Politecnico di Torino

Politecnico di Torino

Master's Degree in Computer Engineering
Graphics and Multimedia
A.y. 2023/2024
Graduation Session July 2024

Rethinking Apple Calendar with Student-Centric Features on iOS

Supervisors

Prof. Luigi DE RUSSIS

Prof. Zdeněk MÍKOVEC

Candidate

Angelica COSTANTINI

Summary

The thesis endeavors to address the needs of university students by enhancing the functionality of an iOS application. As modern education systems become increasingly complex, students face challenges in efficiently managing their academic schedules, which often involve many classes, assignments, exams, and extracurricular activities. Recognizing these challenges, the thesis aims to redesign the existing application to better cater to the unique requirements of university students.

By integrating new templates specifically tailored for university schedules, the redesigned application seeks to provide students with a more intuitive and streamlined way to organize their academic commitments. The addition of these templates not only offers practical benefits such as improved time management and organization but also contributes to enhancing the overall user experience of the application.

Through the adoption of User-Centered Design (UCD) principles, the redesign process prioritizes the needs and preferences of the end-users, ensuring that the new features align closely with their expectations and usage patterns. Additionally, by incorporating User Interface (UI) and User Experience (UX) design strategies, the thesis aims to create an interface that is visually appealing, easy to navigate, and conducive to efficient task completion.

The choice of Figma for prototyping and design, followed by implementation in Xcode for iOS app development, reflects a comprehensive approach to the redesign process. By leveraging these tools, the thesis endeavors to create a seamless transition from concept to execution, ensuring that the final product meets the highest standards of usability and functionality.

Overall, the thesis seeks to contribute to the improvement of student life by providing a digital solution that addresses the unique challenges of managing university schedules. Through thoughtful design and implementation, the redesigned iOS application aims to empower students with the tools they need to succeed academically while enhancing their user experience.

Acknowledgements

At the end of this journey of growth, I would like to express my gratitude to all those who have supported me throughout these months.

First and foremost, I would like to extend my sincere thanks to my Erasmus supervisor for accepting my thesis proposal and providing unwavering support throughout the entire process. Your guidance and encouragement have been invaluable, and I am deeply grateful for your commitment to my academic and personal development.

I would also like to thank my supervisor at my home university for constantly pushing me to improve and for instilling in me the importance of striving for excellence. Your belief in my abilities, your constructive feedback and your boundless patience have been instrumental in shaping this project and enhancing my skills.

Furthermore, I am immensely grateful to my family for their constant support and wise advice, which have always motivated me to persevere. Your encouragement has been a source of strength and inspiration during challenging times.

Finally, I wish to extend my heartfelt thanks to all my friends who have been with me through both the challenges and triumphs of these years. Your companionship and understanding have made this journey more enjoyable and rewarding.

Thank you to everyone who has contributed to this endeavor. Your support has been essential in helping me achieve my goals, and I am profoundly appreciative of all your efforts.

Thank you for everything.

"To my parents for believing in this path for me."
Angelica

Table of Contents

List of Figures IX

Acronyms XIII

1 Introduction 1
- 1.1 User Experience (UX) . . . 2
  - 1.1.1 Key Features of UX . . . 2
- 1.2 User Interface (UI) . . . 3
  - 1.2.1 Key Features of UI . . . 3
- 1.3 User Experience VS User Interface . . . 4
- 1.4 Aim of the project . . . 6

2 State of the Art 7
- 2.1 What does the market already offer? . . . 7
- 2.2 So why create something that already exists? . . . 10

3 Methods 11
- 3.1 User Centered Design (UCD) . . . 11
- 3.2 Methodologies . . . 13
  - 3.2.1 Persona Development . . . 14
  - 3.2.2 Scenarios . . . 16
  - 3.2.3 Use-Cases . . . 17
  - 3.2.4 Storyboard . . . 18
  - 3.2.5 Hierarchical Task Analysis (HTA) . . . 19
  - 3.2.6 Sketches . . . 22
  - 3.2.7 Mock-ups . . . 23
  - 3.2.8 Prototyping . . . 25
  - 3.2.9 Testing . . . 29

4 Design 32
- 4.1 User Research . . . 32

4.2 Persona Development . . . 35
4.3 Scenario and Storyboard . . . 39
4.3.1 First Scenario: How to consult the timetable with a click . . . 39
4.3.2 Second Scenario: Odd or even week? . . . 40
4.3.3 Third Scenario: Reminder via notification . . . 42
4.3.4 Hierarchy of scenarios . . . 43
4.4 Hierarchical Task Analysis . . . 43
4.4.1 First HTA . . . 44
4.4.2 Second HTA . . . 45
4.4.3 Third HTA . . . 46
4.5 Sketches . . . 47
4.5.1 How to insert the schedule . . . 48
4.5.2 How to set the schedule . . . 49
4.5.3 Different Views . . . 50
4.5.4 2-Weeks View . . . 51
4.5.5 Marks . . . 52
4.6 Mock-ups . . . 53
4.6.1 Add a new timetable . . . 53
4.6.2 Views . . . 55
4.6.3 Colors . . . 58

5 Implementation 60
5.1 Prototyping . . . 60
5.2 Figma . . . 61
5.2.1 Add a new timetable . . . 61
5.2.2 Views . . . 64
5.2.3 Colors . . . 66
5.2.4 How to make changes to the timetable . . . 68
5.3 Xcode . . . 70
5.3.1 Timetable View . . . 70
5.3.2 ColorCalendar View . . . 72
5.3.3 Holidays View . . . 73
5.3.4 Courses View . . . 74
5.3.5 Views . . . 76

6 Test 79
6.1 Test . . . 79
6.1.1 Figma: First part of the test . . . 80
6.1.2 Figma: Second part of the test . . . 81
6.1.3 Xcode: Test . . . 82

7 Conclusion 84

A Test 85
A.1 Test 1 85
A.2 Test 2 87
A.3 Test 3 87

Bibliography 89

List of Figures

1.1 UX vs UI [1] 4
1.2 The world of UX and UI [2] 5

2.1 Todoist 8
2.2 Microsoft To Do 8
2.3 Calendly 8
2.4 iStudiez 8
2.5 myHomework 9
2.6 Power Planner 9
2.7 ClickUp 9

3.1 Persona Template 16
3.2 Example Persona 16
3.3 Example of a storyboard 19
3.4 HTA tree 20
3.5 Diagrammatic HTA 21
3.6 Example of a wireframe sketch 23
3.7 Example of a mock-up 24
3.8 Lo-Fi Prototype 27
3.9 Hi-Fi Prototype 29

4.1 Mobile Vendor Market Share Europe from 2021 to 2023 35
4.2 Ideal User Persona 38
4.3 First Storyboard 40
4.4 Second Storyboard 41
4.5 Third Storyboard 42
4.6 First HTA 44
4.7 Second HTA 45
4.8 Third HTA 46
4.9 Insert your schedule 48
4.10 Set your schedule 49

4.11 Day View . . . 50
4.12 Week View . . . 50
4.13 2-Weeks View . . . 51
4.14 Adding marks . . . 52
4.15 Add a new Timetable . . . 53
4.16 Set the Timetable . . . 54
4.17 Add a new Course . . . 55
4.18 Set Alert and Repeat . . . 55
4.19 Day View . . . 55
4.20 Week View . . . 56
4.21 Month View . . . 57
4.22 Year View . . . 57

5.1 Add a new Timetable . . . 62
5.2 Set the Timetable . . . 62
5.3 Colour . . . 62
5.4 Duration . . . 62
5.5 Month View . . . 63
5.6 New Course . . . 63
5.7 Set Repeat and Alert . . . 64
5.8 Day and Week Views . . . 65
5.9 Month and Year Views . . . 65
5.10 Color Palette . . . 67
5.11 Texture: Second option, first variant . . . 68
5.12 Texture: Second option, second variant . . . 68
5.13 Texture: Fourth option . . . 68
5.14 Update Lesson . . . 69
5.15 Delete Lesson . . . 69
5.16 Delete Course . . . 70
5.17 TimetableView . . . 71
5.18 TimetableView Updated . . . 71
5.19 ColorCalendar View . . . 72
5.20 New Holiday . . . 73
5.21 Holidays . . . 73
5.22 Set Date . . . 74
5.23 New Course . . . 75
5.24 Courses . . . 75
5.25 Calendar . . . 76
5.26 Lecture . . . 76
5.27 Lab . . . 76
5.28 Day View . . . 77

5.29 1-Week View 78
5.30 2-Weeks view 78

Acronyms

UI
: user interface

UX
: user experience

UCD
: user centered design

HTA
: hierarchical task analysis

Lo-Fi
: low fidelity

Hi-Fi
: high fidelity

Chapter 1

Introduction

"Design isn't just
what it looks like and feels like
— design is how it works."
Steve Jobs

In the past decade, we've witnessed a remarkable transformation in the realm of digital applications. Gone are the days of static interfaces and clunky user experiences. Today, applications have evolved into dynamic, intuitive platforms that seamlessly integrate into users' daily lives. This evolution has been driven by a multitude of factors, including advancements in technology, changes in user behavior, and the rise of design-centric methodologies.

One of the most notable changes in modern applications is the shift towards mobile-first design. With the proliferation of smartphones and tablets, users now expect applications to be accessible anytime, anywhere, and on any device. This has led to the emergence of responsive design principles, where applications dynamically adapt their layout and functionality based on the device's screen size and orientation. Mobile apps have become the primary interface for many users, necessitating a focus on touch-friendly interactions, intuitive gestures, and seamless navigation.

Moreover, the rise of cloud computing and web-based technologies has enabled applications to break free from the confines of traditional platforms and embrace a more flexible and interconnected approach. Today's applications leverage APIs, microservices, and cloud infrastructure to deliver real-time updates, personalized experiences, and seamless integration with other services and platforms. This interconnectedness has blurred the lines between different applications and ecosystems, leading to a more cohesive and interconnected digital experience for users. At the heart of this evolution lies the importance of design. In an increasingly competitive landscape, where users have countless options at their fingertips, the

design of an application can make or break its success. A well-designed application not only captivates users with its visual appeal but also enhances usability, accessibility, and overall satisfaction. Design is no longer just about aesthetics; it's about understanding users' needs, behaviors, and motivations, and crafting experiences that resonate with them on a deeper level. In today's digital age, where user expectations are higher than ever, the design of an application is more important than ever before. It's not enough for an application to be functional; it must also be intuitive, engaging, and memorable. Design has become a strategic imperative for businesses looking to differentiate themselves in the market, build brand loyalty, and drive user engagement and retention. As we continue to navigate the ever-changing landscape of digital technology, the importance of design will only continue to grow, shaping the future of applications and the experiences they offer to users.

At the heart of this evolution lie two fundamental concepts: User Experience (UX) and User Interface (UI). As we delve into the intricacies of modern application design, it is essential to understand the significance and interplay of these concepts in crafting compelling and intuitive user experiences.

1.1 User Experience (UX)

User experience encompasses the overall journey and perception that users have while interacting with a digital product or service. It extends beyond the visual interface to encompass factors such as usability, accessibility, performance, and emotional engagement. UX design focuses on understanding user needs, behaviors, and motivations to create experiences that are intuitive, efficient, and meaningful.

1.1.1 Key Features of UX

Here are the key features of the UX through which it can be better understood:

1. **User Research**: UX design begins with user research, which involves gathering insights into user behaviors, preferences, and pain points. This research informs design decisions and ensures that the product meets user needs effectively.

2. **Usability**: Usability is a core aspect of UX design, emphasizing the ease of use and efficiency of interactions within the interface. UX designers strive to minimize friction and cognitive load, enabling users to accomplish tasks quickly and intuitively.

3. **Information Architecture**: Information architecture involves organizing and structuring content within the interface in a logical and hierarchical manner.

This includes considerations such as navigation design, content categorization, and labeling, ensuring that users can find information easily and navigate the interface effectively.

4. **Feedback and Iteration**: UX design emphasizes a iterative approach, incorporating user feedback and testing throughout the design process. This enables designers to refine and improve the interface based on real-world usage and user insights, ensuring continuous optimization and enhancement.

1.2 User Interface (UI)

The user interface of a digital product serves as the gateway to its functionality, presenting users with a visual and interactive platform through which they can navigate and engage. UI encompasses the layout, design elements, and interactive components that users interact with, including buttons, menus, forms, and visual elements. Its primary purpose is to facilitate intuitive and efficient interactions, guiding users through their journey and enabling them to accomplish their goals with ease.

1.2.1 Key Features of UI

The key features of the UI are:

1. **Visual Design**: UI design focuses on creating visually appealing and cohesive interfaces that align with the brand's identity and convey information effectively. This includes considerations such as color schemes, typography, iconography, and layout composition.

2. **Interactivity**: Interactive elements such as buttons, links, and menus provide users with ways to navigate the interface and perform actions. UI design ensures that these elements are clear, consistent, and responsive, enabling seamless interactions and feedback.

3. **Accessibility**: UI design prioritizes accessibility, ensuring that the interface is usable and navigable for users of all abilities. This includes considerations such as contrast ratios, text legibility, keyboard navigation, and screen reader compatibility.

4. **Consistency**: Consistency in UI design fosters familiarity and predictability for users, reducing cognitive load and facilitating learning. This involves maintaining uniformity in design elements, layout patterns, and interaction patterns throughout the interface.

1.3 User Experience VS User Interface

Due to their close collaboration, UX and UI designers are frequently mistaken for one another, despite their distinct roles within the design process. While there is certainly overlap between the two disciplines, it's essential to recognize the fundamental differences that distinguish them.

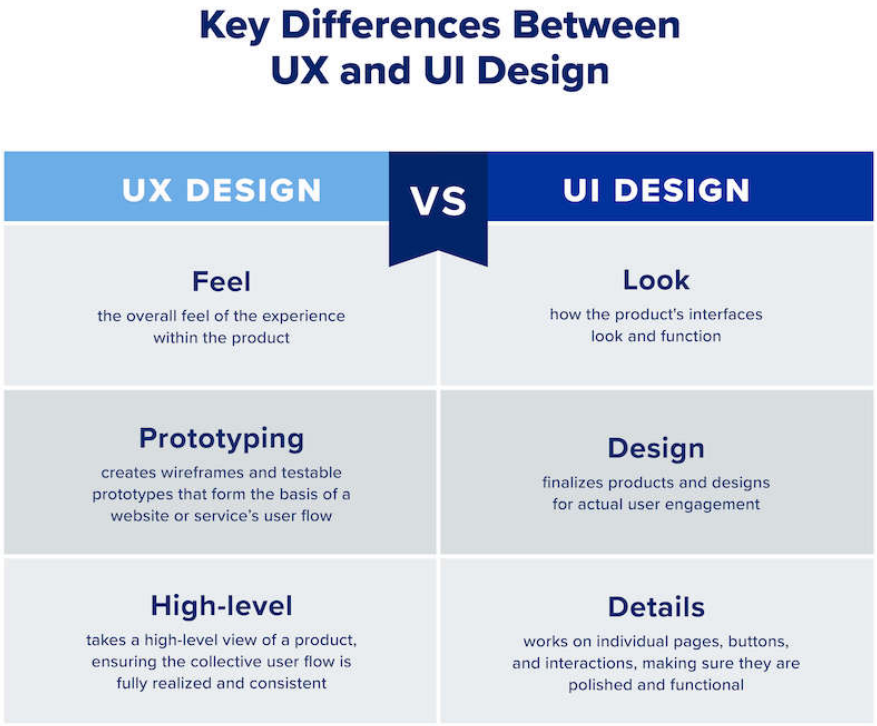

Figure 1.1: UX vs UI [1]

Feel vs. Look

UX and UI design are integral to a product's development, each playing distinct yet interconnected roles. While UX design is more concerned with the holistic feel and functionality of the product or service. In essence, UI design focuses on creating a visually appealing interface, whereas UX design is dedicated to crafting a seamless and meaningful experience for users, ensuring that every interaction contributes to their satisfaction and overall perception of the product, UI design primarily concerns itself with the visual aspects and interactive elements (look) that shape a user's experience.

Prototyping vs. Design

While UX and UI designers collaborate on the same product, their roles and objectives diverge. UI designers concentrate on refining and enhancing the visual elements of the product, aiming to create designs that captivate users and drive engagement. Conversely, UX designers are tasked with developing wireframes and prototypes to establish the user flow of a website or service, focusing on functionality and usability.

High-level vs. Details

Furthermore UI and UX designers diverge in the scope and granularity of their work. UX designers adopt a broader perspective, focusing on the overall user flow and coherence of a product or service across multiple pages or screens. Their goal is to ensure a consistent and fully realized user experience throughout the entire journey. Conversely, UI designers meticulously craft individual elements such as pages, buttons, and interactions, ensuring they are not only visually polished but also function seamlessly.

In summary, UI and UX collaborate to shape unified, user-centered digital experiences. While UI concentrates on the visual and interactive elements of the interface, UX encompasses the holistic user journey and perception, guaranteeing effective fulfillment of user needs and the delivery of a smooth and gratifying experience.

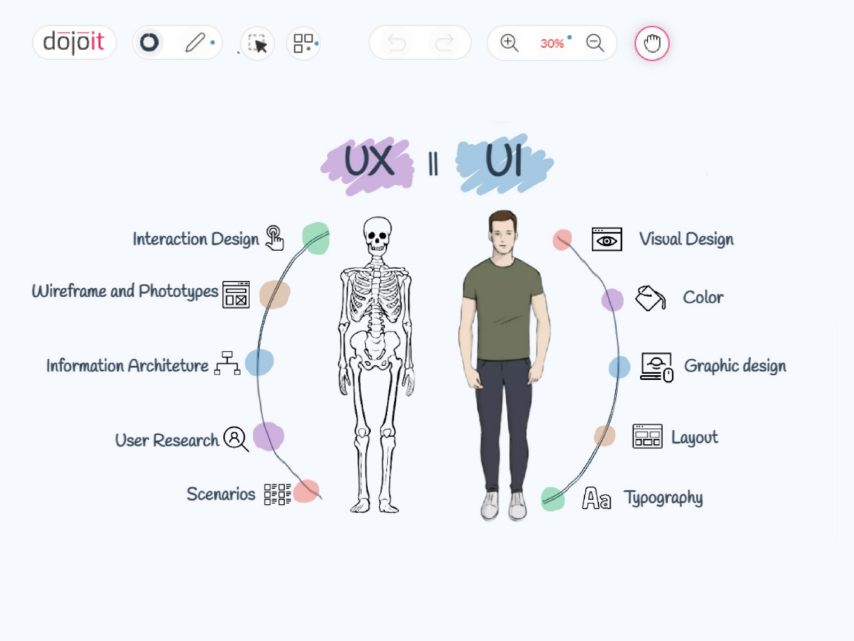

Figure 1.2: The world of UX and UI [2]

1.4 Aim of the project

Building upon the foundational understanding of User Experience (UX) and User Interface (UI) design principles, the aim of this thesis is to redesign an existing iOS application in order to improve its features.

In the realm of mobile applications, the Apple Calendar app stands as a cornerstone for iOS users, facilitating organization and time management in their daily lives. However, despite its utility, the app lacks specialized templates tailored to specific contexts, such as university schedules. This limitation prompts the need for a redesign aimed at integrating new templates, offering iOS users the ability to seamlessly incorporate their academic timetables into their digital calendars.

The motivation behind this project stems from the recognition of a common struggle among students to efficiently manage their academic schedules. While the Apple Calendar app provides a solid foundation for time management, its lack of predefined templates hinders users from easily inputting and organizing complex schedules, such as university courses, assignments and exams.

So, by introducing customizable templates tailored to academic needs, this redesign aims to alleviate the burden on students and enhance their productivity by seamlessly integrating their academic commitments with their daily routines.

Chapter 2

State of the Art

"A user interface is like a joke.
If you have to explain it, it's not that good."
Martin LeBlanc

2.1 What does the market already offer?

Before delving into the proposed enhancements for the Apple Calendar application, it's essential to recognize the existence of several third-party apps that already cater to the specific needs of university scheduling. In fact when it comes to school planner apps, there's no shortage of options available.

According to the article "7 best student planner apps" [3], there are seven highly-rated apps that cater to the needs of high school and college students:

1. **Todoist**: A versatile project planner and to-do list manager with features suitable for students and professionals alike. It offers robust task management capabilities and customizable options.

2. **Microsoft To Do**: An evolved version of Wunderlist, offering task organization, subtasks, and visually appealing design options. It's particularly useful for managing daily tasks and assignments.

3. **Calendly**: Simplifies scheduling meetings with individuals and groups, integrating seamlessly with digital calendars. Ideal for students coordinating group projects and extracurricular activities.

4. **iStudiez Pro**: Specifically designed for students, iStudiez Pro allows users to organize class schedules, assignments, and exams. It offers a comprehensive view of academic commitments and includes GPA tracking features.

5. **myHomework**: A clean and straightforward planner app tailored for educational use. It supports various class scheduling formats and allows for task prioritization and deadline reminders.

6. **Power Planner**: Offers features similar to iStudiez Pro and myHomework, including schedule management and GPA tracking. Notable for its responsive developer support and robust GPA calculator.

7. **ClickUp**: A comprehensive productivity platform suitable for managing complex projects and daily assignments. Features include customizable task organization, document editing, and integration with external tools.

