Frankfurt University of Applied Sciences

Software Engineering Project

GOP(Gauntlet of Power) Smartwatch Project

Authors/Matr. Nr.:

Karthik Bhat

Felix Dallmeyer

The Phi Tran

Arjola Cara

Zhiyu Li

Wenxue Zhao

Supervisor:

Prof. Dr. Jörg Schäfer

Prof. Dr. Peter Thoma

Final Report for the Software Engineering Project – Smart Watch

Team 3

Software Engineering - Analysis

January 17, 2024

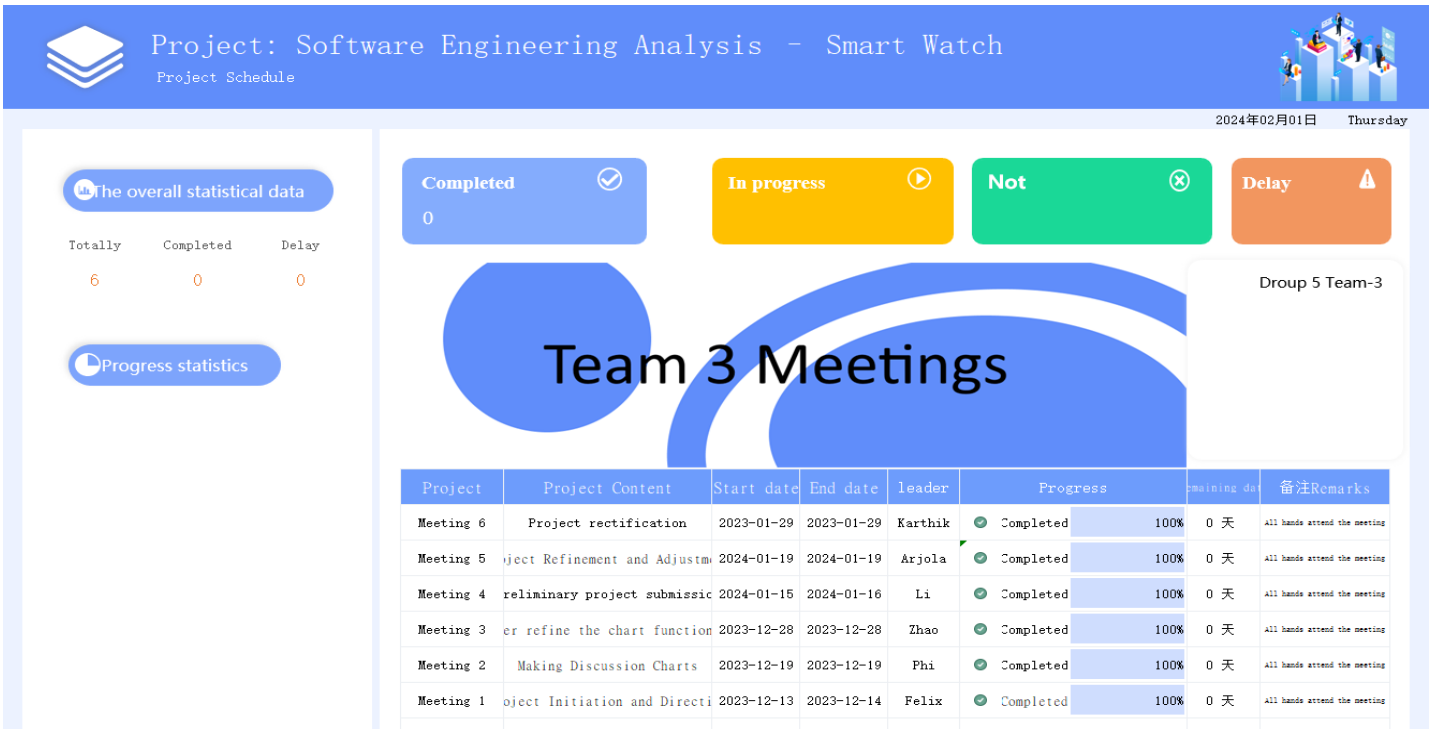
1. Introduction

We started working on a beginner-friendly smartwatch, aimed at a broad audience with its simple controls and general usefulness. This is the final report of Team 3.

In this document, each member's name is added on the first page of the section they worked at.

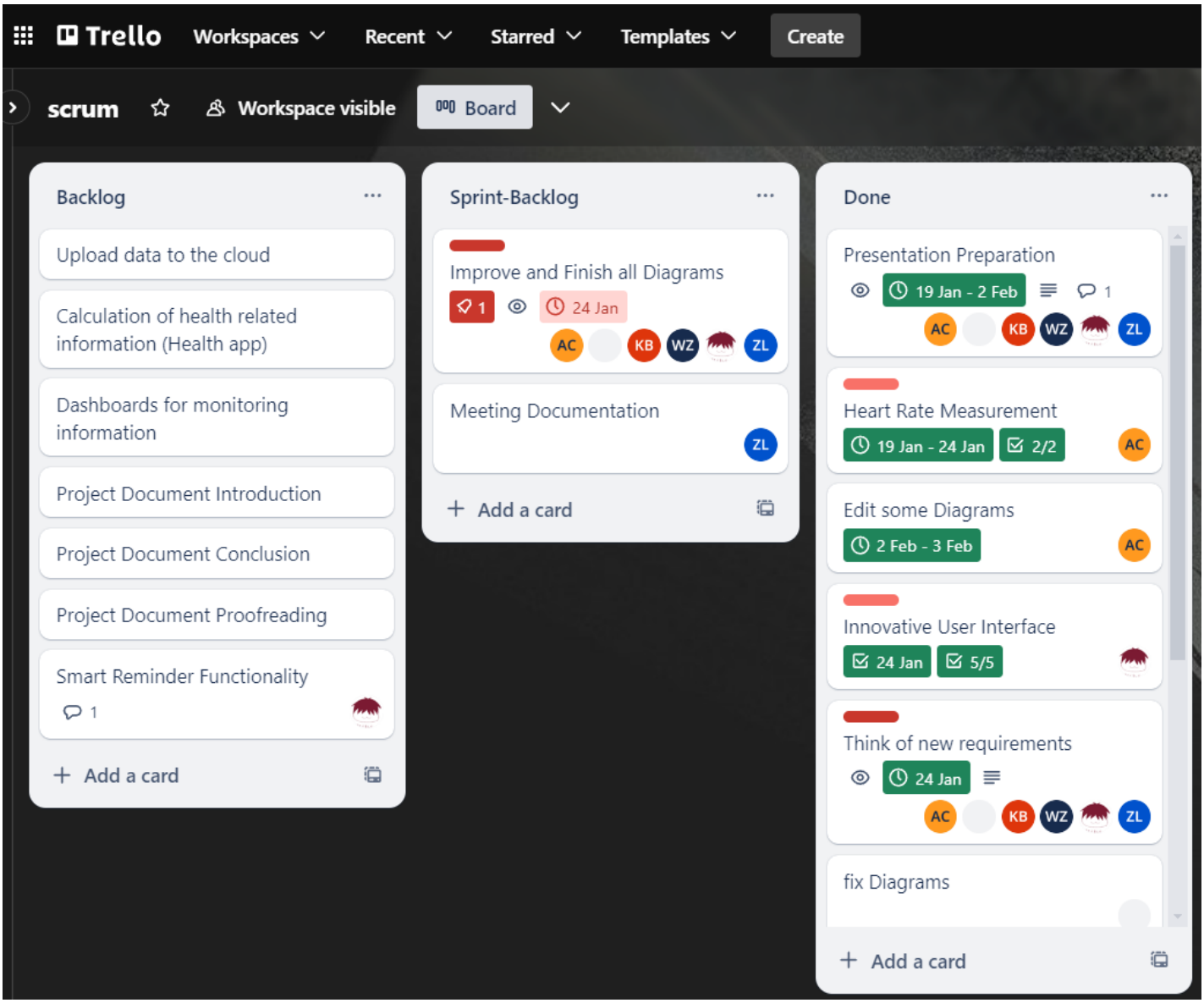
1.1 Meeting Schedule

Our meetings were primarily held online since most member had tight schedules. We met up in person to discuss the most important changes in our project right before Christmas and after the preliminary report. The documentation was done by Zhiyu Li.



1.2 Sprint Meeting using Trello

We also utilized the Trello tool to make our work more organized and streamlined. Everyone created tasks for themselves, assigned them, and of course, added a due date. Here, we consistently worked in sprints.



1.3 Backlog Items

ID	Main Actor	Title	Category	Business Value	Sprint	Team Member
1	user	Intuitive User Interface	Functional	high	1	Phi
2	user	Text Input	Functional	medium	2	Phi
3	user	Measure Heart Rate	Functional	high	1	Arjola Cara
4	user	Emotional Intelligence Analysis	Functional	high	1	Arjola Cara
5	user	Personalized Fitness Routines	Functional	medium	2	Arjola Cara
6	user	Health Assessment and Early Detection	Functional	high	2	Arjola Cara
7	user	Health App	Functional	high	1	Felix Dallmeyer
8	user	Workout App	Functional	high	2	Felix Dallmeyer
9	user	Reminder App	Non Functional	medium	3	Felix Dallmeyer
10	user	upload data to the cloud	Functional	high	1	Zhiyu Li
11	user	Mobile phone information cloud backup	Functional	high	2	Zhiyu Li
12	user	Medical Records Integration	Functional	medium	3	Zhiyu Li
13	user	Location Tracking	Functional	high	1	Karthik Bhat
14	user	Advanced Navigational System	Non Functional	mid	1	Karthik Bhat
15	user	Time Tracking App	Functional	high	2	Karthik Bhat
16	user	Dashboards for monitoring information	Functional	high	1	Wenxue Zhao
17	user	User Social Interaction Features: Comments and Likes	Functional	high	2	Wenxue Zhao
18	user	User Social Interaction Features : Social Sharing	Functional	high	2	Wenxue Zhao

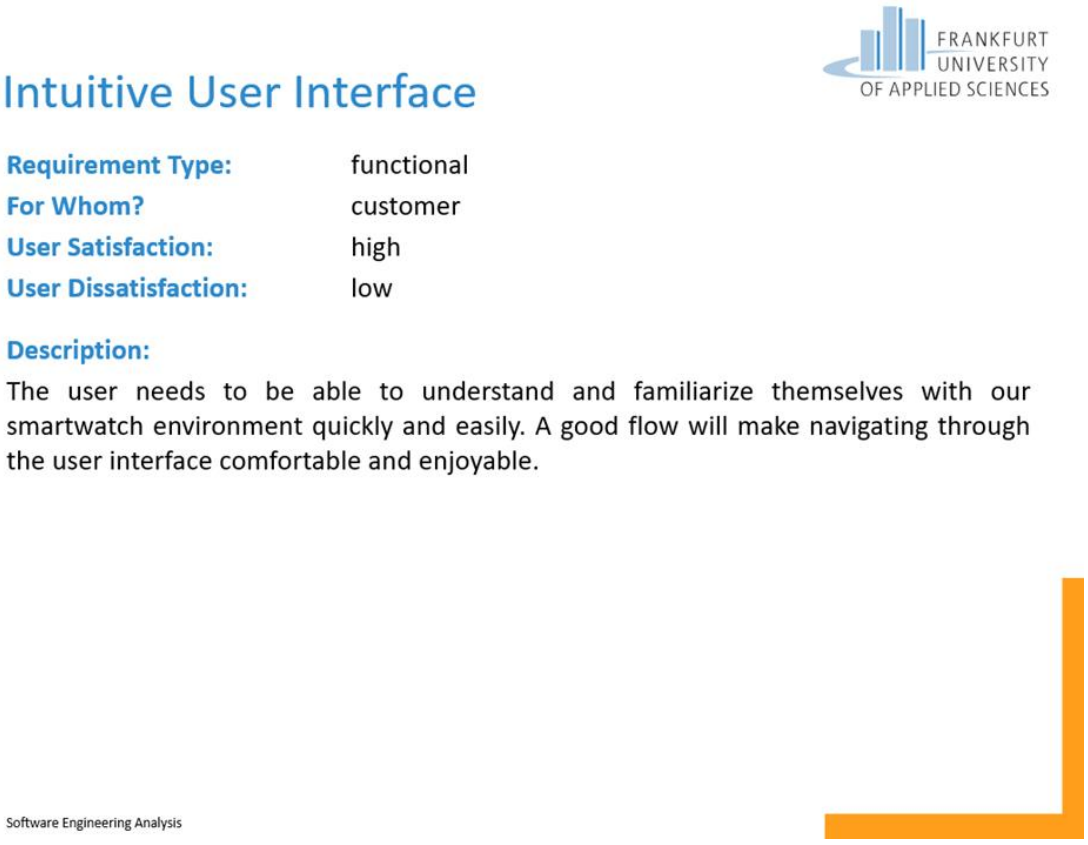
The Phi Tran

2. Intuitive User Interface

2.1 Introduction

The user interface is what brings the smartwatch to life and it's the thing that users can really see and interact with almost all the time. Thus, it is important to create a good experience for the user in making the User Interface as intuitive as possible. To achieve an intuitive User Interface that is both easy to use and straightforward, I decided on a customizable cycle of home screens that lets users access their commonly needed applications within a few swipes.

Starting into it, I thought about what actions can be achieved with clicks and single swipes. Because most touchscreens don't exceed the basic controls, thinking about fancy controls like a two-finger swipe or circles will not be necessary. It will also add an unnecessary layer of complexity to the interface. Another control action that will work in tandem with the basic control is having a button on the side of the smartwatch act as a "back" button for the interface, as well as implemented voice control. Since our screen on the smartwatch is significantly smaller than even a smartphone, it is important to reduce the information greatly and only show the most important bits. Elements must be maximized but still leave enough room for others.



2.2 User Stories

First, we created user stories to help us model out the most important parts of our GUI. These user stories give us a small insight into details of our smartwatch.

Focus on Fitness

As a fitness enthusiast, I want to quickly access the Health Dashboard screen on my smartwatch to view essential health data such as step count, calories burned, and distance covered for the day.

Focus on Heart Rate

As a user concerned about my heart health, I want to easily check my heart rate trends on the Heart Rate screen, which should display a graphical representation of my heart rate throughout the day.

Focus on GPS

As someone who frequently uses navigation apps, I want a straightforward GPS Navigation screen on my smartwatch, allowing me to mark a destination, view the map, and start navigation with minimal effort.

Focus on Customization

As a tech-savvy user, I want to personalize my smartwatch's main home screen by customizing the watch face with options like theme colors, backgrounds, clock layouts, and fonts.

Focus on Notifications

As a busy person, I want a simple clean way to view and manage my notifications, as well as review them without trouble.

Focus on Simple Design

As a user who values simplicity, I want the screen of my smartwatch to display essential information like time, date, heart rate and fitness trackers in a simplified manner.

2.3 Standard Usage of the User Interface

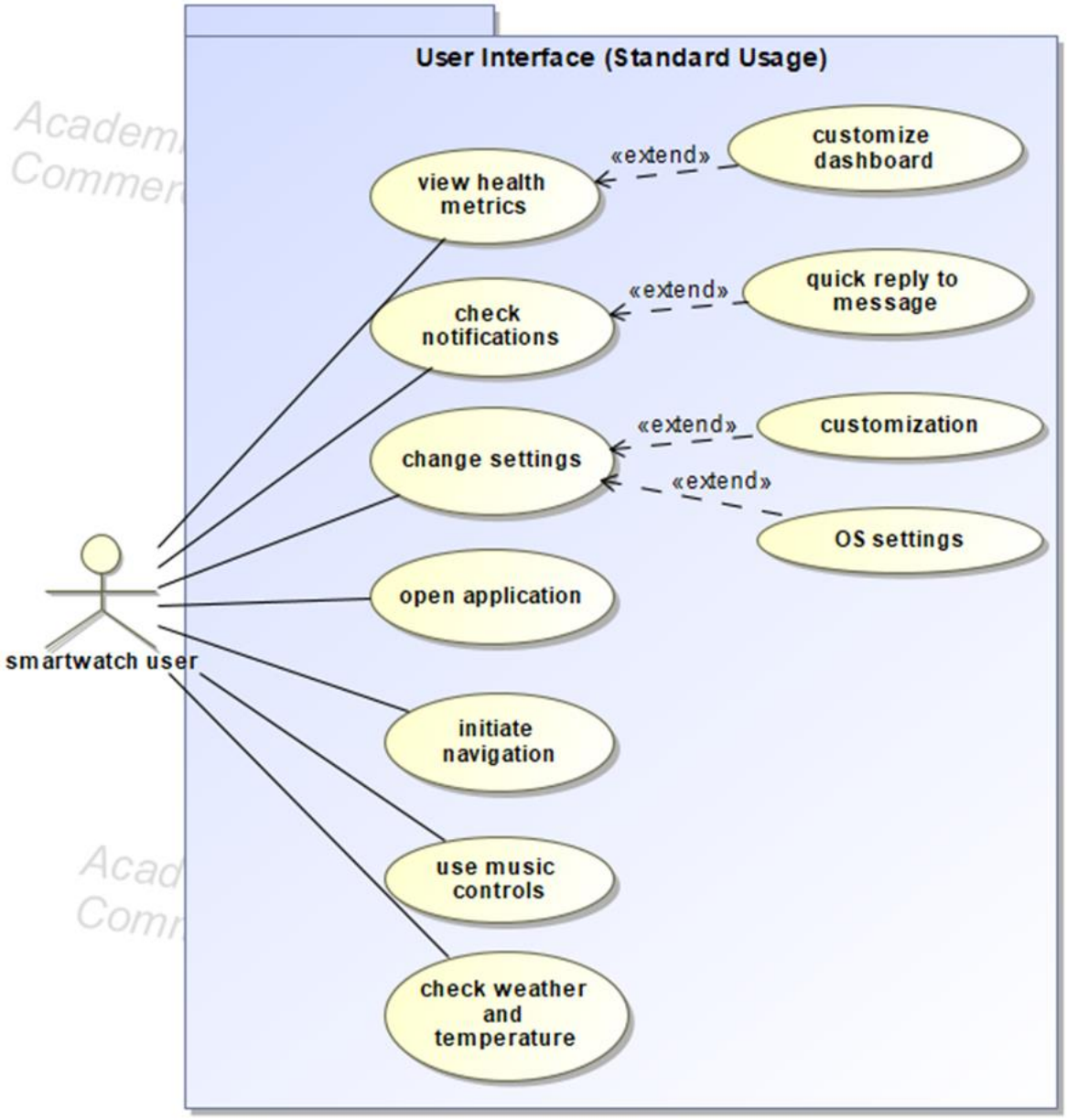
In this simple use case for opening an application, the interaction between user interface is very straightforward. The GUI only acts as a link between the user and the application.

Name	Opening an application
ID	1
Descrip- tion	The contractor opens up an application on their smartwatch
Trig- ger	The contractor wants to open up a specific app
Ac- tors	User, System
Pre-conditions	App installed, (For Flow A: user has created a widget)
Post-conditions	The user has opened the application and switched to the application screen
Basic Flow **Description** **Actions** **1** **2**	This is the main scenario where user opens up the application through the app list The user opens the app list The user clicks on the application icon
Alternative Flow **Description** **Actions** **1** **2**	A The user has created a widget on the quick-access home screens The user swipes to the needed home screen The user clicks on the widget to open the application
Alternative Flow **Description** **Actions** **1** **2** **3**	B The user uses the voice control function The user activates the voice control The user tells the voice control to open the application The system will redirect the user to the application

2.3.1 Use case diagram

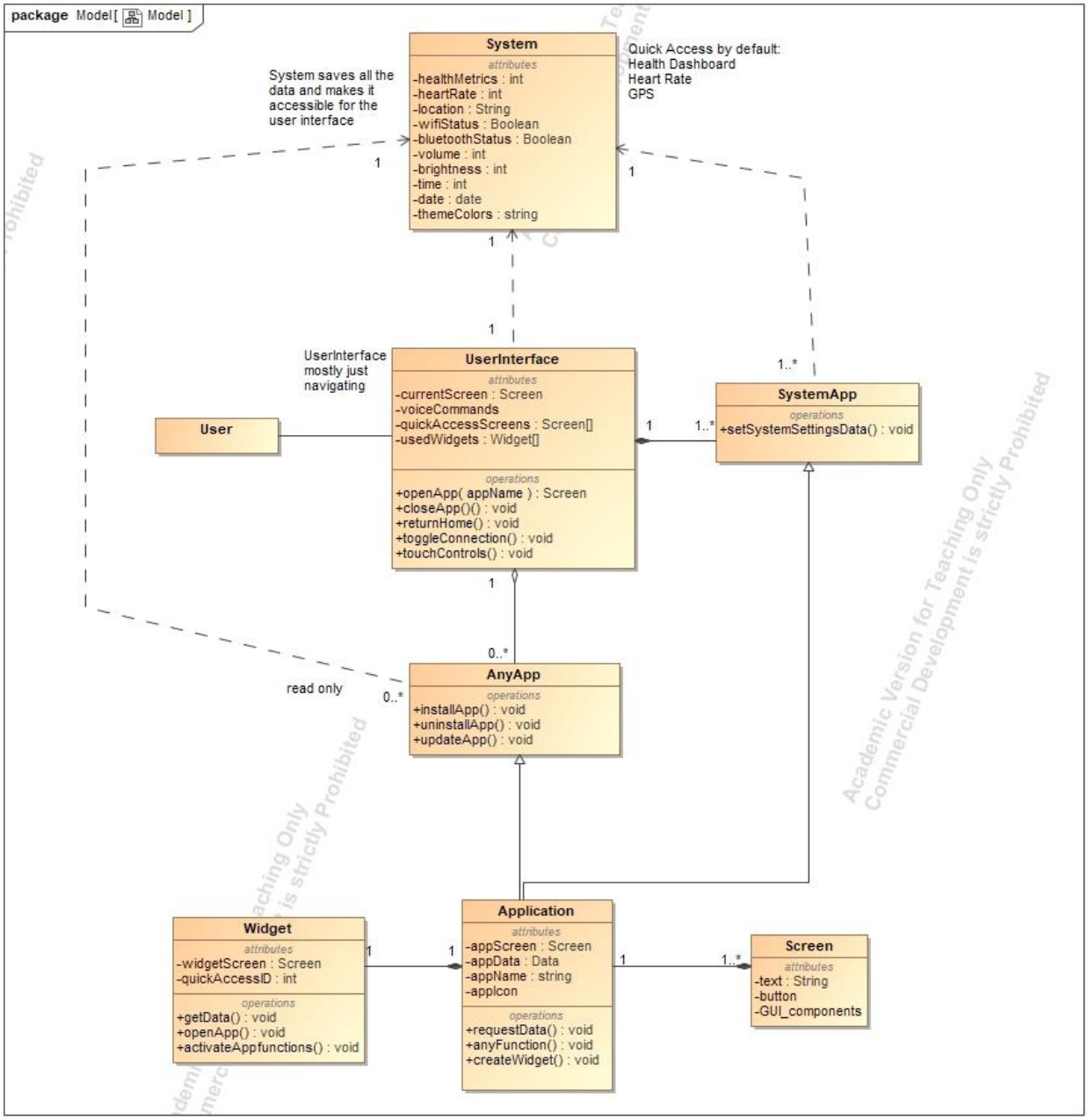
This use case diagram showcases how users would interact with the interface. There are a variety of functions the user can access directly through the interface. These functions are very straight-forward and require just a few taps and swipes on the touchscreen.

They include displaying data, turning functions on or off, opening applications and navigating through the user interface.



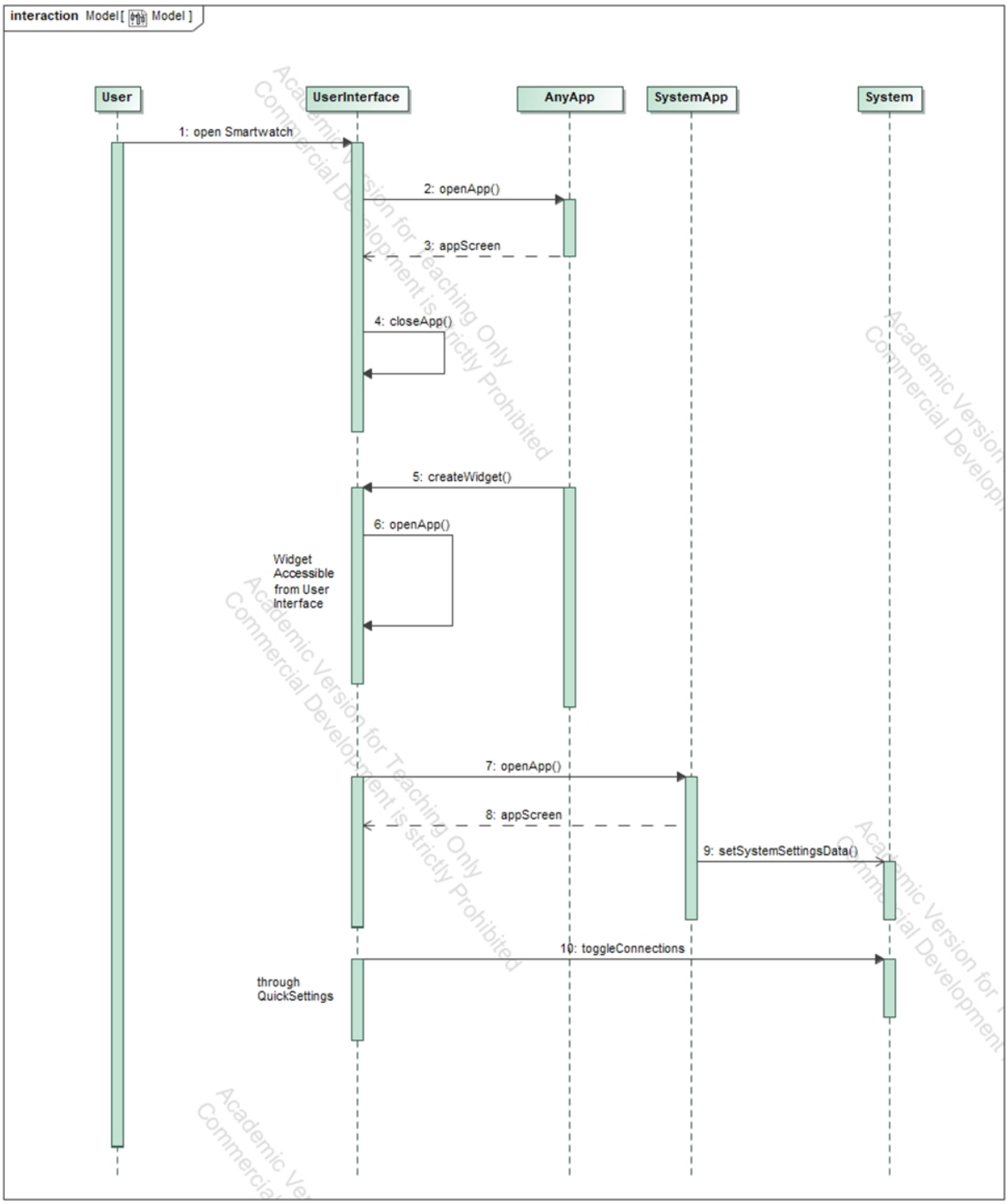
2.3.3 Class Diagram UI

The class diagram consists of a UserInterface class that acts as a bridge between user and system, as well as the applications. It holds a the currentScreen variable that takes a Screen object and displays it on the smartwatch. A screen object can be created from any Application object. The System class will hold all technical data and make it accessible for the UserInterface. The AnyApp class and the SystemApp class are both derived from the Application class and differ in the type of access towards the System class. All Application objects will also be able to create widgets that are accessible in the user interface.



2.3.4 Sequence Diagram UI

The sequence diagram describes how the different classes interact to open an application and to toggle a connection. The User interacts with the UserInterface and from there accesses applications and the System.



2.4 UI Design

Smartwatches only have limited screen space, so we need to focus on simplifying the design as much as possible.

First, we need to have all the essential functions at the front. Chances are, the user bought a smartwatch to be health conscious and keep track of their data throughout the day. Heart Rate and Fitness Tracker applications are a must have to keep on the quick access screens. Additionally, we believe the GPS application will be of great value to keep on here too. If the user needs other apps on quick access, they can add those applications as widgets and rearrange the order of the quick access screens.

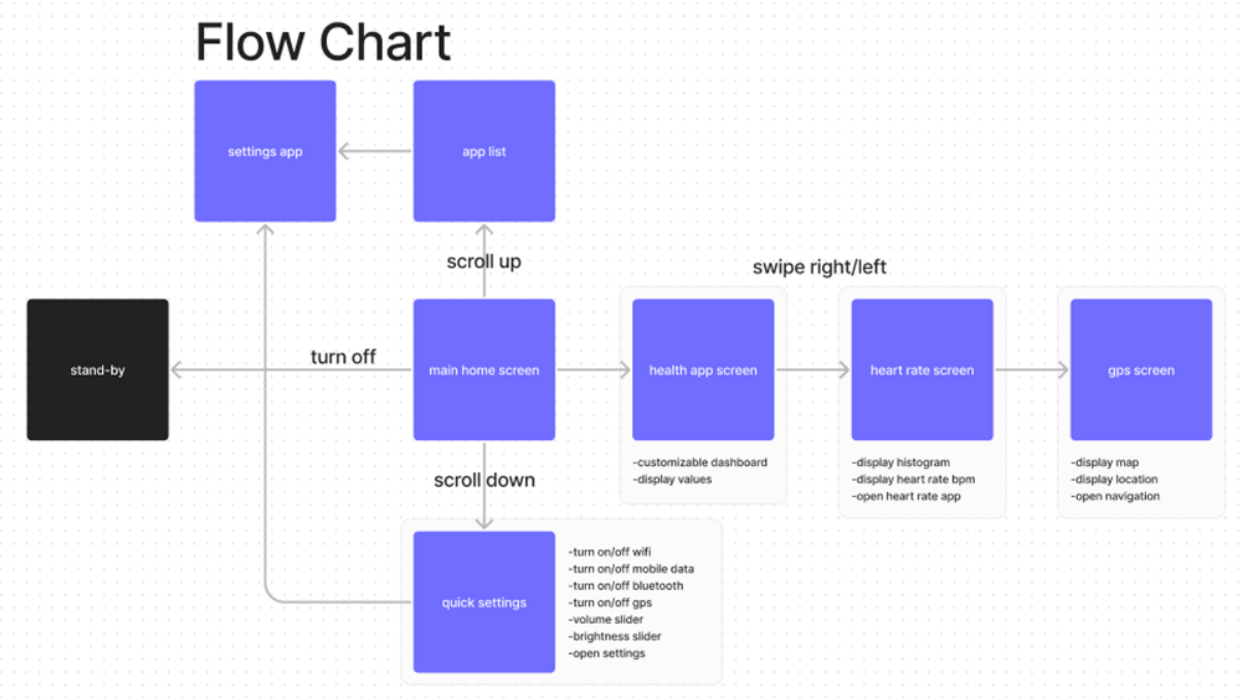
To prevent our screen from getting clustered by UI elements we need simple and recognizable icons, short and precise texts and clear visual cues. They will appeal more to our user and make our smartwatch easy to use. Colors, shapes and fonts have to be consistent in every screen and distracting elements are to be avoided.

As for navigation, we believe the page-based model to be superior. It incorporates the simple gesture of swiping to change the screen. It makes our navigation paths quick and concise. Therefor a flowchart would be more suitable for visualizing how a user would move instead of an activity diagram.

2.4.1 Flow Chart UI

The flowchart works by standardizing the navigation through the screens through the basic controls clicking and swiping so we can just look at how the interface environment is built.

When you open up the smartwatch, you begin at the main home screen. From here, you can start swiping to access another screen. Left and right for quick access to personalized app screens, up to open the app list, and down to access quick settings. When the screen is turned off, the interface will turn into standby mode, where power will be preserved, and time and date are displayed. The following flow chart shows a quick overview of where you can navigate starting from the main home screen.



This will be the minimum number of screens for our smartwatch, as we cover the basic requirements of our smartwatch with these. We modeled the smartwatch to behave similarly to a phone so that the user will feel familiar with the interface environment immediately. I created a user interface prototype to simulate a possible outcome.

2.4.2 UI Prototype

For further visualization we created a smartwatch user interface prototype. It contains a variety of screens. We look at the most important ones.

Standby

The standby screen is set to display information during screen inactivity. Here, the smartwatch should behave like a regular clock, while it can also make use of its digital environment by showing even more information that requires minimal processing like displaying the date or unread notifications. It's necessary to keep it simple as it will be the most-used screen for an important purpose: telling the time.

Watch face (Main Home Screen)

The main home screen can be more personalized. The user needs to be able to connect with their smartwatch emotionally. By giving personalization options, the user can represent their style on the smartwatch and, as a result, will like it more. This can be achieved by giving options to change things like theme colors, backgrounds, clock layouts, and fonts.

Having the default configuration appealing to the user is just as important, though, as it is a first impression of our smartwatch. I believe a simple modern look with a pleasant color scheme will appeal to a majority of our customers already, because it's easy to look at while keeping critical information at the front.

The next screens show the basic requirements of our smartwatch. They are put a few swipes away from the main home screen for quick and easy access. The order of the screens can be changed. These screens are designed with interactive elements on the bottom and displaying elements at the top so that the top elements are not obstructed by your finger touching the screen.

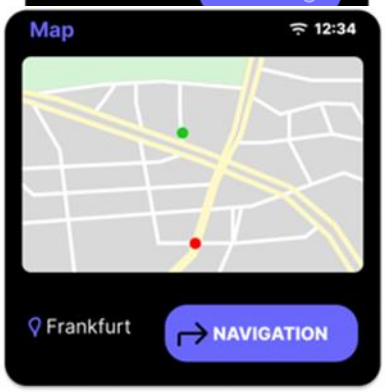
Health Dashboard Screen

The health app screen displays a dashboard of the most important health data for each day. Linear values are displayed with visual elements as well as the decimal numbers, so users can get a better feeling of units and sizes. I created the prototype with a circle bar that fills up the higher the value goes, where the end of the bar signals the daily goal. Which data is shown and what the daily goal is can all be set by the user, depending on their current needs.

Heart Rate Screen

On the heart rate screen, the measured heart rate will be tracked over the course of a day and drawn on a graph. The smartwatch saves the peak and the bottom value of the day and displays them at the bottom of the screen, while also keeping track of the current heart rate value.

GPS Navigation Screen

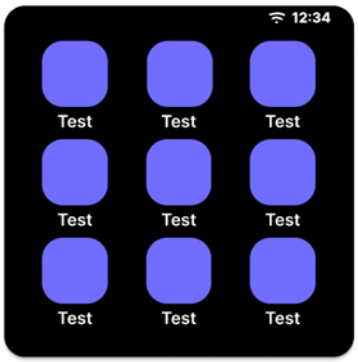


The GPS screen draws a map on the screen which can be interacted with. You can mark a destination on the map and start the navigation with the button below. On the left, the display also shows the current location. When the GPS is activated, your own location will also be marked on the map.

We put these applications into the quick access screens as they are the most useful functions for our users, which they will use every day. Accessible through just a few swipes we keep the interface easy and intuitive.

Now, we look at the general screens. They build the bridge to the applications, so it is important to make them easy to understand as well as capable of showing direction. By showing a direction, I mean because the user wants to do a certain thing or open a certain app, we set the path so that the user can instantly see where to click and swipe to navigate through. We achieve that by using recognizable icons and clever labeling.

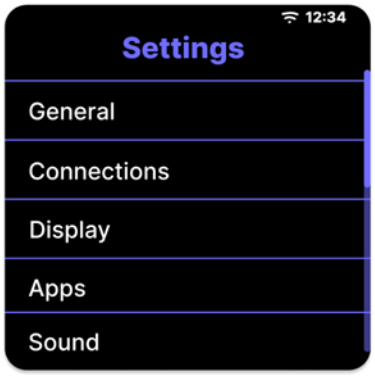
App List Screen



If we swipe up from the home screen, we open up the app list. With a grid design, we can keep the shown apps at a maximum while also keeping a good overview of the apps by only showing 3x3 at a time. You can access all the applications from the health app to GPS, as well as additional apps like the settings application. Apps can be downloaded from a store and will be sorted alphabetically on the grid. When there are more than 3x3 apps, the app list becomes scrollable.

Quick Settings Screen

To open up the quick settings, you can swipe down from the home screen. Here you can manage connections, volume, and brightness with just a few clicks and swipes. At the bottom of the screen, you can also open up the settings application where you can change the appearance and behavior of your smartwatch to your preferences.

