Gamification in Smartphone Applications to Motivate Medication Adherence for Elderly

Sofie Tjønneland Urhaug

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
60 credits

Institute for Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

Gamification in Smartphone Applications to Motivate Medication Adherence for Elderly

Sofie Tjønneland Urhaug

Gamification in Smartphone Applications to Motivate Medication Adherence for Elderly

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Purpose

The number of heart failure patients is a growing issue. Beta-blockers are a medication form that has shown a better prognosis for patients, by lowering their resting heart rates. Still, one in three patients is non-adherent to their medication. Medication Reminder is an application created to remind patients to take their medication when their resting heart rates reach above a boundary. This application was created as a minimal viable product. The purpose of this thesis is to further develop this application, in two steps. First, two health frameworks from Apple Ecosystem, CareKit and ResearchKit will be integrated. Second, methods of gamification will be integrated, using the two frameworks. All to better the patients' engagement and motivation towards achieving medication adherence.

Methods

The process of mapping out the further development of Medication Reminder was done in three steps. First, it was important to understand the background of the problem. In regards to medication adherence, gamification and, especially gamification for adherence and gamification for elderly. Secondly, the Medicine Reminder application was studied, to map out how it worked and how it could be changed. Finally, ResearchKit and CareKit were studied to figure out how they could be utilized to include gamification features in the application. The gamification elements were chosen based on the theoretical background as well as the capabilities of the frameworks. Therefore a streak and feedback feature was introduced to the application. The streak works by the user registering that they have taken their medication every day. A daily reminder to take their medication was also implemented. The features were tested through Alpha testing, in two testing intervals. The application was altered based on feedback between the two intervals.

Results

The two test intervals yielded insight into how the gamification elements were working. The first tester found the differentiation between the reminder to take their medication and the registration of their medication-taking confusing. This was therefore explained in more detail in the onboarding of the application. The second tester reported that the streak had triggered his competitiveness. The feedback on the other hand had not been noticed by the tester. Another important discovery was the importance of patients' routines. Every time the tester lost his streak it was due to his routine changing. The tester also suggested that the time of the reminder should be able to be changed after setting it the first time.

Conclusion

Based on the results of the initial testing the streak feature shows good promise. The feedback needs to be more prominent to the user in the application. Routines and how they change are an important consideration, to support the patients when their routines change the application should also be able to change some of its behavior accordingly.

Acknowledgements

First and foremost, I want to express my gratitude towards my thesis supervisor, Jens Johan Kaasbøll, for his support during this project. This thesis would not have been where it is today without your feedback and guidance.

I am grateful for the testers' feedback and insights, which were crucial when evaluating the application. Thank you for taking the time. In addition, I would like to thank my fellow DHIS2 design lab students, notably Stilrommet. Thank you for engaging discussions, long lunches, and joyful times.

I'd like to express gratitude to my family and friends for their constant encouragement and support. Thank you to my parents for their reassuring words, always just a phone call away. I'm grateful for my brother, who always provides solutions to my problems, academically and generally.

I particularly want to thank Elisabeth, for her friendship and support throughout this master's degree. You have filled my life with laughter, even in the midst of quarantines, intense thesis writing, and freezing winters.

Contents

1 Introduction **1**
1.1 Medical Background 1
1.1.1 Heart Failure 1
1.1.2 Beta-blockers 1
1.2 Adherence 2
1.3 Wearable Heart Rate Monitors 2
1.4 Reminders Through Smart Devices 3
1.5 Gamification 3
1.5.1 Gamification of Snapchat 5
1.5.2 Gamification of Waze 6
1.5.3 Gamification for Adherence 7
1.5.4 Gamification for Elderly 8
1.6 Project Plan 9

2 Technical background **10**
2.1 Apple Ecosystem 10
2.1.1 Apple Watch 10
2.1.2 Xcode 10
2.1.3 Swift 10
2.1.4 TestFlight 10
2.2 Frameworks 11
2.2.1 HealthKit 11
2.2.2 ResearchKit and CareKit 11
2.2.3 ResearchKit Surveys 12
2.2.4 CareKit 14
2.2.5 Integrating ResearchKit with CareKit 15
2.2.6 Notifications 18
2.3 MedRem 19
2.4 Problem Statement 20

3 Requirement Analysis **21**
3.1 Requirement Specification Process 21
3.2 Introducing Gamification 21
3.3 Requirement Specification 24
3.3.1 Stakeholders 25
3.3.2 User Stories 25
3.3.3 Functional requirements 26
3.3.4 Non-functional requirements 29
3.4 Apple Design Principles 29
3.5 Inclusive Design 31

4 Design and Development **32**
4.1 Agile Development Process 32
4.2 Development 32
4.2.1 Studying ResearchKit and CareKit 32
4.3 Integration with the MVP 33
4.4 Implementing New Functionality 35

4.4.1 Core Functionality . . . 36
4.4.2 Streak . . . 38
4.4.3 Feedback . . . 39
4.5 Onboarding . . . 41
4.5.1 Flow . . . 44
4.6 User Interaction & Experience . . . 46

5 Evaluation **47**
5.1 User Testing . . . 47
5.1.1 Test Execution . . . 47
5.1.2 First test interval . . . 48
5.1.3 Second Test Interval . . . 49
5.2 Evaluation of Requirements . . . 50
5.2.1 Onboarding . . . 50
5.2.2 Reminder Notification . . . 51
5.2.3 Streak . . . 51
5.2.4 Feedback . . . 52
5.2.5 Level, Badges and Titles . . . 53
5.2.6 Non-functional Requirements . . . 53

6 Conclusion and Future Work **57**
6.1 Conclusion . . . 57
6.2 Future Work . . . 58
6.2.1 Routines . . . 58
6.2.2 Enhance and promote social interactions . . . 59
6.2.3 Sense of urgency . . . 61
6.2.4 Involving the Care Provider . . . 62
6.2.5 Resting Heart Rate as Control Mechanism . . . 62

Appendices **68**

A Source Code **68**
A.1 MedRem . . . 68

List of Tables

1 How the most prominent gamification elements for elderly is linked to Chou (2019) 8 core drives. 8
2 Definition of each of the three health related kits from Apple. 12
3 Gamification elements prominent in gamification for adherence and elderly 23
4 Core drives linked to the two gamification features 24
5 An overview of the applications non-functional requirements, divided into three categories 30
6 The different type of feedback the user gets on each the different categories of adherence 40
7 Evaluation of Product Requirements 53
8 Evaluation of Organizational Requirements 55
9 Evaluation of External Requirements 56

List of Figures

1 Octalysis that represents the eight core drives of gamification ("Octalysis", 2020) . . . 5
2 The UI of a ResearchKit-Survey Question Step with the code from 1 . . . 12
3 Sequence diagram illustrating how ResearchKit Surveys are integrated with CareKit. . . . 18
4 Use case diagram modeling the functional requirements of the core functionality of the application . . . 27
5 Use case diagram modelling the functional requirements of onboarding of the application . . . 28
6 Use case diagram modeling the functional requirements of the feedback mechanisms of the application . . . 29
7 The user interface of the MedRem MVP . . . 34
8 How the requirements of different categories are connected to parts of the application. . . . 35
9 The application's user interface with a unfinished and finished task. . . . 37
10 The user interface that displays the users current streak . . . 38
11 The MVPs notification of a too high resting HR . . . 39
12 Complete onboarding screen . . . 42
13 The onboarding displayed in the view controller when not completed . . . 43
14 Welcome page to the onboarding . . . 43
15 The reminder the user receives at a predefined time every day to take their medication. . . . 43
16 The Permission step of the onboarding . . . 44
17 The Permission step of the onboarding . . . 44
18 The Boundary Heart Rate Step of the onboarding . . . 45
19 The Medication Reminder step of the onboarding. . . . 45
20 Application with dynamic font sizes . . . 46
21 Example Wordle game . . . 60
22 Example of Wordle statistic and sharing page . . . 60
23 How a shared message of played Wordle looks . . . 61
24 An idea of how such a message could be formatted . . . 61
25 The image that is exported from the Slutta application for sharing . . . 62

1 Introduction

This thesis aims to research, develop, and test how an application for iPhone and Apple Watch can aid adherence to the medication of heart disease patients. The minimal viable product (MVP), named MedRem (Aanesen, 2021), is the basis for further development. The application will integrate two medical frameworks, CareKit and ResearchKit, from the Apple Ecosystem. When integrating CareKit into the application, it becomes more responsive, which requires the user to interact with the application to track their medication adherence. Gamification tactics are used to improve adherence in many fields, such as eHealth-programs. Therefore, to maintain and motivate the interaction needed, game elements will be implemented. This thesis will investigate how gamification can motivate patients to achieve medication adherence. In addition, it will be explored how CareKit can facilitate the integration of gamification elements. Further discussion is given in the problem statement in Section 2.4.

1.1 Medical Background

1.1.1 Heart Failure

Heart failure (HF) is a syndrome caused by cardiac dysfunction that reduces life longevity. The symptoms and signs of HF may require hospitalization or more frequent doctor visits. While HF can have different causes, it ultimately leads to the heart not being able to pump sufficient amounts of blood into the circulation. Kannel and Belanger (1991) reviewed HF in regards to data from the last 34 years. They found that 37% of men and 33% of women died within two years of diagnosis. The 6-year mortality rate was four to eight times greater than for the general population of the same age (Kannel & Belanger, 1991). More than half of the patients admitted for HF will be readmitted within six months (Krumholz et al., 1997). HF is assumed to increase over the next 20 years steadily. Consequently, HF will continue to be a substantial and growing public-health burden (Dunlay et al., 2017).

1.1.2 Beta-blockers

Beta-blockers are currently an essential treatment for HF. Beta-blockers work by blocking the actions of hormones like adrenaline in the nervous system and avoiding the activation of the "fight-or-flight" stress response. Some of the effects of the "fight-or-flight" response are higher blood pressure and heart rate (HR). The beta-blockers reduce these effects and thus relieve the strain on the heart (Øye & Levy, 2021). In addition, beta-blockers give a 10-40% reduction in mortality and hospitalization within one year (Hernandez et al., 2009). Accordingly, the use of beta-blockers lowers the HR, which has shown a better prognosis for patients (Kotecha et al., 2017).

1.2 Adherence

Adherence is defined as "the extent to which a patient acts in accordance with the prescribed interval, and dose of a dosing regimen" (Cramer et al., 2008). Even though beta-blockers positively affect mortality and hospitalization, there is an issue of adherence. Desta et al. (2021) measured the difference between beta-blockers prescribed and dispensed and found that one in three patients was non-adherent. With adherence being a central factor in improved long-term outcomes, missing adherence is essential in avoiding re-admittance and poor outcomes.

There are many different ways to measure if a patient is adherent. Desta et al. (2021) used the number of pills dispensed versus the number of pills prescribed. A weak spot of this method is that it leaves out the possibility of a patient dispensing all their beta-blockers but simply not taking them. Other examples include measurements of plasma and urinary levels and the use of electronic containers that count the number of times the patients open them. These two approaches have a common problem, accessibility. While plasma and urinary level measurements have high accuracy, they are only available in some hospitals and laboratories. Furthermore, they entail a cost and extra work for the patient. In addition, electronic containers are not commonly available. Kociánová et al. (2017) researched how the patient's heart rates could measure their adherence to beta-blockers. Their results indicated that 75.5 beats per minute (BPM) predicted non-adherence to beta-blockers.

An essential aspect of adherence is also what determinants of adherence are. The determinants of why patients are non-adherent are essential to understand to work against them. In a review of systematic reviews, Kardas et al. (2013) highlighted the lack of support, lack of motivation, and forgetfulness as the most critical factors of adherence.

1.3 Wearable Heart Rate Monitors

The HR of a patient can be a good and available measure of the adherence to beta-blockers. However, using HR as a source of adherence introduces concerns that need to be addressed. Firstly, the accuracy of the measured HR has to be high. Secondly, the HR monitor device needs to be of little disturbance or annoyance to the patients. Few people want to be easily identified as suffering from illness. Therefore, the device should be as discreet as possible. Finally, the factor of availability of the device is also essential.

The method and point of measurement are important aspects related to the accuracy of HR monitors. The gold standard for measuring HR is using an electrocardiogram (ECG). The goal is to find a heart rate monitor method close to this standard while still fulfilling the goals for availability and wearability. When compared to standard ECG, photoplethysmography (PPG) was shown to be a valid proxy for measuring HR (Barrios et al., 2019). The technology of PPG is based on the fact that blood is red and thus will absorb green light and reflect red light. Consequently, the amount of blood flowing through the wrist can be found by using green LED light paired with light-sensitive photodiodes. When the heart beats, the volume of blood is more significant in the wrist than between beats. These two factors allows us to identify beats and what the HR is (Apple Inc., 2022f).

A PPG-sensor can be placed on multiple body areas, for instance, in armband-based and wrist-based devices. Armband-based has been shown to have the highest level of accuracy, while the patients preferred wrist-based. Since the accuracy of wrist-based devices is shown to be adequate, the factor of comfort for the patients favours the wrist-based alternative (Barrios et al., 2019).

1.4 Reminders Through Smart Devices

A tactic to improve adherence to medical treatment is to use a reminder system. As smartphone use is quite widespread in society today, the accessibility of these apps is a possible benefit for patients with medical regimens. Therefore, a notification-based system through smartphone apps shows good promise to improve adherence (Choi et al., 2015).

1.5 Gamification

Deterding et al. (2011) defines gamification as "the use of game design elements in non-game contexts." Deterding et al. (2011) highlights the distinction between playful and gameful, stating that gamification does not involve playful interactions or design. Then what differentiates gamification and games? In gamification, the context includes game elements that are not combined to become an entire game. A group of users can make a gamified application into a complete game by, for instance, sharing and comparing their ranks and scores. Still, the purpose of the application is not to be a game. Serious games and gamified applications share the trait where the purpose is not bound to entertainment, but also here, the distinctive factor between them is that the application does not equal a game (Deterding et al., 2011).

A review done by Hamari et al. (2014) indicated that gamification provides positive effects; the majority of the studies reviewed documented positive effects. However, the findings also indicated that the effects depended on the context and the users.

In his book Actionable Gamification: Beyond Points, Badges and Leaderboards, Chou (2019) defines gamification as the "the craft of deriving all the fun and addicting elements found in games and applying them to real-world or productive activities". Chou (2019) describes the eight core drives within gamification.

1. **Epic Meaning & Calling**
   Contributing to something bigger than you, and that you are "chosen" to do it.

2. **Development & Accomplishment**
   The internal drive for making progress, developing skills, achieving mastery, and eventually overcoming challenges. "Challenge" is a keyword here because without a challenge the badge or trophy is not meaningful at all.

3. **Empowerment of Creativity & Feedback**
   The need to see the results of your creativity, receive feedback and adjust in turn.

4. **Ownership & Possession**
   Where users are motivated because they feel like they own or control something. With ownership, you want to increase and improve what you own.

5. **Social Influence & Relatedness**
   Social elements that motivate people. Including mentorship, social acceptance, social feedback, companionship, and even competition and envy. It also taps into that we tend to draw closer to people, places, and events that we can relate to.

6. **Scarcity & Impatience**
   Wanting something because it is rare, exclusive, or immediately unattainable. The fact that people cannot get something right now motivates them to think about it until they can.

7. **Unpredictability & Curiosity**
   Constantly being engaged because you do not know what is happening next. The primary core drive behind gambling addictions, or on a lighter level why people watch movies and read novels.

8. **Loss & Avoidance**
   The motivation to avoid something negative from happening. Like losing previous work, or admitting that everything you did up to this point was useless because you are now quitting. Under this drive, the fading of opportunities plays a part, like special offers only available during a limited time.

Chou (2019) states that if there are none of these eight core drives behind the desired action, there is no motivation to do said action. The core drives have different natures; some make the user feel powerful, others create urgency, obsession, and addiction and, in turn, make the user feel bad. Some give long-term effects, others short-term.

The eight drives are further categorized into the right-brain and left-brain core drives, which correlate with their position in the octagon displayed in Figure 1. Chou (2019) defines the right brain core drives as the ones that focus on creativity, self-expression, and social dynamics. These are primarily associated with intrinsic motivations; the activity itself is rewarding. On the other hand, the drives associated with logic, analytical thought, and ownership are left-brain core drives. These rely on extrinsic motivation or motivation from the desire to obtain something. Chou argues that implementing right brain drives will, due to their nature, yield a motivation that sticks.

Currently, gamification elements are present in most market sectors, such as social media, learning platforms, and GPS navigation applications. Examples of these categories are Snapchat and Waze. These two examples will be presented and compared to the core drives of gamification.