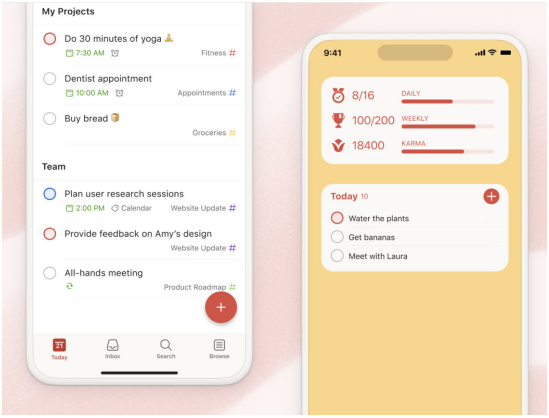

Figure 2.1: Todoist

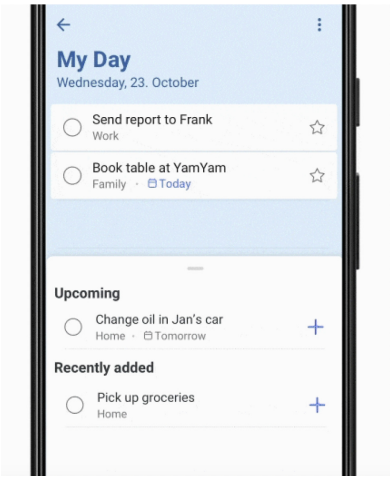

Figure 2.2: Microsoft To Do

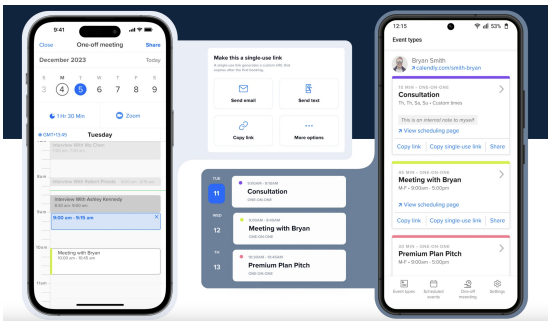

Figure 2.3: Calendly

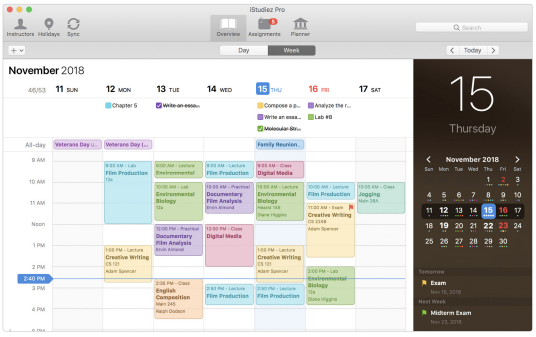

Figure 2.4: iStudiez

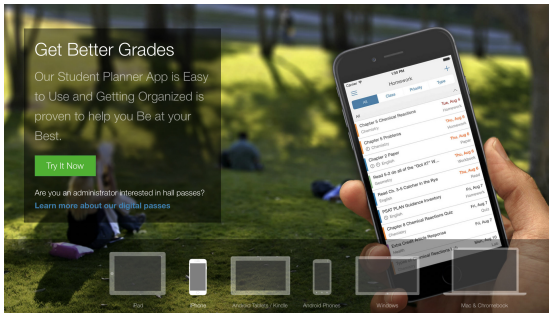

Figure 2.5: myHomework

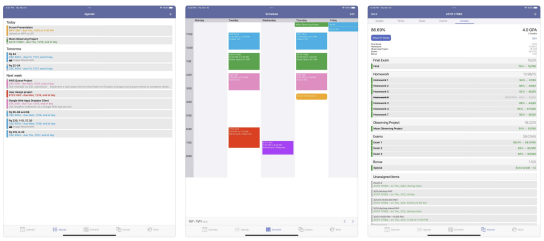

Figure 2.6: Power Planner

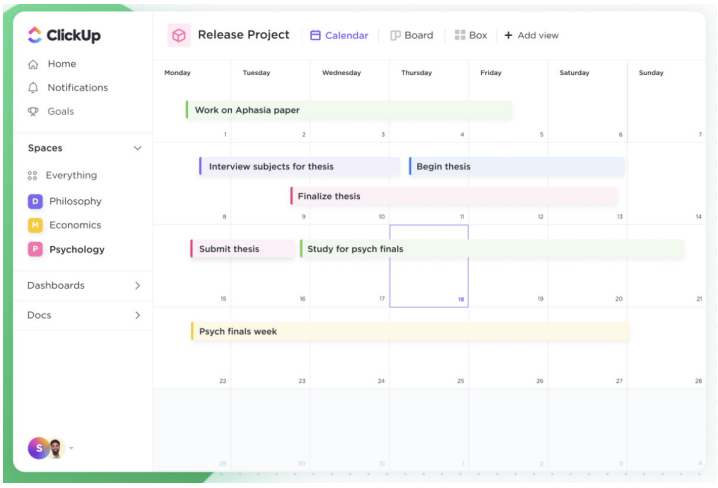

Figure 2.7: ClickUp

2.2 So why create something that already exists?

Despite the availability of these third-party solutions, the rationale behind proposing improvements to the Apple Calendar application remains compelling for several reasons:

- **Enhancing Pre-installed Accessibility**: Apple Calendar comes pre-installed on every Apple device, providing users with immediate access without the need to download additional apps. By improving the functionality of this native application, users can avoid the hassle of installing and learning to use third-party alternatives.
- **Cost Efficiency and Integration**: Unlike many third-party apps that offer premium features for a fee or only provide limited functionalities for free, Apple Calendar is entirely free to use and integrates seamlessly with other Apple services through iCloud synchronization. This ensures a consistent and cost-effective solution, especially for students who may have limited financial resources and rely on their parents for financial support.
- **Widget Functionality**: Apple Calendar offers the advantage of native widget functionality, allowing users to access timely information at a glance directly from their home screens. While some third-party apps may offer widgets, not all are as readily available or customizable as those within the Apple ecosystem. Avoiding the need to download additional widget apps such as Widgetsmith further simplifies the user experience and enhances convenience.

Therefore, the exploration of these apps in their free versions has yielded valuable insights, serving as a wellspring of inspiration for refining the Apple Calendar application. By assimilating and adapting pivotal features from these top-rated apps into the Apple Calendar's design, users can anticipate a harmonized and instinctive scheduling experience tailored precisely to their academic requisites.

Chapter 3

Methods

"If you want to create a great product,
you have to start by understanding the people who will use it."
Don Norman

3.1 User Centered Design (UCD)

The expression "user-centered design" was coined by Donald Norman, an American psychologist and engineer, in 1986 with the publication of the book titled "User-Centered System Design: New Perspectives on Human-Computer Interaction".

User-centered design (UCD) is a specific approach within the realm of user-centric design methodologies. It's characterized by an iterative process where designers prioritize the needs, goals, and feedback of users throughout product development.

The primary objective of UCD is to optimize the usability of the product that refers to the effectiveness, efficiency, and satisfaction with which users can interact with a product or system to achieve their goals. Let's look at these three aspects in more detail:

1. **Effectiveness**: The extent to which users can accomplish their intended tasks accurately and completely within the product.

2. **Efficiency**: The speed and ease with which users can perform tasks once they have learned how to use the product.

3. **Satisfaction**: The overall user experience and satisfaction levels, including aspects such as aesthetics, ease of use, and emotional response to using the product.

Usability testing and evaluation methods aim to assess these aspects systematically, providing insights into areas where the product design can be improved to enhance the overall user experience.

Norman expanded upon the UCD principle in his influential work "The Psychology Of Everyday Things" (POET) (Norman, 1988). In POET, he acknowledges the user's needs and interests while emphasizing the importance of design usability. He outlines four fundamental principles for effective design:

1. Make it easy to determine what actions are possible at any moment.

2. Make things visible, including the conceptual model of the system, the alternative actions, and the results of actions.

3. Make it easy to evaluate the current state of the system.

4. Follow natural mappings between intentions and the required actions; between actions and the resulting effect; and between the information that is visible and the interpretation of the system state.

These suggestions prioritize placing the user at the core of the design process, but, before talking about the functions performed by users during the design process, let us make a distinction between different types of users:

1. **Primary Users**: These are the individuals who directly engage with the product or service as part of their regular activities or tasks. They are the primary audience for whom the product is designed to meet specific needs and goals. For example, primary users of a mobile banking app would be the customers who use it to check their account balances, transfer funds, or pay bills.

2. **Secondary Users**: Secondary users may interact with the product or service indirectly or infrequently. They may be affected by the product or have some involvement in its use, but they are not the primary target audience. For instance, in the context of the mobile banking app, secondary users could include family members who occasionally check account balances but do not actively manage the accounts.

3. **Tertiary Users**: Tertiary users are individuals or entities who have an interest in the product or service but do not directly interact with it. They may include stakeholders, regulators, or other third parties who are impacted by the product's use or outcomes. In the case of the mobile banking app, tertiary users could include government agencies responsible for regulating financial transactions or investors interested in the app's performance.

Considering the needs, preferences, and behaviors of all these user types is essential for creating a product or service that effectively meets the requirements of its entire user base. This comprehensive approach ensures that the design is inclusive and accessible to all relevant stakeholders. Having defined these three types, we can now talk about the role each user plays during the design phase. Here are some key roles users may take on during UCD:

- **Requirements Gathering**: Users are involved in the initial phase to provide insights into their needs, goals, and challenges. This helps designers understand the context in which the product will be used and identify specific requirements to address.

- **Feedback and Evaluation**: Throughout the design and development process, users participate in usability testing, user experience evaluations, and feedback sessions. They interact with prototypes or early versions of the product and provide feedback on its usability, functionality, and overall user experience.

- **Iterative Design**: Users' feedback and insights inform iterative design cycles. Designers refine and iterate on the product based on user feedback, ensuring that design decisions align with user needs and preferences.

- **Validation**: Users validate the final product to ensure that it meets their expectations and effectively addresses their needs. This validation may involve testing the product in real-world scenarios or reviewing final designs to provide final approval before launch.

Overall, users' active involvement in the UCD process helps designers create products that are intuitive, user-friendly, and ultimately more successful in meeting user needs and achieving business objectives.

3.2 Methodologies

User-Centered Design (UCD) encompasses various methodologies and approaches aimed at ensuring that products and systems meet the needs, preferences, and abilities of users.

So, in this chapter there will be a theoretical discussion of all the most commonly used methodologies in UCD, starting with the choice of the user who will use the final product and ending with the testing of the latter. In subsequent chapters, these methodologies will be taken up one by one and adapted to the specific case of the Apple Calendar application, the objective of my thesis.

3.2.1 Persona Development

Personas are fictional characters created to represent different user groups based on demographic and psychographic data. Persona development helps designers empathize with users and design products that cater to their specific needs and goals. The advantages of using personas in the design process are manifold:

- **Facilitates stakeholder discussions**: Personas provide a structured framework for stakeholders to discuss and prioritize critical features of a redesign. By focusing on common user interactions, personas help uncover frustrations and pain points, clarifying actual user priorities over personal stakeholder preferences.
- **Promotes shared understanding among team members**: Personas distill complex user data into compelling narratives, making it easier for team members to share a consistent understanding of the target user group. This shared understanding fosters collaboration and alignment when working towards solutions.
- **Guides informed design decisions**: By focusing on user needs and goals, personas help inform the development of wireframes and site architecture. Teams can walk through user scenarios to determine the optimal placement of content, ensuring that the design effectively supports user objectives. This approach can ultimately save time and resources by avoiding costly rework after launch.
- **Humanizes the user experience**: Personas provide a tangible representation of the target audience, fostering empathy and understanding among designers and developers. By putting a "face" to the user story, personas help prevent the imposition of personal mental models onto the product design, ensuring that the final product aligns with actual user needs and preferences.

Because personas represent fictionalized versions of real users, their effectiveness relies on a thorough understanding of your target audience. The more comprehensive your knowledge of users, the more effective your personas will be. Typically, creating 4-7 personas, derived from various user groups, suffices to encapsulate the majority of user requirements. For projects involving diverse user groups, it's beneficial to categorize them into clusters based on similarities, thereby condensing the user groups into 4-7 distinct profiles.

Depending on the project, you may have access to existing users or not. Fortunately, there are several methods to gather data for either scenario.

For existing users:

- Conduct user interviews, the most valuable method. Patterns emerging from multiple interviews can inform persona creation for the target audience group.

- Analyze help desk calls, website feedback, surveys, questionnaires, focus groups, and purchase records for valuable insights.

- Encourage existing clients to participate in surveys, offering incentives like gift cards or prize entries.

For non-existing users:

- Use original research, brainstorming personas based on product goals and objectives.

- Consult stakeholders for insights into their anticipated customer base, potentially providing contacts for collaboration. Once you have the necessary information, craft persona stories for each user group, using your imagination for this aspect.

Each Persona is characterised by several key points, such as:

- Persona Group (e.g., sales manager);

- Fictional name (e.g., "Government Gail");

- Job title and primary responsibilities;

- Optional: Representative quote;

- Demographics: Age, education, ethnicity, family status;

- Goals and tasks on the website or application;

- Pain points or frustrations;

- Technical expertise and background;

- Working environment: Office setting, distractions, etc;

- Casual photos: Realistic images sourced from dating sites or online searches are preferable to stock photos.

Various persona templates are available online, which can be adapted to better align with the user needs and experiences of your product.

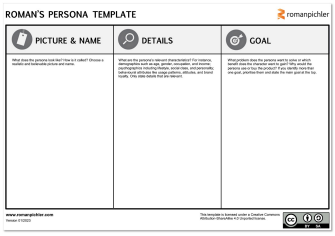

Figure 3.1: Persona Template

Figure 3.2: Example Persona

3.2.2 Scenarios

A scenario is essentially an informal narrative story that depicts the daily experiences or a sequence of events involving the primary stakeholder group as the central character. Typically, the main character of this narrative is based on a persona created earlier in the process. The story should outline specific events relevant to the challenges faced by the primary stakeholder group, addressing the key research questions guiding the design process.

While the scenario may depict the everyday life of an individual, it should include subtle details that offer insights into the users, potentially incorporating emotional or physical characteristics. Scenarios can take different forms, including the "best case scenario", where everything unfolds favorably for the main character, the "worst case scenario", where challenges abound, and an "average case scenario", representing the typical, uneventful day for the individual.

Scenarios provide a social context within which personas operate and help to establish a tangible physical environment, enhancing the understanding of the persona's existence. By presenting information in the form of a story, scenarios are more engaging and easier for people to comprehend compared to abstract character profiles.

Scenario example: Shared Calendar

"The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals' calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people's calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it should be confirmed before it is written in."

3.2.3 Use-Cases

In User-Centered Design (UCD), use cases play a crucial role in understanding how users interact with a system or product. A use case is a description of a specific interaction between a user and the system to achieve a particular goal. It outlines the steps a user takes to accomplish a task within the system, along with any possible variations or alternative paths. Use cases are typically written in a structured format, including:

1. Use Case Number or ID: A unique identifier assigned to the use case for reference and tracking purposes.

2. Use Case Title: A concise name that summarizes the objective or goal of the use case.

3. Text Description: A brief description of the use case, outlining the main activities or interactions between the user and the system.

4. Preconditions: Any conditions or requirements that must be met before the use case can be initiated.

5. Flow of Events: The sequence of steps followed by the user to accomplish the task. This includes the basic path and any alternative paths.

   - Basic Path: The primary sequence of steps leading to the successful completion of the task.

   - Alternative Paths: Optional paths or deviations from the basic path that users may take under certain circumstances.

6. Postconditions: The expected state of the system after the use case has been successfully executed.

7. Special Conditions: Any exceptional or unique circumstances that may impact the execution of the use case, such as error handling or system limitations.

Use cases help designers and developers understand user requirements, clarify system functionality, and identify potential areas for improvement. They provide a clear framework for designing user interfaces and guiding the development process. Additionally, use cases serve as valuable documentation for stakeholders, ensuring that everyone involved in the project has a shared understanding of user needs and system functionality.

Example Use-Case: Path for use case for shared calendar

1. *The user chooses the option to arrange a meeting.*
2. *The system prompts user for the names of attendees.*
3. *The user types in a list of names.*

4. *The system checks that the list is valid.*
5. *The system prompts the user for meeting constraints.*
6. *The user types in meeting constraints.*
7. *The system searches the calendars for a date that satisfies the constraints.*
8. *The system displays a list of potential dates.*
9. *The user chooses one of the dates.*
10. *The system writes the meeting into the calendar.*
11. *The system emails all the meeting participants informing them of them appointment.*

3.2.4 Storyboard

A storyboard is a visual representation used to outline the usage requirements of an application, service, or product during the early stages of a project. Through images, drawings, or collages, a storyboard illustrates the sequential steps of an interaction. Unlike wireframes, which focus on specific sections of an application, storyboards present interactions from the user's viewpoint, often resembling a comic strip. By providing a holistic view of usage requirements, storyboards facilitate discussions on potential weaknesses or areas for improvement. They effectively visualize the user's perspective and facilitate gathering valuable feedback.

So, a storyboard is a tool that integrates scenarios and use-cases by depicting the sequence of tasks involved in user interactions with the future system to achieve specific goals. Each storyboard typically represents one scenario, potentially including alternative endings. Unlike detailed use-case descriptions, storyboards do not provide explicit visualizations of user interface elements or in-depth descriptions of system interactions. Instead, they focus on illustrating the flow of events and user actions within a given scenario.

Additionally, storyboards serve as a communication tool within the development team, ensuring a shared understanding of the project vision, in fact in this way designer has a big picture of the whole situation and, at the same time, users can evaluate intuitively the design plans.

Figure 3.3: Example of a storyboard
[4]

3.2.5 Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis (HTA) is a method used to decompose complex tasks into smaller, more manageable subtasks. It involves breaking down tasks hierarchically, starting from the main goal or objective and progressively dividing it into smaller actions or steps.

HTA Tree

In order to do it, here are the steps to follow:

1. **Identify the Main Task**: Start by identifying the main task or goal that you want to analyze.

2. **Break Down the Task**: Break down the main task into its sub-tasks or sub-goals. Each sub-task should represent a distinct step towards achieving the main task.

3. **Create a Hierarchy**: Organize the sub-tasks hierarchically, with the main task at the top and the sub-tasks branching out beneath it.

4. **Further Decompose Sub-Tasks**: If necessary, further decompose the sub-tasks into smaller, more detailed actions or steps.

5. **Add Detail**: For each sub-task or action, add detailed descriptions or annotations to clarify what it entails. This could include descriptions of inputs, outputs, conditions, and any decision points.

6. **Review and Refine**: Review the HTA tree to ensure that it accurately represents the task structure and relationships. Refine the hierarchy and descriptions as needed.

7. **Validate with Users**: Consider validating the HTA tree with actual users to ensure that it aligns with their mental models and task execution strategies.

8. **Document and Communicate**: Document the HTA tree in a clear and organized format, such as a diagram or spreadsheet. Use it to communicate task structure and requirements to stakeholders, designers, and developers.

This decomposition forms a tree-like structure, with the goal at the root and progressively detailed tasks and subtasks branching out from it.

While the hierarchy represents the task structure, the plans describe the execution process. Plans consist of sequences of steps that need to be performed to achieve each task and subtask. However, it's important to note that only the leaf nodes of the hierarchy, representing the smallest, indivisible tasks, are included in the plans for execution. These plans detail the specific actions users need to take to accomplish each task effectively.

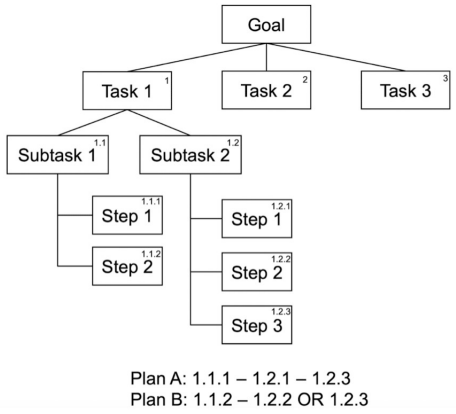

Figure 3.4: HTA tree

Diagrammatic HTA

In addition to traditional tree-style Hierarchical Task Analysis (HTA) diagrams, there exists a diagrammatic version. In this variation, a line drawn under a box

signifies that the task represented by that box does not require further expansion. Additionally, plans for executing the tasks may be depicted directly on the diagram or provided separately.

This diagrammatic approach offers an alternative format for representing tasks and their relationships, providing a visual representation that may complement traditional tree-based HTA diagrams.

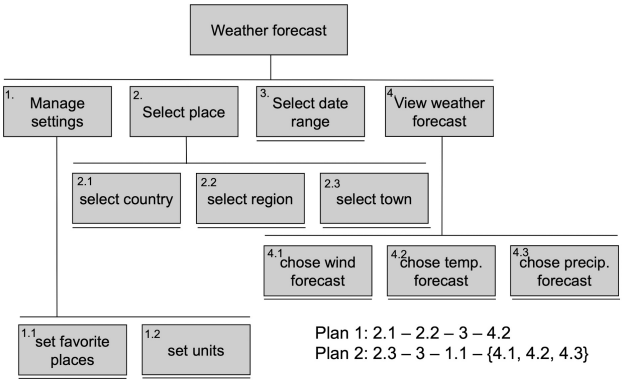

Figure 3.5: Diagrammatic HTA

Alternative description of HTA

HTA can also be presented in a textual or structured form, providing a more condensed representation of tasks and their relationships compared to graphical formats. However, this method may be more challenging to read and interpret due to its compact nature.

Example of a textual HTA

0. *In order to borrow a book from the library*
   1. *go to the library*
   2. *find the required book*
      2.1 *access library catalogue*
      2.2 *access the search screen*
      2.3 *enter search criteria*
      2.4 *identify required book*
      2.5 *note location*
   3. *go to correct shelf and retrieve book*
   4. *take book to checkout counter*

- *plan 0:*
  1. *do 1 - 3 - 4.*
  2. *if book isn't on the shelf expected do plan 2 - 3 - 4.*
- *plan 1:*
  1. *do 2.1 - 2.4 - 2.5.*
  2. *if book not identified do plan 2.2 - 2.3 - 2.4.*

In conclusion, Hierarchical Task Analysis (HTA) serves as a versatile tool for UI design, facilitating the capture of requirements and system design in various ways. It models user interactions with the system, offering insights into what features to include or omit, and prioritizing critical tasks. Furthermore, HTA aids in detailed interface design by mapping plans to dialog paths and informing menu design based on task decomposition. Additionally, HTA-generated scenarios prove invaluable for user evaluation tests, as well as for developing manuals, training materials, and help systems.

