

# Digitales Forschungstagebuch

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medieninformatik und Visual Computing**

eingereicht von

**Mark Peneder, BSc**

Matrikelnummer 51822541

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Purgathofer

Wien, 9. März 2023

---

Mark Peneder

---

Peter Purgathofer



# Digital Research Diary

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Media Informatics and Visual Computing**

by

**Mark Peneder, BSc**

Registration Number 51822541

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Purgathofer

Vienna, 9<sup>th</sup> March, 2023

\_\_\_\_\_  
Mark Peneder

\_\_\_\_\_  
Peter Purgathofer



# Erklärung zur Verfassung der Arbeit

Mark Peneder, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 9. März 2023

---

Mark Peneder



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Danksagung

Zunächst möchte ich mich bei all meinen Familienmitgliedern und Angehörigen bedanken, die mir während meiner akademischen Laufbahn geholfen und mich unterstützt haben.

Darüber hinaus möchte ich die kontinuierliche Unterstützung und Hilfe meines Betreuers, Peter Purgathofer, hervorheben. Er hat meine Arbeit von Anfang bis Ende betreut und war immer da, um konstruktives und produktives Feedback zu geben, wenn ich eine Frage hatte oder nicht weiterkam. Auch bei der Suche und Rekrutierung von Teilnehmern für meine BenutzerInnenstudien war er mir eine sehr große Hilfe.

Das bringt mich zu der nächsten Gruppe von Personen, denen ich danken möchte - allen TeilnehmerInnen, die sich die Zeit genommen haben, meine App über einen längeren Zeitraum zu nutzen. Ihr Feedback war äußerst wertvoll, und ich bin sehr dankbar, dass sie sich auch die Zeit für ein längeres Interview genommen haben.

Ein besonderer Dank geht auch an Assistenzprofessorin Dr.in phil. Mag.a phil. Astrid Weiss, die mir ihr Fachwissen über Forschungstagebücher zur Verfügung stellte und mir bei der Gestaltung der daraus resultierenden Anwendung half.

Weiters möchte ich mich bei meiner Freundin für ihre Geduld und ihre unermüdliche Unterstützung während des gesamten Studiums und insbesondere während des Schreibens meiner Arbeit bedanken.

Zu guter Letzt möchte ich mich bei allen LeserInnen dieser Arbeit dafür bedanken, dass Sie sich die Zeit genommen und Interesse an meiner Arbeit gezeigt haben!



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.



# Acknowledgements

First, I want to extend my gratitude to all my family members and loved ones who helped and supported me during my academic journey.

Furthermore, I really want to highlight the continuing support, always valuable insights and remarks, and help of my supervisor, Peter Purgathofer. He advised my thesis from start to end and was always able to give constructive and productive feedback whenever I had a question for him. He was also immensely helpful in finding and recruiting participants for my user studies.

This brings me to the next group of people I want to thank - all the participants who took the time to use my app for an extended period of time. All of their feedback was extremely valuable, and I am very thankful they all took the time for an extended interview as well.

A special thanks also goes out to Assistant Prof. Dr.in phil. Mag.a phil. Astrid Weiss for lending me her expertise in research diaries and helping me design the resulting application.

Additionally, I want to thank my girlfriend for her patience, never-ending support, and motivating efforts throughout my studies, especially while writing my thesis.

Last but not least, I want to thank every reader of this thesis for taking your time and showing interest in my thesis!



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Kurzfassung

Forschungstagebücher werden immer häufiger von Forschern verwendet, die Studien durchführen, die sich auf viele ethnografische Daten stützen, oder in den Sozialwissenschaften. Es ist ein unverzichtbares Hilfsmittel bei der Datenerhebung, insbesondere bei der Feldforschung, aber es ist auch mehr als ein bloßes Protokoll. Es kann ein wichtiges Werkzeug für die Reflexion bei der Forschung darstellen und ermöglicht ehrlich über Erfahrungen zu schreiben.

Trotz dieser Vorteile und der in der Literatur erwähnten Verwendung von Tagebüchern scheint es jedoch nur wenige detaillierte Berichte darüber zu geben, wie sie geschrieben und geführt werden sollten. Auch zu technischen Lösungen ist derzeit wenig Literatur oder Applikationen vorhanden. In Anbetracht der stetig wachsenden Beliebtheit neuer multimodaler Eingabegeräte wie dem iPad, die verschiedene Eingabemodalitäten wie Sprache, Stift oder Tastatur unterstützen wären solche Geräte ideal geeignet für ein digitales Forschungstagebuch. Es gibt zwar viele Tagebuch-Apps, aber keine App, die speziell für Forschungstagebücher entwickelt wurde.

Ziel dieser Masterarbeit war es, eine iPadOS-App zu entwickeln, die das Verwalten, Erstellen und Analysieren eines Forschungstagebuchs unterstützen kann. Der Austausch mit Experten, der Vergleich bestehender Lösungen und eine Literaturrecherche wurden genutzt, um einen umfangreichen Prototyp zu entwerfen und zu implementieren.

Dieser Prototyp wurde dann in einer Nutzerstudie eingesetzt, die aus einer Tagebuchstudie mit anschließenden Interviews, einer SUS-Evaluation und einem zusätzlichen Fragebogen bestand. Die Ergebnisse zeigen, dass Backup- und Synchronisierungsfunktion sowie die Möglichkeit, ihre Notizen in herkömmlichen Dateiformaten wie PDF zu exportieren, von Usern als besonders wichtig eingeschätzt wurden. Darüber hinaus empfanden einige Nutzer den Wechsel zwischen den Eingabemethoden als umständlich, und einige der Funktionen waren schwer zu erlernen für Beginner. Für diejenigen, die von einem traditionellen papierbasierten Tagebuch umsteigen, sollte die App umfangreiche Unterstützung für den Apple Pencil bieten, um den Bedürfnissen dieser Nutzer am besten gerecht zu werden. Für erfahrenere "Power-User" sind Funktionen wie Rich-Editing zur besseren Formatierung potenziell sehr langer Einträge wichtiger. Zusammenfassend lässt sich sagen, dass das iPad und die Forschungstagebuch-App das Potenzial haben, für digitale Forschungstagebücher hilfreich zu sein. Dennoch gibt es Verbesserungspotenzial bei der Unterstützung multimodaler Eingaben und anderer wichtigen Funktionen.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Abstract

Research diaries are becoming increasingly common among researchers conducting studies that rely on a lot of ethnographic data or in social sciences. It is a vital logging device during data-gathering, especially in fieldwork but it is also more than just a logging tool. It can be a crucial reflective research tool, allowing researchers to be honest about their transformation and experiences.

However, despite these benefits and the use of diaries recorded in the literature, there still seem to be few detailed accounts of how they were written and maintained and which format was chosen. Furthermore, as technology continues to evolve and allows researchers to have more and more tools at their disposal, so should research diaries. Especially considering the steady rise in popularity of new multimodal input devices like the iPad, supporting different input modalities like voice, pen, or keyboard. Many general-purpose and special-purpose diary apps exist, but no app is specifically focused and designed for research diaries.

This master thesis aimed to create an iPadOS app that can support managing, creating, and analyzing a research diary based on a user-driven design approach. Exchanges with experts, comparing existing solutions, and a literature review were used to design and implement a feature-rich prototype.

This prototype was then used in a user study consisting of a diary study and follow-up interviews, a SUS evaluation, and an additional questionnaire. The findings indicate that users highly value a backup and sync capability for their notes across multiple devices, as well as the option to export their notes in traditional file formats, such as PDF.

Additionally, some users found switching between input methods often cumbersome, and some of the app's features and iPadOS were challenging to learn. However, for those transitioning from a traditional paper-based diary, the app should have strong support for the Apple Pencil to accommodate the needs of these users best. For more experienced "power users" features such as rich editing to better format potentially very long entries are of more importance. In conclusion, the iPad and the research diary app have the potential to be instrumental for digital research diaries. Still, there is room for improvement in supporting multimodal input and other crucial features regarding the application and iPadOS.



# Contents

|  |             |
|--|-------------|
| <b>Kurzfassung</b>   | <b>xi</b>   |
| <b>Abstract</b>  | <b>xiii</b> |
| <b>Contents</b>  | <b>xv</b>   |
| <b>1 Introduction/Motivation</b>                             | <b>1</b>    |
| 1.1 Motivation and Problem Statement . . . . .               | 1           |
| 1.2 Goals . . . . .  | 2           |
| <b>2 State of the Art</b>                                    | <b>5</b>    |
| 2.1 Research Diaries . . . . .                               | 5           |
| 2.2 Introduction . . . . .                                   | 5           |
| 2.3 Strengths and Weaknesses . . . . .                       | 6           |
| 2.4 Elements and Content . . . . .                           | 8           |
| 2.5 Analysis/ Coding of (Digital) Research Diaries . . . . . | 11          |
| 2.6 Practical Examples of Research Diaries . . . . .         | 15          |
| 2.7 Currently Existing Digital Solutions . . . . .           | 21          |
| 2.8 Diary Apps . . . . .                                     | 21          |
| 2.9 Note-Taking Apps . . . . .                               | 27          |
| 2.10 "Smart Notebooks" . . . . .                             | 33          |
| <b>3 Design</b>  | <b>35</b>   |
| 3.1 Design Considerations . . . . .                          | 35          |
| 3.2 Design Steps . . . . .                                   | 38          |
| 3.3 Outlook/Additional Features . . . . .                    | 47          |
| <b>4 Implementation</b>                                      | <b>51</b>   |
| 4.1 Setup + Software Architecture . . . . .                  | 51          |
| 4.2 Input Modes Implementations . . . . .                    | 56          |
| 4.3 Document Export . . . . .                                | 71          |
| 4.4 Highlighting Feature . . . . .                           | 73          |
| 4.5 Data Persistence and Cloud Storage . . . . .             | 74          |
|  | xv          |

|   |            |
|---|------------|
| <b>5 Evaluation</b>                             | <b>77</b>  |
| 5.1 Selection of Participants . . . . .         | 77         |
| 5.2 Chosen Methods . . . . .                    | 77         |
| 5.3 Planning and Preparation . . . . .          | 79         |
| 5.4 Analysis and Results . . . . .              | 82         |
| <b>6 Conclusion and Discussion</b>              | <b>89</b>  |
| 6.1 Answers to the Research Questions . . . . . | 90         |
| 6.2 Discussion . . . . .                        | 93         |
| <b>7 Outlook and Future Work</b>                | <b>95</b>  |
| 7.1 Additional Features . . . . .               | 95         |
| 7.2 Future Work . . . . .                       | 97         |
| <b>List of Figures</b>                          | <b>99</b>  |
| <b>List of Tables</b>                           | <b>103</b> |
| <b>Bibliography</b>                             | <b>105</b> |



# Introduction/Motivation

## 1.1 Motivation and Problem Statement

Research diaries are becoming increasingly common among researchers conducting studies that rely on a lot of data that is of ethnographic nature or in social sciences [14]. It is a vital logging device during data-gathering, especially in fieldwork. This can include recording contact information of interviewees, key participants, etc. that were interacted with during the fieldwork which can also prove to be important at later stages of research [4]. Another critical element and distinctive feature of research diaries are thorough accounts of the research process, for example, who has been interviewed, when and where the meeting was and who was present. Burgess additionally notes that these records should also try to answer questions regarding the chosen methodology [15]. Such as the circumstances of observations and interviews, the chosen role of the researcher, who was selected for the study, and under which criteria were they chosen [15]?

It can also be argued that research diaries serve more purposes than being a mere logging tool, but are actually an important reflective research tool. They allow researchers to record and retrospectively understand the decisions and methodological tools they have chosen during the research and data gathering process [14]. It also allows researchers to be honest about their own personal transformation and show the "real inner drama" of research. Its primary purpose is not for it to show the research process to others, but to facilitate the research by recording observations, thoughts, and questions as they happen for later use and to reinforce reflective thinking about the research and the researcher's personal development and change of personal views over the course of a research project [37].

However, despite these aforementioned benefits and the use of diaries recorded in the literature, there still seem to be few detailed accounts of how they were written, and

maintained and which format was chosen. This is why this thesis will take a closer look at this still somewhat underrated tool, especially for qualitative research studies - the research diary.

Furthermore, as technology continues to evolve and allows researchers to have more and more tools at their disposal, so should research diaries. Especially considering the steady rise in popularity of new multimodal input devices like the iPad, which incorporates several input modalities like speech, pen or keyboard, and mouse. These devices would allow researchers to note down their thoughts and notes in various ways that best accommodate their needs and preferences [25].

The overall aim of this entire research project will be to develop an iOS app that can support managing a research diary. The optimal design and layout will be based on user input and currently existing diary apps. Some features will be templates for different research diary components, a synopsis functionality, and process support like goal setting, progress tracking, and notifications. Additionally, the project will also entail writing a research diary.

### 1.2 Goals

The aim of this master thesis is to create an iPadOS app, with a primary focus on the Apple iPad, that can support managing, creating, and analyzing a research diary. There are already many general purpose and special purpose diary apps, such as Day One<sup>1</sup>, but currently, there is no app that is specifically focused and designed for research diaries.

As such, the primary focus of the thesis will be to document, design and implement such an application with a user-driven design approach. The aforementioned existing diary solutions will also be compared and evaluated to serve as a starting point for the application to gain a better understanding of what the ideal research diary app should look like. Some of the key features of the app will be the option for different diary templates, a synopsis functionality, progress tracking, and habit reinforcement via goal settings, reminders/notifications, or "streaks".

Additionally, the thesis will also encompass writing a research diary to document the research process, meetings, possible interviews, and focus groups. This diary will serve as both an important tool for documentation and self-reflection during the research period, as well as to get a better understanding of the process of writing such a diary. This personal experience coupled with the knowledge and expertise of researchers experienced with creating research diaries and the existing literature should help create the optimal layout and range of features that should be included in the application.

Finally, the following research questions will be answered:

---

<sup>1</sup><https://dayoneapp.com/>, last accessed on 9<sup>th</sup> March, 2023

- How can research diaries be technically supplemented and augmented by an iPadOS application?
- In what ways can the multimodal interaction possibilities of the iPad, especially voice input and the Apple Pencil, support the creation and management of research diaries?
- How do experienced and inexperienced users of research diaries evaluate current applications that can be used for creating diaries and what features would they want to include in such applications?

In order to answer these questions, the design and implementation of the planned iOS application and comparable solutions will be evaluated via workshops and usability/diary studies. The data and knowledge gained from answering these research questions should facilitate the creation of the resulting research diary application. They should help determine whether users prefer such an electronic solution compared to more traditional or currently existing solutions. Preliminary interviews and exchanges with experts, research into existing solutions, and a literature review should further help with designing and implementing the application and answering the research questions.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# State of the Art

## 2.1 Research Diaries

This section serves as an introduction to the topic of research diaries. Their history as a tool for (social) scientists, as well as the typical elements of a research diary, the implication, benefits for research and analysis, and the mediums by which they can be recorded or supplemented with, will be discussed.

## 2.2 Introduction

Generally speaking, one could define a diary as a "record of the ever-changing present" in which individuals note down their personal experiences, thoughts, and reflections over a period of time [45].

Research diaries have existed in many shapes and forms, whether they were called diaries, logbooks, lab books, or journals, and played an integral part as a memory device to record fieldwork in many research fields [4]. Especially qualitative social research has utilized this device quite intensely to record participant observations and key interviews. But there is more to research diaries than just recording observations and collecting data. Its introspective nature allows researchers to increase self-understanding, eliminate, or at least become aware of self-delusions, and document, rationalize, and reduce pain and frustrations during the research process. These honest conversations with oneself in a diary be an eye-opening "dialogue with a cruel partner" [4].

Research diaries can also help with one integral part of the research - writing up the actual research [37]. This process of collecting all the data gathered throughout the research period and writing the conclusions drawn from it and what work was done is part of the routine of most researchers. This usually includes many forms of written notes and gathered data, like reports on academic papers or formal research. However,

writing in the form of research diaries has still received comparatively small popularity, despite how useful it can be. Especially in studies that are of qualitative nature, require a lot of fieldwork, or in many social sciences.

As Engin points out, the diary or journal has long since been known as a useful tool for learning a language or to support the teacher learning process [22]. They offer the opportunity for teachers to reflect and introspect on their teaching, thus helping to change beliefs and teaching practices, which in turn can amplify professional development. This self-dialogue helps teachers externalize their knowledge and notions of teaching by writing it out and then internalize them again by reflecting on them simultaneously. But, while the importance of diaries in teacher learning is already well-documented, the same cannot be said about the importance of research diaries and their support in researcher learning. Engin suggests that research diaries can be as integral for researcher knowledge development as diaries are for the development of teachers[22].

Altrichter et al. also mention that a research diary is also vital in its role to document both successes and failures during the path of discovery and learning in research [3]. Especially the latter is an often neglected part, although it might prove instrumental later for further analysis and retrospection. Furthermore, they argue that this documentation of the analysis and reflection should not be separated from the data gathering, as they both reinforce each other. Reflection (and analysis) is an essential part of research diaries, and something we will discuss in more detail in an upcoming section.

### 2.3 Strengths and Weaknesses

The main advantage of using such a time-intensive method like research diaries or journals to document the research process is the opportunity to reflect on oneself as a researcher and the words and beliefs described in the diary [30]. This reflection of collected data, such as interview transcripts, in combination with the documented thoughts and interpretations, allows one to rethink and analyze the decisions taken in a chronological manner as well as observe ones growth as a researcher. Furthermore, it allows checking on one's own interpretations and is almost like a "dialogue" with oneself on paper.

One of the biggest weaknesses of research diaries is their time-consuming nature [45]. It takes time to get used to writing one and choosing the best format that fits best personally. This is further hindered by the still somewhat limited amount of recommendations and accounts on how to actually establish and keep a research diary [14]. It is also vital for research diaries to be written as honestly as possible, to keep both unconscious and conscious editing of entries to a minimum if researchers want to draw the full potential of this tool. This honest account of the process is important and should not be viewed as something that will be shared with others or publicized [45]. It should also be mentioned that there might be some ethical concerns, as diaries cannot be made public that easily since they contain usually quite private accounts and additionally possibly sensible interview or observation data. If research diaries should contain such interview

or observation data from someone else, Altrichter et al. recommend immediately clearing this data with the concerned person [4].

On a more general level, diaries of any kind can be very helpful to strengthen gained knowledge, writing skills, learning, and student reflection. Bontuyan et al. analyzed multiple studies and papers concerning (academic) diaries and their impact on the student's writing and communication skills, among others [11]. For example, Clipa et al. documented how diaries can help with the development and growth of metacognitive strategies[16]. Essentially, the diary teaches students how to *think* about their *thinking* and better understand how they actually learn.

They mention how it can be useful during initial university training, which brings us to the next point. A lot of these aforementioned points are also true for novice researchers.

In a learning environment, such as universities or faculties and even high schools, research diaries or journals could serve as a "low stakes writing assignment", as demonstrated by Vinjamuri et al. [47]. Since it is not as constrained by the strict rules of formal writing, students and (novice) researchers can express themselves more freely, without needing to worry about these rules, and can also write out their own feelings and anxieties. This can also serve as a dialogue between themselves, as well as their work, and their teachers. By taking control and "ownership" of their learning, by voicing the decisions they have taken and challenges faced during research, they could gain a better understanding and connection to their own process, as opposed to just using didactic teaching methods alone.

This requires of course a non-judgmental and open environment between teachers and students, as they are laying their feelings bare than they might be comfortable and used to doing so. Additionally, another drawback of such a teaching method and assignment would be the time-consuming factor for teachers. Reading, judging, and giving timely, constructive feedback is essential for this to work and to further the student/teacher dialog - and, sadly, it is not always easy to make time and energy for such assignments.

Newbury mentions several practical issues and considerations that need to be taken when planning and writing a research diary[37].

Firstly, the medium of the diary has to be carefully considered and might be hard and/or cumbersome to change later. A stack of loose papers might make it easy to add printed images and notes in between, as well as being easy to scan and copy for later use, however single notes can be easily lost or brought into disarray. A notebook is more compact and easier to structure well, however adding additional notes or digitizing it, might be more difficult. A digital research diary could help with that, but it requires to be always kept nearby and cannot be as easily replaced and data security plus backups are an important issue - as we will discuss in later chapters.

It might also tend to happen that authors change their structure, style of note-taking, writing, and contents over the course of creating their research diary, which might make writing up the research project/paper more difficult in the end. Additionally, without

a monthly or weekly summary, or some sort of index system, it might be hard to get information or certain passages from a diary at a later point.

Some of these aforementioned issues could potentially be ameliorated by using a digital research diary, as we will discuss and analyze over the course of this thesis.

### 2.4 Elements and Content

Research diaries can contain a great array of different elements and artifacts [4]. These can include:

- Data gained during interviews, observations, and other fieldwork
- Supplementary gathered artifacts, like photographs, letters, and - for digital diaries - videos and other recordings
- Contextual information such as the place of meetings, who was present, etc.
- Methodological steps and decision and further plans for actions
- Personal reflections and reflections on the chosen methods
- A record of readings, including comments, summaries, and quotes
- Certain keywords and other indexing methods that might help navigate the diary

Altrichter et al. categorize entries and content of a research diary roughly into the following categories [4]:

#### 2.4.1 Memos

*Memos* are all written accounts of an experienced event that happened over a specific time period, such as an interview, a study session, or a lesson. This is often the only or best way to document a continuously changing activity with a steady stream of new information.

Bogdan and Biklan recommend a few tips and habits for writing memos [10]. Firstly, memos should be ideally written as soon as the event happens, if possible. Secondly, it might be good to not immediately talk about an event before writing it down, to not cloud or disrupt your own memory about the event. Thirdly, it is best to detail events in the same chronological order as they happened, but if you remember a crucial detail at a later point, it is better to write it down at any point. Fourthly, it is always good to write keywords and catchphrases during an activity, which can help as a reminder for the creation of the memo.

Altrichter et al. suggest differentiating the content of memos between *descriptive sequences* and *interpretative sequences*[4].



## Descriptive Sequences

**Descriptive sequences** are detailed recounts during memos of events, interviews, and dialogues - including the tone, gestures, expressions, etc., of participants. It should be noted that these details of events should be more important than general descriptions, and the narration of the event is more valuable than an evaluation or interpretation. Quotations are essential additions and should be kept as thorough and true to the original account as possible. When you are a participant or subject of the research yourself, you should also note down your own behavior and feelings during these activities.

## Interpretative Sequences

**Interpretative sequences** of memos are all parts of a memo that contain recounts of personal feelings, emotions, reflections, challenges encountered, notions, future plans, discussions of the theoretical approach, and much more. All of these sequences are completely valid to be in a research diary. They can turn it from a mere description of events and logging device into a personal companion that helps you grow as a researcher and reflect on your work.

These interpretative parts both happen as you write about an event or experience, as well as later when you are reflecting on it. As you reread your writing - which should be done often to better reflect on the detailed events - you might be able to connect the dots between feelings and ideas you might not have found the first time around. It also helps to "separate the wheat from the chaff" and figure out the more insightful sequences, which can lead to the best next steps and ideas that need follow-up.

**Interpretative sequences** can be further broken up into three subcategories: *planning notes*, *theoretical notes*, and *methodological notes*.

## Planning Notes

**Planning notes** are interpretative sequences that can help transform the research diary into a real travel companion that keeps a record of all your planned next steps, things you want to improve in your next interview, for example, or alternative steps that could be taken. These things are often forgotten if not noted down immediately and additionally serve as documentation of how much a researcher has improved in their thinking and approaches over the course of a project.

These **planning notes** are handy and common when the research subject entails practical actions and experiences, such as during field research.

## Theoretical Notes

**Theoretical notes** happen as you try to connect the dots between gathered data and artifacts to gain an understanding of the deeper picture. As you reflect on the presented data, the ideas, and insights written down in the memo help you immortalize these

connections and thoughts. These can help you gain answers and explanations to your research questions further down the line and establish connections between points in time that might need further investigation.

**Theoretical notes** can be handy in many ways. They help create new theories or clarify an existing idea or theory. Additionally, they can help establish connections between pieces of information or connect the information and lived experiences with an existing theory. They also help find confounding circumstances that need a follow-up investigation.

### Methodological Notes

As the name implies, **methodological notes** concern themselves with the methodologies used during research and the general research strategy. It is vital to always reflect and fairly critique the current methodological strategy to grow as a researcher and improve the standard of the research project. This also helps to gain new ideas and insights for alternative methods that could be used instead.

They help in analyzing and answering tough questions about your own research. These could be - among others [4]:

- Why did I use this particular research method in this situation? Do I have certain biases attached to them?
- Did I face any ethical dilemmas and how was I able to solve or at least ameliorate them?
- What was my own role in a certain situation? Did I bring my own prejudices into it?
- Why did I make certain choices regarding the future of my research?

### 2.4.2 General Writing Tips and Suggestions

Evidently, diaries are a melting pot of all the different ingredients of a research project, with a varying degree of "quality", as they can include personal thoughts and reflections, notes as well as detailed analysis of e.g. interviews. This can be confusing and seemingly messy for some, especially when starting out with writing a research diary but it is also one of its biggest strengths and distinctive qualities.

These diverse accounts, thoughts, and reflections on the research process all have their place in a research diary. This also makes it an excellent "travel companion" during fieldwork, as it can document every part of the "journey" and development of research and a chronological collection of every part of it [14]. It also offers a continuous and steady analysis of ongoing research. This can also lead researchers to find currently lacking parts in both the data and the used theoretical framework. Additionally, it can also be used to document an often neglected part of research - the fears, concerns, and

self-doubts of the researcher's own work.

Writing any diary - research diaries are no exception - is always a personal matter and there is no strict formula that one should adhere to [14]. However, even in the still somewhat limited literature for research diaries, researchers can follow certain guidelines and tips.

Altrichter et al. mention a difficult period at the beginning of writing a diary before it becomes a fulfilling experience and recommend writing as regularly as possible and discussing extracts one feels confident to share with collaborators and research partners [3]. Of course, this does not mean that this should take away from the private nature of the diary, as "self-censorship" might diminish its value and flow of thoughts. Furthermore, the text should be structured with paragraphs, headings, etc. to help with later analysis. It should also be made apparent in the text - if possible, as the distinction is often quite fuzzy - which part of it is observation and which is interpretation. Furthermore, a preliminary analysis of the entries might help define this distinction between observation and interpretation and facilitate planning the next research steps.

Browne further recommends to re-read the diary at multiple points to allocate codes to text segments and organize these coded segments into overarching themes [14]. He also recommends using it as an additional scheduling device to keep track of key contacts and if there are any frustrations with fieldwork or unresponsive contacts to write them down and rationalize them for further plans of action.

## 2.5 Analysis/ Coding of (Digital) Research Diaries

In this section, we will look at ways to analyze and codify a research diary. This is a vital step to use a research diary to its fullest potential and in order to transform the data and insights contained in it into new theories, research categories, and much more.

### 2.5.1 Introduction

Research diaries are inherently a process that produces a lot of mostly qualitative data, which might amount to quite an overwhelming amount of raw data if the research/fieldwork has been conducted over a long period of time.

This is where coding as part of the analysis can come in handy, especially at the end of a project where all the entries in a diary should prove quite useful in creating the write-up for a thesis, paper, or similar. As researchers, one of our main tasks is to observe and interact with the world around us, trying to make sense of the created data and present them in a methodically and logically sound way [18]. While a research diary is often concerned with the first part, the second part also often plays a role depending on the type of memo, as we learned earlier in section 2.4.1 with interpretative sequences that are already trying to interpret the recorded and observed data.

Coding is a great way to help reduce the amount of data by breaking it down into specific junks, according to your own judgment, e.g. by certain emerging or presupposed topics and themes. Breaking down data this way is also a great help in organizing and structuring data to make further analysis and searching for specific entries easier. As Cope points out, data reduction via coding is a handy method to deal with and organize the "chaos of brute observations" [18].

One simple, but essential approach towards not only coding but better understanding the content of your research diary, is to often re-read it to start looking for emerging trends, as Browne and Engin noted [14, 22].

What to code and analyze in your research diary is generally up to you and according to what belief you subscribe to. Some researchers are of the opinion that every part of data, which might include images, descriptions of your day, and other minute details are worthy of consideration [32]. Others suggest that only the most noteworthy recorded data should be considered and even suggest that around 50 percent of data can be ignored, deleted, or summarized before the analysis [43]. However, Saldaña suggests that especially when starting out in qualitative analysis, you should consider and code all of the data as you might not even know what is really important in the beginning or overlook emerging patterns otherwise [42]. This point might be especially true for research diaries, as we will see in the next section, where even personal experiences and feelings can become an essential part of your research and write-up.

Coding will also play a role in the design and feature set of the created application, since this process hugely benefits from technological support to automate the grouping and overview of various different categories found during coding, adding tags and categories to entries, among other things. This is also why many software solutions and suites have emerged that are specifically designed just for this purpose and can be used for all kinds of different quantitative data. This so-called *Computer-Aided Qualitative Data Analysis Software (CAQDAS)* exist in many feature ranges, price points, and levels of detail. However, for the context of this research, such tools were not the focal point and might be deemed too sophisticated for this project's purpose and goals.