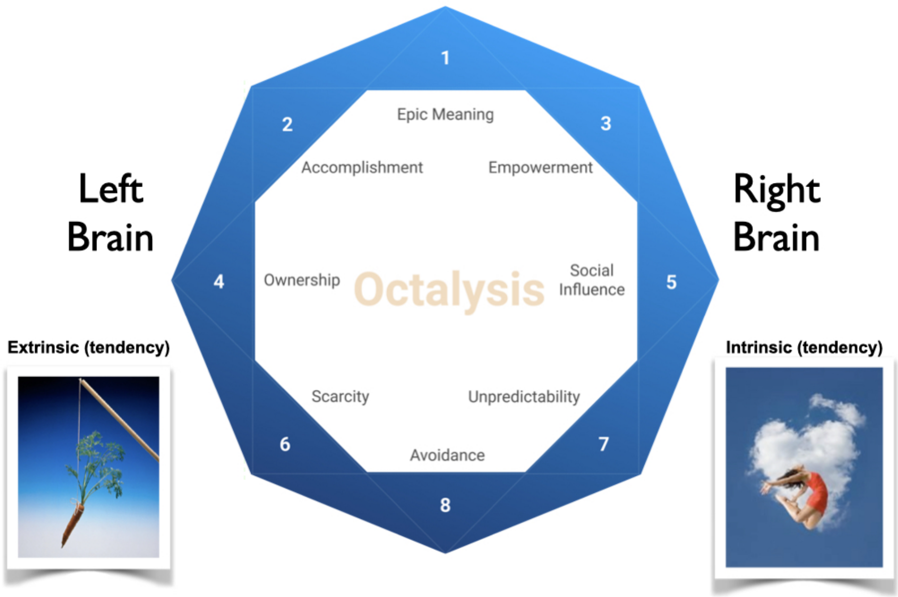

Figure 1: Octalysis that represents the eight core drives of gamification ("Octalysis", 2020)

1.5.1 Gamification of Snapchat

Snapchat is an application for chatting with friends with pictures and videos, only available to watch once. In the spring of 2015, Snapchat implemented a new feature, Snapstreak (Støstad, 2017). Snapstreak indicates how many days in a row two users have both sent pictures or videos to each other. Støstad (2017) wrote an editorial about teenagers that were addicted to keeping their Snapstreaks. They reported of teenagers on school trips walking to mountain tops to get reception on their cell phones and to send Snapchats to keep their Snapstreaks. Some teenagers on the same trip had paid for friends to log onto their accounts to keep their streaks going while they were away. One of the teenagers interviewed has 138 streaks with different people, and the longest one is 648 days. Each day she sends all 138 people two snaps, so-called "streaksnaps". These snaps have no other purpose than to make sure that they keep their streaks. This feature alone taps into three core drives of gamification, 2. Development & Accomplishment, 5. Social Influence & Relatedness, and 8. Loss & Avoidance. The accomplishment of keeping the streak gives even more excitement to the user, but also, the fear of losing it motivates them to use the application. The streak also gives you companionship with the person you share your streak with and competition with others who have the longest and most streaks. When interviewed, the teenager says she wishes to delete Snapchat since it takes so much time of her day, but she feels that she cannot due to the achieved streaks. She even states that it gets harder the longer the streak gets, ”When I have over 600 days with Kaja [friend], it means that we have put effort into it, we have put effort into our friendship.” which especially resonates with core drive number 5, and 8.

1.5.2 Gamification of Waze

Waze is an application of crowd-sourced navigation. This means that it relies on the users to share real-time traffic information such as accidents, traffic jams, and roadworks. Through these feedback mechanisms, Waze calculates new and smoother routes and notifies the users (“Driving Directions, Traffic Reports & Carpool Rideshares by Waze”, n.d.). Therefore, the success of Waze depends on the users being active and engaged. The gamification elements and which drives they relate to are listed in the following.

- **Score and Level System**
  The users that report real-time driving information receive likes and comments on what they report. This feedback is then translated into points, resulting in higher levels. The core drive that is prevalent here is 2. Development & Accomplishment.

- **Badge Reward System**
  Once the user reaches the highest level, they can obtain status symbols and unique virtual goods. For this gamification element, there are two core drives present, 2. Development & Accomplishment, for the same reasons as the previous point, but also 6. Scarcity & Impatience. Badges that are difficult to obtain and thus rare trigger the drive.

- **Avatars**
  The users can share how they are feeling using ”mood”-avatars. Loyal app users can get unique avatars. The avatars can be customized as the user would like. This is a clear example of 4. Ownership & Possession. However, as new avatars unlock after a level increase, it also plays into the 2. Development & Accomplishment. In a sense, the drive of 5. Social Influence & Relatedness play in due to the sharing of feelings and a sense of community with other users.

- **Leaderboards**
  Users can see how they are doing compared to others. The higher participation, the higher the points, and the higher the user’s rank. This amplifies the drive 2. Development & Accomplishment.

Otherwise, the application also uses the drive of 1. Epic Meaning & and Calling. When users contribute, they are a part of something bigger than themselves due to their information creating significance for others driving in their area. Waze’s success hinges on users engaging and being active, and they manage to create motivation and incentive through engaging multiple core drives.

1.5.3 Gamification for Adherence

A systematic review of gamification for adherence Croon et al. (2021) found that 19 out of 27 studies reported a significant positive effect or trend on adherence. No studies reported any negative effects. The majority of the studies implemented points (85%) and feedback (67%). Points can be implemented in various ways, often as simple numerical values reflecting an action or combination of actions. Often, points are displayed on a leaderboard or used towards achievements or badges. Chou (2019) defines feedback mechanisms as "information delivery mechanisms that communicate to the user that their actions are meaningful." For instance, the feedback can include tracking their progress or feeling the urgency of time.

In addition, the possibility of social interactions was implemented in the majority of the studies. Bonding between participants made them more adherent (Scase et al., 2017). This was done in various ways such as social networking, teams, opponents, social status, and communication features (Croon et al., 2021).

However, the combination of adherence and gamification has its issues. Bodduluri et al. (2017) suggested that the motivating effects would be lost in the long term unless there is a variety or progression of challenges in the task. Fotaris et al. (2015) saw how game elements could lead to a decrease in motivation, where participants lost interest once they trailed behind on the leaderboard.

With this in regard, Croon et al. (2021) concludes that their findings support the hypothesis that gamification has a positive effect on adherence. Nevertheless, it is important to take the fading effect of gamification into account.

Kamat et al. (2021) found that the combination of repeated notifications and the use of gamification elements had a positive trend in medication adherence for diabetes patients. The notifications were repeated four times per dose over two hours. These included an in-app reminder when the dose should be taken, a text reminder 60 minutes after, a notification to the caregiver after 90 minutes, and an automated phone call to the patient after 115 minutes. A tablet held in front of the phone camera verified that the patients had taken their medication. An AI algorithm would perform "pill identification and pill counting." When this was done, the application would assume that the patient took the pill and then calculate the adherence as a percentage. When a dosage was taken and verified, the patient was given a random number of "KYT points" in scratch cards. If the patient maintained 80% adherence throughout the study, the "KYT points" would be converted into a proportionate amount of money. The study results showed that the patients enrolled with this application only missed 12.17% of their doses, whereas the control group missed 31.89%. No patients in the control group achieved 80% adherence throughout the study period, while among the patients using the application, 57 out of 59 (96.6%) made it over the 80% mark.

1.5.4 Gamification for Elderly

Patients suffering from heart failure are mainly in the elderly category. Therefore, the gamification tactics applied to better the adherence of people with heart failure should be focused on tactics specifically for elderly. Martinho et al. (2020) found in their systemic review of gamification techniques for elderly care that there are four prominently used.

1. **Feedback**
   Provide feedback regarding user performance in forms such as sound, message, touch, and images.
2. **Sense of progression and improvement**
   Levels, increased difficulty after user performance improves, unlocking features, trophies, and virtual rewards.
3. **Points/scores**
4. **Social Interaction**
   Exchange and promote social interactions between participants.

In Table 1 these gamification elements and their corresponding the eight core drives of Chou (2019) are displayed.

Gamification Element	**Core Drive**
Feedback	3. Empowerment of Creativity & Feedback
Sense of Progression and Improvement	2. Development & Accomplishment
Points/Scores	2. Development & Accomplishment
Social Interaction	5. Social Influence & Relatedness

Table 1: How the most prominent gamification elements for elderly is linked to Chou (2019) 8 core drives.

1.6 Project Plan

Background

The ultimate goal of this phase was to gain knowledge to map out where the gap in current research lies. The domain background was gathered by studying the medical and technical background of the domain. To fully exploit the potential of the MVP, it was important to study how it works thoroughly. Then it is possible to identify areas for change and improvement. The background studied in this phase is presented in Sections 1.1 through 1.5 and Chapter 2.

Requirement Analysis

A requirement analysis was done based on the information gathered. The analysis included identifying stakeholders, creating user stories, and defining functional and non-functional requirements. These requirements are the foundation for the product of this thesis. The analysis as a whole is described in Chapter 3.

Development

The development prominently includes integrating and creating new functionality for the MedRem application. During this phase of the project, the focus will be to work as agile as possible. The initial thoughts found in the prior Research phase are not definite. They can change based on discoveries while developing and with further research. The changes have to be reflected in the requirements as well. Therefore, the Background, Requirement Analysis, and Development ultimately happened somewhat alongside each other. The documentation of this phase is described in Chapter 4.

Testing

Outside of the testing of each iteration done by the developer, two test intervals were executed. After the first test interval was executed, the developer implemented changes based on the feedback from the tester. When these were implemented, the second test interval started to gain insights into the changes as well. These tests and how they are executed are described in Chapter 5.

Evaluation and Conclusion

When the tests were conducted, the product was evaluated, both in terms of whether the requirements were met and the insights gained from the testing. Consequently, a conclusion of the work was presented. The evaluation can be read in Chapter 5, and the conclusion in Chapter 6.

2 Technical background

2.1 Apple Ecosystem

2.1.1 Apple Watch

The Apple Watch has been a great success for Apple. The Apple Watch Series 4 launched in late September 2018 and managed within that last quarter to become the most sold smartwatch of the year (Reisinger, 2019). The Apple Watch integrates with other Apple products, such as the iPhone and iPad. This makes it simple to share data collected from the Apple Watch with your other devices. The Apple Watch can also trigger actions on your other devices by answering messages, switching which song you are listening to, answering phone calls, and even paying for your groceries at the store (Apple Inc., 2022k). In addition, the Apple Watch Series 4 uses PPG to measure HR, as well as an ECG feature that has received FDA approval as a medical device (Apple Inc., 2022f).

2.1.2 Xcode

In Xcode you can build, test, and submit iOS applications within Apple's integrated development environment (IDE) (Apple Inc., 2022l). Xcode allows you to manage your entire development workflow. Xcode is an IDE specifically designed to develop applications on Apple platforms, including the iPhone, iPad, and Apple Watch. There are numerous tools in Xcode that aid in the development process. You can test your applications using the simulator and upload builds for further testing. In newer versions, Xcode integrates with version control software to make the development process even smoother (Apple Inc., 2022m).

2.1.3 Swift

For the development of iOS applications, Apple has provided its powerful programming language, Swift. Apple ensures that Swift is modern, designed for safety, fast and powerful, and a great first language. Apple made Swift with the latest research on programming languages and decades of experience building Apple platforms in mind. When using Swift in Xcode, Xcode gives tips and helpful warnings when, for instance, a variable could be a constant. The combination of the two makes the development as smooth as possible for the developer (Apple Inc., 2022h).

2.1.4 TestFlight

To be able to distribute your application before it is published to the App Store, Apple provides an easy way to distribute and test it through TestFlight (Apple Inc., 2022i). With TestFlight, you can invite up to 10,000 testers to test your application. Getting your application ready for testing only requires two steps. First, upload a beta build to App Store Connect. Then distribute the application either externally using email or public links or internally by assigning Apple accounts to groups and roles of your choice. When the testers are testing the application, they can send feedback directly from the application by taking a screenshot. They can also give feedback and more context if the app crashes, for instance, what they were doing right before it crashed, aiding the developer in finding and fixing the reason for the crash.

2.2 Frameworks

2.2.1 HealthKit

With HealthKit, you get a central repository for health and fitness data on your iPhone and Apple Watch. Using HealthKit, you can collect and store health and fitness data, analyze and visualize the data, and enable social interactions. In addition, the framework is designed to make sharing data between applications and syncing data between devices straightforward. This enables developers to create applications using health data, given that the user permits the application to gain access to them. For instance, when syncing up with an Apple Watch, HealthKit facilitates the use of measures of HR (Apple Inc., 2022d).

2.2.2 ResearchKit and CareKit

ResearchKit is an open-source framework for creating applications for medical research. ResearchKit also works seamlessly with HealthKit, which gives even more insights into the patients' health data. In addition, ResearchKit is specifically used for facilitating patient consent, dynamic, active tasks, and surveys (Apple Inc., 2022g).

CareKit is also an open-source framework for developing apps, but what differentiates CareKit from ResearchKit is that CareKit helps users better understand and manage their health. So, while HealthKit offers developers the data to create meaningful health apps, CareKit aims to provide better care and improve patient outcomes actively. CareKit ensures this by allowing the user to track and share their daily progress from their devices so that care providers can connect with patients and gather insights to provide better care (Apple Inc., 2022b).

ResearchKit and CareKit were, for instance, used by Spaulding et al. (2019) as a means to prevent re-admissions after hospitalization for acute myocardial infarction. The application created was called Corrie. The patients are either enrolled with their own devices or provided with an iPhone and Apple Watch. In this case, ResearchKit was used to get electronic consent and registration. CareKit was utilized to monitor and control the patient's medications and vital signs and for educational purposes. Corrie also uses CareKit's opportunity to connect with care providers to schedule and track follow-up appointments and upload health information. In Corrie, there was no monitoring of real-time data by the study or patient's clinical team, it was solely a self-management tool for patients.

The ResearchKit framework is designed to make it easy for developers and researchers to create apps that facilitate research. With ResearchKit, you can connect the sensors and capabilities of the iPhone to measure and

Framework	Main functionality
HealthKit	Central repository for health and fitness data, like daily steps, calorie use, and heart rate.
CareKit	Framework for building iOS apps to aid patients in following their care plans and facilitates information sharing with their care teams.
ResearchKit	Framework to create apps specifically for medical research, and facilitates both collecting of data as well as consent.

Table 2: Definition of each of the three health related kits from Apple.

record the user's behaviour, together with modules that make creating surveys, getting consent, and creating tasks simple (Inc., n.d.).

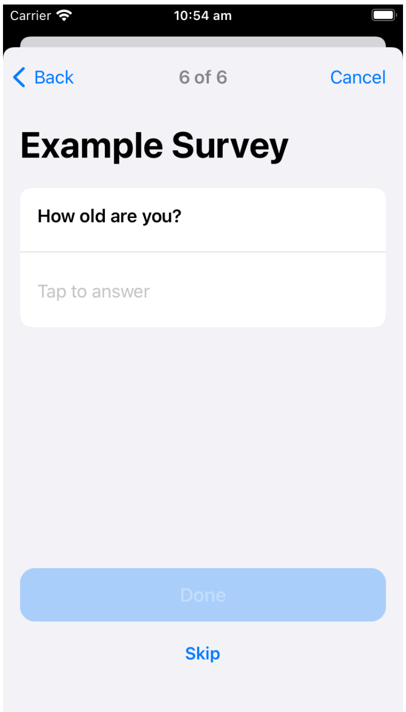

Figure 2: The UI of a ResearchKit-Survey Question Step with the code from 1

2.2.3 ResearchKit Surveys

Surveys are a central part of the ResearchKit framework and enable the researchers to collect data and permissions from the users throughout their research study. A survey consists of a survey task with a collection of step objects. The framework makes it simple to create a survey that gets data of different types as well as asks for permissions (Inc., n.d.).

1. First, since the survey consists of one or more steps, these need to be made. There are three different types of steps, instruction-step, single-question step, and form-step.

    - An instruction step, as the name suggests, explains or instructs the user. The instruction step includes a title, text, detailed text, and an image. This type of step does not collect data but yields an empty result that records how long the instruction was on the screen.
    - A single-question step presents a single question with a short title and longer descriptive text. It is also possible to configure the answer's format, like numeric or text, and select boundaries to this format. A question step will yield how long the user had the question on the screen and the user's answer. Figure 2, illustrates an example of a question step and Listing 1 includes the code to generate it.
    - A form step is used when you have several related questions, then you can present all your questions on one page. A form step supports all the same formats as the question steps but can contain multiple items with each its answer format. The result of a form step works in the same manner as a question step, but with one question result for each form item.

```
//Setting the style of the question to default Numeric Answer
let style = ORKNumericAnswerStyle(rawValue: 1)

//Setting the format to have to unit years, and to have the
//boundaries between 18 and 90 years of age
let format = ORKNumericAnswerFormat(style: style!,
                                    unit: "years",
                                    minimum: 18,
                                    maximum: 90)

//Creating the Question Step and setting the id, title,
//question and the format of the answer.
let questionStep = ORKQuestionStep(identifier: "example.yearsOld",
                                   title: "Example Survey",
                                   question: "How old are you?",
                                   answer: format)

//Making the question optional to answer to be able to
//move on in the survey
questionStep.isOptional = true
```

Listing 1: Creating a ResearchKit-Survey Question Step

2. Then, a Survey Task needs to be created. This is an object that contains all the steps. This can either be an Ordered Task or Navigable

Ordered Task. The former is a task where all steps appear in a specific order, while with the latter, the order of the tasks can change and branch out based on the answer to a previous task. When the task object is created, it must be presented. How this is programmatically done is described in Listing 2.

```
let taskViewController = ORKTaskViewController(task: task,
                                               taskRun: nil)

taskViewController.delegate = self

present(taskViewController, anitmated: true, completion)
```

Listing 2: Presenting a ResearchKit-Survey Step

3. Finally, the results of the survey need to be retrieved. The result of a task can be redeemed as soon as the task is finished. How the result is retrieved programmatically is described in Listing 3.

```
func taskViewController(
                _ taskViewController: ORKTaskViewController,
                didFinishWith reason: ORKTaskViewControllerFinishReason,
                error: Error?) {
    let taskResult = taskViewController.taskResult

    //You could do something with the result here.
    userData.doSomething(result: taskResult)

    // Then, dismiss the task view taskViewController
    dismiss(animated: true, completion: nil)
}
```

Listing 3: Code to handle the result of a finished ResearchKit-Survey Step

2.2.4 CareKit

While ResearchKit was developed first and foremost for researchers, CareKit was primarily made for patients and health personnel. CareKit aims to make it simple to create applications that facilitate users in understanding and managing their health through dynamic care plans, tracking symptoms, and creating a channel to connect to their health team. The motivation behind CareKit was to improve the support and care of patients outside of the hospital (Apple Inc., 2022c).
CareKit is composed of three packages:

- **CareKit**
  Connects the CareKitUI and CareKitStore packages through view controllers, where they provide synchronization between the store and the views.
- **CareKitUI**
  Delivers the views used in the framework.

- **CareKitStore**
  Stores patient data, and provides fast, secure, on-device storage.

The CareKit framework offers tools for tracking the treatment and related symptoms. The patients' treatment and progress data is stored in the CareKitStore and visualized via the CareKitUI framework. Therefore, a view controller is needed to synchronize these two entities. The view controller is a full-screen view that executes queries to get data from the store and display it. Through the view controller, CareKit ensures synchronization between changes in the data and the data shown to the user. A store manager facilitates the synchronization. If any change occurs to the store's values, the store manager notifies the view controllers of this change. The view controllers will then update the views according to the modification. There are two types of view controllers, one that keeps track of tasks for each day and one that displays health contacts from the store.