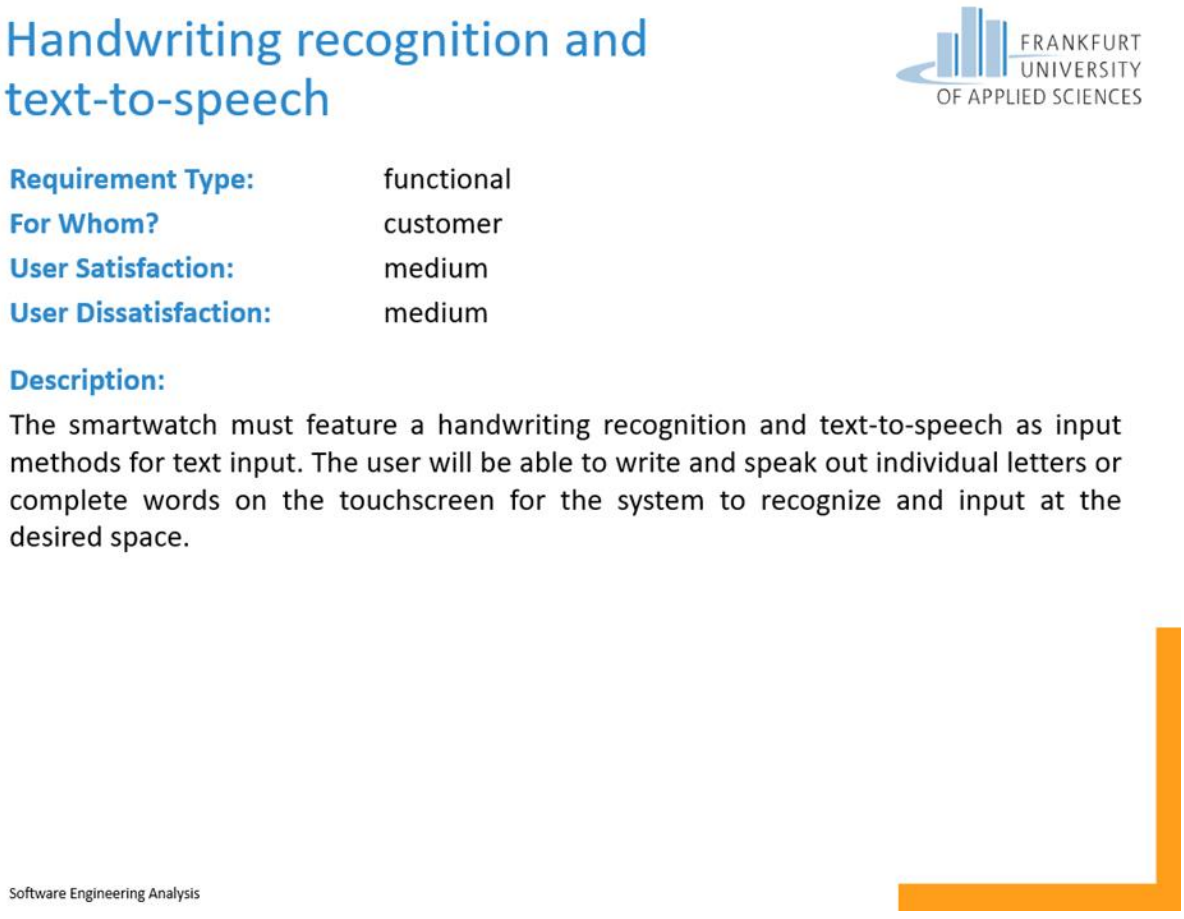


Settings Screen

The settings app is displayed as a list where different categories are labeled, depending on their type of setting. We have general settings, for example, where you can set time and date or the language.

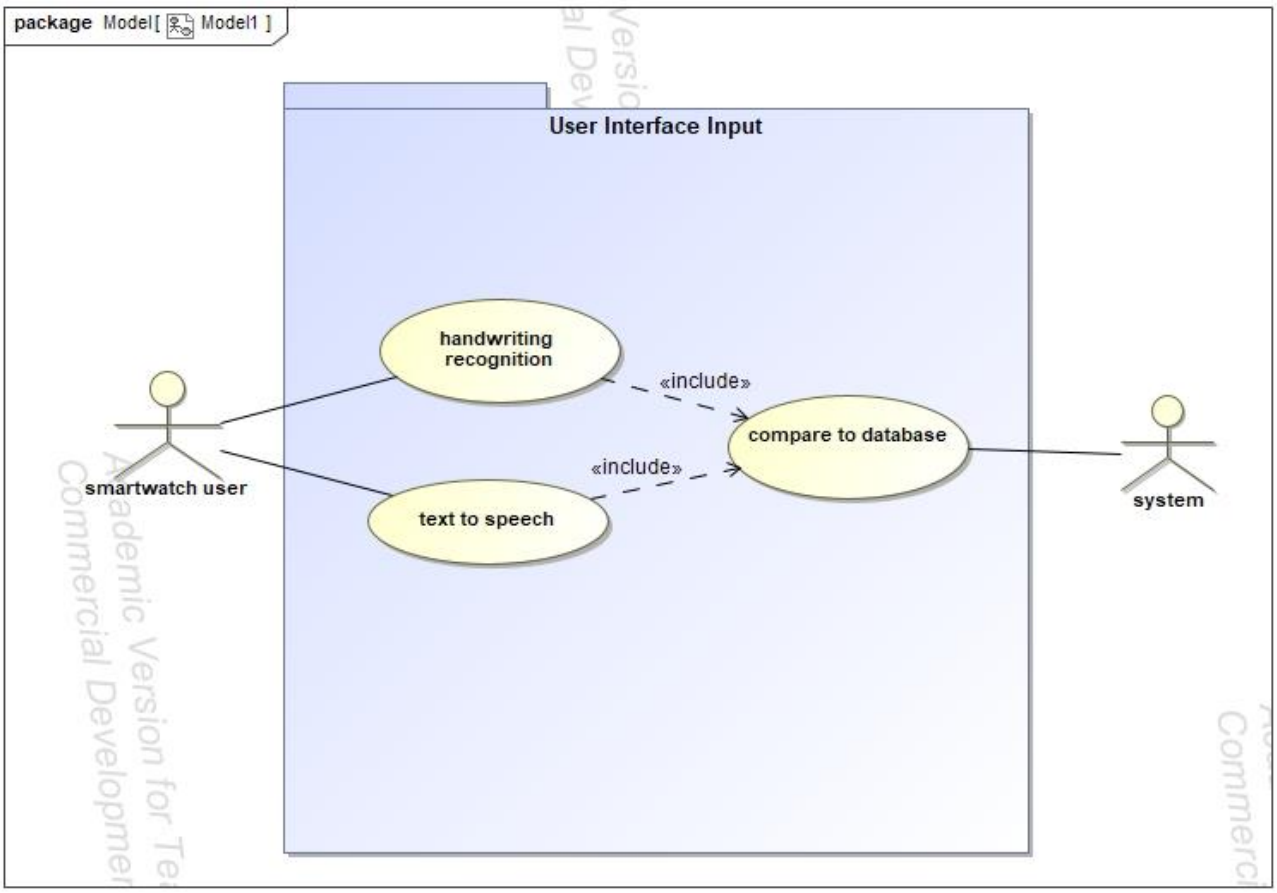
2.5 Text Input

There are many apps that require input by the user, for example when searching something or logging in somewhere. Usually, you would do so on your connected mobile phone, but in case you cannot take it out at the moment, or you do not have it on you, you still want to be able to use those functions. Since our smartwatch screen is too small for placing a typical onscreen keyboard, we thought of implementing handwriting recognition on our smartwatch screen in addition to text-to-speech. The user will be able to write single letters, digits and symbols that are picked up by the smartwatch and inputted at the desired space.



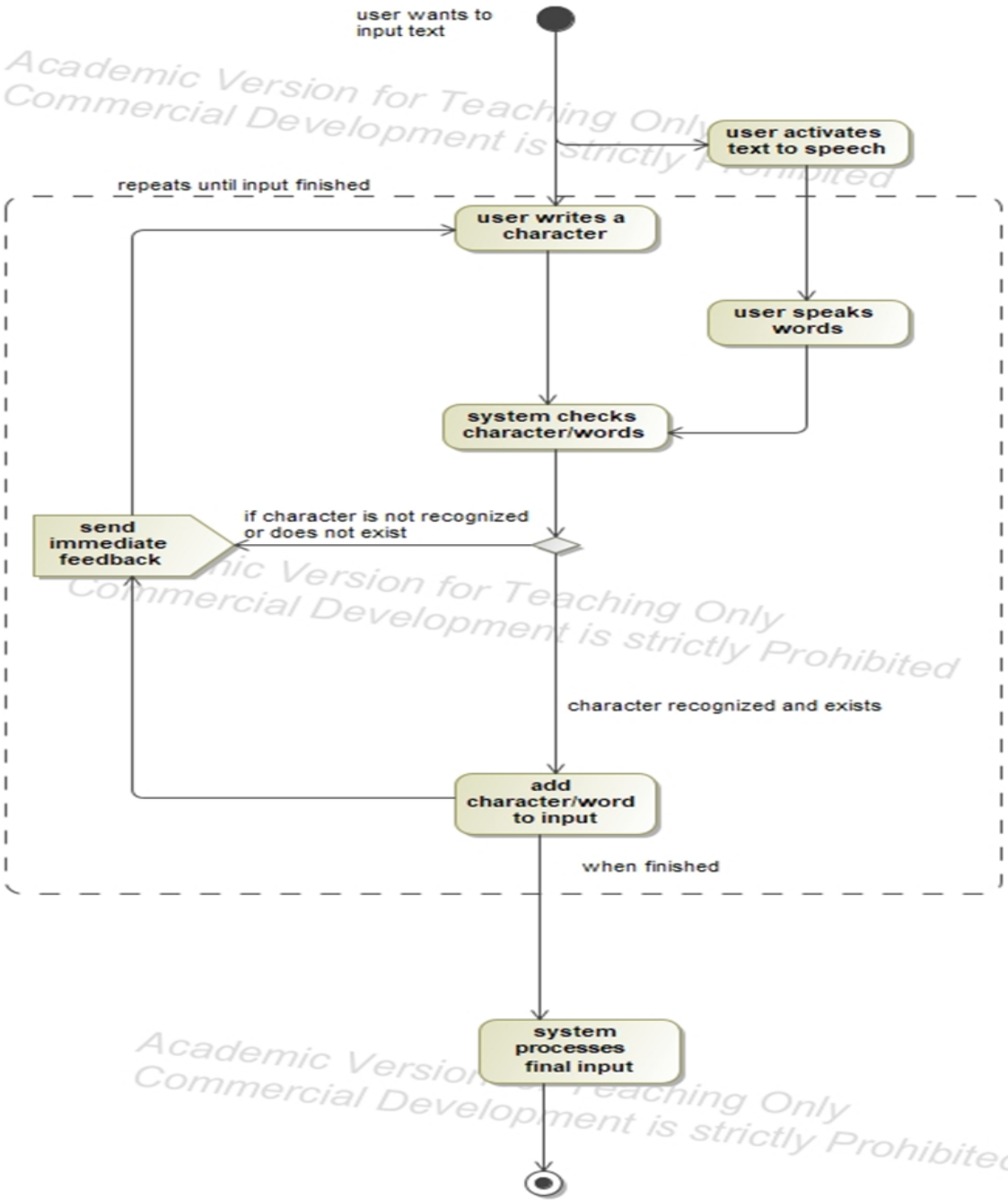
2.5.1 Use Case Diagram for Text Input

This use case diagram shows the user interacting with the user interface input via handwriting or text to speech. Using either of those includes the comparison to the database that will be initialized by the System.

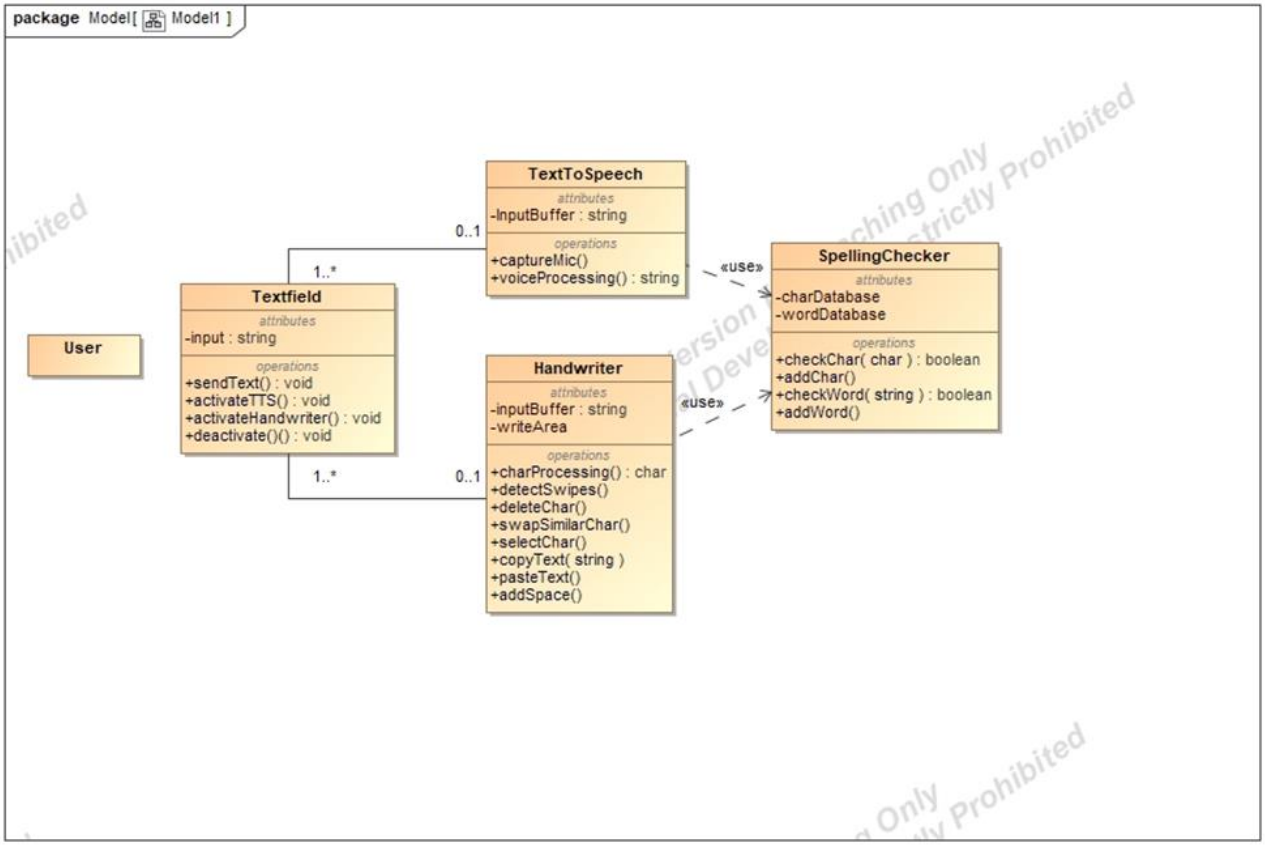


2.5.2 Activity Diagram Text Input

This activity diagram shows the loop, in which the system checks the user input and, in the case it is recognized, it will be saved as input and displayed on the screen. Immediately after that, the user receives feedback depending on whether the input was accepted or rejected. This feedback can come in the form of blinking edges and vibrations.



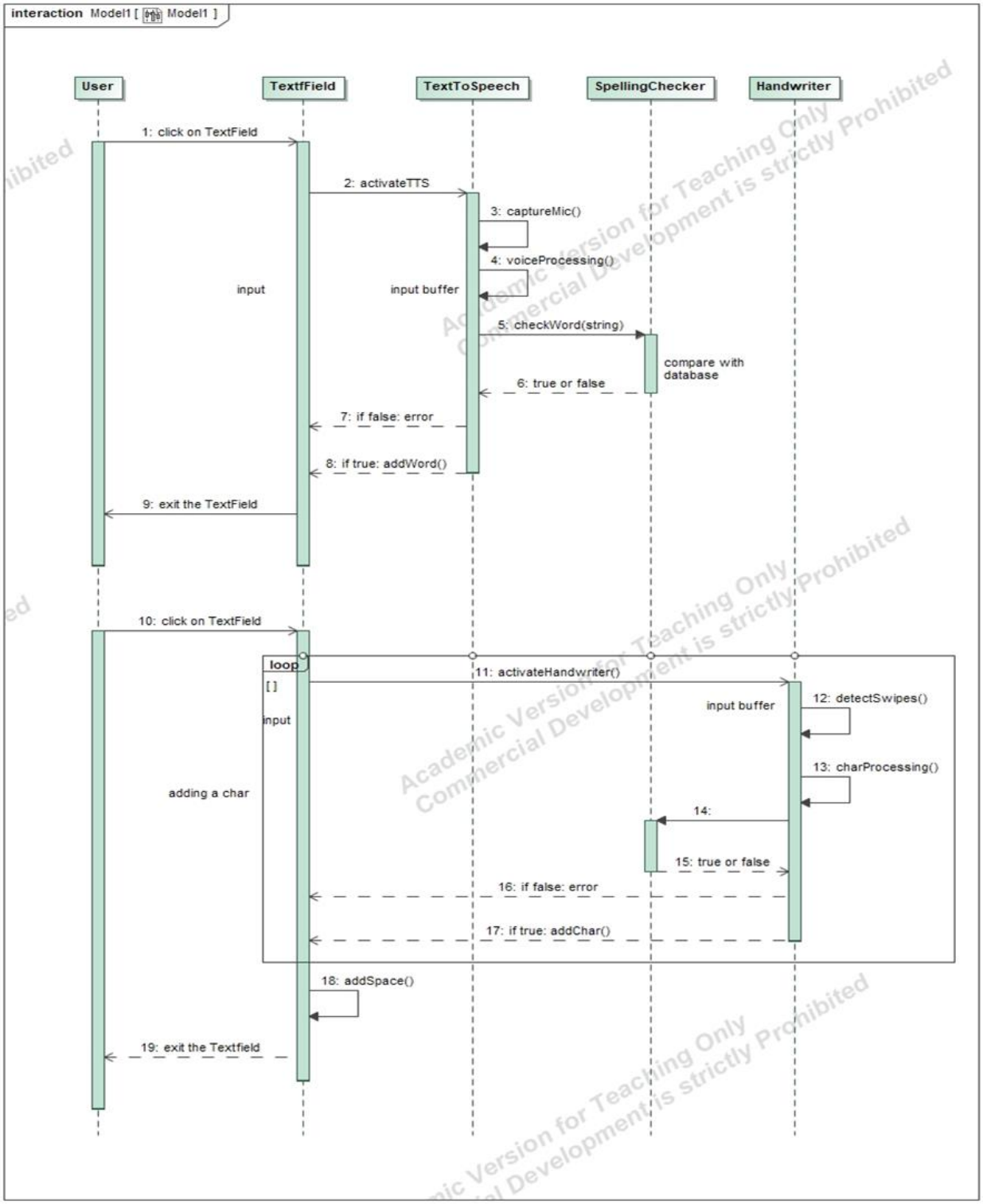
2.5.3 Class Diagram Text Input



In this class diagram we have a TextField class that either uses the TextToSpeech or the Handwriter class to create a string for the input. In both classes, they make use of a SpellingChecker class that confirms the existence of the spoken word or letter. If it does, it will get passed on into a buffer which will later be passed on again into the TextField input.

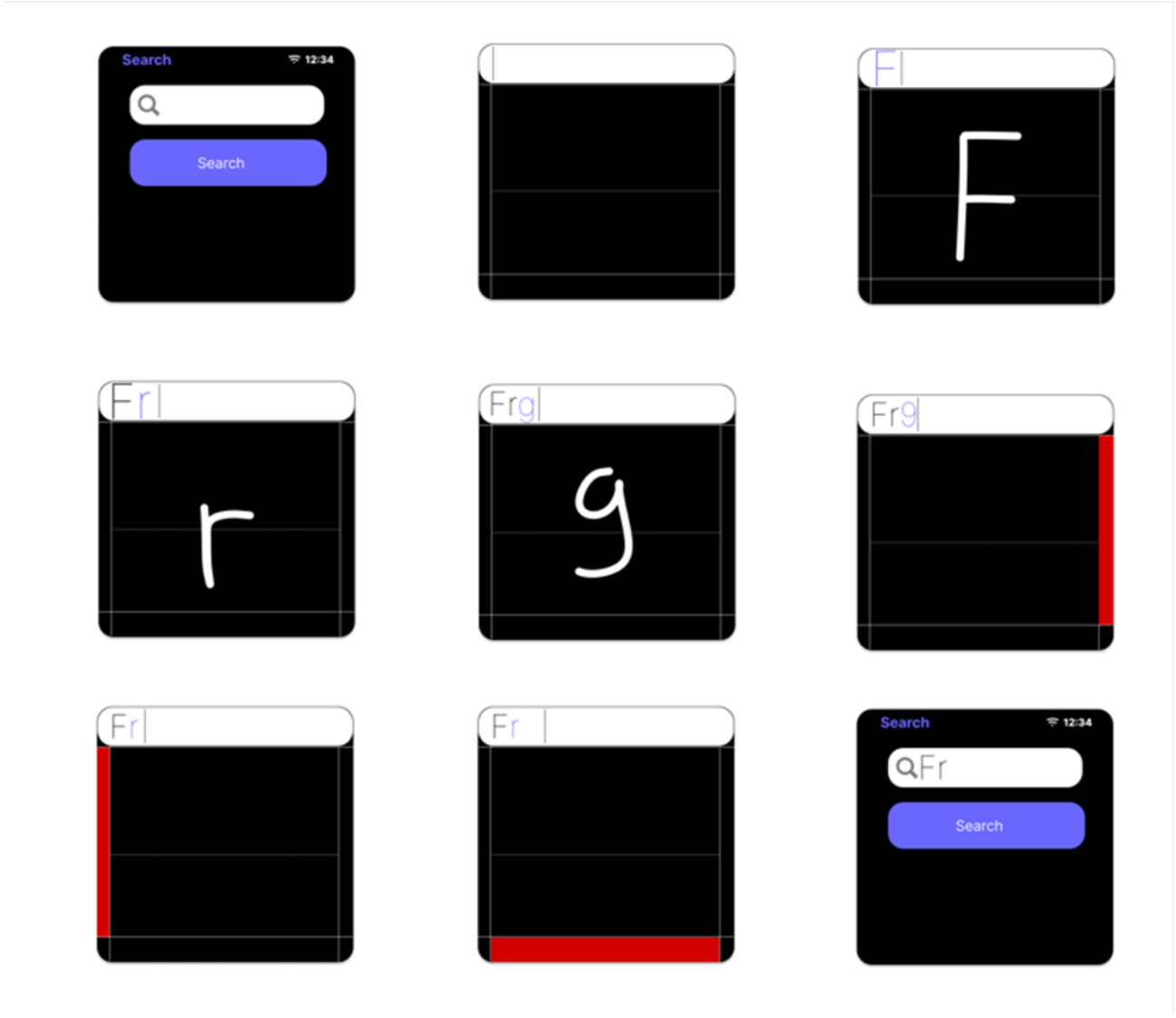
2.5.4 Sequence Diagram Text Input

The sequence diagram describes the usage of the TextToSpeech and the Handwriter class to create an input inside the TextField.



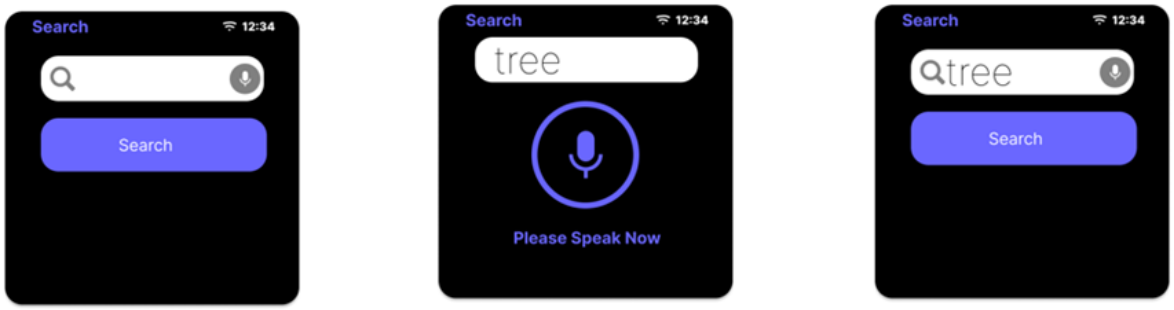
2.5.6 Handwriting Input Example

Here is an example of using the handwriting feature of our smartwatch. As the user clicks on the search bar, it opens a different screen where the user can write the letters into the buffer textfield, where they are highlighted. The letter will stay on the screen until another letter is written. If there are similar characters like 9 and g the system will first use the character that seems to fit into the field. Since there are only letters in the text field the system puts a g in. If the user then wishes to have the other option, they can press the right side of the screen to cycle through the characters. If the user pressed the left side of the screen, the last written character will be deleted. If the user presses the bottom side of the screen a space will be added as the last character. To return, the user can press the smartwatch button. The user can now initiate the search via the search button.



2.5.7 TextToSpeech Input Example

To input an input via TextToSpeech, the user can press on a button, displayed on the Textfield to start up the TextToSpeech. From here the microphone will pick up any word said by the user, process it and put it into a buffer textfield, if that word exists. To return, the user presses the smarwatch button and they can now initiate the search with the search button.



Alternatively, the activation of the TextToSpeech can be integrated into the smartwatch button. Press it twice and the smartwatch microphone will be activated to receive the spoken words and the interface automatically fills them into the text field.

These input examples will work on any text field the user might encounter and provides more control, making the smartwatch work as a standalone too. They will be especially useful when using the GPS, searching something on the web, making small notes (for example grocery list), quickly replying to messages, typing in your credentials and in creating calendar entries.

2.6 UI Goal

In conclusion, our smartwatch UI is designed to provide an effortless experience for users of all levels.

With our user interface, the goal is to attract customers with an easy-to-learn and use but extensive environment to work with. We emphasize a lot on quick access to health and fitness features and ensure that users can effortlessly monitor their health metrics with just a few swipes and taps, thanks to our intuitive navigation.

Our smartwatch also allows users to change the appearance and themes to match their unique style preferences, as well as swap quick access applications to fulfill their needs.

The ability to input text directly on the smartwatch will help distinguish our smartwatch from others, making it an independent and innovative device, not just an extension of a phone.

A well-designed user interface is the basis to creating a good smartwatch.

Arjola Cara

3. Real Time Sensor Data

In my requirements, the focus is on the heart and how heartbeats react to various events in life.

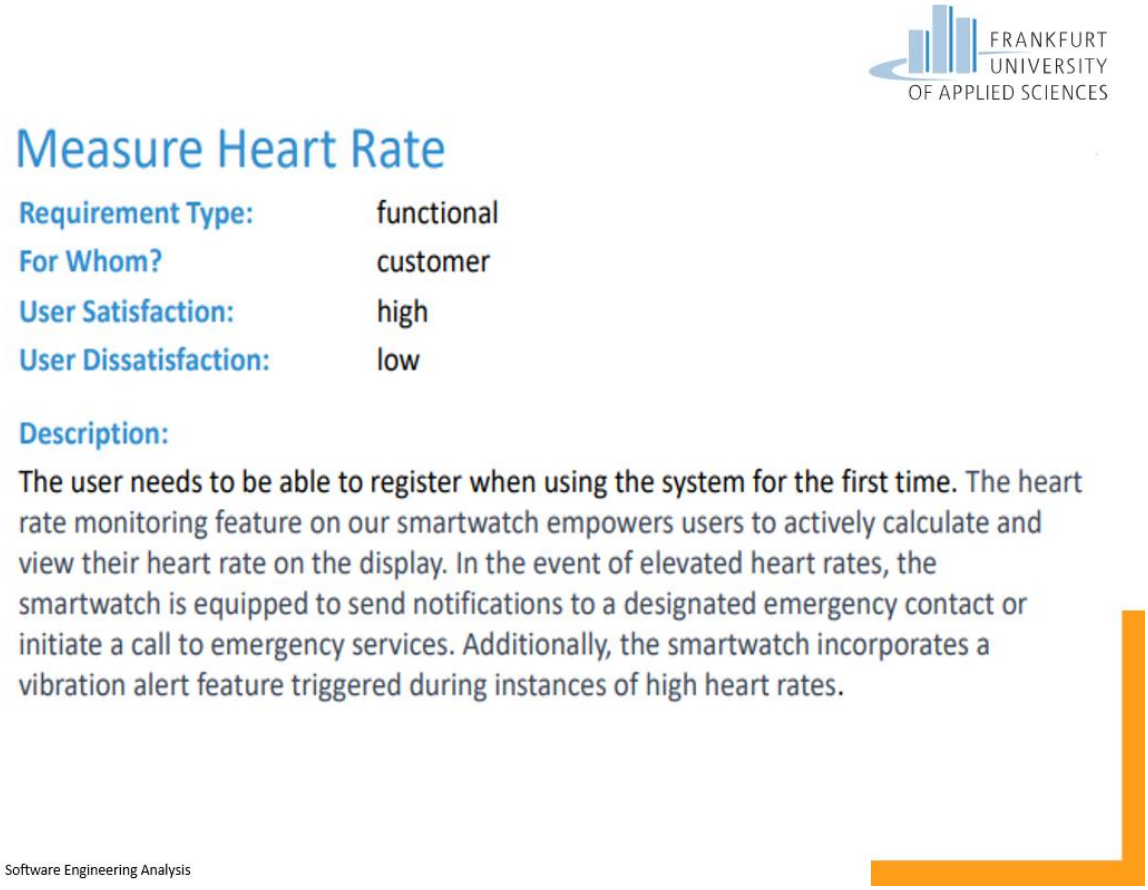
The smartwatch analyzes your emotional state by monitoring factors like skin temperature. Simultaneously, it continuously tracks your heart rate. If you experience stress or excitement, your heart rate may increase. The smartwatch examines this correlation and notifies you when your emotions impact your heart rate. This functionality provides insights into how your feelings influence your physical well-being.

Additionally, this feature can be particularly beneficial for older individuals, especially those with heart conditions. It serves as a proactive tool to monitor and potentially detect heart-related issues, even before they occur. In fitness scenarios, the smartwatch can also assist by sending notifications when heart rates are too high. In such cases, the smartwatch may recommend stopping the exercise or slowing down. This function helps make fitness both safe and effective, providing an additional safety measure for heart health.

3.1 Requirement: Measure Heart Rate

Introducing a feature in smartwatches that can detect signs of unconsciousness and automatically send a message to a trusted contact or initiate an emergency call. This functionality could be a lifesaver, especially for older individuals or those at higher risk of sudden health issues. The ability to swiftly call for help or notify a person in emergencies provides an additional layer of security, particularly for individuals living alone or concerned about their health. Such features highlight the potential of smartwatches not only as fitness trackers but also as life-saving health tools.

3.1.1 Snowcard



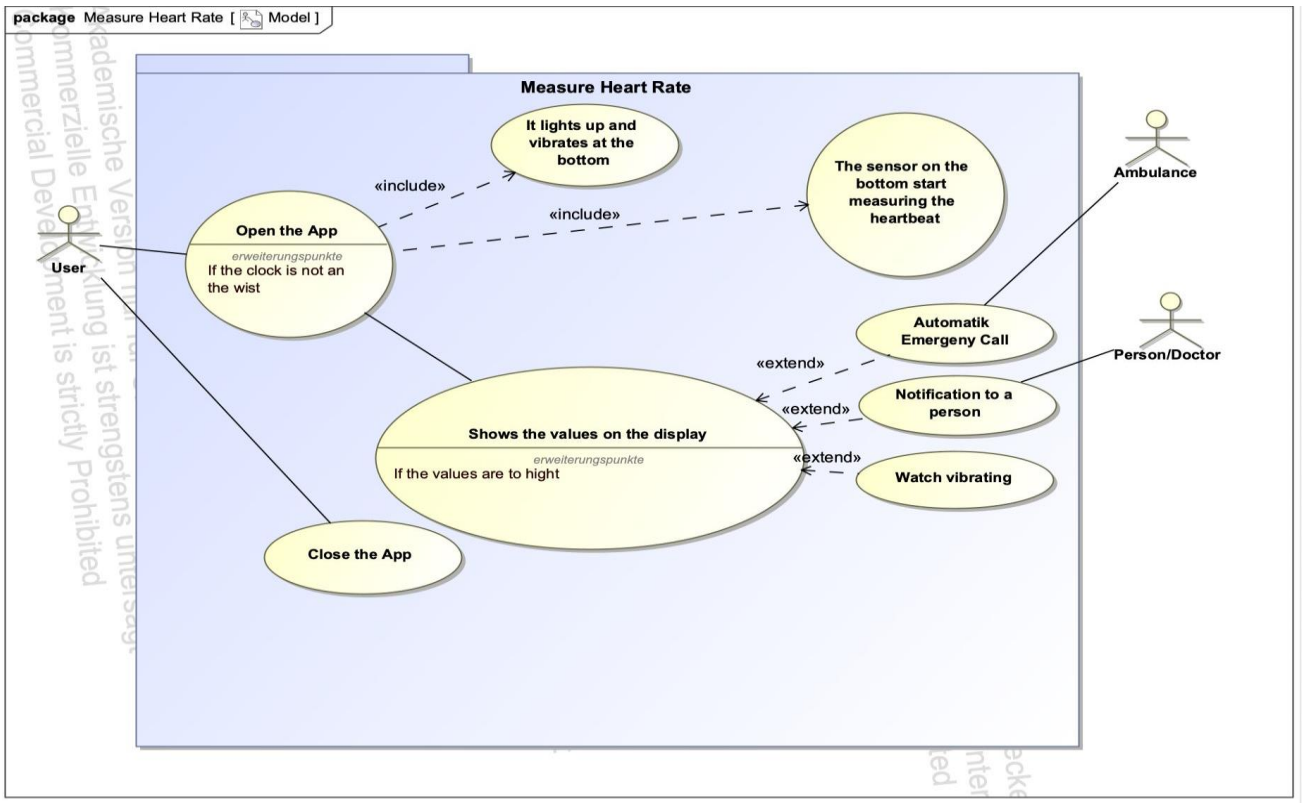
3.1.2 Use Case

This is about a smartwatch always checking your heart. If it sees your heart is acting strange or you might have passed out, it sends a message to someone you trust. The message tells them what's happening and where you are. In really serious situations, if the trusted person doesn't respond, the smartwatch can even call for emergency help, telling them where you are. It's all about making sure you get help fast when you need it.

Name	Measure Heart Rate
ID	01
Description	This use case involves a smartwatch continuously monitoring the user's heart rate and, upon detecting signs of irregularities, automatically sending a message to a trusted contact or making an emergency call.
Trigger	The smartwatch detects an abnormal heart rate pattern or signs of powerless.
Actors	Smartwatch & User & Trust Contact or Emergency Services
Pre-conditions	The smartwatch is on, and it can check the user's heart rate. The user has configured emergency contacts or granted permission for emergency notifications
Post-conditions	A message is sent to a trusted contact. In case of severe emergency, emergency services are automatically contacted.
BasicFlow **Description** **Actions** **1** **2** **3** **4**	The smartwatch notices if there's something odd in the heart rate. **Detection of Irregularities:** The smartwatch identifies abnormal heart rate patterns or signs of unconsciousness. **Emergency Notification:** If irregularities are detected, the smartwatch automatically sends a message to a trusted contact, providing details of the situation. **User Confirmation:** The trusted contact receives the message. **Emergency Call (Optional):** If the situation is critical and no response is received from the trusted contact, the smartwatch automatically initiates an emergency call to designated emergency services.
AlternativeFlow **Description** **Actions** **1** **2** **3**	The smartwatch notices if there's something odd in the heart rate. **Detection of Irregularities:** The smartwatch identifies abnormal heart rate patterns or signs of unconsciousness. **Emergency Notification:** If irregularities are detected, the smartwatch automatically sends a message to a trusted contact, providing details of the situation. **User Confirmation:** The trusted contact receives the message. The trusted contact is available and decides what to do next.

3.1.3 Use Case Diagram UI

The use case diagram outlines how users interact with the interface, considering different situations. If the smartwatch is not worn on the wrist, it initiates vibration and illumination to alert the user. Users have the flexibility to open the app, access its features, and quickly close it without prolonged use. Upon placing the smartwatch on the wrist, it promptly activates, enabling immediate utilization of its various functionalities. This diagram offers a comprehensive understanding of user interactions within the smartwatch interface, covering a range of usage scenarios.