Saldaña describes a code as a short word or phrase which describes the essence of a part of a qualitative datum that is being observed, such as a sentence of an interview transcript, or - like in our case - a passage from a memo in a research diary [42].

Cope further differentiates between two main categories of codes, descriptive and analytic codes [18]. There are also other distinctions, such as differentiating between first and second-cycle coding methods [42].

### **Descriptive Codes**

Descriptive codes might describe or point out more surface-level insights that were either in verbatim stated by research subjects or could be easily observed without deep analysis or investigation. This can often be boiled down to simple "who did what where, when,

and how" inquiries. For example, demographic or location information, or a very topical summary of the data at hand - e.g. when analyzing an interview about safety in a dangerous neighborhood, a descriptive code might be "crime".

For Cope, in vivo codes are a specific subset of descriptive coding, while others, such as Saldanã would view it as a category of their own [42]. A in vivo code is taken directly from a subject's own words to avoid adding your own perception or biases and rather stay as close to the original meaning as possible.

### Analytic Codes

Analytic codes usually cut more under the surface of a research topic and already show an emerging pattern or topic that the researcher has noticed over the course of analysis. This can be an already established pre-existing theory, e.g. based on a research question, or often it is based on a connection made during the analysis. Saldanã would group these codes under second-cycle codes[42].

#### 2.5.2 From Codes to Categories to Themes

Since these pattern or analytic codes can also be seen as a grouping and abstraction of the more straightforward descriptive codes, they are also the primary tool that helps us to move from descriptive and analytic codes to categories.

A category is a further grouping of codes to an overarching topic or a common idea. The way these categories come to be is entirely up to your own intuition and decision what you deem as common property and if they "feel similar".

Keep in mind that coding is also an exploratory, "cyclical" task. As you are re-codifying a text for the second, third, or even fourth time, you might notice new codes, restructure existing codes or new categories that point to a newly found overarching theme [1]. After identifying, re-identifying, moving, adding, and removing these codes and categories over multiple cycles, you might move on to the next step, which is identifying the existing and found themes in your data.

As you look at these categories, you should be able to find both similarities and differences between them that allow you to merge and compartmentalize them. By doing so, you should be able to see beyond the "reality" of your collected memos and data and arrive at the concepts, abstract and pure nature of it, i.e. the concepts/themes. Creating one or more themes or concepts out of your found (major) categories helps you to get one step closer to a working theory that will (ideally) help you with your write-up and answer the research questions at hand. A simplified and general tactic for getting from codes to theory is illustrated in figure 2.1.

In a way categories and themes can be seen as higher levels of codes, as according to Elliott [21].

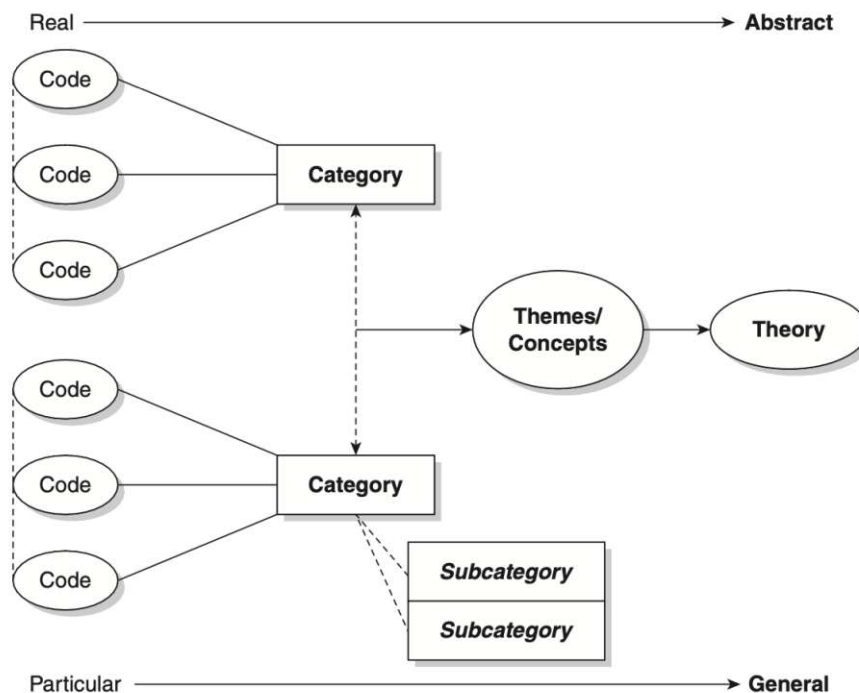


Figure 2.1: A simplified illustration of the codes-to-theory approach described by Saldaña [42].

### 2.5.3 Coding Approaches

Generally, it is up to the researcher to choose more of a top-down (deductive) approach or a ground-up (inductive) approach before starting to code their qualitative data. It might also be valid to combine these two methods, if appropriate for your data.

#### Deductive Approach

In a deductive/a priori approach, a researcher starts out with a preset set of codes that are used to analyze the qualitative data at hand. These codes can be based on your research questions, an existing theory, or a mixture of both. This might be also useful when trying to prove or test a presupposed/existing theory against empirical data at hand or testing data against an existing framework in general.

This approach is often used when dealing with large projects that might also require more than one coder, where a mutually agreed upon set of codes is important to not *muddy* the outcome with two wholly different approaches in analyzing the data [21].

As mentioned, even when starting out with a deductive approach and presupposed coding schemes, it is highly advisable to stay open-minded for new codes and patterns that emerge during the coding process [21].

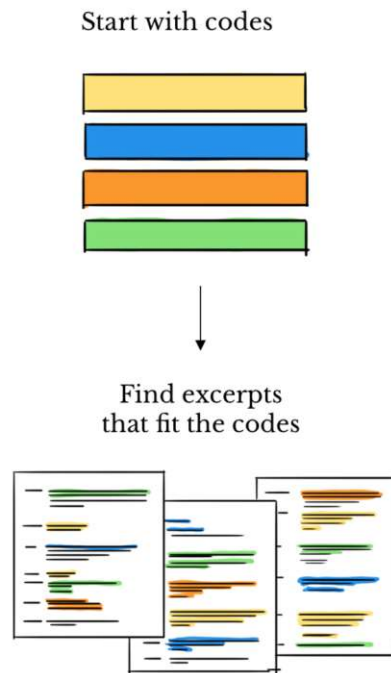


Figure 2.2: A deductive approach would start out with certain codes and try to find parts of the data that fit them<sup>1</sup>

### Inductive Approach

An inductive/emergent approach would try to start from the "ground up" with no existing set of codes beforehand and mostly let the data "speak for itself". This is useful to not include your own existing biases, theories, or beliefs into the initial coding but rather come up with and explore new ideas and theories[20]. Therefore, this approach is very useful when you are doing more exploratory research and you are trying to find new frameworks, and ideas and puts the voice and views of participants into focus [21].

An inductive/emergent code might be words or a phrase of a participant, or concepts/phrases that you have been made aware of or often encountered during your research for a particular project.

## 2.6 Practical Examples of Research Diaries

In this section, we are going to look at some practical examples and excerpts from research diaries. These should give you get a better idea of what an actual entry to a

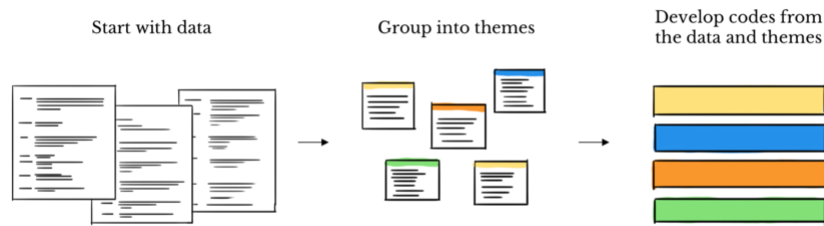


Figure 2.3: An inductive approach derives codes and themes directly from the data at hand<sup>2</sup>

research diary could look like and how they might be able to use and incorporate a diary into their research project. However, as mentioned before, there are no strict rules and guidelines as to what the content and entries of a research diary have to look like and contain.

### 2.6.1 Research Diary: A Tool for Scaffolding

In her article, Engin discussed the role and importance of a research diary to help facilitate learning about research and improving as a researcher [22]. She specifically used a research diary she wrote during her doctoral studies as an example of how a diary can serve as an "expert other" to help scaffold research knowledge.

Scaffolding can be described as the support given by a more experienced peer or teacher in an educational setting [34]. Bruner has defined scaffolding as the actions and support given that help reduce the degree of freedom in executing a task so that a child can entirely focus on the difficult skill he/she is in the process of learning.

Engin thinks, that in (creative) writing this expert other can be the writer him/herself. For researchers, who are often working alone, maybe even far away from their university and/or supervisor, she also believes that a diary can be a companion. It is not part of an assignment and is only a private repository of data and decisions made, as well as reflections on them - in short, all kinds of thought processes that happened during research. She used the research diary that she created alongside a log during her doctoral studies as an example. The research diary transformed more and more into an emotional support as well [12], in addition to being a documentation of the thoughts and talks with the "expert self" during research. The analysis of the research diary was done as she would be analyzing any transcript of an interview. Based on Miles and Richards [35, 39], she analyzed, read, and coded the diary based on viewing it as a scaffolding tool. She read through the diary multiple times, on the lookout for certain codes and emerging broad categories that described how the diary helped her in gathering her thoughts about her research process and her reflections on them. The emerging categories were: questions to



self, reference to "expert other", noticing differences, and justifications for decisions and actions.

We are now going to look at some of these discussions and excerpts, to get a better understanding of what an entry in a research diary could look like.

### Questions to self

Questions to self can be seen as proof of how one might be able to grow as a researcher by asking the right questions and reflecting on them. They also show how you are thinking and not just noting something down and considering other options. They also do not necessarily need answers right away.

Such questions could be about the insecurities during the formulation and choice of the research methods, as in Engin's case. This question shows the self-doubt of a novice researcher as they are struggling with new terms and the literature, but also illustrates the growth and first steps toward problem-solving:

“Reading for methods chapter but getting bogged down on terms – ethnographic what? Also, getting a bit worried about what mine is – seems to be an ethnographic study of myself as a trainer!” (Research Diary, 22.9.08)[22].

### Justifications for decisions made

Justifying actions and decisions show that the researcher has made informed decisions and that they can clearly argue for their taken route, as well as articulating considered options. This can help with confidence in your own research and “believing” more in them. Furthermore, it can also help us in clarifying and understanding our own position by actually writing them down and manifesting ideas and thoughts.

### Noticing

Noticing certain themes in your behavior, reflection is important and can signify growth and knowledge creation. This noticing can happen as you analyze and code the diary or while writing down your behavior and actions, as you verbalize them. One case of noticing occurred as Engin was transcribing an interview and serves as a good example of noticing themes as they happen and how important is to note them down [22]:

I'm transcribing the tapes and listening to everything carefully, writing it all down and noticing how it is going. I'm really listening carefully to my questions, but I'm starting to think it is not the questions, but the format, and structure of the feedback. (RD, 13.11.08)[22]

### Conversation with expert “other”

As initially mentioned, in a narrative sense the conversation with an expert can be "yourself", or with the expert “other”. It can be literature that you are reading and interacting with. It can scaffold your own knowledge and serve as a basis to discuss ideas, and inform your decisions about the framework and used methodologies. They can also justify and give "validity" to your own thoughts. Engin adds an example of how reading and articulating thoughts about literature can scaffold knowledge [22]:

Read some interesting stuff about activity theory and at the first level of analysis is goals. Now I am sure all my students have different goals, which means they are all actually involved in different activities. I should find out more about their motivations. (RD, 11.1.09)[22]

### 2.6.2 Recording the Personal: The Benefits in Maintaining Research Diaries for Documenting the Emotional and Practical Challenges of Fieldwork in Unfamiliar Settings

In this section, we will look at reflections and excerpts from Browne’s own research diary during fieldwork [14]. Like Engis, he re-read the diary at multiple points, and assigned codes to similar sections, per the guidance of [12]. These codes lead to four different broader categories: practical issues faced during data collection, dangerous fieldwork, methodological challenges, and personal challenges.

#### Practical issues

The research diary has proved to be a vital companion to record, reflect, but also vent on various practical issues that the author encountered during his (often dangerous) fieldwork in Ramallah. This can not only lead to finding better solutions for future similar issues but also to record important scheduling and contact information. Additionally, talking about the emotional duress and challenges that such fieldwork or research in general entails is an often overlooked part of research (diaries), as we discussed in earlier sections. By venting in a diary, we might also be able to use the negative energy in a positive way, by channeling them in a productive manner and rationalizing the problems and finding solutions. One such example is an entry that talks both about the difficulties of working and moving in a region plagued by conflict as well as the more common problem of having an important contact that cancels often. In the end, he was able to rationalize the benefits of talking with the contact and had a valuable conversation with him - something that might not have happened without such management of frustrations.

I was up early this morning making my way to Nablus to meet with (N), this involved catching the bus to Ramallah (1-2 hours) and then waiting for the bus to take me on to Nablus. In total, I would be spending in excess of 6 hours traveling to and from Nablus to meet with him. As I had just

pulled out of Ramallah (N) texted me to say that he could no longer make the meeting, this is the 5th time he has canceled on me and I am beginning to doubt if this meeting will ever take place. His would be a great narrative to have and his views would be really welcome but he constantly cancels. I'm getting more and more frustrated with this and am really at odds as to whether to rearrange. (research diary, Jerusalem, February 26, 2011) [14]

### Dangerous Fieldwork

Since the fieldwork was conducted in quite a dangerous region, known for its conflicts and instability, a large part of the diary was concerned with and served as documentation of working in such an environment. So for Browne, it was important to capture these emotions and experiences so that they can also be used for (novice) researchers planning to work in similar environments. This illustrates how a research diary can serve as a sincere document of the emotional and personal challenges that the researcher has encountered - an often overlooked and underdocumented side of research. These accounts are both important documents of challenges but can also help face or at least alleviate these feelings [17]. By documenting these anxieties and dangerous situations, he also created an insightful chronology of the unfolding events. One such account seeks out to deal with the stress, panic, and guilt after a bomb explosion in Jerusalem:

Practically speaking, how is one supposed to manage personal risk in a place like this? Today I got scared for the first time since I have been here, a bomb exploded outside the crowded central bus station, West Jerusalem, where I had bought my ticket to Eilat only a week or so earlier. There seem to be 33 people injured some seriously and it looks like a few may have died as a result of the blast. This is not a great omen and has made me think a lot about why I am here. For the first time I am thinking critically about personal safety. I am beginning to wonder what the hell is the point in getting caught up in somebody else's problems? . . . . (research diary, Jerusalem, February 26, 2011) [14]

### Methodological Challenges

As discussed in section 2.4.1, methodological challenges encountered during the research should be an important part of a research diary, according to Altrichter [4]. Browne had a similar experience in his fieldwork. Memos of methodological challenges were potentially his most important notes when it came to the discussion and critique of his methods when writing up his research. His were mostly practical, such as critiquing his interview style, recruiting participants, or getting participants to trust him as an outsider. The latter was illustrated in the following entry:

The meeting with (I) lasted a lot longer than I had thought it would (50 minutes). I really didn't seem to have her full trust or enthusiasm though.

This is something I have noticed so far especially when looking at something as sensitive as the Nakba. The trust just isn't automatically given to you. I'm trying desperately to work on this but it is perhaps inevitable that I will face this issue as an outsider here. (research diary, Jerusalem, March 8, 2011)[14]

### Personal Challenges

As previously discussed, documenting personal challenges can and should be a part of a research diary, especially when working far from home and your loved ones. As Borg has mentioned, we rarely talk about the emotional part of conducting research, an error that often leads to the misconception that emotions should be removed from it and these emotions should be dismissed and hidden [Borg2001]. Burgess has listed numerous doubts and feelings of distress, both regarding his personal life, as well as anxiety about the research and the validity of his gathered data. However, noting down only negative emotions might not always be a good idea, so you should also include positive and self-affirming motivational entries. The next two entries serve as examples of a "positive" and a "negative" memo:

Doubts over the work have made me really miss home today. I was feeling quite homesick today for the first time in a while. I get like that when I don't leave this place. I know I have interviewed some really high profile figures but I am concerned that I have not interviewed enough. There's not much I can do though to combat these worries. (research diary, Jerusalem, April 11, 2011) [14]

My head feels good and clear. I admit that there is always more that can or could have been done at this stage, but I am feeling confident. I am learning significantly from being in the field and I could not have accessed all these groups had I been behind a desk in Belfast. Being immersed in the field for such a period of time, and during such a tumultuous period has really helped me. (research diary, Ramallah, May 24, 2011) [14]

### 2.6.3 Conclusion

This section should have given you a better idea of what an entry in a research diary could look like and maybe it helped you in seeing the benefit of keeping such a diary for your own research. From documenting personal struggles to questioning your methodological approach, everything can and should be part of your diary, and as you can see it can be really useful in recognizing emerging topics in your research that you should pay more attention to and Additionally, as Browne demonstrated, especially for fieldwork - but not limited to - it can prove to be a personal, supportive companion.

## 2.7 Currently Existing Digital Solutions

There are already many currently existing and maintained applications for the iPad that can be generally used as a digital research diary, even if that is not their intended primary use case. Depending on the user, one might prefer one over the other depending on the offered feature set, approach to diary/note-taking, pencil, and multimedia support, and much more. This section is not to be seen as a definitive recommendation or overview of the currently existing solutions for digital (research) diaries. As discussed in the previous chapter, a research diary is a personal research method, and there are no definitive rules. Therefore, there is no definitive, objective "best" diary or note-taking app. The most important metric should be that it works for you personally, is delightful for you, and invites you to use it as often as possible.

In the following section, we are going to look at different apps that at least partially fulfill the feature requirements one would have for a digital research diary - these are of course different for every potential author of such a diary. Some features that I used as selection criteria were:

- Multi-modal input
- Media support
- Meta-Data
- Chronological support
- The number of downloads / the popularity ranking in their respective category on the Apple App Store

The apps were loosely grouped into apps that call themselves "diary" apps and more general-purpose note-taking apps. The list is not entirely exhaustive, and in the vast landscape of the App Store, it is easy to miss an app. Most of the apps in the coming sections influence the final design in one way or another.

## 2.8 Diary Apps

There are already existing solutions to write diaries/journals on both iOS and Android devices which are primarily designed for a general-purpose diary but not specifically for research diaries. Some of the currently most popular applications are **GridDiary**<sup>3</sup>, **Day-one**<sup>4</sup> and **Journey**<sup>5</sup>. Some of their features, such as different templates, daily reminders and streaks, and the option to add contextual information, will also be included in some shape in the resulting application, but with a more clear focus on strictly research diaries.

<sup>3</sup><https://griddiaryapp.com/>, last accessed on 9<sup>th</sup> March, 2023

<sup>4</sup><https://dayoneapp.com/>, last accessed on 9<sup>th</sup> March, 2023

<sup>5</sup><https://journey.cloud/>, last accessed on 9<sup>th</sup> March, 2023

There are also more specific solutions for other research methods that use diaries, specifically diary studies. Examples include mobile diary study solutions from *indeemo*<sup>6</sup> and *opiria*<sup>7</sup>. However, diary studies and research diaries might have some overlaps but they are two very different tools and research methods and are usually used for very different purposes. Diary studies are especially important to let multiple participants self-report qualitative data over a certain amount of time and are used for many different fields, such as User Experience Research [23]. Some of the required functionality of such solutions can be comparable. Still, they are more focused on documenting, designing, and analyzing multiple diaries at once, focusing on User Experience research for brands and designers.

### 2.8.1 Grid Diary

Grid Diary<sup>8</sup> has an exciting approach towards journaling/diary writing. It uses a titular grid format that users can customize to always start a new entry with prompts that they selected or chose from the predefined templates. The intended purpose is not to have to "stare and be intimidated by a blank page" but rather to start out with a structured approach and reflective prompts that make a habit of keeping a (research) diary easier for beginners.

Users can also create multiple journals which open in separate views if they want to have one dedicated for e.g. work, goal tracking, or weekly planning. The overview page also consists of a sidebar containing multiple tabs that show different overviews of the created data, such as a timeline, a calendar view, and an explore page. On the explore page, users can customize templates, see all their added media, and see a statistical report of their journals. These stats include the best streaks, the number of characters, trends, and other data points. Journals are always opened in a fullscreen view and editing an entry also opens another one in turn.

Each entry can also be enriched by meta-data, such as the current mood when writing the entry or a sticker to e.g. show the current weather. Attachments, such as pictures can also be added to entries, although they can not be resized and easily be moved around. It also supports rich editing functions.

This approach can also be handy for a research diary to answer essential questions about what you have done and achieved in a day. One could, for example, structure it to answer the questions provided by Altrichter [4] or more general ones, such as "What have I learned today?".

Like most popular diary apps that I looked at, it features a *freemium* subscription model, that requires users to pay monthly or yearly to unlock all of its features, such as unlimited journals, exporting PDFs, cloud backup, or markdown tools.

---

<sup>6</sup><https://indeemo.com/mobile-diary-study>, last accessed on 9<sup>th</sup> March, 2023

<sup>7</sup><https://www.opiria.com/mobilediary/>, last accessed on 9<sup>th</sup> March, 2023

<sup>8</sup><https://griddiaryapp.com/>, last accessed on 9<sup>th</sup> March, 2023

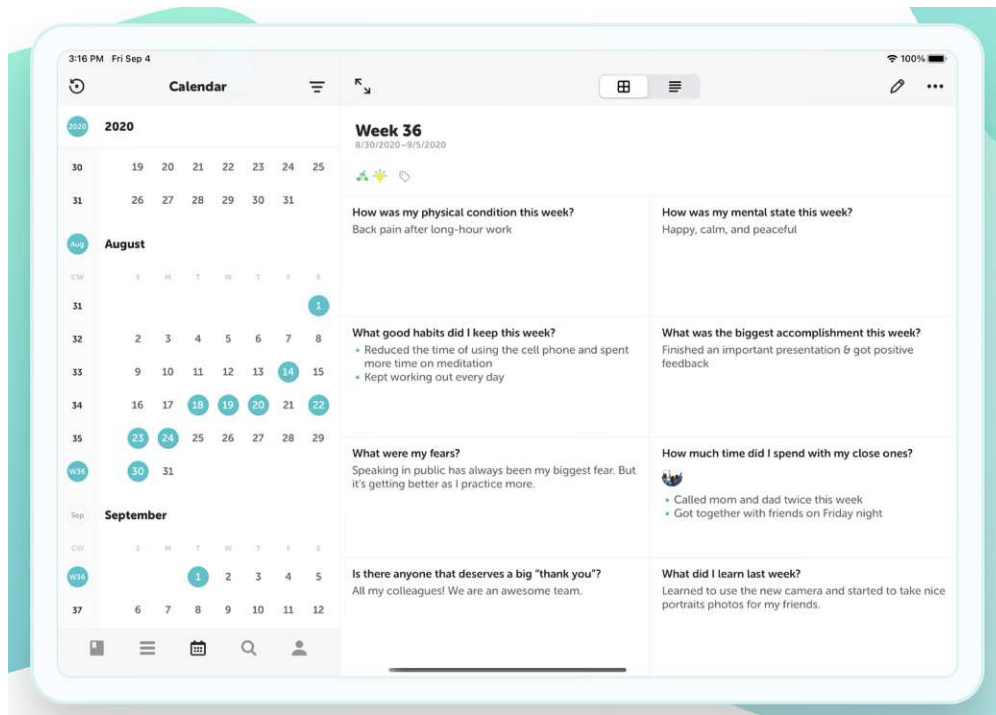


Figure 2.4: Grid Diary offers different overviews via the tab bar on the left, such as the calendar view, as well as a unique grid-based layout with custom prompts for each new entry to help user guide their thoughts<sup>9</sup>.

As mentioned, the grid format is a good way to keep structure in a research diary, especially for beginners or users that often suffer from "writer's block" and the habit tracking plus statistics can be pretty encouraging and insightful. However, it can also be too frigid in its approach for some and may lack proper pencil/handwriting support outside of adding sketches to entries. Additionally, some of the mentioned premium features are almost a requirement to actually use the app daily as a diary.

### 2.8.2 Journey

Journey is a popular digital journaling app for iPad, iOS, web, and macOS. It serves as a good entry app for beginners at journaling, with a particular focus on personal development and well-being. Like most competitors it offers cloud syncing across devices.

Similar to Grid Diary, entries can be organized via tags and viewed in a chronological overview or calendar view. An interesting addition is the "Atlas" view where entries are displayed on a map view based on their geotag. As a whole, the app is quite simple and straightforward to navigate and use.

Since Journey advertises itself as a self-care journal, it offers several ways of tracking (mental) health data, such as the mood while composing an entry or even activities done

## 2. STATE OF THE ART

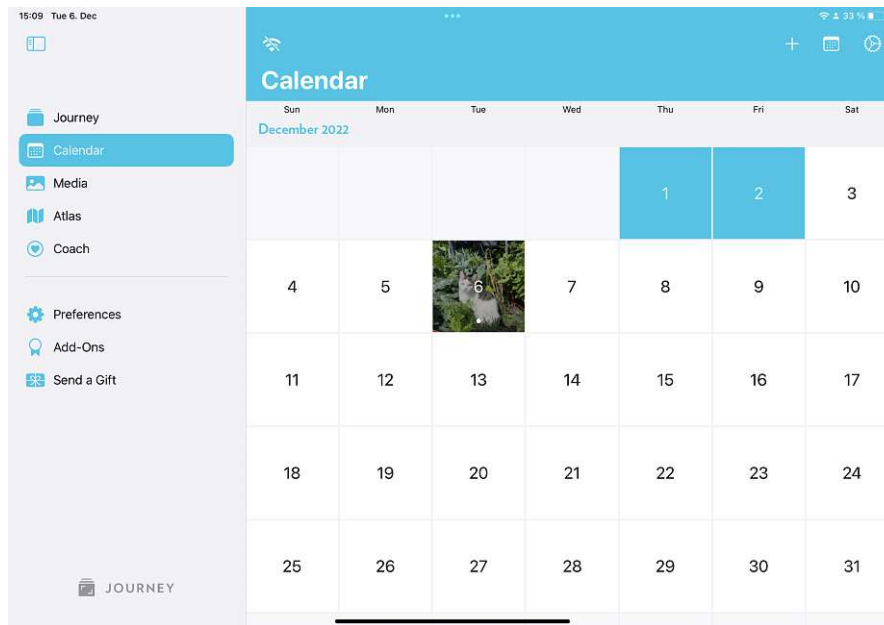


Figure 2.5: Users can switch between different ways to view their entries, one of them being a calendar overview that shows users on which day they created an entry and, if applicable, the attachments they added to an entry.

on that day, including a possible sync to Apple Health data. There is also a "coach" section that provides users with guided prompts and articles teaching self-care on various topics. Creating a new entry is based on a "no-frills" approach. While some rich editing options are possible, there are no custom layout options and images/drawings can only be added to an attachment screen, with no option to change a sketch after it has been added. Users also get a statistical overview of their mood, activities, and other data, as well as "throwbacks" to entries and moments from the past to relive them again.

Journey also operates under a freemium model that disables many features for non-paying users, such as the option to create more than one journal or limiting the number of photos users can add to a journal.

As with the other diary apps that were tested, apple pencil support is a very secondary feature, and other media and input support is as well. Organizing is also not as easy as in other apps, and you cannot organize it into different folders. However, the mood and activity tracking plus the focus on mental health is a great addition, but in general, there might be better applications for research diaries in particular. The design is pleasing, albeit a bit simple, and almost "too clean".



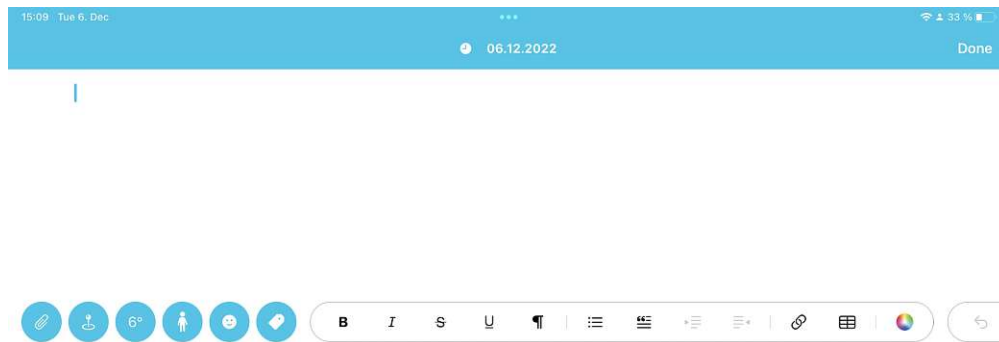


Figure 2.6: Journey’s editor is relatively straightforward with no options for custom templates. Additional meta-data and markdown editor options can be added via buttons on the bottom of the screen, always resting above the onscreen keyboard.

### 2.8.3 DayOne

DayOne is one of the most popular and oldest journaling apps on the app store and was created in 2011<sup>10</sup>. As with the other apps, it features a sidebar containing a tab bar with media, timeline, map, and calendar tabs. A nice addition is the easy addition of a new entry via the plus button in the middle of the tab bar and the fact that the sidebar is collapsible and always accessible. A somewhat questionable choice is the addition of a somewhat hidden "hamburger" menu - containing further options like streaks and journal switching - behind the sidebar which can be easily missed.