The following three steps need to be implemented to set up a view controller to track a patient's progress.

1. A store and a store manager need to be created. The store manager is then attached to the view controller.

2. Each of the tracked medications or actions needs a defined corresponding task. A schedule needs to be defined for each task. The schedule defines the occurrences of a task. The task is then added to the store by the store manager.

3. To connect the tasks in the store to the corresponding user interface, the view controller then fetches the tasks from the store manager and creates the CareKitUI-views for said tasks. When the patient finishes a task, the store manager will record and store this information.

CareKit also provides views to visualize the completion of tasks in the current week. This is done through a chart view that can be customized and styled in multiple ways. In addition, these charts can include and compare one or more tasks.

2.2.5 Integrating ResearchKit with CareKit

CareKit and ResearchKit are designed to integrate easily with each other. This gives the possibility of introducing surveys as a task in the view controller. There are three steps described by Apple to do this (Apple Inc., 2022c). Which are presented in the following. In Figure 3 the flow of the integration is illustrated.

1. First, the task view controller must be subclassed to control the flow and present a ResearchKit Survey instead of a CareKit task. The code behind this step is described in Listing 4.

```
// 1. Subclass a task view controller to customize
//the control flow and present a ResearchKit survey!
class SurveyViewController:
    OCKInstructionsTaskViewController,
```

```
        ORKTaskViewControllerDelegate {

}
```
Listing 4: The first step of integrating ResearchKit with CareKit

2. The method called when a user taps the task has to be altered. The user tap can represent two different actions. The code for this step can be viewed in Listing 5.

    a) The task is marked incomplete, then it will fall back to the super class behaviour, return and delete the outcome.

    b) The task is marked complete, then the view controller marks the task as complete, creates a ResearchKit task, and presents the task to the user.

```
// 2. This method is called when the use taps the button!
override func taskView(_ taskView: UIView & OCKTaskDisplayable,
                       didCompleteEvent isComplete: Bool,
                       at indexPath: IndexPath,
                       sender: Any?) {

    // 2a. If the task was marked incomplete, fall back
    //on the super class's default behavior or deleting
    //the outcome.
    if !isComplete {
        super.taskView(taskView,
                       didCompleteEvent: isComplete,
                        at: indexPath,
                        sender: sender)
        return
    }

    // 2b. If the user attempted to mark the task
    // complete, display a ResearchKit survey.
    let style = ORKNumericAnswerStyle(rawValue: 1)
    let format = ORKNumericAnswerFormat(style: style!,
                                        unit: "years",
                                        minimum: 18,
                                        maximum: 90)
    let questionStep = ORKQuestionStep(identifier: "example.yearsOld",
                                       title: "Example Survey",
                                       question: "How old are you?",
                                       answer: format)
    questionStep.isOptional = true

    let surveyTask = ORKOrderedTask(identifier: "survey",
                                    steps: [questionStep])

    let surveyViewController = ORKTaskViewController(
```

```
                                                task: surveyTask,
                                                taskRun: nil)

    surveyViewController.delegate = self

    // 2c. Present the survey to the user
    present(surveyViewController,
            animated: true,
            completion: nil)
}
```

Listing 5: The second step of integrating ResearchKit with CareKit

3. The last step is to change the method that is called when the user finishes the survey. If there is no result after completion, the task will be set as incomplete, and the method will return. Otherwise, the result is retrieved from the survey and the result is saved to the CareKitStore.

```
// 3. This method will be called when the user
//completes the survey.
func taskViewController(
                _ taskViewController: ORKTaskViewController,
                didFinishWith reason:
                ORKTaskViewControllerFinishReason,
                error: Error?) {
    taskViewController.dismiss(animated: true,
                               completion: nil)
    guard reason == .completed else {
        taskView.completionButton.isSelected = false
        return
    }

    // 3a. Retrieve the result from the ResearchKit survey
    let survey = taskViewController
                               .result
                               .results!
                               .first(where: {
                                   $0.identifier == "example.yearsOld"
                               }) as! ORKStepResult

    let questionResult = survey
                .results!
                .first as! ORKScaleQuestionResult

    let answer = Int(truncating: painResult.scaleAnswer!)

    // 3b. Save the result into CareKit's store
    controller.appendOutcomeValue(withType: answer,
                               at: IndexPath(item: 0,
                                             section: 0),
```

```
                                        completion: nil)
}
```
Listing 6: The third step of integrating ResearchKit with CareKit

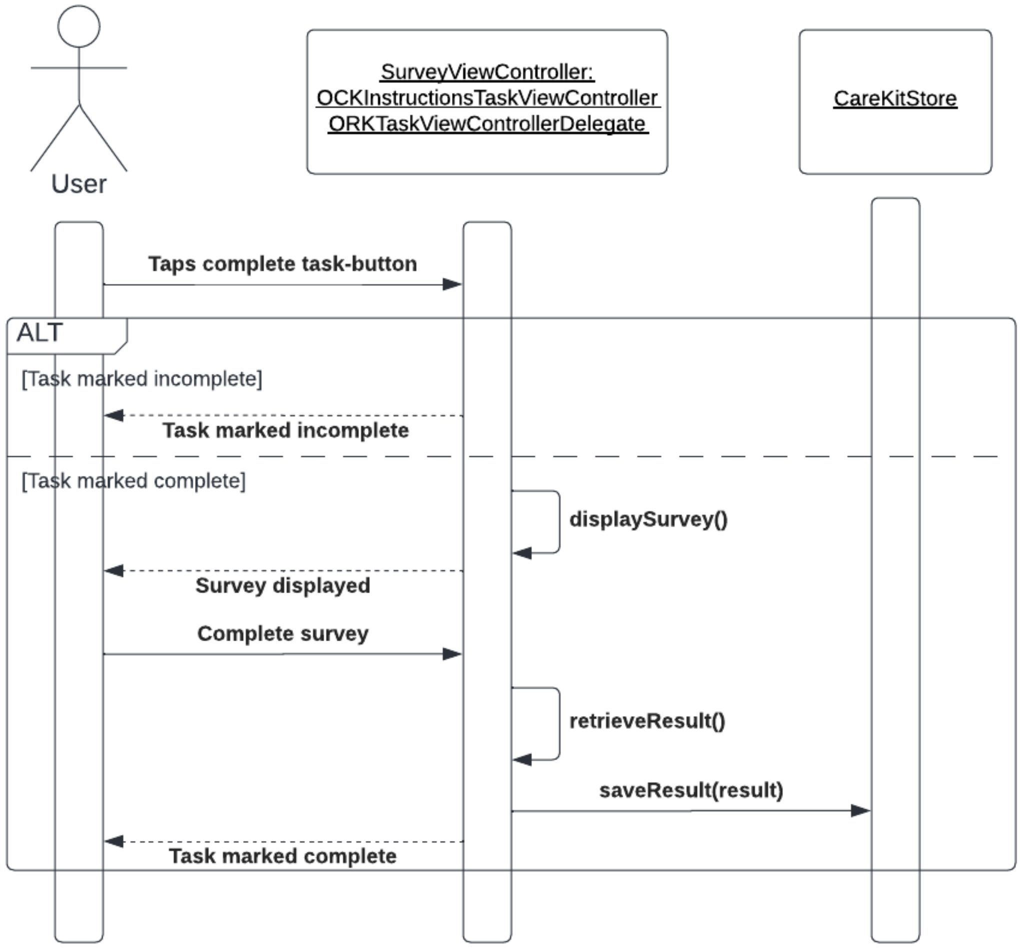

Figure 3: Sequence diagram illustrating how ResearchKit Surveys are integrated with CareKit.

2.2.6 Notifications

Within the Apple platform, there are two ways to generate notifications. It can be done locally from the application or remotely from a server. Local notifications require notification content and a condition specifying a trigger for when the notification should be delivered, like time or location. An essential aspect of notifications is that all applications have to ask for permission from the user to give notifications (Apple Inc., 2022a). How to programmatically ask for permission is given in Listing 7.

```swift
let center = UNUserNotificationCenter.current()
center.requestAuthorization(options:
                            [.alert, .sound, .badge]) {
                            granted, error in

    if let error = error {
        // Handle the error here.
    }

    // Enable or disable features based on the authorization.
}
```

Listing 7: Asking user for permission to give notifications

2.3 MedRem

The work of Jonathan Aanesen's master thesis resulted in four different applications, three of them for testing purposes. MedRem was the final result with all tested functionalities of the three test apps included.
The MedRem application fulfills three functional requirements:

1. Fetching Heart Rate
2. Dispatching Notifications
3. Running in the Background

Every time you unlock your phone, MedRem gets access to HR data from HealthKit. MedRem uses the HR data and determines if the user needs to be reminded or asked to take their medication. If given the notification, the user can dismiss it, press on it, or swipe it. If pressed on or swiped, they will be further given the question of whether they have taken their medication or not. Their answer to this question will update the values of the application to assess further if the user should get more notifications.

The premise of the application is to use the HR data from the Apple Watch and, based on this data, determine if the patient is adherent or not. If the user has an elevated resting HR, they will get notifications to take their medication. These notifications can be redirected to create reminders in Apple's Reminder application, or the user can ask to be reminded some hours later. Therefore, the application does not need much direct interaction with the user to fulfill its functionality and purpose.

An important aspect of this master thesis was to understand and research the MVP to understand how it could be altered to fit the new functional requirements. In the following, the MVP will be described.

In the MVP, the Apple Watch reads the HR of the user throughout the day. Information about the resting HR of the user gets sent to HealthKit, which in turn notifies the MedRem application whenever the resting HR changes. The MedRem then acts on these changes in the following manner. If the updated rested HR is higher than the set boundary HR, the application will warn the user that their HR indicates that they have not taken their medication. If the user responds to this notification that they have taken their medication, the boundary HR value will increase to adjust.

2.4 Problem Statement

HF is a growing problem, and beta-blocker medication has improved patients' prognoses. Simultaneously, a third of HF patients are non-adherent to their medication. The consequences for the patients are re-hospitalization or even fatal. When patients do not take their medication, their resting HR rises, and the MVP Aanesen (2021) created took advantage of this fact. The MVP reminds the patients to take their medication when their resting HR rises. This thesis aims to add to the MVP without removing the already developed functionality. Two prominent causes of non-adherence are forgetfulness and lack of motivation. Gamification is currently used as a tactic to engage and motivate users. In applications such as Snapchat and Waze, they use it to keep users coming back and engaging with the application. Gamification is also used to engage and motivate patients to better their health in various ways. The gamification elements most commonly used for the elderly are feedback, levels, points, and scores.

CareKit and ResearchKit are frameworks that offer components to facilitate health care in mobile applications. CareKit primarily facilitates gathering information on medication use and offers multiple methods for visualizing these. ResearchKit makes it simple to create surveys and ask for permissions. CareKit and ResearchKit require users to interact with the applications. Consequently, introducing them will change the application considerably from being primarily a background application. When demanding interaction from the user, the users need to be motivated to engage with the application regularly. In turn, medication adherence is also an achievement that requires continuous engagement. Consequently, the utilization of gamification elements similar to those of Snapchat and Waze may positively affect medication adherence.

This thesis focuses on two things. Firstly, the integration of ResearchKit and CareKit into the MVP. Secondly, the integration of gamification elements through the artefacts of these frameworks. Because most HF patients are elderly, it made sense that this should be the target group of this application. The thesis will evaluate how we can use gamification to aid medication adherence for elderly patients. The connection between gamification, medication adherence, and elderly is an area that currently lacks research. Therefore, in this thesis, I will link methods of gamification for adherence and elderly together. After identifying a set of strategies, the ones found feasible for this project will be introduced to the MVP.

Based on this the research question that the thesis seeks to answer is:

How can gamification of a smartphone application affect the medication adherence of elderly patients?

3 Requirement Analysis

3.1 Requirement Specification Process

When working out the requirements, the developer must consider the users. A company rarely succeeds with an idea without a basis in the user problems and user interaction. Of course, there are exceptions like Facebook, but this is a rarity. Sommerville (2020) lists three factors that drive the design of software products:

1. **Business and consumer needs that are not met by current products.**
   Heart failure patients need a product that successfully helps them maintain their medication adherence.

2. **Dissatisfaction with existing business or consumer software products.**
   Based on highly occurring non-adherence described in Section 1.2, the current products are believed to be not satisfactory to the patients.

3. **Changes in technology that make completely new types of products possible.**
   CareKit and ResearchKit offer new possibilities to monitor and aid patients in their healthcare journeys.

In the process of defining requirements, Sommerville (2020) suggests creating personas of the users. Then, define scenarios for these personas and construct user stories based on these scenarios. This suggestion is based on the possibility of conducting informal user consultations. However, within the limitations of this thesis, access to users of the target group was minimal. Therefore, the steps of personas and scenarios were omitted. The user stories were instead based on research on medicine adherence, gamification for the elderly, and gamification for adherence. Which are described in Section 1.

3.2 Introducing Gamification

This thesis aims to further develop an MVP, MedRem, to examine further how an application can aid users of the medication beta-blocker to achieve adherence. The MVP focused on working in the background as a control mechanism to determine the user's adherence. The new application aims to incorporate Apple frameworks to promote user involvement and then use gamification elements to motivate and counteract the forgetfulness of the users in their medication journey.

The functional requirements of the previous MedRem application were as follows.

1. MedRem has to be able to check the user's resting heart rate to determine if the user has taken their medicine or not.

2. MedRem must deliver different types of notifications to the users to warn them if the recorded resting-HR values exceed the pre-defined set-point and especially interact with the users.

3. MedRem has to run in the background and do the two previous tasks while doing so. This is required such that MedRem is maintenance-free, and the users do not have to interact with it unless they receive a warning or are re-configuring the app.

It was decided that the next iteration of the MedRem application should implement ResearchKit and CareKit. The structure and features of these frameworks have user involvement as a critical aspect. CareKit offers tasks, and graphs of the execution of said tasks, as a tool to help the patient keep track of their symptoms and medication. ResearchKit allows in-app surveys, permissions, and consent. Therefore, to use these frameworks, MedRem, an application that required close to no interaction with the user, had to be structurally changed. The application now had to require and reward interaction. From this, the idea of using elements of gamification arose. The next iteration of the application partly contradicts the third functional requirement, and thus it needs to be edited. Since the goal is to include the first and second requirements, the demand for the application to run in the background should be met. By removing the "and the users do not have to interact with it unless they receive a warning or are re-configuring the app.", the demand fits.

The further development of the MVP aims to implement elements of gamification. The gamification elements introduce new requirements to the application. Gamification is a broad concept that includes many sectors, elements, and purposes. Therefore, it was essential to find the elements that best resonated with the problem we were trying to solve. The problem addressed in this thesis then presents three concerns.

- How gamification works in general
- How gamification work in regards to elderly people
- How gamification for adherence work

In Chapter 1 these three areas of concern were presented. When assessing which requirements would be applied to the MedRem application, these concerns were set as the foundation for the definition of new requirements. The most important concern was gamification for elderly. If the game elements were in line with medication adherence and the core drives of gamification but did not appeal to elderly, the effects would be close to none. Therefore, starting with the gamification elements for elderly made sense. Then these were compared with the gamification elements most commonly used for adherence. In Table 3 the gamification elements for adherence and elderly are listed. There is significant overlap here. Thus the focus on feedback and points leading to achievements will be considered.

In both fields, there is an emphasis on social interaction between participants having a significant and positive effect. Regardless, this was not emphasized in the requirements due to the privacy issues introduced by such functionality. Sharing and storing medication and adherence data between users complicates development and creates higher demands regarding privacy and sharing and storing of data. Therefore, with the time, scope, and resources of this thesis, the social aspect will not be emphasized.

Gamification For	Gamification Element
Elderly	Feedback Sense of progression and development Points/Scores Social Influence
Adherence	Points/Scores towards Leaderboards and accomplishment Feedback Social Bonding

Table 3: Gamification elements prominent in gamification for adherence and elderly

In inspiration of Snapchat's Snapstreak remarkable success in keeping users engaged in using the application every single day, the way of counting points should be through this measure. After a given number of days, the adherent patient will be rewarded with badges and the possibility to reach levels. The feedback given to the user should also be in different forms. For the application to have an effect, the user must understand how the application works and how they should interact with it. There are also some settings to be set where there should be a care provider present. This establishes a need for an introduction to the application. The introduction has two concerns: the patients have to gain insights into the application, and the care provider must make sure that the correct information is registered.

1. MedRem has an onboarding that gives the user an introduction to the features of the application.
2. MedRem has to deliver notifications at a time that the user can set during the onboarding.
3. MedRem has to count the adherence of the user and give them feedback in the forms of a streak, a weekly summary, and a graph of medication taken that week.
4. The streak accomplished will grant different levels of badges and titles.
5. MedRem has to run in the background and do the two previous tasks while doing so. This is required such that MedRem is maintenance-free.

With these requirements, we can look into how this translates to the core drives of gamification. The 1st, 2nd, and 5th requirements do not directly or indirectly have any core drives behind them. Therefore, the focus will be on the remaining two. An overview of how each feature is connected to core drives is listed in Table 4. Further, an in-depth explanation of why is given in the following.

The streak feature of this application cannot be equated with the Snapstreak. In Snapchat, the streak represents an action between two users, and in this case, the streak represents an action that one user alone does. However, not all core drives are lost. Primarily, the streak will contribute to core drives 2, 4, and 8. The drive 8. Loss & Avoidance are achieved through the fear of losing the streak. It only takes one day without medication before the streak, and all the hard work is lost. 2. Development & Accomplishment is also a drive that becomes central, where the streak serves as a point system towards the accomplishment of adherence. Since the streak is just as hard to accomplish and maintain at all times, it does not offer an increasing level of difficulty and thus an enhanced challenge over time. But, the longer the streak is maintained, the bigger the loss of it. The streak can even contribute to the core drive of 4. Ownership & Possession, since the streak is a metric that the user has ownership over. Especially since it reflects something that is directly related to them, their ability to take their medication every day.

The feedback mechanisms that are in the application also tap into 4. Ownership & Possession because they refer directly to how the user has been doing the last week. The most definite core drive is 3. Empowerment of Creativity & Feedback. Even though the involvement of creativity from the user lacks these requirements, the trigger the feedback represents is for them to gain motivation and an overview of their medication regimen. Which, in turn, leads to an enhancement of the core drive 2. Development & Accomplishment.