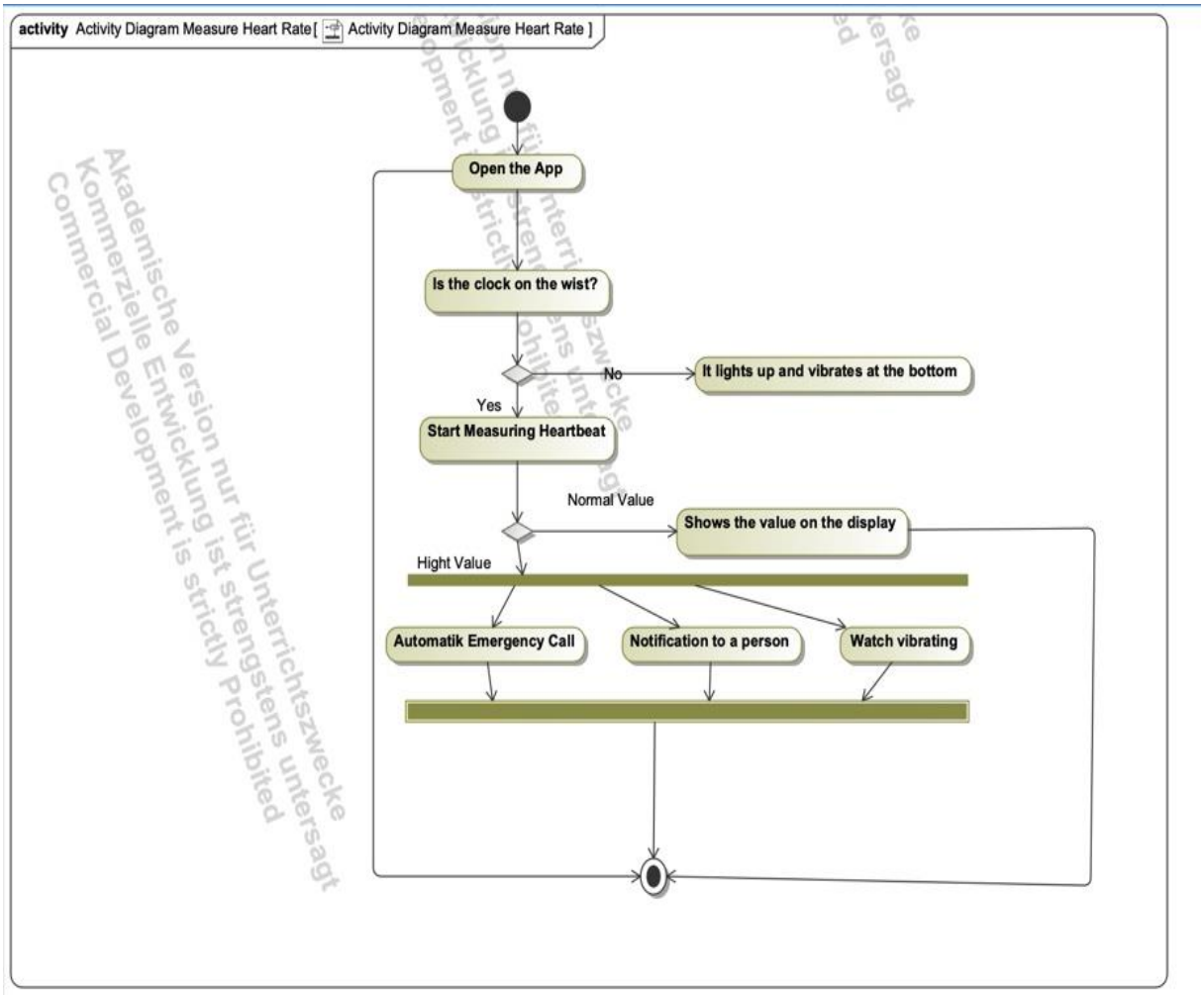
If the measured values reach a critical point indicating that the user is in danger, the smartwatch could automatically perform a check and notification. In such a scenario:

Automatic Check and Vibration:
The smartwatch continuously monitors measured values such as heart rate. If these values surpass a predefined critical range, the smartwatch automatically initiates vibration to alert the user.

Emergency Call or Message:
In addition, as an alternative option, if the situation is deemed critical, the smartwatch could automatically initiate an emergency call or send a predefined message to a specified contact, ensuring a timely response to the potential life-threatening situation.

3.1.4 Activity Diagram UI

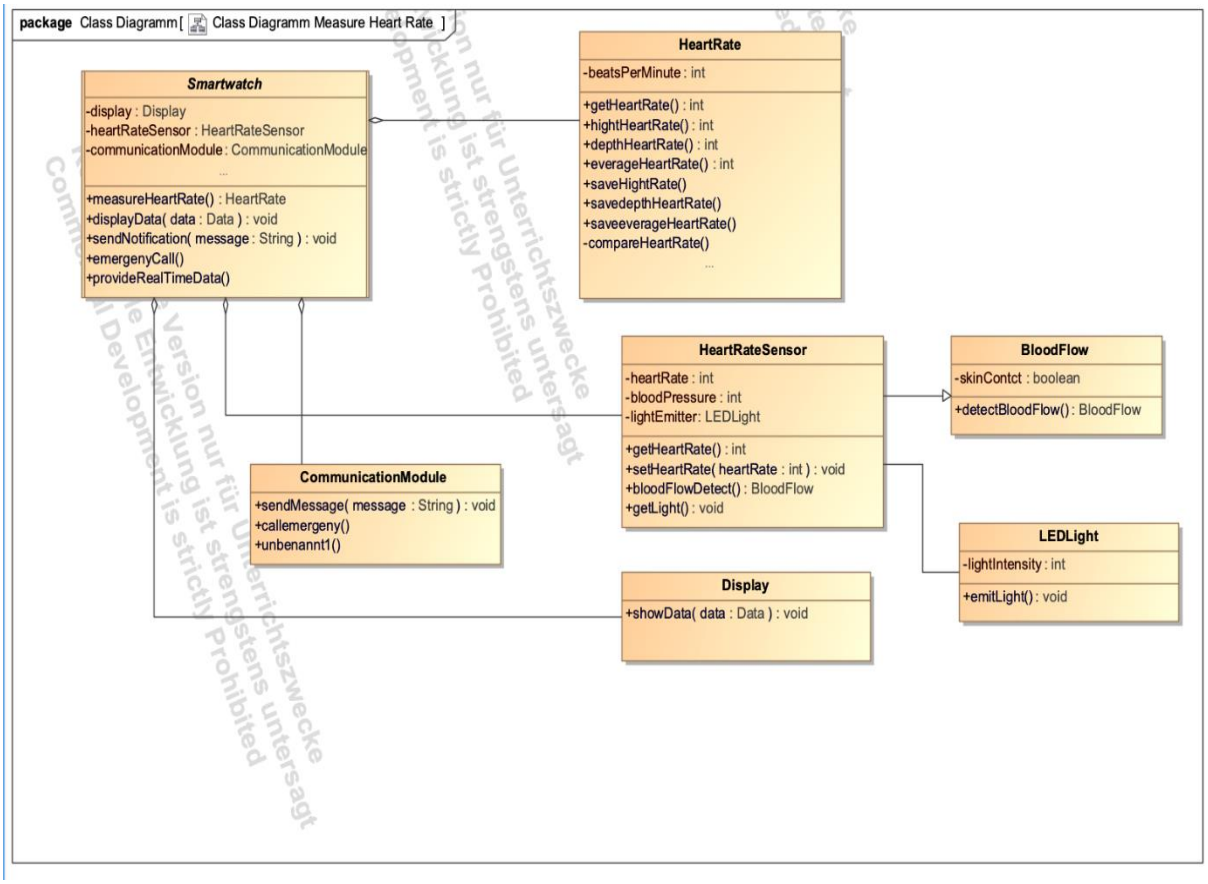
The primary task within the user interface revolves around navigation, achieved through repetitive clicking and swiping actions.
The activity diagram portrays this interaction as a continuous loop, where users consistently swipe and click until the smartwatch begins measuring the heartbeat. If this process is automated, and the smartwatch continually monitors the heartbeat, it immediately triggers a call or sends an emergency SMS if it detects a high heartbeat, assuming that the user is on unconscious or can't react and touch the watch.



3.1.5 Class Diagram UI

In the class diagram, the classes are connected through associations. Here are the connections between the classes:

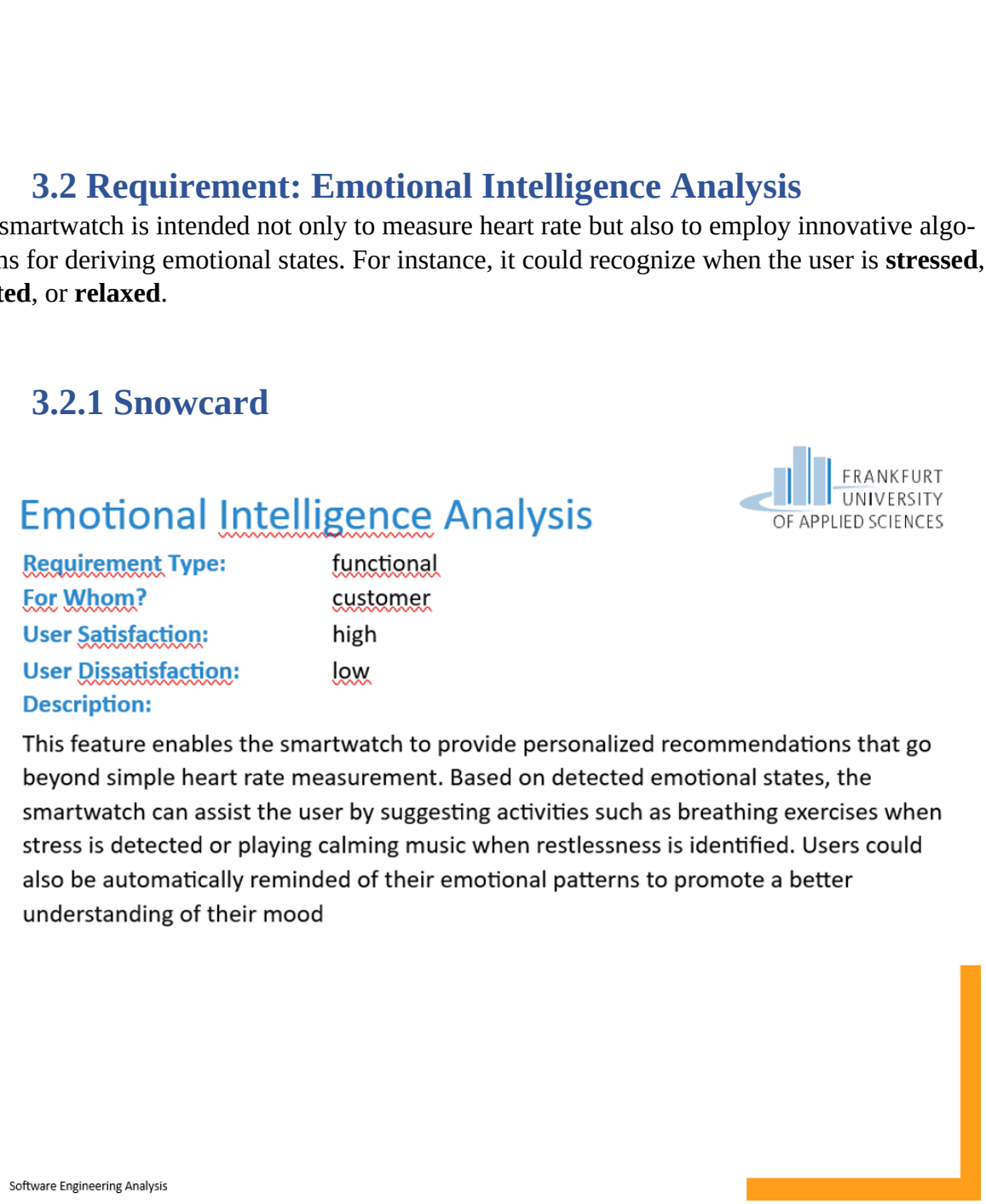
- The class Smartwatch has an association with the HeartRate. This means that the Smartwatch supports heart rate measurement and can access the methods of the HeartRate.
- The class Smartwatch has an association with the class HeartRateSensor. This indicates that a Smartwatch contains a heart rate sensor and can access the methods of this sensor.
- The class Smartwatch has an association with the class Display. This implies that a Smartwatch has a display and can access the methods of this display.
- The class Smartwatch has an association with the class CommunicationModule. This signifies that a Smartwatch has a communication module and can access the methods of this module.
- The Heart Rte Sensor has a associatin with the Blow Flow and with the LEDLight



3.2 Requirement: Emotional Intelligence Analysis

The smartwatch is intended not only to measure heart rate but also to employ innovative algorithms for deriving emotional states. For instance, it could recognize when the user is **stressed**, **excited**, or **relaxed**.

3.2.1 Snowcard



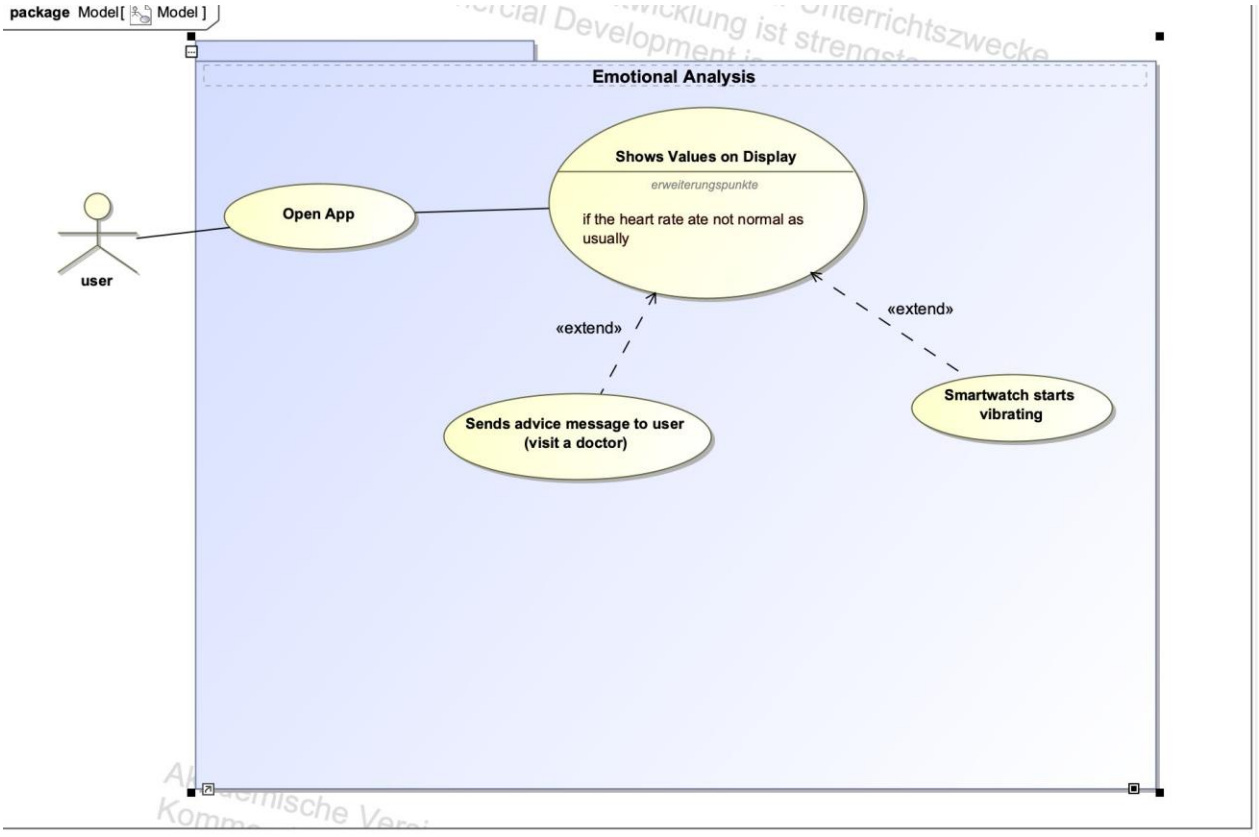
3.2.2 Use Case

The "Emotional Analysis" use case involves the smartwatch monitoring the user's heart rate, identifying prolonged periods of high heart rate as potential stress, and then suggesting personalized activities, such as deep breathing exercises, to help the user manage stress. The user receives notifications with recommendations, and the smartwatch continues to monitor their emotional state during and after the suggested activities, saving this information for future analysis. Users have the option to decline recommendations, and the smartwatch persists in tracking emotional changes to offer additional support.

Name	Emotional Intelligence Analysis
ID	02
Description	This is about the smartwatch figuring out how the user feels by looking at their heart rate and giving them advice.
Trigger	The smartwatch sees the user's heart rate is high for a while.
Actors	Smartwatch & User
Pre-conditions	The user said it's okay for the smartwatch to check their emotions
Post-conditions	The smartwatch sends a message to the user. It saves the information about how the user felt.
BasicFlow	
Description	Noticing if the smartwatch sees the heart rate is high for a long time, it understands the user might be stressed.
Actions	
1	Guessing Emotion: Using its clever programs, the smartwatch guesses how the user feels, like being stressed.
2	Sending a Message: The smartwatch sends a message to the user, suggesting something to help, like deep breathing.
3	User Gets Message: The user sees the message and decides what to do.
AlternativeFlow	
Description	Noticing if the smartwatch sees the heart rate is high for a long time, it understands the user might be stressed.
Actions	
1	Guessing Emotion: Using its clever programs, the smartwatch guesses how the user feels, like being stressed.
2	Sending a Message: The smartwatch sends a message to the user, suggesting something to help, like deep breathing.
3	User Gets Message: The user decides not to do what the smartwatch suggests. That's okay, and the smartwatch keeps watching.
AlternativeFlow	Noticing if the smartwatch sees the heart rate is high for a long time, it understands the user might be stressed.
Description	
Actions	Guessing Emotion: Using its clever programs, the smartwatch guesses how the user feels, like being stressed.
1	Sending a Message: The smartwatch sends a message to the user, suggesting something to help, like deep breathing.
2	User Gets Message:Even after giving advice, the smartwatch and the user doesn't do nothing ,the watches
3	still looks how the user feels. If it sees a change, it might suggest something else.

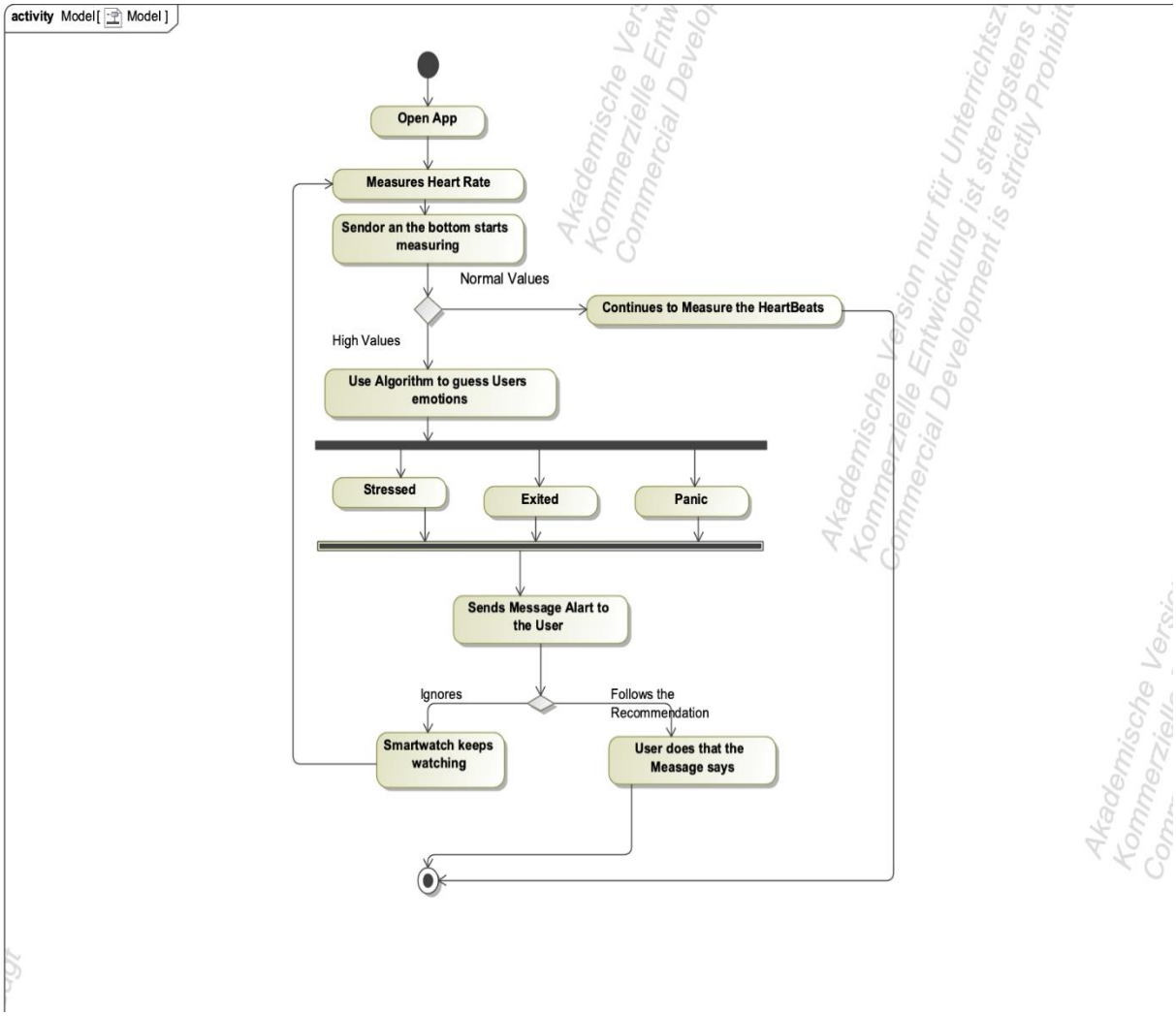
3.2.3 Use Case Diagram UI

In this use case, users utilize the smartwatch to discover their emotional state, which is continuously monitored by measuring heartbeats. The smartwatch employs an algorithm in the background to analyze emotional patterns based on the heart rate data. Users receive notifications about their emotional analysis. Additionally, the smartwatch can independently detect emotional states using the implemented algorithm, providing users with insights into their emotions throughout the day.



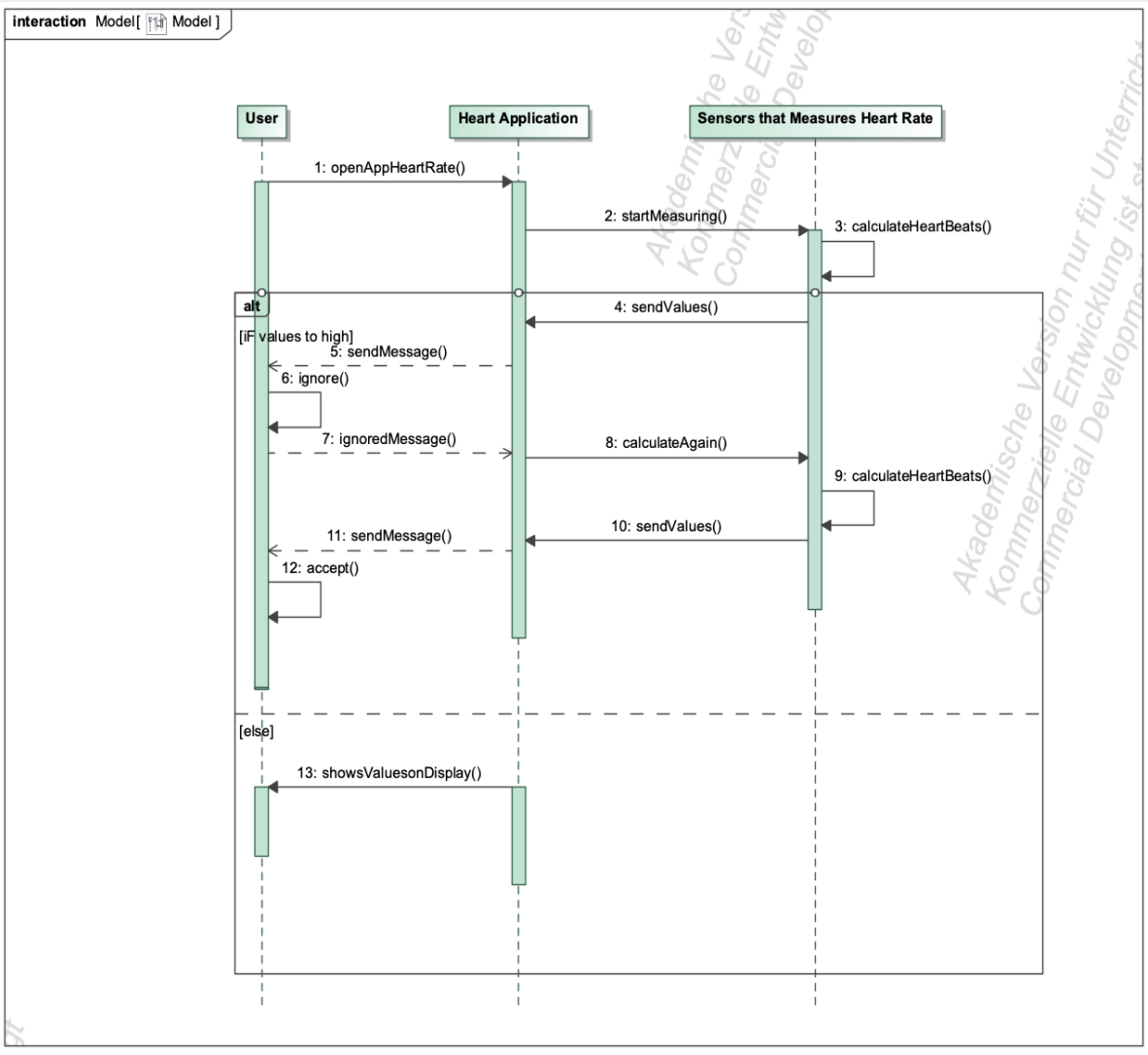
3.2.4 Activity Diagram UI

In this activity diagram, the smartwatch, relying on an algorithm to infer users' emotions such as excitement, stress, or panic, decides when to send information to the user. The user has the option to either ignore the message, considering it might be a false alarm, while the smartwatch continues monitoring the heart rate. Alternatively, the user can accept the message and follow the recommended actions for a more personalized experience in managing their emotional well-being.



3.2.5 Sequence Diagram UI

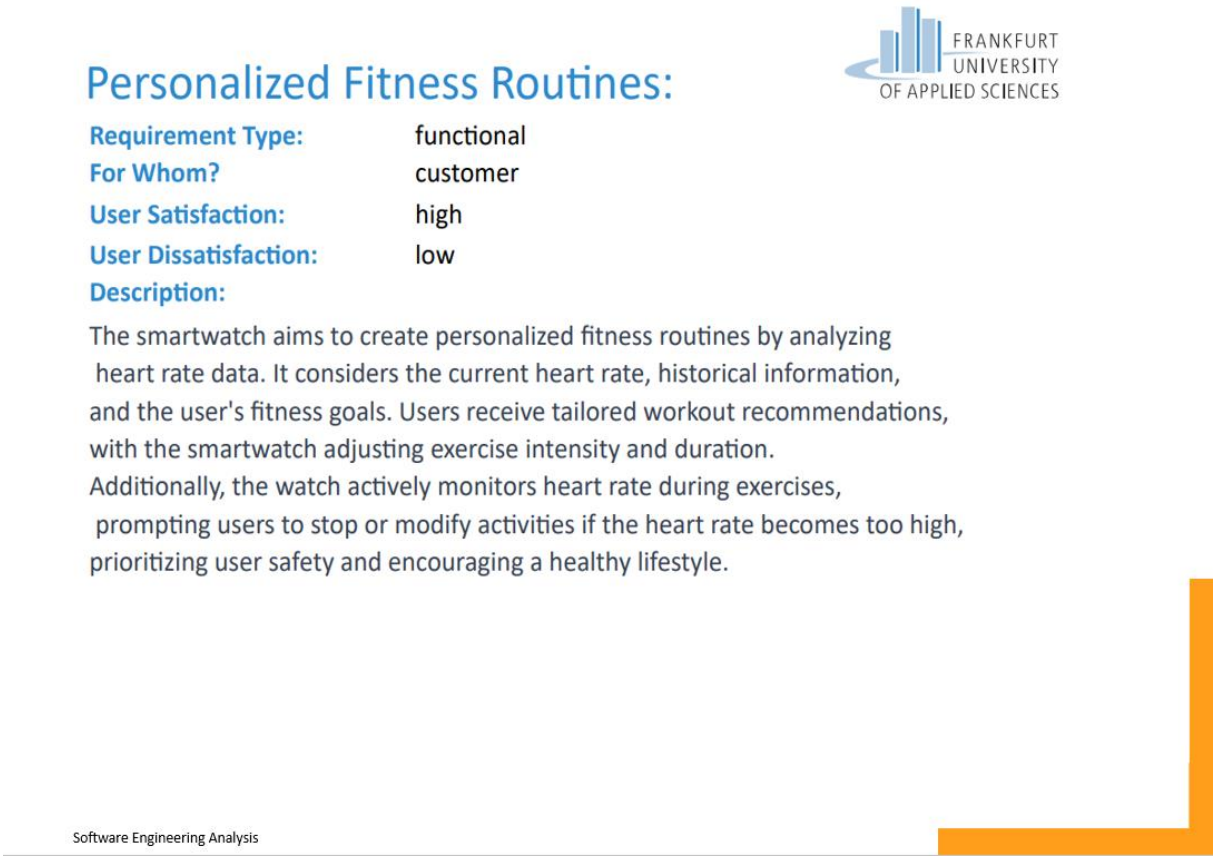
This diagram illustrates how the function operates, involving the user, the smartwatch and in the background, sensors working in conjunction with the algorithm to determine the user's emotional state. It outlines various possibilities regarding the messages sent when the heart rate is elevated, taking into account user interaction. The goal is to provide a comprehensive view of how the system functions in assessing and responding to the user's emotional well-being.



3.3 Requirement: Personalized Fitness Routines:

Purpose: The smartwatch aims to analyze heart rate data to create individualized fitness routines, considering not only the current heart rate but also historical data and personal goals. The smartwatch adjusts the intensity and duration of exercises and motivates the user to stay active.

3.3.1 Snowcard



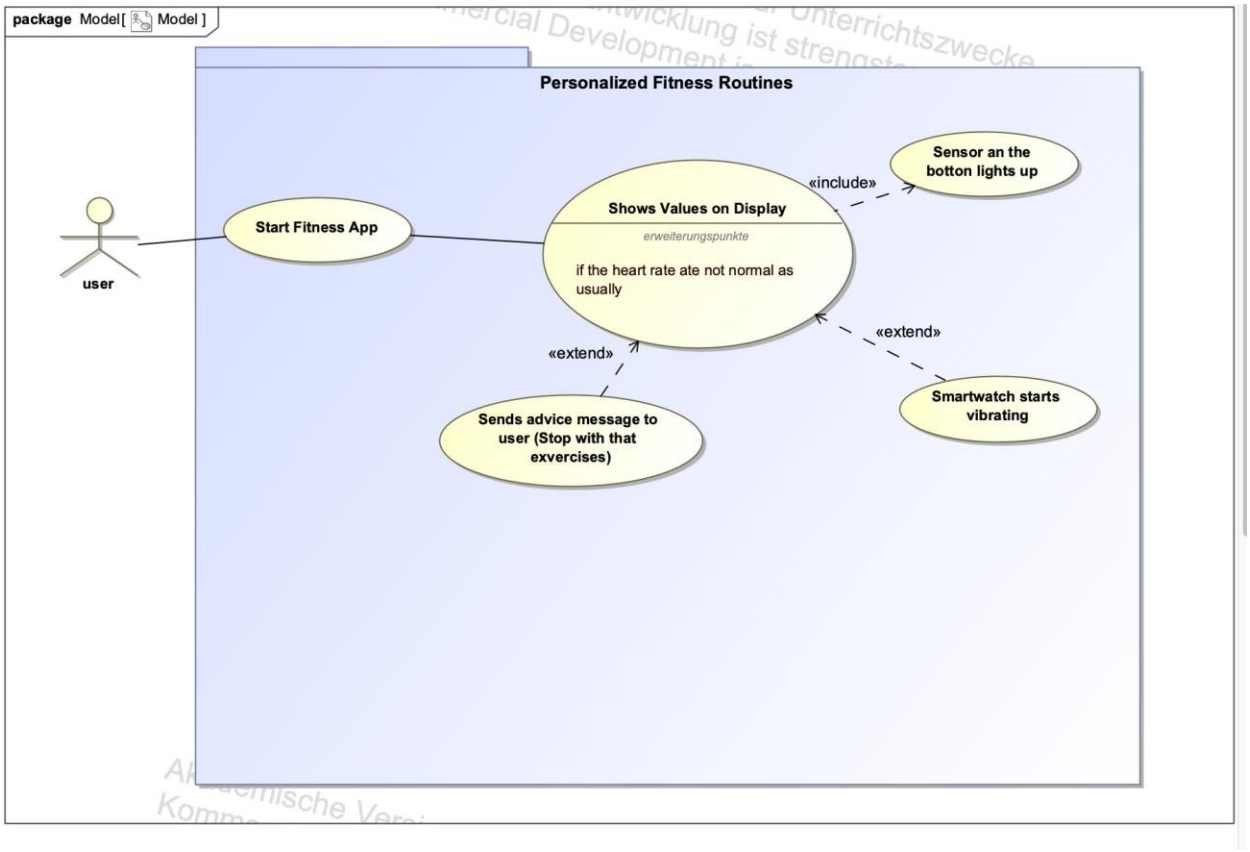
3.3.2 Use Case

The smartwatch makes special exercise plans for the user and helps them during workouts. It watches the heart rate and adjusts the exercise plan in real-time. If the heart rate is too high, it sends a message and vibrates to tell the user to stop or change what they're doing. After the exercise, the smartwatch sends a message about how well the user did and suggests improvements.

Name	Personalized Fitness Routines
ID	03
Description	This is about the smartwatch making special exercise plans for the user and helping them during workouts. By hight heart beat it will send messages, or vibrate
Trigger	The user starts to exercise with the smartwatch.
Actors	Smartwatch & User
Pre-conditions	The smartwatch is on, and it can watch the user's heart rate. The user told the smartwatch what exercise goals they have.
Post-conditions	The smartwatch keeps a record of what the user did during the workout. The user gets a message about how well they did
BasicFlow	
Description	The user begins exercising with the smartwatch. The smartwatch keeps an eye on the user's heart rate all the time.
Actions	
1	**Helping in Real Time:** While the user exercises, the smartwatch helps them by adjusting how hard and how long they should do each exercise based on their heart rate.
2	**Stop or Change if Heart Rate is Too High:** If the heart rate gets too high and might be dangerous, the smartwatch tells the user to stop or change what they are doing, sending a message and vibrating for attention.
3	**Message About Exercise:** The smartwatch sends a message to the user, saying how well they did and what they could do better.
AlternativeFlow	
Description	The user begins exercising with the smartwatch. The smartwatch keeps an eye on the user's heart rate all the time.
Actions	
1	**Helping in Real Time:** While the user exercises, the smartwatch helps them by adjusting how hard and how long they
2	should do each exercise based on their heart rate.
3	**Stop or Change if Heart Rate is Too High:** If the heart rate gets too high and might be dangerous, the smartwatch tells the user to stop or change what they are doing, sending a message and vibrating for attention. **Message:** The user ends the exercise.