DayOne also supports multiple media attachments, such as sketches, images, and videos, and even a speech-to-text option for premium users - available on the Apple Watch as well. Metadata, like automatic weather and geotags, can also be added in DayOne. However, accessing the metadata submenu of an entry is somewhat hidden and not as intuitive as in Journey, for example.

It can inspire users via a daily prompt, custom reminders, and customizable templates to help start a new entry. However, the customization and template are not as feature-rich and easy to use as in GridDiary.

### 2.8.4 Conclusion/Comparisons

In summary, all the examined apps were similar in look and feel, with some standout features and minor differences in design and navigation between all of them. All of them allow users to add meta-data, e.g. weather or mood, whether automatically or manually, allowing for further statistics and other tracking plus metrics for users to track. These features are a common denominator and seemingly important for both users and designers. Rich editing possibilities are an (expected) common feature across

<sup>10</sup><https://dayoneapp.com/>, last accessed on 9<sup>th</sup> March, 2023

## 2. STATE OF THE ART

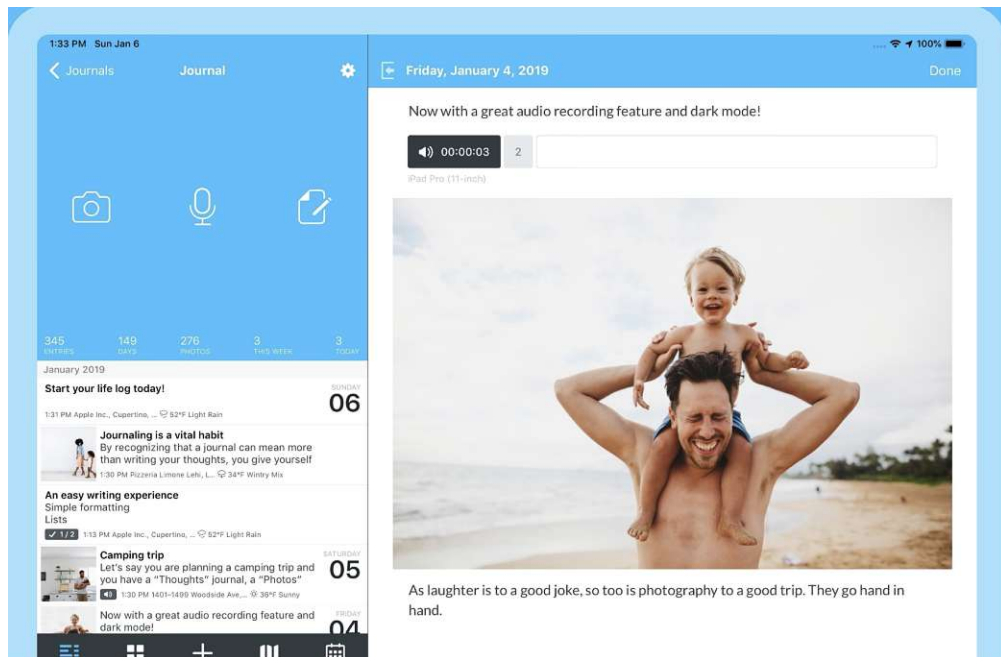


Figure 2.7: DayOne's overview page offers quick access to creating a new entry, adding other media, as well as a well-structured timeline with additional stats of the current journal<sup>11</sup>.

modern diary apps and almost a requirement to structure and organize longer entries in a journal/(research) diary.

Pencil support seems to be lacking in the examined apps - there are others that are focused explicitly on hand-written diaries, however, these often don't have typed text support at all. There does not seem to be a good hybrid solution, compared to note-taking apps, like Notability.

Statistics, timely reminders, and "streaks" encourage users to keep journaling as often as possible - this is especially important when starting out with this habit.

Overview of all captured media, the timeline, and calendars are essential features of diary apps since they are more chronological in their nature and are more built upon (personal) reflection than general note-taking apps.

Since all the tested apps use a subscription model, users might have to commit to one app for a certain amount of time, and the switch between apps might not be too easy.

Encryption and protection are also a high priority for users and all diary apps highlight their data protection and the option to lock a diary behind a passcode.

In general, if someone needs and relies upon reminders and positive reinforcement to form and keep the habit of using a research diary, such journaling apps might be more suited for them and help them more in their journey. In turn, if someone needs a more

structured approach, prefers organizing their diary into different folders, and wants better multi-modal support, a note-taking app like GoodNotes might be a better idea.

## 2.9 Note-Taking Apps

In this section, we will look at examples of the numerous note-taking apps available on the iPad. There are many, many more that would also be suitable for a research diary, so this list is in no way exhaustive or definitive. They can range from simple, like Apple Notes, to quite complicated and extensive, such as Notion or Bear.

### 2.9.1 Notability

Notability is a popular, multi-modal, and multi-media note-taking app that has been at the top of the iPad App Store popularity ranking quite often and even won the "App Store Editor's Choice" award.

It is both a (handwritten) note-taking app and a tool for PDF annotation by supporting the import of images, GIFs, and PDFs.

It also has support for typed text, both in the form of freely typed text or text boxes with a specific size and location. In both cases, there is support for rich text editing and different fonts, however, text alignment can only be done in text boxes. Furthermore, handwritten text can be recognized and copied as regular text to other applications.

Notes can be organized into *Subjects*, which can in turn be organized into *Dividers*. The layout of the app is very intuitive and straightforward, with a sidebar containing all *Subjects*, *Dividers*, and recent notes. On the right side, there is a grid overview of all available notes in the current context. Notably, a note is always opened in fullscreen, with no access to the sidebar, however, notes can also be opened in a new window, allowing users to view two notes at the same time.

An interesting and unique feature is the support of audio recordings that can even be synced to the other contents of the note. That means you can play an audio recording and see visual feedback of what was written/drawn during an exact moment or tap on a specific word or stroke to play what was said during that time.

Additionally, you can create and import custom templates for notes, in order to start out with a specific background or spacing to speed up and streamline your note-taking process.

A main point of criticism, according to the user's App Store reviews, and a disadvantage of Notability is the fact that the app has recently switched to a "freemium" subscription plan that requires users to pay a fee monthly or yearly to access all of its features, as compared to a one-time payment as before, and what other apps, such as GoodNotes offer. It can also only create and use custom layouts and templates on a per-note basis, while e.g. GoodNotes is capable of doing it on a per-page basis.

## 2. STATE OF THE ART

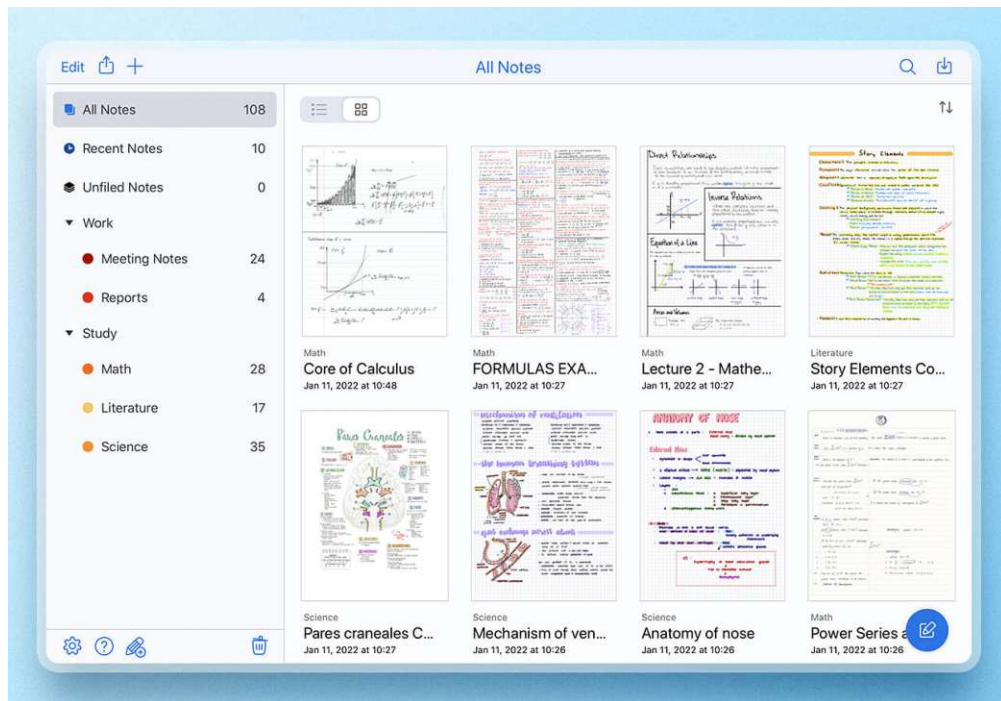


Figure 2.8: The layout of Notability is as simple and basic as an app like this can be. It offers a sidebar with different folders/"subjects" on the side, the option to edit a note by tapping on it or creating a new one by tapping on the "plus" button on the bottom right<sup>12</sup>, last accessed on 9<sup>th</sup> March, 2023.

The different drawing tools, such as the available pencil options are not as extensive and customizable as in other apps, but mostly enough for most use cases.

### 2.9.2 GoodNotes

GoodNotes is another popular note-taking app that has even been awarded by Apple as the best iPad app of 2022<sup>14</sup>.

GoodNotes offers a central place via the *Documents* tab to view all bookmarked pages, folders, and notebooks across your account allowing you to view them at any time. They are organized in a grid view and subfolders are opened in a subpage. Most of the individual and detailed organization is done via Notebooks, where each individual page layout and background can be changed and set to the users' needs and demands. The organization of different notebooks and their respective folders can be quite useful depending on the use case, but it can also be counterintuitive. E.g., for a research diary, it would make more sense to have deeper categorization and subfolder creation within a notebook, as one usually does not have more than one research diary at a time.

<sup>14</sup><https://www.apple.com/newsroom/2022/11/app-store-awards-celebrate-the-best-apps-and-games-of-2022/>, last accessed on 9<sup>th</sup> March, 2023

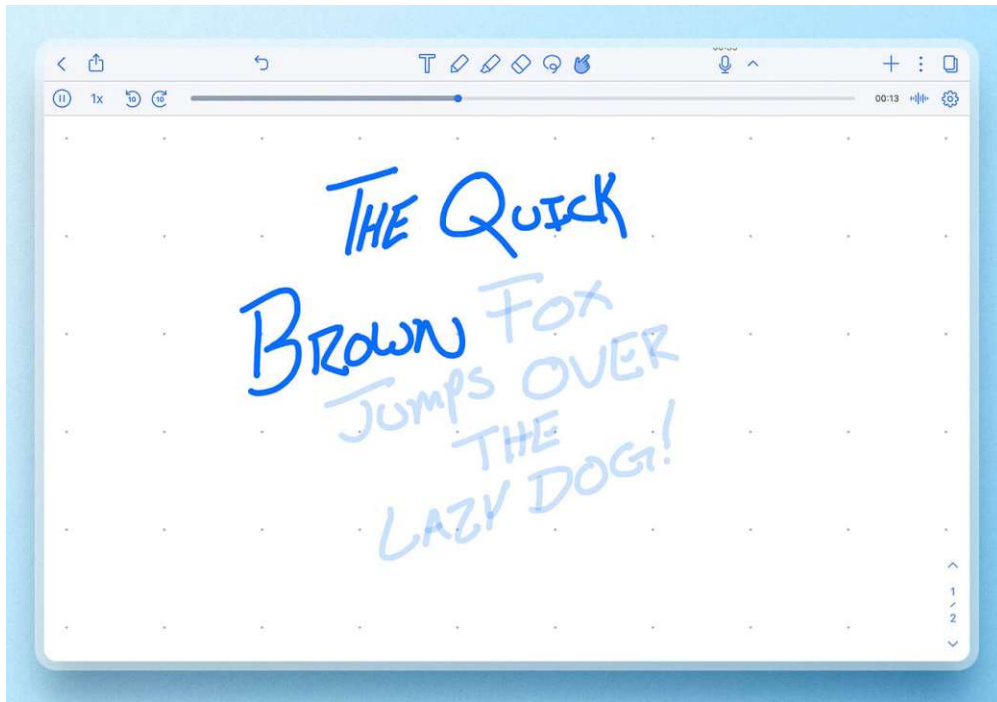


Figure 2.9: A standout feature of Notability is the way it can synchronize audio with other content in a note, allowing users to jump to the exact point a stroke was created in the audio, and vice-versa.<sup>13</sup>, last accessed on 9<sup>th</sup> March, 2023.

Content is also automatically synced to Apple iCloud or other cloud storage solutions, such as Google Drive and users can also share notes or notebooks with others. GoodNotes is also available on the iPhone and the Mac.

The app also makes use of its own handwriting detection algorithm, allowing users to easily search through their digital handwritten notes, and offers very detailed and deep pencil customization options, usually only found in drawing apps, like Procreate<sup>16</sup>. Users can also create custom elements, either by importing them or selecting elements of a notebook page to create new ones. These elements are somewhat similar to "stickers" in other apps and can be conveniently reused and inserted whenever needed.

Typed text can be inserted at any part of the page via text boxes, however, the text input lacks more sophisticated rich editing capabilities that other apps offer, further reinforcing the impression that this app is clearly focused on handwriting with a pencil first. It also does not offer audio input, like Notability does, for example.

<sup>16</sup><https://procreate.com/>, last accessed on 9<sup>th</sup> March, 2023

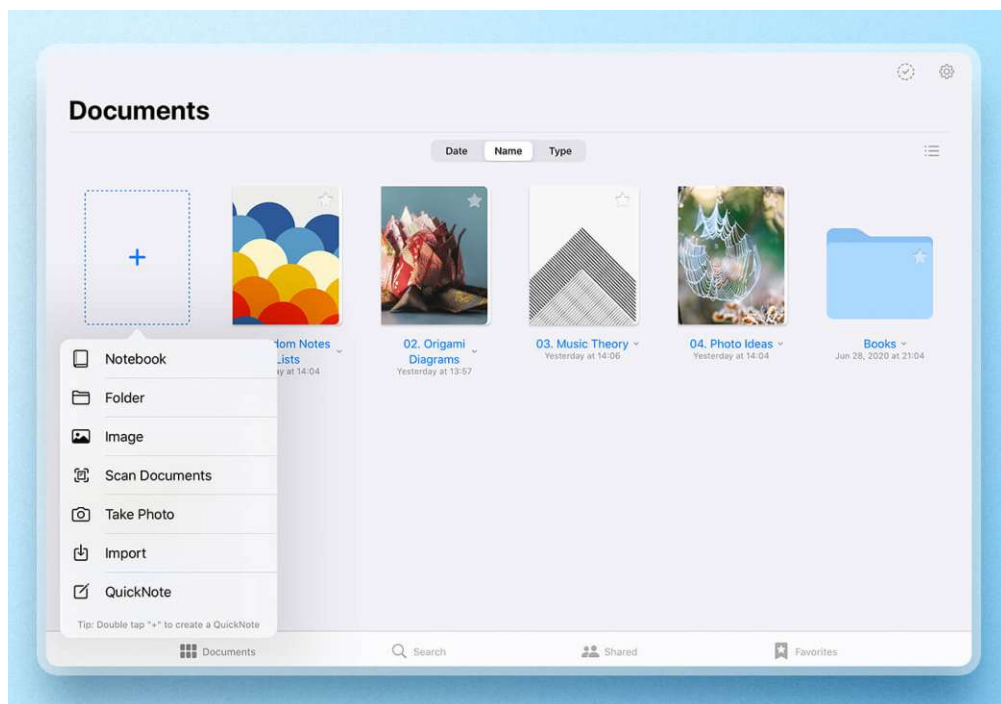


Figure 2.10: GoodNotes structures content into folders, subfolders, and Notebooks. Notebooks can contain as many pages as desired with many layout templates to choose from for each page<sup>15</sup>

### 2.9.3 Apple Notes

Most people that have used one of the many Apple products and operating systems, such as iOS for iPhones or macOS for MacBooks, have come across the Apple notes app, since it comes preinstalled with iPhones, iPads, and Macs. It has been an Apple staple for many years and has undergone various iterations. It offers a very streamlined and straightforward way of taking and creating both handwritten and typed notes and has the unique benefit of being deeply integrated into the Apple ecosystem, which allows - among other things - for automatic syncing across devices, encryption, and easy creation and adding of new notes thanks to Quick Notes. Quick Notes is a new feature since macOS Monterey and iOS/iPadOS 15<sup>17</sup>, that allows quick access to a new note to write down thoughts, links, etc. immediately.

It offers rich editing capabilities in the trademark, *minimal* "Apple way", as well as the option to add media from your library or iCloud, scan documents, and much more. It also allows users to organize their thoughts and notes via folders and tags (automatically created by prefacing a word with "#").

<sup>17</sup><https://www.guidingtech.com/how-to-use-quick-notes-on-iphone-ipad-and-mac/>, last accessed on 9<sup>th</sup> March, 2023

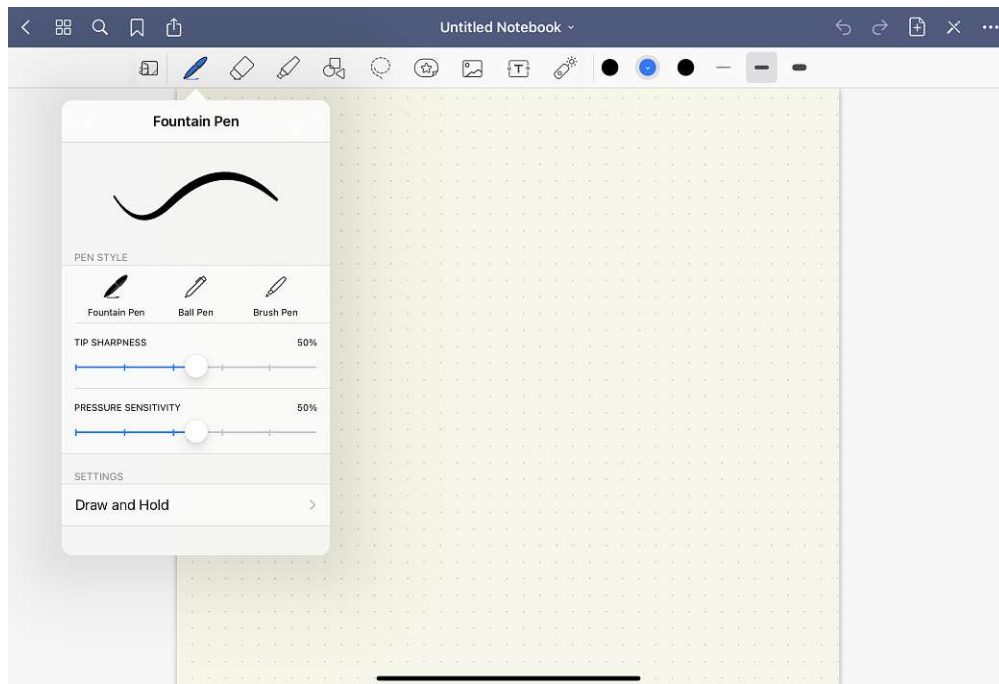


Figure 2.11: Since GoodNotes is quite focused on pen input, it offers very extensive ways to customize the behavior, colors, and look and feel of different drawing tools, such as the pen, usually only seen in drawing apps.

As one can imagine, the design is mostly adhering to Apple standards set in the Human Interface Guidelines[28], featuring a collapsible sidebar across platforms, showing the folders and tags on the leftmost side, followed by an overview of notes either as a grid or list in the context of the current folder or set filter and on the rightmost view is the note editor. The latter features a simple menu with corresponding submenus and a search bar on top.

While users can add both pencil and keyboard input, there is no way for them to scribble or draw over existing text as this input is then added in a separate view, as illustrated in figure 2.13.

One main advantage of Apple Notes is that it is completely free and, as mentioned, comes preinstalled with Apple devices, without hiding any features behind a paywall, as other apps do. It is also quite easy to use and can be easily and quickly opened anytime thanks to Quick Notes and allows collaboration with others via shared notes - a feature not easily found in other apps.

To sum up, Apple Notes offers a comprehensive, if somewhat basic user experience that is quite suitable for most needs for notetaking, but if users want more specific features, e.g. for PDF annotation, combining handwritten and typed text, or want specific custom layouts for specific entries, they would require other more in-depth solutions.

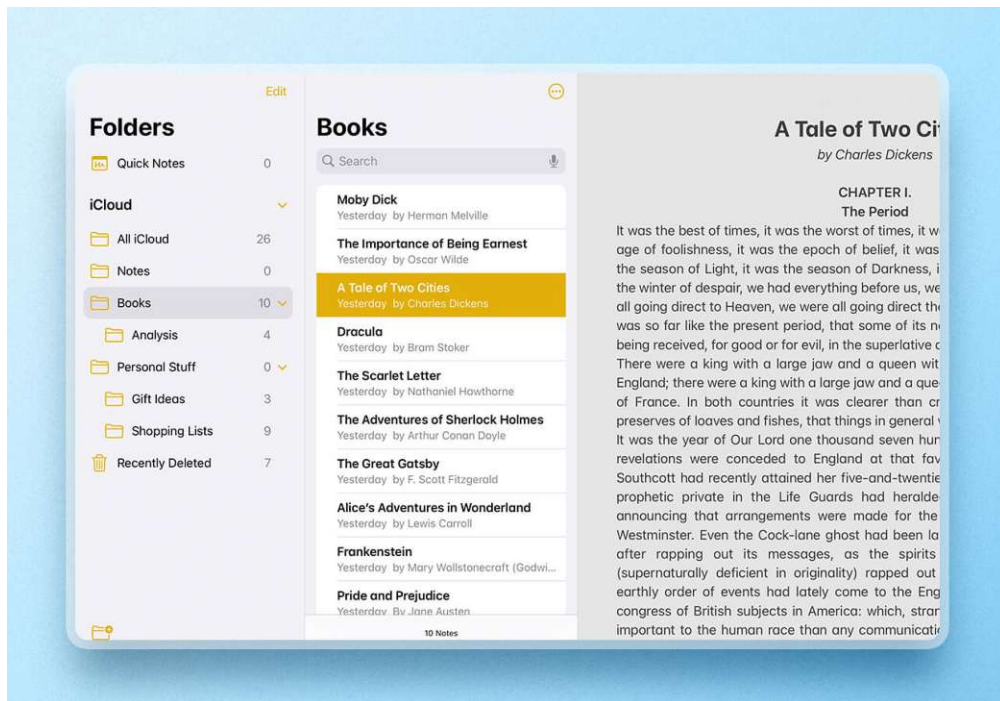


Figure 2.12: Apple Notes has a general structure of three collapsible hierarchical views, from left to right: the folders (both local and in the iCloud), the current list of notes, and the currently opened note in the editor view<sup>18</sup>.

### 2.9.4 Conclusion/Comparisons

The examined note-taking apps all provide a very good way of organizing different entries via tags, favorites/bookmarks, folders, and other convenient features. Apple notes offers a more straightforward approach which might make it easier to start out with, especially for users that mostly like to type and not draw with their pencil.

Notability and GoodNotes are both a good fit for note-taking with the iPad and choosing between them is mostly up to personal preference. The main way of input seems to be focused on handwritten text with the apple pencil, with typed text being more of a secondary focus. For someone that prefers to type (especially with an external keyboard) mostly, there might be better-suited apps, such as Evernote<sup>20</sup>, Apple Notes, or Notion<sup>21</sup>.

Multi-modal input via the Apple pencil is significantly better supported and more straightforward with all of these applications compared to the apps from the previous section.

Suppose one often encounters "writer's" block or needs (daily) encouragement to write in a research diary. In that case, the mentioned diary apps might be better suited as they

<sup>20</sup><https://evernote.com/>, last accessed on 9<sup>th</sup> March, 2023

<sup>21</sup><https://www.notion.so/>, last accessed on 9<sup>th</sup> March, 2023



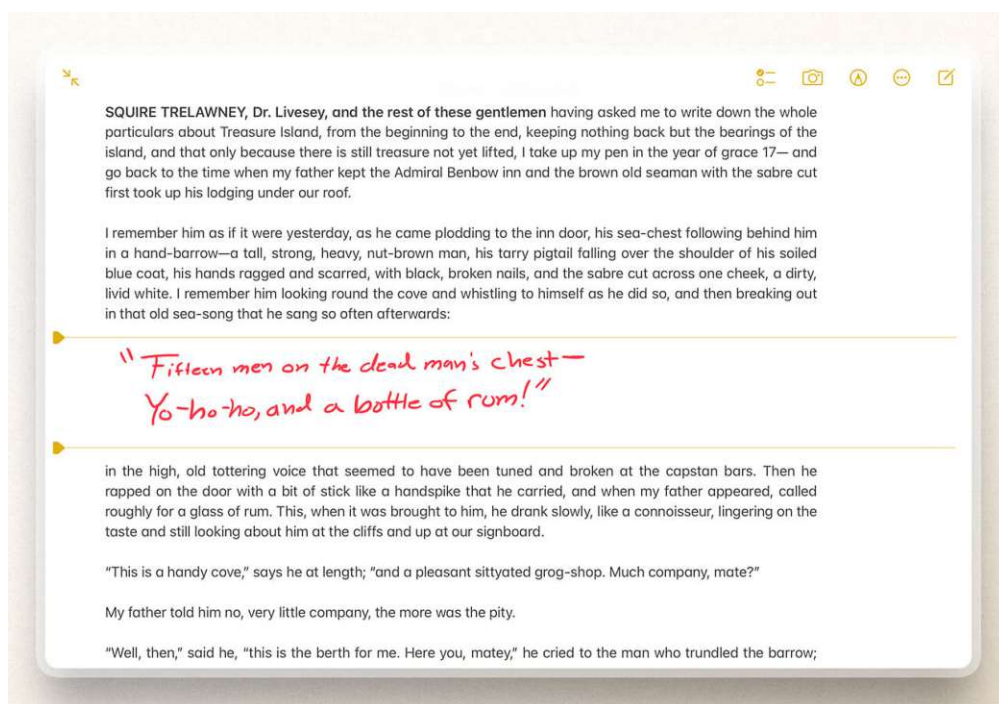


Figure 2.13: Apple Notes does offer both handwritten pencil input, as well as typed text. However, it makes a point to separate the two into different sections, further illustrated by the two yellow lines above and below the handwriting<sup>19</sup>.

provide customizable prompts for each entry that allows users to always have a good starting point and other support such as habit and "streak" reminders. However, if users prefer a custom background and layout for their research diary, an app like GoodNotes or Notability might be better suited and provides more options in that regard.

The mentioned apps also provide better support for in-line images and importing PDFs to annotate them which might be nice for users that try to transition from an analog diary to a digital one or often write something down on a piece of paper that they later want to continue working on or annotate digitally.

## 2.10 "Smart Notebooks"

Between digital apps for the iPad and the traditional *analog* notebook/journal sit the so-called "smart notebooks" [38]. These hybrid solutions combine these two mediums by offering the look and feel of "real paper" and the convenience of automatically digitizing its content, mainly in PDF or image formats.

## 2. STATE OF THE ART

---

Examples for this include the Moleskine Smart Writing Set<sup>22</sup> or the Rocketbook Smart<sup>23</sup>. In Moleskine's case, this is achieved via a special "smart" notebook, as well as a smart pen that automatically transfers new pencil strokes to the digital companion app, *Moleskine Notes*<sup>24</sup>. The Rocketbook relies on a reusable "smart" notebook that should only be used with special *Pilot Frixion* pens and whose content is then scanned with the companion app.

Another hybrid solution would be the reMarkable (2)<sup>25</sup>. It is basically a tablet with an e-Ink display, similar to that of an e-book reader, aiming to simulate the feel of traditional paper while being entirely digital, allowing quick and easy edits, erasing, and automatic cloud backups.

What unites all of these solutions is that they are solely focused on handwritten notes. While this might be enough for the needs of some users, this thesis aims to explore all the unique multi-media and multi-modal options that the Apple iPad provides, such as adding images/audio to a notebook, additional meta-data, categorization, and the option to type, either with the on-screen keyboard or an external one, like the Apple Magic Keyboard<sup>26</sup>.

---

<sup>22</sup><https://www.moleskine.com/en-us/shop/moleskine-smart/>, last accessed on 9<sup>th</sup> March, 2023

<sup>23</sup><https://getrocketbook.com/>, last accessed on 9<sup>th</sup> March, 2023

<sup>24</sup>[https://www.moleskine.com/en-us/app\\_notes.html](https://www.moleskine.com/en-us/app_notes.html), last accessed on 9<sup>th</sup> March, 2023

<sup>25</sup><https://remarkable.com/store/remarkable-2>, last accessed on 9<sup>th</sup> March, 2023

<sup>26</sup><https://www.apple.com/ipad-keyboards/>, last accessed on 9<sup>th</sup> March, 2023

# CHAPTER 3

## Design

In this chapter, we are going to discuss, document, and elaborate on the design process, steps, and evolution from sketches to wireframes, to the (final) screen designs.

### 3.1 Design Considerations

The basis of the design and main inspirations were drawn from the different note-taking and diary apps we looked at in the previous chapter.