Feature	Core Drive	Rationale
Streak		
	2. Development & Accomplishment	Streak serves as a points system towards the accomplishment of adherence
	4. Ownership & Possession	The streak reflects something directly related to them
	8. Loss & Avoidance	Fear of losing the streak
Feedback		
	3. Empowerment of Creativity & Feedback	Gains insight and motivation over their medication regiment
	4. Ownership & Possession	Refers to the users own adherence

Table 4: Core drives linked to the two gamification features

In the following chapter the new requirements, both functional and non-functional, will be further analyzed and defined.

3.3 Requirement Specification

This section will present the analysis of what new requirements should be implemented. First of all, the stakeholders will be defined, and then each of the stakeholders' user stories will be presented. Based on these functional and non-functional requirements, the further development of the MVP will be presented.

3.3.1 Stakeholders

1. **Patients**
   Is the primary user of the application. Where the application aids the user in bettering their health through medication adherence.

2. **Care providers**
   Is the secondary user of the application. They aid the patient in introducing and setting up the application. The application will both take some of their responsibility to follow up away and make their follow up-session simpler by providing information about the patient to them.

3. **Government**
   The Government does not have a direct role in the application but will benefit from positive results of the application. If more patients take their medication every day, the result would be fewer re-admissions. The costs of re-admissions are high, and thus the government has an interest in the decrease in such cases.

4. **Developers**
   The developers are stakeholders due to their responsibility to maintain and further develop this application.

5. **Friends and Family**
   While friends and family were not included as stakeholders in the MVP, the medicine process will be more visualized in the next version, making it easier to share.

3.3.2 User Stories

A user story is a detailed way of describing a single thing that a user wants from a software system (Sommerville, 2020). Sommerville (2020) presents this format of an user story:

As a <role>, I <want/need> to <do something>

If the intention behind the user story is not clear, Sommerville (2020) suggests supplementing the format:

As a <role>, I <want/need> to <do something> so that <reason>

Creating user stories is a part of planning an application, and it can help with coming up with alternative ideas to provide what the user wants. User stories can often be placed directly into the backlog in agile development. Therefore, user stories should focus on one single system feature. The user stories are also a tool to explain the purpose of each functionality (Sommerville, 2020). In the following, the user stories of the stakeholders are listed.

1. As a patient, I want to get a reminder when it is time to take my medication because I do not want to forget my medication.
2. As a patient, I want to get a visual overview of my medication adherence within the last week, to help me remember if I have taken my medication in the last few days.
3. As a patient, I want to get instructions on how to use the application, to be able to use the application in the intended manner.
4. As a patient, I want to get feedback on my medication adherence, to get more motivation to follow the medication plan that is set for me.
5. As a patient, I want my sensitive health information to be stored in a secure way.
6. As a care provider, I want to be able to set important settings together with the patient, to make sure that these are correct.
7. As a care provider, I want to go through important information with the patient, to make sure that they understand the application and avoid additional consultations.
8. As a care provider, I want to have an overview of the medication adherence over time, to be able to follow up with the patients better.
9. As a government, I want patients to stay adherent to medication, to avoid expensive re-admissions.
10. As a developer, I want an application that is easy to maintain and alter.
11. As a developer, I want to be able to set the boundary heart rate, to test parts of the application.

3.3.3 Functional requirements

The user stories defined in Section 3.3.2 can further be categorized into categories of requirements. Primarily, requirements are classified as functional or non-functional requirements. Sommerville (2016) defines functional requirements as "Statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations." The functional requirements define what the system should do, and they should ideally be complete and consistent. Where completeness refers to the fact that all services and information required by the user should be defined, completeness opposes contradictory requirements. (Sommerville, 2016).

A use case describes interactions between users and a system. A use case includes the actors involved in an interaction and specifies the type of interaction. A use case diagram illustrates the set of all use cases for a system. Here the actors are represented as stick figures, and each interaction is listed in a named ellipse (Sommerville, 2016). In the following, the use cases based on the functional requirements from Section 3.3.3 will be presented in a use case diagram.

Based on the user stories in Section 3.3.2, in the following, functional requirements will be listed along with reflecting use case diagrams for the different parts of the application.

Core Functionality

1. The application should have an overview of the medication taken within the current day
2. The application should have to go back and look at previous adherence to medication
3. The application should give the user a way of registering that they have taken their medication
4. The application should count the patient's adherence in the form of a streak
5. The application should use the resting heart rate data from the Apple Watch to control if the user has taken their medication

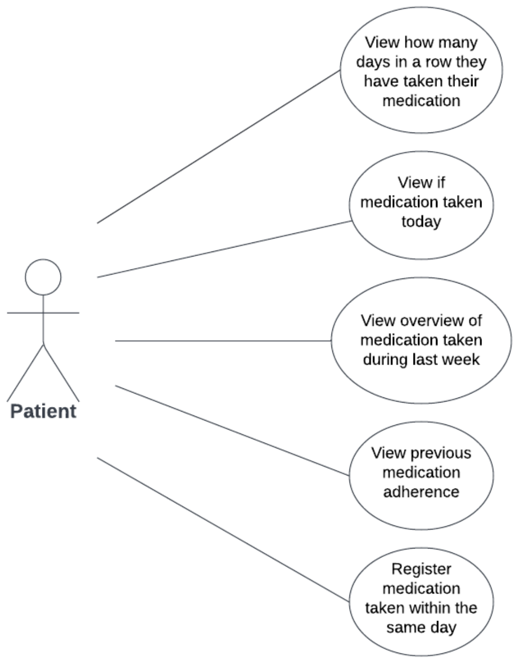

Figure 4: Use case diagram modeling the functional requirements of the core functionality of the application

Onboarding

1. The application should have an onboarding process

2. The application's onboarding process should provide instruction on how to use the application
3. The application's onboarding should provide information about how data is stored
4. The application's onboarding should include entering settings, these should include boundary heart rate and a time for a medication reminder notification
5. The application should give a notification to remind the user to take their medication at a predefined time every day

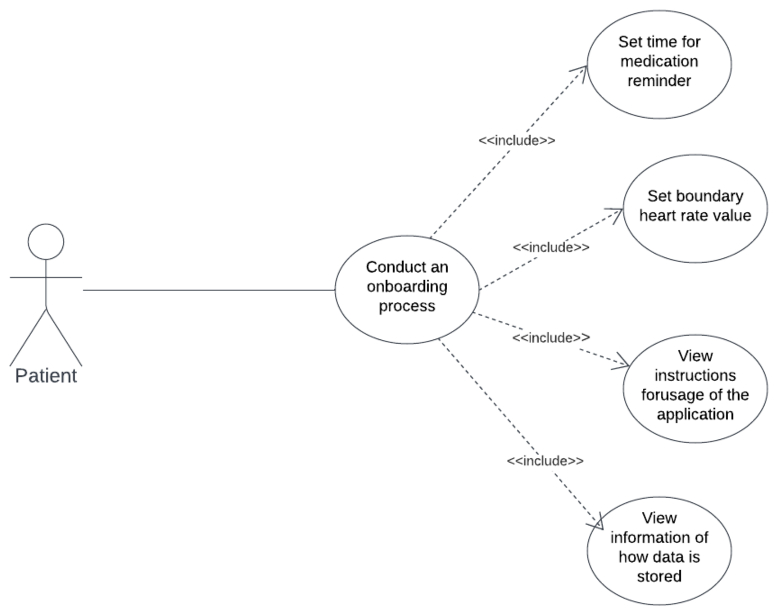

Figure 5: Use case diagram modelling the functional requirements of onboarding of the application

Feedback

1. The application should visualize the medication adherence of the user in the form of a graph
2. The application should give the user a weekly summary of their medication adherence
3. The application should give the user the possibility to share and compare their weekly feedback with others

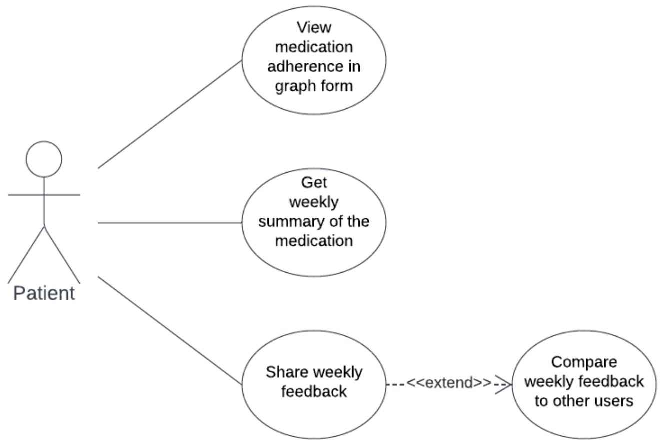

Figure 6: Use case diagram modeling the functional requirements of the feedback mechanisms of the application

3.3.4 Non-functional requirements

Non-functional requirements, on the other hand, are defined as "Constraints on the services or functions offered by the system ... Non-functional requirements often apply to the system as a whole rather than individual system features or services" (Sommerville, 2016). Thus, the requirements in this category are not directly connected to the services delivered by the system to its users. Instead, they specify or constrain the characteristics of the system. Since non-functional requirements include the entire system as a whole, the requirements are further categorized into product, organizational, and external requirements (Sommerville, 2016).

- **Product requirements**: Specifies the run-time behavior of the software, for instance how much memory the system uses.
- **Organizational requirements**: Requirements of the policies and procedures of the customer and developers organization. Included here is for instance what programming language is used, which tool has to be used for version control, and what development process should be used.
- **External requirements**: Defines requirements come from the external "world" that applies to the system. For instance laws and ethical requirements.

From these definitions the non-functional requirements of the application was defined, these are described in Table 5.

3.4 Apple Design Principles

An important aspect of creating an application is to ensure that the design neither weakens nor amplifies the functionality. To do so, some design

Category	Requirement
Product	The application must store sensitive data in a secure and encrypted manner.
	The application must support dynamic text sizes in the user interface
	The application should not need any further instructions or training than the onboarding for a user to use it.
	The application should be a one-page application
Organizational	The application should follow the Apple Design Principles, listed in section 3.4
	The application should follow Apples best practices for Inclusive Design, listed in Section 3.5
	The application must be written in Swift
	The application must integrate the ResearchKit and CareKit frameworks
External	General Data Protection Regulation (GDPR) laws and regulations.

Table 5: An overview of the applications non-functional requirements, divided into three categories

principles need to be kept in mind. Therefore, when developing an iOS application using frameworks with an Apple-developed UI, it makes sense to manifest the primary needs and measures of the design to Apple's own Design Principles.

- **Aesthetic Integrity**
  That the user interface/design of the application is in line with its functionality. An application with a serious purpose should not include expressive graphics and animations, whereas this would be better suited for a game application.

- **Consistency**
  Suggests that the application implements interface elements, such as icons, text style, and terminology, in a well-known and uniform way. For example, an icon should not have two different meanings. The application should behave as people would expect.

- **Direct Manipulation**
  The user should be able to see the immediate, visible results of their actions. Their gestures affect the on-screen content.

- **Feedback**
  Provide visual feedback to every user action. Feedback refers to, for instance, small events like an element being highlighted when tapped and progress indicators to indicate the status of a process.

- **Metaphors**
  Using objects and actions that are metaphors for familiar experiences makes people more quickly learn. An example of this is flicking through the pages of books, newspapers, and magazines.
- **User Control**
  An app should make the user feel like they are in control and strike a balance between enabling users and avoiding unwanted outcomes. This can be done by keeping interactive elements familiar and predictable, confirming destructive actions, and making it easy to cancel operations.

(Apple Inc., 2022j)

3.5 Inclusive Design

Inclusive design is all about giving the most users the possibility of using an application. MedRem's target group, elderly, has other needs than the rest of the population, and a focus on inclusive design should be emphasized when developing this application. Apple defines three best practices for creating an inclusive application (Apple Inc., 2022e).

- **Design with Accessibility in Mind**
  The two characteristics to keep in mind are **Simplicity** and **Perceivability**. Simplicity enables familiar, consistent interactions to make more complex tasks simpler to perform. Perceivability ensures that content can be perceived when people use either sight, hearing, or touch.
- **Support Personalization**
  The application should support the accessibility features people use to personalize and interact with their devices. These include preferences such as bold text, larger text, and increased contrast.
- **Audit and Test Your App for Accessibility**
  Test to make sure that all tasks in the application can be done no matter how they interact with the device. This can easily be done with Apple's accessibility features turned on. In particular, make sure that critical flows of the application can be executed without a problem.

4 Design and Development

4.1 Agile Development Process

A key aspect of working agile is to develop iteratively, and after each iteration, there is a working system with new functionality. Therefore, in the beginning, the application will only offer a subset of the requirements. However, by providing an end-to-end user experience, it is simple to test it and provide feedback. Furthermore, there is an emphasis on agile to deliver early and often. The deliveries ensure that the development is going in the right direction. There would be no way back if there were only one delivery at the end of the development, and the customer and users were unhappy with the outcome. Consequently, by providing many opportunities for change and feedback, the result is more likely to be successful (Meyer, 2014). The application developed in this thesis has no success if the users' needs are not satisfied. Therefore, this principle has been a central part of development. This approach also facilitates testing, which is described in Section 5.1.

When developing iteratively, the first concern is to define what an iteration is for this project. Because it is hard to estimate how much time learning and developing with the frameworks would take, it made sense not to have time-boxed iterations, such as one or two-week-long iterations (Meyer, 2014). Still, this thesis has a deadline and there needed to be some time limits to the development to deliver a product. Therefore, the deadline was set at the year's turn. Then at least some of the new functionality should be implemented for testing. Next, after the testing is done, changes and new functionality should be implemented to be tested in a second testing interval.

4.2 Development

This section will present how the application was designed and developed based on the requirements from the previous chapter.

In short, the process of development had these steps.

1. Figure out how ResearchKit and CareKit works separately
2. Understand how ResearchKit and CareKit can be integrated one another
3. Integrate said frameworks with the MVP
4. Integrate the new functionality within the use of these frameworks

4.2.1 Studying ResearchKit and CareKit

In the first step, the documentation of the two frameworks was the only source of knowledge. Although CareKit and ResearchKit are well-designed frameworks, their use is limited. Therefore, it is difficult to find information and tutorials from other developers. Usually, when you stumble upon an edge case that causes a strange error, someone else has experienced the same thing. This was not always the case, with CareKit and ResearchKit not being widely used. Therefore, troubleshooting to figure out what was causing the errors and behaviour was challenging. While the documentation of ResearchKit and CareKit is thorough, the documentation only shows a few examples of uses of the frameworks. These do not always fit entirely with the requirements. However, because these early troubleshooting issues took so much time, I decided to keep more of the application similar to the examples and tutorials that Apple provided to minimize the danger of errors setting back development.

Since ResearchKit and CareKit are additional frameworks, they need to be imported into the development environment. It is common practice to include the stable version of a framework. As the name suggests, this ensures that the framework does not produce bugs and uncertainties. Therefore the stable version was chosen at first. However, after some weeks of development, it became apparent that these stable versions were missing a lot of crucial elements that the examples and tutorials were showing. Thus, the frameworks were changed to the main branch. Consequently, the errors that had been difficult to troubleshoot disappeared.

When getting to know the ResearchKit framework, the only source of knowledge was their documentation. There were two documentations. GitHub provided a short documentation with examples using both Swift and Objective-C (Apple Inc., 2022g). Comprehensive documentation could be found on the ResearchKit web page, but only in the language Objective-C. Objective-C was the standard development language for iOS applications before Apple created Swift. To translate the code examples into Swift was challenging and demanded a lot of time and effort.

The documentation of the two frameworks was beneficial when figuring out how ResearchKit and CareKit can work together. How it was integrated is described in Section 2.2.5.

How integrating the frameworks and introducing new functionality was done is described in Section 4.3 and 4.4.

4.3 Integration with the MVP

After figuring out how and what parts of ResearchKit and CareKit would be included in MedRem, the next step was integrating them with the MVP. The application had to be modified to meet both the new and existing requirements during this process. These requirements are listed in 3.2.

Since the MVP is almost solely a background application, all functionality could be kept, and further development could be done almost separately without altering the previous code. The application's user interface was the most vital piece that needed to be changed. The MVP's user interface can be viewed in Figure 7. The user interface of the MVP offers the following functionality.

1. Shows the boundary heart rate.
2. Shows the current resting heart rate.
3. Shows the average resting heart rate.

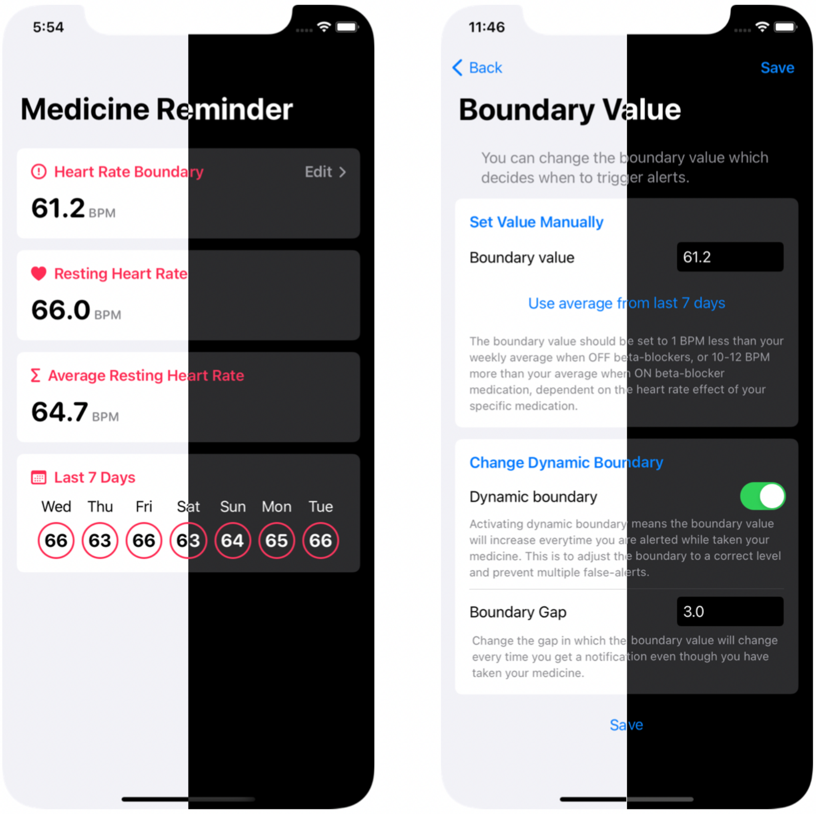

Figure 7: The user interface of the MedRem MVP

4. Shows how adherent the patient has been the last seven days based on their resting heart rate.
5. Change the boundary heart rate.
6. Change boundary heart rate gap.