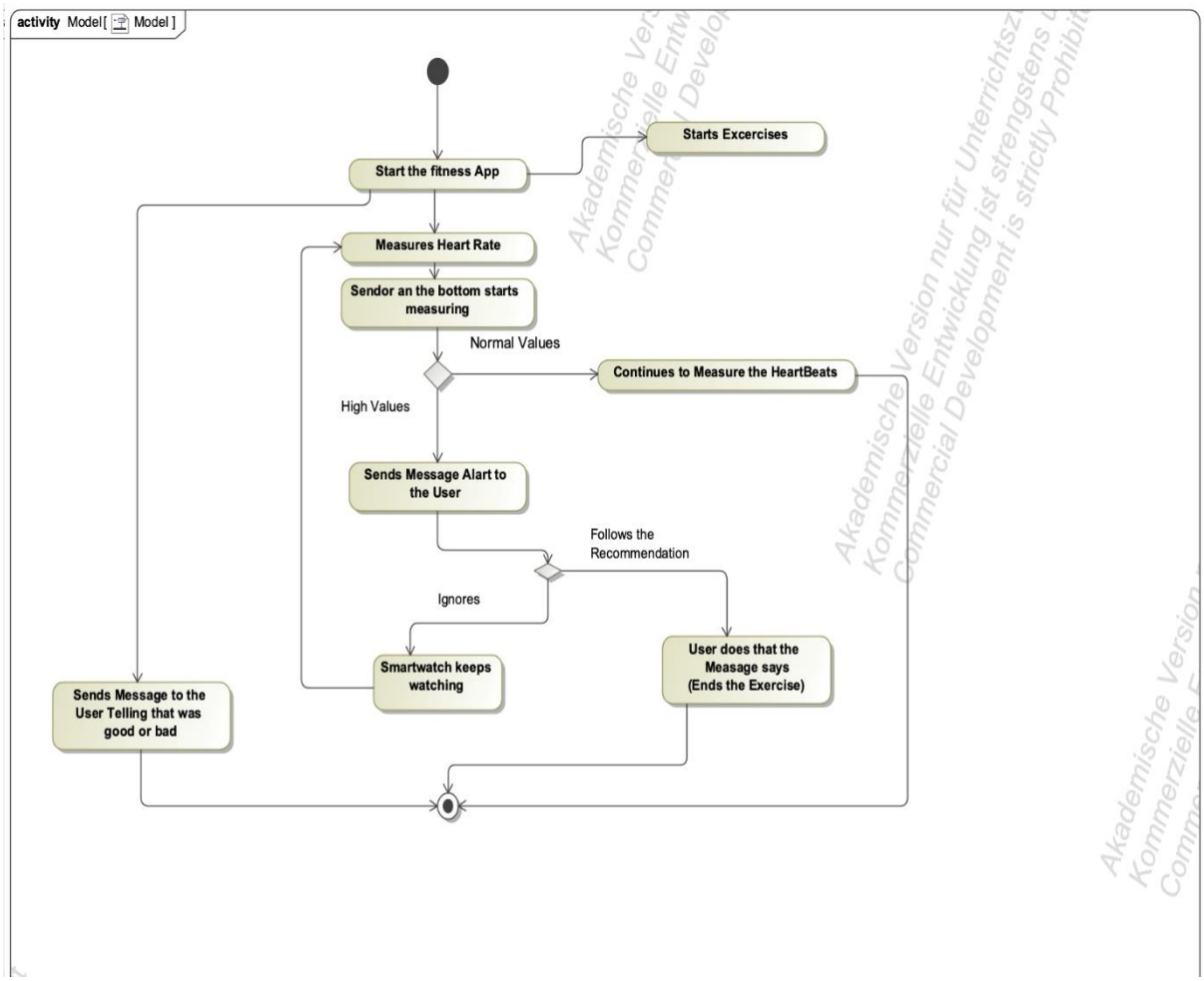
3.3.3 Use Case Diagram UI

In this use case, the user wears the watch during a fitness session. They open the fitness app and are continuously notified if there are any irregularities with their heartbeats.



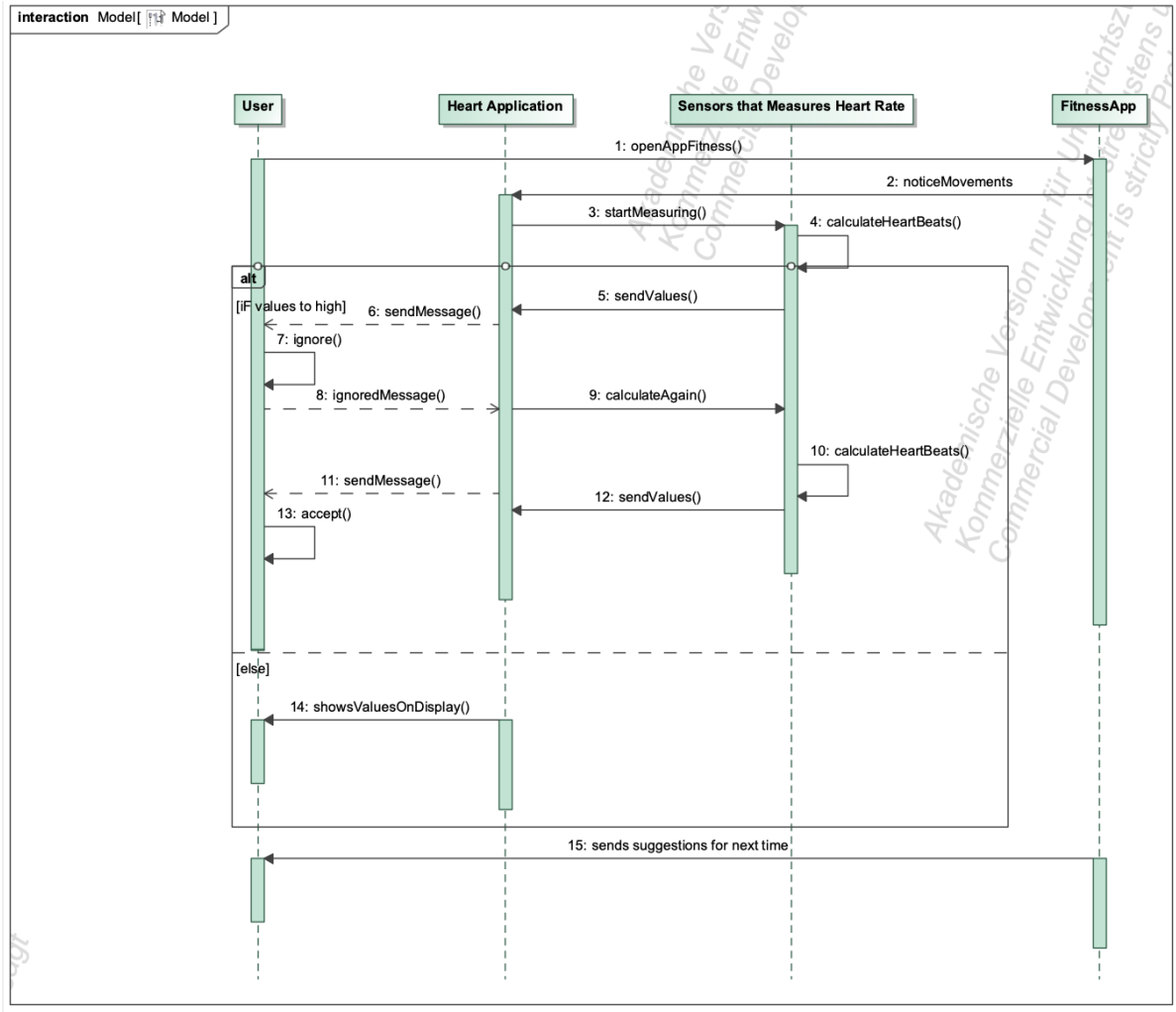
3.3.4 Activity Diagram UI

In this diagram, the flow is illustrated once again. When the user starts exercising and their heart rate increases, the app will send an alert message. The user can then decide what action to take. If they ignore the message, the app will continue to monitor the heartbeat to potentially notice anything noteworthy.



3.3.5 Sequence Diagram UI

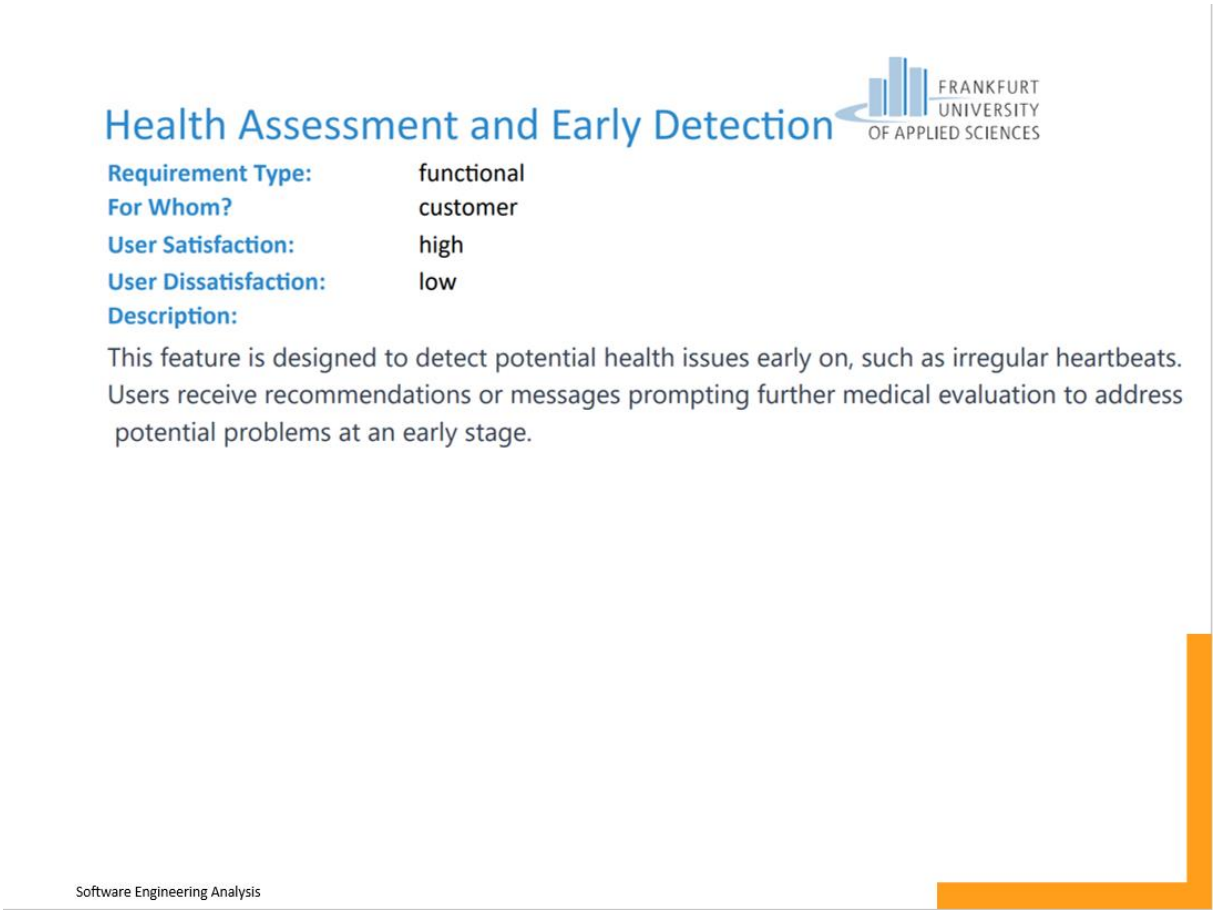
Sequence diagram involves user, heart app, fitness app, and the sensors of the watch, they will communicate together to determine the heartbeats, notify the user, and this happens repeatedly in a loop with conditional statements and measuring the heartbeats. The user will be notified, and it's up to the user to decide which direction to go with the watch.



3.4 Requirement: Health Assessment and Early Detection

The smartwatch utilizes heart rate data to generate health assessments. It identifies potential deviations from normal patterns and alerts the user in case of irregularities.

3.4.1 Snowcard



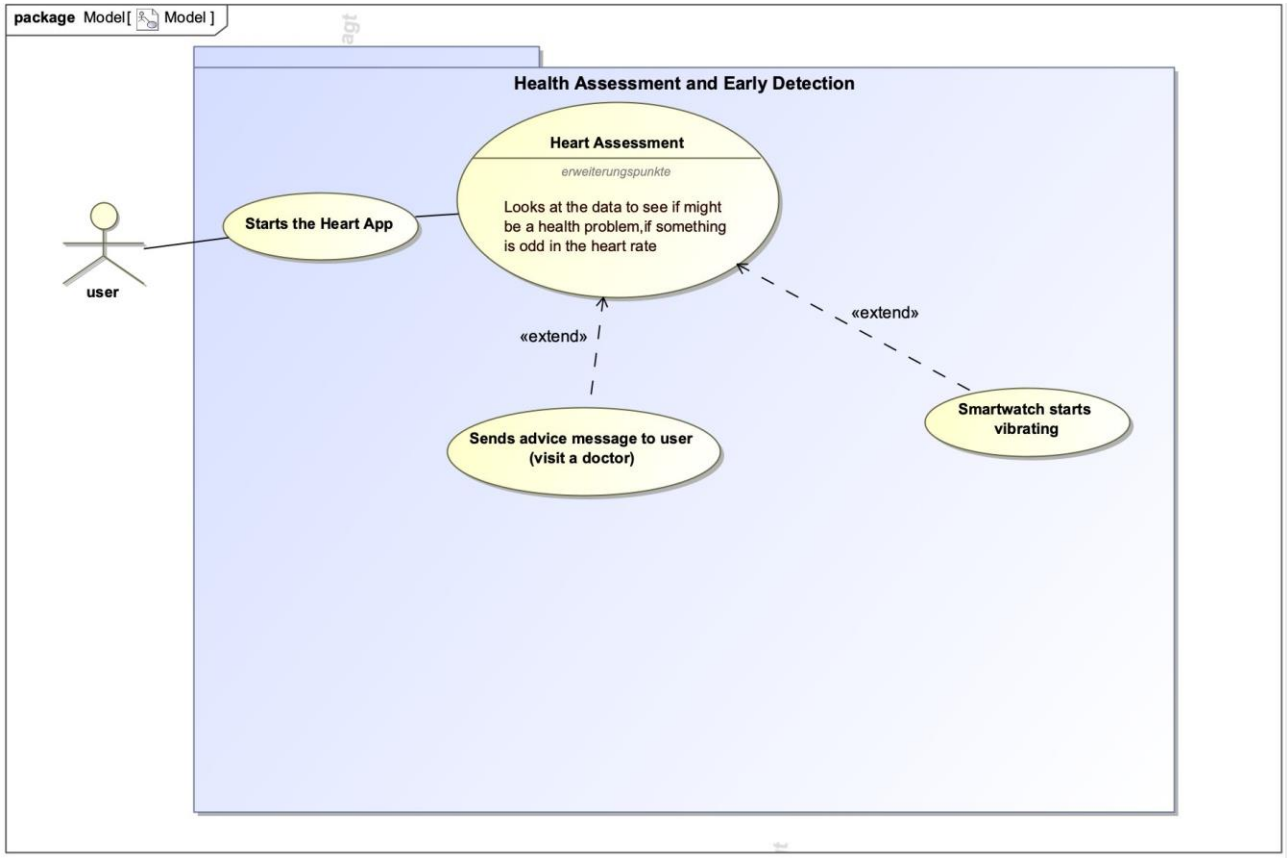
3.4.2 Use Case

This use case highlights the smartwatch's pivotal role in health monitoring by continuously assessing the user's heart rate. Leveraging heart rate data, the smartwatch detects potential deviations from normal patterns and promptly alerts the user to irregularities. In the event of identified concerns, the smartwatch sends a notification to the user, advising them to seek further medical evaluation. The overarching aim is to facilitate early detection of potential health issues, providing timely recommendations for professional medical consultation. Through continuous monitoring and advanced algorithms, the smartwatch becomes an indispensable tool for assessing and safeguarding the user's health, ensuring proactive and personalized care.

Name	Health Assessment and Early Detection
ID	04
Description	This use case is about the smartwatch keeping an eye on the user's heart rate all the time. If it sees anything unusual, it sends a message to the user and suggests they see a doctor to check for possible health problems.
Trigger	The smartwatch notices something strange in the user's heart rate.
Actors	Smartwatch & User
Pre-conditions	The smartwatch is on, and it can check the user's heart rate. The user said it's okay for the smartwatch to do health assessments and send messages.
Post-conditions	The smartwatch sends a message to the user about possible health concerns. The user gets suggestions to see a doctor for more advice.
BasicFlow	
Description	The smartwatch notices if there's something odd in the heart rate.
Actions	
1	**Health Assessment:** The smartwatch looks at the data to see if there might be a health problem.
2	**Sending a Message**: If there's something wrong, the smartwatch sends a message to the user, telling them there might be a health concern and suggesting they see a doctor.
3	The user gets the message and agrees to think about seeing a doctor.
AlternativeFlow	
Description	The smartwatch notices if there's something odd in the heart rate.
Actions	
1	**Health Assessment:** The smartwatch looks at the data to see if there might be a health problem.
2	**Sending a Message**: If there's something wrong, the smartwatch sends a message to the user, telling them there might be a health concern and suggesting they see a doctor.
3	The user can say they don't want to think about it right now. The smartwatch keeps watching but might send reminders sometimes.

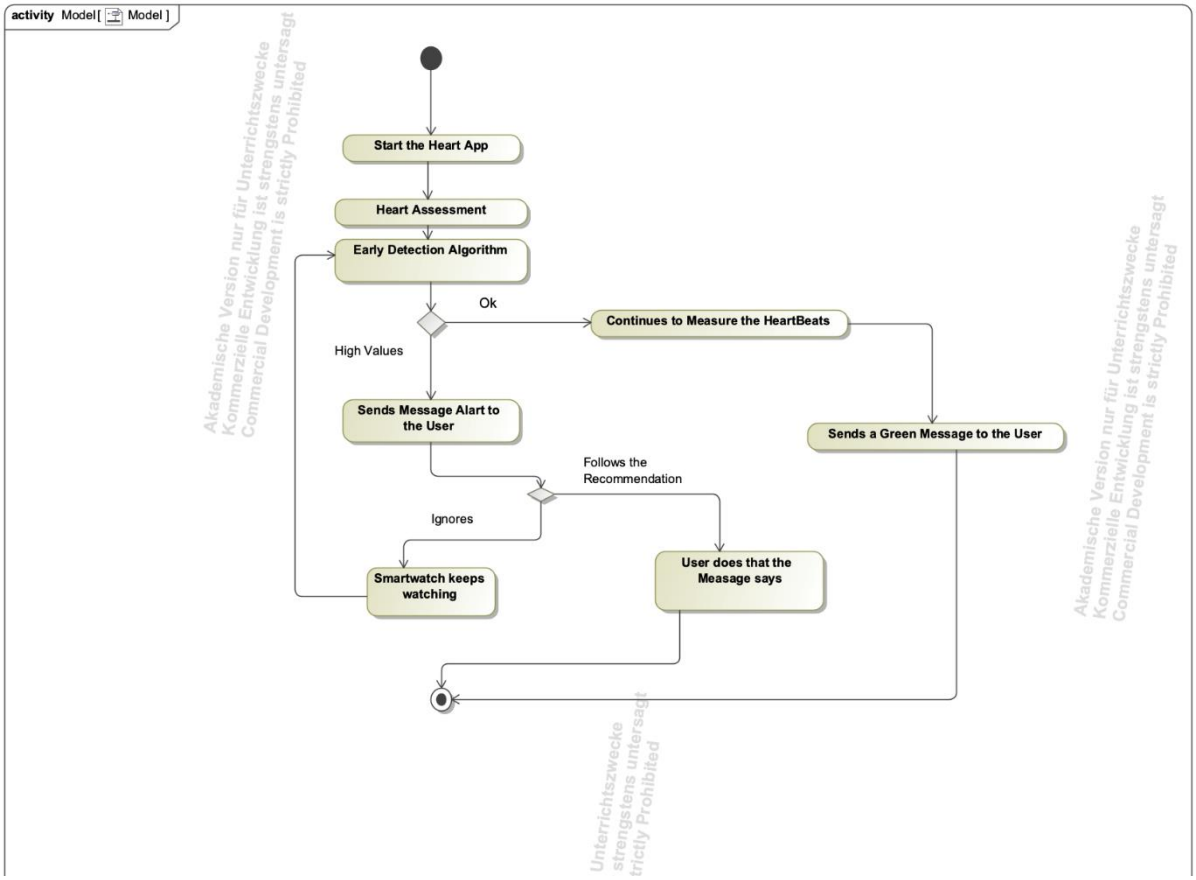
3.4.3 Use Case Diagram UI

In this use case diagram, it is shown that the user can also use the Heart app to determine in advance if their heart has any problems or if a visit to the doctor is necessary.



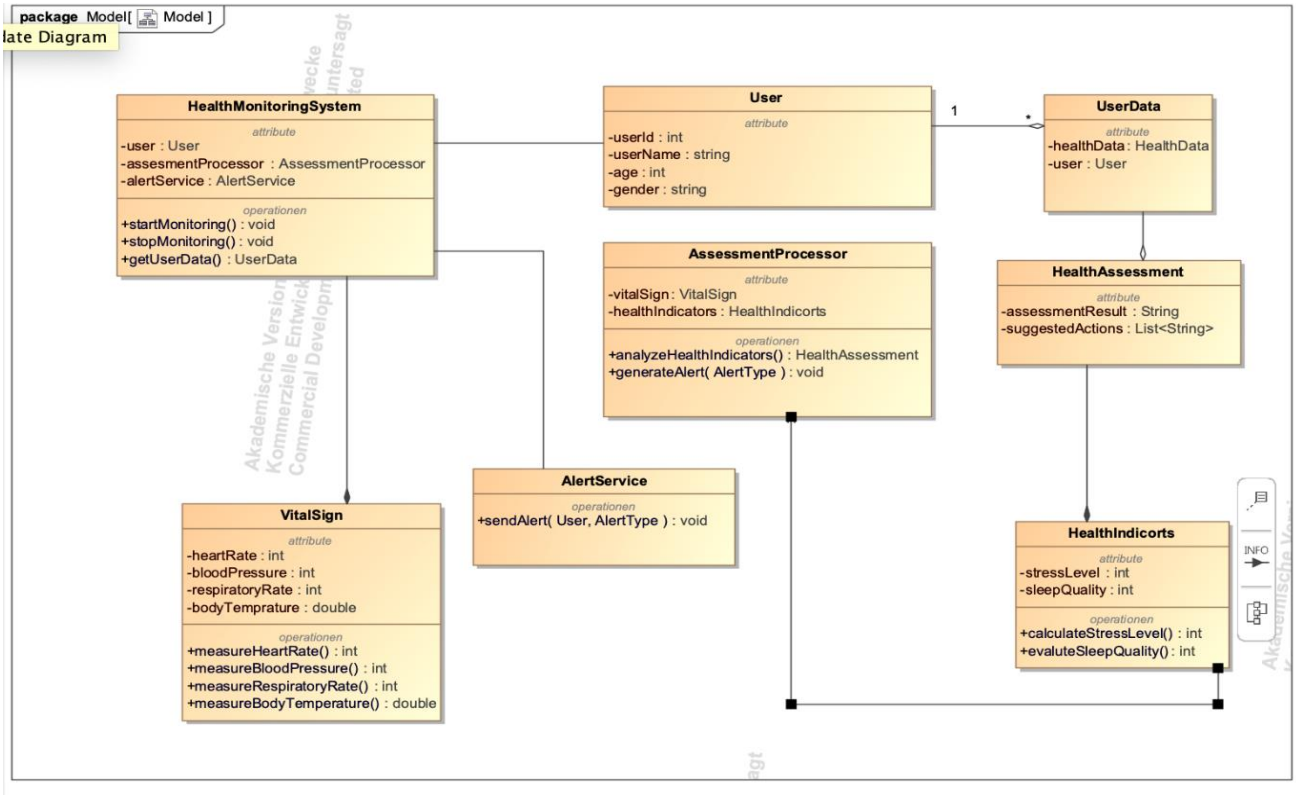
3.4.4 Activity Diagram UI

In this activity diagram, the flow of how everything works with the Early Detection Algorithm is shown. While the app measures the heartbeat, the algorithm continuously analyzes whether everything is okay with the user's heart. If everything is in the green zone, the user is always informed with a green message. If that's not the case, the user is notified to visit a doctor, and consequently, it's up to the user what to do. If the user declines, this algorithm will continue to be executed, and everything would run in a loop



3.4.5 Class Diagram UI

The class diagram depicts the structure of a health monitoring system. Key classes include HealthMonitoringSystem, User, UserData, AlertService, AssessmentProcessor, HealthAssessment, VitalSigns, and HealthIndicators. The system allows monitoring vital signs, analyzing health indicators, generating health assessments, and sending alerts. The User class represents the user with personal information, while VitalSigns and HealthIndicators measure physiological parameters. The AssessmentProcessor analyzes health indicators and creates health assessments. The AlertService sends alerts based on the assessment results. UserData maintains collected health data for the user.



3.5 UI Prototype for the App Heart Rate

Felix Dallmeyer

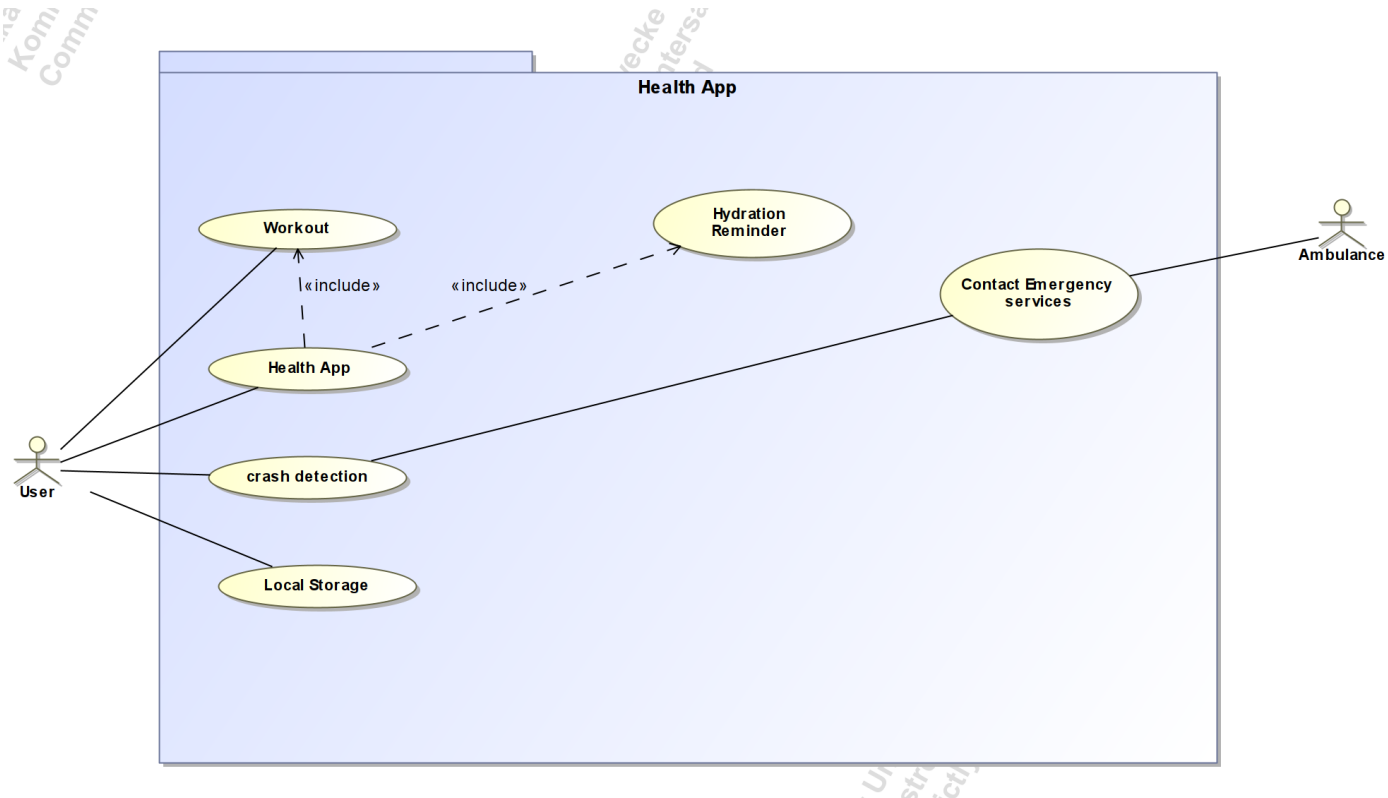
4. Health App Requirement

Goal of the health app is to enable the user to keep track of their health, inform them about irregularities and contact emergency services in case of an emergency. The application should save the data for a short period of time in a local database, so that the first responders can review the history of the cardiogram. The Idea is that the different parts of the App, like for instance the Heartrate, Workout recorder and crash detection are modular and could be used standalone within the app itself.

The Health app can be split up into 4 different requirements:

1) Healthapp
2) Workout(recorder)
3) Local Storage
4) Crash detection

4.1.1 Use case Diagram



4.1.2 Snowcards

Heartrate

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Requirement Type: functional
For Whom? customer
User Satisfaction: high
User Dissatisfaction: high

Description:
Application for the user to interact with. Displays the users Heartrate and other Health and Fitness related data.

Software Engineering Analysis

Local Storage

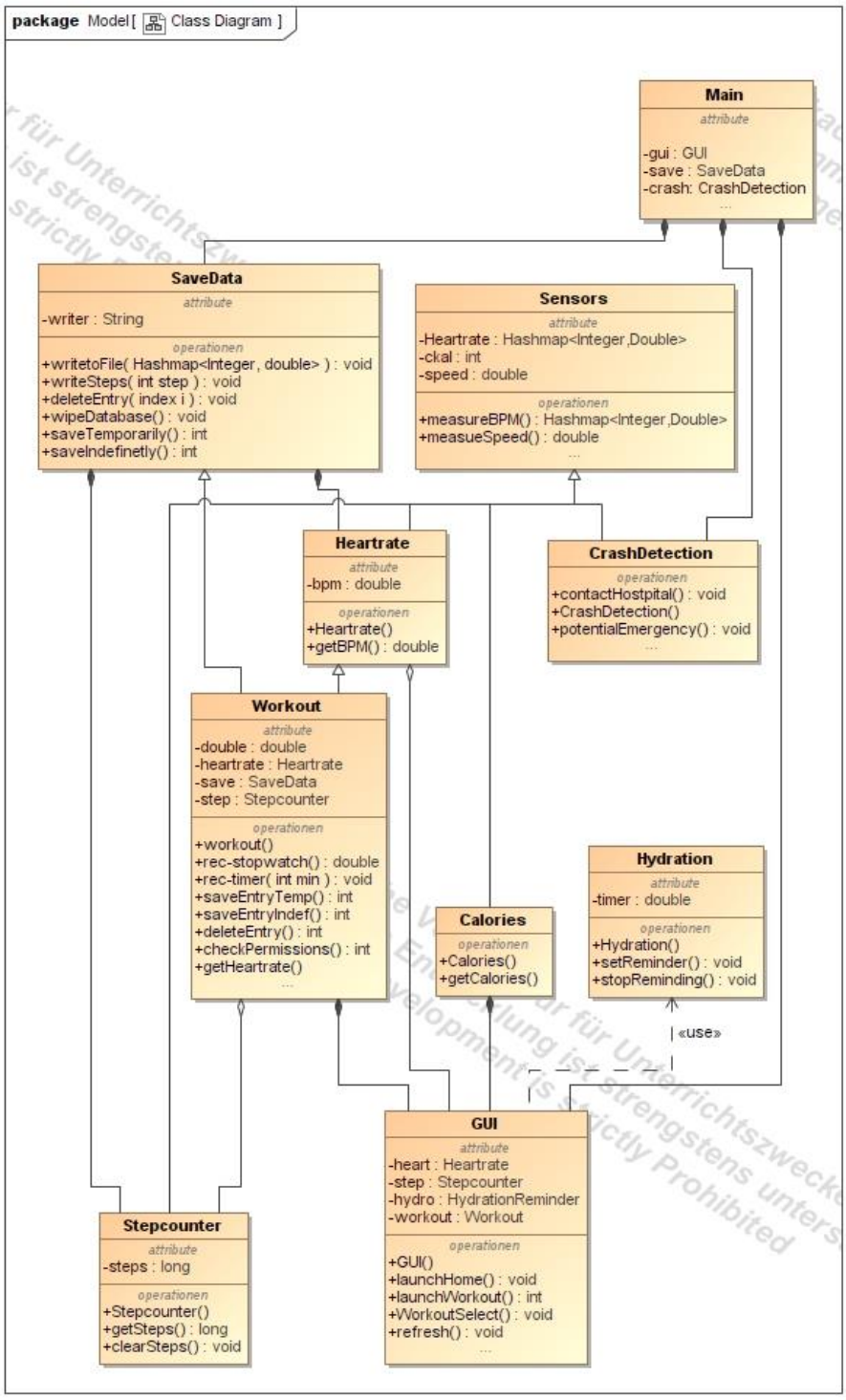
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Requirement Type: functional
For Whom? customer and maybe emergency services
User Satisfaction: low
User Dissatisfaction: low

Description:
Stores the Health apps recorded data either temporarily or indefinetely until the user decides to delete the saves.

Software Engineering Analysis

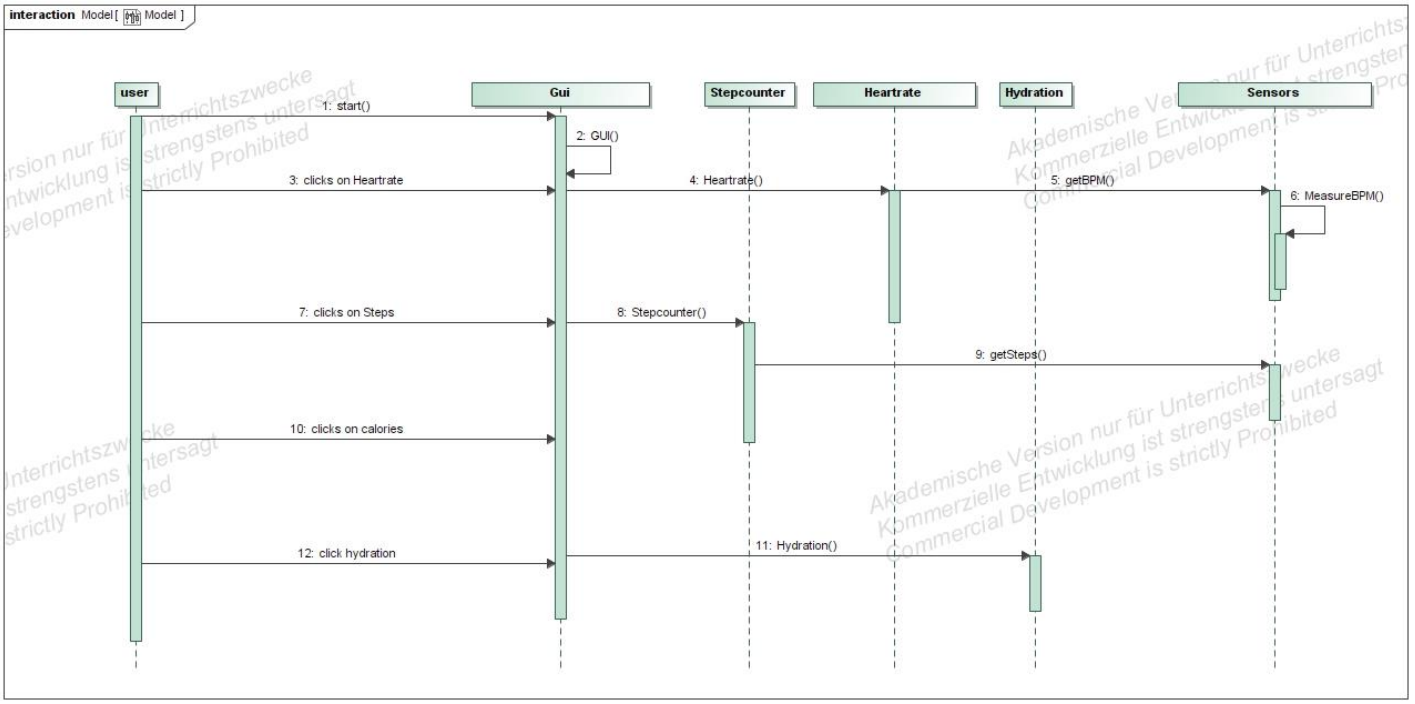
4.1.3 Class Diagram



The three most important classes in the Application are: Main, Sensors and GUI class, as they start all the classes and processes, or are needed for the other classes to work.

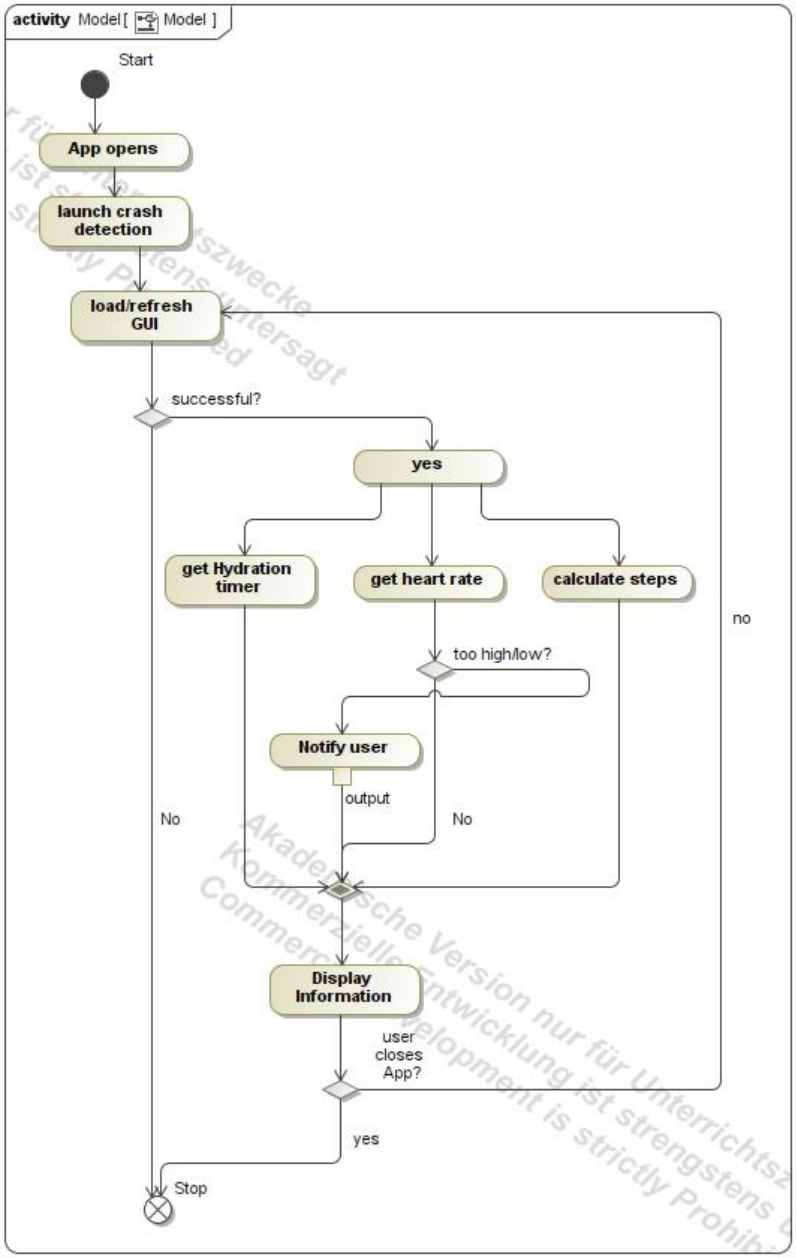
4.1.4 Sequence Diagram

This Sequence Diagram displays how the different classes of the health-app interact between each other according to user input.