The main goals and criteria for the final design were similar to the selection criteria for existing apps, namely good multi-modal support, easy overview and creation of entries, and a simple, but powerful entry editor view.

#### 3.1.1 Navigation/Entry Overview

Navigation is a key factor for every modern app. Confusing navigation quickly leads to frustration for a user and usually results in them not using the app as much as they would otherwise, or maybe not at all [36]. It is also one of the things that users notice first and helps them to get a grasp of all its features. The famous notion that "good design is invisible" also holds true for navigation[46].

Users rely on good navigation without noticing it, except when it is poorly designed. One more important thing to note is that since mobile device screens are still small compared to a laptop or a computer screen, it is impossible to display all the features of an application on one screen. This is less the case for modern iPads, depending on the used model since they range from around eight inches up to almost 13 inches<sup>1</sup>

---

<sup>1</sup><https://www.macworld.com/article/670670/best-ipad-2021-which-ipad-to-buy.html>, last accessed on 9<sup>th</sup> March, 2023

When looking at most default apps provided by Apple as well as other diary apps we looked at last chapter, there seems to be a general consensus in navigation for a so-called sidebar, as Apple likes to call it and as specified in their Human Interface Guidelines (HIG) [28].

Some apps combined this sidebar with an additional tab bar contained in the sidebar, something not explicitly recommended by Apple. Still, it gives users even more options to view different overviews of their content. Apple's default apps go a different route by providing not one secondary pane but an additional tertiary pane. There are advantages and disadvantages to both of these approaches. Still, for the sketches and wireframes, I mostly went with the combined approach of a sidebar and tab bar, before settling on just a sidebar for the time being, as we will discuss later.

#### 3.1.2 Entry Editor

The editor view of an entry is the most important and main way of interacting with the app for most users - or at least should be - so a lot of design efforts and considerations were spent on exploring different approaches and options to arrive at the best solution for the desired use case.

Since I was keen on exploring the combination of pencil and keyboard, an early decision was to combine these two so that users did not have to feel like they had to sacrifice their preferred way of creating entries, whether via handwriting or typing. Many apps focused either on typing, e.g. Grid Diary, or hand-writing, e.g. GoodNotes. Realistically, it is already hard to do one of these things very well, and two would be even harder. However, the developed app should be viewed more as a proof-of-concept and an exploration of what would be possible with more time and resources.

Rich editing possibilities were also an important feature since one entry could potentially span multiple pages, and users would expect some way of formatting the text to increase readability and organize their thoughts better. Most existing diary and note-taking apps provide such features, although some might hide them behind a paywall. Beyond being able to add styling to their text, users accustomed to other diary apps would also expect a (research) diary app to be capable of adding meta-data, such as location, weather, tags, and mood, to their entries. These would help them organize their entries better and provide potentially interesting statistics later on. As we will discuss later, the placement and organization of all these tools would prove to be a design (and implementation) challenge.

The addition of other media and documents was also a challenge. Still, an early decision was not to add a PDF annotation feature or allow images to be directly inserted into the text, as other apps like GoodNotes do since this would be too complex to implement for this thesis and would not help much in answering the research questions at hand.

### 3.1.3 Coding/Analysis Capabilities

Since I wanted the application to be more than just a diary app with multi-modal input support, I explored different options for adding coding and categorization support. This would allow users to directly code their diary in the app without needing another external solution and hopefully help them with their final write-up.

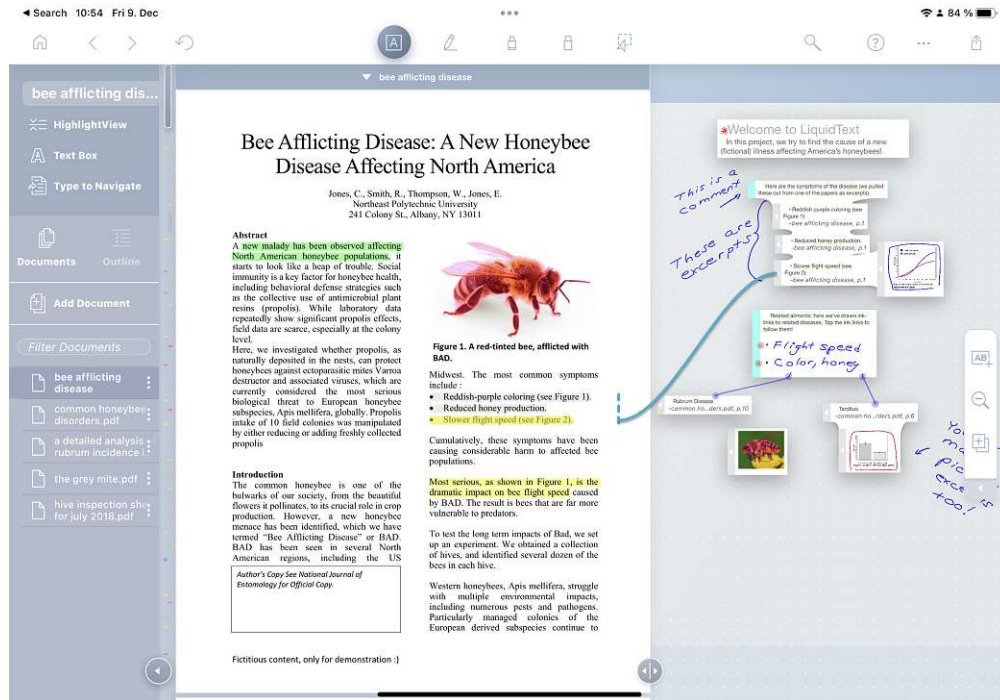


Figure 3.1: LiquidText provides a unique approach and way to view and connect comments, highlights, and excerpts. This visual representation was a big inspiration for designing and implementing the coding views.

There are apps like LiquidText<sup>2</sup> that provide exciting and unique ways of connecting different notes and remarks with another, creating an almost *mindmap*-like connection between highlights, notes, and quotes in a way that really elevates it from being just another PDF annotation app. However, replicating this exact behavior would be too complex, and the app is mainly focused on going through dense, long (PDF) documents, instead of one's own written thoughts but this connection of highlights across entries and pages is something that I wanted to replicate, albeit in a simpler and more manageable way.

Overlapping and multiple applicable codings for one section/text were also considered. However, this idea was not further pursued for two reasons. Firstly, designing and implementing a feature that allows different highlighting for the same text passage would

<sup>2</sup><https://www.liquidtext.net/>, last accessed on 9<sup>th</sup> March, 2023

have been quite complex and time-consuming. Secondly, it might have still not been the best way of doing it, according to Creswell [19]. He encourages researchers to identify the main idea being passed on and assign this idea a single code.

## 3.2 Design Steps

The design process underwent the traditional and well-established process of starting with very low-fidelity sketches, moving to medium-fidelity wireframes, and lastly high-fidelity screen designs. This was an iterative process with various steps and I went through refinements by myself and my advisor to improve the functionality. Starting out with low-fidelity sketches allowed me to quickly explore vastly different ideas and approaches for different app feature sets, without needing to re-write existing code or create highly complex fleshed-out user interfaces each time [13].

Sketching has been done with an Apple Pencil in GoodNotes while the steps afterward were created using Adobe XD<sup>3</sup>.

### 3.2.1 Sketching/Wireframing Phase

In the early sketching phase, I was mostly concerned with three different, difficult design questions to answer:

- How would the navigation/overview of the app work and what are the most essential views and features for my prototype?
- Which features and layout should the entry editor have and what additional data can and should be added?
- How and in which detail can a user code and categorize an entry and its contents?

### 3.2.2 Navigation/Overview

The first phase and sketches were concerned with answering the first question of navigation. We discussed some of the considerations in the earlier section 3.1.1. Early sketches and wireframes included a sidebar with a tab bar. Mixing these two types of views can be potentially confusing/overbearing and not explicitly recommended by Apple, since users would miss out on a significant advantage of a tab bar - the fact that it is always visible and users can easily navigate between different sections of the app. However, I would argue that in the case of a digital (research) diary app for an iPad, this approach would make sense since users would want to focus mainly on editing or creating a new entry in their diary. It could also lead to synchronization issues and higher complexity if users were able to open the same entries from different tabs at the same time. This is impossible with a sidebar, as there is always only one primary view.

---

<sup>3</sup><https://www.adobe.com/products/xd.html>, last accessed on 9<sup>th</sup> March, 2023

I also noticed this while testing other note-taking and diary applications, such as Notability or Journey. Switching between and editing different entries quickly was somewhat cumbersome since they were opened in a fullscreen context. This was, in my opinion, better achieved by apps like Apple Notes, which - as one could imagine - adheres closer to the Apple Human Interface Guidelines regarding navigation and sidebars. While editing an entry in Apple Notes, the user does not lose the option of top-level navigation via the sidebar and iPadOS offers the option for collapsible sidebars out of the box. This way users can always decide to hide the sidebar if they do not need it, but they also always have the opportunity to navigate between different entries.

The final product and designs do not feature a tab bar. This is mostly due to the fact that implementing all these different views and features was considered out of scope for the implemented prototype but would be featured in a fully-fledged application.

I also considered a separate tab for a *bibliography*/list of references. This would allow users to directly write down and manage books and other sources they read. But, although it might serve as a documentation of read publications and books, it should be argued that a diary is not the best place for a (sole) bibliography and should be kept separately [37]. This is why I decided against designing and implementing such a feature in more detail. Furthermore, this bibliography functionality was also deemed out of this project's scope and would have required much more time and research. It is also questionable whether users would have made use of such a feature/reference management tool, especially on an iPad.

In the early phase, the concept featured the option for multiple research diaries that users could switch between. This is, however, not featured in the final design because of multiple considerations. Firstly, it is doubtful that many users would have multiple active research diaries at a time, not just one primary one they want to devote their time to. Secondly, this would have further increased the implementation complexity and would not have helped much in answering the research questions.

Thirdly, depending on the chosen concept and implementation, this would have repercussions for the ease of use and intuitivity of the navigation strategy. One way of solving this would have been to open a diary in an entirely new fullscreen context losing the previous overview. Another would be to introduce a tertiary view in the sidebar that allows switching between diaries, similar to how Apple Notes handles folder switching. This would increase complexity, add another "hidden" navigation, and add further questions for the different overviews. For example, should a calendar overview only show entries of the currently selected diary or allow users to mix and match? And if the latter is the case, how can users be quickly aware that they now switched contexts and possibly opened another notebook by tapping on a different entry?

### 3.2.3 Entry Creation/Editing

The second phase was focused on trying to come up with different ways of creating a new entry and editing an existing one.

The first problem to solve was the possible layouts for creating an entry and if users should be supported with prompts or other forms of templates, or if they should always start with a clean slate.

An early sketch and wireframe of the entry creation included a step-based layout that used a customizable template. A step was included for each prompt in the template, inspiring users to add something to each step. This would allow users to really focus on each question at hand but it had more problems than it provided solutions or help, in my opinion. Firstly, users could only type or write in a small view limiting the already smaller screen real estate, unless the editor would have then opened in fullscreen which would somewhat defeat the purpose of such a layout in the first place. Secondly, users would always have to follow the same cumbersome process, and picking and choosing between prompts would have been more challenging and annoying than just showing a grid from the start. A created entry would then be presented in a grid format, and further editing would have been possible. This was largely inspired by *Grid Diary* 2.8.1 and its titular grid layout with a similar template approach. In addition to the earlier mentioned drawbacks, it would have been questionable whether this approach would have worked for every user or even a research diary in general. As discussed in the previous chapter, a research diary should be a personal companion that allows any kind of thoughts and content. Such an approach might constrain users in their freedom of expression and thoughts. Additionally, limiting the screen space would have made drawing and writing with a pencil more difficult. One advantage of this approach - other than providing an easier structure with prompts - would have been that coding and tagging could have been applied more granularly for each section. The next decision in designing the editor was to combine handwriting and text editing in one place, allowing users to use both input modes interchangeably as Notability achieved. This was a common missing feature in other apps focused on diaries, such as DayOne, and I wanted to combine some of the advantages of these apps with the better input support of dedicated note-taking apps. This would mean that users would have the option to switch between a rich text editor and a canvas view for their pencil in the same space, e.g. scribble over their typed text. As we will discuss in the next chapter, this decision severely increased the implementation complexity and introduced several problems, some of which were not wholly solvable.

This brings me to the next design consideration - the placement and layout of the different toolbars of the editor. The earliest designs featured all the toolbars at the top of the screen, somewhat separated by categories, e.g. text styling and meta-data.

This concept allowed for more space for the text and pencil input. However, grouping all these items into one view - even with some visual distinctions between categories - could be too confusing and overwhelming for users.

Another approach was inspired by Apple Notes' more streamlined and minimal UI design. Instead of displaying each text styling option and metadata option individually, they would be contained in a menu button that displays a popover submenu and only the most important features would be immediately available. This would clean up and simplify the user interface, however, it would also decrease the discoverability of available features



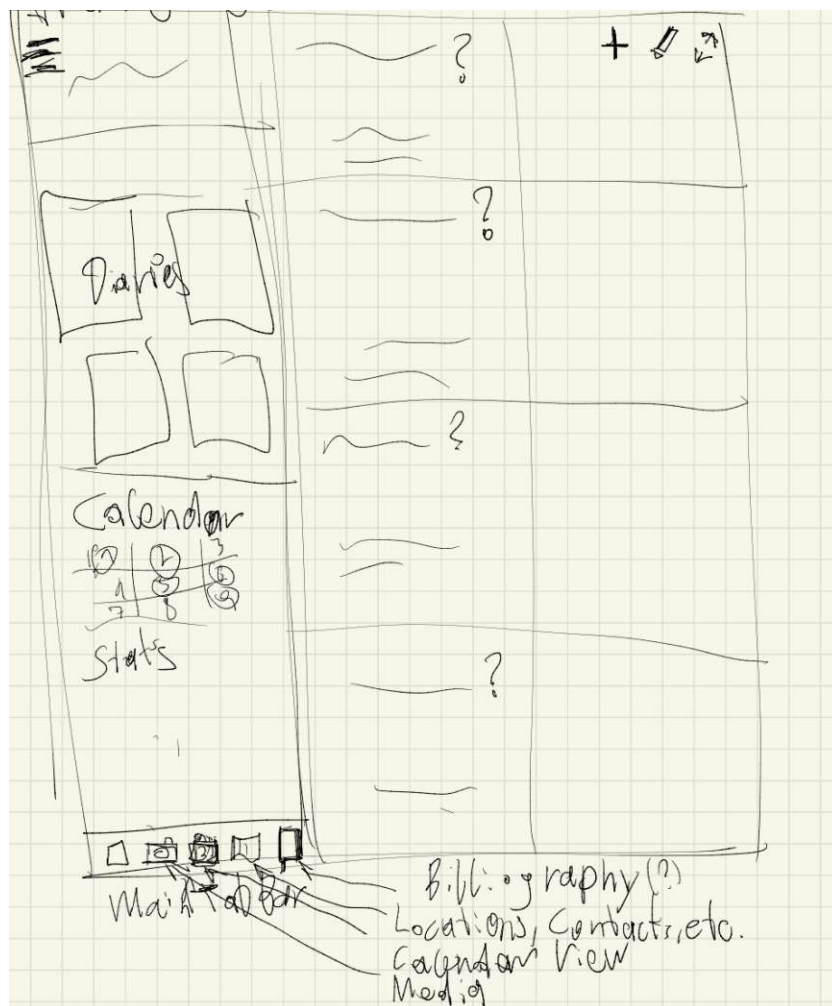


Figure 3.2: An early sketch featuring a sidebar containing a tab bar with potential overview tabs on the left. On the right, an entry screen with a grid-based layout and prompts for each section.

and users would not necessarily know where to look for and expect them in favor of a smaller, possibly less overwhelming interface. Additionally, I did not want to make the decision of what would be the most "important" feature and meta-data option for the users. Plus, I wanted to find out in my user tests what feature users were using the most or which was the most exciting/useful to them without influencing their decision.

### Meta-Data

Other existing diary apps mostly informed the inclusion of meta-data options to help users understand their habits better and gain better statistical insights into them. What kind of data to include was discussed with my advisor and was based on ease of implementation,



Figure 3.3: A concept for a grid-based entry editor consisting of multiple editable views based on a changeable list of questions/prompts.

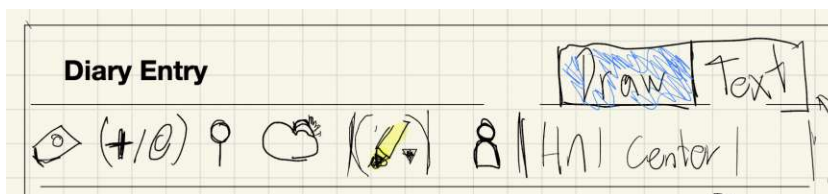


Figure 3.4: The earliest entry toolbar wireframe, featuring a combined entry toolbar on top of the screen.

and perceived usefulness for the end users, and ranked accordingly. As discussed in the previous section, the final placement and design would be on the top left of the entry screen, in a horizontally scrollable view. All the meta-data options would be opened in a popup view which can be easily dismissed by tapping outside it while only taking up a small portion of the screen. Furthermore, another design iteration/improvement was to ensure that the currently set value of a meta-data category, such as the name of the current location or the selected mood, is displayed to the users in the meta-data toolbar as well. This way users can easily see if they have already added meta-data and what its value currently is, without needing to open the corresponding view first.

In the final implementation of the app, a statistical overview of this data alongside other data, such as habits, number of entries, and length of "writing streaks" would be included.

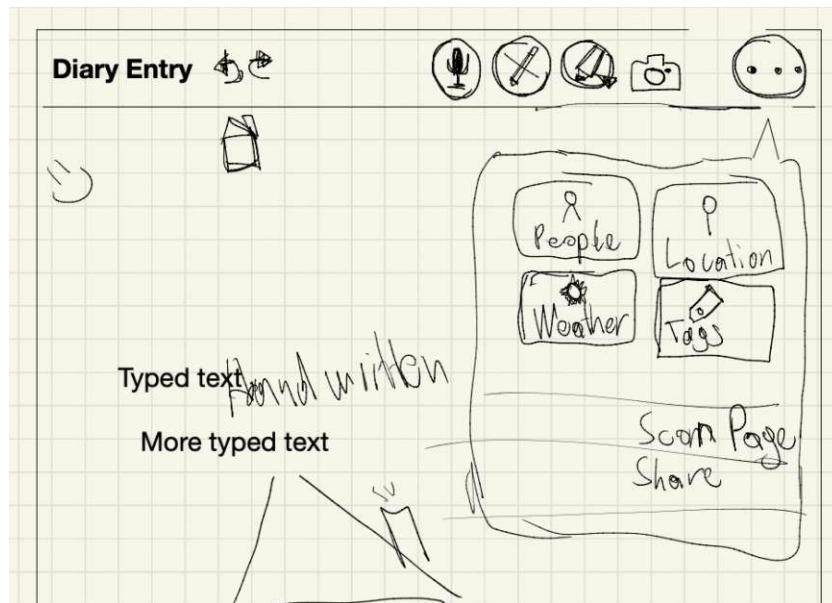


Figure 3.5: A design sketch, inspired by *Apple's Notes* app featuring a streamlined, straightforward design with different sub-menus.

### 3.2.4 Attachments

The possibility of adding digital attachments was an important goal and design consideration from the beginning. It was essential for me that users have the option to add other digital media, e.g. scans of written documents or other notes. Early drafts and concepts included the option to directly add images and PDF documents to the editor, allowing users to draw over them and annotate them directly. However, this idea was abandoned because of the hugely increased implementation complexity. Instead, a separate view was conceptualized that would allow users to add and view their media. This was designed as a dismissable **page sheet**<sup>4</sup>. Such a sheet allows users to add media without losing the current context and users can adjust the size of the attachments sheet to their preferences.

### 3.2.5 Highlights, Notes, and Codes

The option to add highlights, notes, and codes was designed as a standout feature compared to existing diary apps and was inspired by other apps, such as LiquidText, as discussed in the previous section 3.1.3.

Some of these were scaled down in complexity, such as the option to link and connect highlights in a visual view. Instead, a separate view was designed that would display all highlights/codes in a tabular manner, grouped by entry and filterable by category.

<sup>4</sup><https://developer.apple.com/design/human-interface-guidelines/components/presentation/sheets/>, last accessed on 9<sup>th</sup> March, 2023



Figure 3.6: A wireframe of a concept for a step-based entry creation screen, based on a template/list of questions.

The first sketches included a togglable highlights overview section on the bottom section of the editor screen which could be resized to the user's wishes. The problem with this would be that this view would firstly not adhere to Apple's Human Interface Guidelines. Secondly, it might be too overwhelmingly present for most use cases, since users usually want to code and highlight after a re-read or during later phases of maintaining their research diary and not while creating or editing an entry.

Another approach would have been another sidebar with a list of the current highlights. The problem with this concept was increased complexity, as well as the limited horizontal space for potentially long highlights, as well as highlights of drawn content.

### 3.2.6 Final Design Decisions

The final design decisions and the resulting screen designs were mostly informed by discussions with my advisor, research and comparison with existing solutions, as well as research into their implementation feasibility.

In the editor screen, I decided to move the pencil and rich editor toolbar to the bottom of the screen. The rich editor toolbar would be attached above the keyboard, and the pencil toolbar would be implemented as a floating view that can be moved according to the user's preferences. All other edit options were moved to the top and grouped into meta-data on the left, highlighting category, and input mode selection on the right. This would allow for better separation of different concepts and makes it easier to use the

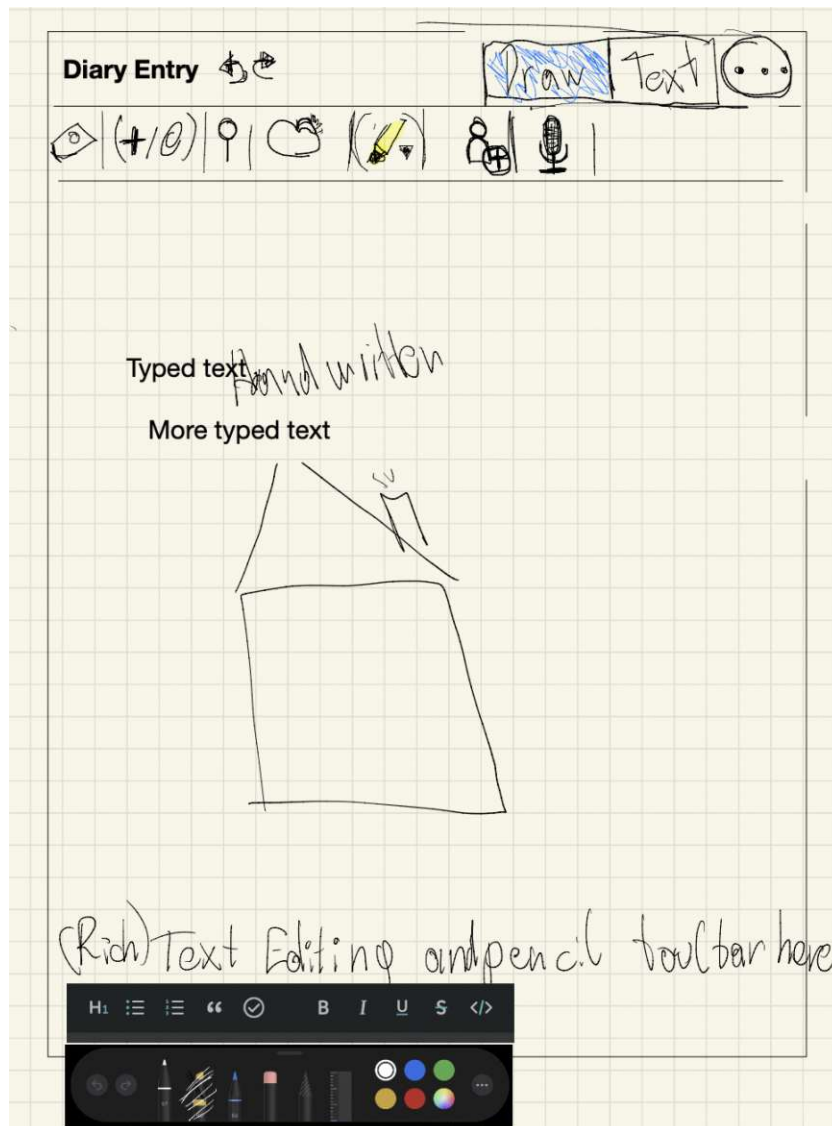


Figure 3.7: The final entry editor sketch, dividing the editor screen with a toolbar for input mode switching, meta-data, and other edit options on top and a context-appropriate toolbar for pencil or text input.

editing toolbars while drawing/typing in an entry.

Additionally, an optional text field was added that allows users to set custom titles for an entry beyond just a date to further differentiate between entries and sort them better.

The attachments and highlights overview screen designs both feature a collapsible sheet, as described in the attachments section. This is a prevalent concept in iOS/iPadOS <sup>5</sup>

<sup>5</sup><https://developer.apple.com/design/human-interface-guidelines/components/presentation/sheets>,

and allows the app to make more use of the width of the screen, as well as allowing users to access the previous context beneath it quickly. It should be mentioned that the default view provided by the UIKit SDK since 15.0<sup>6</sup> does not make use of the full width and I decided to adhere to Apple's decision and their guidelines.

#### 3.2.7 Input Mode Switching

One of the most challenging aspects of designing the user interface of the entry editor was the question of the placement, look, and functionality of the input mode switching between the apple pencil and the keyboard. In the early stages, a simple button was used to signify the currently selected mode, similar to how Apple's Notes app has done it. This proved to be a problem since it lowered the feature's discoverability and users might not have found the feature, and users would not be aware of the available input options at a glance with this design. Furthermore, this button could have only served as a binary option - either pencil or keyboard input is active. This was later changed to three options, as we will discuss.

Since the previous design could have led to possible confusion, it was changed to a so-called "*Segmented Control*"[5]. This UI element consists of several visually segmented views, each serving as a possible option that can be toggled via a button. This UI element better conveys the possible input modes and the currently selected mode to the users. Additionally, I decided to distinguish between the highlighting mode and the general drawing and scribble mode of the pencil. This was due to the fact that the highlighter in the canvas toolbar serves a particular function in the app and can be used to highlight and code drawings. The highlighted drawings are automatically added to the highlights screen as an image, as described in the implementation chapter 4.2.

A segmented control allowed me to add this third option to the interface quickly.

#### 3.2.8 Onboarding / Tips

Since the resulting design and application is quite complex, with several submenus, input modes, and editing capabilities, I decided to include an onboarding/tutorial page. The goal was to explain and condense all functionalities into one example page that should illustrate the features and show them in an exemplary setting.

This was especially important for the later user tests since it would hopefully jumpstart their writing process and allow them to explore the different features of the application. However, as we will discuss in the user tests chapter, this only partially succeeded as most users decided to skip or only glance over the tutorial.

It should feature both hand-drawn and typed content, as well as show off the editing, highlighting, and meta-data features, and would be included with every fresh install as

---

last accessed on 9<sup>th</sup> March, 2023

<sup>6</sup><https://developer.apple.com/documentation/uikit/uimodalpresentationstyle/pagesheet>, last accessed on 9<sup>th</sup> March, 2023

the first entry in the diary. Users would also always have the option to delete the entry if they no longer need it to clean up their diary.

However, during my own testing, I decided that only this tutorial page might not suffice for some of the more unique features, such as highlighting with the **marker** tool from the pencil toolbar. Thus, I sparingly included *tooltips/alerts* to inform users of certain functionalities. These would only be displayed once to the users and can be quickly dismissed to limit the possible annoyance of too many disruptive messages.

### 3.2.9 Implementation Limitations

Several of the discussed features and views were limited and constrained either after researching during the design phase or due to discoveries in the implementation phase and will be discussed in more detail in the next chapter.

The most significant limitation and challenge for design and implementation turned out to be the entry editor and the interplay between a rich text editor view and a canvas view for the pencil. Several strides were made to improve or alleviate some problems, such as suitable size calculation or switching between landscape and portrait mode and the resulting rearrangement of the drawn and typed context. Sadly, not all problems can be fully solved. Other features were cut or scaled down due to their complexity of implementation compared to their probable help in answering the research questions. This included - among others - streaks, reminders, stats, and a calendar overview page.

## 3.3 Outlook/Additional Features

(Research) diary apps and notetaking apps in general are primarily feature-rich, as we saw from the apps we examined in the previous chapter. So, during the design (and implementation) phase multiple features were planned, sketched, or thought of that were not included in the prototype application used for the user tests. However, end users would expect or appreciate many of them, as we will also discuss in the user tests chapter.

In this section, we are going to discuss some of these planned features and what they would entail.

### 3.3.1 Gamification, Motivation

As discussed in the previous chapter, diary/journal keeping of any kind can and often is quite a hard habit to start out with and keep over a long period. This is where a digital solution could help users reach their goals via different motivational techniques and reminders.

Timely reminders with encouraging messages that are ideally context-aware and/or could be set to specific times and days according to the users are a common and helpful way to encourage users to open the app and write a new entry. These reminders are pretty standard and can be found in many apps, e.g. in *GridDiary*.