These views could have been included in either the application's landing page or made a part of the application in an insights page. Unfortunately, this was made difficult due to the implementation of the CareKit view controller. The view controller only accepts views of type UIViews, where the views from the MVP are of type View. In this iteration of the MVP, the resting heart rate and boundary heart rate are no longer a key focus for the user but an additional background functionality. Therefore it was decided that the work to conform these views to the right type would have too little gain compared to the time it would take. Therefore, the three first functionalities were not kept in the new iteration of the application. Furthermore, because the view controller visualized adherence satisfactorily, the adherence visualization (4) was removed. As well, the streak gives an additional indication of adherence, as displayed in Figure 9. Finally, changing the boundary heart rate was switched with the possibility of setting the boundary heart rate in the onboarding process presented in Section 4.5. This removed the possibility of setting it more than once and the possibility of changing the boundary heart rate gap. Since this functionality was applied mainly for testing purposes in the MVP, it was thought that it was not needed to be included in the new iteration. If the application were to be tested, there would only be a need to set the boundary HR upon start.

4.4 Implementing New Functionality

After realizing that the MVP's user interface entities needed to be integrated differently, a new application interface was created. In addition, the new functionality had to be integrated. How this is done and connected to the different categories of the application is described in the following sections. The categories include Core Functionality, Streak, Feedback, and Onboarding. Figure 8 shows how the categories of functionality, excluding onboarding, are tied to the application's user interface. The onboarding user interface is further displayed and explained in Section 4.5.

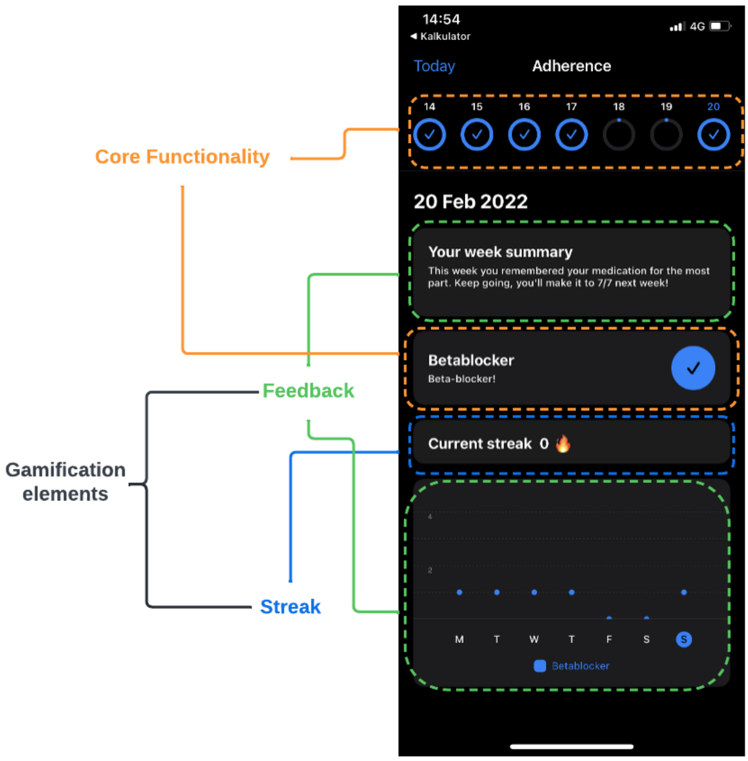

Figure 8: How the requirements of different categories are connected to parts of the application.

4.4.1 Core Functionality

This part of the application should apply the functional requirements for Core Functionality that are listed in Section 3.3.3. Since the core functionality was shifting to encourage and need user involvement the landing page of the application needed some drastic changes. The new landing page was to give the user information and insight into their medication adherence.

In the CareKit framework, a view controller shows and tracks the execution of tasks in a calendar format. This was deemed to be a good fit to satisfy the application's requirements. How such a view controller looks is displayed in Figure 9. The calendar at the top gives the user an overview of their adherence within the current week. If the user wants, they can scroll back on this calendar and view the adherence of previous weeks. A task is included in the view controller for the user to register that they have taken their medication. This task synchronizes with the calendar at the top. To urge the user to interact with the application daily, the user should not be able to go back in time and register that they have taken their medication in the past. Therefore, the view controller will only show the task on the current day. Through these visualizations and possible actions, the view controller fulfills all the functional requirements for core functionality. To test the task's behavior over time, the application had to be used for several days to see how the data structure changed. This includes the order in which the results were kept and how the dates were stored and updated for each result. This type of testing takes time to execute because of the apparent need for multiple days to pass and to analyze and compare the data between these days.

Figure 9 displays how the whole user interface of the view controller synchronizes, in the calendar display of the week, in the insight graph, and in the task itself.

Flow

The flow of the application, especially on the background fetching is for the most part the same as the MVP. Therefore this will not be described further in this thesis.

1. User opens the app
2. The application fetches outcomes of the onboarding and checks if this has been successful or not.
    a) If not successful it will follow the flow given in Section 4.5 and then continues step 2 again after completing the onboarding.
3. The application then checks if the date displayed in the view controller is the current day. If not it will show an empty view controller.
4. The application checks if the date displayed is a Sunday.
    (a) If true, a view with feedback on the medication adherence over the past week is displayed. The flow of how this is done is described in Section 4.4.3

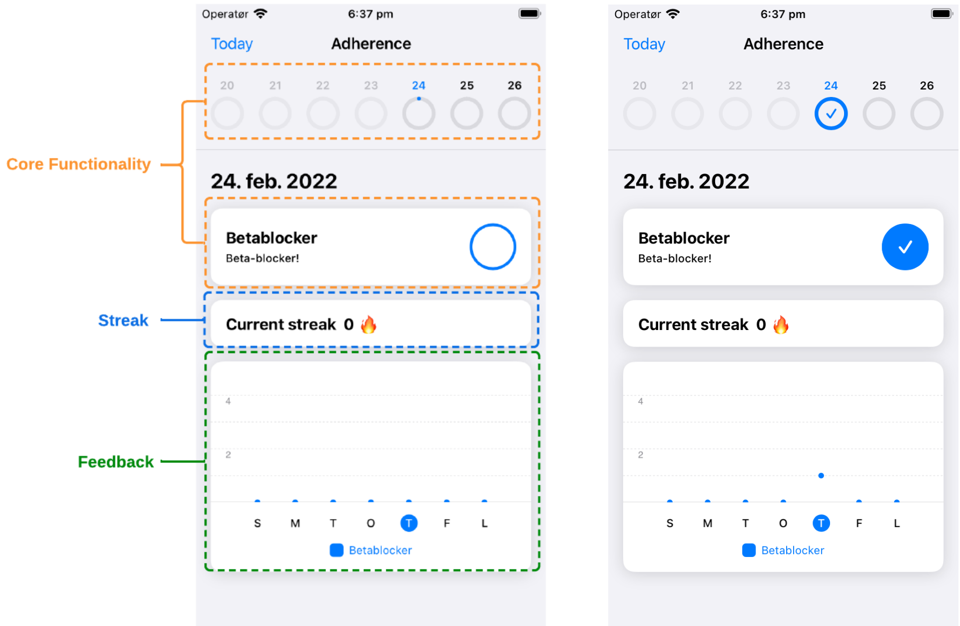

Figure 9: The application's user interface with a unfinished and finished task.

5. The application then fetches the outcomes of the beta-blocker task.
6. The screen then shows the beta-blocker task, a view that states how many days of streak the user has, as well as a table with a graph showing this week's medication adherence. This can be viewed in Figure 9.

   X-1 If the user has not completed the beta-blocker task within the current day the user can mark the task as completed.

   X-2 Then the user presses the task to complete it.

   X-3 Due to the change in the values of the CareKitStore, the Store Manager notifies the view controller to change its views according to the modification.

   Y-1 If the user has completed the beta-blocker task within the current day the user can un-do the task.

   Y-2 The CareKitStore will then delete the entry.

   Y-3 Store Manager notifies the view controller of modification.

   Y-4 View Controller changes views accordingly.

7. The User Data module will then be notified of the change.
8. The streak is calculated.
9. The view presenting the streak is updated.

4.4.2 Streak

The streak in the MedRem application was thought to be directly connected to the patient's medication adherence. Therefore, the streak was calculated using the number of days in a row the patient had taken their medication, minus one. If the patient forgot their medication one day, the streak would go back to zero. The streak feature thus only gives a reward to the patient if they take their medicine every day, not most days.

To implement the streak, this had to be tightly connected to the task described in Section 4.4.1. To calculate this attribute, the outcomes of the task had to be fetched from the CareKit Store. The streak would be calculated based on the stored outcomes of the beta-blocker task to keep continuity between sessions. CareKitStore stores the results of tasks on the phone. A task's outcomes are stored in an array, with each having a date attribute indicating when it was created. The application only allows the user to finish a task within the current day, so the user cannot finish or un-finish a task before or after the day that it occurs. Therefore, the streak could be calculated by traversing the list of outcomes and then counting how many days in a row from the current day the user had finished a task.

The user interface displaying the streak has the same styling as the other views in the view controller. This is achieved by using the CareKit framework and implementing a subclass of a CareKit class. Figure 10 shows how the streak is displayed. The streak uses the flame emoji to mimic the appearance of the Snapstreak from Snapchat. If the user were familiar with the Snapstreak, they most likely would be able to connect this to the same concept of a streak. This can trigger the core drive of 5. Social Influence & Relatedness, due to the familiar appearance of the streak.

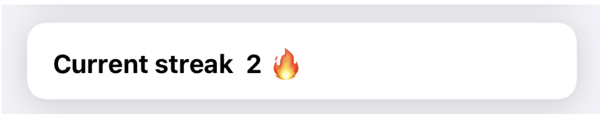

Figure 10: The user interface that displays the users current streak

The streak is calculated through the following flow.

1. A method takes in the list of outcomes connected to the task for beta-blocker medication.
2. The last outcome is retrieved from the list.
3. The difference in the number of days between the last outcome and today is calculated.
    a) Case 1: Number of days is larger than 1 - The streak is set to 0 because that means that the last outcome is older than yesterday. The method returns and terminates
    b) Case 2: Number of days is less than 1 - There is a streak to be counted, and the method continues.
4. The list of outcomes is then traversed.

5. The number of days between the current outcome and the previous one in the list is calculated.
    a) Case 1: Number of days is larger than 1 - Then the streak is set to be what was counted before comparing, and the function returns and terminates.
    b) Case 2: Number of days 1 - The streak is added with 1
6. If all outcomes are gone through and there has been no case of 5b) then the streak is set to the number that has been counted, and the method terminates.

Connection to the Apple Watch

The core functionality requirements state that the resting HR data from the MVP should be used as a control mechanism. It was thought that the heart rate data would ensure the actual mediation intake of the patient by matching it with the collected resting HR. This was to connect the functionality of the MVP further and take advantage of the use of the Apple Watch. Unfortunately, this functionality was not implemented, but the logic behind it would be to alter the notification triggered by the MVP, as well, as a higher resting heart rate would lead to a loss in the streak, even though the user was adherent in the application. The notification that is currently as displayed in Figure 11. The notification would need to be different to include the aspect of the streak. If the user has no record of adherence in the application, the notification can remain the same as in Figure 11. However, the notification text should be altered if the user is adherent in the application, but the resting HR says otherwise.

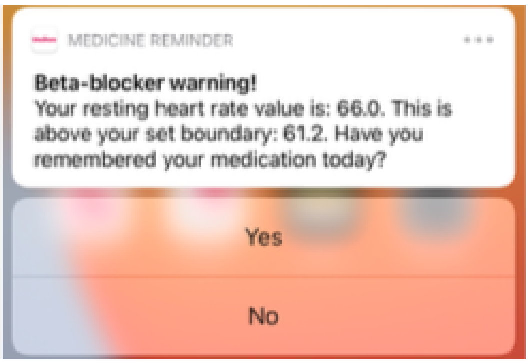

Figure 11: The MVPs notification of a too high resting HR

4.4.3 Feedback

The streak addresses the continuous adherence to the medicine and gives the user some reward. The literature emphasized user feedback, and it is the most commonly used gamification element for elderly. The following will describe the functionality created to meet the three feedback requirements listed in Section 3.3.3.

Number of days	Category	Feedback
7	Perfect	”Well done! You have taken your medication every day this week”
5-6	Good	”This week you remembered your medication for the most part. Keep going, you’ll make it to 7/7 next week!”
3-4	Average	”This week you only remembered your medication half of the time, work better to remember your medication next week!”
1-2	Bad	”This was not a good week, to improve the effect of your medication, take your medication everyday next week! You can do it!”
0	Horrible	”You haven’t taken your medicine all week, try to put the medication somewhere you can see them so that you take them next week as well!”
Any other number	Error	”An error has occurred”

Table 6: The different type of feedback the user gets on each the different categories of adherence

Weekly Feedback

The first functional requirement related to feedback was to give the user feedback at the end of each week on how they had been doing. A text card presents the feedback. Therefore, when the user finishes their task in the application, they can see how well they have been doing. The logic and flow of assigning the feedback are given in the following.

1. Traverses through the list of outcomes stored in the CareKitStore from start to end of the list
2. A new outcome is chosen
3. Checks if the date of the outcome is within the same week
   a) True - the number of days is added with 1, the code then continues at step 2
   b) False - the number of days-attribute is returned, and the function is terminated
4. When all outcomes are traversed through, the method returns the number of days counted
5. That number is then associated with a category, where each category returns a feedback text. The number of days, categories, and feedback are listed in Table 6

Graph

CareKit offers a simple way of connecting the outcomes of a task to a graph. The outcomes are connected by associating it with the task ID of a task in the CareKitStore. Therefore, when the outcomes of a task are changed, the graph automatically changes. How this is done programmatically in the application can be viewed in Listing 8. The user interface of this code can be viewed in the bottom half of the view controller in Figure 9. It is also clear how it synchronizes with the task being unfinished and then finished.

```
//1: Create a Data Series based on a Task with a given Task ID
// in the CareKitStore
let betablockerSeries = OCKDataSeriesConfiguration(
                              taskID: "betablocker",
                              legendTitle: "Betablocker",
                              gradientStartColor: self.view.tintColor,
                              gradientEndColor: self.view.tintColor,
                              markerSize: 3,
                              eventAggregator: .countOutcomes)

//2: Create a Chart of a given type, with the data series,
//and connect to CareKitStore
let betablockerInsight = OCKCartesianChartViewController(
                              plotType: .scatter,
                              selectedDate: Date(),
                              configurations: [betablockerSeries],
                              storeManager: self.storeManager)

//3: Append to the view controller
listViewController.appendViewController(betablockerInsight,
                                   animated: false)
```

Listing 8: The steps to create a graph based on the outcomes of a task.

Sharing of Feedback

Due to the sensitivity of the data on a patient's adherence, this step was not prioritized. By sharing their progress, such as their streak, with other patients or external parties, the security measures had to be thoroughly investigated to ensure that all data sharing was done according to privacy law. This work was expected to take a long time, and so when the development process ended, this functionality was not implemented. Although the user can always screenshot their phone screen and send it to people, it is not promoted to the user in the application. Therefore, this requirement is evaluated as not being completed.

4.5 Onboarding

With the application introducing user interaction, some needs have arisen. A guide to the application was deemed necessary for the patient to interact with it successfully. The guide will be implemented as an onboarding process. It should be executed before the application has started to be used, together with a health professional. The onboarding would also ensure that the data added to the application was correct. Having an onboarding process fulfills this need.

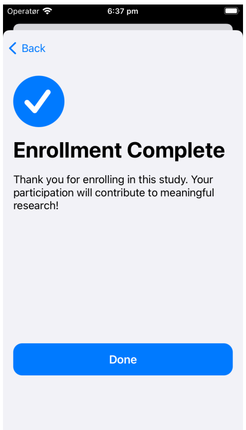

Figure 12: Complete onboarding screen

For the onboarding to seamlessly integrate with the remainder of the application, it was decided to use the ResearchKit survey. ResearchKit surveys make it simple to integrate asking for permissions, such as notifications and access to health data, within this onboarding process. ResearchKit surveys are also well documented by Apple through one of their Apple Worldwide Developers Conferences. Therefore, there was a greater possibility that developers had used it for this purpose, and thus the online resources could be better. During the survey, the patient would also submit their boundary heart rate and which time of day they wanted to take their medicine. Using the ResearchKit survey gives a simple way to add advanced features to the onboarding form in the future, such as signing authorization to be part of a research project.

The onboarding process has four main steps. Before and after these steps, there was presented a Welcome- and Enrollment Complete-step, Figure 14 and Figure 12 display these steps. In Section 2.2.5 the integration of ResearchKit surveys, such as the onboarding, into CareKit is presented. When integrated, the survey will be presented as a task. The task will be the only thing present within the view controller until it is completed. How the task is presented is displayed in Figure 13.

The following describes the onboarding process steps in detail. Because this is a collaborative process, it will be defined when a care provider or patient is expected to control the applications interaction. If both the care provider and the patient can perform the interaction, the term "user" will be used to describe it.

1. **Information step**
   In this step, no action from the user is needed, only that they read through every part and then press next. This was designed to give the patient insight into what the app does and also what information is stored about the patient. The step informs them about permission to share health data, the tracking of regular intake of beta-blockers, the notification that will remind them to take their medication, and data security. This step of the survey is displayed in Figure 16. This step uses the Instruction Step-type from ResearchKit.