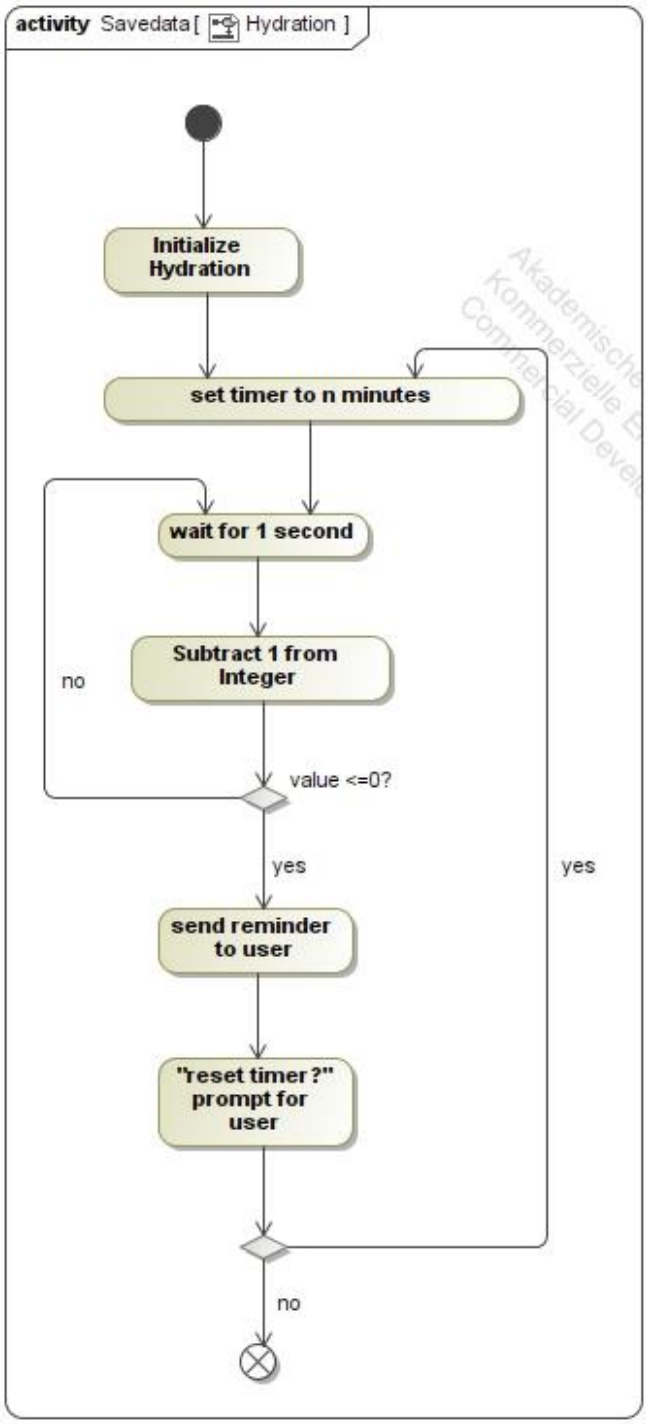


4.1.5 Activity Diagram for App loop at idle

Shown here is the basic process that gathers the Data form the user and passes it to the Graphical User Interface, which then displays it to the screen. The Process is being looped infinitely unless the user closes the Application or if the GUI fails to initialize/refresh. The Program starts with loading the components from the GUI as these are likely to be resource intensive. Then the GUI initializes the instances of the Classes: Heartrate, Stepcounter, Hydration and uses the getter functions to access the Sensordata. This part of the Program should be multithreaded for performance optimisation.

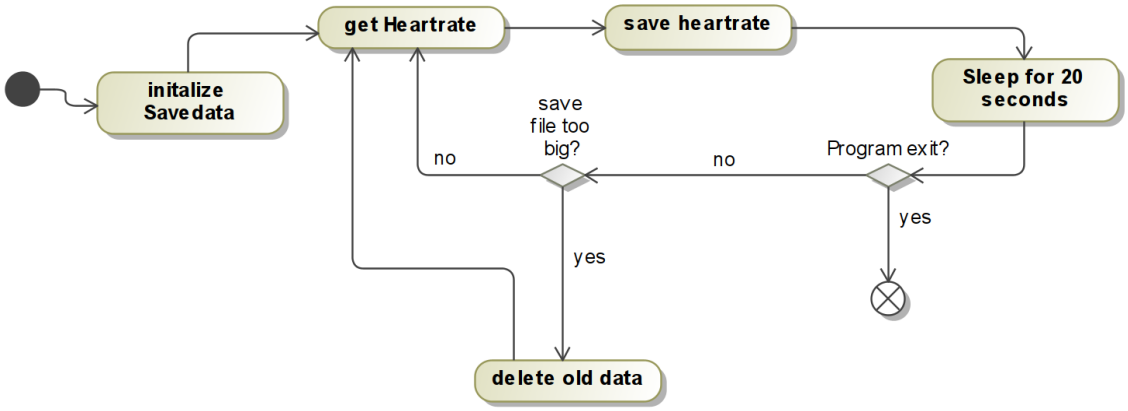


4.1.6 Activity Diagram – Hydration Reminder



The Hydration timer and crash detection could be running outside of the health App as background processes. Meaning they are not visible to the user unless the user is in an accident, or the hydration timer reaches zero.

4.1.7Activity Diagram - Savedata

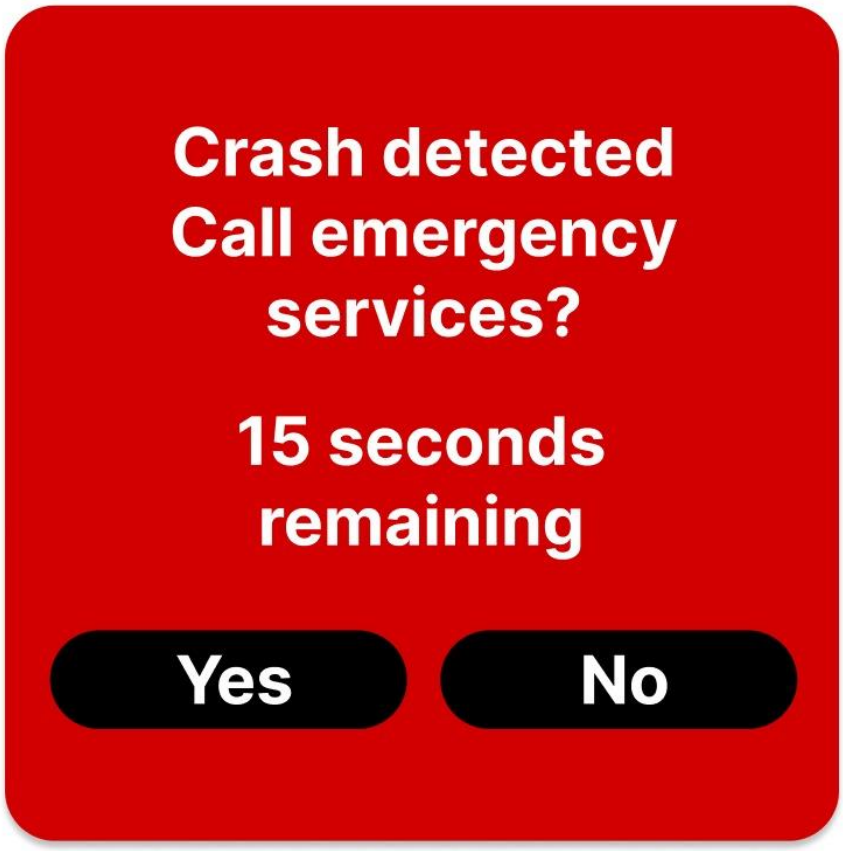


The diagram describes how the How the Heartrate gets measured and saved outside of the workout recorder. It recursively writes and deletes the save data on the smartwatch. This might be useful if a doctor needs to know any details about the users Heartrate to make a diagnosis. The data will be accessible to only the user, as it would otherwise be a privacy concern. This diagram also applies to the other data measured by the app. The workout module has a different way of saving the data, as shown later.

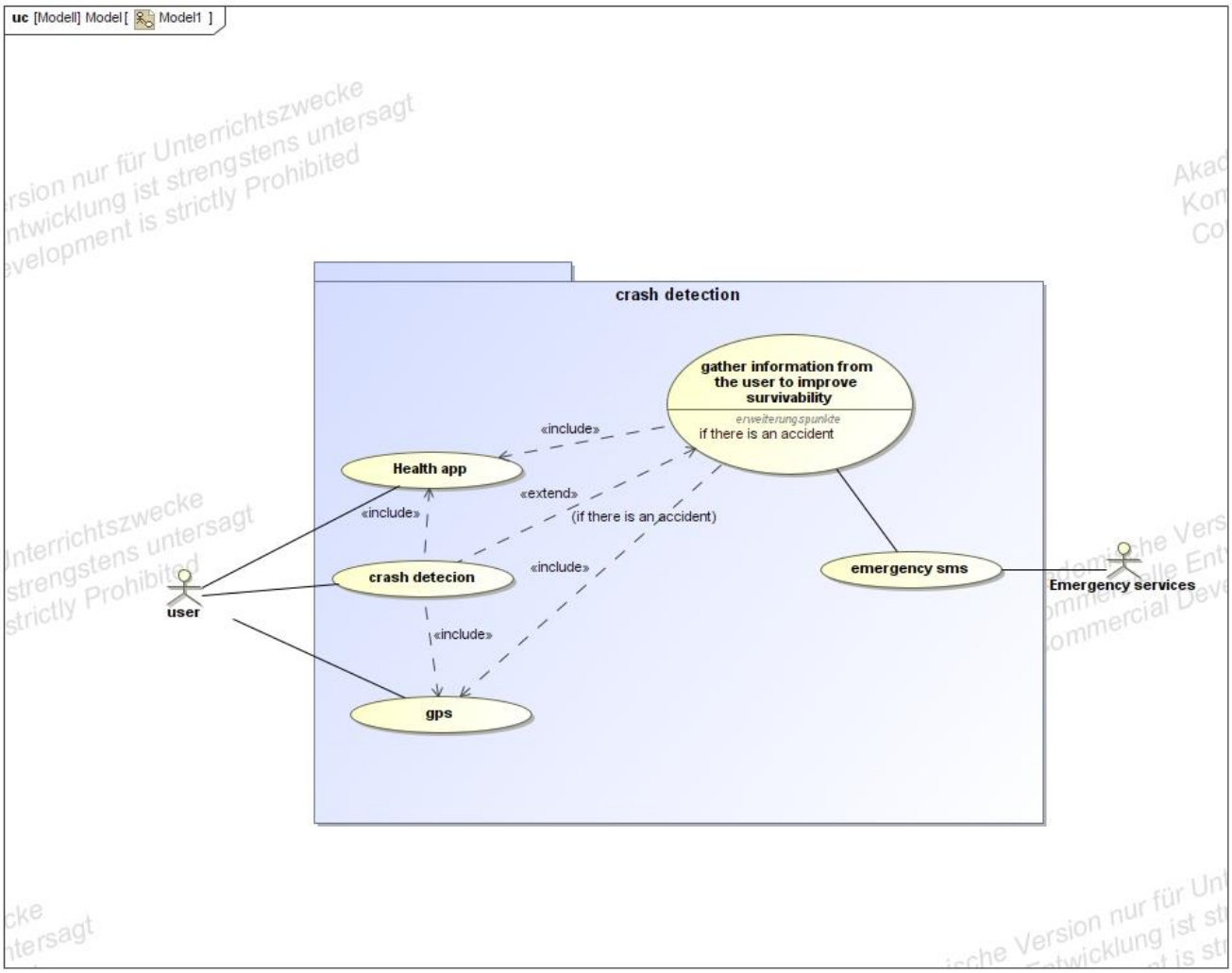
4.2 Crash detection

Crash detection is a feature implemented into the health app, that can, as the name implies, detect crashes and accidents, such as trips, falls and anything in that regard. Aside from just detecting said situations, it can send a sms containing the medical history of the past few minutes/hours as well as coordinates of the accident to the nearest Hospital, Ambulance, Doctors Office to reduce response times. Some people might not want to share their personal data with anyone, because they value their privacy, or see it as a security risk. That's why the crash detection and emergency-sms feature will be opt in and the owner of the watch can decide what data gets sent via the sms. The Heartrate measurement, Stepcounter and crash detection service run on startup of the watch, because it would be relatively useless, if the crash detection would only run if the health app is open.

4.2.1 UI mockup



4.2.2Use case diagram

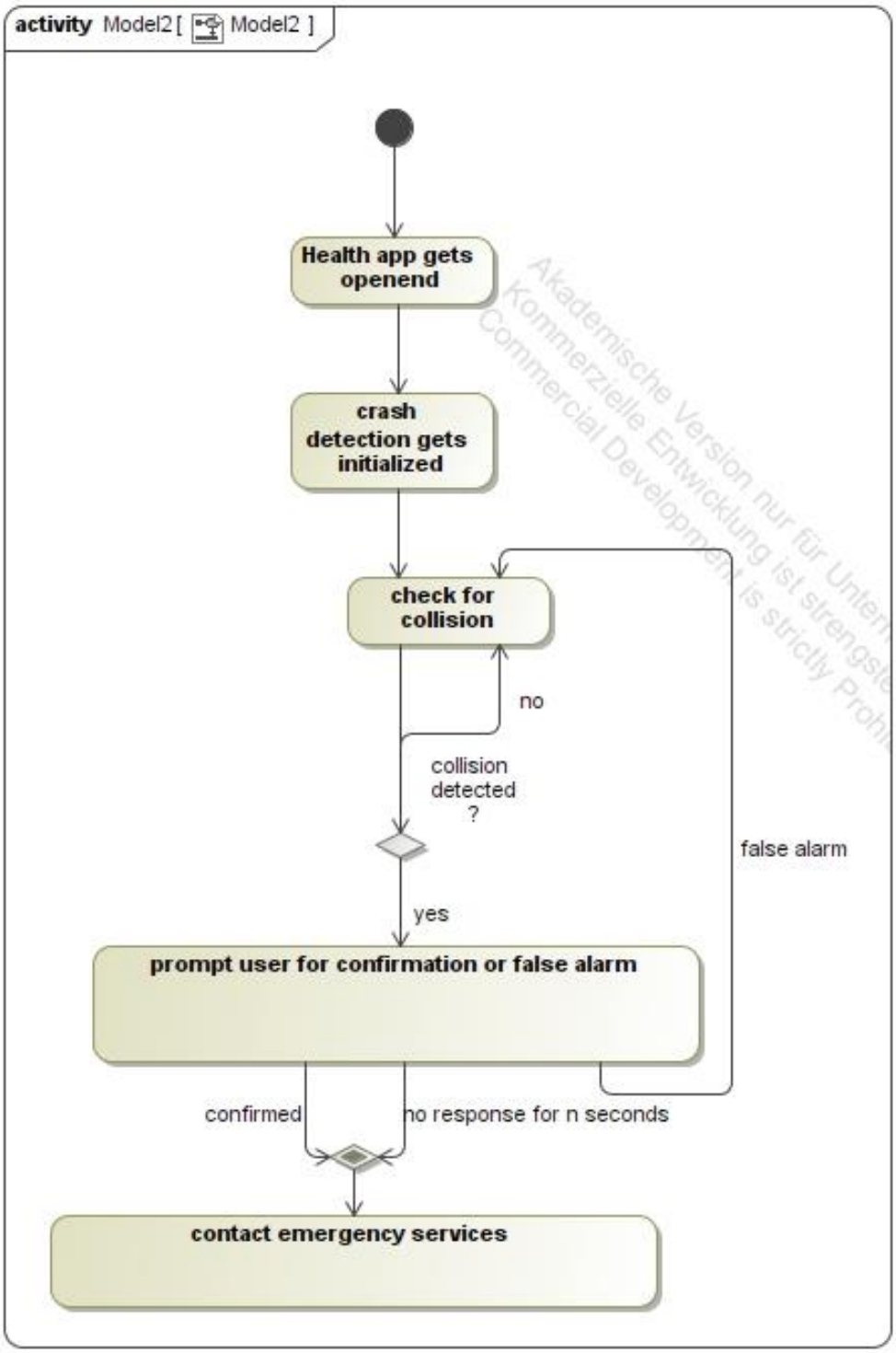


As the usecase diagram shows, is the intended functionality of the crash detection to detect an accident for example during exercise or a car crash and send an emergency sms to Emergency service containing the users location and his medical data of the last n minutes. This can help to improve response times and survivability rates in the future. The medical data comes from the health app's idle record function, which is always recording the heartrate and counted steps in the background, as it is very unlikely the user wants to actively have the app be open to record his steps or heartrate and just wants to see his history over the course of the past few hours. As some people might not like this feature, it will be opt in.

4.2.3 Use case Crash detection

Name	Crash detection
ID	2
Description	The user wants to be assured that he will be safe in case of an accident
Trig-ger	The user trips, falls or something else in that regard, and triggers the crash detection
Actors	User, emergency services
Pre-conditions	The watch is charged, turned on and is being worn by the user
Post-conditions	The user gets saved by emergency services or it was just a false alarm
Basic Flow **Descrip-tion**	In this scenario the crash recognition activates without the need for medical atten-tion
Actions	
1	The user triggers the crash detection system
2	The watch prompts the user if there was an emergency or if he is altright
3	Within n seconds the user chooses the option that nothing happened and the crash detection resets
Alternative Flow **Descrip-tion**	A This time the user is involved in an accident
Actions	
1	The watch triggers the crash detection
2	The user gets prompted to verify the emergency
3	The Emergency services get contacted
Alternative Flow **Descrip-tion**	B The user is involved in an accident and unconscious
Actions	
1	The watch triggers the crash detection
2	The user gets prompted to verify the emergency
3	The user doesn't select an option in n seconds,
4	The emergency services get contacted

4.2.4 Activity Diagram – process of crash detection



4.3 Workout Recorder

While the Workout recorder is currently part of the Health app, as seen in the previous class diagram, it would probably best to make it a separate application. But as all the functionality required by the workout recorder is already available in the classes of the Health app, it is part of it for now.

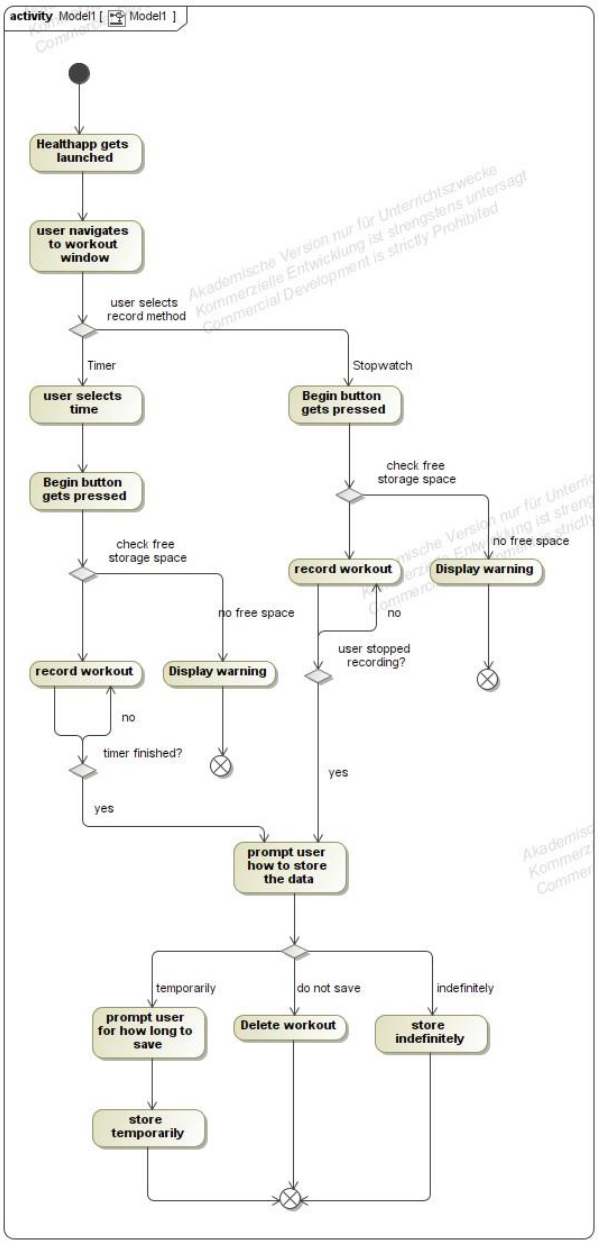
4.3.1Usecase

Name	workout-recorder
ID	1
Description	A feature of the health app to record and save the users condition during a workout. The user can decide between recording either with a timer or a stopwatch. He also can decide how the data should be saved(Indefinitely or temporarily).
Trigger	Actor goes to the workout menu in the health app
Actors	User
Pre-conditions	The application needs permissions to access the watches storage and sensors
Post-conditions	Enough unused storage space
Basic Flow **Description** **Actions** **1** **2** **3**	 The User starts a workout and decides to record with timer instead of stopwatch User enters the length for how long they would like to Exersize User selects 'Begin' and the program starts to record the workout The timer finishes, plays a sound and can select for how long the workout should be stored, if at all.
Alternative Flow **Description** **Actions** **1** **2**	A The user starts a workout without the required permissions(internal error) The user selects the duration of the workout and selects 'Begin' The watch displays an error message with instructions to fix the issue
Alternative Flow **Description** **Actions** **1** **2**	B The user starts a workout but the watches storage is full or reserved for the operating system The user selects the duration of the workout and selects 'Begin' The watch prompts the user to free up storage before recording a workout

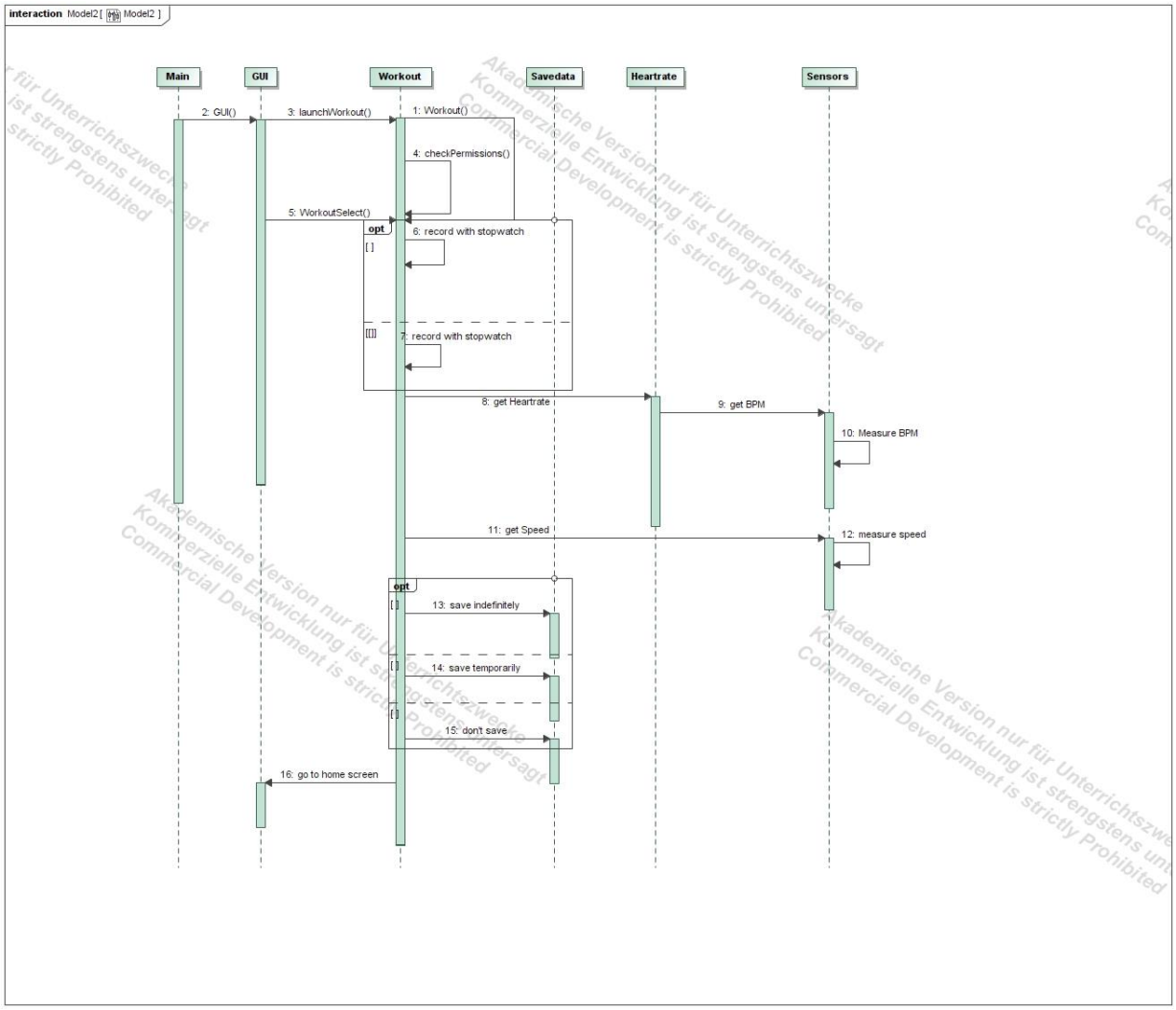
This usecase diagram describes how the user should interact with the workout module of the health app and how the user gets informed when an unexpected error occurs. In this scenario the user decides to record the workout with a timer. The flow and error handling will of course be the same for recording with a stopwatch.

4.3.2 Flow of recording a workout

The Idea for how to record a workout is, that the user opens the health app and goes the menu displaying the workout recorder. They can then decide if they want to record with a stopwatch, meaning they can workout until they are done. They can also record with a timer counting down if they want to workout for a fixed length. Before the program starts recording it does background checks permissions for accessing the sensors of the watch for monitoring the users motions and heartbeat and the storage for reading, writing and possibly deleting data related to the workout. If none of the checks fail, the workout begins and the program starts to record.

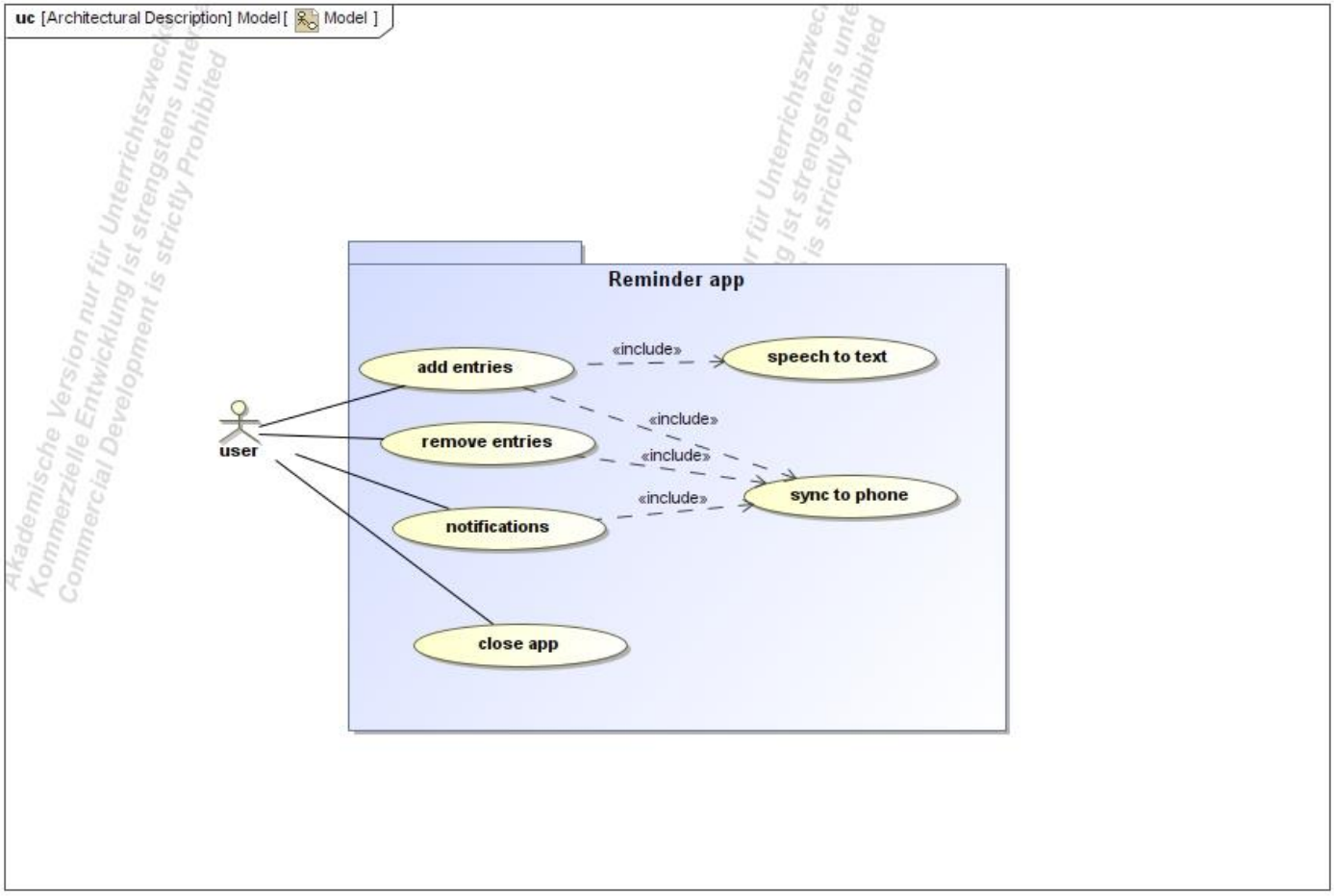


4.3.3Sequence Diagram



4.4Reminder App

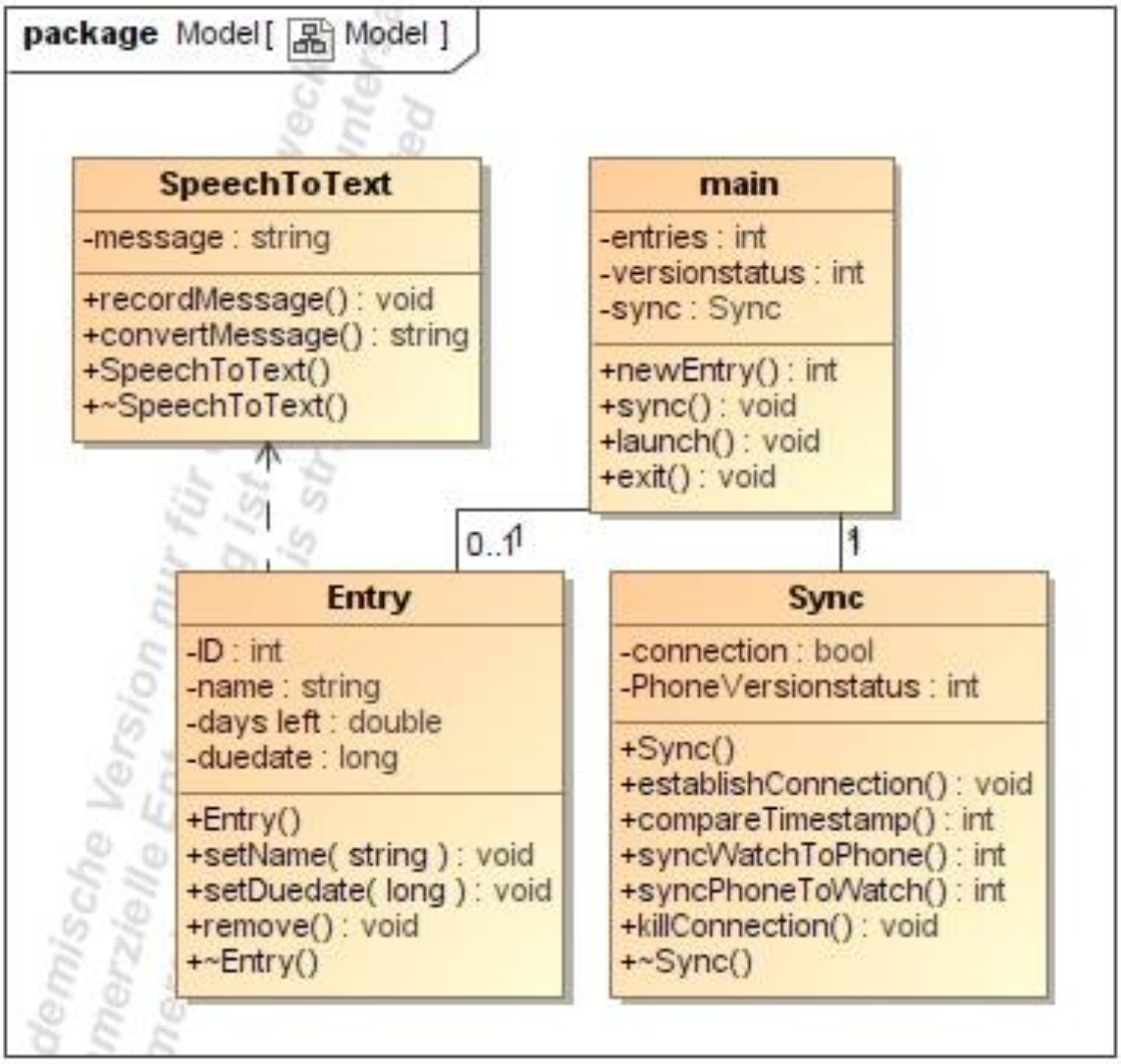
The Reminder App is a very simple application for keeping track of important events. Essentially like every Todo- or Calendar- App ever.