3.2.6 Sketches

A sketch serves as a means to describe ideas, prompting questions and inspiring inventions. It typically consists of hand-drawn illustrations and does not necessarily originate from the designer. It does not provide explicit or detailed solutions, but rather encourages exploration and creativity.

There are many benefits derived from sketches, shown below:

- **Cost-Effectiveness:** Sketches offer a high cost-effectiveness ratio. Since they only require paper and pencil, or simple digital tools, they are a budget-friendly way to explore and communicate design ideas.
- **Speed of Production:** Sketches can be created rapidly. This speed allows designers to explore a wide range of design options in a short amount of time and obtain timely feedback.
- **Creativity Exploration:** Sketches encourage creativity and innovation. As there is no pressure to create perfect or complete designs, designers can freely explore new ideas and solutions without fear of failure.
- **Overcoming Language Barriers:** Sketches overcome language barriers. They can be understood by people from different cultures and languages, as they are based on images and symbols rather than written text.

Figure 3.6: Example of a wireframe sketch
[5]

- **Support for Collaboration:** Sketches facilitate collaboration among team members. They can be used during brainstorming and design thinking sessions to actively engage all participants in the ideation process.

- **Documentation Tool:** Sketches can be used as a documentation tool during the design process. They can be archived and used to track the evolution of design ideas over time.

- **Practical Testing:** Sketches can be used to conduct practical tests with users. Designers can create prototypes based on sketches and present them to users to assess their understanding and acceptance of the proposed interfaces.

In summary, sketches are a versatile and powerful tool in the user-centered design process, offering numerous advantages in terms of creative exploration, communication, iteration, and collaboration.

3.2.7 Mock-ups

Mock-ups are static representations of design concepts, typically created using design software or prototyping tools. They provide a visual representation of the final product and allow stakeholders to provide feedback on the design.

Here are some key points about mockups:

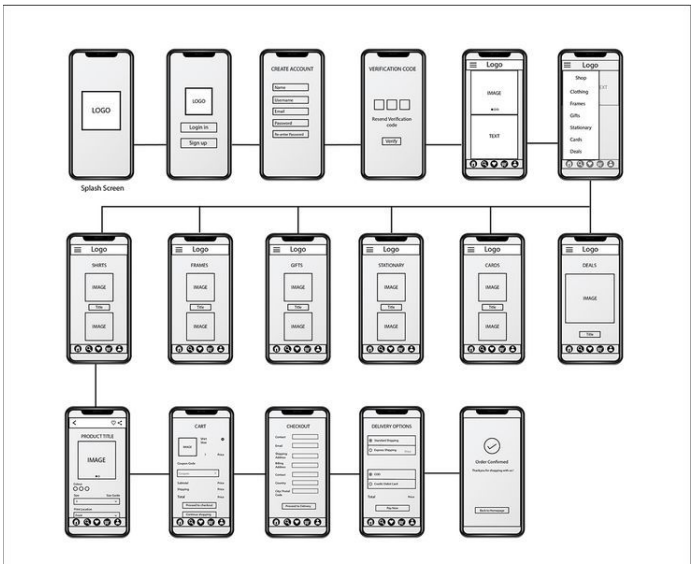

Figure 3.7: Example of a mock-up
[6]

- **Visual Representation:** Mockups provide a visual representation of the user interface, including layout, typography, colors, and imagery. They allow stakeholders to visualize the overall design and aesthetics of the product.

- **Functional Demonstration:** Mockups can demonstrate the basic functionality and interaction flow of the product. While they may not include fully functional features, they should provide enough interactivity to simulate user interactions and demonstrate key user flows.

- **Iterative Design:** Mockups facilitate an iterative design process by allowing designers to quickly create and test different design concepts. They can be easily modified based on feedback from stakeholders and users, helping to refine the design and improve usability.

- **Feedback Gathering:** Mockups are valuable tools for gathering feedback from stakeholders and users. They can be presented in meetings or user testing sessions to solicit feedback on the design, layout, and functionality of the product.

Overall, mockups play a crucial role in the UCD process by helping to visualize, test, and refine design concepts before moving into the development phase. They contribute to the creation of user-friendly and visually appealing products that meet the needs and expectations of users.

3.2.8 Prototyping

Prototyping is a vital aspect of the design process, characterized by the creation of successive iterations of a product. With each iteration, feedback and enhancements from stakeholders and users are integrated, leading to incremental improvements.

These prototypes serve as tangible representations of design concepts, providing a concrete visualization of the product's functionality and user interface. Unlike mere descriptions or sketches, prototypes offer a more comprehensive understanding of the design.

A key objective of prototyping is to elicit feedback from users, enabling designers to pinpoint usability issues, gather suggestions for enhancement, and validate design choices. By testing prototypes with real users, designers can refine their designs to better meet user needs.

Furthermore, prototyping mitigates the risks associated with product development. By identifying potential issues early on, designers can proactively address them, minimizing the likelihood of costly revisions later in the development process.

Prototypes also serve as effective communication tools, facilitating discussions among stakeholders, clients, and development teams. Visual prototypes make it easier for non-designers to grasp the intended functionality and user experience of the product.

Additionally, prototyping enables designers to explore various design solutions. By creating multiple prototypes with different features and interactions, designers can compare and evaluate alternatives to determine the most optimal solution for the product. Prototypes typically iterate on top of each other as they are meant to test specific interactions of a product and improve upon them. With this concept, prototypes are commonly split into two types:

- Low-Fidelity Prototypes (Lo-Fi);
- High-Fidelity Prototypes (Hi-Fi).

Lo-Fi Prototyping

Lo-Fi (Low-Fidelity) prototyping involves creating simple, low-tech prototypes using paper, cardboard, or digital tools. These prototypes allow designers to quickly test and iterate on design concepts before investing in more detailed prototypes.

There are two main types of low-fidelity prototypes commonly used by designers:

- **Paper prototyping**: Paper prototyping involves sketching basic user interfaces on paper or a whiteboard. Designers and stakeholders use these simple prototypes to quickly capture ideas for new or redesigned user experiences. While traditional paper prototypes can generate valuable internal feedback

within small teams, online whiteboards enable prototyping with larger, hybrid teams.

- **Wireframing**: Wireframing, on the other hand, utilizes simple blueprints of digital screens to outline basic layouts and content hierarchy. Wireframes can also illustrate user interactions in a user flow, which is beneficial for usability testing. For example, Figma's wireframing tool facilitates the creation of low-fidelity assets that can seamlessly integrate with design system components.

Low-fidelity prototypes offer several benefits for the overall user experience:

1. Rapid Collection of User Feedback: Users are more inclined to provide feedback on works in progress, such as rough sketches or wireframes, compared to near-final designs. Prototypes with fewer features or less detail can be assessed more quickly.

2. Risk Reduction in Product Development: Addressing user interface issues early in the design and development process helps mitigate risks associated with product development.

3. Workflow Optimization: By identifying and addressing usability issues early on, low-fidelity prototypes help streamline workflows, avoiding costly and time-consuming revisions later in the process.

4. Agile Design Process: Low-fidelity prototypes enable agile design processes by allowing for quick and easy adjustments and iterations.

5. Cross-Team Collaboration: Various teams, including strategists, developers, and project managers, can collaborate on low-fi prototypes, particularly with online prototypes that don't require specialized design skills.

Here's a simplified guide on how to create a low-fidelity prototype in four straightforward steps:

Step 1: Define the problem

Clearly articulate the design challenge or problem you're aiming to address. Collaborate with your business and UX research teams to ensure alignment on the goals of your design concept.

Step 2: Outline essential functionality

Identify the core elements and interactions that are crucial for solving the problem. Sketch out a basic design concept, mapping out its structure and flow using a simple storyboard or user flow diagram.

Step 3: Develop the prototype

Begin drafting the screens for your prototype. You can use traditional paper sketches, an online whiteboard, or leverage pre-made screen layouts available in

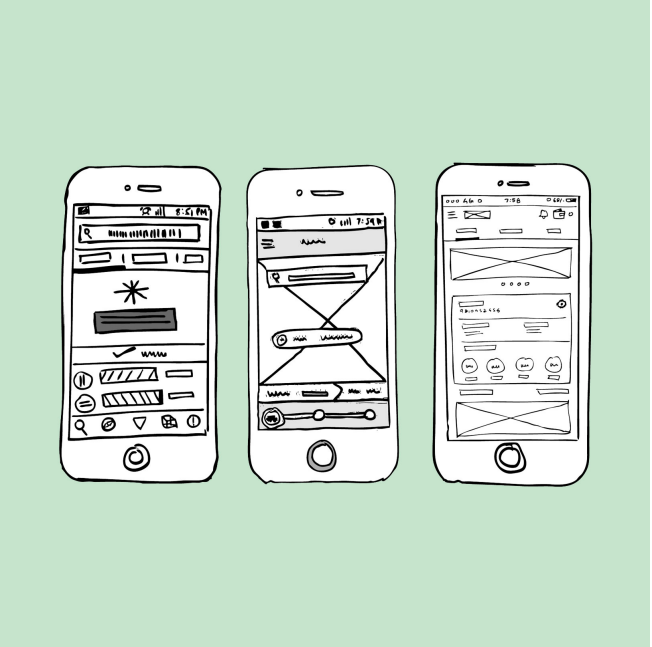

Figure 3.8: Lo-Fi Prototype
[7]

prototyping tools like FigJam. Keep each screen straightforward and high-level, focusing on key interactions and content. Remember to maintain a clear focus on addressing the identified problem.

Step 4: Test and iterate

Share your prototype(s) with internal teams to gather feedback and align on a design direction. Once you've selected a design, conduct user testing using the prototype to refine and improve your design. Consider adding annotations or comments to guide team members and users through the prototype and explain key features and interactions.

By following these four steps, any team can quickly create and iterate on low-fidelity prototypes to explore and validate design ideas without the need for extensive design expertise.

Hi-Fi Prototyping

Hi-Fi (High-Fidelity) prototyping involves creating more detailed, interactive prototypes that closely resemble the final product. These prototypes simulate user interactions and help designers evaluate usability and functionality more accurately. Refining design features and user flows through high-fidelity prototyping offers several advantages:

1. Enhanced user feedback: Clickable, interactive prototypes immerse end users, leading to deeper insights and detailed feedback on usability and UX design.

2. Stakeholder buy-in: High-fidelity prototypes enable team members, leadership, and investors to visualize the design, facilitating approval and generating enthusiasm for the final product.

3. Prevention of costly mistakes: By validating the final product design, high-fidelity prototypes help teams avoid expensive fixes during the development process.

4. Accelerated product development: Developers can use high-fi prototypes as references for product features and functionality, speeding up the development process with fewer iterations.

Creating high-fidelity prototypes can indeed be time-consuming. however, there are many prototyping tools including Figma (we will see its use in Chapter 5) that allow to add polish and functionality quickly without the need for coding.

With the right tools, the team can bring UX designs to life through hi-fi prototyping in just four steps:

Step 1: Research and plan

Determine the goals you aim to achieve with your high-fidelity prototype. Utilize UX research to define your purpose and objectives.

Step 2: Define key functionality and interactions

Identify the functionality and user interactions that will enhance your hi-fi prototype. Apply design thinking to pinpoint key features for your testers and stakeholders.

Step 3: Fill in your high-fidelity prototype

Initiate the process with detailed mock-ups of a user flow. Integrate visual assets, interactions, animations, and transitions from your design library.

Step 4: Test and refine your prototype

Enhance your designs through user testing. Present your high-fidelity prototype to key stakeholders to ensure that the UX aligns with your business and brand.

Lo-Fi vs Hi-Fi Prototyping

Low-fidelity prototypes differ significantly from their high-fidelity counterparts in that they do not closely resemble the final product. Instead, they span from rudimentary paper sketches to simple digital wireframes. These low-fidelity prototypes lack intricate interactions, animations, or transitions, which are typically present in high-fidelity prototypes.

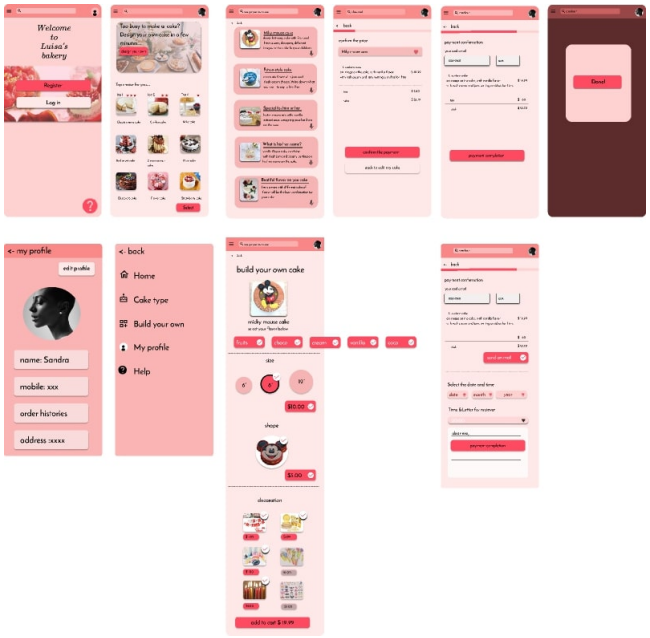

Figure 3.9: Hi-Fi Prototype
[8]

Hi-fi prototypes, in fact, have the look and feel of a finished product, with interaction designs that are practically pixel-perfect, they may include animations and clickable screens.

However, both low-fidelity and high-fidelity prototypes are integral to the product development process, each serving distinct purposes. Low-fidelity prototyping occurs in the initial phases of design, aimed at modeling and evaluating concepts before committing resources to development. On the other hand, high-fidelity prototyping occurs at later stages, allowing for comprehensive user testing to assess design functionality thoroughly.

3.2.9 Testing

Testing involves evaluating the usability and effectiveness of a design through user testing, usability testing, or other evaluation methods. It allows designers to gather feedback from users and identify areas for improvement in the design.

Testing is an integral aspect of the entire lifecycle of a product or service, spanning from the initial stages of ideation to the final live service available to the public. As user behaviors and interactions evolve continuously, ongoing testing and feedback are essential to ensure that the product or service remains relevant, efficient, and aligned with user needs.

Before conducting any testing, it's crucial to establish clear objectives, whether they stem from identified problems, hypotheses, or assumptions. Testing should focus on discovering the most effective solutions to address these objectives, rather than seeking evidence to support preconceived notions.

Prototypes serve as valuable tools for testing new ideas or concepts before their implementation in live environments. By allowing for experimentation and feedback gathering in controlled settings, prototypes enable teams to refine and optimize ideas prior to their public release. Once the prototyping phase is over, we proceed with the usability tests. Usability testing is a crucial component of the user-centered design process, aimed at evaluating a product's usability by testing it with real users. This method involves observing users as they interact with the product or system to identify any usability issues, gather feedback, and assess overall user satisfaction.

Here's an overview of usability testing:

1. **Objective Definition**: Before conducting usability testing, it's essential to define clear objectives and goals. What specific aspects of the product's usability are you aiming to evaluate? Establishing these objectives will guide the testing process and help in interpreting the results effectively.

2. **Participant Recruitment**: Identify and recruit participants who represent your target user demographic. The number of participants may vary depending on the scope of the project, but even testing with a small number of users (around 5-10) can uncover significant usability issues.

3. **Test Environment Setup**: Set up a conducive testing environment where participants can comfortably interact with the product or system. Ensure that the necessary equipment, such as computers, mobile devices, or prototypes, is available and functioning correctly.

4. **Test Scenario Creation**: Develop realistic test scenarios or tasks that reflect common user interactions with the product. These tasks should align with the objectives of the usability test and cover key features or functionalities.

5. **Observation and Data Collection**: During the testing session, observe participants as they attempt to complete the tasks. Take note of their actions, comments, and any difficulties they encounter. Use various methods for data collection, such as video recording, screen capture, or note-taking.

6. **Analysis and Iteration**: After completing the usability testing sessions, analyze the collected data to identify patterns, recurring issues, and areas of improvement. Prioritize the identified usability issues based on their severity and impact on the user experience. Use this feedback to iterate and refine the design of the product.

7. **Iterative Testing**: Usability testing is often conducted iteratively throughout the design and development process. After implementing design changes or improvements, repeat the testing process to validate the effectiveness of the changes and ensure that new issues do not arise.

8. **Reporting and Documentation**: Document the findings of the usability testing, including identified issues, user feedback, and recommendations for improvement. Share these findings with relevant stakeholders, such as designers, developers, and project managers, to inform decision-making and drive product enhancements.

By incorporating usability testing into the design process, teams can gain valuable insights into user behavior, preferences, and pain points, ultimately leading to the creation of more intuitive, user-friendly products and systems.

Once all these stages have been addressed from persona development to testing, the next step is typically to move forward with development and implementation. This involves translating the finalized designs and prototypes into actual working software or products. Throughout this process, close collaboration between designers, developers, and other stakeholders is crucial to ensure that the final product aligns with the intended design vision and meets user needs. Additionally, ongoing testing and iteration may continue during development to address any unforeseen issues or improvements. Ultimately, the goal is to deliver a high-quality product that provides an exceptional user experience and fulfills the objectives outlined during the design and prototyping phases.

Chapter 4

Design

"If you think good design is expensive,
you should look at the cost of bad design."
Dr Ralf Speth

As mentioned earlier, this chapter will cover all the necessary steps to design features that we would like to add to the native iOS Apple Calendar application.

4.1 User Research

Understanding the target users is fundamental to the success of product development. By gaining insights into their needs, preferences, and behaviors, designers can create user-centric solutions that resonate with users, remain relevant in the market, and mitigate risks associated with product development.

Since the goal of the thesis is to develop models for school timetables, our main target user is the student population. Understanding students' specific needs and preferences is critical to creating effective, easy-to-use planning models. By tailoring our designs to meet students' needs, we aim to provide them with tools that simplify their academic organization, improve productivity and promote overall well-being.

However, it's essential to delve deeper into the user demographics to ensure that the templates are truly effective and inclusive.

Here are some key considerations:

- **Age and Educational Level**: Students' needs vary significantly depending on their age and educational level. For example, elementary school students may require simpler and more visually engaging templates, while high school or university students might need more detailed and structured schedules. By considering the age and educational level of the target users, we can tailor

the templates to suit their cognitive abilities and organizational requirements effectively.

- **School/University Specifics**: Each educational institution may have its own unique scheduling requirements and formats. Therefore, it's crucial to adapt the templates according to the specific policies, timetable structures, and academic calendars of different schools or universities. Customization options should be available to accommodate variations in class durations, breaks, extracurricular activities, and other scheduling factors.

- **Geographical Considerations**: The organization of schedules can vary significantly based on the geographical region or cultural context in which the student resides. For example, scheduling norms and practices may differ between central Europe, South America, Asia, and other regions. Factors such as school start times, holiday schedules, and academic priorities may influence the design and functionality of the templates. By acknowledging these geographical differences, we can ensure that the templates are relevant and applicable to a diverse global audience.

- **Localization and Adaptation**: To address the diverse needs of students across different regions, the templates should be localized and adaptable. This involves offering customization options for language preferences, date formats, and regional holidays. Additionally, providing guidelines or tips for optimizing the templates based on local scheduling practices can enhance their usability and relevance for users worldwide.

After careful consideration of various factors, we have decided to focus on university students as the primary target users for our school schedule templates. Additionally, we have chosen to restrict the geographical area to Europe. Here's why:

University Student Focus: University students represent a diverse and significant user group with distinct scheduling needs and preferences. By targeting university students, we can address the specific challenges and requirements they face in managing their academic schedules, coursework, and extracurricular activities. This includes considerations such as class schedules, study sessions, project deadlines, and social engagements.

Rationale for Geographical Focus: While universities worldwide may have unique characteristics, focusing on a specific geographical area allows us to streamline the customization process and ensure greater relevance and usability for our target users. Europe has been selected as the geographical focus due to several compelling reasons:

1. **Similar Organization**: Universities across Europe exhibit a relatively consistent organizational structure in terms of school hours, academic activities,

examination schedules, and semester-based systems. This uniformity simplifies the design and customization process, as the templates can be tailored to align with common scheduling norms and practices prevalent across European universities.

2. **Cultural and Academic Alignment**: European universities share certain cultural and academic values and traditions, which influence the scheduling and organization of academic activities. By focusing on Europe, we can better understand and address these cultural nuances, ensuring that the templates resonate with European students and cater to their specific needs and expectations.

3. **Accessibility and Usability**: Concentrating on a specific geographical area allows us to delve deeper into the unique requirements and preferences of European university students. This targeted approach enables us to create templates that are more accessible, user-friendly, and relevant to the daily lives of our target users, ultimately enhancing their overall experience and satisfaction.

To ensure the accessibility and widespread adoption of this revamped application, we conducted research to understand the iPhone user base, since, as already mentioned, Apple Calendar is a native iOS application, therefore present only on Apple devices.

My research used data from Statcounter Global Stats, a reputable web traffic analytics website founded in 1999. Statcounter provides valuable insights into device usage trends across different regions and time periods.

we narrowed this analysis to the period from 2021 to 2023 to capture recent trends and changes in iPhone usage patterns. This time frame allows us to obtain relevant and up-to-date statistics for our research. Furthermore, my analysis focuses specifically on Europe to align with the target market for this application. By focusing on European iPhone users, we can adapt our strategies and features to meet the preferences and behaviors of this demographic.

The analysis of iPhone usage trends in Europe yielded noteworthy findings that inform the potential user base for this application.

Apple products, including iPhones, emerged as the most prevalent devices among European users, capturing a substantial share of 32.98% of the market. This indicates a significant opportunity to reach a large audience through iOS devices.

However, while iPhones are popular, it's important to note that the data doesn't differentiate between students and other demographics. Therefore, while the application may appeal to a broad user base, we need to consider the specific needs and preferences of students when designing and marketing the app.