2. **Permission step**
   ResearchKit makes it easy to ask for different permissions in their Permission Step. This makes it possible to ask for categories of permission, for instance, Health Data, and then define what types of Health Data you want. In this case, it only asks for permission to get the resting HR values. This step also requests to send notifications. The step is displayed in Figure 17.

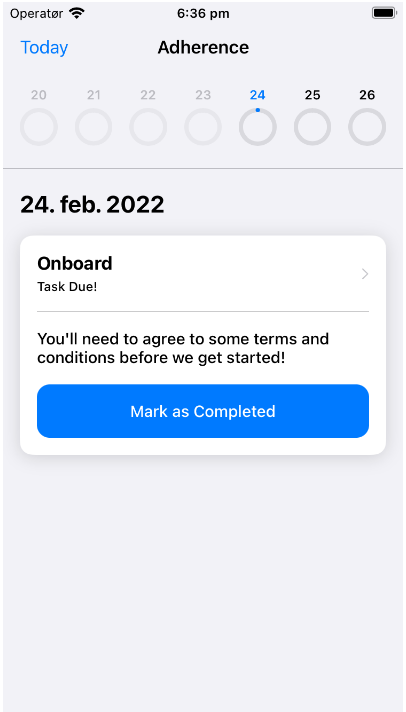

Figure 13: The onboarding displayed in the view controller when not completed

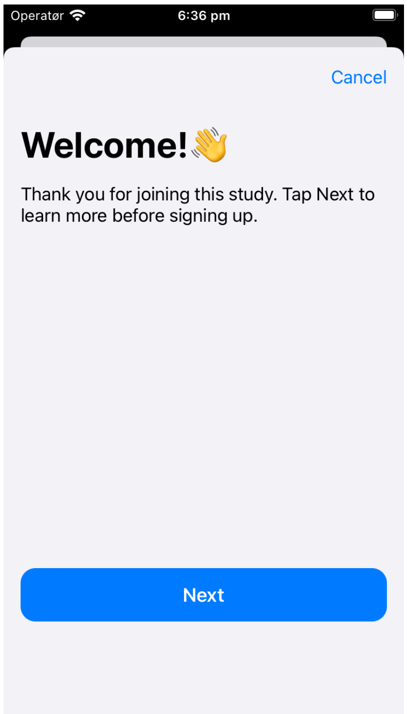

Figure 14: Welcome page to the onboarding

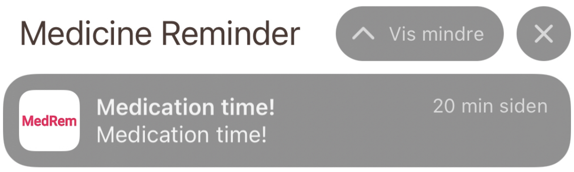

Figure 15: The reminder the user receives at a predefined time every day to take their medication.

3. **Boundary Heart Rate step**
   This step was included for the testing of the application. In the previous MedRem, there was a possibility to change the boundary HR. This was made to be able to set the boundary HR lower than reality and thus provoke notifications. This could not be integrated in the same way into the current application. Therefore, it was added to the onboarding and can easily be removed from the onboarding when the application is no longer used for testing purposes. The figure uses the Questions Step with a numeric format answer. The boundaries set for the answer are between 0 and 200, and any other number will give an error message. The step is displayed in Figure 18. This step should be completed by the care provider.

4. **Time of Medication step**
   With the need for more interaction within the app, the need to remind the patient to take their medication at a certain time was also added. This was to create a link between the application and the action of taking their medication. Therefore, the notification was made to be a regular, simple notification. How the user views the notification is displayed in Figure 15. To ensure that this was set, it had to be included in the onboarding. Here the Question Step is used, with the format being the time of day. The step is displayed in Figure 19.

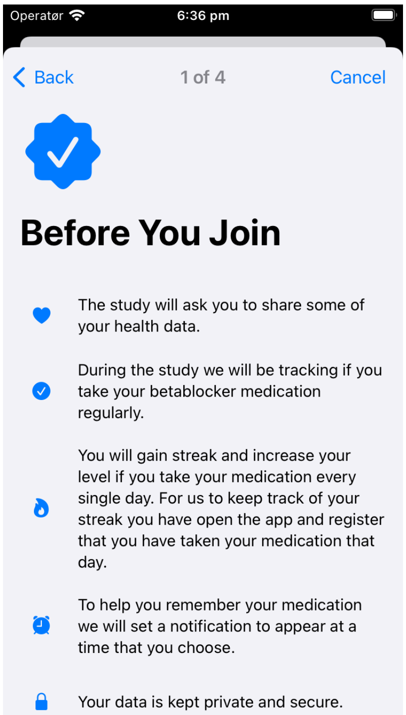

Figure 16: The Permission step of the onboarding

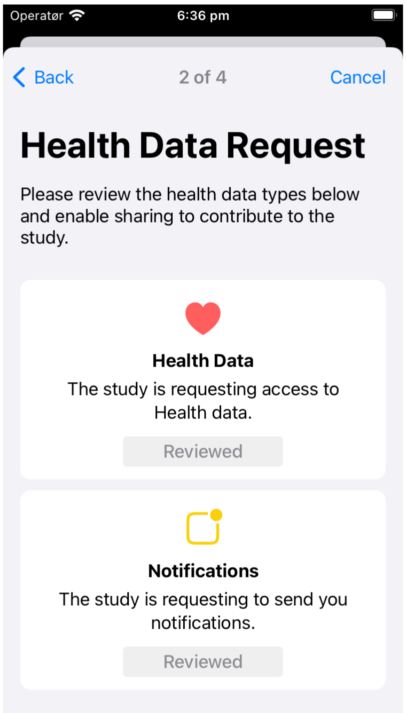

Figure 17: The Permission step of the onboarding

4.5.1 Flow

The onboarding will only show if it is not completed. This is checked every time the CareFeedViewController is shown. This flow follows the start of the flow from Section 4.4.1 and continues in the case onboarding is not successful.

1. If the user has not completed the onboarding the onboarding task will be the only thing shown in the app. This can be displayed in Figure 13.

2. The user presses mark as completed button on the onboarding task.

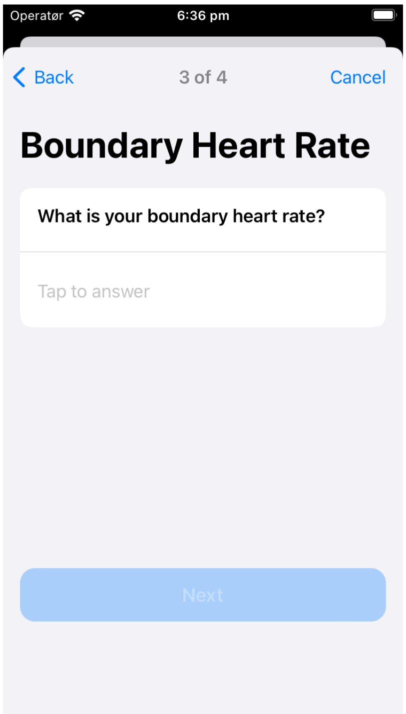

Figure 18: The Boundary Heart Rate Step of the onboarding

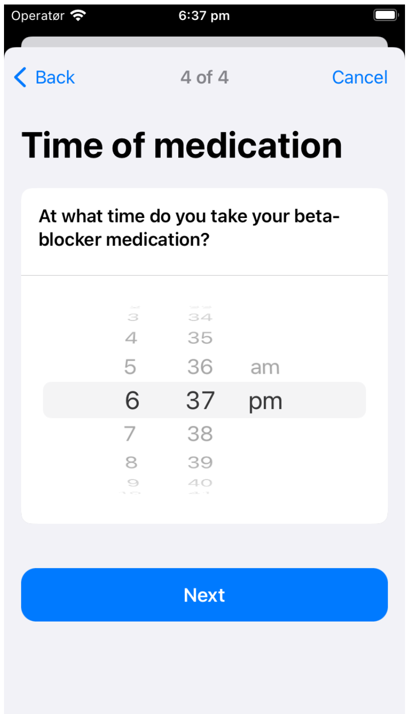

Figure 19: The Medication Reminder step of the onboarding.

3. A popover sheet will then pop up, and show the user the Welcome screen, the user then presses the “Next”-button. This popover screen can at any time be dismissed using the “Cancel”-button in the top right corner. This screen is displayed in Figure 14.

4. The popover shows the Before you join-screen, displayed in Figure 16, then the user presses the “Next”-button.

5. The popover shows the “Health Data Request”-screen, displayed in Figure 17, and will only be able to go to the next screen if the user reviews both requests. The user presses the “Next”-button.

6. The popover shows question of boundary heart rate, displayed in Figure 18. The user can only press next when the field is filled with a value between 0 and 200. The user presses the “Next”-button.

7. The patient gets asked to choose a time of day that they want a reminder, displayed in Figure 19. The ”Next”-button is disabled until a time is chosen. The user presses the ”Next”-button.

8. The popover shows a the ”Enrollment completed”-step shown in Figure 12, the user presses ”Done”.

9. When the onboarding is finished, the result is retrieved form the ResearchKit survey is stored in the CareKitStore. How this is programatically done is described in Listing 6.

10. The application will then set the trigger boundary, and store this locally on the phone.
11. The Notification Handler will then schedule a recurring notification at the date and time set in the onboarding and deliver it to the Notification Center.

4.6 User Interaction & Experience

ResearchKit and CareKit offer a user interface (UI) when using the classes and types of the framework. These UI views follow the design guidelines and principles of Apple. Since Apple develops the views, their appearance resembles other Apple applications. When using only these views, the application's design is familiar to the user and creates continuity throughout the application. This also supports the principles listed in Section 3.4. Apple's UI components are configured to ensure accessibility. Since the target group of this application is elderly, the likelihood of having special needs when it comes to accessibility is high. Therefore, to avoid spending time on features like scalable text and speaker assistant to be integrated with new UI views, the focus was to keep as much as possible within the already configured Apple UI components. Two views were made as customized UI views from these frameworks. These, therefore, do not include all accessibility functionality but still support dynamic fonts. These two views are the streak and the weekly feedback. The Apple UI components give both the feature of speaker assistant and dynamic font sizes, while the streak and weekly feedback views only support dynamic font sizes. The dynamic font sizes are displayed in Figure 20.

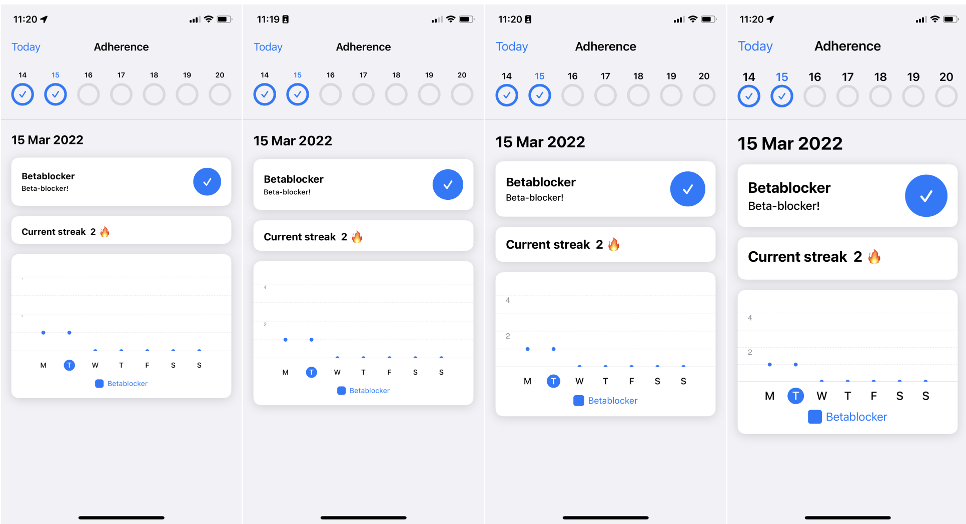

Figure 20: Application with dynamic font sizes

5 Evaluation

5.1 User Testing

Based on the principle of early and frequent delivery, the goal of the development was to go through multiple intervals of user testing. This was to provide feedback and assess and further develop the application in line with the concerns of users of the target group.

There are three types of user testing (Sommerville, 2016).

- **Alpha testing**, where there is a selected group of users that work closely with the development team to test early releases of the software.
- **Beta testing**, where a larger group gets access to a release where they can experiment and report of problems to the developers.
- **Acceptance testing**, where customers test software to determine if it is accepted and can be deployed to the customer environment.

Beta testing of the application would be technically possible due to the TestFlight framework described in Section 2.1.4. With TestFlight, it is simple to distribute the application for testing and get feedback from users. The issue with this method lies in the accessibility of users within the target age group. Also, the tasks of supporting and training the testers, managing feedback, and analyzing results would exceed the resources available for this thesis. Beta testing is a good fit for software products that are used in many different settings (Sommerville, 2016), this is not the case for this application, where the application has one definite setting.

Acceptance testing is largely based on the relationship between the developers and a customer. However, in this case, there is no prominent customer awaiting the delivery of the application. Therefore, this type of testing may not be the most relevant form of user testing.

On the other hand, Alpha testing focuses on a smaller group of testers, which is easier to manage. This form of testing also relies on the users to identify problems and issues that are not apparent to the developers. Whereas developers work from requirements, these do not always reflect external factors that affect the practical use of the software (Sommerville, 2016). Since there is a prominent gap both in age and understanding of technology between the developer and the end-users of the application, these insights have high significance for the success of the application. Based on these factors, Alpha testing was determined to be the user testing method that was the best fit for this thesis.

5.1.1 Test Execution

The focus of the application is to work against the forgetfulness and lack of motivation that most commonly lead to decreased adherence for the elderly. Forgetfulness and lack of motivation are not isolated cases of beta-blocker patients. Therefore, it was considered sufficient to test the application outside of this specific target group. The main criteria for the testers were that their age fit the age group elderly. In addition, none of the testers suffered from heart disease or used beta-blocker medication.

Two external testers tested the application. They were ages 63-66, thus being a part of the age group. The testing was performed in two intervals, with one tester in each interval. In the first test interval, the goal was to map out if the new functionality worked and if it made sense to an elderly user. In addition, it was essential to discover if parts of the application that were obvious to the developer were not that straightforward to an outsider in the suitable age group. In the second interval of testing, the feedback from the first test was implemented, and the application was altered accordingly. The primary aim of the second interval was to test, over a more extended period of time, three weeks, if they could keep their streak. The secondary aim was to get further insights into the introduced functionality and the changes that were made since the first test interval.

Both of the testers have used devices from the Apple ecosystem for over six years and used their own iPhones to test the application. Therefore, it is probable that the testers were familiar with the Apple design and user interface.

The application was distributed through TestFlight, a platform described in Section 2.1.4. TestFlight enables the testers to download the application as if it were in the App Store for testing. It was also possible for the testers to send feedback if the application crashed or by taking a screenshot.

5.1.2 First test interval

Execution

The first test interval was primarily to test the application's usability and, if present, discover defects. The interval was executed by one tester. The tester used an Apple Watch provided by the developer and installed the application on their own iPhone. The duration of the test was five days. Since the tester was not on beta-blocker medication, the test was designed for them to do another task that would simulate the same behavior as a patient using beta-blockers would have. The tester was asked to drink a glass of water at a set time each day as an independent action, like taking a beta-blocker pill.

The developer assisted the tester with the application setup and guided them through the onboarding. In addition, the developer was available to the tester for questions or comments during testing.

Results

One problem identified was the distinction between having to do an action in the application to register that the medication has been taken and getting a notification to remind you to finish the task. Therefore, when the tester saw the notification with the reminder to take their medication and did the task that was given to her, she thought that was enough. The concept of what streak was unclear to the tester. From this feedback, two things could be done. The application could be changed to clarify how it works, or the information given to the user could be more clearly explained. Since the frameworks used are somewhat rigid and time-consuming to customize, it was decided that the onboarding of the application should be changed to clarify the steps for the user.

The tester also provided comments on the reminder to take their medication. She thought that the reminder should be more in the form of a text message and thus provide more context and information than the current solution. The tester also emphasized that the reminder should have some sound to draw more attention to the phone and the task the user has to do.

5.1.3 Second Test Interval

Execution

The second interval had one tester, a different person than the tester from the first interval. During this interval, the tester took an unrelated medication to complete the task in the application. This test aimed to see if the tester would take their medicine for three weeks straight and report it in the application.

Covid-19 brought some difficulties into the second interval of testing. Due to quarantine and the time restrictions for this master's degree, it was decided that the second test would be executed remotely. The application setup was done over the phone, where the developer gave detailed instructions for each step, and the tester sent screenshots if there were any uncertainties. Unfortunately, the tester did not have an Apple Watch available. Therefore, this test interval only included interaction with the iPhone part of the application. The rationale for the decision to proceed without an Apple Watch, was that the new functionality added was focused on the iPhone application rather than the Apple Watch. Therefore, the test would give valuable insight into the gamification functionalities and their effects without using an Apple Watch.

Results

During the onboarding of the application, a question about the definition of "streak" arose. After explaining what streak meant and how it worked in the application, the tester understood the concept. However, this emphasized that the streak concept is not as evident as the developer expected.

After the entire three weeks had passed, the tester had a maximum streak of eight, thus completing the task in the application nine days in a row. The streak was broken on Thursdays, Saturdays, and Sundays. Furthermore, the task was not completed for four days out of the 21 days of the test. Even though the tester did not complete the task every day, the tester took their medicine all 21 days.

The tester commutes Monday through Thursday and stays in a hotel. He said that the differences in routines at home versus at the hotel were significant when failing to complete the task. The tester took the medication when he was in bed and kept it at the bedside table. The tester also had his phone at the bedside table when in the hotel. Consequently, he always remembered to do the task on the application. However, that routine is different in his home, where he puts his phone away to charge in the hallway before he goes to bed. The tester reported that several times when he took the medication, he was about to fall asleep and remembered that he had forgotten to complete the task in the application. Even though he remembered it then, he did not want to get up and potentially struggle to fall asleep afterward.

The change in routine on the weekends also affected the capability of the tester to complete the task. On a Saturday, when the tester did not complete the task, the reason was that he did not get home till past midnight. Since his routine for medicine taking is to take it before he goes to sleep, and the application limits the registration of medicine taking to the same day, the tester could not register that the task was finished.