Notifications in iOS can be pretty complex to implement, depending on the complexity of delivery, and might require remote servers for some features. However, locally scheduled notifications are also possible as a first step [8].

To help with writer's block on some days or for additional inspiration, a daily prompt or question could be directly displayed in the app or in a timely notification to help users write a new note for a day.

These reminders and goals could be further enhanced by methods of habit tracking and repeated, daily entries - similar to the popular concept of "*streaks*" used in other applications<sup>7</sup>, one of the most famous apps being Duolingo<sup>8</sup>. This streak counter could increase every time a new entry is written, or an existing one is edited each day or for each day the user has set as a goal and would reset back to zero otherwise. This method of "gamification" would encourage users to stay on their goal, especially if they have already managed to garner a long-running streak and don't want to lose it.

Other forms of encouragement or motivation could be a statistical overview of all created data and habits so far. These could be a character/word count across all entries, a calendar view of all entries, a graphical representation of e.g. mood/weather over a specific timeframe, and many more data points. These could serve as positive reinforcement for the users and as a way to look back at what they achieved and already managed to write down in their research diary.

All of these features were considered and discussed but abandoned in favor of other features for the prototype but indeed something that would require further investigation and testing for a final product.

#### 3.3.2 Document Export / Cloud Syncing

Document Export and the option to sync the diary's content with cloud services, such as Apple iCloud, was another consideration and discussed feature that often came up during user tests.

As a digital research diary grows over time, containing many important thoughts and information, and users pour more and more time and effort into them, it is increasingly essential for them to have better peace of mind about the security of their data in case they lose the data on their current device.

For these reasons, a final application would require a feature to allow users to sync their diary with a cloud storage solution to store their data in their cloud safely. This would allow them to access it on a different device or quickly transfer their data to a new one.

In addition, a general export in a different file format such as PDF or plain text files would be a useful feature for many users. It would allow them to use the created text

---

<sup>7</sup><https://www.zacfitzwalter.com/articles/how-streaks-motivated-me-to-exercise-100-days-in> last accessed on 9<sup>th</sup> March, 2023

<sup>8</sup><https://www.duolingo.com/>, last accessed on 9<sup>th</sup> March, 2023



in other programs, e.g. for further analysis in a more sophisticated coding software, such as the aforementioned **CAQDAS solutions** 2.5.1 or transfer to other note-taking applications or devices that don't support the application, e.g. a Windows computer [18].

### 3.3.3 Handwriting Detection

Handwriting detection that would bridge the gap between hand-drawn and typed content and allow users to easily copy and paste text from a hand-drawn note to other applications as with typed text was also considered.

This was planned because Apple Notes has offered this feature since iOS 15.0 thanks to their scribble technology which allows users to select drawn text easily and allows them to copy it as plain text<sup>9</sup>.

However, during the implementation, I discovered that this seems to be a private API not available for developers in other apps. Other apps like GoodNotes or Notability offer this feature, as well. However, that is not a readily available feature and requires a custom (machine learning) model to recognize handwritten text and translate it to a string. Thus, this idea was no longer pursued due to its complexity.

### 3.3.4 Content and Entry Management

For now, different entries can only be viewed chronologically or sorted by their tags. For a final product, it would be nice to introduce folders and subfolders to allow users to organize their entries and thoughts better, something which a lot of note-taking apps offer but not many diary apps. These could either be different notebooks or folders with encompassing subfolders.

The sidebar could also be further extended with a customizable appearance, allowing users to rearrange the overview and other potential "widget-like" views, e.g. a calendar view, current "streak", statistics, and many more. This would allow for better personalization, giving users the option to adjust the overview to their wishes and more easily access other views.

### 3.3.5 Multiple Editors

Many tested apps also featured the option to open another instance/window of the app to allow users to view two entries simultaneously. This was not done in my test application but could be added later on. Adding support for this feature would require supporting the multiple multitasking features available on the iPad, such as a split view or slide-over view[6]. This would require an adaptable layout and design, that would still be functional as a small slide-over view and in fullscreen. Designing and implementing such an interface would require additional testing and logic beyond this thesis's scope.

<sup>9</sup><https://support.apple.com/en-us/HT211774>, last accessed 9<sup>th</sup> March, 2023



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Implementation

## 4.1 Setup + Software Architecture

### 4.1.1 Chosen Programming Language + IDE

Swift was chosen as the programming language for this project and is also Apple's recommended programming language. Since its release in 2014, it has quickly overtaken Objective-C, its predecessor for developing iOS applications, and is used for most new apps solely intended for native iOS development. The User Interface was implemented with UIKit, as opposed to SwiftUI since a lot of the components and libraries used in the application are not supported by the latter as of the time of development.

Xcode was the obvious integrated development environment (IDE) of choice since it is the de-facto standard for developing apple's products, whether for iOS, iPadOS, watchOS, or tvOS devices and is also the only way to compile and distribute apps for these products.

There would also have been the option of creating a cross-platform app capable of running on iPads and Android phones and tablets. There are several frameworks that can be used to achieve this. React Native<sup>1</sup>, created by Meta, would be one such option, or the increasingly popular Flutter SDK<sup>2</sup>, released by Google in 2017.

However, in the case of this project and use case none of these cross-platform SDKs would have been a fitting option. Firstly, the developed app heavily uses the apple pencil capabilities and a specific library for *rich* text editing. Neither of these are either available in Android in the same fashion or would have required lots of additional custom

---

<sup>1</sup><https://reactnative.dev/>, last accessed on 9<sup>th</sup> March, 2023

<sup>2</sup><https://flutter.dev/>, last accessed on 9<sup>th</sup> March, 2023

code and fine-tuning for both Android and iOS to make them work in a similar fashion. E.g. the *CanvasView* used to detect and display Apple pencil input is part of the native iOS SDK "Pencilkit". This SDK would not have been supported by an Android device and would have required a completely different view and SDK for Android - if it would even exist in the first place.

Secondly, the project was designed and conceptualized for Apple's iPad from the beginning. It is designed with Apple's Human Interface Guidelines in mind and adheres to and uses standard iPad UI elements. Support for Android would require additional design adjustments and custom behavior that better suits the behavior and architecture of Android applications.

Last but not least, iOS and Swift were also chosen because I am experienced with native iOS development but not with cross-platform development.

### 4.1.2 Chosen Architecture

#### Architecture Pattern - MVVMC

An architecture pattern can be defined as a reusable solution to prevalent problems and challenges found in software engineering when designing the architecture of software projects. These can address various challenges and topics regarding software architecture and are not necessarily mutually exclusive to each other [44].

In this section, we will look at two architecture patterns that mostly concern themselves with *designing* interactive software solutions, i.e. programs with a user interface.

The chosen architecture pattern for implementing the practical part of this thesis is the so-called MVVM-C (ModelViewViewModel-Coordinator) pattern.

This is a more sophisticated approach compared to the standard Model-View-Controller (MVC) pattern which is still often used, especially in smaller projects. It is also officially encouraged by Apple as most of their provided frameworks and SDKs are built upon that concept<sup>3</sup>. This also explains why it is the default basic structure for new projects in Xcode (Xcode is Apple's IDE to develop macOS, iOS, watchOS, tvOS, and iPadOS applications).

This architecture might work great for smaller apps or projects that heavily use other patterns, like delegates or protocols in general. Still, more often than not, this architecture lends itself to a problem most iOS developers encounter in their career - the "Massive

---

<sup>3</sup><https://developer.apple.com/library/archive/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>, last accessed on 9<sup>th</sup> March, 2023

ViewController<sup>4</sup>. These are controllers whose codes stretches over hundreds of line of code, massively increasing in complexity and responsibilities over time.

This is why MVVM has increased in popularity as iOS apps, and their capabilities have become increasingly complex. It aims to create a better “separation of concerns” between objects and heavily regulate what they are responsible for. ViewControllers, for example, now only mostly contain the presentation, i.e. the UI and user interaction logic, instead of all the additional business and data logic [2].

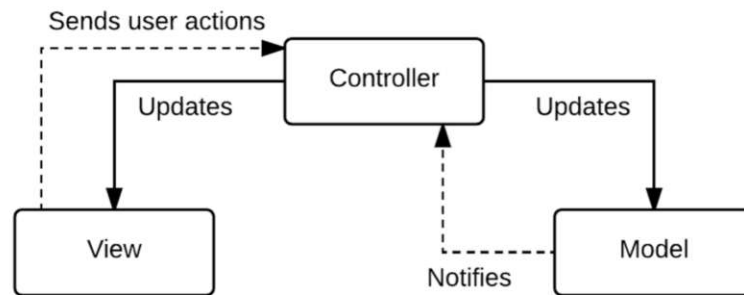


Figure 4.1: Apple’s version of MVC [2].

MVVM-C is an iteration on the MVVM software design pattern that tries to solve some of the latter’s issues with navigation/routing and dependency injection which can be a problem when using only the navigation options provided by the native iOS SDK, such as segues<sup>5</sup>. This can lead to a strong coupling of the ViewController with the navigation which may worsen its reusability and testability.

In the following section, we will look at the MVVM-C model and its components in more detail with a brief overview of each of its components.

### Model (M)

The model is mostly a plain representation of the data used in the application. This may include but is not limited to the data models used in backend requests and responses or data saved to the local storage of the device that the app is installed on.

It is vital that the model is as “simple” as possible and does not contain any business logic.

### View (V)

The view contains all the logic needed to display the desired and designed user interface to the app’s users and is in the case of iOS/iPadOS mostly part of the UIKit and often so-called ViewControllers. The view contains a reference to the ViewModel which it

<sup>4</sup><https://www.hackingwithswift.com/articles/159/how-to-refactor-massive-view-controllers>, last accessed on 9<sup>th</sup> March, 2023

<sup>5</sup><https://www.marcosantadev.com/mvvmc-with-swift/>, last accessed on 9<sup>th</sup> March, 2023

needs to access the data and logic it should present but also to propagate events - in most cases user interactions - to the ViewModel via so-called “bindings” which we will look at later.

### ViewModel(VM):

The ViewModel is the beating heart, the quintessence of the MVVM pattern, and contains most business logic. It is responsible that the view displays the right data in the right format and is also the main bridge between the view and the model, hence “ViewModel”. So, for example, it is responsible for fetching data from an endpoint. If there is a result or error, it tells the view via an event/signal to display the appropriate error message or to update the UI with the new data.

### Coordinator (C)

As mentioned before, the coordinator is mostly concerned with routing, i.e. showing the right views with the right dependencies in the current context. This “dependency injection” is used to “inject” the view with its corresponding ViewModel, for example.

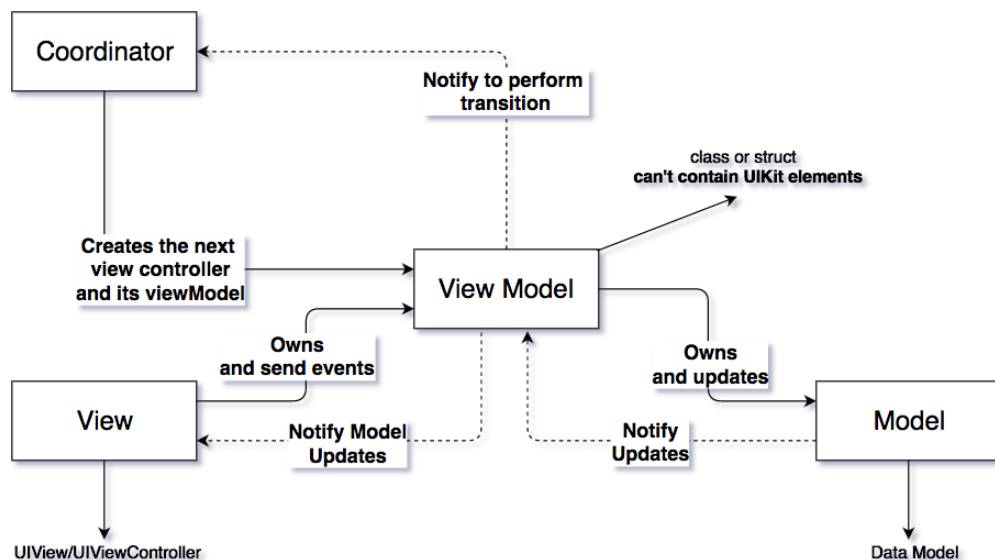


Figure 4.2: The MVVM-C(ModelViewViewModel-Coordinator) pattern <sup>6</sup>.

### Data binding

As previously before, the “binding”, i.e. the propagation of updates, like user interactions, etc., is very important in MVVM. In this project, this data binding has been done using RxSwift<sup>7</sup>. This reactive library has been implemented across different programming languages, e.g. RxJava or RxJs, and is very useful for this kind of event-based programming

<sup>7</sup><https://github.com/ReactiveX/RxSwift>, last accessed on 9<sup>th</sup> March, 2023

paradigm. An alternative in Swift would have been “Combine”, Apple’s official SDK<sup>8</sup>. The main reason for choosing RxSwift was the framework’s robustness since it has been published and constantly updated for several years more than Combine and my own inexperience with Combine.

### Reasons for choosing MVVM-C

As discussed in the previous sections, there are several reasons for choosing MVVM-C over Apple’s standard MVC. The separation of concerns is a huge factor in helping to keep the complexity of components, especially the ViewControllers, as low as possible.

Additionally, the data binding implemented with RxSwift is an effective way of propagating events and updates of the model through the ViewModel to the View. E.g., when the user updates the tags of an entry, they are easily updated in the *sidebar* when using this reactive approach. An alternative would have been a delegation pattern via Swift’s protocols. However, this pattern often creates unnecessary boilerplate code and more complicated UI and threading management.

The increased testability might not be a big factor for now, but since an app like this would grow in features, and complexity and be improved and further developed with more and more features, good support for creating both UI and unit tests is vital to guarantee the stability and functionality of already existing features with each change.

#### 4.1.3 Project / File Structure

As discussed in the previous section, the project architecture was designed based on the MVVM-(C) pattern which is also reflected in some parts of the file structure. On the root level, the folders - or groups as they are called in Xcode - were loosely separated based on the part of the app that they belong to or, if they are used across the app at several points, e.g. the various utility functions and extensions that can be found in the "Utilities" folder.

A good example of the aforementioned adherence to the MVVM-(C) pattern can be seen in the folder "Overview" in figure 4.3. Its subfolders are separated into "Views", containing all the basic, building block views of the UIKit, e.g. a `UITableViewCell` or a basic `UIView`.

These types of views are used as building blocks in so-called **UIViewController**s which usually contain an assortment of `UITableViews`<sup>9</sup>, which display data in rows with a single column, `UICollectionViews`<sup>10</sup>, which display a collection of data items in a more

<sup>8</sup><https://developer.apple.com/documentation/combine>, last accessed on 9<sup>th</sup> March, 2023

<sup>9</sup><https://developer.apple.com/documentation/uikit/uitableview>, last accessed on 9<sup>th</sup> March, 2023

<sup>10</sup>[https://developer.apple.com/documentation/uikit/views\\_and\\_controls](https://developer.apple.com/documentation/uikit/views_and_controls), last accessed on 9<sup>th</sup> March, 2023

flexible way using customizable layout guides or more static layouts using other container views using e.g. *UIStackViews*<sup>11</sup>. These **UIViewController**s can be found in the "ViewControllers" folder.

ViewModels which contain most of the business logic and are the bridge between Views and ViewModels are situated in the folder with the same name.

As discussed in the previous section, Models are often tied to a remote network call/response, e.g. the **WeatherModel** containing the response of the Weather API used in the application, or as a representation of the data saved to the disk, e.g. the *EntryModel* which contains all the information of one diary entry that is persisted on the iPad's storage. Since these models are also used across the app, it would not make sense to put them into one specific feature folder which is why they are grouped into the "Services" folder. "Services" also contains the so-called "Controllers" for functionality like the request of weather data. The job of these *Controllers* is to handle all the logic that is needed to request or persist the models associated with them. For example, it is the responsibility of the *WeatherController* to retrieve the weather data for a given set of coordinates from the weather API, handle and propagate possible errors and map the JSON response to the WeatherModel used in the app.

All the images and color assets used in the app are grouped together into a folder containing an asset catalog which can be identified by its unique file format, *.xcassets*. These assets catalogs are the recommended way of managing a vital portion of many apps: their assets, consisting of custom fonts, icons, images, and color assets<sup>12</sup>. These xcasset catalogs are an easy way to manage these files in one location and allow developers to e.g. set different icons and colors for the dark mode, among other things.

### 4.2 Input Modes Implementations

In this section, we will discuss the chosen approaches to implement user input detection with both the Apple Pencil and the (rich) text input, the difficulties I encountered, and my reasons for implementing it that way.

#### Pencil Input

The implementation of the pencil input itself was quite straightforward thanks to a native component offered by Apple's appropriately named SDK, **PencilKit**. It was first introduced on the *Worldwide Developers Conference (WWDC)* in 2019<sup>13</sup> and has

---

<sup>11</sup>[https://developer.apple.com/documentation/uikit/views\\_and\\_controls](https://developer.apple.com/documentation/uikit/views_and_controls), last accessed on 9<sup>th</sup> March, 2023

<sup>12</sup><https://developer.apple.com/documentation/xcode/asset-management>, last accessed on 9<sup>th</sup> March, 2023

<sup>13</sup><https://developer.apple.com/videos/play/wwdc2019/221/>, last accessed on 9<sup>th</sup> March, 2023



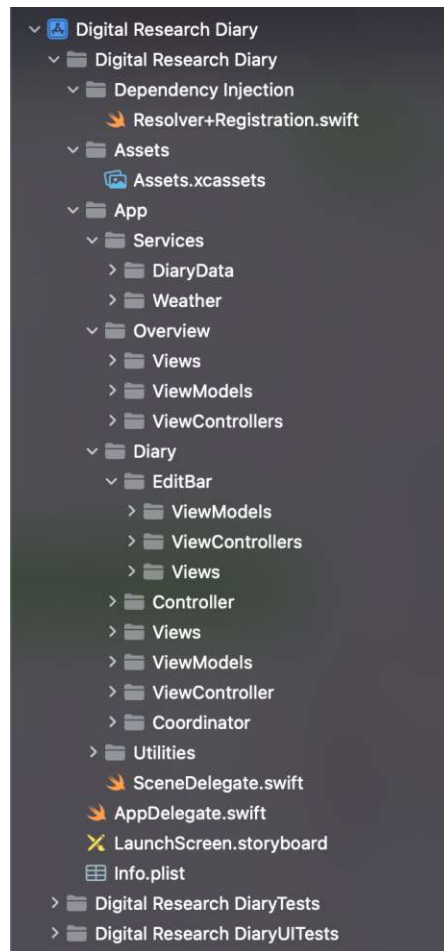


Figure 4.3: The project and folder structure of the developed application, as displayed in the chosen IDE, Xcode.

received many updates since then. It allows for easy, low-latency capture of pencil input. The main way of capturing this pencil input is via the **PKCanvasView**. This is a scroll view, meaning the content size can be larger than the frame size of its visible view. This application also uses this feature to always increase the content size after the user has drawn at the lower edges of the screen. This ensures the users have enough space to add more strokes to their drawings. The **PKCanvasView** offers a delegate protocol, called **PKCanvasViewDelegate**. This protocol allows classes to listen and react to certain events, e.g. when a drawing has changed or when a user has used a particular drawing tool from the toolbar. The applicable method in our use case is **canvasViewDrawingDidChange(\_: canvasView: PKCanvasView)**<sup>14</sup>. When this method is called, we can react to these changes and can be assured by the system that it

<sup>14</sup><https://developer.apple.com/documentation/pencilkit/pkcanvasviewdelegate>, last accessed on 9<sup>th</sup> March, 2023

is not right after every small change but, only after a user has completely finished their stroke. The implementation of this logic looks as follows:

```
func canvasViewDrawingDidChange(_ canvasView: PKCanvasView) {
    //...
    updateContentSizeForDrawing()
    //...
}

func updateContentSizeForDrawing() {
    let size = getOptimalContentSizeForDrawing()
    if size.height > currentRichEditorHeight {
        canvasView.contentSize = size
    }
}

private func getOptimalContentSizeForDrawing() -> CGSize {
    let drawing = canvasView.drawing
    let contentHeight: CGFloat

    // Adjust the content size to always be bigger
    // than the drawing height.
    if !drawing.bounds.isNull {
        contentHeight = max(canvasView.bounds.height,
            (drawing.bounds.maxY + canvasOverScrollHeight)
            * canvasView.zoomScale)
    } else {
        contentHeight = canvasView.bounds.height
    }
    return CGSize(width: DataModel.canvasWidth * canvasView.zoomScale,
        height: contentHeight)
}
```

The optimal content size of the drawing is calculated by getting the current content height of the canvas. If there is no drawing, set it to the size of the canvas view's frame. Otherwise, set the height to the result of adding the drawing's maximum height, i.e. the `maxY` value of its bounds, to the static constant `canvasOverScrollHeight`, taking into account the current zoom scale of the canvas view.

If this value is higher than the view's frame, it is used as the new content size. However, we use the rich editor view in tandem with the canvas view. This means that when switching between the two views the content size of one view is already set to the other's size. We only update this value if it exceeds the rich editor's content size. Other methods and logic are also called when `canvasViewDrawingDidChange` is called, such as intermittently saving the diary entry after a certain number of changes made to the entry, such as strokes on the canvas.

Another prominent feature of the PencilKit is the option to change between different drawing tools, such as a highlighter or a pencil. The settings of these tools can also be changed to allow for different line strengths or colors. This **PKToolPicker** view is automatically displayed when the user interacts with a view that is accepting pencil input and can be freely dragged around the screen<sup>15</sup>. Changes to the selected tool can be observed by implementing the `PKToolPickerObserver` delegate protocol. This protocol is also used to detect the selection of the highlighter tool, which serves a special purpose in the app.

The users can create highlights of parts of the entry they deem to be especially useful, interesting, or insightful and which might be relevant for the later write-up or coding of the research diary - a very important aspect of research diaries, as discussed by Engin [22]. The color of the stroke detects the category of a highlight. To give users easy access to an overview of all their written and drawn highlights, they are also created and stored when an entry is saved and can be viewed on a separate screen. The drawn highlights are saved as images to save resources and decrease complexity when displaying a potentially huge collection of highlights. This can be easily achieved thanks to a method of `PKDrawing` called `image(from rect: CGRect, scale: CGFloat)`. The necessary rectangle can be extracted from the `renderBounds` of a `PKStroke`, and I decided to use the full width of the screen as horizontal bounds in case users to increase the chance of capturing all the intended information that a user wanted to highlight in this snapshot.

Additionally, one approach at the beginning was to display an adapted tool picker that would better show the highlighting feature or remove it from the tool picker altogether and move the activation of the highlighting tool to a separate place. But as it turned out, an application cannot alter the standard toolbar and would require a fully custom-designed and coded toolbar. This would have been beyond the scope of this thesis. As such, I decided to make a similar approach by detecting when a tool has been changed to and from the highlighter and updating the user interface accordingly. Additionally, suppose the user activates the highlighting or drawing mode via the segmented control at the top of the screen. In that case, the selected tool is also manually set to the according tool in the toolbar.

Since iOS 15, a canvas view can also detect and extract drawn text from a canvas and allow users to copy it as a text to other applications<sup>16</sup>. However, as of writing this thesis, I could not find a source or SDK method that would allow programmers to access this functionality or manually kick off this text processing. It seems to be an internal method not yet publicly available. This would have allowed me to combine the written and "drawn" text more easily and would have made searching for specific highlights or text across diary entries more convenient for users.

<sup>15</sup><https://developer.apple.com/documentation/pencilkit/pktoolpicker>, last accessed on 9<sup>th</sup> March, 2023

<sup>16</sup><https://support.apple.com/en-gb/guide/iphone/iphd939f0a1c/ios>, last accessed on 9<sup>th</sup> March, 2023

This seemed to be the case for a lot of **PencilKit**, despite its first appearance already being more than three years ago. One can only hope that Apple will expand on its features, support, and documentation for iOS programmers in the future.

### 4.2.1 (Rich) Text Input

The implementation of the rich text input view went through multiple considerations. An important goal for the text input was to allow advanced text editing, allowing users to use different headline styles, add ordered and unordered lists, change text alignments, and much more.

This kind of rich editing possibility is sadly not possible with the standard input views offered by the UIKit SDK. Usually, text input is handled by the UITextView<sup>17</sup>. However, this view does not support the text editing and formatting capabilities that would be required in this project. This is because it can only display plain text. Live formatting and text styling capabilities can only be done with other views and custom solutions. In my research, the most common solution was to use a WKWebView to display a basic HTML page that can be used as a text editor. The webview either directly displays an editing toolbar or uses UIKit's support of so-called inputAccessoryViews<sup>18</sup>. These views can be optionally associated with input views. Whenever these views become the first responder, the accessory view is displayed over the system-supplied input view, e.g. the keyboard, allowing users more context-appropriate, custom ways to interact with the displayed content.

Apple also allows users to format their thoughts more extensively with their native app, "Notes", as we discussed in section 2.9.3. Sadly, it was not possible for me to find out if they were also using a web view or, more likely, a custom solution. Apple's apps are not open source and it is often up to the iOS programmer community to find out what they actually used to program their solutions.

Since developing this custom view was not the primary aim of this thesis, I decided to use a currently existing, open-source solution available on GitHub. The main selection criteria were as follows:

- How lightweight is the library? Is it bloated with many unnecessary features and files that are not needed for this application?
- Is it easy to customize? Can the views and edit toolbar easily be customized to my needs and use cases?
- Is it easy to implement and integrate? Can the library be added and used with a few lines of code or does it need a lot of fine-tuning and custom code to work with my application and needs?

---

<sup>17</sup><https://developer.apple.com/documentation/uikit/uitextView>, last accessed on 9<sup>th</sup> March, 2023

<sup>18</sup><https://developer.apple.com/documentation/uikit/uiresponder/1621119-inputaccessoryview>, last accessed on 9<sup>th</sup> March, 2023

- Is it buggy? Does it fulfill its feature promises and has no frustrating user interactions?
- Is it well-maintained? Are there a lot of open pull requests, and GitHub issues and how long ago was the last update of the library?

In the end, I decided to use the library **RichEditorView**<sup>19</sup>. This library is a more recent extension of another solution, also called "RichEditorView". It is based on the aforementioned WKWebView and offers an easy, lightweight implementation and high customization. It was also decently well-maintained and was ported to the latest Swift version. Some other considered options were "Proton"<sup>20</sup> and "AztecEditor"<sup>21</sup>. Proton is a very interesting approach for a rich editor. However, it was still in the early stages of development, and the last release was already some time ago. Additionally, the implementation and capabilities were too complex, and its feature set was too extensive for this project. Similarly, AztecEditor was too feature-rich and not as easily customizable for the desired use cases. It is designed and maintained by WordPress<sup>22</sup> and primarily intended as an editor for websites and editors supporting websites built using WordPress.

The integration of the *RichEditorView* library was as easy as promised and mainly consisted of adding the titular RichEditorView to the DiaryEntryViewController. Some customization was added to the view, such as making its containing web view's background color translucent and disabling horizontal scrolling which would have interfered with the combined input of the canvas view too much.

Changes to the editor's content and whether the user has started editing can also be detected thanks to its delegate protocol, RichEditorDelegate. This was used to save the editor's content in the diary entry's view model and occasionally save the entry on the disk.

The other main part of this library is the RichEditorToolbar. This toolbar is used as the aforementioned inputAccessoryView of the rich editor and is displayed over the keyboard. This toolbar is the main way to add rich formatting to the editor view and is easily customized. I customized the displayed formatting options by creating an enum that implemented the associated protocol RichEditorOption that allows setting the title, image, and action for a toolbar option. SF Symbols<sup>23</sup>, Apple's iconography library, was the source of most of the toolbar icons.

Similarly to the RichEditorView, the RichEditorToolbar also offers a delegate protocol to listen for specific events and actions. This was used to react when specific toolbar

<sup>19</sup><https://github.com/Andrew-Chen-Wang/RichEditorView>, last accessed on 9<sup>th</sup> March, 2023

<sup>20</sup><https://github.com/rajdeep/proton/>, last accessed on 9<sup>th</sup> March, 2023

<sup>21</sup><https://github.com/wordpress-mobile/AztecEditor-iOS>, last accessed on 9<sup>th</sup> March, 2023

<sup>22</sup><https://wordpress.com/>, last accessed on 9<sup>th</sup> March, 2023

<sup>23</sup><https://developer.apple.com/sf-symbols/>, last accessed on 9<sup>th</sup> March, 2023

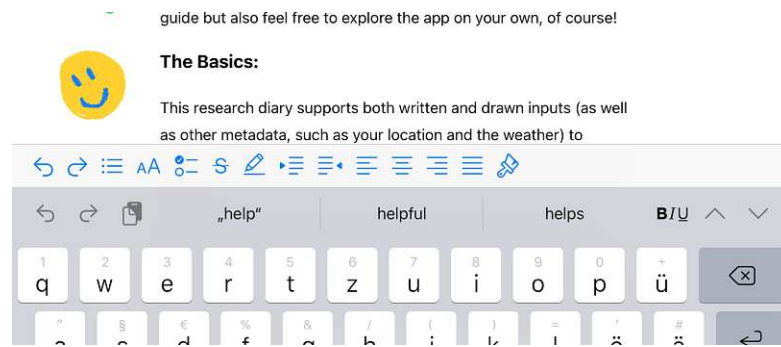


Figure 4.4: The custom toolbar "*RichEditorToolbar*", which is displayed above the standard keyboard. This toolbar contains all the rich editing options that users can choose from to customize their diary entry's layout.

options were pressed that needed custom interactions and extra work. In this project, this was used to detect when a user has pressed the highlighter, heading, or list option on the toolbar.