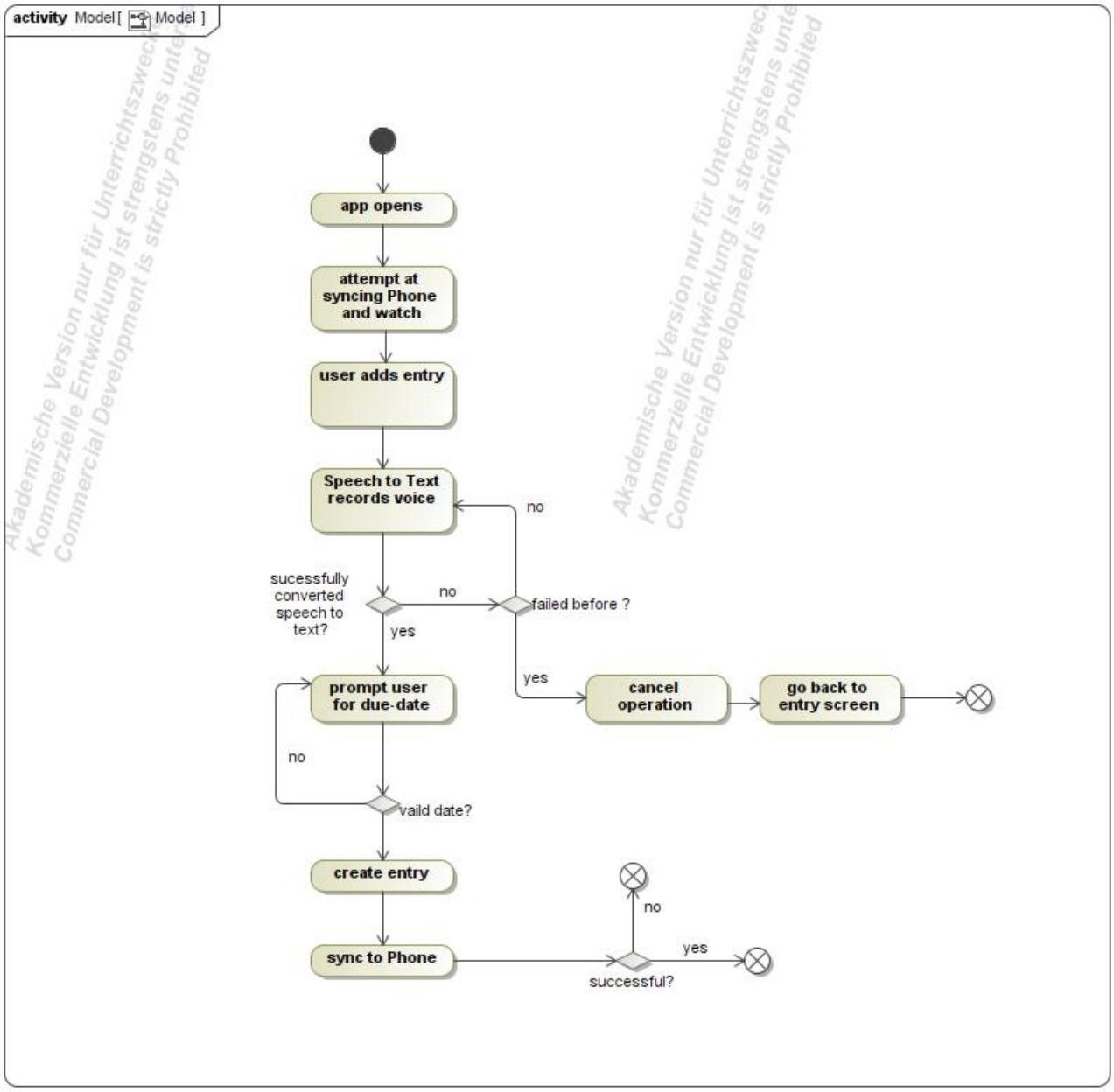
4.4.1Reminder Use case

Name	Reminder app
ID	2
Description	The user wants to plan their everyday life. This app consists of a to-do list where the user can add and remove entries on their watch. They can set due dates and get notifications to remind them about upcoming events. They can also use their phone for more functionality and ease of use.
Trigger	The user opens the app
Actors	User
Pre-conditions	The Reminder app is installed
Post-conditions	All changes are being saved by the program and synced to the phone, if available.
Basic Flow **Description** **Actions**	The user wants to add an entry
1	The user opens the app
2	The user selects the "+" Button
3	The user speaks out their prompt/arrand
4	The app asks the user about the due date
5	The user enters due date and time using a pop up number pad
6	The user selects save, the reminder app adds the entry, saves and synchronizes to the users phone if available
Alternative Flow **Description** **Actions**	A The user wants to remove an entry
1	The user opens the app
2	The user selects the entry/entries they want to remove
3	The user selects the delete button
4	The reminder app removes the entry/entries and syncs the changes to the users phone if available
Alternative Flow **Description** **Actions**	B The user has entries in his reminder and one of them is due
1	The reminder app opens a pop up window and displays the entry
2	The user gets notified and can either remove the entry or extend the deadline

4.4.2 Class Diagram

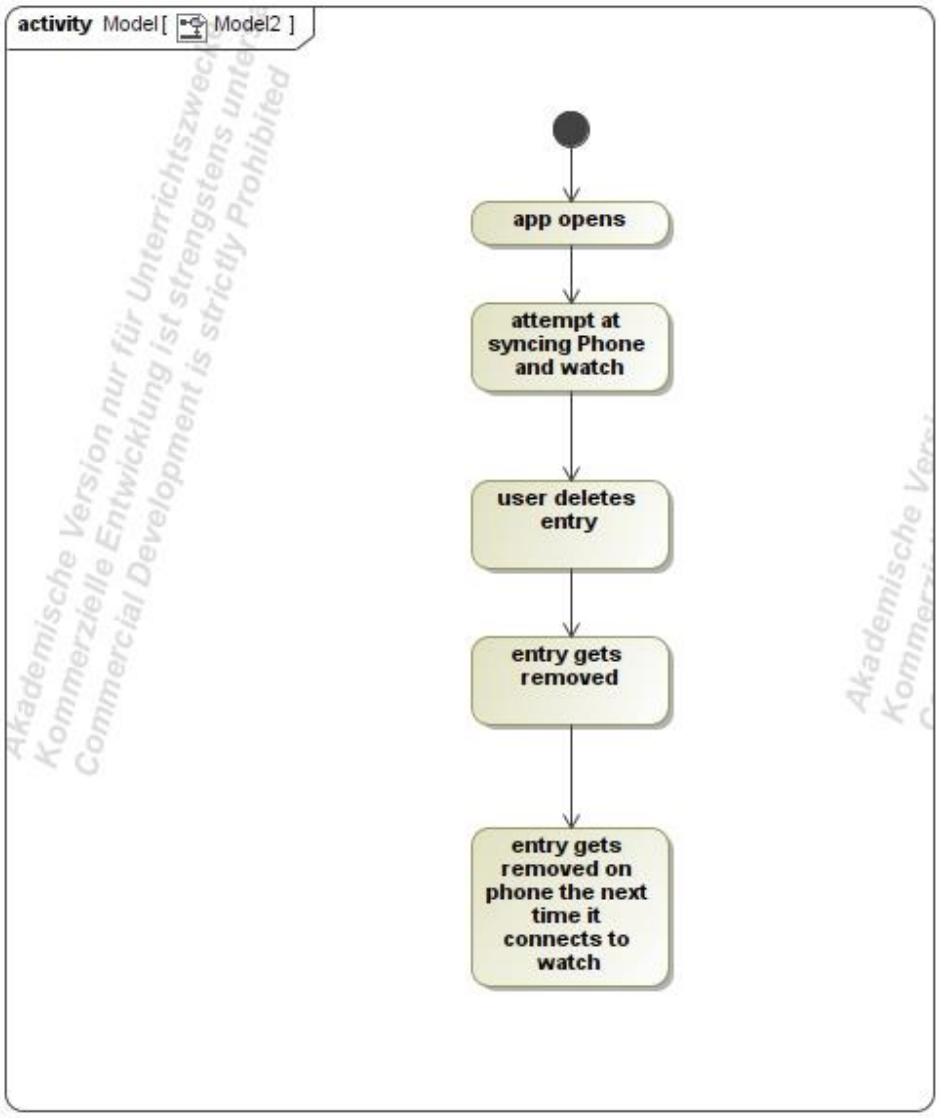


4.4.3Activity Diagram – add entry to reminder app

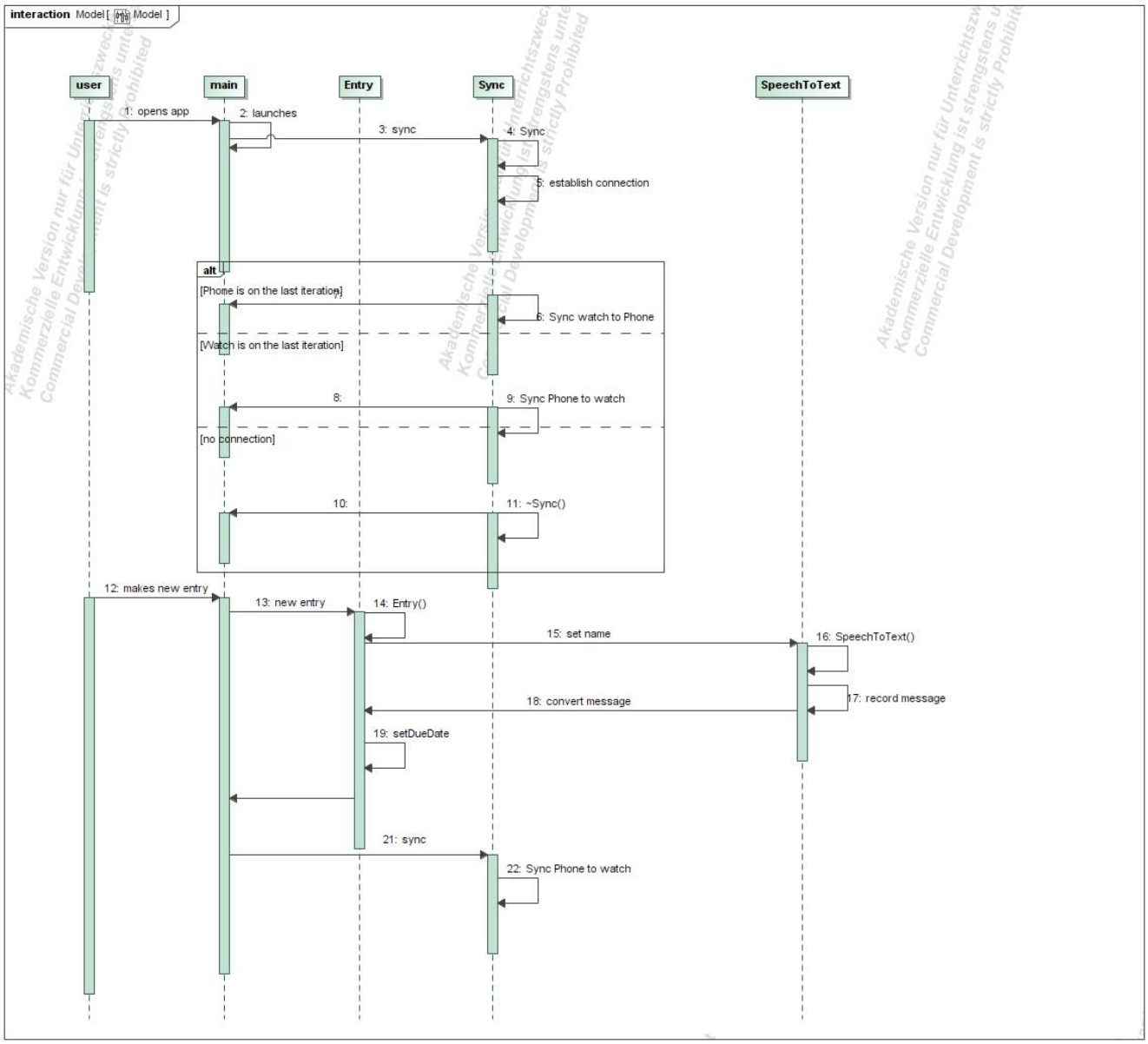


4.4.4 Remove entry from app

This is a very short diagram because it is easier to remove entries than to add them



4.4.5 Sequence diagram for adding an entry



4.4.6 Snow Cards

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Speech to Text

Requirement Type:	functional
For Whom?	customer
User Satisfaction:	medium
User Dissatisfaction:	high

Description:

As it is impossible to but a full screen on screen keyboard on a tiny Display like a smartwatch, alternative input methods are required. Speech to text functionality is therefore useful for example for naming entries for a reminder app

Software Engineering Analysis

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Sync to Phone

Requirement Type:	functional
For Whom?	customer
User Satisfaction:	High
User Dissatisfaction:	High

Description:

The smartwatch should be able to communicate with the users Phone, as many applications run on both devices. The user should not need to do anything twice, if possible. That's why the watch should sync to the Phone and one device updates the other depending on which devices has the lastest changes

Software Engineering Analysis

Wenxue Zhao

5. Dashboards for monitoring information

5.1 Use case for dashboards for monitoring information

5.1.1

Name	Dashboards for monitoring information
ID	5
Description	Users should be able to view a personalized health monitoring dashboard through their smartwatch to comprehensively understand their health and activity information.
Trigger	1.User Activates Watch: The user initiates the functionality by pressing a button, tapping the touchscreen to activate the smartwatch. 2.Voice Activation: The user activates the health monitoring dashboard feature by using voice commands. 3.Gesture Activation:a specific user gesture could be configured to trigger the display of the health monitoring dashboard. 4.Intelligent Alerts: The smartwatch, based on analyzing user activity patterns, health data, or set goals, might automatically remind the user to check the health monitoring dashboard
Actors	user
Pre-conditions	1.The smartwatch is connected to the internet. 2.The user has successfully logged into their smartwatch account.
Post-conditions	The user successfully views a personalized health monitoring dashboard, gaining a comprehensive understanding of their health.
Basic Flow	
Description	The smartwatch shows the health monitoring dashboard
Actions	
1	The user activates the health monitoring dashboard feature on the smartwatch through touch or voice activation.
2	The smartwatch retrieves the user's health and activity data
3	The smartwatch shows the personalized health monitoring dashboard
Alternative Flow	A
Description	Viewing Detailed Information
Actions	
1	The user view detailed view
2	System retrieves and presents comprehensive health and activity data, allowing the user to gain deeper insights.
Alternative Flow	B
Description	Customizing Dashboard Layout
Actions	
1	The user customizing dashboard layout
2	The system guides the user to the dashboard customization settings, allowing them to rearrange and add health metric modules.
3	Once the user completes the customization, the system saves and displays a personalized health monitoring dashboard.

• This case emphasizes users' ability to access real-time data through the Monitoring Information Dashboard.

• User's personalized needs are met, as they can freely choose the information displayed on the dashboard.

• The system has an intelligent feedback mechanism, promptly notifying the user in case of detecting abnormal conditions.

Through this use case, users gain intuitive insights into their health and activity status, with the system providing a comprehensive monitoring experience characterized by real-time updates and customization options.

5.1.2 Use Case Extension

Monitoring Information Dashboard Enhancements:

The Monitoring Information Dashboard can be further enhanced to provide a more comprehensive and user-friendly experience.Some of the key enhancements include:

5.1.2.1 Data Visualization:

The dashboard can incorporate advanced data visualization tools,such as charts, graphs,and maps,to present the user's health and activity data in a more intuitive and easy-to-understand format.This will help users to quickly analyze their performance and identify trends or patterns in their data.

5.1.2.2 Notification Customization:

Users should be able to customize the notification settings according to their preferences. They can choose the type of notifications they want to receive,such as audio, visual, or vibration alerts. Additionally,users can set custom thresholds for triggering alerts,ensuring that they are only notified when genuinely needed.

5.1.2.3 Integration with Third-Party Apps:

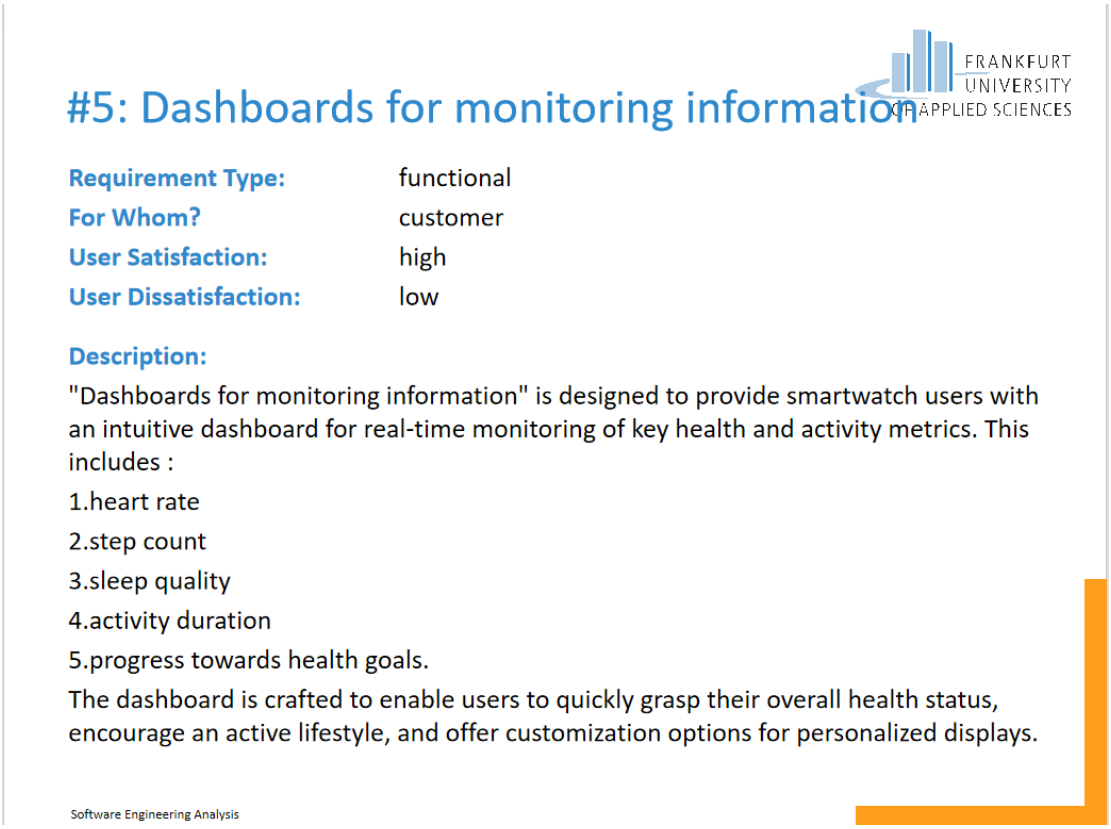
The Monitoring Information Dashboard can be integrated with popular health and fitness apps, such as MyFitnessPal or Nike+,to provide a more seamless user experience.This will allow users to access and analyze their data from multiple sources in one convenient location.

5.1.2.4 User Interface Improvements:

The dashboard's user interface can be updated to enhance usability and aesthetics. This can include improvements such as a responsive design, intuitive navigation,and a cleaner layout. Colors and fonts can be chosen to create a visually appealing and easy-to-read interface,ensuring that users enjoy using the dashboard.

5.2 Snow card for dashboards for monitoring information

Intelligent monitoring real-time information and personalized display on the dashboard



5.2.1 Requirement Categorization

Functional Requirement

This requirement is categorized as a functional requirement as it describes specific functionalities the system needs to provide, namely the design and implementation of the Monitoring Information Dashboard. This includes users being able to access real-time data, customize dashboards to display specific information, and the system triggering alerts or notifications.

5.2.2 Business Value of the Functionality

High

The business value of this functionality is assessed as high because it directly enhances the user experience, allowing users to monitor health and activity data in real-time. Additionally, it addresses the growing market demand for health-conscious wearable technology, contributing to the product's competitiveness.

5.2.3 Advantages

5.2.3.1 Enhanced User Experience

The Monitoring Information Dashboard provides users with a seamless and intuitive interface, tracking their health and activity status in real-time.

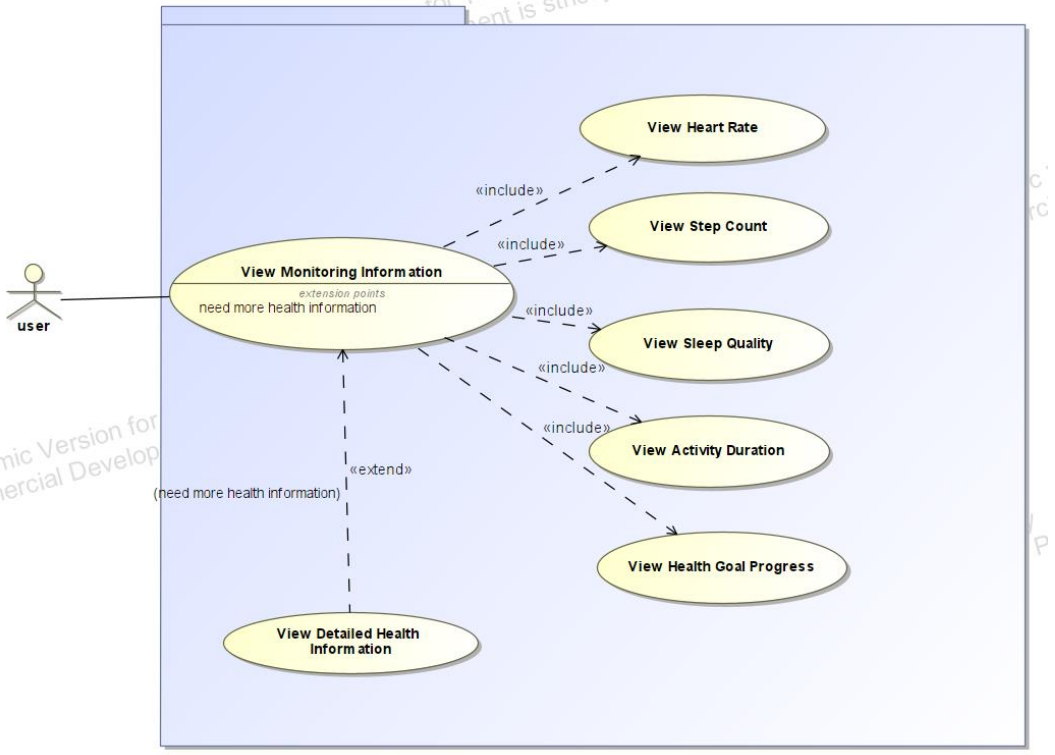
5.2.3.2 Flexibility

Customization options empower users to tailor the dashboard to their individual preferences, focusing on the metrics that matter most to them.

Alignment with Business Goals: This feature aligns with the business goal of delivering a Smart Watch that not only encompasses traditional functionalities but also excels in providing innovative health and activity monitoring solutions. It caters to a market increasingly seeking wearable technologies that prioritize both user experience and health-conscious features.

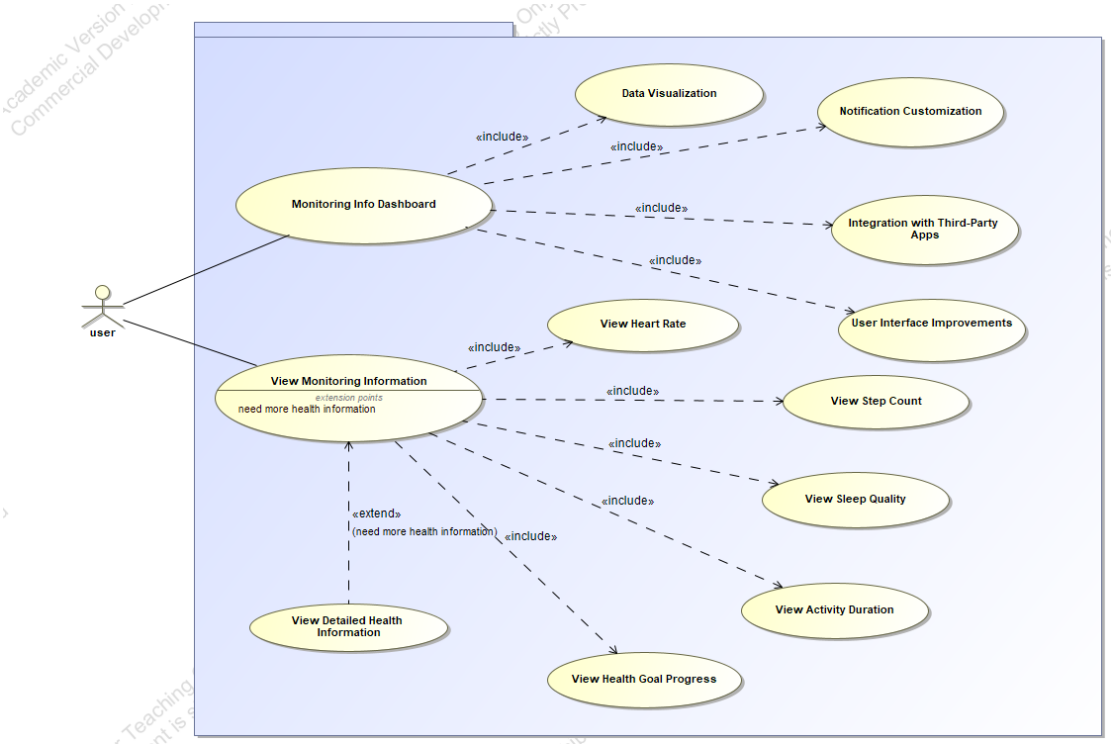
5.3 Use case diagram

The various functions that can be viewed in the dashboard



5.3.1 Use Case diagram extension version

Functional extension



5.3.1.1Use Case details

Detailed Use Case Description: Dashboards for monitoring information

Use Case Name: The various functions that can be viewed in the dashboard

Actor: User

Use Case List:

Use Case Name	Description
1. View Heart Rate	Users can view real-time heart rate information; the system provides heart rate monitoring functionality.
2. View Step Count	Users can view daily and historical step counts; the system provides step monitoring functionality.
3. View Sleep Quality	Users can view sleep quality assessments; the system provides sleep monitoring and evaluation functionality.
4. View Activity Duration	Users can obtain the duration of specific activities; the system records and provides activity monitoring functionality.
5. View Health Goal Progress	Users can check the progress of achieving health goals; the system monitors based on the set objectives.
6. View Health Details	Users have the right to access more detailed health information, including historical data and trend analysis.

5.3.1.2 Relationships

Users interact with all use cases through the Smart Watch, which records and provides relevant monitoring information.

5.3.1.3 Notes

This use case diagram focuses on users accessing health and activity information through the Monitoring Information Dashboard.

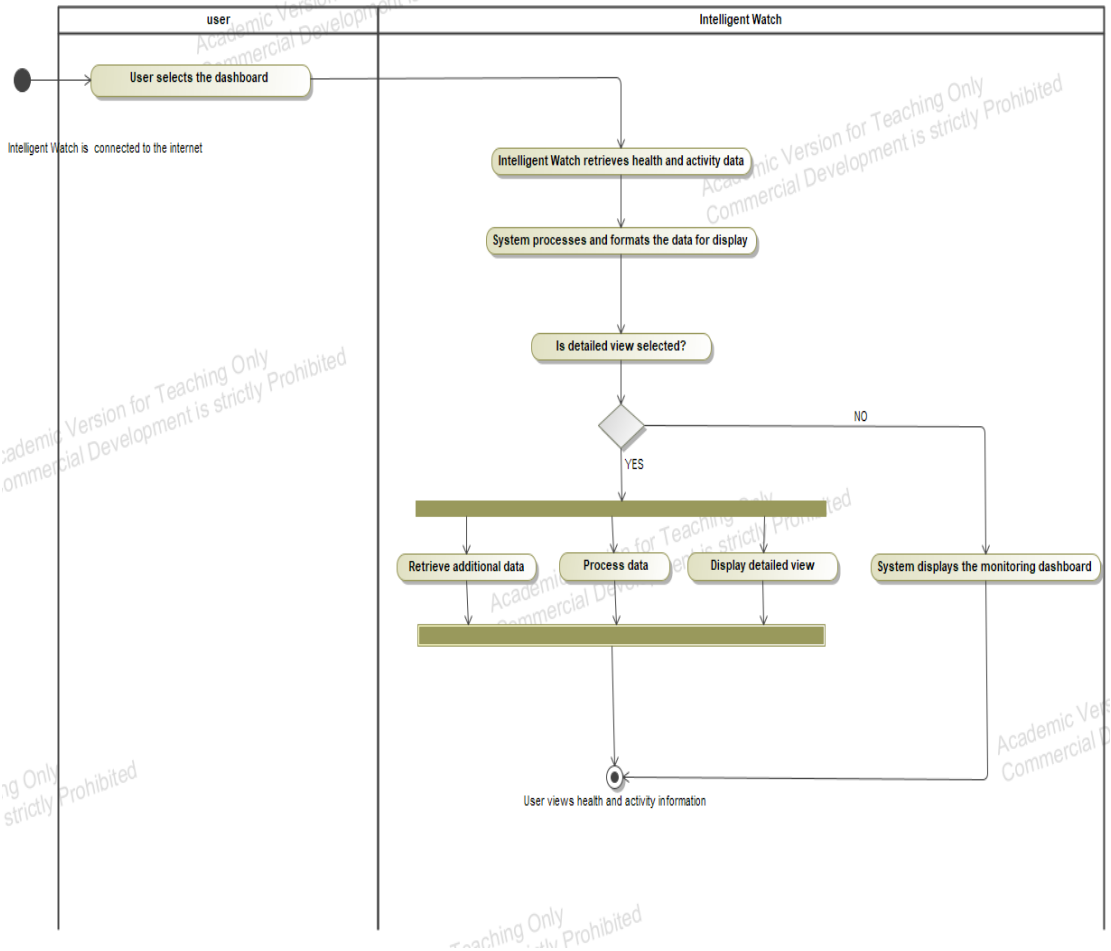
Data sources related to each use case include heart rate sensors, pedometers, sleep monitors, etc.

Setting and monitoring health goals may require user pre-configuration, and the system will monitor based on these objectives.

5.3.1.4 Use Case Diagram Explanation

This use case diagram illustrates the interaction between users and the Monitoring Information System through the Monitoring Information Dashboard. It primarily includes viewing heart rate, step count, sleep quality, activity duration, health goal progress, and more detailed health information. Users conduct these operations through a Smart Watch, which utilizes various sensors to record and provide monitoring information. Users can access this information at any time to monitor their health status and activity progress, allowing for better understanding and adjustments as needed.

5.4 Activity diagram



5.4.1 Activity Diagram Name

The Process of Viewing Health and Activity Information

5.4.2 Activities

- User Selects Dashboard:

The user initiates the activity by selecting the Monitoring Dashboard on the Smart Watch.

The Smart Watch responds to the user's selection.

- Smart Watch Retrieves Health and Activity Data:

Assuming the Smart Watch is connected to the internet, it retrieves health and activity data from the system.

The system processes the raw data and formats it for display.

- Prompt User for Detailed View:

The Smart Watch prompts the user with a message asking if they want to view detailed information.

The user decides whether to proceed to detailed view or not.

- User Chooses Detailed View (Yes):

If the user chooses the detailed view, the Smart Watch retrieves additional data and processes it for display.

The detailed view is presented to the user.

- User Views Health and Activity Information (Detailed View):

The user explores the detailed view of health and activity information.

- User Chooses Direct View (No):

If the user decides not to view detailed information, the system directly displays the Monitoring Dashboard.

- User Views Health and Activity Information (Direct View):

The user explores the health and activity information directly through the Monitoring Dashboard.

5.4.3 Notes:

This activity diagram describes the sequence of actions when a user interacts with the Monitoring Information System.

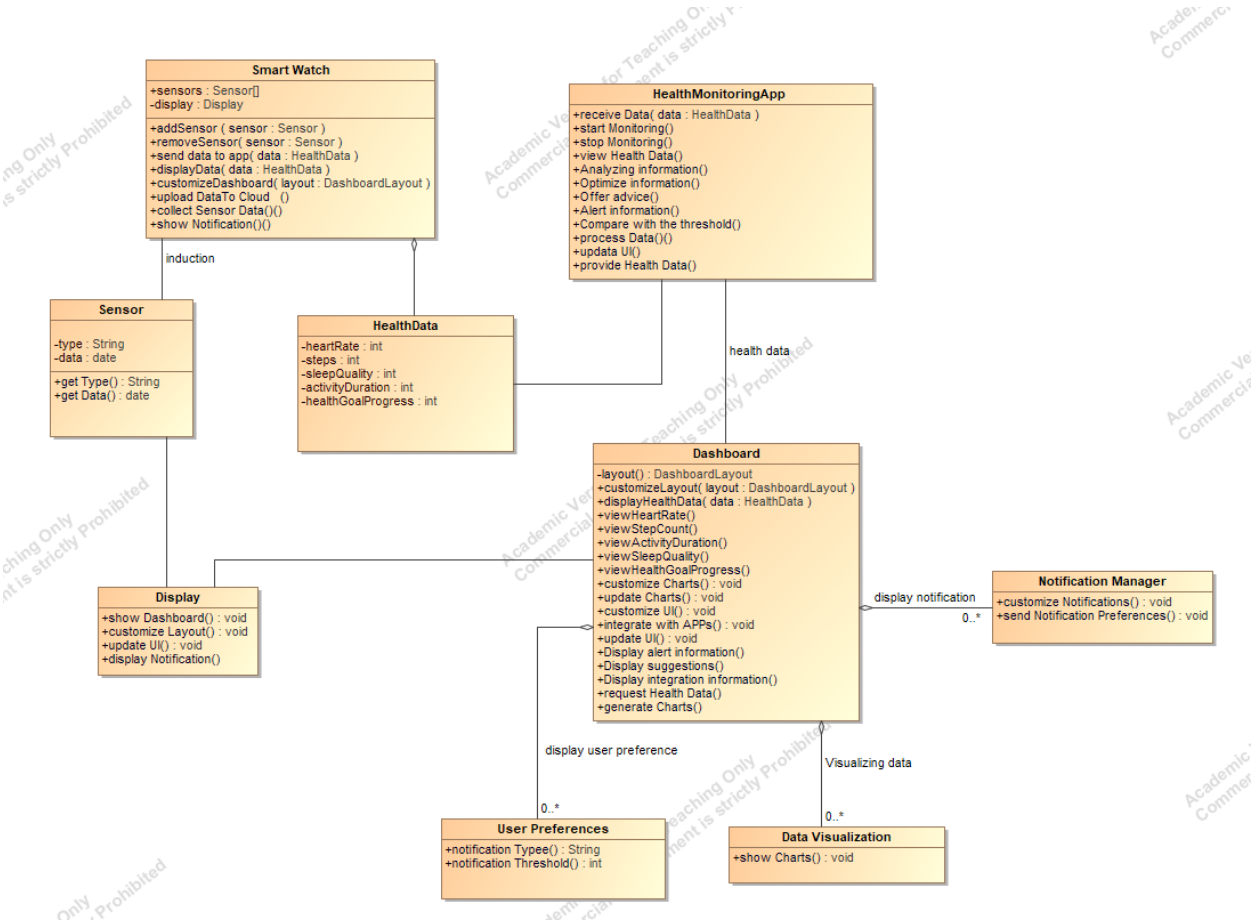
The assumption is that the Smart Watch is already connected to the internet for data retrieval.

The decision point at Activity 3 represents the user's choice to view detailed information or stick to the standard dashboard.

Detailed views involve additional data retrieval and processing before presentation.

The activities illustrate a user-centric flow, providing insight into how users can navigate between standard and detailed views of health and activity information.

5.5 Class diagram



5.5.1 Class Diagram Name

Monitor and display health information

5.5.2 Classes

·Smart Watch Class: The Smart Watch is a crucial hardware device responsible for collecting and transmitting health-related sensor data. This class includes methods for processing this data and interacts with the Health Monitoring App, passing the collected data for further processing and display.

·Health Monitoring App Class: This class is responsible for handling and managing health data from the Smart Watch. It may include methods for communication with the Smart Watch, receiving, and parsing sensor data. The aggregation relationship with the Smart Watch indicates that the Health Monitoring App contains objects of the Smart Watch class, but the Smart Watch can exist independently.

·Health Data Class: This class is used to represent and store health-related data, including various attributes such as heart rate, step count, sleep quality, etc. The aggregation relationship with the Smart Watch indicates that the Smart Watch contains objects of the Health Data class, but Health Data can exist independently.

·Dashboard Class: The Dashboard is responsible for displaying and monitoring summarized information of health data. It includes methods for customizing layouts, displaying real-time data, and providing data trend analysis. The aggregation relationship with Health Data indicates that the Dashboard contains objects of the Health Data class, but Health Data can exist independently. The aggregation relationship with the Health Monitoring App suggests that the Dashboard may also contain objects of the Health Monitoring App class.

·Sensor class

·Display class

·User Preferences class

·Data Visualization class

·Notification Manager class

5.5.3 Relationships

·Aggregation relationship between SmartWatch and HealthData, indicating that the SmartWatch contains and uses health data.

·Aggregation relationship between HealthData and Dashboard, signifying that health data is included and used to construct the monitoring dashboard.

·Association relationship between Dashboard and HealthMonitoringApp, representing communication between the monitoring dashboard and the health monitoring app.

Notes:

·The class diagram primarily includes key classes related to the monitoring information dashboard.