At the end of the first week, the tester reported that the reminder to take their medication was not appearing. However, he later discovered that these notifications had been delivered at the right time every day. The issue was that the tester had a setting to go into focus mode when the notification was issued. When in focus mode, all notifications are silent, not showing up on the home screen or giving a sound alert. They are present in the notification center, but the user has to look for them actively.

The tester stated that it was annoying to lose the streak, "even though it was only a competition among himself." He also admitted that the streak triggered his competitiveness. The feedback that was given every Sunday was informed about at the start of the test, but the tester did not register it. This might be a consequence of the two Sundays that the streak was lost. When asked about the graph showing an overview of tasks completed that week, the tester had not registered or studied it before it was mentioned at the end interview.

Finally, the tester had some suggestions for the application to better fit his needs. His main concern was the lack of the possibility to go back and change the settings that were set in the onboarding. He suggested the possibility of changing the notification time dynamically, for instance, having a different time of the day on the weekends and weekdays. This would not just apply to him with his commuting work schedule, but because people usually have different sleeping schedules during the weekends. The notification should also be possible to change due to vacations or other changes in the usual schedule.

5.2 Evaluation of Requirements

For each of the requirements given in Section 3.2 the requirement will be assessed and deemed successful or not. For simplicity, the broader, less specified requirements will be used for this evaluation since they all embody the same functionalities as the more specified ones.

5.2.1 Onboarding

The purpose of the onboarding was to introduce the user to the application and give instructions on how to use it. After being iterated and changed, the primary outtake of the onboarding is that the streak feature was difficult to understand. This will be further discussed in Section 5.2.3. It was evident in both testing intervals that the onboarding was what took the most time for them to get through, assisted and explained by the developer. While this took some time, it is important to ensure that the applications are used correctly, so it should not be of much concern. More illustrations from the application could be used to connect the information with what they see afterward to make it simpler for the users to understand. It is also possible to include a tour of the app as part of the onboarding process, where each application component is explained. This is common when an application is opened for the first time, or its design and structure have changed.

5.2.2 Reminder Notification

The reminder notification has some work to be done. An important aspect is that it is in addition to the MVP's notifications. The main issues with this notification were that it did not appear due to the focus mode setting and could not be changed. However, by introducing the possibility of changing the time of the medication reminder notification, the application can also alter changing routines. This is further discussed in Section 6.2.1.

The phrasing of the reminder was commented on by one tester, with them wanting it to be more like a text message. However, since there was confusion about what the notification did, a short sentence emphasizing that the user should interact with the application might have avoided possible misunderstandings.

For simplicity, the reminder notification was only set to be a notification without any other action when pressing it than to open the application. In the MVP, a notification appears when the resting HR is higher than the boundary. This notification can schedule a reminder for users to take their medication at a given number of hours. This functionality could also have been implemented for the medication reminder to give the user more flexibility. In the study of Kamat et al. (2021) they tried having multiple notifications. The notification reported to be the most effective was the one that happened at the exact time the medication should be taken. Therefore, the extension of having multiple scheduled reminders was not considered.

5.2.3 Streak

The streak is the most prominent game element in the application today. The main issue with the streak during testing was the lack of flexibility. In a sense, the streak should be strict because it should be challenging to maintain. The whole goal of the streak is to make the user take their medication every day. On the other hand, if the user loses their streak, they have to start all over again, which can be frustrating and lead to the opposite of what we want—less motivation. This type of behavior was experienced by Fotaris et al. (2015), where participants lost interest when trailing behind on the leaderboard. Therefore, the measures for counting the streak could be reconsidered. For instance, there is no possibility of going back and changing what happened the days before. This is to prevent users from cheating the streak by going back and checking if they took their medicine in the past. But perhaps the application should open up to going back to the day before. This will also lead to the opportunity for people to go to bed and take their medication after midnight. In addition, the user gets the chance to register their medication task in the application the morning after. Giving the user one day to correct such behavior makes the streak more flexible, but the user still has to interact with the application daily to keep the streak.

Another measure that can be taken can be to measure the streak less strictly. Currently, a 100% adherence is the only thing that qualifies for maintaining a streak. While in the study of diabetes by Kamat et al. (2021) their measure of adherence was only 80%. By introducing this type of logic, the user would not have to take their medication multiple days in a row to lose their streak. While this logic would give more flexibility to the user, the health of the patient is the primary priority. An issue with such a limit is that the application indicates that an adherence lower than 100% is good enough. Therefore, there has to be a form of medical background that sets which percentage the adherence can have before it has any health complications for the patient. There can be badges, levels, and titles connected to having a high percentage over time to motivate the patients to have the highest adherence percentage possible.

The streak had an engaging effect. The tester, in the second interval, expressed that it had been like a competition with himself. After he lost his streak two weekends in a row, he promised himself to keep it throughout the testing period. When asked how it felt when he lost the streak, if he had felt any anger or irritation, the tester said that he did not care *that* much.

The testers were confused when the concept of the streak was introduced. The term had to be explained thoroughly for the testers to understand what it meant and its purpose. This was surprising because of Snapchat's great success using streak. The users of Snapchat have a common understanding of the term "streak." However, the elderly might not be the biggest user group, nor the user group most focused on the streak on Snapchat. Consequently, the concept is not that widely known to them.

5.2.4 Feedback

One issue was that the tester's weekly feedback was not registered, even though they were present three times. Since the feedback is shown on Sundays, there might be a correlation between this being the day the tester missed his streak the most. The feedback is implemented to motivate and encourage adherence. Therefore, it should also be presented to users who are not perfectly adherent. Ensuring that the non-adherent users get feedback can work as a trigger to change their behavior and become more adherent. Consequently, the core drive of 3. Empowerment of Creativity & Feedback is present for all users. In addition to in-app feedback, users could get notification-based feedback to increase their likelihood of seeing it. That means users get a weekly notification summarizing their adherence for the previous seven days. The message inside the application can be kept and just emphasized with the notification.

Currently, the feedback is personalized by reflecting the adherence of the user. Simple steps of using the user's name can amplify the feeling of the application speaking personally to them and further tap into the core drive of 4. Ownership & Possession.

5.2.5 Level, Badges and Titles

This requirement was not implemented in the application due to time restrictions. The idea was that there would be a level associated with the streak. For instance, if the user reached a streak of 10, they would get to level 1 and get a badge and title. If the user lost their streak, their title, badge, and level would not be lost. This would make up for the possible loss of motivation after a losing streak. To get increasing challenge and accomplishment, the gaps between the levels and badges should tend to be larger. Bodduluri et al., 2017 reported the importance of increasing challenge in the task to keep adherence. The first five days of a streak would yield a level, while the next one is not until 10, then 50, and so on. Setting these boundaries for each level, it must not feel impossible even though the difficulty becomes greater. The streak itself somewhat contributes to the drive of 2. Development & Accomplishment, but these measures would more effectively trigger this drive due to the more defined ”challenges”.

5.2.6 Non-functional Requirements

Product Requirements

All product requirements but one were fulfilled. An overview of the evaluation can be viewed in Table 7. All sensitive data in the application is stored in the CareKit Store. The CareKit Store is an on-device option for storing. Therefore, the data is not sent to an external server for storage. This reduces the risk of leakage. CareKit Store also ensures that the data is fully encrypted anytime the device is locked. As described in Section 4.6 and in Figure 20, the application provides dynamic font sizes. The application is also a single-page application. While testing the application, the features, especially the streak, were discovered to need more explanation than expected. The onboarding and a tour of the application should happen together with a care practitioner. Therefore, the onboarding itself only provides the patient with some of the needed training. However, the care provider can fill the gap between the onboarding and the remaining knowledge needed.

Requirement	**Evaluation** (Fulfilled, Partly Fulfilled, Not Fulfilled)
The application must store sensitive data in a secure and encrypted manner.	Fulfilled
The application must support dynamic text sizes in the user interface	Fulfilled
The application should not need any further instructions or training than the onboarding for a user to use it.	Partly Fulfilled
The application should be a one-page application	Fulfilled

Table 7: Evaluation of Product Requirements

Organizational Requirements
An overview of the evaluation of all organizational requirements are given in 8. The requirements of Apple's Design Principles consists of six points which are listed in Section 3.4. Based on the arguments of the following the requirement is evaluated to partly fulfilled.

- **Aesthetic Integrity**
  The design of the application is simple and fits its purpose. There are no graphics and animations that do not contribute to the functionality and seriousness of the application.

- **Consistency**
  Since the application uses exclusively CareKit and ResearchKit components and their views, the design is similar to other Apple applications. In addition, the views have a consistent design, with the same colors, shapes, and fonts. A part of consistency is that the application works as the user expects. The majority of the application does so, but there was an inconsistency in the onboarding during testing. In the second test, when the user was onboarded over the telephone, they struggled to start the onboarding process from the screen displayed in Figure 13. The issue was that when pressing the arrow in the top right corner, a popover with the exact text appeared. The button with the text "Mark as Completed" had to be pressed to actually start the onboarding, which was not found to be intuitive to the tester. The CareKit view controller treats the onboarding as a CareKit task. All CareKit tasks have a set design, and therefore, the button name is locked. This is not considered an issue since the onboarding will be a guided experience for the patients.

- **Direct Manipulation**
  The application gives the user visible results of their actions. For instance, in Figure 9 all views that have something to do with the completion of the task change when the user completes the task.

- **Feedback**
  As described in the previous item, the application always gives users feedback when taking their medication. In addition, the gamification elements for providing feedback strengthen the

- **Metaphors**
  No part of the application was found to have a real-life metaphor that matched the feature's purpose. Therefore, this point was not found relevant for this application.

- **User Control**
  In the application, the user can only control and change the tasks for the current day. However, this ensures that the user takes and registers their medication on the same day. Therefore, this does not limit the users' control more than necessary.

The requirements of Apple's best practices for Inclusive design is listed in Section 3.5. In the following I will evaluate the application towards each of the practices.

- **Design with Accessibility in Mind**
  The application's design is simple, with no more views than necessary, with four views at the most. The application is also a one-page application that supports simplicity. All content can be perceived with sight, and all but two views can be heard with Voice Over.

- **Support Personalization**
  The application supports personalization in terms of dynamic text throughout all views.

- **Audit and Test Your App for Accessibility**
  This requirement is not fully fulfilled since the Voice Over does not work for all views. These views include the functionality of Streak and Feedback, which is a critical part of the application.

Based on these points the requirement is evaluated to partly fulfilled.

Requirement	**Evaluation** (Fulfilled, Partly Fulfilled, Not Fulfilled)
The application should follow the Apple Design Principles, listed in section 3.4	Fulfilled
The application should follow Apple's best practices for Inclusive Design, listed in Section 3.5	Partly Fulfilled
The application must be written in Swift	Fulfilled
The application must integrate the ResearchKit and CareKit frameworks	Fulfilled

Table 8: Evaluation of Organizational Requirements

External Requirements

The GDPR guidelines for privacy rights and how they are fulfilled are given in the following. Since all data about medication adherence is only saved locally on the user's phone, the storage of this sensitive data will not be taken into account.

- **The right to be informed**
  The user is informed of what data is stored about them when consenting to it during onboarding.

- **The right of access**
  When the application asks to collect the users' health data, it is specified that the collected data is only the resting heart rate.

- **The right to erasure.**
  The user can at any time remove permission for the application to access their health data.

- **The right to restrict processing.**
  The user can restrict the processing of the application by turning off notifications for the application. This is the only part of the application that processes the health data.

- **The right to data portability.**
  The health data recorded by the Apple Watch can be viewed in the Health-application on their phone, and there are several mechanisms for exporting this data into other systems.

- **The right to object.**
  The user can object to giving access to the health data in the onboarding process.

- **Rights in relation to automated decision making and profiling.**
  The only automated decision currently in the application is to send a notification to the user at times when the resting heart rate is too high. This is a functionality that is for the benefit of the user and can be silenced by turning off permission to send notifications.

Requirement	**Evaluation** (Fulfilled, Partly Fulfilled, Not Fulfilled)
General Data Protection Regulation (GDPR) laws and regulations.	Fulfilled

Table 9: Evaluation of External Requirements

6 Conclusion and Future Work

6.1 Conclusion

During the work of this thesis, I integrated gamification elements into the MedRem MVP. Two frameworks, CareKit and ResearchKit, facilitated the integration. As a result, the MedRem application has gone from being a background application, expecting close to no user interaction, to demanding it every day. In addition, the user interaction has introduced new concerns that need to be considered. The two gamification elements integrated with the application were streak and feedback.

The streak counts the number of days that the user completes the task in the application to take their medication. The initial testing of this feature has given good promise. However, in the second interval, the tester stated that he experienced his competitiveness getting triggered, even after losing his streak. An issue with the streak was that this was a relatively new concept to the testers. Therefore, there is a need for a more thorough explanation of the streak. Either during the onboarding or by the care provider.

The feedback feature was twofold. First, the user can see feedback in the form of a graph demonstrating adherence over the previous week. Every Sunday, the user receives a written summary of their current week's adherence. These types of feedback were not visible to the tester. Therefore, it is difficult to evaluate if the feedback could have had an effect or not. The only conclusion that can be drawn here is that the feedback needs to be more prominent in the application.

(Martinho et al., 2020) presented the four most relevant gamification effects for elderly as feedback, sense of progression and improvement, points/scores, and social interaction. The application implemented all except social interaction. The streak, which represents the sense of progression and improvement as well as points/scores, had, as (Martinho et al., 2020) suggested, a positive effect.

ResearchKit was used to conduct the onboarding process and was a framework that gave the developer simple tools to collect permissions and give the user information. CareKit, on the other hand, facilitated the implementation of the gamification elements. Both recording and storing the vital data for the medication adherence streak and creating the feedback graph. Both these frameworks can contribute to the further development of this application.

Due to the streak, the application demands interaction every single day. Consequently, the user has to adhere to the application and their medication. The patients' medication-taking concerns can also translate to the application's interaction needs. The testing showed how the routines of the tester affected their streak in the application. The tester's routines changed during the week because he commuted for work between two cities. This change led him to lose his streak when the routine changed. His change in routine is not unique. Usually, people's routines change between weekdays and weekends. Retired elderly may have an additional challenge lacking a defined routine. A retired patient stated that due to their lack of routines, it was difficult to remember if they had taken their medication or not because their morning routine shifted from day to day (Ruppar & Russell, 2009). Based on this discovery, the further development of the application should consider the importance of routines. Currently, the application is static in regards to routines. The time of day for reminders is always the same, and the user cannot change it after the onboarding. To further facilitate changing routines, this should be more dynamic. The user should be able to change their routines based on different days of the week and go back and change the reminder time if they wish to change it.

So when answering the research question, *"How can gamification of a smartphone application affect the medication adherence of elderly patients?"*. The initial tests showed that the streak is promising and illustrated the importance of visibility towards the feedback. However, with the introduction of gamification elements, the issue of routines arose and how significantly them changing was. Therefore, routines should be a primary focal point for further development.

6.2 Future Work

6.2.1 Routines

A key reason the tester broke the streak in the second test interval was that their routines changed. At the same time, most people do not work in a different city than where they live and thus move from city to city during the week. However, people's routines change slightly from weekdays to weekends. In addition, most elderly are retired, which entails that they have the freedom to go on vacations, visit their cabins, and thus change their routines from time to time.

The problem with routines is not unique to this application and its effort to obtain medication adherence. Kidney transplant recipients had trouble remembering their medication when they had changes in their routine. For example, not eating breakfast at home became a reason for forgetting their medication. There are two typical strategies for remembering to take their medication; either having a location for the medication that makes them remember it or incorporating the medication-taking into their routines. For instance, a patient reported that they always took their medication after breakfast and when they had a glass of milk in the evening. The connection to routine actions worked as a trigger for the medication taking. Others reported that the location was significant and even made the changing of the environment when travelling less challenging (Ruppar & Russell, 2009).

Therefore, a suggestion for further development of the application is to create different routines for different parts of the week. The routine consists of the time of the day the user gets a reminder to take their medication. It should not include anything related to the medication dosage or other health values that the care provider should set. In addition, these routines should be able to be changed after creation, based on the tester's experience with regretting the time that he had set for his medication reminder after the onboarding finished. With the dynamic nature of the routines, the user gets an opportunity to change up the routine if it is not working as they thought it out in the beginning. In addition, the user can set up new routines when they are on vacation or in other periods where their environment will change their routine. Being able to create their own routines adapted to their lives will also contribute to the core drive of 4. Ownership & Possession, and thus give the user a more personalized feeling and ownership of the application.

6.2.2 Enhance and promote social interactions

After development, the MedRem application has three out of the four most significant game elements for elderly and adherence. Unfortunately, the aspect of promoting social interactions between participants was not included in the application. The current application does not have any mechanisms for this other than perhaps sharing a screenshot of the feedback given each Sunday. However, this would have been outside of the application's intended functionality.

The core drive of 5. Social Influence & Relatedness is a part of the intrinsic, right-brain drives described in the Octalysis framework (Chou, 2019). As mentioned in Section 1.5 these drives make the actions themselves rewarding, and Chou, 2019 believes it is what will yield the most motivation over time. The value of social bonding and interaction between participants is emphasised in the studies of Scase et al. (2017) and Martinho et al., 2020, which supports Chou (2019) emphasis on it. Perhaps the concept of Snapstreak is so addictive and works so well due to social intention. A Snapstreak has two people involved, and both need to do an action for it to work. This creates a companionship, here referring to the core drive of 5. Social Influence & Relatedness, between the two of them. Also, the loss is greater when losing the streak because the person also ruins it for someone else. This may amplify the core drive of 8. Loss & Avoidance.

The concern of promoting social interaction is that people's medical intake data concerns privacy regulations. Therefore, the application itself cannot share this information among users. However, a solution is to encourage users to share their progress outside of the application. Then the application's responsibility is removed since it is done by choice, not by default.