The native menu, offering the standard cut, copy and paste commands, that pop up when a user double taps or long taps on a selected text was also expanded to show the custom commands "add highlight" and "remove highlight". This offers users an alternative way of adding and removing highlights from a familiar interaction method. The highlight was set programmatically using a method offered by *RichEditorToolbar*, called `setTextBackgroundColor(_ color: UIColor)`. The color is based on the currently selected highlight category.

Additionally, since the *RichEditorView* is based on a *WKWebView* it was possible to "inject", i.e. executing, short JavaScript code which was used in multiple ways. E.g. to check the current heading size or list style of selected text.

Also, The current size of the web view was more accurate when using JavaScript which was very helpful for size calculation and matching the canvas view height when the size of the web view has changed. JavaScript was also used to increase the size of the web page's body to match the canvas view.

Finally, I also planned a more convenient highlight function that would allow users to highlight text simply by swiping/selecting it, without needing to press an additional button - similar to the highlight feature in most PDF viewers, such as *Adobe's Acrobat Reader*<sup>24</sup>. However, I could not find any existing solution, approach, or supported methods offered for either the *UITextView* or a *WKWebView*. Furthermore, since this is not a common feature in other similar note-taking apps it might have confused users, so I decided against further researching implementation ways.

<sup>24</sup><https://www.adobe.com/acrobat/pdf-reader.html>, last accessed on 9<sup>th</sup> March, 2023

### 4.2.2 Combination of Pencil and (Rich-)Text Input

Various other apps that we looked at in section 2.7 offer the option to input text with a (hardware) keyboard or with a digital stylus pen like the apple pencil. However, as far as I know, no app offers free-flowing rich text input in the same input area as the pencil input while being able to switch between these two input modes at any time.

Take the official Apple Notes app for example. It does allow switching between pencil and keyboard input, however, the drawing is strictly separated from the text and it is impossible to, e.g. scribble over text, as seen in figure 2.13. This user experience is far different from traditional media, where one might regularly scribble, underline or add arrows and drawings right next to a text. So my goal was to implement a prototype that would recreate this experience.

This would mean that these two input modes and their accompanying views would need to be overlapping and users would need to have an easy way of switching between the two. This mode switching has gone through various iterations, as discussed in section 3.2.7.

The rudimentary approach was to overlay the pencil input, i.e. the `CanvasView`, with the text editor view and switch their order in the view hierarchy depending on the current input. This is easily achievable by calling the method `bringSubviewToFront` in their containing `ViewController`, in this case the `DiaryEntryViewController`. This method changes the order of a given subview to display it on top of its sibling views.

It was also essential to make sure that the input view that now would be on the top of the view hierarchy to also become first responder. A first responder is a concept in UIKit that basically handles and defines which view will first react to an UI event [26]. A typical example would be for a `UITextView` to become first responder to react to keyboard input events first. This can be achieved by calling the methods `resignFirstResponder` and `becomeFirstResponder` to ensure that the correct view is the first responder and other views resign their right as first responder.

As we will discuss later in the following sections, this approach came with a host of problems and implementation challenges; not all of them were solvable while developing the application for this thesis.

#### Pencil Detection

Initially, I planned to detect pencil input while the text input mode is active, i.e. the rich editor view is in the foreground, and automatically change the input mode to the drawing mode. This would require detecting user inputs, especially a "tap gesture" and checking whether it was a tap with a finger or pencil. This would have been easily achievable thanks to the existence of a specific recognizer for this kind of gesture, called

UITapGestureRecognizer<sup>25</sup>. As the name suggests, this recognizer can be created and added to a view to listen for single or multiple taps. Furthermore, it can be configured to only listen to pencil touches by setting the `allowedTouchTypes` property to only "pencil" `UITouchTypes`.

However, this and other similar approaches to recognizing taps and other gestures did not seem to work for my project. This might have been due to the fact that the rich editor view was already set to be the aforementioned "first responder" and did not correctly relay the tap events to other views. I also tried to detect these events already in the `RichEditorView` component by forking the library, but even then the pencil taps were not always properly recognized.

The iPad also supports an easy way to input text with the Apple pencil, called "**Scribble**"<sup>26</sup>. This allows users to hand-write letters with their pencil which are automatically translated to text. Automatically detecting pencil input and switching to the drawing mode would have also interfered with that functionality. As we will discuss in the section , most users that are writing with the pencil, prefer to use their handwriting instead of waiting for the system to translate it to text, with the potential of not recognizing their writing. However, they might want to use the scribble mode's revising features for other text, allowing users to easily delete or select text, among other things. Additionally, such an automatic change might be confusing or unintuitive to users without good visual and/or haptic feedback when switching modes.

Because of all these reasons, I decided against pursuing implementing this feature further and prioritizing other functions of the application.

### 4.2.3 Synchronization of input views / View Size Calculation

Since the drawing view, i.e. the canvas view, and the rich editor view were layered on top of each other, it was vital to make sure that these views stayed synchronized in their scrolling position and view height when the user switched between drawing and text mode.

This means that the content height of the two views needs to stay the same when the input mode changes to avoid weird scrolling behavior that would only move the drawn portion, possibly resulting in unwanted overlapping of text and drawing.

As discussed in the previous section, to ensure that users can always add drawings to a long text or vice versa, the content size of the two needs to stay roughly the same. This means that each time when a user switches between drawing and text input, the size of the two views needs to be calculated and the smaller view needs to increase in size, if applicable. In the previous section, we looked at the canvas view's content size

---

<sup>25</sup><https://developer.apple.com/documentation/uikit/uitapGestureRecognizer>, last accessed on 9<sup>th</sup> March, 2023

<sup>26</sup><https://support.apple.com/en-gb/guide/ipad/ipad355ab2a7/ipados>, last accessed on 9<sup>th</sup> March, 2023



calculation. The size of the web view was calculated by "injecting" JavaScript code into the underlying web view of the rich editor since it led to far more reliable results than using the content size property of the webview. The evaluated property was the `scrollHeight` of the current HTML document<sup>27</sup>. This property was then compared to the content height of the canvas view, and the smaller view's height was set to that of the taller view. In the case of the rich editor, padding at the bottom was added via JavaScript to make sure that the whole view could be selected and new text could be added.

Another drawback of this method is that users can not easily change between portrait and landscape mode, i.e. rotate their device, when using both drawn and text input. Of course, it is possible to rotate the device and use the app in landscape and portrait mode to accommodate for different user needs. Thanks to the auto layout constraints set for both views, they automatically adapt to these changes and take up the according screen size. *Auto layout* is a basic concept of Swift's UIKit, that calculates the size and position of views according to the constraints set for them<sup>28</sup>. The canvas view's constraints are set to always take up the entire screen below the toolbar view, while the rich editor is set up to take up around 62% of the screen's width, resembling the "*golden ratio*"<sup>29</sup>. This allows users to draw on the sides of the text more easily while also making the reading and writing experience on larger iPads more enjoyable. However, since the aspect ratio changes when rotating the device, so does the size relation of the two views change. That means drawings that would not have overlapped with the text in one rotation, might do so when rotating.

#### 4.2.4 Toolbar

Since one of my main research questions was also concerned with inquiring whether users of research diaries would use and appreciate various metadata information that they could easily include with a digital diary, I implemented a toolbar that allows users to quickly add this data, among other things.

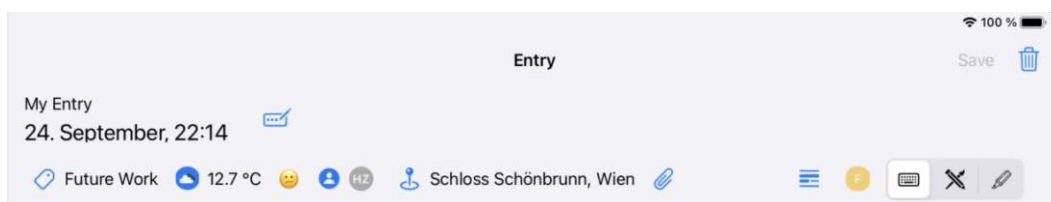


Figure 4.5: The custom-designed and implemented toolbar, displays all relevant information of the currently selected diary entry as well as the options for input switching.

<sup>27</sup>[https://www.w3schools.com/jsref/prop\\_element\\_scrollheight.asp](https://www.w3schools.com/jsref/prop_element_scrollheight.asp), last accessed on 9<sup>th</sup> March, 2023

<sup>28</sup><https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutoLayoutPG/index.html>, last accessed on 9<sup>th</sup> March, 2023

<sup>29</sup><https://www.canva.com/learn/what-is-the-golden-ratio/>, last accessed on 9<sup>th</sup> March, 2023

This "toolbar" is a custom UIView that is displayed on the top of an entry screen, layered above the editor and canvas view. This view allows users to edit all relevant additional (meta-)information and data of the entry, as well as allowing users to switch between the input modes, both of which we will discuss in the following sections in more detail.

### Entry (Meta-)Data

The first additional info that can be added and edited in an entry is the option to change its name by tapping on the title view in the upper left section of the toolbar view. By tapping on the view, a text field is displayed that allows users to update and/or add a custom title to an entry.

Most other metadata options can be found at the bottom of the screen. This collection of views was implemented as a UICollectionView since this type of view allows its subviews to be horizontally scrolled. This means that is both easier to add new metadata options to these views as well as making sure that even on smaller screens, all options can always be accessed.

The currently available options are as follows:

- **Tags:** users can create and add tags to an entry. This allows them to later search for entries containing them in the sidebar, as well as group them by their content. This screen also serves as the tag management view where people can add and remove tags for the entire diary.
- **Weather:** users can add and update the weather data of an entry. This is achieved by calling the OpenWeatherMap API using the users' current longitude and latitude<sup>30</sup>. Additionally, this API is used to get an image representation of the weather conditions that are displayed next to the temperature.
- **Mood:** users can represent their mood when they created or worked on the entry by selecting one of five emojis, ranging from quite unhappy to very happy.
- **People:** users can also add people that were mentioned or relevant to a certain entry. As Browne pointed out, one main advantage and use-case of a research diary is that it should be a collection and documentation of all the work and possible fieldwork of a research project [14]. The people they met with and keeping their information is a very relevant part of that. Currently, people can only be added by selecting them from their contact list. However, in the future, it is planned to create a contact in the application directly.
- **Location:** the location where the entry was created is automatically saved when it is created, but only when users give permission for the app to access their location. Since location data is very sensitive, Apple is quite restrictive about apps

---

<sup>30</sup><https://openweathermap.org/api>, last accessed on 9<sup>th</sup> March, 2023

accessing this data and requires explicit user consent<sup>31</sup>. This can be done with the `CLLocationManager` method `requestWhenInUseAuthorization`<sup>32</sup> which displays a popup requesting the location access. Users can also update or delete a location at any point. The name of the location is derived by sending a reverse geocoding request to the `CLGeocoder`<sup>33</sup>. This converts the users' location from coordinates to a human-readable format. This format may contain a country, city, street, and region, as well as a corresponding landmark or other points of interest. In the app, I chose to display an area of interest first and if there is no applicable area, the placemark's name, which most often corresponds to a street name, is displayed.

- **Images/Media:** since one of the main benefits of a digital research diary is the easy addition of other digital media, such as images or videos, instead of e.g. having to print a photo to add it to a physical diary, I wanted to include the option to add other media to an entry. I first planned the option to directly embed images and maybe even videos in the text or canvas view. However, the canvas view does not support including images, and the rich editor view did not support adding images in a suitable way for my application.

So, I decided to include an additional screen and tool in the toolbar that allows users to add images from their gallery. These images are saved alongside an entry's information. As we will discuss in the user tests, a future improvement of this feature would be to make the already added images more visible and discoverable to the user and also show thumbnails of them in the sidebar underneath an entry.

Most of these views display additional content and views as a "popover" controller. This is a presentation mode reserved for iPadOS that displays temporary views and interfaces on top of the currently displayed content<sup>34</sup>. Apple recommends using this type of presentation for content that is only shortly displayed and can be dismissed when no longer needed, such as configuration options or other tools. This popover should be anchored to a specific view. In our case, this would be the icon in the toolbar the user pressed. A popover can also easily be dismissed by tapping outside its bounds, making them less intrusive and easily removed when no longer needed by users while taking up less of the screen real estate.

If users allow the app to access their device's location, the weather and location are automatically added to a newly created entry.

<sup>31</sup>[https://developer.apple.com/documentation/corelocation/configuring\\_your\\_app\\_to\\_use\\_location\\_services](https://developer.apple.com/documentation/corelocation/configuring_your_app_to_use_location_services), last accessed on 9<sup>th</sup> March, 2023

<sup>32</sup><https://developer.apple.com/documentation/corelocation/cllocationmanager/1620562-requestwheninuseauthorization>, last accessed on 9<sup>th</sup> March, 2023

<sup>33</sup><https://developer.apple.com/documentation/corelocation/clgeocoder>, last accessed on 9<sup>th</sup> March, 2023

<sup>34</sup>[https://developer.apple.com/documentation/uikit/windows\\_and\\_screens/displaying\\_transient\\_content\\_in\\_a\\_popover](https://developer.apple.com/documentation/uikit/windows_and_screens/displaying_transient_content_in_a_popover), last accessed on 9<sup>th</sup> March, 2023

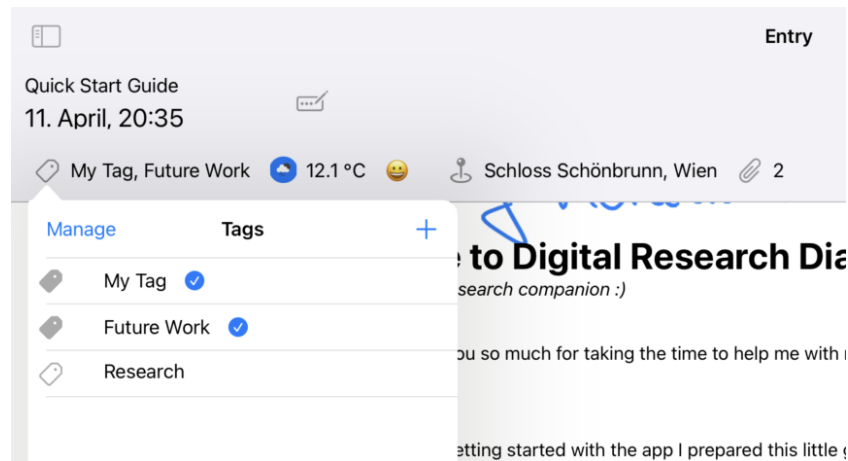


Figure 4.6: The tags screen is displayed as a popover screen, making it easier for users to dismiss when no longer needed, as well as the source view being easily identified.

### Input Switch

The input switching functionality has been through various iterations before arriving at the final one, as we already discussed. In the end, I settled on a switch that is always displayed on the right side of the toolbar.

This switch was implemented using the so-called `UISegmentedControl`<sup>35</sup>. A segmented control may contain several, horizontally aligned child views, each serving as a button. The main advantage of this approach is that the currently set value is always visible alongside its alternative settings, as opposed to earlier implementations, where it was just a single button that changed its icon based on the currently active input method. Additionally, a segmented control allows more than two different states, allowing me to add a separate state for the highlighting mode.

The segmented control also reacts when the user selects the highlighter/marker tool from the `PKToolPicker` and changes the selected segment accordingly. This is done thanks to the aforementioned `PKToolPickerObserver` protocol method `toolPickerSelectedToolDidChange` that allows us to observe changes to the selected tool.

#### 4.2.5 Navigation

The navigation was based on the "sidebar" navigation paradigm, as defined in Apple's "Human Interface Guidelines" [7]. This is a typical navigation layout on iPad applications, especially Apple's pre-installed apps, such as "Mail" or "Files". A sidebar consists of

<sup>35</sup><https://developer.apple.com/documentation/uikit/uisegmentedcontrol>, last accessed on 9<sup>th</sup> March, 2023

several app navigation areas, displayed side-by-side in a so-called **split view**. A split view manages multiple adjacent views. These views can be composed of components, from table views to an image or a collection view. A sidebar is a very common part of such split views.

When a user interacts with an item in a sidebar, it is most often expected that this item's details are shown in the primary view of the split view. Or, if a split view consists of three adjacent views, the leftmost, primary view often shows items that contain an array of items, such as a folder with notes and the secondary view displays this array. Interacting with an item in the secondary view then displays the item's details in the tertiary, rightmost view.

This navigation hierarchy was implemented using the so-called `UISplitViewController`<sup>36</sup>. This view controller manages its child view controllers in a horizontal hierarchical interface with view controllers leveraging the navigation of other controllers. This `UISplitViewController` is set as the application's root view controller since it is responsible for displaying the outermost view controllers. It has no actual custom appearance since this is defined by the view controllers it displays. In this case, the `DiaryEntriesSidebarViewController` is on the left, and the `DiaryEntryViewController` is on the right.

The `DiaryEntriesSidebarViewController` offers a chronological overview of all the user's entries, as well as a button for adding new entries and a tag filtering section. The tag filtering was closely modeled after Apple's "Notes" app, as described in section 2.9.3. This allows users to quickly filter and look for entries with specific tags, enhancing the organization capabilities of their diary.

The `UISplitViewController` supports multiple arrangement options for its child view controllers, by setting its `preferredDisplayMode` property. In this project, mainly two arrangement options were chosen. If the iPad is in landscape mode, it is set to `oneBesideSecondary`, meaning that the sidebar is always displayed side-by-side with the secondary view controller. In portrait mode, it is set to `secondaryOnly`, i.e. only the secondary viewcontroller which contains the diary entry is displayed. This is done to better use the more limited horizontal screen real estate in portrait mode.

Starting with iOS 14, the `UISplitViewController` supports up to two sidebars, but currently, only one sidebar is used. However, this would make it easy to support e.g. creating multiple research diaries and letting users navigate between them in the outermost sidebar, similar to how the "Notes" app lets users switch between different folders.

In the future, it would make sense to allow users more customizing options and display additional, informational views here. Such as the aforementioned, motivational trackers that other apps display, such as "Journey". Furthermore, letting users rearrange, add, and delete them would allow for a better, personalized experience. Finally, if an iPhone app is considered in the future, this layout might need adjustments. Apple recommends

<sup>36</sup><https://developer.apple.com/documentation/uikit/uisplitviewController>, last accessed on 9<sup>th</sup> March, 2023

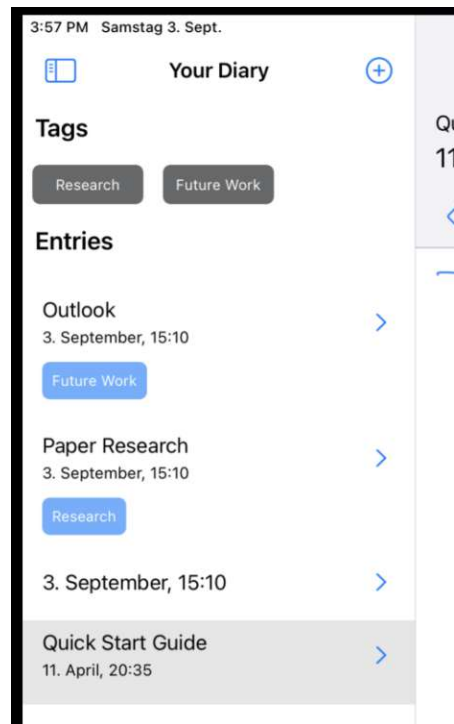


Figure 4.7: The sidebar as it appears in the app. It consists of a list of entries that shows its title, creation date, and the diary's tags, and the "hide" and "add new entry" buttons on the top.

using the more common tab view [7]. This is because of the aforementioned need for bigger space on the screen - which is less of a problem on larger iPads - and because it allows users to switch between the top-level navigation more easily on the phone.

#### 4.2.6 Learnings

Some of these drawbacks regarding view synchronization could have probably been prevented or at least significantly decreased in complexity by using the native `UITextField` instead of a custom rich editor, based on a `WKWebView`. This is for multiple reasons. Firstly, the size calculation would have been much easier with a traditional text field since there is no additional HTML and CSS rendering logic. This would have made switching between drawing and keyboard mode easier. Secondly, the document export would have also been more manageable since it seems that the web view is not rendering all of the web page when it is loaded, or is at least not aware of the total size of the page.

Thirdly, layering the canvas view and rich editor view has led to most of the implementation challenges in this application. While it was quite an exciting challenge, both from a design and implementation perspective, I am not sure if it was the best approach. However, to my knowledge, there is no other iPad notes or diary app with a similar

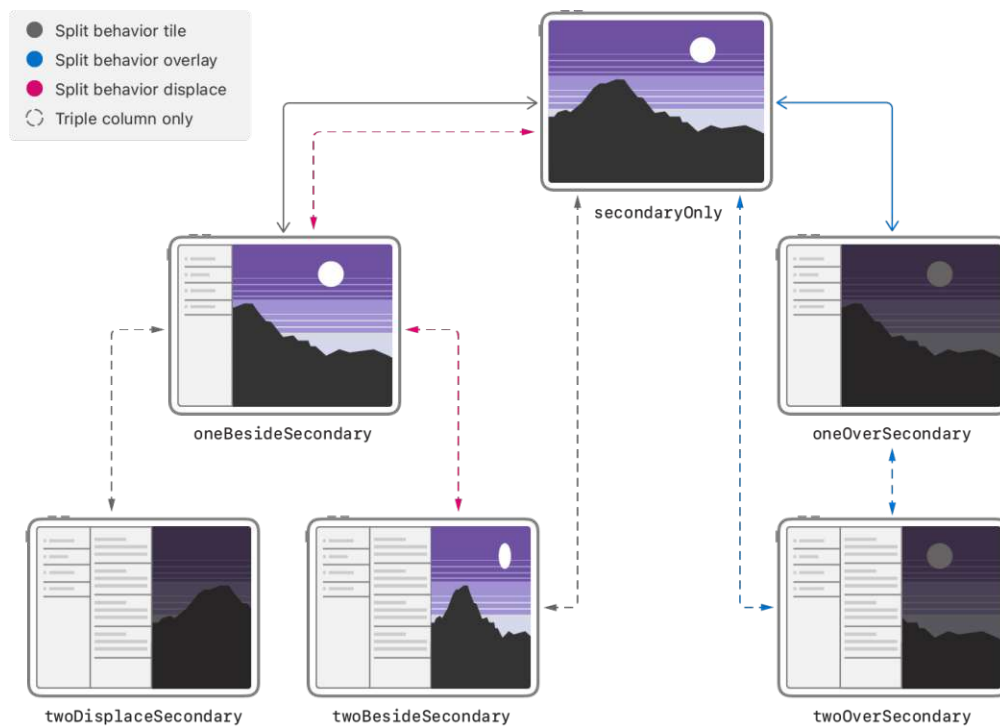


Figure 4.8: The many possible display mode settings of a UISplitViewController, allowing full customization of when to display which sidebar<sup>37</sup>

design approach which makes this project more unique and worthwhile both as an app on the Apple app store as well as a research project since it allowed me to test how users are making use of the different input methods, especially the apple pencil and keyboard, and whether they are using them in combination.

However, I still decided against using a text field because it would have meant losing all the rich editing capabilities that one would expect from a fully-fledged diary app that contains entries that might stretch over multiple pages. With more time and experience in web and mobile development, I am sure some of the problems could have been ameliorated. Still, all the aforementioned methods have not yielded entirely satisfactory results. Additionally, both the Swift image export API and the native UITextField are still sorely lacking in the use cases needed for this project. Especially the PDF API in Swift is quite convoluted and hard to use.

### 4.3 Document Export

A common and often required use-case for (research) diary applications is the option to export a created entry or the entire diary, e.g. as a PDF or other document formats. Consequently, one goal of the implementation was to create an export feature that allows users to create PDFs of their diary, e.g. use them in other applications, back them up on

the cloud, or share it with others.

The document export was implemented using the `UIScreenshotServiceDelegate`<sup>38</sup>. This protocol listens for user screenshot requests, commonly done with the hardware buttons on the iPad. When a user requests a screenshot, the method `screenshotService(UIScreenshotServiceDelegate generatePDFRepresentationWithCompletion: (Data?, Int, CGRect) -> Void)` is called. This method requests the app to create a PDF representation of the current content on the screen and allows users to save and edit the PDF in the native screenshot pop-up view.

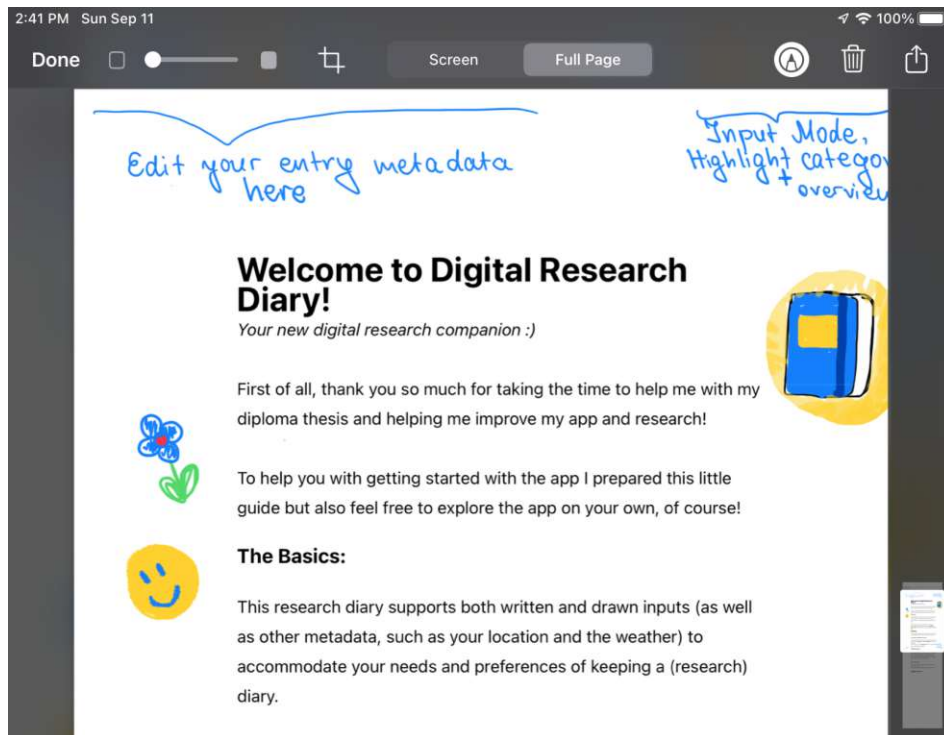


Figure 4.9: The PDF preview that pops up after users request a screenshot of an entry. Users can edit, markup, share and save the PDF on their device or the cloud.

Admittedly, this is a relatively niche part of the typical iOS screenshot functionality, that most users are unaware of or would be expecting to be there in the first place - as we will also discuss in the user tests section.

This functionality could have been further expanded with a dedicated button on the entry toolbar view to either export one entry or the whole diary as a PDF. However, implementing the PDF creation was not as easy as initially expected.

<sup>38</sup><https://developer.apple.com/documentation/uikit/uisccreenshot servicedelegate>, last accessed on 9<sup>th</sup> March, 2023



Creating a PDF representation of the canvas view was easily done, even when the content was spanning over multiple pages. This was done by creating a PDF rendering context using the method `UIGraphicsBeginPDFContextToData`. In this rendering context, data can be added by rendering or drawing views and/or images. For example, a view can be rendered by calling `view.layer.render(in: context)`. For the canvas view, better results were achieved by iterating over its contents, strip for strip at a fixed distance. At every step, an image representation of the drawing in the canvas view was created, and this image is drawn into the context at the correct render location.

The challenge was to ensure that not only the content of the canvas view was included properly over multiple pages, but also the content of the rich editor view while not rendering any of the toolbar or the sidebar. I could not correctly align the rendered image of the drawing with the editor's content. Either view not in the foreground was always cut off, which might be due to optimization. Rendering each view separately onto the context also did not seem to be working as the positions and coordinates of the editor content, and canvas drawing always were out of sync and drawn into the wrong position. Apparently, when converting the drawing object to an image, it uses a different coordinate system, and fusing these two together leads to misaligning the content.

Just rendering the appropriate parts of the parent view, containing both the canvas view and rich editor view, was also not an option since this only seemed to work for the currently visible parts of the view, but not the parts above or beneath it. Either the canvas or the text was always cut off or not visible when using this approach.

This has led me to not further work on this feature, since it seemed like the two views were incompatible with being rendered simultaneously with the same drawing coordinates. This might be due to battery or other energy optimization done by the system, and I could not find a similar problem or solution that could have pointed me in the right direction. The `UIGraphicsPDFRenderer` API is quite convoluted and render options beyond just rendering a view or an image into the context are very limited and do not seem to be very suited for my use-case, especially when using two scrollable views that apparently do not render their complete view beyond what they are currently displaying.