Although the focus is on iOS devices for the initial development phase, the application can be adapted to accommodate other mobile vendors in the future.

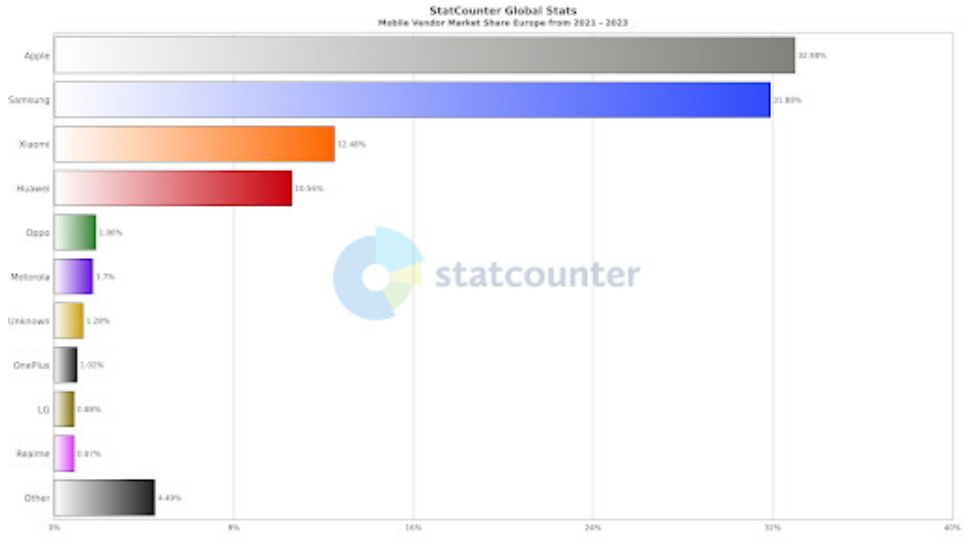

Figure 4.1: Mobile Vendor Market Share Europe from 2021 to 2023

However, to streamline the development process and maximize efficiency, I've chosen to prioritize iOS compatibility for now.

4.2 Persona Development

Once the user target has been defined, the next step is to develop the Persona.

This target represents a heterogeneous group with varying backgrounds, interests, and academic pursuits. Persona development allows designers to capture this diversity and create tailored user experiences that address the specific challenges and goals faced by students.

So, this section delves into the process of persona development specifically tailored to university students. By exploring the goals, needs, and challenges of our target audience, the aim is to create personas that serve as empathetic representations of the student experience. Through these personas, it can uncover valuable insights that will inform the design and development of solutions tailored to the needs of university students.

When crafting personas for university students, it's essential to consider factors such as:

- **Academic Background**: Students pursuing different majors or disciplines may have distinct requirements for academic tools and resources. For example, engineering students may prioritize software for technical calculations, while liberal arts students may require tools for research and writing.

- **Lifestyle and Schedule**: University students often juggle academic responsibilities with extracurricular activities, part-time jobs, and social commitments. Understanding their daily routines, time constraints, and preferences for managing their schedules is crucial for designing intuitive and efficient user interfaces.
- **Technological Proficiency**: Today's university students are digital natives who are comfortable using technology for various tasks. However, their familiarity with specific devices, platforms, and software tools may vary. Persona development helps identify students' technological preferences and proficiency levels to inform design decisions.
- **Communication Preferences**: University students rely heavily on digital communication channels for staying connected with peers, professors, and academic institutions. Understanding their preferred communication methods, whether it's email, messaging apps, or social media, is essential for designing effective communication features within applications.
- **Academic Goals and Challenges**: Students have diverse academic goals, ranging from achieving high grades to gaining practical skills and experiences. Persona development allows designers to empathize with students' aspirations and challenges, ensuring that the products they design support their educational journey effectively.

Taking all these aspects into account, we created this example of Persona:
Persona Name: Cloe Toms
Goals and Needs:

1. Efficiently manage academic schedule for both daily classes and long-term assignments.
2. Stay organized and prioritize coursework effectively.
3. Easily access and navigate class locations using Google Maps integration.
4. Stay informed about any changes or exceptions to the regular schedule, such as holidays or classroom changes.

Preferences:

1. Prefers using Apple products and services for their reliability, ease of use, and seamless integration across devices.
2. Values simplicity and user-friendly interfaces when it comes to organizing academic tasks and schedules.

3. Appreciates features that enhance productivity and streamline daily tasks, such as direct links to Google Maps for class locations.

Challenges:

1. Time Management: Balancing academic responsibilities with extracurricular activities and personal interests can be challenging. Cloe may struggle to allocate time effectively to ensure she stays on top of her coursework while also participating in campus activities.

2. Academic Pressure: As a university student studying Computer Science, Cloe may encounter challenging coursework and demanding assignments. Managing academic stress and meeting academic expectations could be a challenge for her.

3. Technology Overload: While proficient with Apple devices, Cloe may also face challenges related to technology overload. Constantly being connected to digital tools and devices could lead to distractions and difficulty focusing on academic tasks.

4. Social Pressure: Active involvement in campus activities may also come with social pressures to fit in or excel in certain social circles. Balancing social obligations with academic commitments could pose a challenge for Cloe.

5. Adaptation to Change: Cloe's busy schedule and involvement in various activities may make it challenging for her to adapt to changes or unexpected events, such as last-minute schedule changes or new academic requirements.

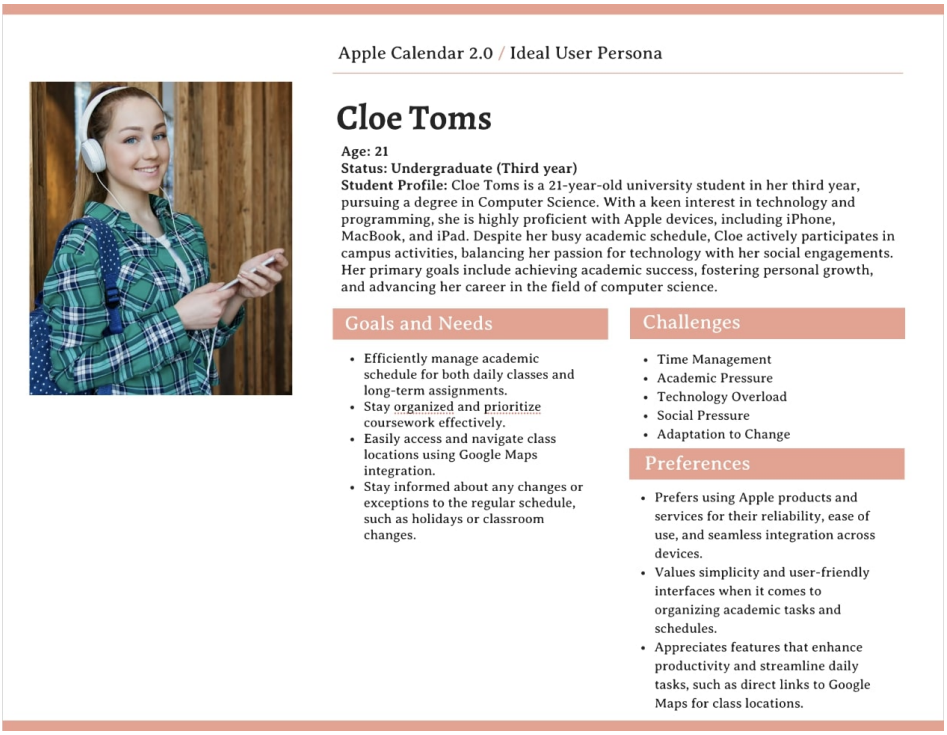

Figure 4.2: Ideal User Persona
[9]

Persona Profile:

- Age: 21
- Education: University student (third year)
- Field of Study: Computer Science
- Interests: Technology, Programming, Campus Activities.
- Technology Proficiency: Highly proficient with Apple devices, including iPhone, MacBook, and iPad.
- Lifestyle: Busy student with a demanding academic schedule and active involvement in campus activities.
- Goals: Academic success, personal growth, career development

With the creation of the Persona for this project, we now have a clear picture of our target user and their needs. It is now time to proceed to the next step.

4.3 Scenario and Storyboard

After the definition of the personas involved, wemove on to the design of the user scenarios. The definition of the user scenario involves the writing of a fictional story with the protagonists the personas.

Three scenarios are presented below that illustrate specific situations in which the proposed additions to Apple Calendar are extremely useful for university students. These scenarios aim to improve the academic schedule management experience by addressing common challenges such as organising classes, managing schedule exceptions, etc. The proposed solutions offer a practical and intuitive approach to optimising scheduling and ensuring that students remain informed and prepared for their academic activities.

4.3.1 First Scenario: How to consult the timetable with a click

Beca is a student and attends Czech Technical University.
Her school schedule is very varied and every day she has several classes in different places; sometimes it also happens that the class is moved to another location due to a change in the timetable, so it is difficult to remember and manage all these situations.
Today is Monday, it is 8 am and Beca has just left the dormitory to catch the bus to university.
Arriving at her destination, Beca knows her first class will be Theory of Algorithms, but she is not sure she remembers the classroom number. Oh wait, the mobile data on her phone doesn't work, Beca can't go to the university homepage to check the classroom. She could call her classmates but it's already 8:15 am, they must be in the classroom by now. How can she do that?
Fortunately, thanks to Apple Calendar 2.0, Beca can check her timetable directly from her phone, without any need for an Internet connection. Everything is within that application: course, start and end time, classroom, professor, etc.
Thanks to this, Beca can now control the classroom and in addition can also find out which topic will be covered on that day.

Figure 4.3: First Storyboard
[10]

4.3.2 Second Scenario: Odd or even week?

Tai is a student and attends law school at Charles University. His courses are organized into lectures and tutorials: the lectures have the same timetable in each week unless there are variations due to holidays or other reasons; the tutorials, on the other hand, consist of discussions on topics covered in the lectures, however, these topics are explained over a two-week period, therefore, the tutorials take place once every fortnight.
It's April, it's a Wednesday morning, the semester started a while ago, there was the Easter holidays in the meantime, and Tai has now lost count of the weeks, so he doesn't know whether or not there will be a tutorial today. How can he do that? Tai could log on to the university homepage and check, but his credentials have

expired so he has to go through the whole procedure again to get access. It takes too long.
He could ask on the university's WhatsApp group, but as usual no one answers.
Tai finally remembers that he has Apple Calendar 2.0 on his iPhone and with a simple click he can now check whether or not the exercise is on that day. The application, in fact, counts the weeks for you, if no engagement is present, then it means that the two weeks have not yet passed, vice versa, if the engagement is present, it is debating day! At this point, the mystery unraveled, Tai finally decides to set the application as a widget on his phone, so that he can have a general and quick overview of the upcoming engagements of the day/week.

Figure 4.4: Second Storyboard
[10]

4.3.3 Third Scenario: Reminder via notification

It is September and a new school year has begun. Chris is a student and attends the Faculty of Computer Engineering at CTU. It is Monday morning and it is the first Information Security class, the lesson begins and as usual the professor gives a general introduction to the course.
The professor also explains that there will be two tests during the semester, one during week six and one during week nine; he also adds that the possible exam dates will be during the first two weeks of June. Chris marks all this information on his iPhone in Apple Calendar 2.0 and the professor begins to explain the first topic of the course.
Weeks pass and the first test gets closer, but of course Chris has totally removed this information, but he absolutely has to start studying!
Who is responsible for reminding Chris of this test?
Once again, Apple Calendar 2.0 comes to the rescue and sends a reminder to Chris, he will then see the notification on the app and can start practicing for the test.
Chris himself will decide how soon to receive the reminder, how often and also how far apart.

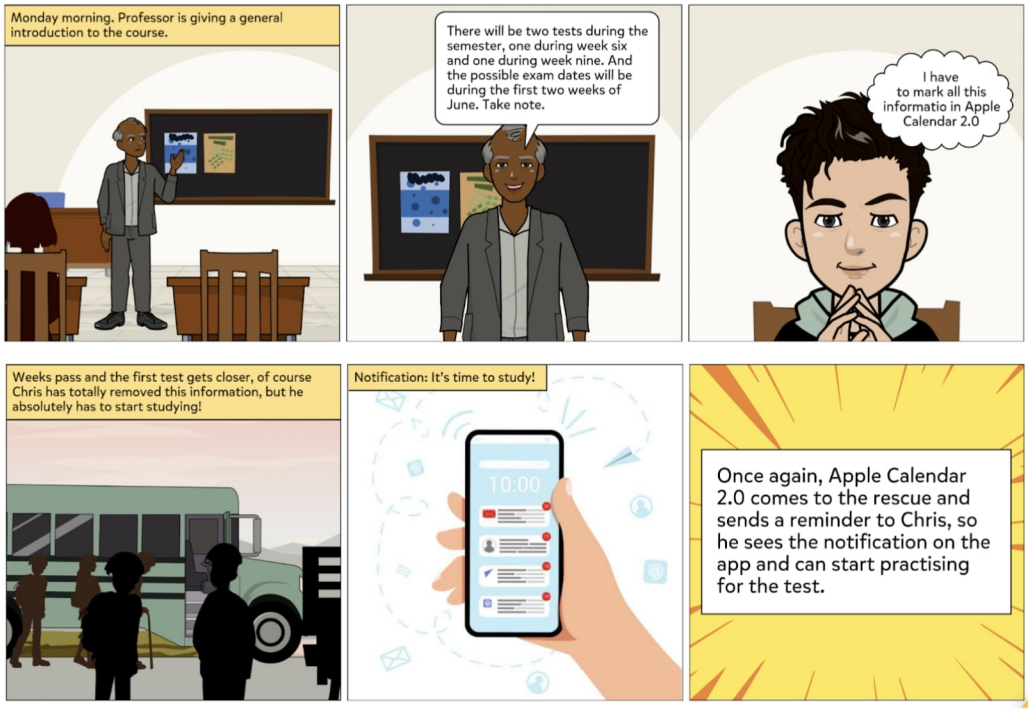

Figure 4.5: Third Storyboard
[10]

4.3.4 Hierarchy of scenarios

The three scenarios are organised in a hierarchy that reflects their relative importance and logical sequence within the application use process. Each scenario builds on the previous ones, thus contributing to a complete picture of the functionalities and situations covered by the application.

The first scenario serves as the foundation, focusing on the essential features that the application aims to provide. This includes generating a timetable that encompasses all classes, specifying details such as professors, classrooms, and other relevant information.

The second scenario addresses potential exceptions to the standard weekly timetable, particularly the alternation between odd and even weeks. It delves into how the application handles these differences, ensuring that users are adequately informed and prepared for any deviations from the norm.

Finally, the third scenario centers around the implementation of notifications within the application. It is crucial for the application to alert users about upcoming events, such as classes or deadlines. This scenario builds upon the foundation laid out in the first scenario, enhancing user experience by providing timely reminders without introducing new properties.

Overall, this hierarchical structure ensures a systematic approach to application development, with each scenario building upon the previous one to create a comprehensive and user-friendly solution.

4.4 Hierarchical Task Analysis

Having defined the Persona for this project and written down the scenarios describing the various situations in which the application will be used by university students, we now proceed to the next step: analysis by means of Hierarchical Task Analysis (HTA).

The following Hierarchical Task Analysis (HTA) describes three distinct scenarios in which the application is used to manage the academic schedules of university students. Each HTA provides a detailed overview of the activities involved in accomplishing a specific task, highlighting the user's actions and the steps required to achieve a goal. In these HTAs we will find recurring tasks, such as:

1. Main Task: Start by identifying the main task or goal that the scenario is analyzing.

2. Timetable View: This is a menu option for selecting different views of the timetable. It includes sub-options for viewing the timetable by Day, Week (with further options for 1-week and 2-week views), Month, Year, and as a List.

3. Add New: This option allows users to add a new class or event to their timetable.
4. Inbox: This represents a feature where users receive notifications or messages related to their classes or timetable.

Now let us go deeper with the specific scenarios.

4.4.1 First HTA

In the first scenario, Beca, a university student, needs to quickly find information about her upcoming class while on her way to the university. She faces the challenge of not having access to the internet to check the university homepage for classroom details.

To address this challenge, Beca uses the Apple Calendar 2.0 application on her phone. The application allows her to access her class timetable without requiring an internet connection. It provides comprehensive information about her classes, including the course name, start and end times, classroom number, professor's name, and the topic to be covered on that day.

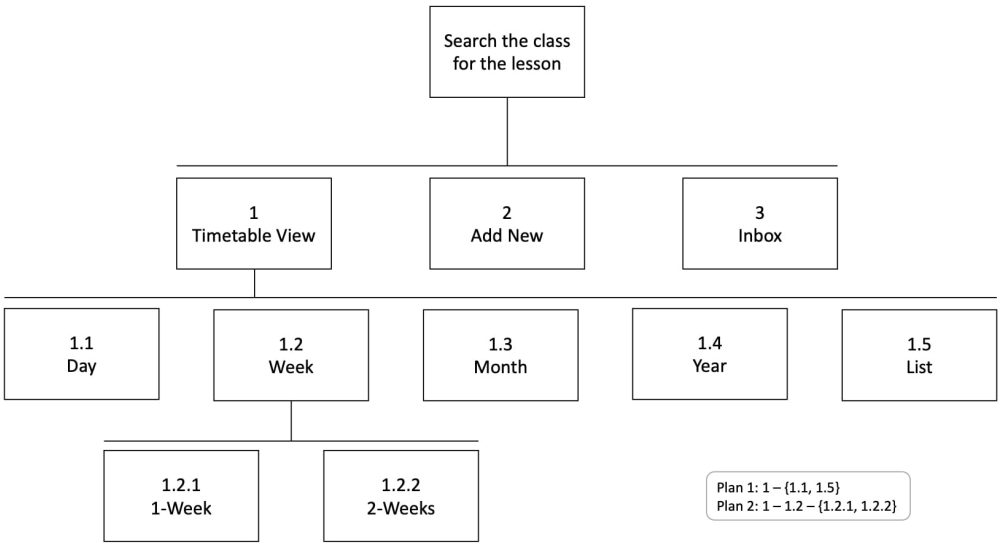

Figure 4.6: First HTA

Several paths can be followed to perform the main task:

Plan 1: This describes a sequence of actions where Beca navigates to view the timetable (1) and either selects the Day view (1.1) or accesses the List view (1.5).

Plan 2: This sequence involves navigating to the Week view (1.2) and then selecting either the 1-Week view (1.2.1) or the 2-Week view (1.2.2).

Different plans for the same objective.

4.4.2 Second HTA

In the second scenario, Tai, a law student at Charles University, faces the challenge of determining whether there is a tutorial scheduled for the day. Due to the irregular scheduling of tutorials, with sessions occurring once every fortnight, Tai has difficulty keeping track of the schedule, especially after the recent Easter holidays.

Once again, Tai uses the Apple Calendar 2.0 application on his iPhone. The application offers a feature that automatically counts the weeks and indicates whether a tutorial is scheduled for the day. Tai can quickly check this information with a simple click, saving him the hassle of logging into the university website or relying on unresponsive WhatsApp groups. Now, let's break down the steps of this scenario into a Hierarchical Task Analysis (HTA):

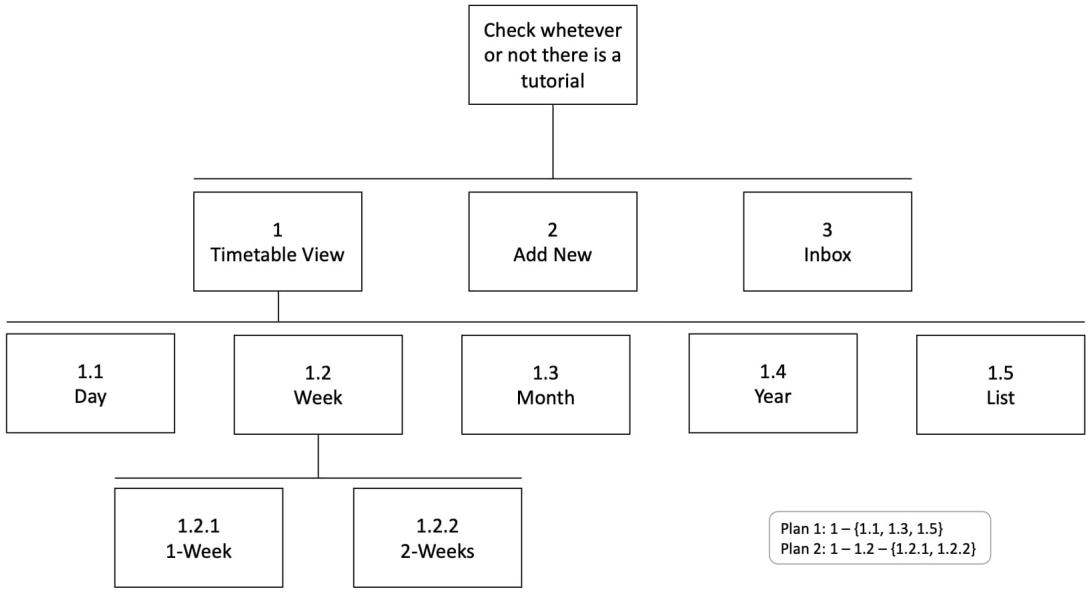

Figure 4.7: Second HTA

Once again, several paths can be followed to perform the main task:

Plan 1: Tai opens the Apple Calendar 2.0 application on his iPhone and navigates to the Timetable view (1) within the application to check for any scheduled tutorials.

Now, he can check whetever or not there is a tutorial by selecting the Day(1.1), Month(1.3) or List(1.5) view. If there is no engagement listed for the day, it means

that two weeks have not yet passed since the last tutorial. Instead, if there is an engagement listed for the day, it indicates that it is the day for the tutorial session.