Wordle is a game that conceptually is one-player, but they have created an effortless way for users to share their results with others. The game itself is simple; you have six chances to guess a five-letter word. Each time you guess, you get to know if your letters are correct and in the right place. They will be green if they have the correct position and yellow if it is in the word but not in the right place. If the letter is not correct, it will be black. An example of a game is displayed in Figure 21. Every day there is a new word, and after you have completed the day's game, you can easily share your result. This is displayed in 22. Wordle will generate a string that you can easily share. An example of this can be viewed in 23. Therefore, without any extra technology, databases to create a leader board or any other programmatical difficulties, the game has become multi-player due to the users frequently sharing daily scores.

By creating such a solution for sharing the results of their adherence, the application could easily include this share-functionality within the weekly feedback without much extra work. The patients could then share their results with other patients they know are using the application or with friends and family to get cheered on and give them a sense of security that they are taking their medication. The solution's string pattern could be similar to Wordle's, with a string like Figure 24.

By creating such a solution, the application will gain the core drive of 5. Social Influence & Relatedness. The users' sharing of their streak strengthens the core drive of 8. Loss & Avoidance. Because now the people they share their streak with will know if they lose their streak.

Another application where adherence to a regime is essential is "Slutta", a Norwegian application for people who are quitting nicotine, such as cigarettes or snus. The application does not require the user to open the application every day but creates ways for users to share their success in quitting by sharing their no-nicotine streak of days, hours, minutes, and seconds. The user can export their streak in picture format from the application. Figure

Figure 21: Example Wordle game

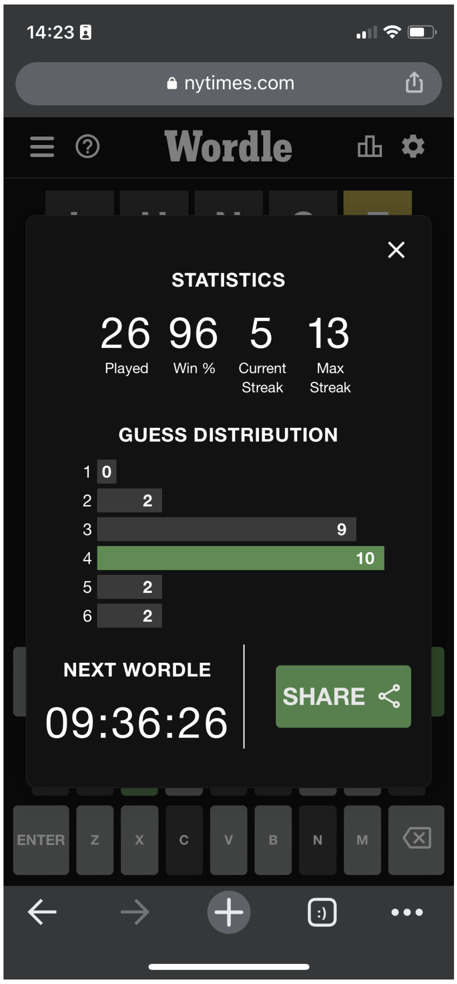

Figure 22: Example of Wordle statistic and sharing page

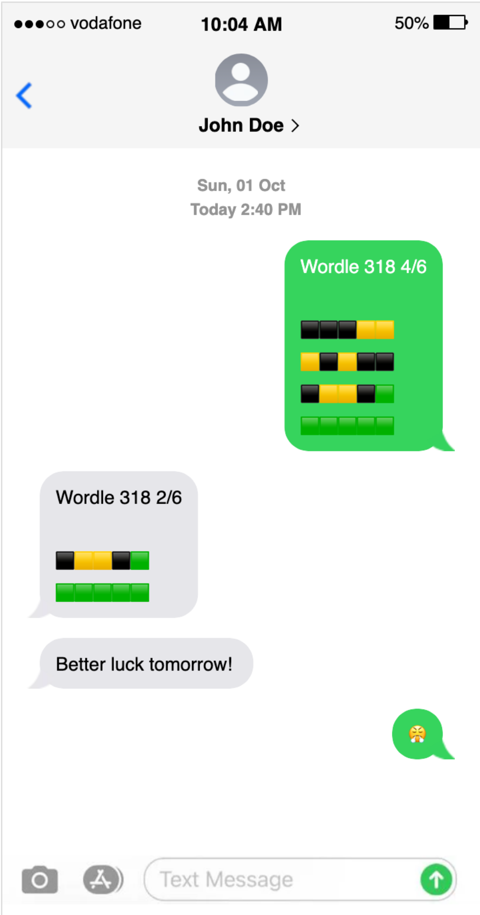

Figure 23: How a shared message of played Wordle looks

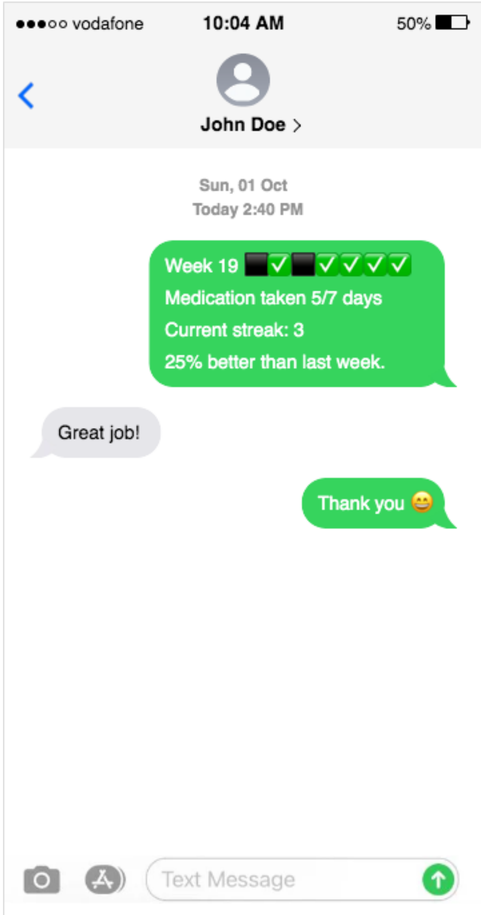

Figure 24: An idea of how such a message could be formatted

25 shows an example of this. The string generated in the Wordle-like solution is easy to alter. Therefore, generating a picture would result in trust between the users that the results are real.

6.2.3 Sense of urgency

While the application at the current moment reminds the user to take their medication, there is no emphasis on keeping their streak in the notification text. If the notifications were to impose a level of urgency on the user by emphasizing that the user would lose their streak if they did not take their medication, an additional trigger of the 8. Loss & Avoidance core drive would be present. Then there is no need to add any reminders, and still, the urgency of the medication-taking is emphasized. To further build upon this sense of urgency, it makes sense to introduce another notification just alerting the user that the streak is about to be lost. The application should send the notification at a time when the user is awake and active on their

Figure 25: The image that is exported from the Slutta application for sharing

phone. If the application gives the notification in the middle of the night, it will not have any effect.

6.2.4 Involving the Care Provider

CareKit can connect the patient with the care provider. However, the application has not implemented this functionality. By implementing an accessible channel for contact between the care provider and the patient, the follow-up of the patient's progression may improve for both patient and care provider. If the care provider were to get access to the patient's progress and adherence, it would be possible to schedule follow-ups. If the patient knows that their care provider can see how well they are doing, that may motivate them to do well and adhere to their medication.

In this scenario, the amount of time required for the care provider to monitor the progress must be considered. However, if it leads to fewer rehospitalizations of patients, which is valuable both financially and for the patients' well-being, that time may be well spent. In addition, as described in Section 1.2, one of the main reasons for non-adherence is a lack of support. Therefore, if the application can offer the patient further support from their care provider, they might have a better prognosis towards adherence.

6.2.5 Resting Heart Rate as Control Mechanism

To connect the MVP with the new iteration of the application, it would make sense to include the fetching of resting heart rate data in the new functionality. Currently, there is no control mechanism for registering medication-taking in the application. As a result, a patient can, in theory, have perfect adherence in the application but be non-adherent in real life. In the MVP, the role of the resting heart rate data was to determine if the patient was adherent or not and, if not, to remind the patient to take their medication. A possible way to solve the problem would be to use the resting heart rate data as a control mechanism to ensure that the application's data matches the real-life adherence. A mismatch between these two values may result in the user losing the streak or getting a warning.

References

Aanesen, J. (2021). *App for Increasing Patient Adherence To Medical Therapy* (Master's thesis). University of Oslo.

Apple Inc. (2022a). Asking Permission to Use Notifications — Apple Developer Documentation. Retrieved April 16, 2022, from https://developer.apple.com/documentation/usernotifications/asking_permission_to_use_notifications

Apple Inc. (2022b). CareKit. Retrieved May 11, 2022, from https://www.researchandcare.org/carekit/

Apple Inc. (2022c). CareKit [original-date: 2016-04-14T21:12:24Z]. Retrieved May 11, 2022, from https://github.com/carekit-apple/CareKit

Apple Inc. (2022d). HealthKit — Apple Developer Documentation. Retrieved May 11, 2022, from https://developer.apple.com/documentation/healthkit

Apple Inc. (2022e). iOS - Human Interface Guidelines - Apple Developer. Retrieved March 17, 2022, from https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/

Apple Inc. (2022f). Monitor your heart rate with Apple Watch. Retrieved March 12, 2022, from https://support.apple.com/en-us/HT204666

Apple Inc. (2022g). ResearchKit Framework [original-date: 2015-03-10T20:05:32Z]. Retrieved May 15, 2022, from https://github.com/ResearchKit/ResearchKit

Apple Inc. (2022h). Swift - Apple Developer. Retrieved April 12, 2022, from https://developer.apple.com/swift/

Apple Inc. (2022i). TestFlight. Retrieved April 12, 2022, from https://developer.apple.com/testflight/

Apple Inc. (2022j). Themes - iOS - Human Interface Guidelines - Apple Developer. Retrieved May 11, 2022, from https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/

Apple Inc. (2022k). Watch – Hvorfor velge Apple Watch. Retrieved May 11, 2022, from https://www.apple.com/no/watch/why-apple-watch/

Apple Inc. (2022l). Xcode. Retrieved April 12, 2022, from https://developer.apple.com/xcode/

Apple Inc. (2022m). Xcode — Apple Developer Documentation. Retrieved April 12, 2022, from https://developer.apple.com/documentation/xcode

Barrios, L., Oldrati, P., Santini, S., & Lutterotti, A. (2019). Evaluating the accuracy of heart rate sensors based on photoplethysmography for in-the-wild analysis. *Proceedings of the 13th EAI international conference on pervasive computing technologies for healthcare*, 251–261.

Bodduluri, L., Boon, M. Y., Ryan, M., & Dain, S. J. (2017). Impact of gamification of vision tests on the user experience [Publisher: Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA]. *Games for health journal*, *6*(4), 229–236.

Choi, A., Lovett, A. W., Kang, J., Lee, K., & Choi, L. (2015). Mobile Applications to Improve Medication Adherence: Existing Apps, Quality of

Life and Future Directions. *Advances in Pharmacology and Pharmacy*, *3*(3), 64–74. https://doi.org/10.13189/app.2015.030302

Chou, Y.-k. (2019). *Actionable Gamification: Beyond Points, Badges, and Leaderboards* [Google-Books-ID: 9ZfBDwAAQBAJ]. Packt Publishing Ltd.

Cramer, J. A., Roy, A., Burrell, A., Fairchild, C. J., Fuldeore, M. J., Ollendorf, D. A., & Wong, P. K. (2008). Medication Compliance and Persistence: Terminology and Definitions. *Value in Health*, *11*(1), 44–47. https://doi.org/10.1111/j.1524-4733.2007.00213.x

Croon, R. D., Geuens, J., Verbert, K., & Vanden Abeele, V. (2021). A systematic review of the effect of gamification on adherence across disciplines. *International Conference on Human-Computer Interaction*, 168–184.

Desta, L., Khedri, M., Jernberg, T., Andell, P., Mohammad, M. A., Hofman-Bang, C., Erlinge, D., Spaak, J., & Persson, H. (2021). Adherence to beta-blockers and long-term risk of heart failure and mortality after a myocardial infarction [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ehf2.13079]. *ESC Heart Failure*, *8*(1), 344–355. https://doi.org/10.1002/ehf2.13079

Deterding, S., Khaled, R., Nacke, L., & Dixon, D. (2011). Gamification: Toward a definition, 12–15.

Driving Directions, Traffic Reports & Carpool Rideshares by Waze. (n.d.). Retrieved March 31, 2022, from https://www.waze.com/waze/

Dunlay, S. M., Roger, V. L., & Redfield, M. M. (2017). Epidemiology of heart failure with preserved ejection fraction. *Nature Reviews Cardiology*, *14*(10), 591–602. https://doi.org/10.1038/nrcardio.2017.65

Fotaris, P., Mastoras, T., Leinfellner, R., & Rosunally, Y. (2015). From hiscore to high marks: Empirical study of teaching programming through gamification. *European Conference on Games Based Learning*, 186.

Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does Gamification Work? – A Literature Review of Empirical Studies on Gamification [ISSN: 1530-1605]. *2014 47th Hawaii International Conference on System Sciences*, 3025–3034. https://doi.org/10.1109/HICSS.2014.377

Hernandez, A. F., Hammill, B. G., O'Connor, C. M., Schulman, K. A., Curtis, L. H., & Fonarow, G. C. (2009). Clinical Effectiveness of Beta-Blockers in Heart Failure. *Journal of the American College of Cardiology*, *53*(2), 184–192. https://doi.org/10.1016/j.jacc.2008.09.031

Inc., A. (n.d.). CreatingSurveys Document. Retrieved March 12, 2022, from http://researchkit.org/docs/docs/Survey/CreatingSurveys.html

Kamat, T., Dang, A., Dang, D., Rane, P., et al. (2021). Impact of integrated medication reminders, gamification, and financial rewards via smart phone application on treatment adherence in uncomplicated type II diabetes patients: A randomized, open-label trial [Publisher: Medknow Publications]. *Journal of Diabetology*, *12*(4), 447.

Kannel, W. B., & Belanger, A. J. (1991). Epidemiology of heart failure. *American Heart Journal*, *121*(3, Part 1), 951–957. https://doi.org/10.1016/0002-8703(91)90225-7

Kardas, P., Lewek, P., & Matyjaszczyk, M. (2013). Determinants of patient adherence: A review of systematic reviews. *Frontiers in Pharmacology*, *4*, 91. https://doi.org/10.3389/fphar.2013.00091

Kociánová, E., Václavík, J., Tomková, J., Ondra, P., Jarkovský, J., Benešová, K., Václavík, T., Kamasová, M., & Táborský, M. (2017). Heart rate is a useful marker of adherence to beta-blocker treatment in hypertension [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/08037051.2017.1346458]. *Blood Pressure*, *26*(5), 311–318. https://doi.org/10.1080/08037051.2017.1346458

Kotecha, D., Flather, M. D., Altman, D. G., Holmes, J., Rosano, G., Wikstrand, J., Packer, M., Coats, A. J., Manzano, L., Böhm, M., van Veldhuisen, D. J., Andersson, B., Wedel, H., von Lueder, T. G., Rigby, A. S., Hjalmarson, Å., Kjekshus, J., & Cleland, J. G. (2017). Heart Rate and Rhythm and the Benefit of Beta-Blockers in Patients With Heart Failure. *Journal of the American College of Cardiology*, *69*(24), 2885–2896. https://doi.org/10.1016/j.jacc.2017.04.001

Krumholz, H. M., Parent, E. M., Tu, N., Vaccarino, V., Wang, Y., Radford, M. J., & Hennen, J. (1997). Readmission After Hospitalization for Congestive Heart Failure Among Medicare Beneficiaries. *Archives of Internal Medicine*, *157*(1), 99–104. https://doi.org/10.1001/archinte.1997.00440220103013

Martinho, D., Carneiro, J., Corchado, J. M., & Marreiros, G. (2020). A systematic review of gamification techniques applied to elderly care. *Artificial Intelligence Review*, *53*(7), 4863–4901. https://doi.org/10.1007/s10462-020-09809-6

Meyer, B. (2014). Agile [Publisher: Springer]. *The good, the hype and the ugly. Switzerland: Springer International Publishing*.

Octalysis: Complete Gamification Framework - Yu-kai Chou. (2020). Retrieved March 29, 2022, from https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/

Øye, I., & Levy, F. O. (2021). Betablokkere. Retrieved March 12, 2022, from http://sml.snl.no/betablokkere

Reisinger, D. (2019). Apple Watch Series 4 Outsold Every Other Smart Watch in 2018—in Just One Quarter. Retrieved May 11, 2022, from https://fortune.com/2019/06/25/apple-watch-smartwatch-sales-2018/

Ruppar, T. M., & Russell, C. L. (2009). Medication Adherence in Successful Kidney Transplant Recipients [Publisher: SAGE Publications Inc]. *Progress in Transplantation*, *19*(2), 167–172. https://doi.org/10.1177/152692480901900211

Scase, M., Marandure, B., Hancox, J., Kreiner, K., Hanke, S., & Kropf, J. (2017). Development of and adherence to a computer-based gamified environment designed to promote health and wellbeing in older people with mild cognitive impairment. *eHealth*, 348–355.

Sommerville, I. (2016). *Software engineering* (10. ed., global ed). Pearson.

Sommerville, I. (2020). *Engineering Software Products: An Introduction to Modern Software Engineering*. Pearson.

Spaulding, E. M., Marvel, F. A., Lee, M. A., Yang, W. E., Demo, R., Wang, J., Xun, H., Shah, L., Weng, D., Fashanu, O. E., Carter, J., Sheidy, J., McLin, R., Flowers, J., Majmudar, M., Elgin, E., Vilarino, V., Lumelsky, D., Bhardwaj, V., . . . Martin, S. S. (2019). Corrie Health Digital Platform for Self-Management in Secondary Prevention After Acute Myocardial Infarction: MiCORE Rationale and Design. *Cir-*

culation: Cardiovascular Quality and Outcomes, *12*(5). https://doi.org/10.1161/CIRCOUTCOMES.119.005509

Støstad, M. N. (2017). Tallet som har blitt en mani. Retrieved March 25, 2022, from https://www.nrk.no/slukt-av-snap-1.13811834

Appendices

A Source Code

The complete source code of the application is available on GitHub.

A.1 MedRem

https://github.com/sofieurhaug/MedicineReminder