## 4.4 Highlighting Feature

As mentioned before, the drawn highlights are saved as an image snapshot of their location on the image. However, it was impossible for me to add the text that a user might want to highlight with their pencil to the created image. While `CanvasView` offers an easy and intuitive method for creating images with `PKDrawing.image(from: CGRect)`, the same cannot be said for creating an image of two different views. There is a general method offered by iOS, however, it did not produce the necessary results in this case. While we do know the bounds of the stroke in the drawing, the coordinate system and bounds seemed to be different from the rest of the views, and I was not able to translate the coordinate system in a way that allowed me to generate a snapshot of the intended target location in the rich input view. This might also have to do with the fact that

the underlying view of the rich input was a web view, which might render the text in a different way that did not allow me to calculate the correct coordinates.

## 4.5 Data Persistence and Cloud Storage

Currently, all of the user-created data in the app, i.e. the created diary entries and metadata, such as the tags used for the aforementioned entries, are only saved on the user's local storage. To be precise, the entries are saved in the file system using Swift's natively provided **FileManager**. This FileManager allows easy access to the file system and is also contained in a *sandboxed* environment specifically created for each app on the iPad [27]. This means that they are contained in their own environment on the file system where they cannot easily interfere with other apps and their created files. The diary file model which consists of an array of entry models is saved as a .data file in the documents directory of the app's sandbox environment. The app also contains a default diary .data file which is loaded and saved on a new install of the app if there is no diary already saved to the local storage. This default diary contains a tutorial to guide and explain to new users.

In order to not block the user interface by performing these load and save operations on the main thread, all data loading is done a background thread. This is achieved via a so-called **DispatchQueue**<sup>39</sup>. A DispatchQueue is one of the main ways to handle concurrency in Swift and allows apps to perform and execute tasks serially or concurrently either on the main or a background thread.

The files were saved in the *property list* format on the disk [29]. This is a file format primarily used by apple and can also be found when creating a new project in Xcode. Each bundle in an Xcode project contains a info.plist file, which contains info about how the system should interpret, compile and build this bundle<sup>40</sup>. A property list file stores hierarchical data in the form of named values and arrays of values of various types. The data format is based on the XML format. They offer a structured and performant way of organizing hierarchical data and are supported by various Apple APIs which allow quick and easy serialization and deserialization of this data for further use in applications.

The tags and other metadata are saved in the *UserDefaults*<sup>41</sup>. UserDefaults is a hierarchical key-value store that is optimized to persist user settings, such as their preferred unit of measurement or in our case the tags used to categorize diary entries. The data

---

<sup>39</sup><https://developer.apple.com/documentation/dispatch/dispatchqueue>, last accessed on 9<sup>th</sup> March, 2023

<sup>40</sup>[https://developer.apple.com/documentation/bundleresources/information\\_property\\_list](https://developer.apple.com/documentation/bundleresources/information_property_list), last accessed on 9<sup>th</sup> March, 2023

<sup>41</sup><https://developer.apple.com/documentation/foundation/userdefaults>, last accessed on 9<sup>th</sup> March, 2023

is saved to the *defaults system* database by assigning values to custom keys set by the application.

This also includes other preferences like the last selected input mode and the last opened diary entry to make sure that the same mode and entry are selected the next time a user opens the app. This is also why it is called `UserDefaults`. An app should load the defaults set by a user at startup to make sure it behaves and looks like the user expects by default based on their set preferences.

### 4.5.1 Cloud Storage

This data persistence is currently done locally on the user's iPad. The biggest reason for this is the increased complexity of implementing such a feature for the various Cloud Services, e.g. iCloud or Google Drive. The developed solution is still mostly a prototype in many ways. This is also a much-requested and expected feature in such an app, as we will discuss in the user feedback from the design section. Most people trust cloud storage solutions to store and access their data safely across devices. Consequently, they are also expecting apps to support such cloud features, especially in the case of a research diary app, where data loss would mean potential hours upon hours of lost time and effort.

So one important feature for a more final version of the app would be to include support to upload and backup files on the cloud. This would be especially useful when using the digital research diary app on their other apple devices, such as their iPhone or MacBook, however, the main goal of this thesis was to create only an iPad app, and a phone or mac version would have required their own design and implementation.

### 4.5.2 Conclusion/Discussion

For the most part, as we will see in the user tests, the developed application can already be used as a complete digital research diary and offers an extensive feature set, with different input methods, a highlight and markup feature, and much more.

However, at the same time, it is still very much a prototype, thanks in many parts to the chosen layering of the two input methods. A choice that no other diary application I am aware of has implemented similarly - probably for a good reason. The document export, more reliable input switching, and better synchronization of text and drawings would need to be overhauled and fixed for a final solution ready for the app store. This is in addition to other design issues and improvements made apparent in the user test sessions, as discussed in the next chapter. I am still very proud of the results and the choice to implement it this way has offered an exciting challenge. It has helped me gain more knowledge of iOS and Swift APIs that I have not used before in my experience as an iOS developer. Some of these libraries like the `PencilKit` and the PDF rendering still are quite poorly documented and not always easy to use which might also be due to the

#### 4. IMPLEMENTATION

---

fact that they are not very commonly used or required APIs for most iPhone and iPad apps.

# Evaluation

## 5.1 Selection of Participants

To get the most valuable insight from the user study, it was important to me to select participants that were already at least somewhat familiar with the concept of research diaries and ideally already have written their own research diaries.

Additionally, I tried to recruit participants with different ranges of experience with the iPad and whether they preferred keeping a digital or a traditional, analog diary. Another limiting factor was that the developed app required an iPad, ideally with an Apple Pencil and/or hardware keyboard to test all of the application's features properly. Thankfully, some hardware was available for participants to borrow and others already possessed an iPad.

This of course meant that the possible number of participants was quite limited. In exchange, I opted for a more time-intensive user research method to gain more extensive and thorough feedback from my study participants.

Participants were mainly recruited with the generous help of my advisor, Professor Purgathofer. He was also helping with organizing the additional hardware for the user studies. Additionally, it was important for me to gather participants from interdisciplinary fields to gain more valuable and broader insights.

All these limiting factors guided my choice of user research methods for my user trial which will be discussed in the next section.

## 5.2 Chosen Methods

The selection of the most suitable user research methods was based on multiple considerations.

Firstly, it was important to me to gain an understanding of how users would actually use the digital research diary in a more real-world application, so I decided against user research methods that would only test certain interactions in a specific setting, e.g. a usability-lab study or focus groups [40]. I wanted to see how users familiar with research diaries would use the application and gather their thoughts on it.

This would have also been accomplishable with a field study by observing participants during their fieldwork and how they would take notes in their diaries after or during their research. I decided against this method because it would have been even harder to find participants I could reliably observe during their work. It would have also most likely affected their note-taking habits while being watched. It would not have led to the most valuable insights in my eyes. Additionally, as one might expect judging by the name, a research *diary* is a very personal tool, and for many, it might have been seen as too intrusive.

### 5.2.1 Research Diary

A diary study allows for a similar experience, although one more reliant on the description of its participants and their generated artifacts and thoughts. The Nielsen Norman Group also fittingly describes it as a "poor man's field study" [41]. Because of all these aforementioned reasons, I decided that a research diary would be the most fitting method to use as the basis for my user tests, despite concerns such as possibly influencing existing note-taking behavior or possible inconsistencies in data gathering of the participants [33].

Additionally, to gather more of my participants' insights and experiences with the application, I planned a one-on-one follow-up post-study interview with each participant after a test period of around two weeks. I prepared a consent form in order to be allowed to record the audio of the interview for later analysis and to gather informed consent. Moreover, while I prepared an interview guide to help structure the interviews around my key research questions, I mostly decided on a loose, unstructured interview format to keep the conversation more natural.

### 5.2.2 Questionnaire / System Usability Scale (SUS)

Furthermore, it was important to me to better quantify the experiences and assessments of the developed software. This led me to use a de-facto industry standard questionnaire for quantifying software, the **System Usability Scale**, or in short **SUS** [31].

The SUS contains ten questions asking participants to rate how much they agree with statements about the usability of an evaluated system. The answers are ratings from one to five on a *Likert* scale, ranging from "strongly disagree" to "strongly disagree". These

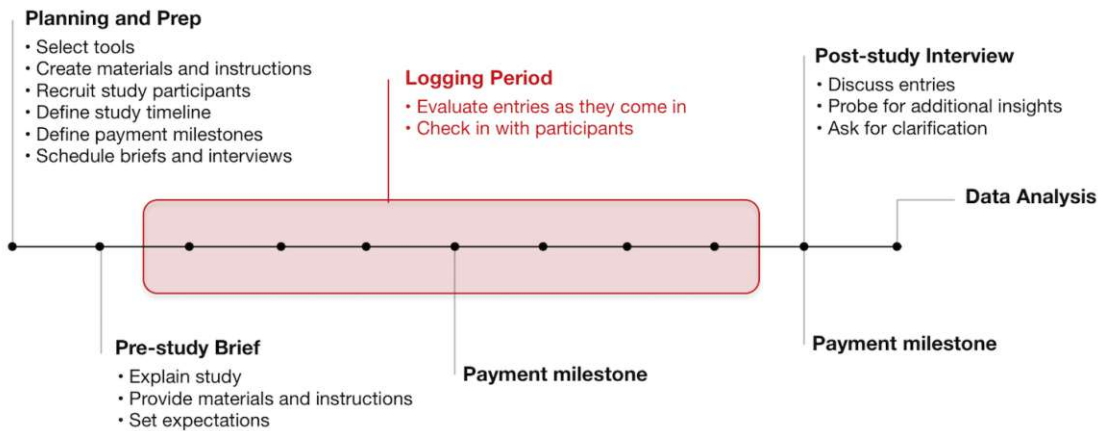


Figure 5.1: The timeline of a typical diary study, as described by NNG [41].

scores are then added together to calculate a score from 0 to 100, with higher scores pointing to higher perceived levels of usability.

Its ease of use and widely known status has made it an obvious choice for my study. It should be added that the SUS is quite a subjective way of measurement and cannot point to an absolute, objective quantification of the usability of a system.

Since the coding and highlighting were one of the stand-out features and specifically aimed at digital research diaries, I wanted to gain more specific insights about the design of these tools and how "usable" they deemed them.

Thus, to gather more and quantify more of my participant's thoughts, I added another brief questionnaire about some of the app's key features with the same scale as the SUS, ranging from "Strongly Disagree" to "Strongly Agree". With these additional questions, I wanted to gauge these features to help better understand what worked well for users and what they deemed to be the most useful part of the application. Since I did not want to take too much of my participant's time, I limited the length of the additional questionnaire to seven questions.

### 5.3 Planning and Preparation

After selecting my choice of user research methods, I selected and recruited my participants based on the criteria discussed in the previous section, 5.1. Additionally, the required hardware was gathered in case the participants did not own an iPad plus Pencil. The planning and conduction of the diary study were based on the timeline outlined by the Nielsen Norman Group, depicted in figure 5.1.

The software was distributed using Apple's Testflight<sup>1</sup> platform, a tool ideally suited to distribute an app that is not yet released in the Apple App Store to invited beta testers.

The plan was that participants in the study should use the app as a "digital research diary" for a current research project spanning over a timeframe of around two weeks.

While I offered my contact information to each participant and offered to help with any technical problems should they arise, I decided against checking in on them during the logging period, to get the most realistic and unbiased results.

Participants were asked to conduct in-Situ logging preferably, but only snippets of thoughts and recollections were also fine. It should be as natural as possible and resemble their normal note-taking habits. Since this is one of the main advantages of diary studies, allowing participants to log their thoughts and observations as they take place in a natural environment unlike a controlled lab setting [24].

The goal and key questions plus potential takeaways were defined as follows:

- Do they prefer a digital method or a more traditional paper-based one?
- Do they think additional meta-info like location or people can be useful?
- Is the mixture of input media (markdown-text, apple pen, optional images) useful and something they would use?
- Is the highlighting/coding feature something they would like?
- Do they think it helps with writing down the research/paper at the end?
- Generally, what can be improved/what do they like and dislike?

### 5.3.1 Tutorial/Introduction

Since the application has a set of very different features that partially use native iPadOS interactions which might not be intuitive for participants that are not well experienced with mobile Apple devices, it was apparent that I needed to include a more thorough explanation/tutorial that users could access while using the app.

Another approach could have been a more interactive approach with tooltips or other pop-up prompts, however, I decided against that, as discussed in section 3.2.8. It was decided that users should have an additional onboarding tool and reference they could come back to at any point to hopefully clear up any confusion they might have about features in the app. Furthermore, unique tools like the highlighter were programmed to show an informational dialog on first-time usage. The end result can be seen in figure 5.2.

---

<sup>1</sup><https://developer.apple.com/testflight/>, last accessed on 9<sup>th</sup> March, 2023



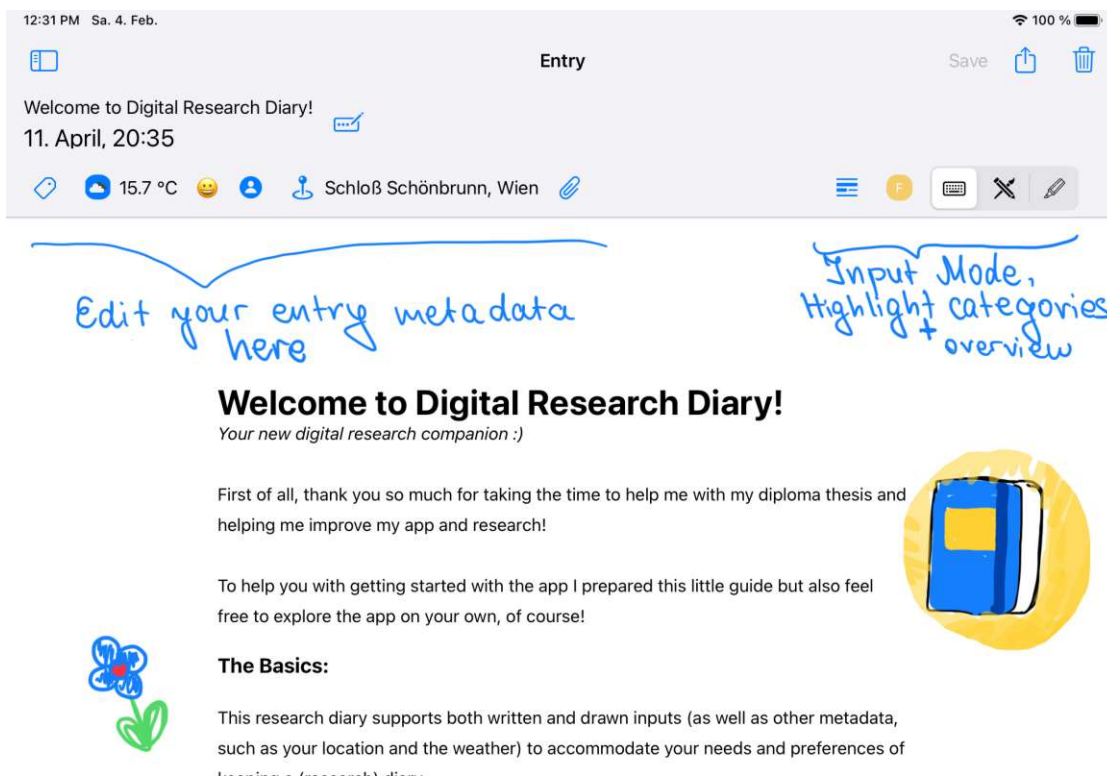


Figure 5.2: An excerpt from the quickstart entry, explaining the app's core functionalities in an example entry.

### 5.3.2 Participant briefing

Users were invited to a setup and explanation briefing session before the logging period to clear up any possible misunderstandings and help with installing the app on their iPad or the iPad they were provided with.

The aim and goals of the study were explained to them, as well as the expectations for their usage, i.e. that they should use the research diary app as a companion for a current research project for around two weeks with daily notes or at least during the week.

If they were not conducting a project at the moment of their participation, they should use it for meetings/diaries/current research topics as an alternative.

During this briefing, the software was briefly explained to them but not in complete detail since I wanted them to use the app without knowing too much beforehand in order to get more authentic user feedback.

I further explained that "In-Situ" logging, i.e. actually writing down the research notes as they occur to them "in the situation", would be preferred but only snippets of thoughts and recollections are also acceptable. It should be as natural as possible and resemble their normal note-taking habits - whatever they might be.

### 5.3.3 Post-Study Interview

After a period of the agreed two weeks, participants were invited to a post-study meeting that consisted of several steps.

Firstly, a consent form was handed out to the participants, asking them for informed consent to record their voices for further analysis later. After receiving the consent, the questionnaire containing the SUS and the additional questions about the coding and highlighting tool was handed out to the participant and they were asked to fill it out.

Next, a semi-structured interview was conducted with the participant, asking them about their honest experience, feedback, bugs they encountered, and other insights and thoughts they had during their time with the application.

The aforementioned key questions were used as a basic structure for the interview. I explicitly decided against a rigid interview structure since I wanted the users' unfiltered feedback and thoughts.

Finally, some sweets were handed out to the participants as a little thank you for their valuable help.

## 5.4 Analysis and Results

The analysis was done using a thematic analysis based on the concepts and approaches described in section 2.5.1. I decided to use an inductive approach as mentioned in section 2.5.3, since I wanted to start with a "blank slate", in hopes that my analysis and codes were more informed by the topics and thoughts raised by my participants and less informed by my own biases [21].

All audio material was transcribed and then analyzed thematically via multiple re-reads until I arrived at emerging categories, moving to themes/concepts on which I could base my theory on.

The topics were loosely separated into "what worked well" and "what did not work well / requires further work and research" and will be discussed in more detail in the following sections.

### 5.4.1 SUS Results

The results of the **SUS** with the additional questionnaire were analyzed, and an average for each question was calculated. While these statistics might not prove to be entirely statistically significant, they still paint an interesting quantification of the *usability* of the application and show what worked better than other things in addition to the calculated **SUS score** [31].

The resulting calculated SUS score amounted to approximately **82** across participants. According to Bangor et. al. this rating is quite respectable and would suggest an *adjective*

| Question  | Answer |   |   |   |
|---|--------|---|---|---|
| 1. I think I would like to use this tool frequently.  | 4      | 4 | 5 | 5 |
| 2. I found the tool unnecessarily complex.  | 2      | 1 | 2 | 2 |
| 3. I thought the tool was easy to use.  | 4      | 5 | 4 | 4 |
| 4. I think that I would need the support of a technical person to be able to use this system. | 2      | 1 | 1 | 1 |
| 5. I found the various functions in this tool were well integrated.                           | 3      | 5 | 4 | 4 |
| 6. I thought there was too much inconsistency in this tool.                                   | 1      | 1 | 1 | 1 |
| 7. I would imagine that most people would learn to use this tool very quickly.                | 3      | 3 | 5 | 4 |
| 8. I found the tool very cumbersome to use.   | 2      | 2 | 2 | 2 |
| 9. I felt very confident using the tool.  | 4      | 5 | 3 | 4 |
| 10. I needed to learn a lot of things before I could get going with this tool                 | 1      | 1 | 3 | 2 |

Table 5.1: The answers and results of the **SUS** (with a score of 1 being "strongly disagree" and 5 "strongly agree").

rating of "very good", very close to "excellent" [9]. This would suggest that users value the usability of the resulting app generally relatively high and enjoy using the app. The design was deemed as very consistent, not too complex and the participants indicated that they would use such an app frequently.

There were some outliers that would require more work and research. It seems that the ease of use for users unfamiliar with the iPad and other note-taking apps would still need more work and refinement to help inexperienced users to get started with creating their digital research diary. This is reflected in the feedback from one participant who was a first-time user of the Apple iPad and her low confidence in using the app and the iPad in general.

### 5.4.2 Questionnaire Results

| Question  | Answer |   |   |   |
|---|--------|---|---|---|
| 1. I prefer a digital method over a more traditional paper-based one.                       | 1      | 4 | 4 | 2 |
| 2. I think that additional meta-info, e.g. entry location or weather, are useful additions. | 3      | 5 | 5 | 5 |
| 3. The mixture of input media (text, pen, images) are useful and something I would use.     | 4      | 5 | 5 | 5 |
| 4. I would use the highlighting/ coding feature frequently.                                 | 4      | 4 | 5 | 4 |
| 5. I found the various functions in this tool were well integrated.                         | 4      | 4 | 4 | 4 |
| 6. I think the highlighting/coding feature is easy to understand/use.                       | 4      | 5 | 5 | 4 |
| 7. I think this tool would help with writing up a paper/thesis at the end of my research.   | 5      | 5 | 4 | 5 |

Table 5.2: The results of the additional questionnaire (with a score of 1 being "strongly disagree" and 5 "strongly agree").

The (average) results of the questionnaire would indicate that the participants did like and appreciate the additional developed features to support a digital research diary. They all would use the highlighting feature and different input modes and would appreciate the app for the write-up of their research. This could probably be even higher by extending the highlight feature and making it easier to use.

However, while they said they would use different input modes, the result of the diary

study and the interviews would indicate that users do not like to swap between different input modes as often as I first assumed. Most people generally prefer using one input method over another.

One noteworthy outlier was the first question, where two participants answered that they do not generally prefer digital methods over traditional ones. This is a perfectly valid opinion as there are many use cases, scenarios, and usage types where an analog diary or journal is a better or preferred option. As we will discuss later, there are also many cases, such as quick notes, e.g., a sticky note attached to a monitor with an important reminder, where the *permanent presence* of a sheet of paper is preferable.

### 5.4.3 General Feedback

In general, the app's design, functionality, and approach were very well received by all participants. They highlighted the easy-to-grasp and straightforward design and mentioned that they prefer this simple layout and navigation over other more complex apps for digital diaries.

Participants also highlighted that they enjoyed using the app not only for specific research but also as an auto-ethnographic tool for their academic life as a researcher, e.g., to note down what they did today during their work.

Users also appreciated the methods of organization available to them thanks to the tagging and filtering feature. Most users easily understood and used this functionality. Users familiar with more expansive note-taking apps like Bear said they would also need folders to boost their diary organization further.

Common positive feedback was the "endless page" design of the app, and most said that they preferred this approach to traditional page-based layouts. This also meant that most participants did not require or want a template feature, grid-based layout, or similar. This could indicate that users want a straightforward, primarily no-frills design and layout for their digital research diary. They want to get started with writing down their thoughts and experiences as quickly and efficiently as possible and might indicate that they don't want to be constrained and limited by predefined questions. Especially since they have to worry less about their entries being "pretty"/well-structured or making their handwriting legible when using the pencil. This would also align with the way the participants have reportedly used the app and the approaches described in the literature. At least for the first rough draft, they would use this approach. Two participants mentioned that they like to iterate, re-structure and format their entries multiple times during their research - as described by Engin and Browne [22, 14]. Here, better rich text editing, the highlighting feature, and tagging would help with that and was be appreciated by the users.

Participants also mentioned this as one of the most significant advantages of digital research diaries - being able to iterate, change and re-structure existing entries in a way that is impossible with traditional paper.

Interestingly, participants were not interested in drawing, e.g., sketches directly in the application. This might be due to them not being the type to create sketches in general, but two participants mentioned that they prefer to create drawings and sketches in other apps and then export them. This led me to a compelling observation: users noted that they prefer to have dedicated apps for specific workflows, e.g., an app for a (research) diary, an app for quick notes and reminders, an app for drawing, and so forth. This would indicate that there is no need to strive for a "jack of all trades" app, including all possible features from other (note-taking and diary) apps. Instead, it is better to have a smaller, more manageable, well-thought-out, and functional feature set.

#### 5.4.4 (Cloud) Backup and Content Syncing

One of the most discussed and mentioned topic by all participants was the option to backup the content of the diary on a cloud service, such as iCloud, and by extension the ability to sync their content between devices.

As one participant put it, a digital research diary without the option to backup all the sensitive content on it would again resemble a traditional, analog diary in the sense that users would have the same fear of losing their data. When something happens to their iPad or the software, all their data is lost and cannot be easily transferred and reproduced.

Furthermore, especially avid users of Apple devices are usually very much bought into the Apple ecosystem. They would expect to be able to use their data and apps across multiple devices. Apple's own apps, such as Notes, exemplify this by being available on the iPad, iPhone, and Mac with real-time syncing between all devices on the same account. This was also noted by participants experienced with Apple products and they would expect a digital research diary app to work the same way.

A compelling point brought up by one participant was possible concerns about privacy and processing of private data, especially considering the EU's General Data Protection Regulation (GDPR)<sup>2</sup>. To alleviate these concerns, it would be good to include more backup options than just iCloud or Google Cloud, as well as allow manual, *traditional* backup on file storage.

#### 5.4.5 Attachments

All participants mentioned the option to add other media and a more fleshed-out attachment feature. At the time of testing, only images could be added, and the visual feedback of added attachments in an entry was still lacking.

Additionally, it is still often the case that one might have to note something down on a sheet of paper quickly. Interestingly, this kind of analog notetaking is still the preferred way of jotting down critical reminders and information despite the prevalence

<sup>2</sup><https://gdpr.eu/>, last accessed on 9<sup>th</sup> March, 2023

of smartphones in most peoples' pockets. The participants wanted the option to *digitize* these sheets of paper by taking a picture or scanning them and then having the option to add them to an existing entry or create a new one. This approach also has the benefit that they could then easily discard the sheet of paper, and they wouldn't have to keep a stack of loose papers - a situation described by one participant. A participant also suggested quick voice memos in the app for this purpose - similar to what Notability is doing.

One participant also mentioned that this would be a required feature for them to consider switching to a digital diary since they often work with small sticky notes and other small sheets of paper.

### 5.4.6 Metadata

The support of multiple kinds of metadata for entries was generally seen as an exciting and fun addition to a digital research diary. The type of metadata used depended on the person. E.g., some said that they would not use the emotion metadata at all, and others said it was a valuable addition and would be interested in the conclusion that could be drawn from them when analyzed over a more extended period.

Users agreed that it could be a good "anchoring point" for their memory when seeing the location and weather as it might help them remember the day better - especially when they are often changing locations or writing while on the move.

The automatic addition of the weather and location was pleasing for users, although two suggested better distinguishing between automatic and manually added metadata.

The real value for users would require further investigation as it would require more data over more extended usage periods and the addition of a statistical overview for this data. Three participants reported that they would be interested in seeing this data to understand better how they are researching and feeling during certain weather conditions and how their mood affects this.

### 5.4.7 Export

An often-discussed emerging topic in addition to cloud backups was the option to support the export of entries or the whole diary as a document file, most often as a PDF or a ZIP file, containing all attachments. They all wanted the option of not being constrained by a proprietary format in the app and felt safer when all their created data could also be saved in a standard file format.

It is compelling that despite the prevalence of so many digital solutions and the private nature of a diary, users still highly value having the option to export their data in a standard file format such as PDF. This might indicate that they are not sure of the longtime support and longevity of whatever app they are using and understandably need to have the extra feeling of security that they can use their data elsewhere when exported as a PDF.

### 5.4.8 Highlight Feature

Generally speaking, the highlight feature was well-received, albeit being mostly underused and somewhat difficult to understand for more inexperienced users. The results from the additional questionnaire also support this.

Since most users did not read or fully understand the introduction, they did not use the feature much - especially participants that mostly used typing since they were unaware of the highlighting through the submenu and the rich editor toolbar.

Users also reported that they tried to highlight typed text which is sadly not possible in the prototype but would make sense for a fully-fledged application. This was also one of the few use cases where users that preferred to type would switch to the pencil to highlight since that was seen as much more convenient than the somewhat cumbersome native text selection on the iPad. One participant suggested letting the highlight tool "snap" and wrap around the text instead of having unsteady, hand-drawn strokes.

The participants liked the option of having differently colored highlighting categories. Still, of course, for a final product, these categories would need to be customizable and not pre-defined as in the prototype.

It should also be mentioned that the highlight feature would be more useful in later stages of a research diary where coding/analysis would play a more significant role and for a limited usage time of two weeks like in this study it is hard to gauge its usefulness fully.

### 5.4.9 Common Issues

While participants reported that they experienced relatively few bugs and no crashes, some common issues were mentioned. The biggest issue was the input switching and the sometimes faulty size calculation. This meant that users sometimes had to switch between modes multiple modes for their input to be registered and detected. Additionally, some users noticed the displacement of their drawn content when rotating the iPad but most said it was not a deal-breaker for them.

Other problems included some participants' general inexperience with the iPad which led to them being unsure if the problem was that they were not using the app correctly or had problems with iPadOS.

Another common problem for inexperienced users was the rich editing toolbar. Since they were provided with a hardware keyboard, they mostly used the app this way without opening the software keyboard. This led to them not noticing the toolbar on the bottom of the screen - an intended behavior by the operating system. Since this toolbar is not too big, it can be easily missed by users who have never used a custom toolbar in other apps. This can be seen as a general problem with the iPad and not strictly applicable to only this application.

This might have been somewhat avoided by more extensive context-sensitive tooltips hinting at the feature.