Plan 2: Another way to obtain the same result is to go throw the Week view (1.2) and, then, go deeper by selecting 1-Week (1.2.1) or 2-Weeks (1.2.2). Tai can make the same points as in plan 1.

So, this HTA illustrates how Tai efficiently utilizes the Apple Calendar 2.0 application to manage his university schedule, particularly for irregularly scheduled tutorial sessions. By leveraging the application's features, Tai can quickly access relevant information and plan his day accordingly.

4.4.3 Third HTA

In this scenario, Chris, a Computer Engineering student at CTU, adds important test dates to his Apple Calendar 2.0 during his first Information Security class. As the first test nears, he forgets about it, but Apple Calendar 2.0 sends him a timely reminder, allowing him to prepare adequately. Chris adjusts the reminder settings to his preference and studies for the exam, ensuring he stays organized and on top of his academic responsibilities.

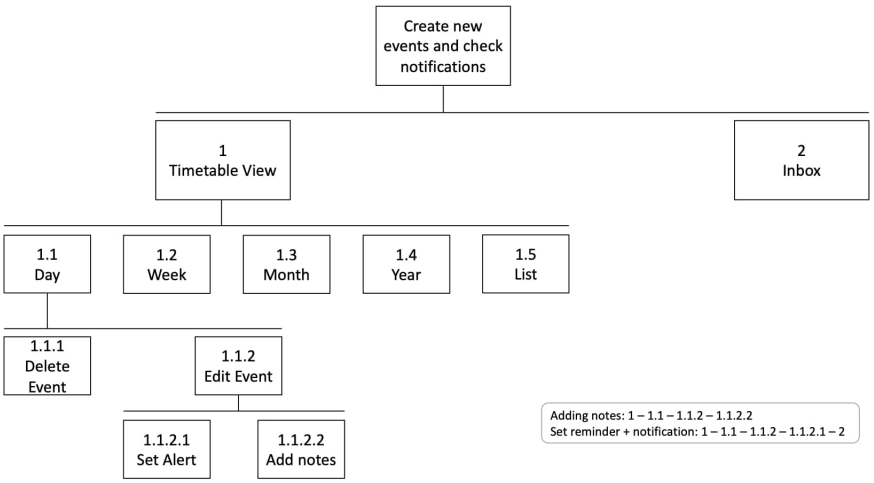

Figure 4.8: Third HTA

This HTA outlines the steps Chris takes to set a reminder for an upcoming test using the Apple Calendar 2.0 application.

Here we have two plans that Chris should follow to perform the main task.

He has to set reminder in order to have a notification before the test. He begins by accessing the Timetable View (1), specifically the Day view (1.1), where he can

create a new event for the test. He then navigates to the Edit Event (1.1.2) option to add any necessary notes or details about the test.

To ensure he doesn't forget about the test, Chris sets an alert by selecting the Set Alert option (1.1.2.1). He can customize the reminder settings to his preferences, including how soon to receive the reminder, how often, and how far apart.

Once the reminder is set, Chris can check his notifications in the Inbox (2) section of the application to confirm that the reminder has been scheduled successfully. This ensures that he will receive timely notifications to start preparing for the test.

If he wants to add new notes to the event, e.g. test topics, homework to be delivered, equipment to be brought, etc., Chris will have to follow the previous steps until Edit Event (1.1.2) and then complete the task by selecting Add notes (1.1.2.2).

This concludes the HTA discussion.

4.5 Sketches

With the user research, Persona development, scenario exploration, and Hierarchical Task Analysis (HTA) completed, we are now well-equipped to proceed with creating sketches for the redesigned iOS application. In fact sketching serves as a crucial step in the design process, allowing us to visualize and iterate on various interface elements and user interactions.

These sketches will serve as the foundation upon which we will build the prototype, enabling us to translate our conceptual ideas into tangible design solutions. By focusing on simplicity, clarity, and user-centricity, our sketches will aim to capture the essence of the application's new features and functionalities, ensuring that they resonate with the needs and preferences of our target users. Through this iterative design approach, we will refine and iterate on our sketches, continuously incorporating feedback and insights gained from each stage of the design process. Ultimately, our goal is to create a visually compelling and intuitive interface that enhances the overall user experience of the iOS application.

The sketches that were produced during this phase can be seen below.

It's worth noting that the sketches were not entirely hand-drawn, as this project involves a redesign where a foundation already exists. Therefore, we will build upon the existing framework to incorporate the desired features. The hand-drawn elements will represent the new additions and modifications to the existing design. This approach ensures continuity and efficiency in the redesign process, allowing us to focus on implementing the requested enhancements while leveraging the existing structure as a solid foundation.

4.5.1 How to insert the schedule

The first sketch depicts the initial interface of the Apple Calendar application on an iPhone. The main screen features a prominent "plus" button, which not only allows users to add a new event but also enables them to create a new schedule. The default view displays a weekly schedule, showcasing the various classes and events for each day. Users can also access a more detailed view of a specific class by tapping on it, revealing additional information such as the teacher's name, room number, and a clickable link to Google Maps for the location. Each class is accompanied by a description field where users can add relevant notes or details.

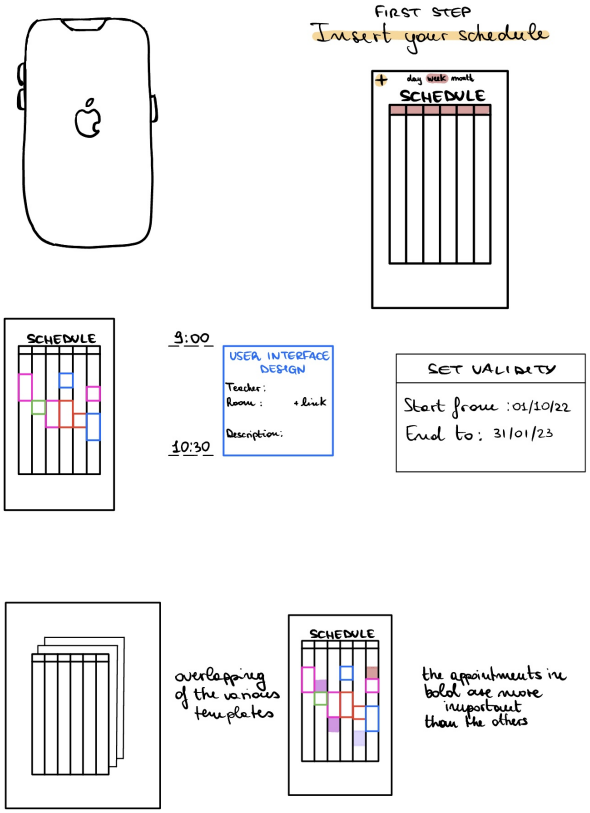

Figure 4.9: Insert your schedule

Furthermore, the sketch includes a visual representation of the schedule's validity period, spanning from October 2022 to January 2023. Additionally, there is a section dedicated to displaying all the created templates, highlighting the most important appointments. It's worth noting that while this last feature was initially conceived, it was not implemented in subsequent design iterations. Instead, it is earmarked for future development efforts.

4.5.2 How to set the schedule

Here's a sketch of how the scheduling application would work:

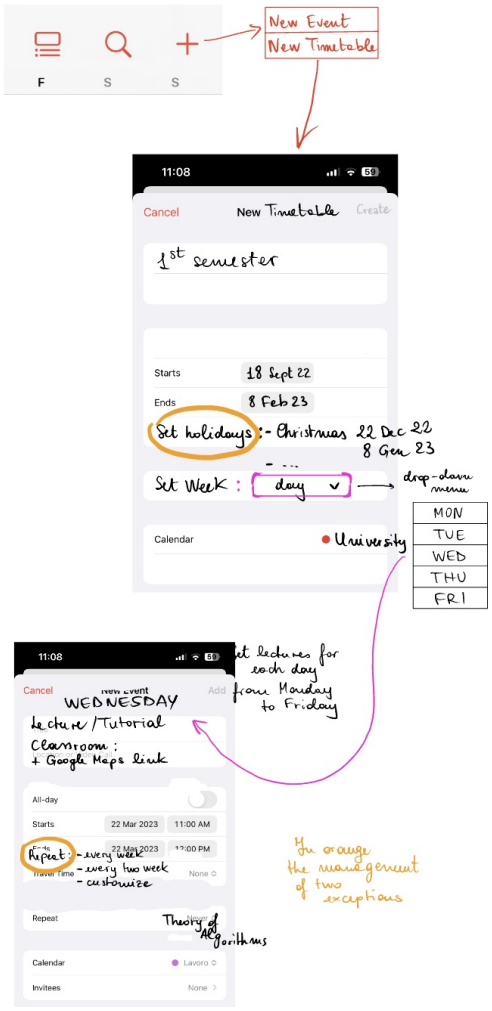

Figure 4.10: Set your schedule

Setting up the Semester Schedule:

- The user sets the start and end dates of the semester.
- They mark holidays or breaks when there will be no classes.

Adding Daily Lessons:

- The user selects a day to add lessons or tutorials.
- Upon selection, a new screen opens up.
- For each day, the user inputs:
  1. Lesson or tutorial title.
  2. Class location with a link to maps.
  3. Time of the lesson.
  4. Frequency of repetition (e.g., weekly, bi-weekly).
  5. Color association for the lesson.

Handling Exceptions:

- The scheduling system considers exceptions such as holidays when there are no classes.
- Users can specify if a lesson doesn't occur every week, allowing for flexibility in scheduling.

Visual Representation:

- The schedule will be visually represented as a calendar, with different colors indicating different types of lessons.

- Holidays or breaks without classes will be clearly marked on the calendar.
- Users can easily navigate through weeks or months to view the schedule.

This scheduling system provides a user-friendly interface for managing semester schedules, accommodating exceptions like holidays or irregular lesson frequencies. It ensures efficient organization and visualization of daily lessons throughout the semester.

4.5.3 Different Views

Once the timetable, with all the various lessons, is set up, users can proceed to view the lessons. The viewing options include daily, weekly, or monthly displays. This flexibility allows users to choose the level of detail they need based on their preferences and scheduling requirements. Whether they need a detailed breakdown of their daily agenda, an overview of the week's commitments, or a broader perspective for the entire month, the application accommodates their viewing preferences accordingly. This feature enhances user convenience and enables efficient planning and organization of their academic schedules.

Below, we will focus solely on the first two options, while the monthly view will be addressed in the mock-ups.

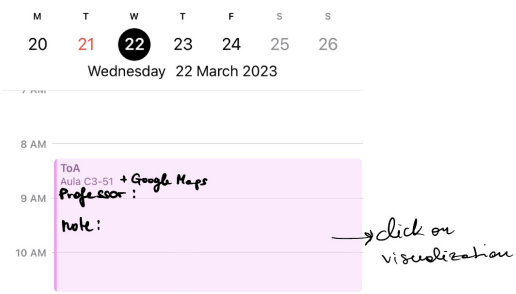

Figure 4.11: Day View

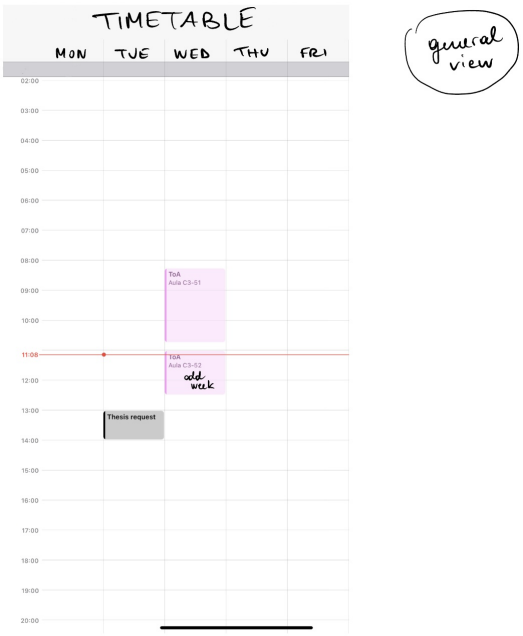

Figure 4.12: Week View

When a user clicks on a specific lesson, they will be directed to a detailed view containing comprehensive information about that lesson. This detailed view includes the lesson or tutorial title, class location with a clickable link to maps for easy navigation, the time of the lesson, its frequency of repetition, and the associated color for visual identification. By offering these details, users gain a deeper understanding of each lesson's specifics, enhancing their scheduling experience and ensuring they have all the necessary information at their fingertips.

The weekly schedule view provides users with a streamlined display of their schedule, focusing solely on the weekdays from Monday to Friday.

Designed particularly for mobile devices, this view optimizes screen space and ensures better visibility of the schedule's details. Each day of the week showcases the lessons scheduled for that specific day, allowing users to navigate through the week effortlessly. By excluding weekends from the display, users can efficiently manage their weekday schedules, ensuring a clear and concise overview of their weekly commitments on smaller screens.

4.5.4 2-Weeks View

An interesting addition to this application is the ability to view two weeks simultaneously. This feature addresses the common scenario where certain lessons or tutorials occur on alternate weeks, allowing students to absorb the lecture material one week and then apply it the following week. With the dual visualization of two successive weeks, users can easily verify this pattern. Furthermore, the design incorporates a synchronized scrolling mechanism, where scrolling down one week automatically scrolls down the corresponding week, enhancing user experience and facilitating seamless navigation. This feature provides users with a comprehensive overview of their schedule, enabling them to plan and manage their academic commitments effectively.

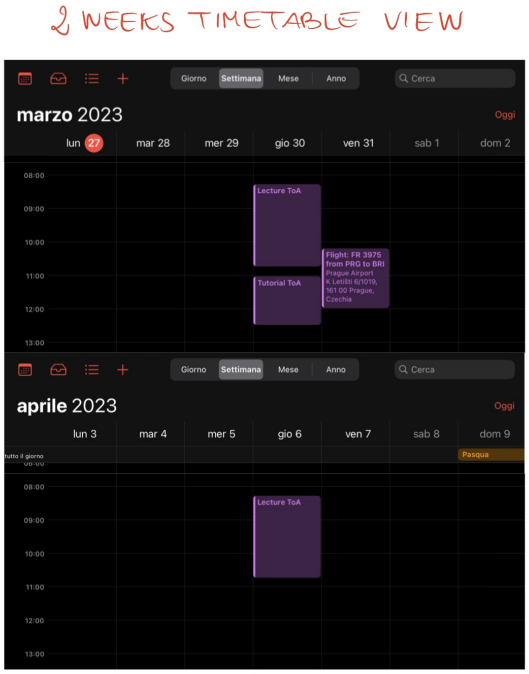

Figure 4.13: 2-Weeks View

4.5.5 Marks

In considering additional features to integrate into the Apple Calendar application, one intriguing idea that emerged was the possibility of incorporating a section to track academic grades obtained throughout the semester. This envisioned feature aimed to provide users with a convenient platform for monitoring their performance, identifying potential exemptions or exam eligibility criteria, and even facilitating the calculation of weighted averages.

While the concept holds potential, it may arguably offer limited value beyond existing functionalities provided by most university-specific applications. As a member of the target demographic, it is conceivable that I, along with others, might not find substantial utility in such a feature, particularly given that many university portals already automate the storage of grades and facilitate the calculation of averages.

However, despite this consideration, a potential implementation of this feature is presented below merely as a conceptual exercise, with the understanding that it will not be actively integrated into the current iteration of the application. This demonstration serves primarily as a reference point for potential future endeavors, acknowledging that its practical implementation is not deemed necessary at present.

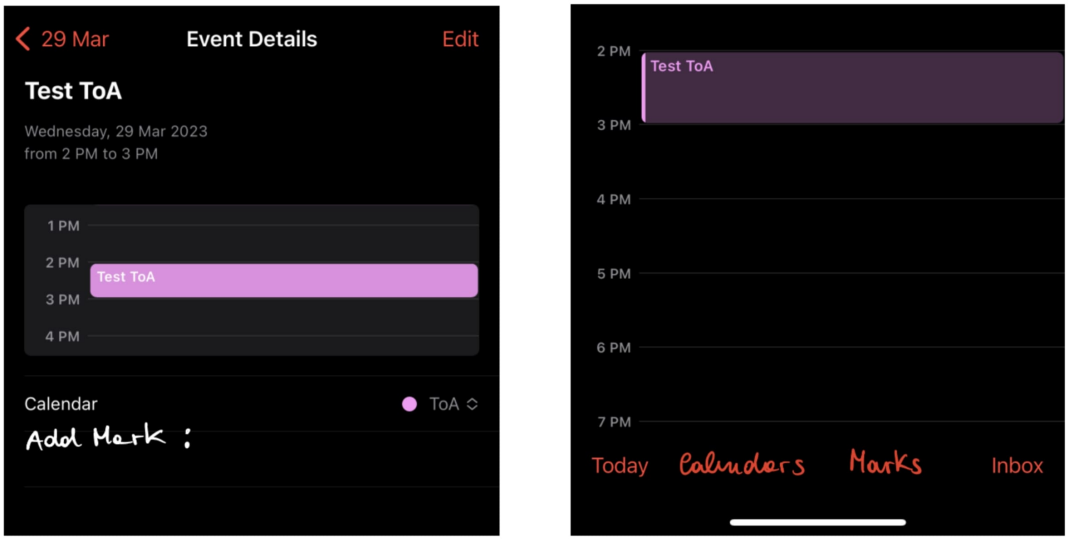

Figure 4.14: Adding marks

As evident, the idea involves introducing an "Add Mark" field corresponding to the exam date, where the user can input their grades. Behind the scenes, an algorithm would discern whether it's a final exam or a partial exam and proceed with the necessary computations to update the user's weighted average. At any

time, the user can navigate to the "Marks" section to review their average and the list of completed exams.

However, it's essential to acknowledge that such an algorithm would be exceedingly complex, considering the diverse grading systems and exam structures across European universities. Even narrowing the scope to a single country would present significant challenges due to the variations among university assessment frameworks. Consequently, despite the intriguing nature of the concept, it was decided not to pursue this avenue.

4.6 Mock-ups

Let's move on to a more detailed discussion.

The mock-ups presented below build upon the sketches, materializing the concepts envisioned during the sketching phase but in a more detailed manner. It's as if the sketches were the initial ideas conceived at the outset, while the mock-ups represent what can actually be achieved with a more precise design.

4.6.1 Add a new timetable

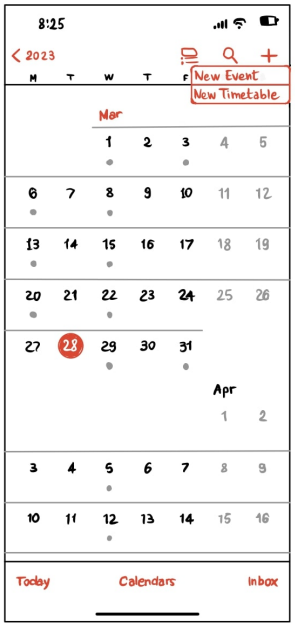

Figure 4.15: Add a new Timetable

When starting the application, the user is greeted by the home screen. Here, they are presented with two primary options upon clicking the plus icon:

1. **New Event**: This option allows the user to create a new event, similar to the functionality available in the base version of the application. Users can input details such as event name, date, time, location, and any additional notes.

2. **New Timetable**: Selecting this option initiates the creation of a new timetable, which serves as the central objective of this thesis. Users can include the duration of validity, any specific customization options and so much more.

Upon selecting “New Timetable”, the user is directed to a new window where they can begin setting up their timetable. Here, the user is prompted to provide a name for the timetable, such as “First Semester.” Once the name is entered, the user proceeds to specify the start and end dates of the semester, defining the duration of the timetable.

In addition to setting the semester dates, the user has the option to add school holidays to the timetable. Initially, the interface presents a button labeled “Set Holidays”. Upon clicking this button, the user can input details for each holiday, such as the name and date. After adding at least one holiday, the button transforms into “Update Holidays”, allowing the user to modify or add additional holidays as needed.

Following the setup of semester dates and holidays, the user moves on to adding their courses for the duration of the semester.

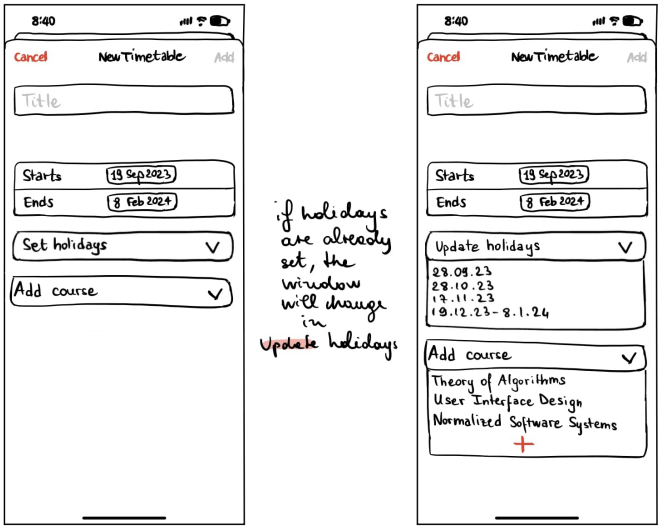

Figure 4.16: Set the Timetable

Upon selecting “Add Course”, a new window appears, allowing the user to input details for the course. Here, the user can enter the course name, total credits associated with the course, and the name of the professor teaching the course. Additionally, the user can choose a color for the course by clicking on the “Calendar” option, maintaining consistency with the original application’s labeling.

Once the basic course details are provided, the user proceeds to specify the type of event being added, with options such as “Lecture”, “Practice”, or “Lab”. For each event type, the user must input the start and end times, specify the frequency of the event and configure any desired alerts. The user has the flexibility to set multiple alerts if they wish to receive notifications before the event start time.