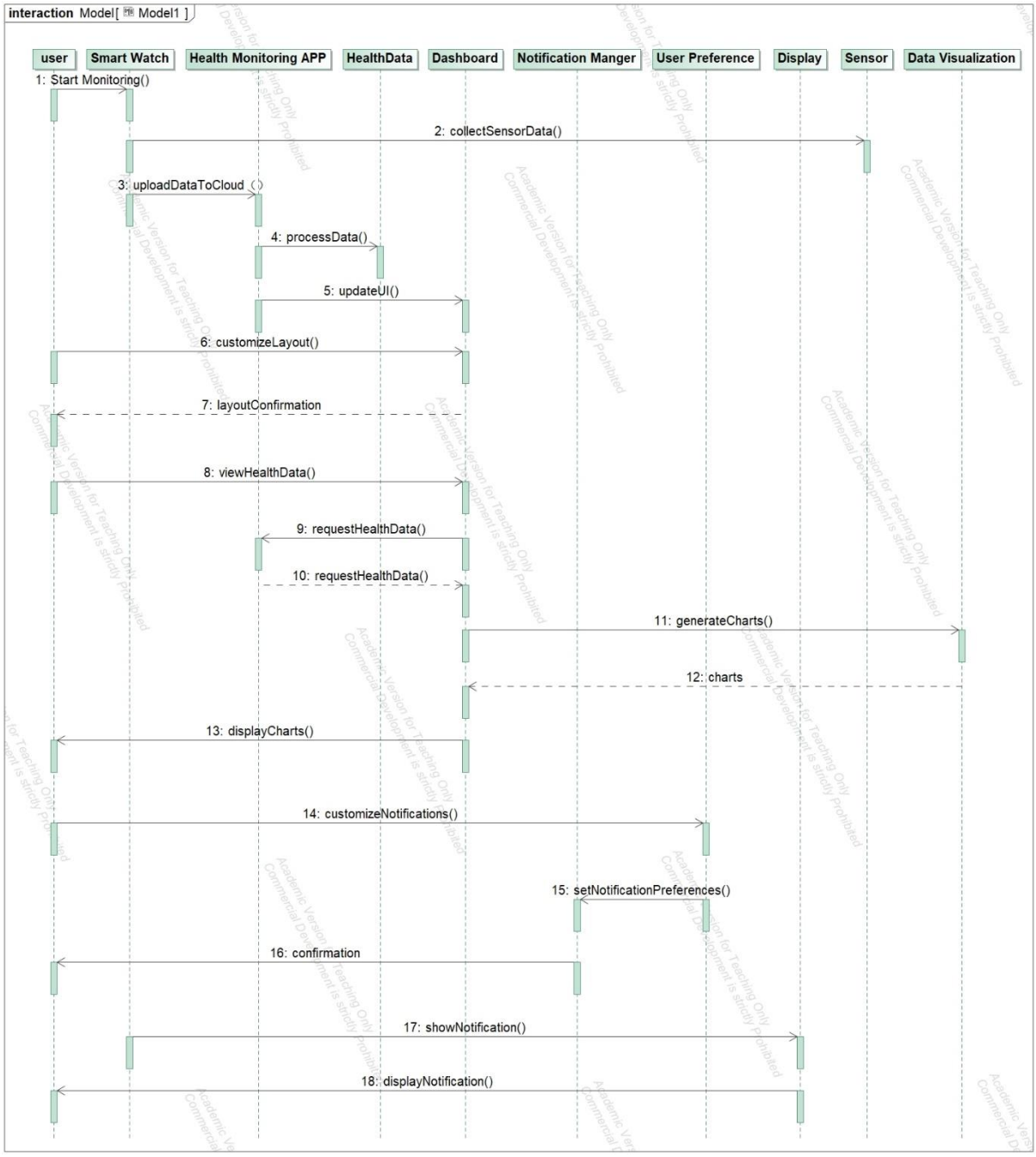
·Aggregation relationships denote a whole-part relationship, with the SmartWatch containing and using health data, and health data being included in the construction of the dashboard.

·The association relationship illustrates communication between the dashboard and the health monitoring app.

·These relationships more accurately reflect the dependencies and interactions between the classes.

·The purpose of generating this class diagram is to clearly depict the relationships between core components in the system. It aids developers and designers in better understanding the system architecture, facilitating the design and implementation of the system.

5.6 Sequence diagram



5.6.1 Sequence diagram Name

The sequence of health information displayed on the dashboard

5.6.2 Note

This sequence diagram illustrates the interaction flow between various participants in a system named "Monitoring Sequence Diagram." Here's an explanation of the main steps in the diagram:

• User starts monitoring (start Monitoring):
The user initiates the monitoring process through Smart Watch.
Smart Watch begins collecting sensor data.

• Upload data to the cloud (upload Data to Cloud):
Smart Watch uploads the collected sensor data to the cloud database.
Health Monitoring App receives the uploaded data.
Health Monitoring App processes and stores the data in the cloud.

• Update dashboard interface (update UI):
Health Monitoring App retrieves processed data from the cloud.
Health Monitoring App updates the user interface of the Dashboard to display the latest health information.

• Customize dashboard layout (customize Layout):
The user customizes the layout of their health monitoring dashboard through Dashboard.
Dashboard confirms the customized layout to the user.

• View health data (view Health Data):
The user views their health data through Dashboard.
Dashboard requests Health Monitoring App to provide health data.
Health Monitoring App supplies the data to Dashboard.

• Generate and display charts (generate Charts and display Charts):
Dashboard utilizes Data Visualization to generate charts for health data.
The generated charts are displayed on Dashboard for user viewing.

• Customize notification preferences (customize Notifications):
The user customizes their notification preferences through User Preferences.
User Preferences passes the notification preference settings to Notification Manager.

• Set notification preferences (set Notification Preferences):
Notification Manager configures the user's notification preferences.
Notification Manager confirms the user's notification preference settings.

• Display notifications (show Notification and display Notification):
Upon receiving a notification, Smart Watch uses Display to show the notification.
Display presents the notification to the user.

The entire process covers steps such as starting monitoring, uploading data to the cloud, updating the dashboard interface, customizing layout, viewing health data, generating and displaying charts, customizing notification preferences, setting notification preferences, and displaying notifications. It demonstrates the collaborative work of various components in this monitoring system.

5.7 For UI Design Regarding Data Visualization

The dashboard can incorporate advanced data visualization tools, such as charts, graphs,and maps, to present the user's health and activity data in a more intuitive and easy-to-understand format. This will help users to quickly analyze their performance and identify trends or patterns in their data.

5.8 Use case for User Social Interaction Features

5.8.1use case: Comments and Likes

Name	Commenting and Liking Health Data
ID	6
Description	Users can view and interact with the health and fitness data of other users in the health monitoring application by commenting on or liking the data.
Trigger	The user wants to interact with the health data of other users by expressing opinions or The user wants to interact with the health data of other users by expressing opinions or preferences.
Actors	user
Pre-conditions	1.The user has successfully logged into the health monitoring application and can access the health data of other users.
Post-conditions	The comment or like information is visible under the related post.
Basic Flow **Description** **Actions** 1 2 3	Commenting and Liking the post The user navigates to the "Social Feed" or "Activity Stream" section of the application. The user selects a specific post to comment on or like. The user can choose one of the following actions: Write a comment: The user enters the comment content. "Like": The user clicks the "Like" button.

5.8.1.1

Description: Enable users to post comments and give likes on the fitness and health data of other users. This feature encourages positive interaction among users, contributing to the creation of a supportive social environment.

Objective: Increase interaction among users, fostering a positive social environment through comments and likes.

5.8.2use case: Social Sharing

Name	Share Health and Fitness Achievements
ID	7
Description	This use case involves the user sharing their health and fitness achievements on a social media platform through the Health Monitoring App.
Trigger	The user wants to showcase and celebrate their health and fitness achievements on a social media platform.
Actors	user
Pre-conditions	The user is logged into the Health Monitoring App and has achieved health or fitness milestones.
Post-conditions	The user's achievements are successfully shared on the chosen social media platform.
Basic Flow	
Description	Share Health and Fitness Achievements
Actions	
1	The user navigates to the "Social Interaction" section within the app.
2	The user selects the "Share Achievements" option.
3	The app generates a formatted post with the user's selected health and fitness achievements.
Alternative Flow	A
Description	Cancel Sharing
Actions	
1	At any step during the process, the user can cancel the sharing operation.

5.8.2.2

Description: Users should be able to share their health and fitness data on social media platforms such as Facebook, Twitter, or Instagram. This allows users to showcase their fitness achievements to their social circles, fostering positive interaction among users.

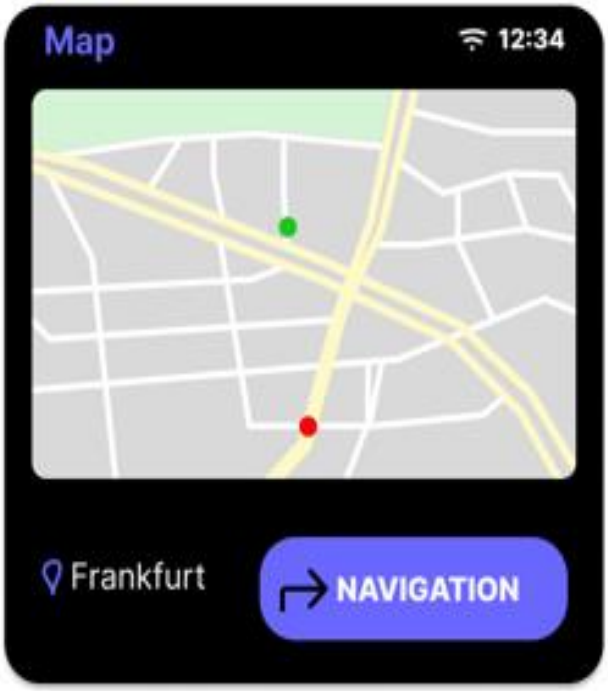
Objective: Increase user visibility on social media by showcasing their fitness achievements.

Karthik Bhat

6.1 Motion and Positioning Data – GPS

6.1.1 Introduction to GPS

The Global Positioning System, commonly known as GPS, is a satellite-based navigation system that provides precise positioning and timing information anywhere on Earth. This technology relies on a constellation of orbiting satellites that transmit signals to GPS receivers. And I have used this GPS Technology in our Smartwatch, which helps the User in many ways.This Smartwatch which is equipped with GPS, allows the User to track their precise location in real-time, for example running, hiking, or cycling, where the User can monitor their routes and distance 95escribes. It also allows the User to navigate themselves to a certain Location by giving directions. In the event of an emergency, the Smartwatch can provide accurate location information to emergency services. This feature enhances personal safety.



How does GPS work ??

GPS satellites circle the Earth twice a day in a precise orbit. There are 24 Satellites orbiting the Earth. Each satellite transmits a unique signal that allow GPS devices to decode and compute the precise location of the satellite. GPS receivers use this information to calculate a user's exact location. The GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a User's position and display it to measure User’s running route, find a way home.

What is GPS Lock ??

To calculate your 2D position (latitude and longitude) and track movement, a GPS receiver must be locked onto the signal of at least three satellites. With four or more satellites in view, the receiver can determine your 3D position (latitude, longitude and altitude) and compute accurate position and time. This is called GPS Lock.

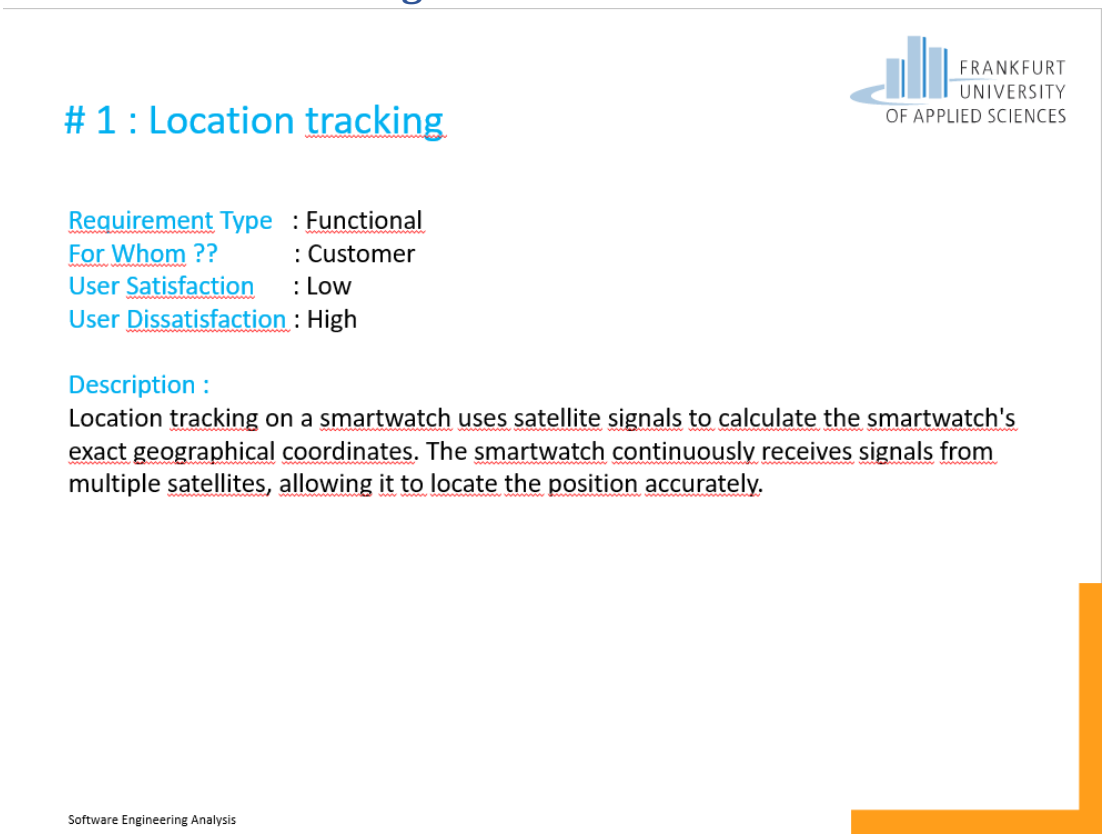
6.1.2 Use Case for Tracking Motion and Positioning Data – GPS

Use Case Name	Tracking Motion and Positioning Data - GPS
ID	1
Description	GPS in a Smart watch provides the User with accurate location data, enables navigation, and improves the overall User experience
Trigger	User trying to find his/her Location
Actors	User, GPS System
Pre-condition	User should have GPS enabled Device which is powered ON and fully functional
Post-condition	GPS System monitors and provides accurate Location data
Basic Flow	• User activates GPS application on the Smart Watch • The Watch establishes a connection with GPS Satellite • User searches for his/her Location or Destination • User starts the navigation to the Location.
Exceptional Flow	1) User experiences Signal loss: • Watch loses connection with the GPS Satellite because of Infra-structure like Tunnels, Skyscraper Building etc • Watch informs the User that the Location my not be accurate • It tries to reconnect with Satellite
Alternate Flow	2) System goes Offline Mode: • If the device is in Offline mode, The GPS system notifies the user that it may not have access to real-time map data. • The system informs the user about limited functionality due to offline mode.

6.1.3 Requirement Table

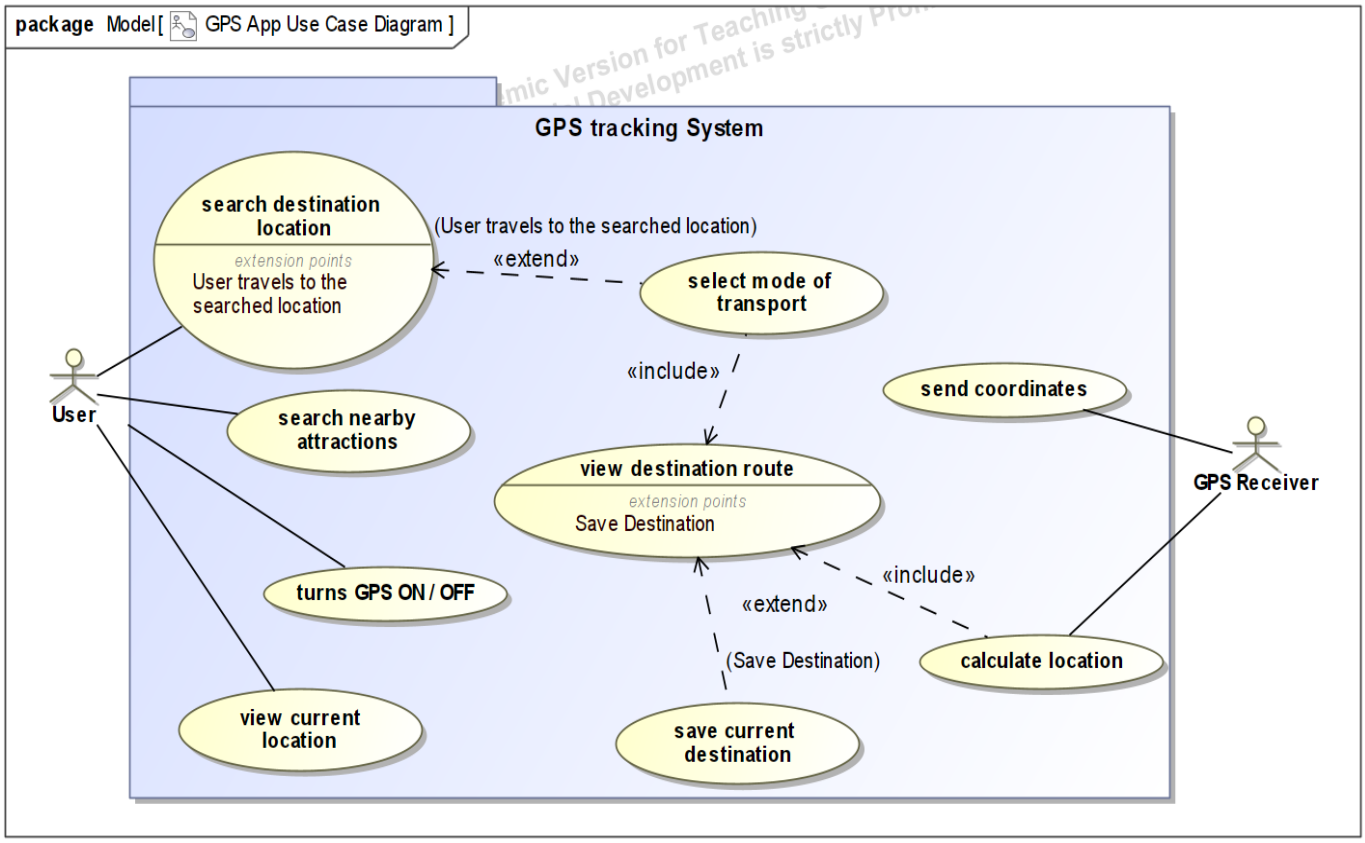
ID	Requirement	Requirement type	Business Value
1	Location Tracking	Functional	High
2	Map Diaplay	Functional	High
3	Navigational Assistance	Functional	Mid
4	Accracy and Precision	Non Functional	Low
5	Fast Respone Time	Non Functional	Low
6	Power Effcient	Non Functional	Mid
7	Weather Information	Non Functional	Mid
8	Location History	Non Fuctional	Mid
9	Offline Support	Non Functional	High
10	Location Sharing	Non Functional	Mid

6.1.3 Location Tracking



Out of all the Requirements from the Requirements Table, I selected Location Tracking and designed all UML Diagrams based on Location Tracking needs, because its one of the most functional and important Requirement.

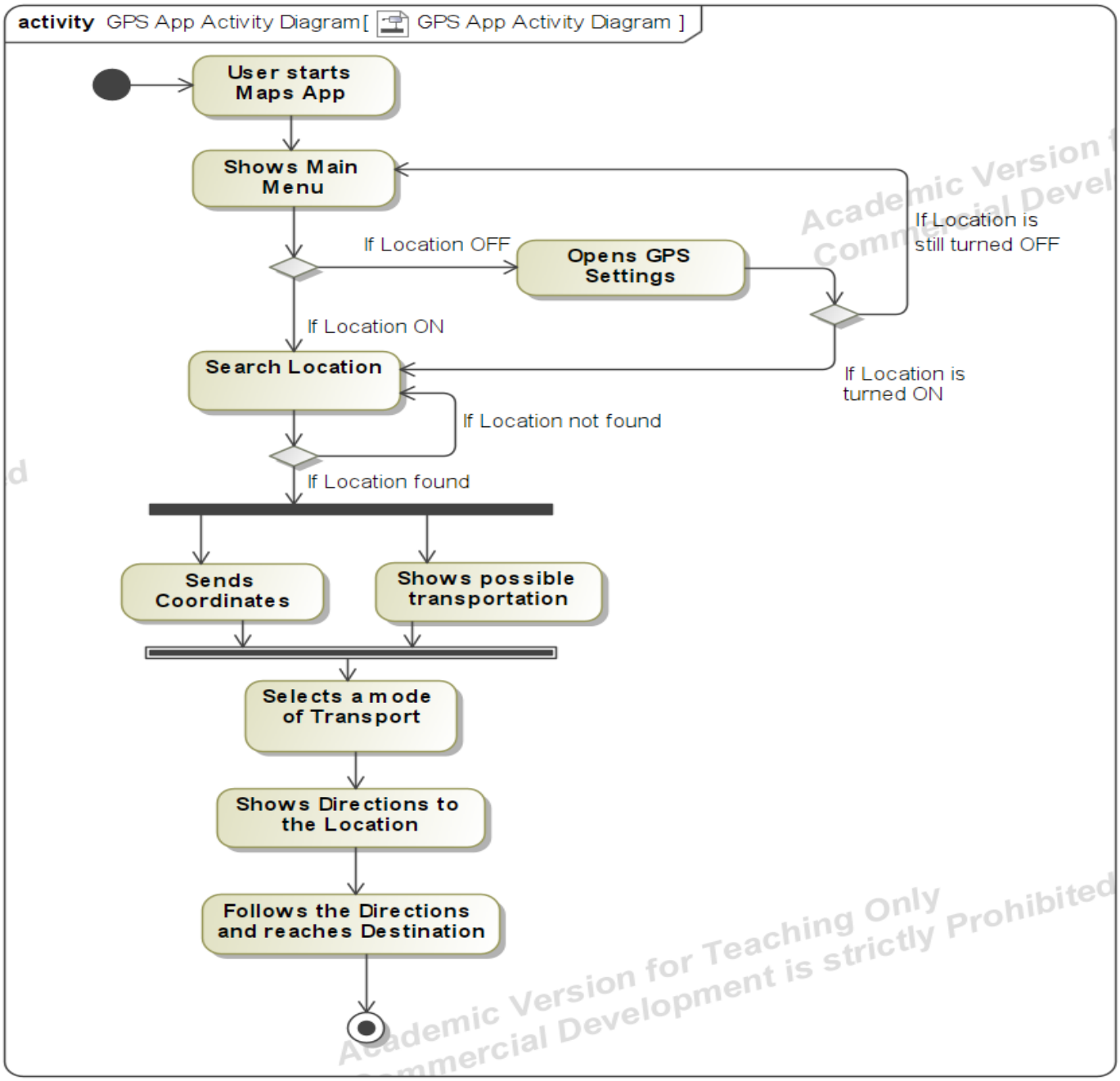
6.1.4 Use Case Diagram for Location tracking



This Use Case Diagram describes how the User would interact with the GPS System. The User can turn ON / OFF the GPS in his/her Device, view his/her current Location, search desination and also nearby Attractions etc. For example the User needs to travel to a particular Location. The User then can Search the Location. Afterwhich the User gets an option of choosing the mode of Transport, with which the User is going to reach the Location. After selecting the appropriate mode of transportation The User gets the suggested Route to this Location. After the User reaches the Location, the User simply can exit the Map. On the other hand we have to the GPS Receiver which Calculates location and sends coordinates.

Use case diagrams provide a representation of the system's functionalities. It is very helpful for stakeholders, Product Owners, developers, to understand the requirements of the GPS system on the smartwatch. Business value is often realized through the development of features. Use case diagrams act as a roadmap for implementing these features. These Features and other information are valuable for marketing the product in the market.

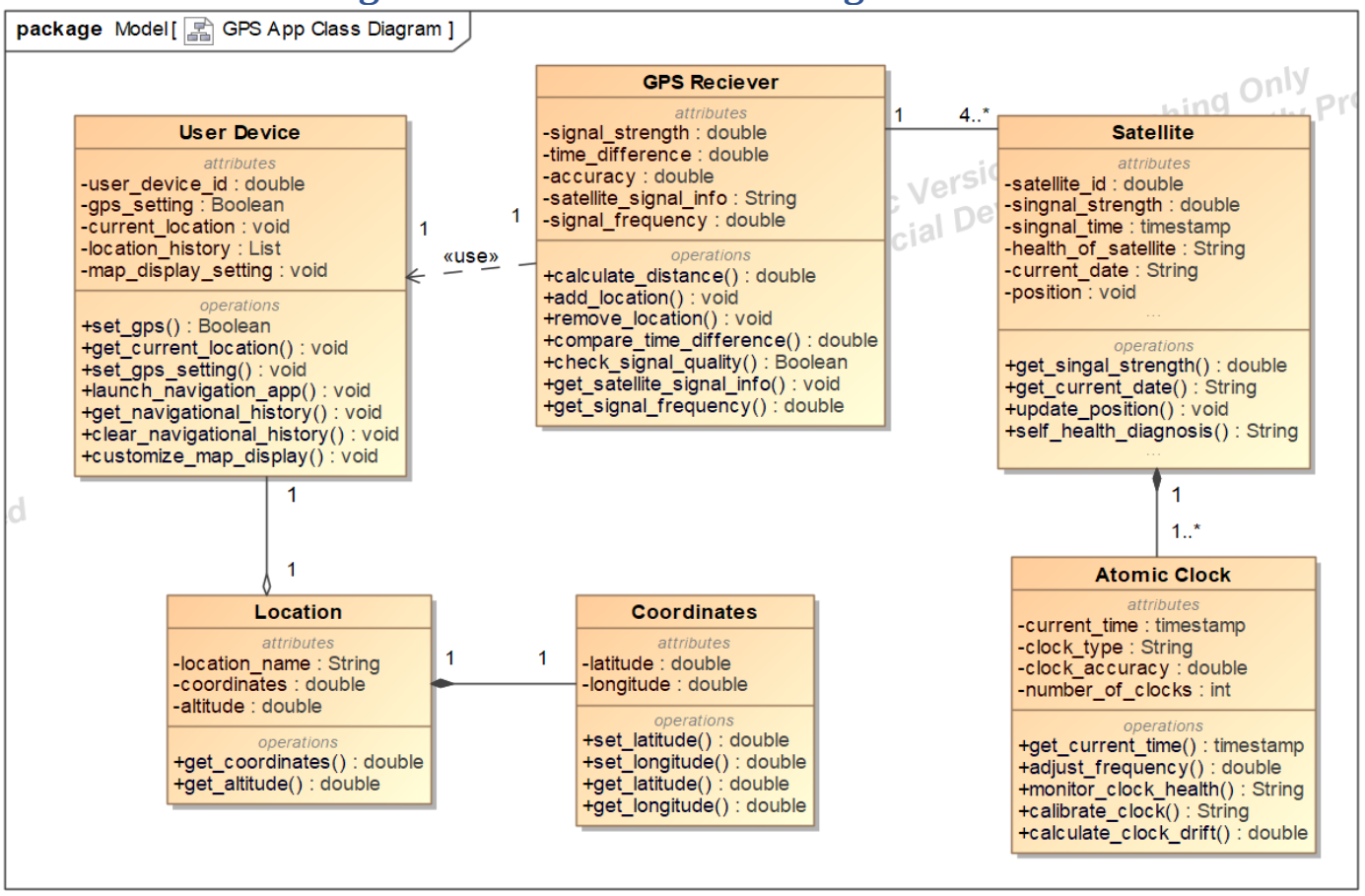
6.1.5 Activity Diagram for Location tracking



Based on the earlier Use Case Diagram, I designed a similar Activity Diagram. The activity diagram begins with the start state, when the User starts the Maps App. The Activity Diagram ends, When the User follows the given Directions and reaches the Destination.

This activity diagram provides a overview of the key steps and interactions involved with GPS System in the smartwatch. It helps in visualizing the flow of activities and events from user interaction to the presentation of location-related information.

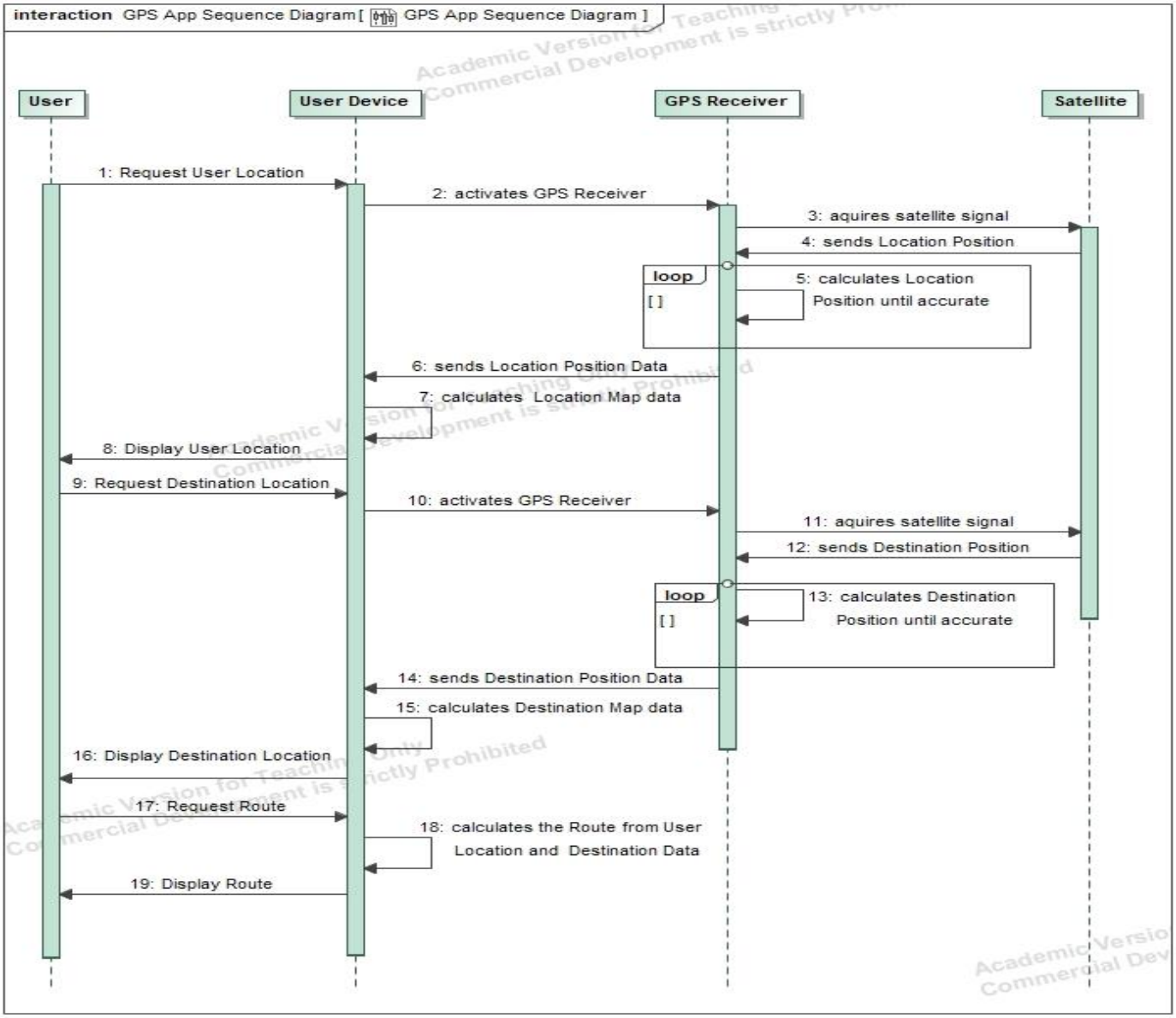
6.1.6 Class Diagram for Location Tracking



This class diagram for GPS in a smartwatch represents the relationships and structures of key classes involved in the GPS functionality. In this Class Diagram, there are 3 main Classes (User Device, GPS Receiver, Satellite)

- The Location Class is connected to Coordinate Class with the composition relationship with the Cardinality 1 , because every Location has one and only unique Coordinates and without Location Class Coordinate Class cannot exist.
- Every User Device has one Location, and instead of writing Location as a method of User Device Class, I connected User Device Class with Location Class with Inheritance relationship.
- The User Device Class is then connected to GPS Receiver Class, and GPS Receiver Class is further connected to Satellite Class with Association relationship.
- Inoder for the GPS System to be accurate, the GPS System has to be connected to atleast 4 or more Satellite, thy why the Cardinality for Satellite between GPS Receiver Class and Satellite Class is 4..*
- The Satellite Class is further connected to Atomic Clock Class, wherein every Satellite has atleast 1 Atomic Clock.

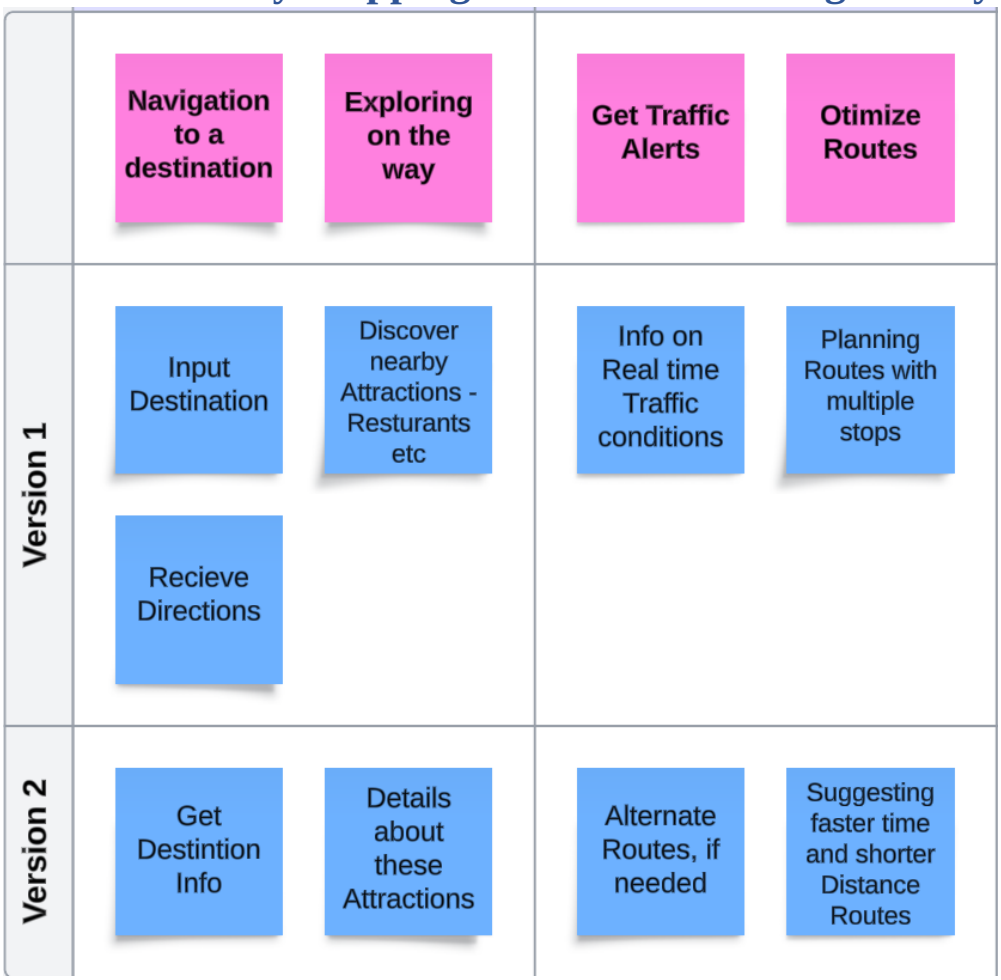
6.1.7 Sequence Diagram for Location Tracking



This Sequence Diagram for GPS in a smartwatch shows the interactions in chronological order of events between various actors involved in GPS functionality. This is a Sequence Diagram based on the earlier Class Diagram, wherein we can see what exactly happens when the User initiates and requests User and Destination Location and its Route. The Sequence Diagram has 3 Classes (User Device, GPS Receiver, Satellite) from previous Class Diagram and not all Classes, because the other Classes cannot be integrated in a Sequence Diagram.