Some users also tried to draw with their fingers but were consequently confused why their input was not registering. This is because finger drawing has to be explicitly activated in the iPad settings and is not activated by default because users can then no longer scroll since their finger input is now used for drawing.

While users generally liked the idea of setting a custom title for entries, they had some suggestions/problems with the current design. First, for pencil users, the context switch to a traditional input view was somewhat annoying, especially because Apple's handwriting detection with "Scribble" is relatively slow and error-prone. Second, two users reported that they generally write the title of an entry in its first line and would like it if the app uses this title if there is no custom one set - similar to Apple Notes' approach.

In general, some design elements and features had a steeper learning curve for newer users than expected and would require more work and better context-sensitive explanations.

### 5.4.10 Discussion and Summary

In summary, the results suggest that users value the ability to backup and sync their notes across multiple devices, as well as the option to export their notes in PDF format. The ability to switch between inputs, such as a keyboard and a pencil, can be frustrating for some users, and the learning curve for some of the app's features can be steep.

Users appreciate the simplicity of the app's design and tagging features, but they would like to see more user-defined highlight categories. Context switching between inputs and drawing with a finger can also be confusing for some users.

Users also appreciate the ability to revise their notes, especially for research purposes, but they would like to see more discoverable features and better handwriting detection. The need for a secure backup solution and privacy concerns are also important considerations for users. Overall, the results suggest that the iPad and the research diary app can be a useful tool for a digital replacement of a research diary. However, there is still room for improvement in terms of user experience and feature discovery.



## Conclusion and Discussion

In this thesis, we looked at the state-of-the-art of research diaries and existing solutions that can be used for (digital) research diaries. We strove to answer the question of what is the best way to conceptualize and implement a digital research diary on the Apple iPad.

Furthermore, this thesis should also help readers better understand what a research diary is and, more importantly, help them get started with their own research diary, whether digital or analog. The provided practical examples and tips for the possible content of a research diary should help with this process.

The developed application was designed from the ground up based on expert feedback, a literature review, and the iPad's existing note-taking and diary apps. This design was then implemented using modern standards of native mobile app development using an experimental approach of layering the drawing and typing layers directly over each other. This was done in hopes of getting the most insightful results and feedback regarding users' preferences and habits of making use of the multimodal input that the iPad supports. As discussed in previous chapters, this approach has led to many development challenges.

Next, this app was evaluated in user tests which consisted of diary studies plus accompanying interviews and questionnaires, using the SUS, among other things.

The achieved SUS score was around 82 indicating excellent usability which was further reinforced and supported by the follow-up interviews of the conducted diary study.

The findings from the user tests indicate that users highly value the backup and sync capability for their notes across multiple devices, as well as the option to export their notes in PDF format. On the other hand, some users may find switching between input methods often cumbersome, and some of the app's features can be challenging to learn for some users.

Users favor the simplicity of the app design and tagging features, but they would like to see more customization, e.g., for highlight categories and more extensive digital attachment support.

Users also appreciate the ability to revise their notes, particularly for research purposes, but they would like to see more discoverable features and improved handwriting recognition. Security and privacy are also crucial factors for users regarding backing up their notes.

In conclusion, the iPad and the research diary app have the potential to be a useful tool for digital research diaries. Still, there is room for improvement in terms of user experience and feature accessibility both regarding the application and iPadOS.

### 6.1 Answers to the Research Questions

Several research questions were formulated at the beginning of my research that I strove to answer during my thesis. The following should not be seen as a definitive answer to each of these questions, as different users have different needs and more and less experience with devices like the iPad. Some would require more research or better support from the iPad itself before they can be fully answered.

#### 6.1.1 How can research diaries be technically supplemented and augmented by an iPadOS application?

As discussed in the previous chapter, there are many aspects of a digital research diary app on the iPad that users expect to be present.

In general, users liked having an app dedicated to their research diary instead of having one app that contains all of their notes, memos, and to-do lists from their private and academic life.

Especially for those transitioning from a traditional paper-based diary, the app should have strong support for the Apple Pencil to accommodate the needs of these users best. For more experienced "power users" features such as rich editing to better format potentially very long entries are of more importance.

One of the important features discussed by all participants was the option to backup the diary's content on a cloud service. Without the option to backup all their sensitive data it would lose all the advantages of being easily reproducible instead of being committed to one piece of paper. Especially for more avid Apple users, the option to access, sync, and change their diary across multiple devices was also an essential aspect of a digital research diary. Ideally, such an application should also offer meaningful privacy options and, for example, adhere to the EU's General Data Protection Regulation (GDPR)<sup>1</sup>.

---

<sup>1</sup><https://gdpr.eu/>, last accessed on 9<sup>th</sup> March, 2023

A digital research diary should also feature the option to export all or specific entries in a traditional file format, such as PDF. All user study participants wanted the option of not being constrained by a proprietary format in the app and felt safer when all their created data could also be saved in a standard file format.

Another important aspect is the option to add other media. E.g., one user reported that they often work with small sticky notes and other small sheets of paper to note something down quickly. An option to digitize these sheets of paper by taking a picture or scanning them and adding them to an entry was quite crucial for users. Voice memos for such short notes would also be a helpful option.

A digital research diary can especially shine during the later stages of research, where the created content is analyzed, re-read, highlighted, and coded. The option to create highlights with custom-defined categories grouped in an overview can further elevate a digital research diary.

Finally, to support users in their journey using a digital diary, such an application should feature context-appropriate tooltips for unique interactions and features and slowly introduce them over time.

### **6.1.2 In what ways can the multimodal interaction possibilities of the iPad, especially voice input and the Apple Pencil, support the creation and management of research diaries?**

It appears that despite the iPad offering multiple ways of interaction and input methods, users expectedly prefer to use the input method they are most accustomed to - whether that is a hardware keyboard, the onscreen keyboard, or a pencil. However, users are also less inclined to switch these methods as initially assumed. This seems to be further hindered by somewhat lackluster support for developers and the current capabilities of the iPadOS when it comes to switching between input methods. However, there are some notable exceptions. Such as creating highlights of typed text where text selection with the pencil would be easier than with the native text selection.

There might also be a meaningful difference between users more accustomed to paper-based diaries and those already used to the iPad and writing down all their notes and reminders in a digital application.

But this is also where a multimodal design and the iPad with its pencil might shine as it allows researchers used to traditional paper-based diaries to easily switch to a digital one since their main writing mode would not have to meaningfully change. This is further supported by remarks from users that they can feel comfortable using the pencil in a digital diary since they have to worry less about their handwriting being legible.

On the other hand, handwriting detection features, such as Apple's Scribble feature, are often seen as more of an annoyance and hindrance in their workflow instead of a benefit. This is mainly because it slows users down and they have to conform their handwriting to the algorithm and have to work around this technology instead of it helping them.

In general, few apps have been able to combine handwriting and typed text really well, indicating that more research and support are needed. Users also prefer to switch to an app dedicated to one input method instead of creating everything in one app, e.g., when drawing sketches for a note, they would prefer to do it in a dedicated app.

Adding voice memos and audio recordings is a fascinating concept that some users were generally keen on using but would require more research and user studies. One example where voice might be especially useful is for quick audio memos and reminders that a user wants to note down when they are short on time and don't have time for a fully written reminder or how Notability 2.9.1 implemented audio recordings that can be synced with written notes.

### **6.1.3 How do experienced and inexperienced users of research diaries evaluate current applications that can be used for creating diaries and what features would they want to include in such applications?**

The results of the SUS survey showed that users rated the app's usability as high (82), found the design consistent and non-complex, and would use the app frequently. Participants also appreciated the features added to support a digital research diary and would use the highlighting feature and various input modes. The score could potentially be improved by making the highlighting feature more reliable and extending it.

Some design elements and features had a steeper learning curve for inexperienced users. They would require more work and better context-sensitive explanations. They also had trouble noticing and using the rich editing toolbar when using a hardware keyboard.

The results of the additional questionnaire suggest that the participants liked and appreciated the new features added to support a digital research diary. They would use the highlighting feature and different input modes, and the app would help with writing up their research. However, the diary study and interviews showed that users don't often switch between input modes and prefer using one method over another. One participant answered that they do not prefer digital methods over traditional ones, which is a noteworthy exception.

Adding multiple types of metadata to a digital research diary was received positively by users, with different types of metadata being used differently by individuals. The automatic addition of weather and location information was appreciated, but a clearer visual distinction between automatic and manually added metadata should be added. The full value of the metadata requires more data and a statistical overview for extended usage periods. Some users expressed interest in seeing this data to understand the potential relationship between weather conditions, time of the month, mood, and their researching behavior.

A consensus among users was their preference for an "endless page" design over a page-based design with different backgrounds as done in other apps, such as GoodNotes 2.9.2.

Additionally, they were not as interested in custom templates or other writing prompts as initially assumed. Users of a digital research diary seem to prefer a simple and intuitive way of writing down their thoughts and experiences - as one user study participant described it as a *"stream of consciousness"*.

## 6.2 Discussion

### 6.2.1 Chosen Design and Implementation

The designed and implemented application can already serve as a comprehensive digital research diary, as demonstrated in the user tests. It boasts a wide range of features, including multiple input methods, highlighting and markup capabilities, and more.

As such, it remains a prototype in some aspects, mainly due to the implementation of the two input methods. This approach is unique compared to other diary applications, which may be because it's a challenging aspect. Improvements in areas such as document export, more reliable input switching, and better synchronization of text and drawings are needed to bring the application to its full potential. Additionally, there are design considerations and other improvements identified through the user testing sessions that need to be addressed.

As discussed in previous chapters, some of these input method problems could potentially be solved by a larger, more experienced development team or a different design approach that could better accommodate different screen sizes, aspect ratios, and input methods.

Despite these challenges, I am proud of the outcome and see the implementation as an exciting opportunity for growth. It allowed me to expand my knowledge of iOS and Swift APIs, including those from somewhat poorly documented libraries such as PencilKit and PDF rendering, which are not frequently used or required in most iPhone and iPad apps.

### 6.2.2 Chosen Methodology

Overall, the chosen methodology was in my opinion the right choice for the research topics at hand and has produced valuable results. All the interviews were very insightful and over half an hour long, leading to productive discussions about the tested application and the participants' approaches with research diaries, their gripes and problems with multimodal interactions on the iPad, and much more.

There were some problems with getting timely answers from participants and one participant also got sick during the testing period which resulted in them spending less time with the application. However, all of their input was still quite extensive and insightful and has helped me better understand how people would and want to interact with a digital research diary.

Admittedly, a more extended period of testing the app could have yielded more results. Still, firstly, I did not want to ask too much of my participants, and a timeframe of

two weeks was in my opinion and according to other projects a good middle ground to let them use the app and form their opinions on it. Secondly, since their entries were intended as a realistic use-case scenario and the private nature of a research diary, I did not want to collect their usage data or created entries. I was rather interested in their experience with the app and digital journaling in general.

Additionally, it would have been interesting to have more participants testing different note-taking and journaling apps and directly comparing them with the developed application instead of just me testing them. This could have given a better feel of where the app excels and where more inspiration could have been drawn from. However, this would have required even more time and effort for the recruited participants, and I did not want to ask too much of my voluntary participants. Or different versions of the app could have been distributed and tested with different users. But it would have been hard to meaningfully compare these results without each of them comparing all the versions and since I strove to create a prototype that could come close to a more fully-fledged replacement of a research diary, designing and implementing multiple, feature-rich prototypes would have simply been out of scope for this thesis.

In order to produce and design a commercial app for a digital research diary, further user research would be required. This would entail more design and implementation iterations, A/B testing, and more prototyping of different approaches and design philosophies.

The discussed usage behaviors and insights by the user evaluation are gathered based on a limited first-time usage scenario where users are still adjusting to the app and the iPad in the case of inexperienced users. How these behaviors would change over time remains to be seen and could prove to be a compelling future user evaluation topic. Examples could include how users would adjust their sketching behavior, e.g. would they start creating more sketches in the app or create placeholders for them? Or would they start moving away from analog (quick) notes over time?

# Outlook and Future Work

The results of this thesis and the accompanying application are intriguing and enlightening in regards to how individuals engage with the iPad and its stylus, as well as digital journal and note-taking applications in general. It was fascinating to observe the feasibility of a digital research journal and the effort necessary to produce a cutting-edge note-taking app with a comprehensive feature set.

However, more design, development, and user research work would be required to elevate a digital research diary to its fullest potential. Many features would need to be added, revamped, or extended.

## 7.1 Additional Features

During the user tests and also earlier in the design/implementation phase there were a lot of additional features that came up. All these features would help transform this application into an even more reliable and valuable digital research companion.

### 7.1.1 More Extensive Multimodal Support

One area that would need further work is extending and improving the support of multimodal input. Currently, the input switching can be slightly cumbersome at times, and the size calculation of both input views is not always perfect. Additionally, it would be interesting to further research the best ways of allowing multimodal input support on the iPad as it seems that there is still no "ideal" solution for solving this problem and users usually seem to be mainly gravitating towards one input method and see switching user input methods as mostly cumbersome. Some apps - and Apple's Scribble feature - have found clever solutions for this problem via creative pencil gesture recognition, e.g. highlighting text or selecting text by drawing a circle around it.

However, general support from the operating system still seems to be lacking, as participants of the user study reported as well. It would require better APIs for developers, more customized settings and options for users, plus a better tutorial and explanation from Apple, especially for less experienced users.

### 7.1.2 Attachments/Other Digital Media

Adding more extensive support for digital media as attachments was a popular request from the user study participants. This would include better visual feedback of added attachments, support of videos and images, and the option to add voice memos.

Many users still rely on small paper sheets for quick notes and reminders for both their private and academic life. These notes could be easily digitized via photo import or a scan feature like in Apple Notes. Or if Apple would allow third-party apps to use this feature, the native **Quick Notes** action could be extended or customized<sup>1</sup>.

Another idea would be to let users quickly and easily record short voice memos via a shortcut or even a **Siri Extension**<sup>2</sup> for this kind of quick notes and thoughts.

### 7.1.3 Statistics

There are a lot of insightful statistical insights about one's researching behavior and general mood/behavior that could be extracted from how a user interacts with the app. Adding the various metadata options makes this data even more valuable.

To make this data visible and engaging for users, a statistical overview should be added, presenting the various data points in different ways - as other diary apps like Grid Diary 2.8.1 or Journey 2.8.2 do. This might include details about their writing habits, moods at certain times of the year or under certain weather locations, and so forth. It might be valuable to see how these data points influence each other and how it affects different personalities and types of researchers and could lead to interesting future work.

Users could also be further encouraged to add this data if they feel that they might gain valuable information about themselves through it and could be encouraged to write more often and regularly, especially when coupled with reminders and "*streaks*".

### 7.1.4 Better Tutorials/Feature Introductions

While the app's design and structure were well received and deemed as pleasing and straightforward, there were some unique features that had a steeper learning curve. For this reason, an introduction entry that served as a tutorial was devised but users mostly only skimmed over it and instead wanted to try out the app on their own. This

---

<sup>1</sup><https://www.guidingtech.com/how-to-use-quick-notes-on-iphone-ipad-and-mac/>, last accessed on 9<sup>th</sup> March, 2023

<sup>2</sup><https://developer.apple.com/documentation/sirikit>, last accessed on 9<sup>th</sup> March, 2023



would suggest that some form of context-sensitive *tooltips* would be better suited for this application rather than frontloading all the information in one example entry.

For example, when a user first creates a new entry an (easily dismissable) tooltip could be shown to them explaining the meta-data and letting them add their first mood information, for example. Other features could be slowly shown and explained to the user, e.g., after typing a certain number of words, the highlight option for text could be explained to them.

### 7.1.5 User Tests

All of these proposed features and extensions would require even more extensive user tests with a larger participant group to get a better feel for what the users actually want and need and which features would require more explanation or refinement.

## 7.2 Future Work

Several issues came up during development and user tests that would be fascinating to research in further detail.

For example, one such topic that arose during user tests is the question of why so many people often still resort to quick notes on sheets of paper to write down important reminders and information quickly instead of e.g., using their smartphones, which we all usually carry around nowadays in this digital world.

Another topic would be the support and interaction of the Apple Pencil. Lots of apps are either focused on the pencil or typing and it would be exciting to further research how these two things (plus other inputs like audio) can be better combined in a way that makes this mode switching easier and more delightful.

Additionally, the various meta-data added for the developed digital research diary and their potential statistical usefulness for studies that would want to look into how our surroundings and mood influence our researching and journaling behavior could warrant an exciting topic for future work.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | A simplified illustration of the codes-to-theory approach described by Saldaña [42]. . . . .   | 14 |
| 2.2  | A deductive approach would start out with certain codes and try to find parts of the data that fit them. . . . .   | 15 |
| 2.3  | An inductive approach derives codes and themes directly from the data at hand. . . . .   | 16 |
| 2.4  | Grid Diary offers different overviews via the tab bar on the left, such as the calendar view, as well as a unique grid-based layout with custom prompts for each new entry to help user guide their thoughts. . . . .  | 23 |
| 2.5  | Users can switch between different ways to view their entries, one of them being a calendar overview that shows users on which day they created an entry and, if applicable, the attachments they added to an entry. . . . .   | 24 |
| 2.6  | Journey's editor is relatively straightforward with no options for custom templates. Additional meta-data and markdown editor options can be added via buttons on the bottom of the screen, always resting above the onscreen keyboard. . . . .                              | 25 |
| 2.7  | DayOne's overview page offers quick access to creating a new entry, adding other media, as well as a well-structured timeline with additional stats of the current journal. . . . .  | 26 |
| 2.8  | The layout of Notability is as simple and basic as an app like this can be. It offers a sidebar with different folders/"subjects" on the side, the option to edit a note by tapping on it or creating a new one by tapping on the "plus" button on the bottom right. . . . . | 28 |
| 2.9  | A standout feature of Notability is the way it can synchronize audio with other content in a note, allowing users to jump to the exact point a stroke was created in the audio, and vice-versa. . . . .  | 29 |
| 2.10 | GoodNotes structures content into folders, subfolders, and Notebooks. Notebooks can contain as many pages as desired with many layout templates to choose from for each page. . . . .  | 30 |
| 2.11 | Since GoodNotes is quite focused on pen input, it offers very extensive ways to customize the behavior, colors, and look and feel of different drawing tools, such as the pen, usually only seen in drawing apps. . . . .  | 31 |
|      |  | 99 |

|      |  |    |
|------|--|----|
| 2.12 | Apple Notes has a general structure of three collapsable hierarchical views, from left to right: the folders (both local and in the iCloud), the current list of notes, and the currently opened note in the editor view. . . . .      | 32 |
| 2.13 | Apple notes does offer both handwritten pencil input, as well as typed text. However, it makes a point to separate the two into different sections, further illustrated by the two yellow lines above and below the handwriting. . . . | 33 |
| 3.1  | LiquidText provides a unique approach and way to view and connect comments, highlights, and excerpts. This visual representation was a big inspiration for designing and implementing the coding views. . . . .                        | 37 |
| 3.2  | An early sketch featuring a sidebar containing a tab bar with potential overview tabs on the left. On the right, an entry screen with a grid-based layout and prompts for each section. . . . .  | 41 |
| 3.3  | A concept for a grid-based entry editor screen consisting of multiple editable views based on a changeable list of questions/prompts. . . . .  | 42 |
| 3.4  | The earliest entry toolbar wireframe, featuring a combined entry toolbar on top of the screen. . . . .   | 42 |
| 3.5  | A design sketch, inspired by <i>Apple's Notes</i> app featuring a streamlined, straightforward design with different sub-menus. . . . .  | 43 |
| 3.6  | A wireframe of a concept for a step-based entry creation screen, based on a template/list of questions. . . . .  | 44 |
| 3.7  | The final entry editor sketch, dividing the editor screen with a toolbar for input mode switching, meta-data, and other edit options on top and a context-appropriate toolbar for pencil or text input. . . . .                        | 45 |
| 4.1  | Apple's version of MVC [2]. . . . .  | 53 |
| 4.2  | The MVVM-C( <b>M</b> odel <b>V</b> iew <b>V</b> iew <b>M</b> odel- <b>C</b> oordinator) pattern. . . . .   | 54 |
| 4.3  | The project and folder structure of the developed application, as displayed in the chosen IDE, Xcode. . . . .  | 57 |
| 4.4  | The custom toolbar " <i>RichEditorToolbar</i> ", which is displayed above the standard keyboard. This toolbar contains all the rich editing options that users can choose from to customize their diary entry's layout. . . . .        | 62 |
| 4.5  | The custom-designed and implemented toolbar, displays all relevant information of the currently selected diary entry as well as the options for input switching. . . . .   | 65 |
| 4.6  | The tags screen is displayed as a popover screen, making it easier for users to dismiss when no longer needed, as well as the source view being easily identified. . . . .   | 68 |
| 4.7  | The sidebar as it appears in the app. It consists of a list of entries that shows its title, creation date, and the diary's tags, and the "hide" and "add new entry" buttons on the top. . . . .                                       | 70 |
| 4.8  | The many possible display mode settings of a UISplitViewController, allowing full customization of when to display which sidebar. . . . .  | 71 |

|     |   |    |
|-----|---|----|
| 4.9 | The PDF preview that pops up after users request a screenshot of an entry. Users can edit, markup, share and save the PDF on their device or the cloud. | 72 |
| 5.1 | The timeline of a typical diary study, as described by NNG [41]. . . . .  | 79 |
| 5.2 | An excerpt from the quickstart entry, explaining the app's core functionalities in an example entry. . . . .  | 81 |



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# List of Tables

|     |   |    |
|-----|---|----|
| 5.1 | The answers and results of the <b>SUS</b> (with a score of 1 being "strongly disagree" and 5 "strongly agree"). . . . .   | 83 |
| 5.2 | The results of the additional questionnaire (with a score of 1 being "strongly disagree" and 5 "strongly agree"). . . . . | 83 |



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.



# Bibliography

- [1] Andrew Delano Abbott. *Methods of discovery: Heuristics for the social sciences (contemporary societies)*. WW Norton & Company, 2004.
- [2] Mariam Aljamea and Mohammad Alkandari. “MMVMi: A validation model for MVC and MVVM design patterns in iOS applications”. In: *IAENG Int. J. Comput. Sci* 45.3 (2018), pp. 377–389.
- [3] Herbert Altricher et al. *Teachers investigate their work: An introduction to action research across the professions*. Routledge, 2005.
- [4] Herbert Altrichter and Mary Louise Holly. “Research diaries”. In: *Research methods in the social sciences* (2005), pp. 24–32.
- [5] Inc. Apple. *Multitasking*. 2022. URL: <https://developer.apple.com/design/human-interface-guidelines/components/selection-and-input/segmented-controls> (visited on 01/05/2023).
- [6] Inc. Apple. *Multitasking*. 2022. URL: <https://developer.apple.com/design/human-interface-guidelines/patterns/multitasking/> (visited on 12/29/2022).
- [7] Inc. Apple. *Sidebars*. 2022. URL: <https://developer.apple.com/design/human-interface-guidelines/components/navigation-and-search/sidebars/> (visited on 01/02/2023).
- [8] Inc. Apple. *User Notifications*. URL: <https://developer.apple.com/documentation/usernotifications> (visited on 12/29/2022).
- [9] Aaron Bangor, Philip Kortum, and James Miller. “Determining what individual SUS scores mean: Adding an adjective rating scale”. In: *Journal of usability studies* 4.3 (2009), pp. 114–123.
- [10] Robert Bogdan and Sari Knopp Biklen. *Qualitative research for education*. Allyn & Bacon Boston, MA, USA: 1997.
- [11] Jessie Bontuyan. “Academic diary in the teaching of oral communication in context”. In: *International Journal of English Language Studies* 4.3 (2022), pp. 10–18. DOI: 10.32996/ijels.2022.4.3.2.
- [12] Simon Borg. “The research journal: A tool for promoting and understanding researcher development”. In: *Language teaching research* 5.2 (2001), pp. 156–177.

- [13] Stephen Brown and Andreas Holzinger. “Low cost prototyping: Part 1, or how to produce better ideas faster by getting user reactions early and often”. In: *People and Computers XXII Culture, Creativity, Interaction 22* (2008), pp. 213–214.
- [14] Brendan Ciaran Browne. “Recording the personal: the benefits in maintaining research diaries for documenting the emotional and practical challenges of fieldwork in unfamiliar settings”. In: *International Journal of Qualitative Methods* 12.1 (2013), pp. 420–435.
- [15] Robert G Burgess. “Keeping a research diary”. In: *Cambridge Journal of Education* 11.1 (1981), pp. 75–83.
- [16] Otilia Clipa, Aurora-Adina Ignat, and Mihai Stanciu. “Learning diary as a tool for metacognitive strategies development”. In: *Procedia-Social and Behavioral Sciences* 33 (2012), pp. 905–909.
- [17] Joanne E Cooper. “Telling our own stories: The reading and writing of journals or diaries”. In: *Stories lives tell: Narrative and dialogue in education* (1991), pp. 96–112.
- [18] Meghan Cope. “Coding transcripts and diaries”. In: *Key methods in geography* 440 (2010).
- [19] John W Creswell and Johanna Creswell Báez. *30 essential skills for the qualitative researcher*. Sage Publications, 2020.
- [20] John W Creswell and Cheryl N Poth. *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications, 2016.
- [21] Victoria Elliott. “Thinking about the coding process in qualitative data analysis”. In: *The Qualitative Report* 23.11 (2018), pp. 2850–2861.
- [22] Marion Engin. “Research diary: A tool for scaffolding”. In: *International Journal of Qualitative Methods* 10.3 (2011), pp. 296–306. DOI: 10.1177/160940691101000308.
- [23] Kim Flaherty. “Diary studies: Understanding long-term user behavior and experiences”. In: *Retrieved January 7* (2016), p. 2018.
- [24] Rúben Gouveia and Evangelos Karapanos. “Footprint tracker: supporting diary studies with lifelogging”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, pp. 2921–2930.
- [25] Amie S Green et al. “Paper or plastic? Data equivalence in paper and electronic diaries.” In: *Psychological methods* 11.1 (2006), p. 87.
- [26] Paul Hudson. *What is the first responder?* May 2019. URL: <https://www.hackingwithswift.com/example-code/system/what-is-the-first-responder> (visited on 01/02/2023).
- [27] Paul Hudson. *Writing data to the documents directory*. Dec. 2021. URL: <https://www.hackingwithswift.com/books/ios-swiftui/writing-data-to-the-documents-directory> (visited on 01/02/2023).

- [28] Apple Inc. *Human interface guidelines*. URL: <https://developer.apple.com/design/human-interface-guidelines/guidelines/overview/> (visited on 01/02/2023).
- [29] Apple Inc. *Property list programming guide*. Mar. 2010. URL: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/PropertyLists/Introduction/Introduction.html> (visited on 01/02/2023).
- [30] Valerie J Janesick. "A journal about journal writing as a qualitative research technique: History, issues, and reflections". In: *Qualitative inquiry* 5.4 (1999), pp. 505–524.
- [31] James R Lewis. "The system usability scale: past, present, and future". In: *International Journal of Human-Computer Interaction* 34.7 (2018), pp. 577–590.
- [32] John Lofland et al. *Analyzing social settings: A guide to qualitative observation and analysis*. Waveland Press, 2022.
- [33] Elisa D Mekler et al. "A diary study exploring game completion and player experience". In: *Proceedings of the first acm sigchi annual symposium on computer-human interaction in play*. 2014, pp. 433–434.
- [34] Neil Mercer. *The guided construction of knowledge: Talk amongst teachers and learners*. Multilingual matters, 1995.
- [35] Matthew B Miles and A Michael Huberman. *Qualitative data analysis: An expanded sourcebook*. sage, 1994.
- [36] Theresa Neil. *Mobile design pattern gallery: UI patterns for smartphone apps*. "O'Reilly Media, Inc.", 2014.
- [37] Darren Newbury. "Diaries and fieldnotes in the research process". In: *Research issues in art design and media* 1.1 (2001).
- [38] Jonas Nygren. *Print & Digital: A qualitative study about students perceived differences and how they move between the mediums*. 2020.
- [39] Jack C Richards. *Communicative language teaching today*. SEAMEO Regional Language Centre Singapore, 2005.
- [40] Christian Rohrer. "When to use which user-experience research methods". In: *Nielsen Norman Group* 12 (2014).
- [41] Kim Salazar. *Diary studies: Understanding long-term user behavior and experiences*. June 2016. URL: <https://www.nngroup.com/articles/diary-studies/> (visited on 01/05/2023).
- [42] Johnny Saldaña. *Coding manual for qualitative researchers*. SAGE, 2015.
- [43] Irving Seidman. *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. Teachers college press, 2006.
- [44] Mary Shaw and David Garlan. *Software architecture: perspectives on an emerging discipline*. Prentice-Hall, Inc., 1996.

- [45] Laura Sheble and B Wildemuth. “Research diaries”. In: *Applications of social research methods to questions in information and library science* (2009), pp. 211–221.
- [46] Jared M Spool et al. *Web site usability: A designer’s guide*. Morgan Kaufmann, 1999.
- [47] Mohan Vinjamuri, Bryan Warde, and Patricia Kolb. “The reflective diary: An experiential tool for enhancing social work students’ research learning”. In: *Social Work Education* 36.8 (2017), pp. 933–945.