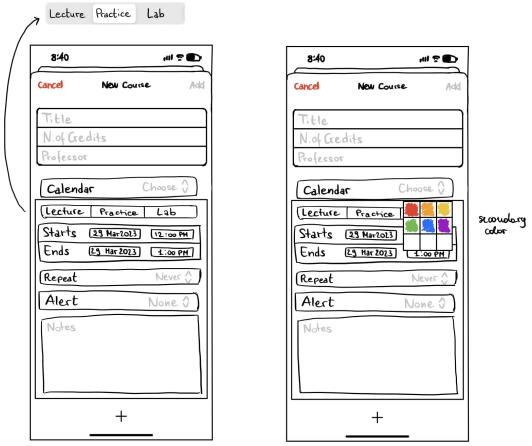

Figure 4.17: Add a new Course

Figure 4.18: Set Alert and Repeat

4.6.2 Views

Day View

In the presented mock-up illustrating the daily view, three classes related to two courses are scheduled on the same day. It is immediately apparent that the essential details (course name, type of class - lecture, exercise, or laboratory, room, and professor) are prominently displayed. The differentiation of class types will be further elucidated in the weekly view.
Another notable aspect depicted in this mock-up is the delineation of highlighted classes by marking the border, in contrast to Apple's method of entirely coloring the background of the event.

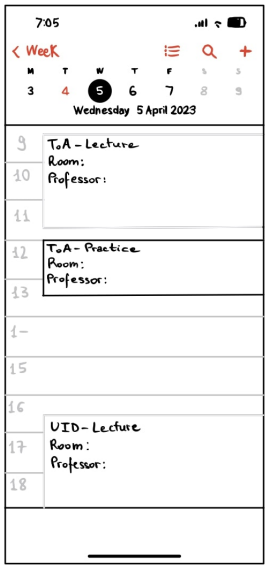

Figure 4.19: Day View

This departure from conventional design choices addresses potential usability issues, particularly in scenarios where only two events are scheduled. In such cases, if one event has a darker color and the other a lighter shade, discerning the selected event at a glance becomes challenging. By implementing border markings, users can readily identify the selected event without ambiguity.

Week View

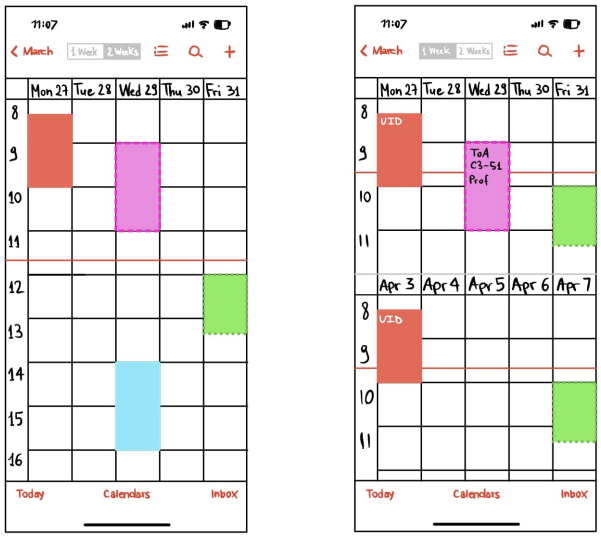

Figure 4.20: Week View

In the presented mock-up on the left showcasing the weekly view, a notable departure from the native application is immediately apparent: the ability to visualize the week as a vertical screen without the need for device rotation. This enhancement was implemented by considering only the days from Monday to Friday, as previously explained, to optimize space and facilitate viewing.
An additional significant detail is the presence of a button at the top (1 Week, 2-Weeks),

enabling users to choose between viewing one or two weeks simultaneously. This feature allows for convenient comparison between the current and subsequent weeks, facilitating quick identification of any differences or scheduling conflicts.

Furthermore, an initial implementation of color usage is observed in this mock-up, but refer to a more detailed discussion in the following paragraph.

Month View

We introduce another mock-up showcasing the monthly view, which distinguishes itself from the original version by presenting significantly more information.

In the basic version, the calendar merely indicates that the user has at least one commitment on a particular day of the month. In contrast, the new version allows the user to see how many courses they have scheduled for that day, along with an abbreviation of those courses (the abbreviation is formed by the initials of the words composing the course name). However, similar to previous designs, only the days from Monday to Friday are displayed, for the same reasons. During the implementation process, additional features were incorporated,

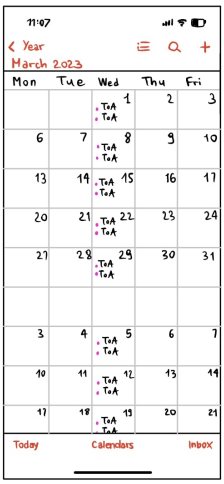

Figure 4.21: Month View

including the inclusion of event start times and experimentation to ascertain the maximum number of courses that could be feasibly added and displayed. It was determined that the system could effectively accommodate up to four courses.

Year View

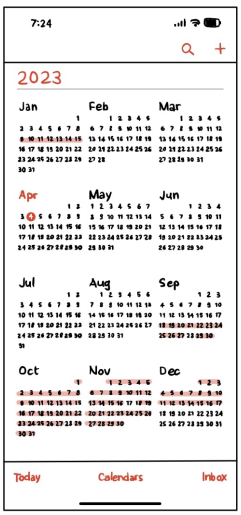

Figure 4.22: Year View

In the last mock-up about annual view we can see another visual information, in fact when creating the timetable, users designate a distinctive colour to represent their schedule. In the annual view, all days associated with that semester will be highlighted with the chosen colour, giving users an immediate visual reference for their timetable commitments, but also and especially for holidays.
However, it should be emphasised that the decision to assign a color to each semester was made at a later stage of development. As a result, the earlier mock-ups depicting timetable creation do not include the option to select a color.

This feature will be introduced during the implementation phase.

4.6.3 Colors

An essential aspect addressed in the development of these mock-ups is the selection of colors assigned to various courses. Color plays a pivotal role in user interface design, serving as a visual cue to convey information, establish hierarchy, and enhance usability.

In the context of a timetable application, distinct colors for different courses aid users in quickly identifying and differentiating between various academic engagements. This deliberate use of color not only facilitates organization but also improves user comprehension and navigation within the interface.

When designing text for colored backgrounds, it's essential to ensure readability and contrast to maintain a positive user experience. Typically, black text is recommended for use on light backgrounds, while white text is preferred on dark backgrounds. This ensures optimal contrast and legibility for users.

However, when dealing with colored backgrounds or typography, the rules for text opacity and contrast may change. Instead of using grey text or icons on top of colored backgrounds, it's advisable to create better contrast by displaying white or black text with reduced opacity. For instance, black text displayed at 75% opacity on a green background can provide the text with a black appearance, subtly infused with the color of the background. Another approach is to calculate the color of the text dynamically based on the background color. This involves placing a reduced-opacity black text over the colored background, identifying the resulting darkened color, and using that color's hex value for the text. By dynamically adjusting the text color based on the background, you ensure that the text remains legible and contrasts well, regardless of changes in the background color. It's crucial to note that if the surface behind the text changes color, the hex color of the text should also be updated accordingly to maintain optimal contrast and readability.

Therefore, the judicious use of color can have a significant impact on the user experience of an app or game. Furthermore, in the context of app interface design, it is crucial to consider other factors such as inclusivity for users with visual impairments or allowing the use of custom colors to express the unique personality of the user who decides them, but also test color combinations in various lighting conditions and on different devices to ensure optimal visibility and readability.

All these principles guided the strategic implementation of colour choices in the Apple Calendar design, ensuring a visually intuitive and user-friendly experience for users.

Text Color

Thanks to these principles, we adhered to using black text on light backgrounds and white text on dark backgrounds. However, initially, the contrast appeared too stark, prompting the decision to reduce the text opacity. After several iterations, it was determined that 80% opacity provided the optimal balance, ensuring readability while integrating seamlessly with the background.

Background Color

Regarding the choice of background colors, we initially opted to utilize the secondary colors in addition to the primary ones: red, yellow, orange, green, blue, and then we decided to add brown, to maintain consistency with the existing application. To ensure visual coherence and readability, we adjusted the saturation of these colors to 70%. Although Apple offers extensive color customization options, we decided to limit the user to these seven options, considering the practicality that most users are unlikely to be enrolled in more than seven courses in a single semester. While it's essential to accommodate user preferences, particularly regarding color selection, for this initial version, we restricted the choices to these options. However, this aspect could be expanded upon in future iterations of the application.

Texture

Another aspect to consider is that each course may include lectures, practices, and labs. Therefore, we decided to visually differentiate these components by maintaining the same background color but adding specific details to distinguish them.

We assumed three options:

1. change the brightness of the colour, so that we start with a darker tone for lessons and gradually lower it.
2. change the border line, for lessons continuous line, for exercises dashed line, for workshops line with dots.
3. keep the same colour for all lessons, lessons will have no texture, drills will be overlaid with a dense grid and labs also with a grid but rotated.

However, visual limitations were encountered, as not everything rendered well on the small screen of the phone. Therefore, modifications were implemented, which we will discuss in the following chapter.

This marks the final phase of design, and in the next chapter, we will proceed with prototyping.

Chapter 5

Implementation

"The next big thing is the one
that makes the last big thing usable."
Blake Ross

5.1 Prototyping

Prototyping stands as a pivotal phase in the journey of transforming abstract concepts into tangible digital solutions. It serves as the gateway between ideation and execution, allowing designers and developers to test, iterate, and refine their ideas before committing to full-scale development. At its core, prototyping is the process of creating preliminary versions of products or services to evaluate their feasibility, functionality, and user experience.

In the realm of digital design, prototyping encompasses a spectrum of methodologies and tools, each tailored to address specific project requirements and objectives. In the context of this thesis, we have chosen to utilize Figma and Xcode as our primary prototyping tools.

Figma is a collaborative web application for interface design that also offers additional offline features through desktop applications for both macOS and Windows. It provides a comprehensive set of tools tailored for user interface and user experience design, emphasizing real-time collaboration among team members. Figma allows users to create and edit vector graphics, design prototypes, and share and comment on designs seamlessly. Additionally, Figma's mobile app for Android and iOS enables users to view and interact with prototypes in real-time on mobile and tablet devices, enhancing accessibility and usability across different platforms.

Instead, Xcode is Apple's integrated development environment (IDE) for macOS, commonly used by developers to create software for Apple platforms such as iOS, macOS, watchOS, and tvOS. It provides a comprehensive suite of tools and

resources for building, debugging, and testing applications across various Apple devices.

We opted to begin with Figma because it offers a more intuitive approach and doesn't require coding. This allowed us to create purely visual prototypes that could be easily tested by users. By utilizing Figma, we were able to identify and refine design aspects before transitioning to the development phase with Xcode. This iterative process enabled us to gather valuable feedback and insights, ensuring a more streamlined and user-friendly final product.

However, the prototyping landscape extends beyond these flagship tools, offering a diverse array of alternatives to suit varying project needs. From Adobe XD's all-in-one UX/UI solution to Sketch's professional digital design platform, designers have a plethora of options at their disposal. Additionally, platforms like Microsoft Visual Studio and Android Studio cater to developers seeking robust IDEs for building cross-platform applications.

5.2 Figma

As mentioned earlier, we utilized Figma for an initial high-fidelity prototyping approach. Here is the link to the project: **Apple Calendar 2.0**.

We now embark on a comprehensive exploration of the various windows within the application, meticulously crafted through the utilization of the aforementioned software, akin to the methodology employed during the mock-up creation phase. These windows serve as the primary interface through which users engage with the application's functionalities, offering a cohesive and intuitive experience. Each window has been meticulously designed to cater to specific user requirements, aiming to optimize usability, enhance clarity, and streamline scheduling and task management processes.

5.2.1 Add a new timetable

Upon opening the application, the user is greeted with the initial screen displaying the monthly calendar view of the application on an iPhone 14. The current date is highlighted in red (6th September).

The top navigation bar shows the year 2023 and the month of September. Here, the user can click on the plus icon (+) located in the top right corner, which presents two options, as previously described: adding a new event or creating a new timetable. In this exploration, we will focus on the second option, as it is the primary subject of this thesis.

By selecting "New Timetable", the user is taken to a new window dedicated to timetable creation. This window allows the user to specify various characteristics of the timetable, including the title, color, duration, holidays, and courses.

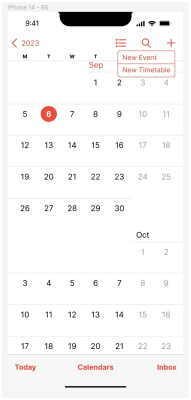

Figure 5.1: Add a new Timetable

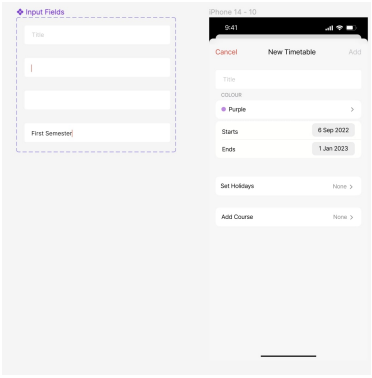

Figure 5.2: Set the Timetable

When the user clicks on the title input field, the cursor begins to blink, indicating that text can be entered. However, in this prototype, the title setting is automated. A second click on the title input field automatically sets the title to "First Semester". This feature streamlines the process for the user, ensuring a consistent and quick setup of the timetable's essential details. This simulation is realized by the use of the input field behavior in Figma that mimics real user interactions, providing a visual cue with a blinking cursor to signify that the field is active and ready for input, even though the actual text input is handled programmatically in this scenario.

Regarding the color settings and the duration of the timetable, we have remained consistent with Apple's design choices, merely replicating them.

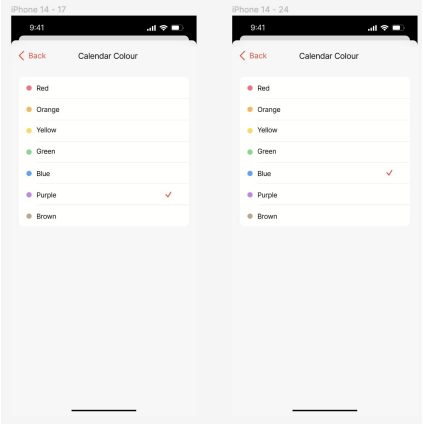

Figure 5.3: Colour

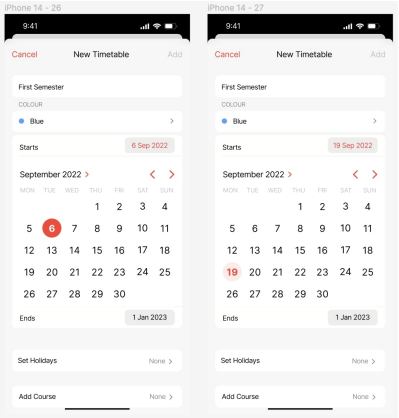

Figure 5.4: Duration

The most interesting and challenging aspect, instead, is undoubtedly the addition of holidays and various courses. Starting with holidays: initially, the "Set Holidays" field is set to "None." When the user clicks on this option, they are directed to another window where a "+" button is present. By clicking this button, the user can add holidays. Here, we have both a start date and an end date. If the holiday corresponds to a single day, the start and end dates will coincide. Conversely, if it corresponds to a longer period (e.g., Christmas holidays), the dates will be different.

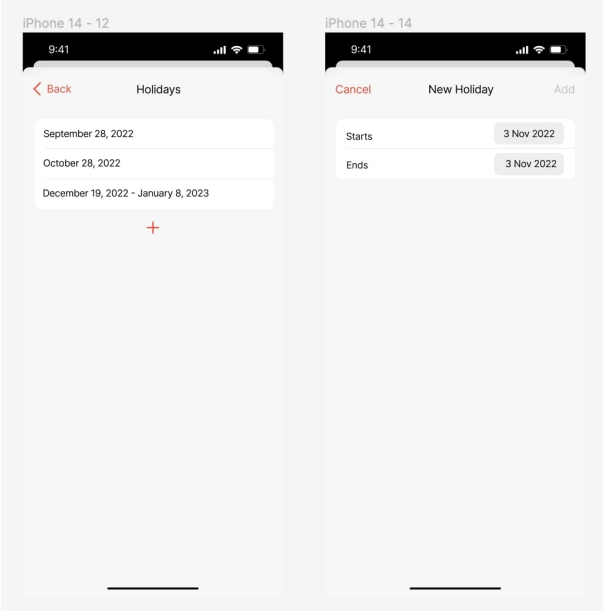

Figure 5.5: Month View

Moreover, in the previously presented mock-ups, we considered using a dropdown menu that would display all holidays in the initial screen. However, in this phase of high-fidelity prototyping, we decided to update the "None" field with the count of holidays. Only after clicking and navigating to another window does the user obtain the list of holidays with the option to add more.

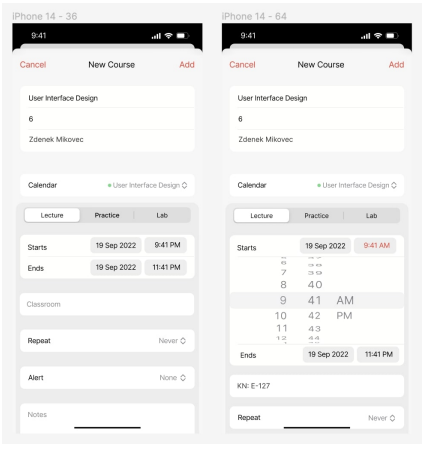

Figure 5.6: New Course

Now, addressing the addition of courses: the creation of a course closely mirrors the creation of a standard event in the Apple Calendar, with necessary modifications. The user needs to input the course name, number of credits, and professor. Following this, a box appears where the user can select the type of lesson (lecture, practice, or lab) they wish to add, set the time and day, and add the room where the lesson will take place.

Subsequently, the user can set how often the lesson should repeat, typically every week or every two weeks, but there is also a custom option for special cases. Additionally, an alert can be set to notify the user a certain amount of time before the lesson begins.

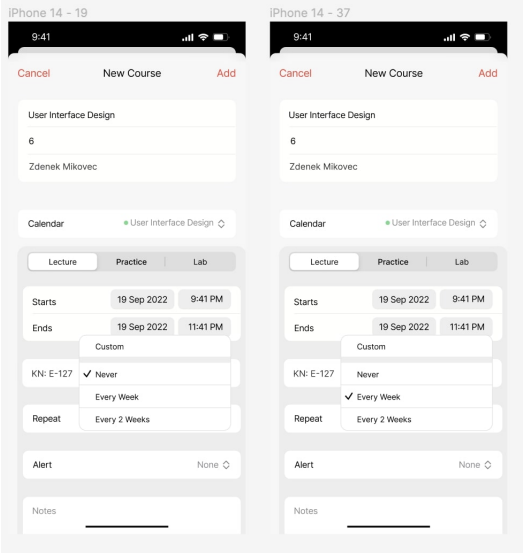

Figure 5.7: Set Repeat and Alert

5.2.2 Views

The screens presented below illustrate the various views within the application: day, week, month, and year views. All design choices made during the mock-up phase have been meticulously implemented in these screens.

Day View

In the day view, the highlighted lesson is distinctly marked with a darker border, ensuring it stands out for better visibility.

Week View

The week view is structured to display only five days, corresponding to the days when university classes are held. This view includes an option to switch between a one-week and a two-week view, facilitated by a button at the top of the screen, allowing users to toggle between these views for reasons elaborated upon in the previous design phase.

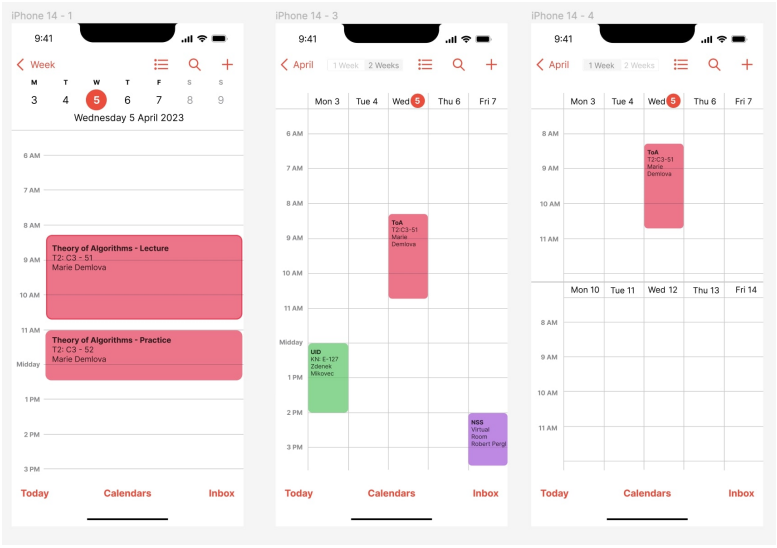

Figure 5.8: Day and Week Views

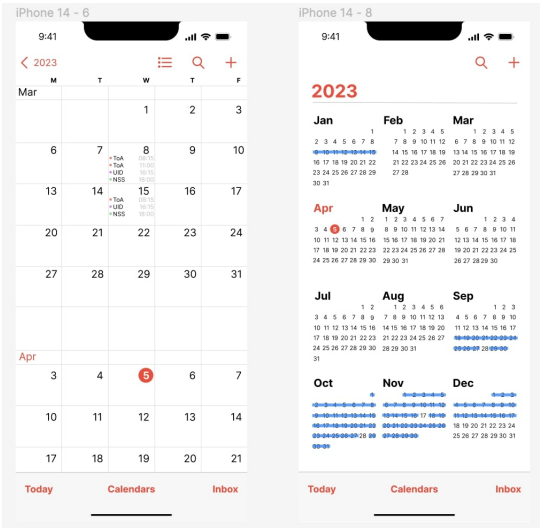

Figure 5.9: Month and Year Views

Month View

The month view remains perfectly aligned with the decisions made during the mock-up phase. Due to the inherent space constraints within the month view, it has been observed that a maximum of four lessons can be displayed per day. These lessons are differentiated by a colored dot on the left, representing the respective course, followed by an acronym of the course name. Additionally, the start time of

each lesson is indicated on the far right of the box, providing a clear and concise schedule for the user.

Year View

For the year view, the design highlights all the days associated with a particular semester using the color selected during the creation of the new timetable. This visual differentiation is intended to provide users with a clear and immediate understanding of their semester schedule, adhering to the rationale previously discussed in the design phase.

Overall, these views reflect the detailed and user-centric design choices aimed at enhancing the usability and functionality of the application.

5.2.3 Colors

Text and Background Color

In our decision-making process regarding color selection, we conscientiously followed the guidelines established in the earlier stages of development, particularly those delineated during the creation of mock-ups. However, recognizing the paramount importance of maintaining continuity and user familiarity within the Apple Calendar interface, we deemed it imperative to conduct a thorough evaluation of the actual colors utilized within the application.

Our overarching objective remains rooted in the seamless integration of new features while preserving the core user experience. Hence, our initial inclination was to ascertain how any alterations to color palettes might impact user interaction and comprehension. This deliberation prompted us to juxtapose the proposed colors with those native to the Apple Calendar interface, thereby ensuring that any introduced changes would not disrupt the established visual hierarchy or impede user navigation.

Concurrently, we explored the viability of employing more saturated colors in conjunction with black text, a departure from the initial approach of utilizing slightly darker shades for text legibility. This experimental divergence was motivated by a desire to enhance readability and ensure that pertinent information stood out prominently against the background. Following meticulous comparison and analysis of both options, our team unanimously concluded that the utilization of saturated colors with black text offered superior legibility and visual clarity.

While acknowledging the departure from the original color scheme, we collectively agreed that the enhanced readability and visual appeal of the revised approach warranted its adoption. Thus, despite the variance from the initial plan, we confidently proceeded with the decision to implement this alternative color scheme,

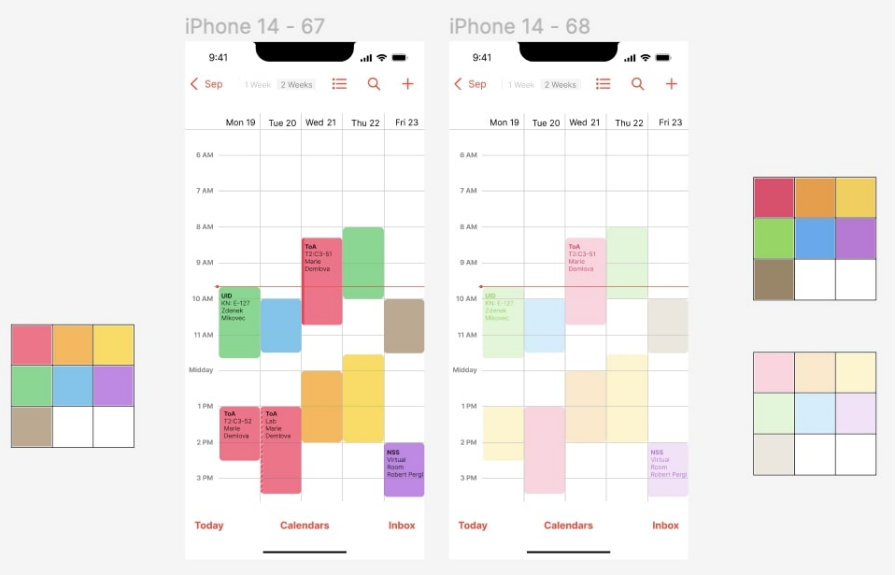

Figure 5.10: Color Palette

confident that it would ultimately contribute to an enriched user experience within the Apple Calendar interface.

Texture

In the preceding chapter, we explored also various approaches to visually differentiate between different components within each course, such as lectures, practices, and labs. These methods included altering the brightness of colors, changing border lines, and applying distinct textures. However, during the implementation phase, we encountered challenges with rendering these options effectively on smaller screens, particularly mobile devices.

Upon testing, it became apparent that the first and third options, which involved modifying color brightness and applying textures, respectively, did not adequately enhance readability on smaller screens. Consequently, we opted to proceed with the second option, which entailed utilizing different border lines for lectures, practices, and labs. This approach proved satisfactory in maintaining visual distinction between course components while ensuring optimal legibility across various screen sizes.

Subsequently, we explored an additional approach, introducing a fourth option, wherein lecture, practice, and lab sessions were differentiated using emojis within the event box in the top right corner. Specifically, a book emoji represented lectures, a pencil emoji denoted practices, and a test tube emoji symbolized labs.

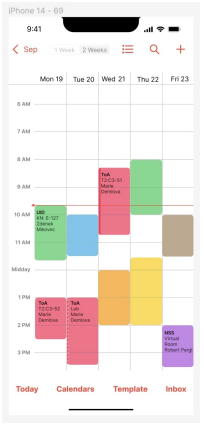

Figure 5.11: Texture: Second option, first variant

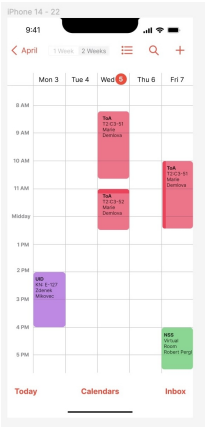

Figure 5.12: Texture: Second option, second variant

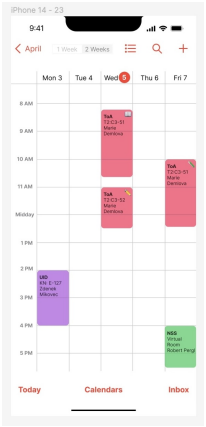

Figure 5.13: Texture: Fourth option

As we transition to the testing phase, the subsequent stage of development, we eagerly anticipate gathering user feedback to ascertain their preferences and further refine our interface design accordingly. This iterative process underscores our commitment to optimizing user experience and ensuring the seamless integration of visual elements within our course scheduling platform.

In the forthcoming testing phase, we shall endeavor to ascertain the relative favorability of the two implemented approaches through feedback solicited from users who have undergone testing procedures. This empirical assessment aims to elucidate which of the implementations has garnered a more positive reception among our user base.

5.2.4 How to make changes to the timetable

Having designed the application interface with the timetable creation functionality, it is natural to question how one might modify the schedule of a lesson, delete a lesson, or remove an entire course.

Below, we can see how these operations have been implemented.

To modify a lesson's schedule, the user clicks on the desired lesson. This action opens a window displaying all the lesson's details. In the top right corner of this window, there is an "Edit" button. Clicking this button navigates the user to another window where they can update the day, time, classroom, repetition frequency, and alerts for the lesson, as described earlier.

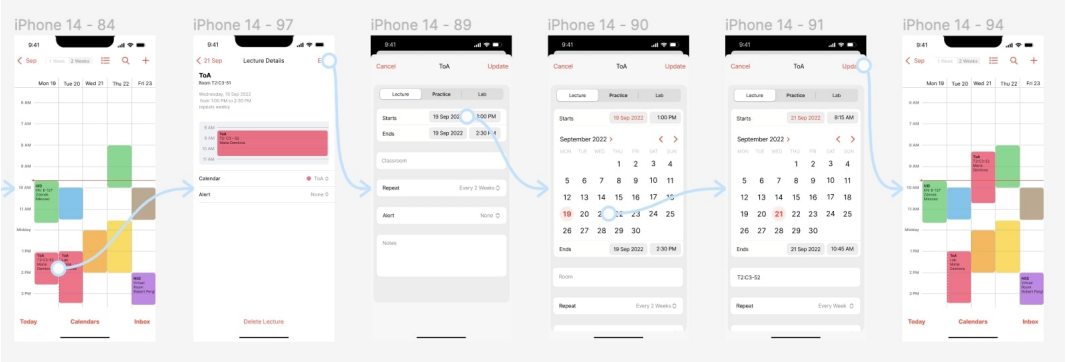

Figure 5.14: Update Lesson

To delete a lesson, there is a "Delete Lecture" button located at the bottom center of the lesson details window. By clicking this button, the user can remove the selected lesson from their timetable.

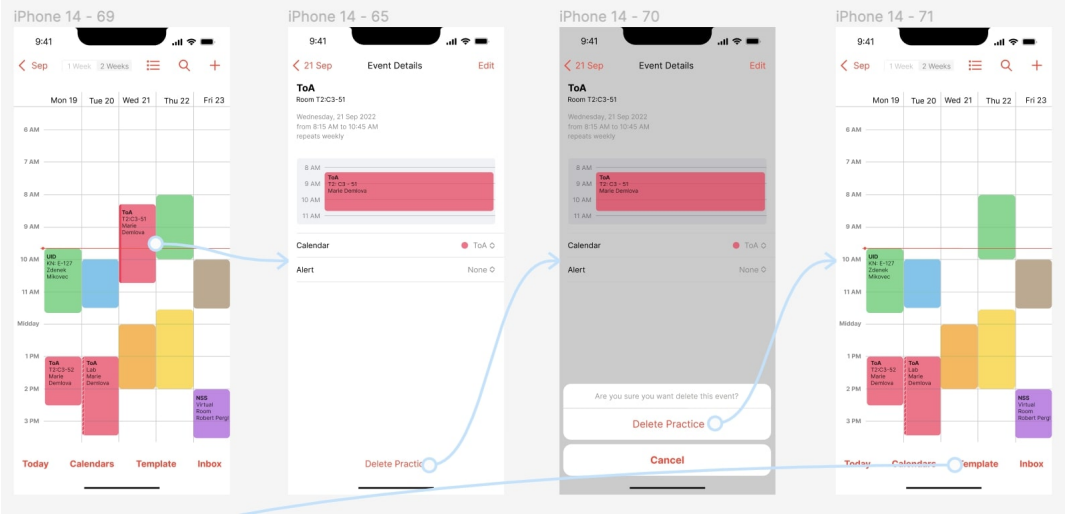

Figure 5.15: Delete Lesson

To delete an entire course, an additional button has been introduced at the bottom of the main screen. Previously, the native application provided only three buttons: Today, Calendars, and Inbox. Now, a fourth button labeled "Template" has been added. When the user clicks on this button, they are taken to a list of all the courses. By selecting a course from this list, they can then delete the entire course by clicking the "Delete Course" button located at the bottom of the course details window.

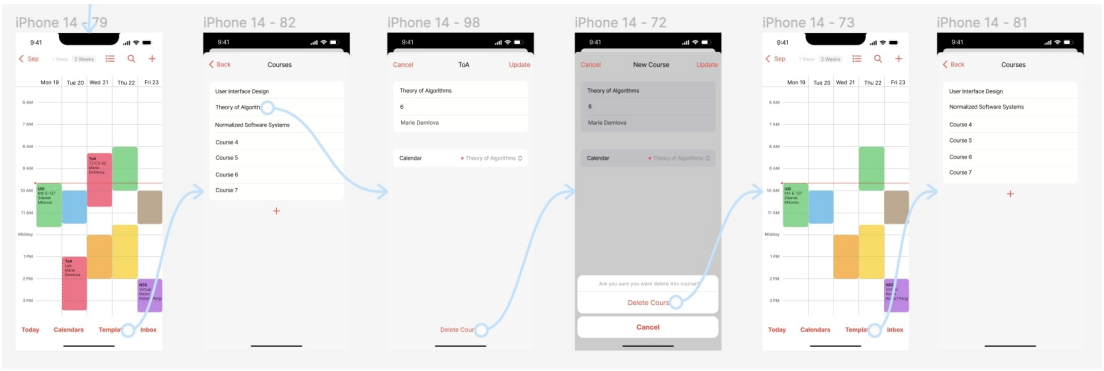

Figure 5.16: Delete Course

This concludes the detailed discussion on the implementation of the application interface using Figma, showcasing the process of creating and managing timetables. In the subsequent section, we will delve into the next phase of development, exploring how these designs and functionalities could be partially realized using Xcode. This will involve transitioning from design to actual code implementation, examining how Xcode's tools and features can bring the Figma prototypes to life, and addressing the practical aspects of developing an iOS application.

5.3 Xcode

With a clear and detailed prototype from Figma in hand, the transition to Xcode began. Utilizing SwiftUI, a modern framework for building user interfaces across all Apple platforms, I meticulously reproduced the visual elements and interactions defined in the Figma prototype. SwiftUI's declarative syntax and seamless integration with Xcode provided an efficient and effective workflow for implementing the application's core features.

Throughout the development process, various functionalities were built and refined. These included dynamic form handling, data binding, and interactive navigation—all designed to offer a user-friendly experience. This chapter will explore the technical details and challenges encountered during this implementation phase, illustrating the journey from prototype to fully functional application.

5.3.1 Timetable View

The code begins with the initial screen, TimetableView, which serves as the primary interface for users to create a new timetable. This screen includes various components that allow users to input and customize their timetable details.

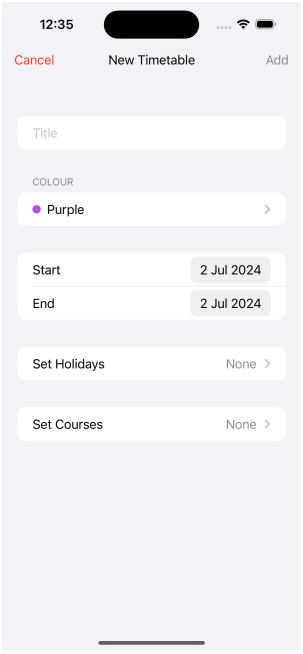

Figure 5.17: TimetableView

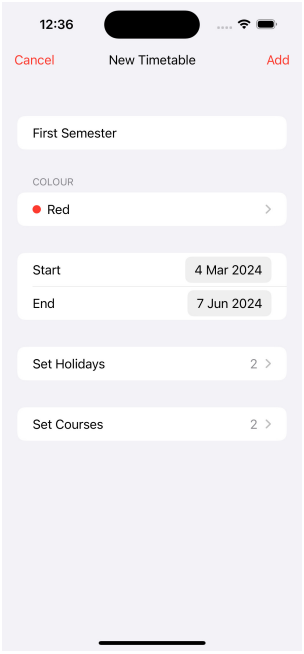

Figure 5.18: TimetableView Updated

The main features and functionalities of TimetableView include:

1. **Title Input**: Users can input the title of their timetable through a text field. This title is a mandatory field, and the presence of text in this field enables the "Add" button, ensuring that no timetable is created without a title.

2. **Color Selection**: Users can select a color to represent their timetable visually. A navigation link leads to a color picker view, where users can choose their preferred color. The selected color is then displayed alongside the link, providing immediate visual feedback.

3. **Date Management**: The start and end dates of the timetable are defined using date pickers. The application ensures logical consistency by preventing the end date from being earlier than the start date. This is achieved through a custom binding that automatically adjusts the end date if necessary.

4. **Holidays**: Users can set holidays within their timetable through a dedicated navigation link that opens a holidays management view. Here, users can add, edit, or remove holidays. The number of holidays set is displayed in the main view, providing an overview of the timetable's holiday schedule.

5. **Courses**: The timetable allows users to add courses via another navigation link leading to a course management view. Users can input course details, and the number of courses added is shown in the main view. This feature supports the organization and planning of academic schedules within the timetable.

The TimetableView screen mirrors the design created in Figma, replicating its layout and functionality with precision. However while the main screen remains true to the Figma prototype, subsequent screens feature several modifications to enhance usability and functionality. Now, let's take a closer look at these screens.

5.3.2 ColorCalendar View

The ColorCalendarView in the application allows users to select a color for their calendar.

When this view is presented, it displays a list of color options, each represented by a colored circle and its name. The user can tap on any color to select it, and the selected color is then marked with a checkmark. Once a color is chosen, the view automatically dismisses, returning the user to the previous screen with their selected color applied.
The interface is designed to be simple and intuitive, with clear visual cues for the color options and the currently selected color. The navigation bar includes a "Back" button for easy navigation, and the entire view maintains a consistent design with the rest of the application, featuring a background color that matches the app's theme.

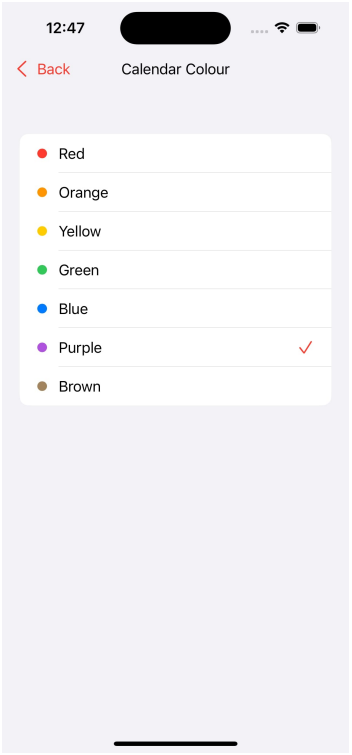

Figure 5.19: ColorCalendar View

5.3.3 Holidays View

The HolidaysView1 screen is designed to manage holidays within a timetable application. It presents a structured list where each holiday is displayed with its corresponding start and end dates. Users can effortlessly remove holidays by tapping on the trash icon next to each holiday entry. The view also features a button labeled “New holiday”, which, when tapped, transitions to the HolidaysView2 screen. This second screen allows users to specify the start and end dates for a new holiday. The entire interface is styled with a consistent background color and navigation setup, including back buttons and a principal title “Holidays” at the top. Dates are displayed in a user-friendly format that handles cases where holidays span multiple days.

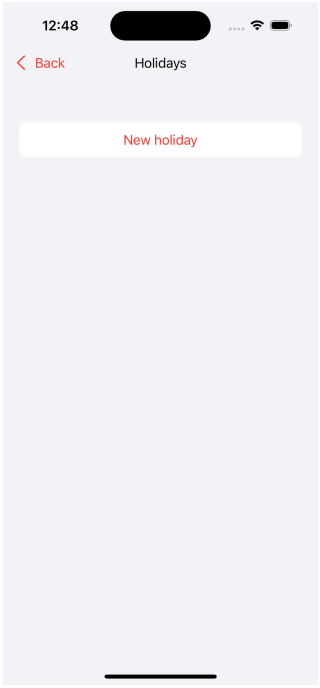

Figure 5.20: New Holiday

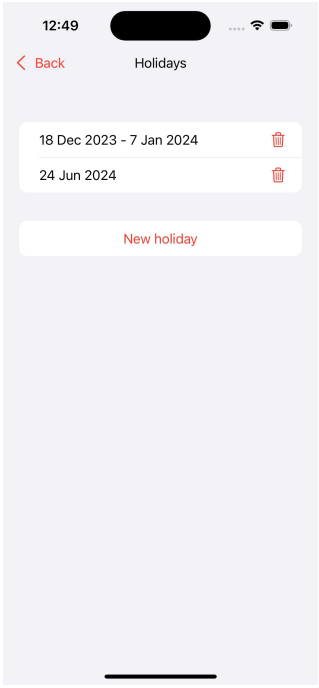

Figure 5.21: Holidays

The HolidaysView2 serves as the entry point for adding a new holiday to the timetable. It provides date pickers for selecting both the start and end dates of the holiday. To ensure data integrity, the end date picker dynamically adjusts based on the selected start date, preventing users from choosing an end date that precedes the start date. The view includes essential navigation elements such as a back button for returning to the previous screen and an "Add" button to confirm the holiday's addition with the selected dates. Like its predecessor, HolidaysView2 maintains a consistent visual style with a uniform background color and title "Holiday" prominently displayed in the navigation bar.

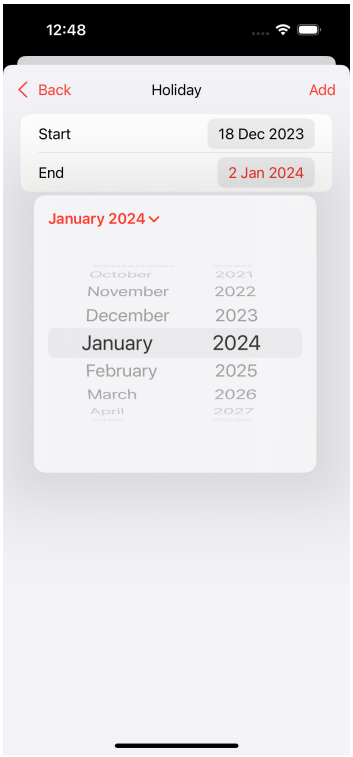

Figure 5.22: Set Date

5.3.4 Courses View

The CoursesView1 screen serves as a hub for managing courses within a timetable application. It features a list of existing courses displayed in a structured format using CourseRow components. Each course entry includes a title and a delete button that allows users to remove courses from the list. The screen also includes a prominent "New course" button, which, when tapped, navigates users to CoursesView2. This second view enables users to add new courses by providing details such as course title, number of credits, and professor. Navigation between these views is facilitated by a back button in the navigation bar, ensuring a seamless user experience.

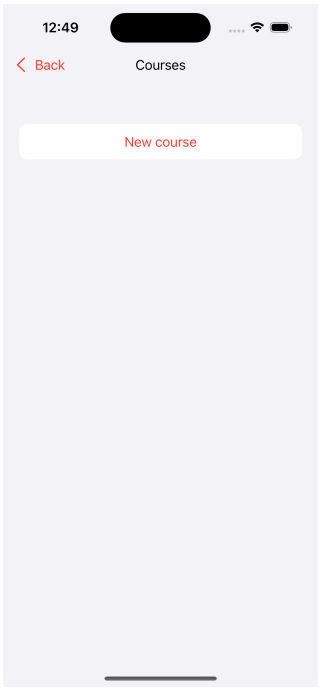

Figure 5.23: New Course

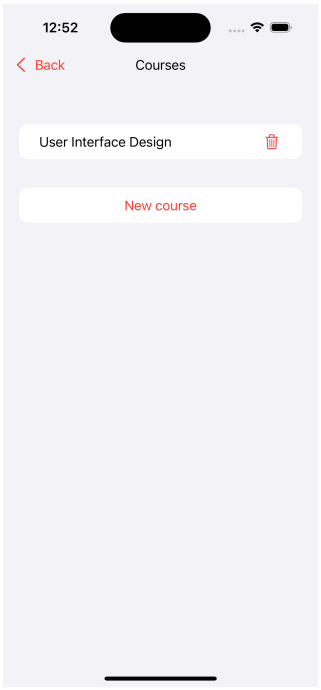

Figure 5.24: Courses

The CoursesView2 screen is dedicated to adding new courses to the timetable. It presents users with input fields for entering course details such as title, number of credits, professor, and additional notes. Users can also select the type of course from a predefined list using a segmented picker, where the selected type influences the visual representation within the interface. The view supports dynamic addition of multiple lessons for each course, with options to specify lesson start and end times, location, recurrence pattern, and alerts. A "Add Lesson" button simplifies the process of adding new lessons dynamically. The interface is complemented by a navigation bar that includes a back button for returning to the previous screen and buttons for canceling or confirming course addition, with validation to ensure required fields are filled.

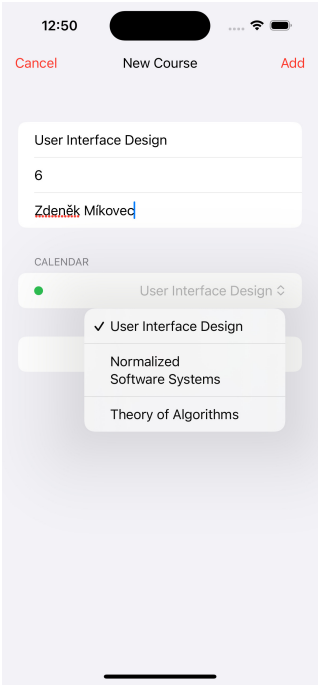

Figure 5.25: Calendar

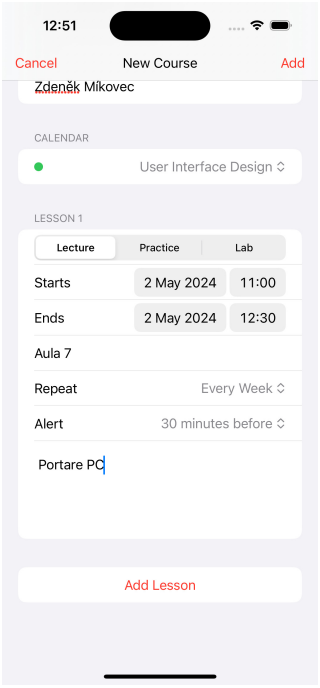

Figure 5.26: Lecture

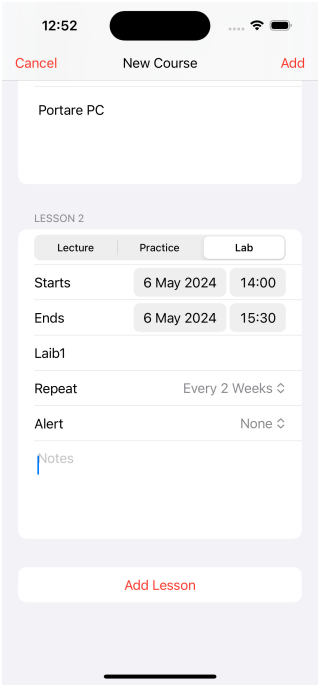

Figure 5.27: Lab

5.3.5 Views

Upon entering data for the timetable, users gain access to a comprehensive display of classes across various views. This includes detailed daily and weekly views designed to enhance scheduling efficiency and provide clear insights into upcoming lessons. Below, these views are introduced to facilitate effective time management and organization of academic or professional commitments.

Daily View

The "DayView" screen in the app serves as a detailed daily schedule view. It displays the current date and the days of the current week, with an emphasis on the current day through distinct visual markers. The interface shows each hour of the day in a vertical timeline format, allowing users to view scheduled events and appointments. A specific highlight is given to the current hour, which is dynamically updated to reflect the current time. Events, such as a lecture, are visually represented within the schedule with detailed information. The top navigation bar provides options for returning to the weekly view, filtering events, searching, and adding new entries, ensuring users can manage their daily schedule efficiently.

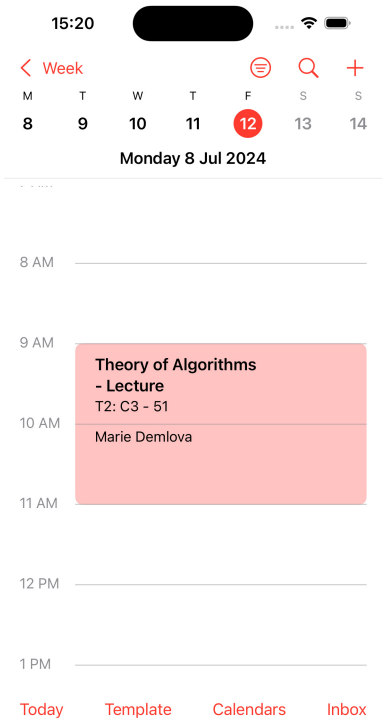

Figure 5.28: Day View

Weekly View

The "WeekView" screen in the app provides a comprehensive view of the user's weekly schedule, allowing toggling between single-week and multi-week views through options in the navigation bar. The interface organizes each weekday in a vertical timeline, highlighting the current day and hour. Events are detailed within this timeline, including course titles, locations, and instructors. The "MultiWeekView" struct presents a comprehensive schedule spanning two weeks, structured with headers for each week and detailed day-by-day event timelines. This design provides users with a complete overview of their schedule, accommodating scenarios where lessons may occur bi-weekly rather than weekly. This approach ensures clarity and efficiency in organizing and planning across a broader timeframe.

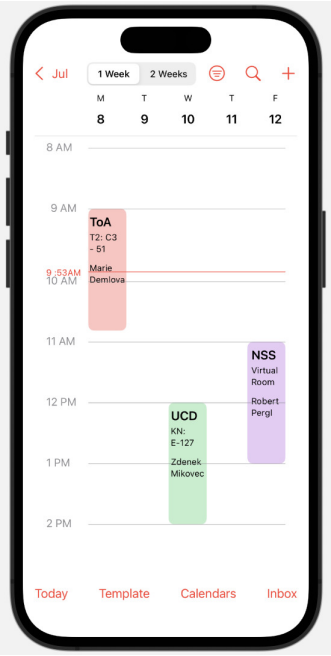

Figure 5.29: 1-Week View

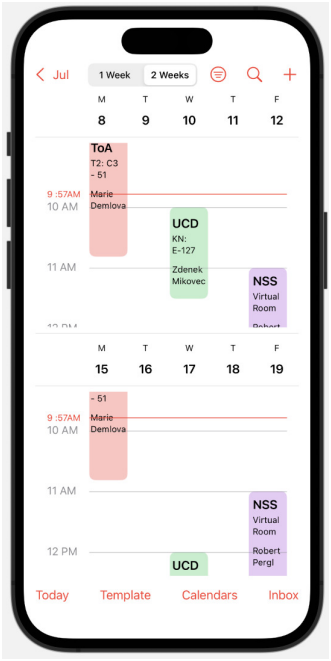

Figure 5.30: 2-Weeks view

So, the transition from the Figma prototype to the final implementation of the application, some enhancements have enriched the user experience and improved usability. One of the most notable changes was the introduction of distinct buttons such as “Add Lesson”, “Add Holiday” and “Add Course” instead of the generic “plus” button. This not only enhanced the interface’s aesthetics but also made it clearer for users where and how to add new elements to their timetable.

Furthermore, functionality to delete courses and holidays was incorporated, providing users with greater control and flexibility over managing their schedules.

Additionally, the date picker for setting start and end dates has also undergone significant changes. In the Figma prototype, the date picker uses a static calendar view, however, in the final implementation on Xcode, the date picker appears as an additional overlay window, which opens on top of the existing screen. The functionality remains the same, this is primarily an aesthetic choice.

Overall, considerable efforts have been made to adhere as closely as possible to the design choices established during the design phase. The modifications that have been implemented stem from either limitations in representing certain elements through code or purely aesthetic considerations. Nonetheless, the fundamental logic remains unchanged, thereby ensuring consistency and reliability in the functionality of the application.

Chapter 6

Test

"Pay attention to what users do,
not what they say."
Jakob Nielson

6.1 Test

The testing phase served as a critical step in assessing the effectiveness and usability of the application. We began by designing the application's interface using Figma, a collaborative interface design tool, which allowed for the creation of interactive prototypes. These prototypes were then translated into a functional application using Xcode, Apple's integrated development environment for macOS.

Once the initial implementation was complete, we conducted user testing to gather feedback on various aspects of the application. This involved presenting a questionnaire to a sample of potential users, including students and individuals familiar with timetable management applications. The questionnaire sought to assess user satisfaction, ease of use, clarity of interface elements, and any potential areas for improvement.

The testing phase involved a carefully selected group of approximately ten individuals, chosen to represent a broad spectrum of users. This cohort included undergraduate and graduate students from the university where I completed my Erasmus program, students from my current university, my thesis supervisors, and family members. This diverse selection ensured a comprehensive evaluation of the application, encompassing a variety of perspectives and experiences with timetable management. This limited number of participants was selected due to the consistent feedback patterns observed during preliminary testing phases and the logistical challenge of distributing the application for independent testing without supervision.

The testing sessions were carried out in a controlled environment, typically in university computer labs or quiet study areas, to ensure that participants could focus without distractions. Each participant was provided with a brief introduction to the application, followed by a series of tasks designed to simulate real-world usage scenarios, such as adding classes, setting up alerts, and navigating different calendar views. Detailed descriptions of the tests conducted can be found in Appendix A.

These tasks were carefully chosen to reflect common use cases and to highlight the new features introduced in the redesigned application. Participants were then asked to complete the questionnaire, which included both quantitative ratings and open-ended questions to capture detailed feedback.

The decision to use this structured approach was driven by the need to obtain actionable insights into the usability and functionality of the application. By closely observing participants as they interacted with the application and gathering their feedback, we were able to identify specific areas for improvement and make iterative enhancements.

The test results indicated a high level of satisfaction with the overall design and functionality, particularly the new vertical weekly view and the ability to view two consecutive weeks simultaneously. However, participants also provided valuable suggestions for further refinements, such as additional customization options and clearer labeling of certain interface elements.

Through the feedback collected from the questionnaire responses, we gained valuable insights into the user experience and identified areas where adjustments were needed. This feedback guided us in refining the application's design, addressing usability issues, and enhancing functionality to better meet user needs.

6.1.1 Figma: First part of the test

The simulated test scenario initiates with a preamble, emphasizing the purpose of the application and the specific functionalities to be evaluated. Users are guided through a series of tasks designed to assess the performance and usability of the application's additional features.

The test begins with an instruction to create a new timetable, mirroring a common user scenario where manual input of school schedules is necessary. Users are directed to input various details such as the title, color, start and end dates, and school holidays. These tasks aim to evaluate the application's ability to efficiently manage and organize academic timetables, including handling recurring events and special occasions.

Subsequent tasks involve adding individual courses to the timetable, each with specific instructions regarding calendar settings, start times, repetitions, and color coding. Users are then instructed to navigate between different calendar views, including weekly, bi-weekly, monthly, and yearly views, to assess the application's

flexibility and user interface across various timeframes.

The test concludes with a reverse navigation sequence, guiding users from the annual view down to the daily view, and prompting them to explore additional details of scheduled lessons. Throughout the simulation, users are encouraged to provide feedback on their experience, including any challenges encountered, suggestions for improvement, and overall impressions of the application's functionality and usability.

Overall, the structured nature of the simulated test provides a comprehensive evaluation of the application's performance, user interface, and user experience, with the ultimate goal of gathering valuable insights to inform iterative refinements and enhancements.

Feedback

The feedback obtained from users who tested the application highlighted initial difficulties stemming from the lack of an introductory tutorial. Without explicit guidance, users resorted to trial and error to familiarize themselves with the application's interface. However, as the test was conducted purely for design evaluation purposes, not all features were implemented, leading to some confusion during the initial trial.

Upon repeating the test, users demonstrated improved performance, indicating that familiarity with the application positively influenced their interaction. However, both users identified areas for improvement. They suggested implementing a feature that allows for the modification of entered courses, particularly in cases of input errors. Additionally, users noted that the transition between screens felt abrupt and could be refined to enhance the overall user experience.

So, while no further requests were made, the feedback underscored the importance of providing clear guidance for users, refining interface transitions, and incorporating features to address user input errors. These insights will inform future iterations of the application, aiming to enhance usability and user satisfaction.

6.1.2 Figma: Second part of the test

In response to the initial feedback, we have worked on enhancing the application, focusing primarily on the course modification or addition experience. This led us to introduce the second part of the test, specifically designed to assess this functionality. Here are the feedback received so far.

Feedback

Users found the addition of the template section to be intuitive and helpful. They commented that executing the test was straightforward and smooth, with no

hiccups encountered along the way.

Regarding the colors chosen for courses, they were appreciated by users and were deemed sufficient for the potential number of courses in a semester. However, users expressed a desire to have the option to choose additional colors beyond those provided. When comparing the chosen colors with those from the original application, users preferred the new colors for their improved visibility and distinguishability. Although the colors were similar in general, the intensity varied.

During the test, we collected feedback on the visual cues used to differentiate between lectures, practices, and labs also.

Users evaluated two options: one entailed utilizing different border lines for each category, while the other incorporated emojis. They found both solutions to be effective and visually striking. However, when expressing their preference, users leaned towards the option with distinct border lines. The majority agreed that this approach was more elegant and better aligned with the design conventions seen in various Apple applications.

6.1.3 Xcode: Test

The simulated test scenario begins with a clear introduction, emphasizing the goal of thoroughly testing the functions implemented with Xcode. This simulation mirrors the tests conducted for the Figma prototype, enabling a comparative analysis of various feedback to inform improvements to the application.

The test starts in the Timetable view, where users are guided through a sequence of tasks to evaluate the application's performance. Initially, users are instructed to input the title of the timetable, change its color, and set the start and end dates. Following this, they are prompted to add school holidays.

Next, users proceed to enter various courses into the timetable. The tasks involve adding a course, setting the calendar color, specifying the start and end dates, and setting the lesson repetitions. Users then add two other courses.

The final task requires users to attempt deleting one or more courses by clicking on the bin icon, observing the application's response to deletion actions.

Feedback

The test did not encounter any issues; the user was able to complete all the steps smoothly and efficiently without requiring any assistance or support. Each function, from entering the timetable title to changing its color, setting start and end dates, and adding holidays, worked as intended. The user successfully added multiple courses with different schedules and managed to navigate the interface with ease. Furthermore, the deletion of courses was executed without any complications, demonstrating the robustness of the implemented features.

Additionally, users appreciated the change from a generic "plus" button to more specific buttons such as "New Holiday", "New Course" and "Add Lesson", finding it significantly enhanced the clarity and usability of the interface.

Overall, the user experience was seamless, confirming that the application meets the design and functionality expectations.

In conclusion, the iterative testing process has been invaluable in identifying both strengths and areas for improvement in the redesigned Apple Calendar application. The feedback underscores the importance of user-centered design principles, highlighting the need for intuitive onboarding, flexible course modification options, a wider color palette, and clear visual differentiation of course types. These insights will inform ongoing refinements, ensuring the application continues to evolve to meet the diverse needs of its user base.

Chapter 7

Conclusion

"To find ideas, find problems.
To find problems, talk to people."
Julie Zhou

This thesis aims to redesign an existing iOS application to enhance its functionality, particularly by integrating specialized templates tailored to accommodate university schedules into the Apple Calendar app.

In pursuit of this objective, the design process followed a comprehensive approach, encompassing all stages from identifying the target user and scenarios, to storyboarding, Hierarchical Task Analysis (HTA), sketching, mock-up creation, prototyping, and culminating in the testing phase. This meticulous approach ensured that the redesigned application would align closely with user needs and preferences, thereby enhancing the overall user experience.

The feedback obtained from user testing provided valuable insights into areas for improvement and potential enhancements. Users expressed appreciation for the addition of specialized templates for university schedules, noting the improved organization and efficiency it brought to their daily routines. However, they also identified areas where further refinement could be made, such as providing users with the flexibility to choose any color from a complete spectrum, integrating features like calculating university GPA, and offering the ability to switch between template and standard calendar modes seamlessly.

These insights serve as valuable considerations for future iterations and enhancements of the application. By incorporating user feedback and implementing additional features the application can further enhance its usability and meet the evolving needs of its user base.

In summary, this thesis serves as a foundation for continued exploration and refinement in the realm of mobile application design, with a focus on improving user experiences and meeting user expectations.

Appendix A

Test

A.1 Test 1

Surely you have already dealt with the application you are about to use, what you are going to test, however, will be additional functions.

How many times, when opening the calendar application on your iPhone, have you found yourself manually entering your school timetable, without a proper template that did not take into account various aspects, such as school holidays, teachers, classrooms, etc.?

Well, this application was created to make this process automatic and more efficient.

The objective of this simulation is to test the functionality of the presented application by receiving feedback from a possible user. The simulation will be guided and only certain functionalities may be executed.

Let us start!

The simulation starts from a monthly calendar view, now follow my instructions:

1. Add a new event/timetable. Now we want to create a new timetable to include all the courses you are taking at the university.
2. Do to enter the title. (Now click on the title again, the keyboard function is not yet enabled, the title will be added automatically).
3. Change the color of the timetable to blue.
4. Set the start and end dates of the timetable:
   - Put 19 September as the start date;
   - Set the end date as 15 January.
5. Add school holidays: enter 17 November as the holiday.

Now we start entering the various courses. Then return to the start page of the timetable.

6. Add a course and set the calendar by changing its color to green.

7. Check that the start time is 9:41 AM by opening and closing the window.

8. Set the repetitions. This lesson repeats every week.

9. Add the course and enter another one, the course in question is Theory of Algorithms.

10. Choose a new calendar.

11. Set the repetitions. This lesson repeats every 2 weeks.

12. Add the course and enter another one, the course in question is Normalised Software Systems.

13. Set the repetitions. This lesson repeats every 2 weeks.

14. Add course.

You have now entered all courses.

15. Add the timetable.

You are now in the weekly view. Try the other views:

16. 2-Weeks view.

17. Monthly view.

18. Yearly view.

Now do the reverse procedure, from annual view to weekly view.

19. Daily view of 21 September.

20. Get more details of the lesson and return to the daily view.

And that's all for now, thank you for your time!

A.2 Test 2

The following test aims to modify the current template of the application. The simulation starts from the week calendar view, follow the instructions below:

1. Get more details from Wednesday's Theory of Algorithms lecture.
2. Delete that practice.
3. Get more details from Monday's Theory of Algorithms lecture.
4. Edit the lesson by changing the date from September 19 to September 21 and update the template.
5. Now go to the template section and try deleting the Theory of Algorithms course.
6. Having done this, return to the template section and check whether or not the course is still present in the course list.

And that's all for now, thank you for your time!

A.3 Test 3

The objective of this simulation is to thoroughly test the functions implemented with Xcode. This simulation will partly mirror the tests conducted for the Figma prototype, allowing to compare various feedback and make informed improvements to the application.
The simulation will be guided.
Let us start!

The simulation starts from the Timetable view, now follow my instructions:

1. Do to enter the title.
2. Change the color of the timetable to blue.
3. Set the start and end dates of the timetable:
   - Put 19 September as the start date;
   - Set the end date as 15 January.
4. Add school holidays: enter 17 November as the holiday.

Now we start entering the various courses.

5. Add a course and set the calendar by changing its color to green.
6. Set the start and end dates.
7. Set the repetitions. This lesson repeats every week.
8. Add the course and enter another one, the course in question is Theory of Algorithms.
9. Choose a new calendar.
10. Set the repetitions. This lesson repeats every 2 weeks.
11. Add the course and enter another one, the course in question is Normalised Software Systems.
12. Set the repetitions. This lesson repeats every 2 weeks.
13. Add course.

You have now entered all courses.

14. Add the timetable.

Now try to delete one or more courses.

15. Click on the bin icon and see what happens.

And that's all, thank you for your time!

Bibliography

[1] Columbia Engineering. *UX vs. UI*. 2021. URL: https://usa.bootcampcdn.com/wp-content/uploads/sites/108/2021/08/Key-differences-ux-ui-design.jpg (cit. on p. 4).

[2] Istiak Ahmed. *The world of UX and UI*. Pinterest. URL: https://dribbble.com/shots/16604089-UI-Design-vs-UX-Design (cit. on p. 5).

[3] Thad Thompson. «7 best student planner apps». In: (Jan 21, 2022). URL: https://calendly.com/blog/best-planner-apps-for-students (cit. on p. 7).

[4] Rachel Roseberry. «How To Create Digital Comics with Pixton As Told In Comic Form». In: (Jan. 2022). URL: https://www.nypl.org/blog/2022/01/28/how-create-digital-comics-pixton (cit. on p. 19).

[5] Sam Grainger, Feng Mao, and Wouter Buytaert. «Environmental data visualisation for non-scientific contexts: Literature review and design framework». In: *Environmental Modelling & Software* 85 (Nov. 2016). DOI: 10.1016/j.envsoft.2016.09.004 (cit. on p. 23).

[6] Visme. *25+ Stunning Wireframe Examples for Web Designers*. 2022. URL: https://visme.co/blog/wireframe-examples/ (visited on 02/13/2024) (cit. on p. 24).

[7] Emily Esposito. *Low-fidelity vs. high-fidelity prototyping*. URL: https://www.invisionapp.com/inside-design/low-fi-vs-hi-fi-prototyping/ (visited on 05/18/2018) (cit. on p. 27).

[8] Figma Community. *high-fidelity prototype*. URL: https://www.figma.com/community/file/988854619809579450/high-fidelity-prototype (cit. on p. 29).

[9] *Cloe Toms*. URL: https://infograph.venngage.com/infographics (cit. on p. 38).

[10] *Pixton*. URL: https://www.pixton.com/welcome (cit. on pp. 40–42).