6.2 Advanced Navigation System

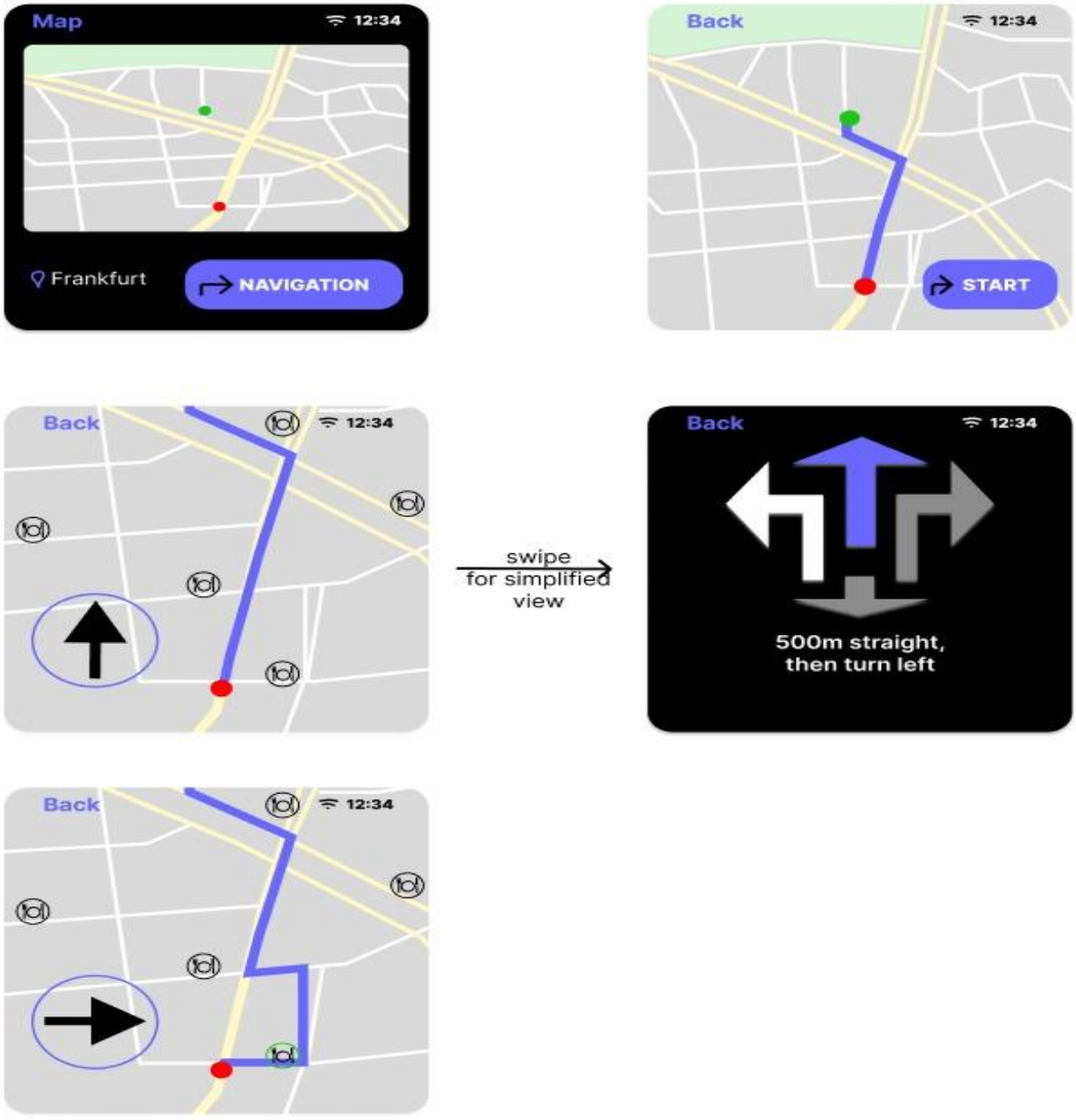
6.2.1 User Story Mapping for Advanced Navigation System



This User Story Map provides a visual representation of the User's journey and helps prioritize things based on User needs and preferences.

The new Feature of this Smartwatch is that, it gives you different options of nearby Attractions on your way to the Location. For example Lets say, You are travelling with your Family to a different City for a Vacation. You Start your Journey with the help of the Smartwatch giving you Directions to your Destination. On the way, the Smartwatch shows different nearby Attractions, one of which is Gas Station, and Suddenly you remember that you have low Fuel in your Vehicle and then you can fill your Gas Tank. Then on the way, the Smartwatch informs you that there is a Traffic Problem on the way, and it suggests you a better Route to your Destination, helping you save a lot of time and fuel. Then the Smartwatch gives you Recommendations and Info of nearby Restaurants on the way, where you can get good Food and etc. Most of the Smartwatches does not have this Feature, which makes this Smartwatch special and has a Unique Selling Point in the Smartwatch Industry.

6.2.2 UI Prototype for Advanced Navigation System

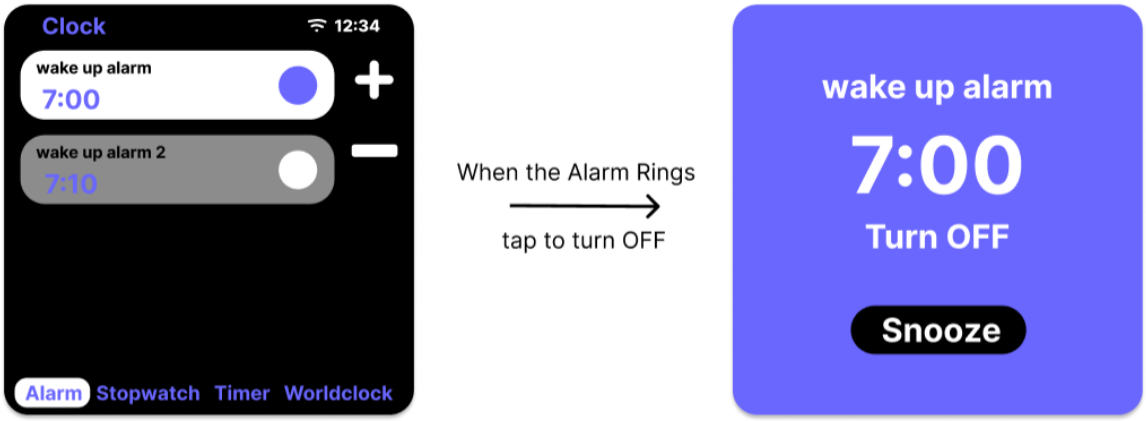


Based on the earlier User Story Mapping for the Advanced Navigation System, some UI Prototype were made. Just to get an Idea how the Advanced Navigation could look like. This is something new that I was working on and as its non Functional Requirement, not many Users might use this Feature, thats why no UML Diagrams were created, just UI Prototypes.

6.3 Clock App – (Timer, Stopwatch, Alarm)

6.3.1 UI Prototype for the Clock App

6.3.1.1 Alarm

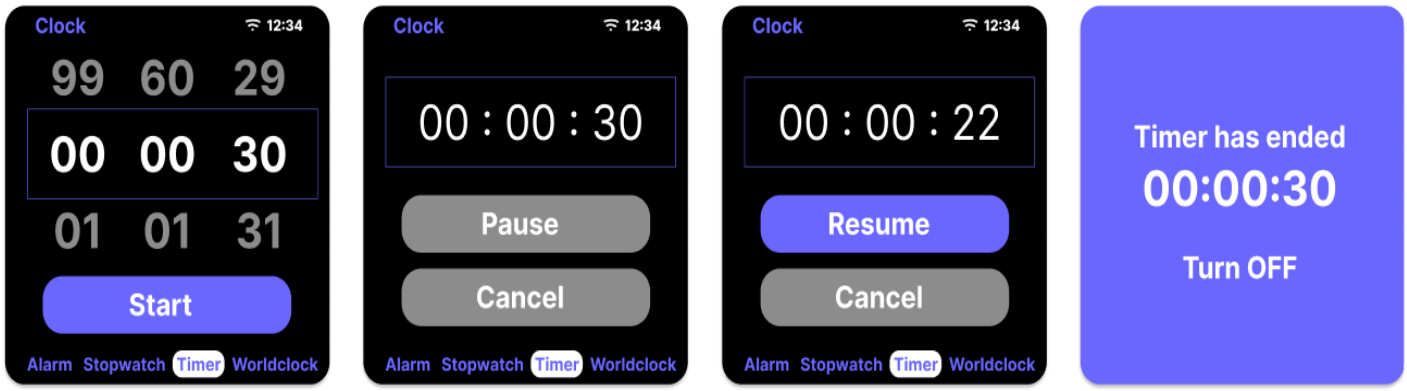


The main screen of the Clock App displays a list of existing alarms. Each alarm could be having further details such as the set time and alarm tone. Users can add a new alarm by tapping a + button, and delete with – button. Additional settings such as snooze duration, vibration for the alarm are also available. A save button confirms the alarm settings and adds the new alarm to the list. When the Alarm rings the User can tap the Turn OFF button.

6.3.1.2 Stopwatch

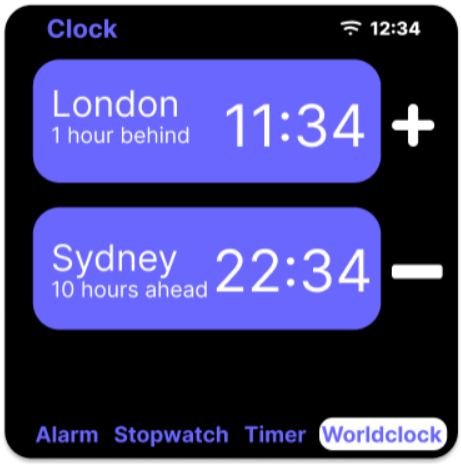
The stopwatch feature is accessible from the Clock App's main screen. The stopwatch has a interface that displays elapsed time and updates in real-time as the stopwatch runs. The User can start the Stopwatch just by clicking the Start button, there are also buttons for lap , restart , stop. A lap button allows users to record lap times during timing. To reset the time on Stopwatch, the User can click restart button.

6.3.1..3 Timer



The timer feature is also accessible from the Clock App's main screen. Users can set the desired countdown duration using a sliders for hours, minutes, and seconds. Start, pause, resume and cancel buttons are provided for controlling the timer. The remaining time is displayed and updating as the timer counts down. Users receive audible and visual notifications when the timer reaches zero, also by vibration if enabled. User can stop the Timer with the Turn OFF Button.

6.3.1.4 World Clock



The main screen of the world clock displays a list of cities, time zones, and time delay. Users can add a new city to the world clock by tapping a + button, and remove with - button. An option is provided to automatically detect and add the user's current location to the world clock.
* World Clock not featured in Use Case Diagram, because 3 Use Case Diagrams for Alarm, Stopwatch, Timer are already made.

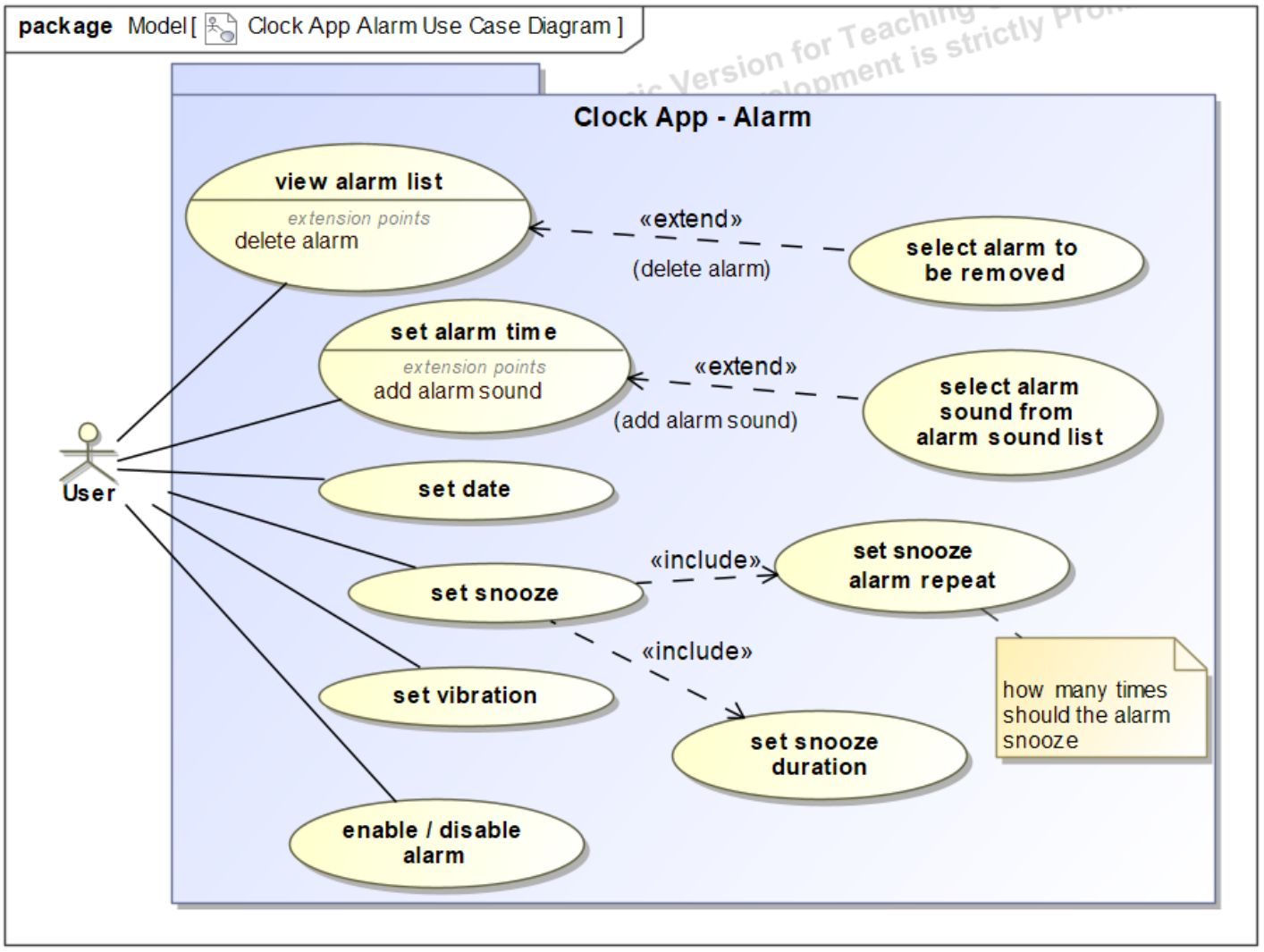
6.3.2 Use Case for the Tracking Time

Use Case Name	Tracking Time with Timer, Alarm, Stopwatch, World Clock
ID	2
Description	The Clock App has alot of features like Timer, Stopwatch, Alarm etc. It is a valuable tool for various time-related activities such as waking up, cooking, exercising, and more
Trigger	Timer - user sets a specific duration and then starts the countdown. Alarm - user sets for a specific time in the future. When the current time matches the set alarm time, the alarm ring. Stopwatch – user wants to measure the elapsed time of an event.
Actors	User
Pre-condition	The Clock App is installed and accessible on the user's smartwatch
Post-condition	App display updates in real-time to show elapsed time, time remaining for Alarm to ring etc
Basic Flow	• User starts Clock App on smartwatch • User starts time tracking (Turns Stopwatch / Alarm / Timer ON) • System shows elapsed time on Stopwatch / existing Alarms / remaining time on Timer • User can stop or lap times in Stopwatch / edit or delete Alarms / pause or cancel time in Timer
Alternate Flow	1. User cancels/stops time tracking • The Clock App resets time and goes to Main menu • The System also gives the User opportunity to track time again • User can start time tracking or close the Clock App

6.3.3 Requirement Table

ID	Requirement	Requirement type	Business Value
1	Start / Stop time tracking	Functional	High
2	Recieve Notification	Functional	Mid
3	Sharing / Exporting Time date	Non Functional	Mid
4	Manual Time Entry	Non Functional	Low
5	Integration with Calendar	Non Functional	Mid
6	Compatibility to other Devices	Non Functional	High
7	Bed Time Mode	Non Functional	Mid
8	Event Countdown	Non Functional	Low
9	Multiple Time Zone Display	Non Functional	Mid

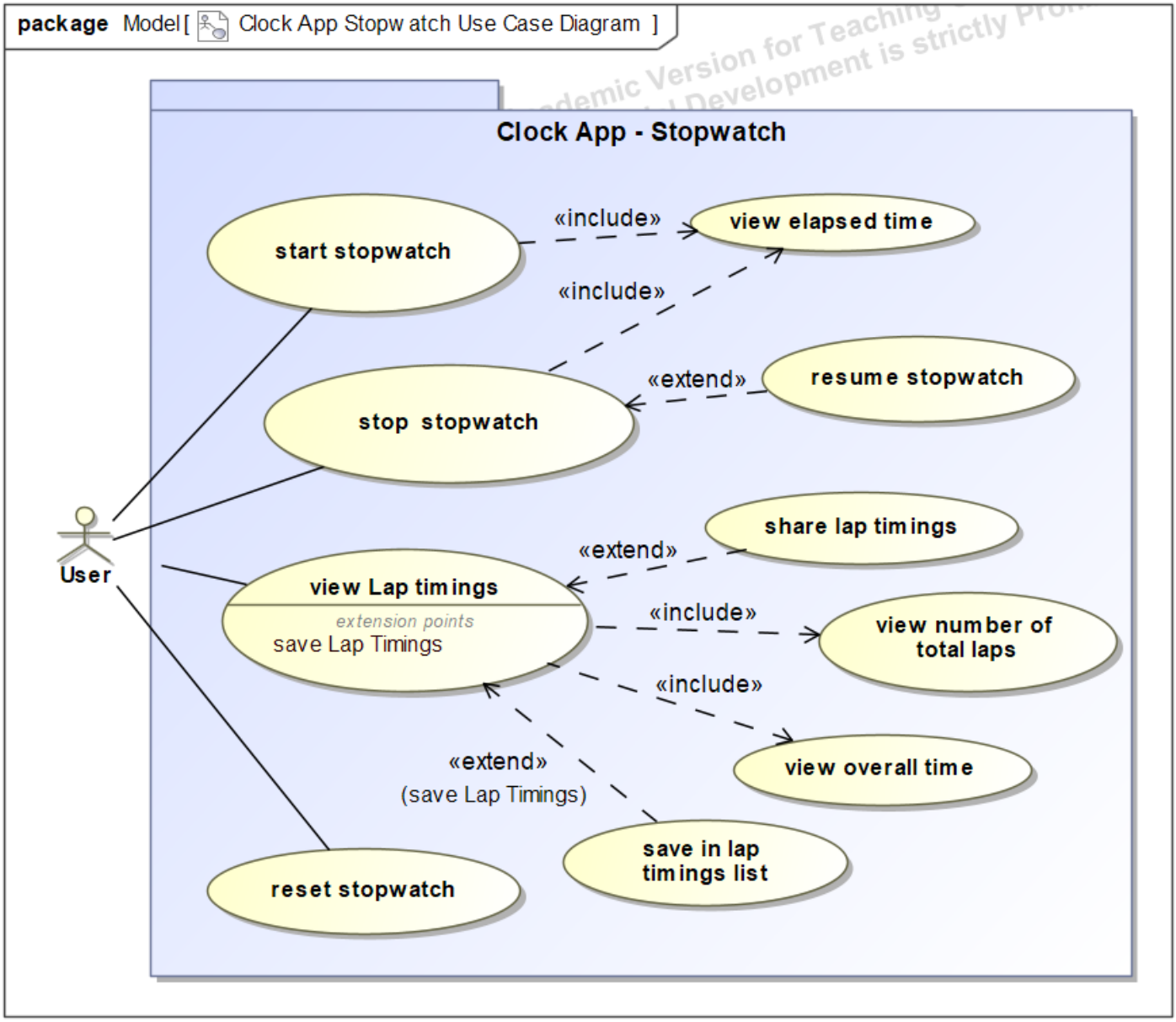
6.3.4 Use Case Diagram for Alarm



The User can do multiple things in our Alarm system in smartwatch like setting Alarm time, date, snooze and vibration also the User can view the Alarm list and enable or disable alarm and more. If the User decides to set Snooze on the Alarm, he/she has to also set the Duration of snooze and also set how many times it should snooze, that's why they have include relationship.

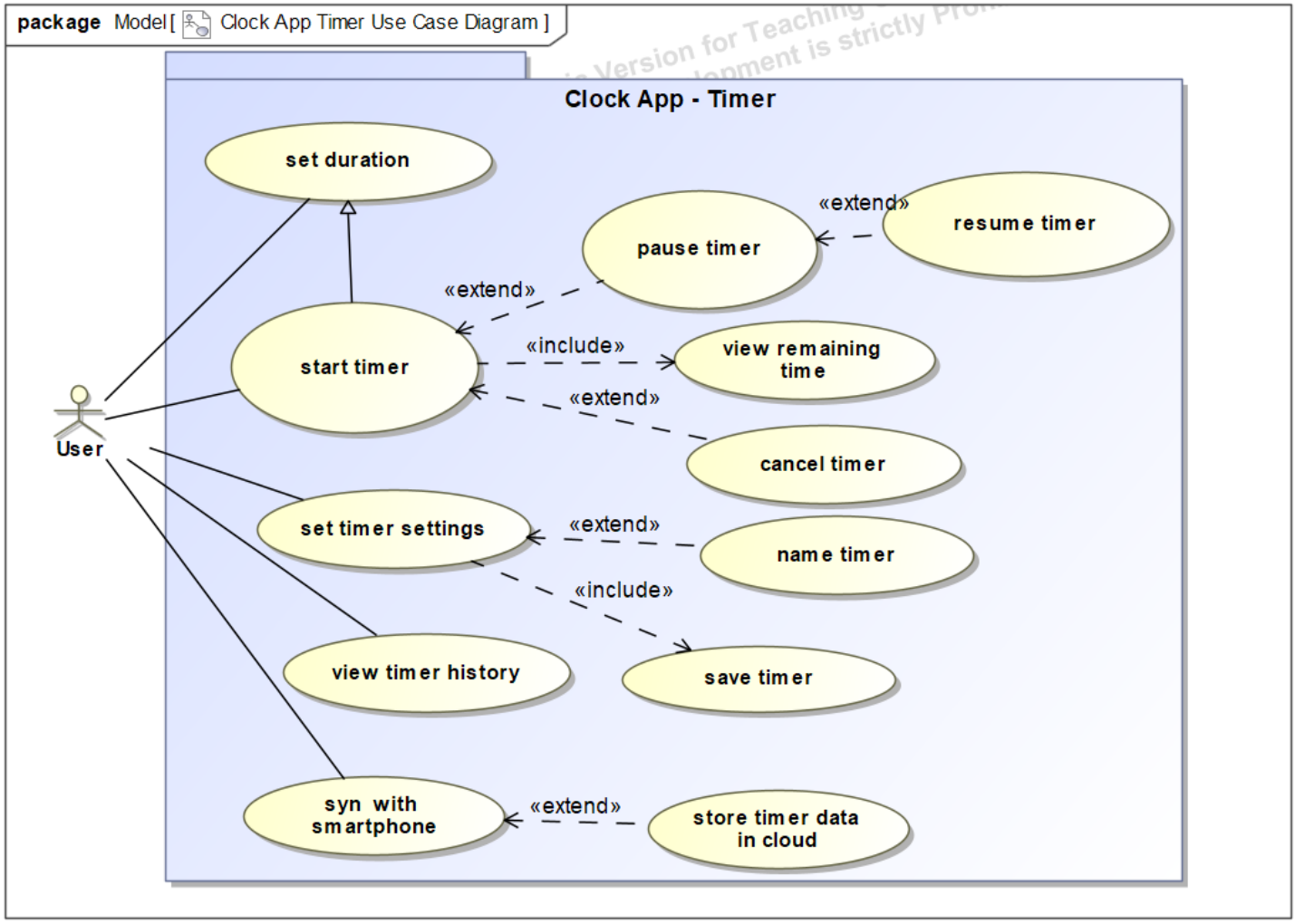
These are the main Use Cases that the User needs to use Alram.

6.3.5 Use Case Diagram for Stopwatch



The User can start, stop and reset the Stopwatch. T the Stopwatch shows overall time and number of total laps everytime the User views his/her Lap Timings, thats why i used include relationship. Also the User has the option to further share the lap timings and also save them in the lap timing list.

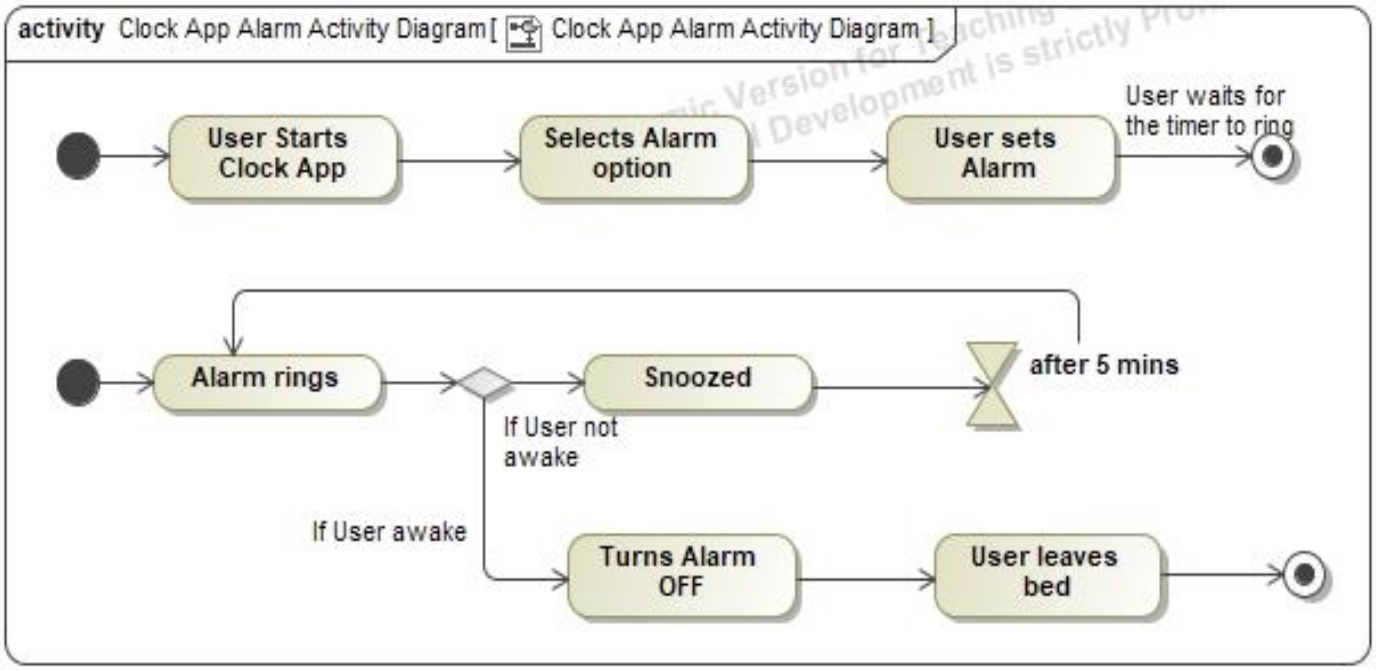
6.3.6 Use Case Diagram for Timer



The User can start timer and further pause and cancel the timer, the Timer always shows the remaining time to the User. Not only this, but the User also can set Timer duration, Timer Settings and more. A new feature that is pretty cool is that, the Timer allows the User the sync the timer Data with his/her Smartphone also helping in storing data in cloud.

6.3.7 Activity Diagram for Tracking Time

6.3.7.1 Alarm

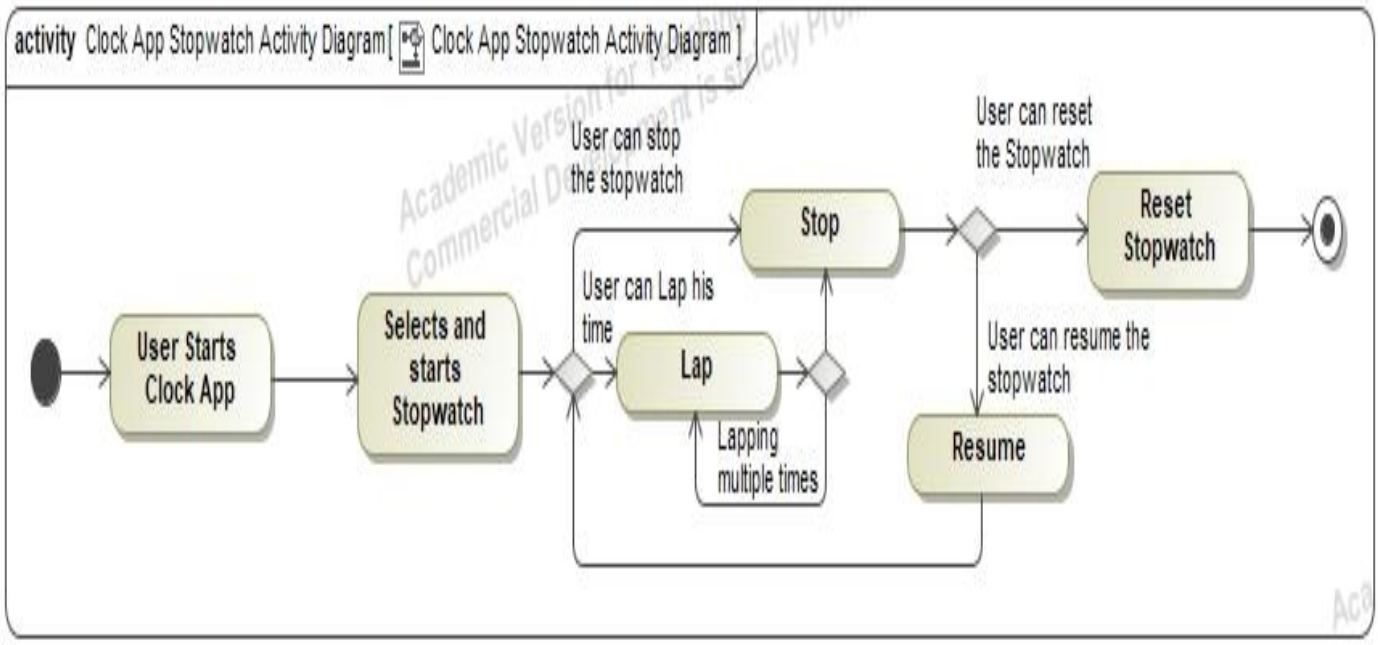


This activity diagram provides a visual representation of the steps involved in setting and managing alarms in the Clock App, guiding users through the process from configuration to alarm triggering and response.

In the first Activity Diagram, it basically shows how the User can set the Alarm and then waits for the Alarm to ring.

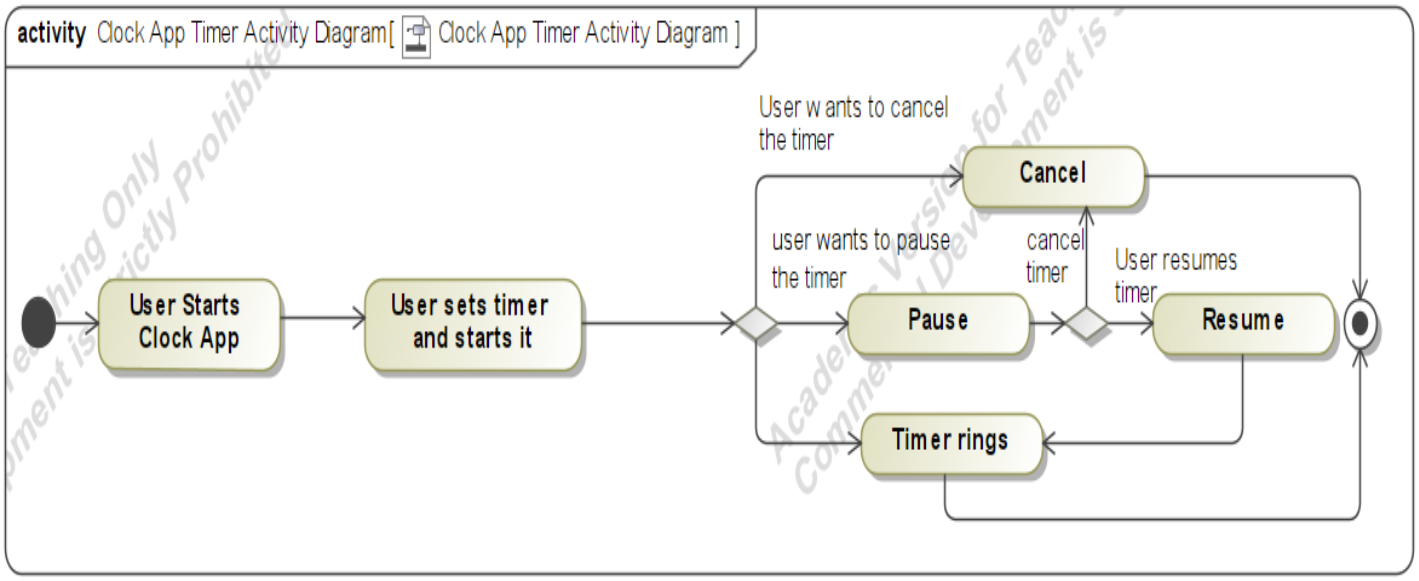
In the second Activity Diagram, it shows the Activity what happens after the Alarm rings. Lets say the User has set the snooze time to be 5 mins. After the Alarm rings, if the User is awake the User has the option to turn the Alarm OFF or if the User is still not awake User has a option to snooze it. As the User set the snooze time of 5 mins, after 5 mins the Alarm will ring again and this goes on in loop, The loop ends when the User turns OFF the Alarm.

6.3.7.2 Stopwatch



The above Activity Diagram is for Stopwatch. After the User starts the clock app he/she starts the Stopwatch. After starting the Stopwatch, the User can either stop the stopwatch or lap his/ her time. The User can then Lap the Stopwatch as many times as the User wishes and this could go on in loop. The loop ends when the User stops the Stopwatch further the User can also resume the Stopwatch. But at last the User has to Stop the Stopwatch, in order to get the timings. If the User wishes the start the Stopwatch, User has to just reset the Stopwatch.

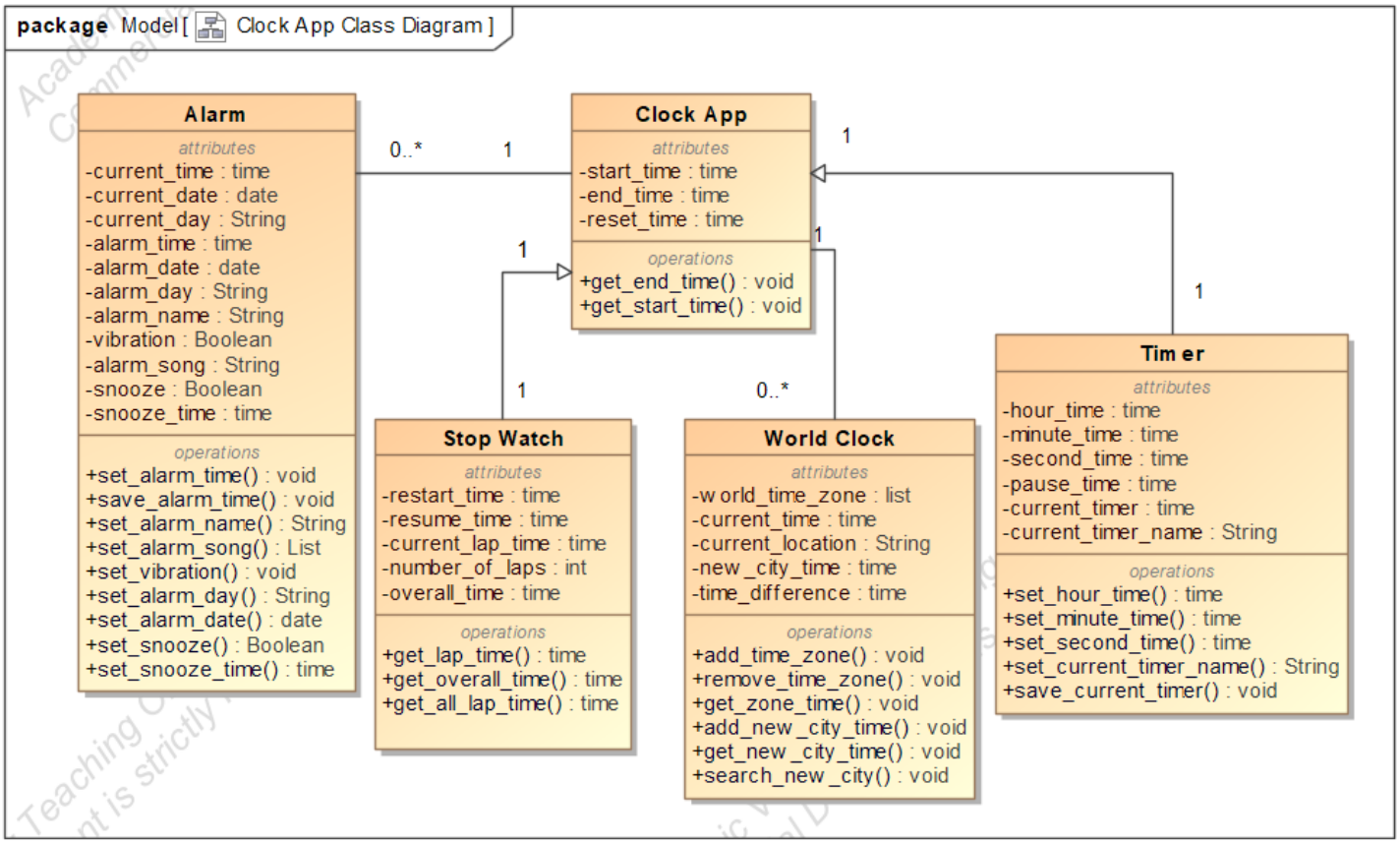
6.3.7.3 Timer



This is a Activity Diagram for Timer. After the User starts the Clock App , User then first sets the timer and then starts it., Then 3 things could happen.

1. The Timer rings and the User can again set a new Timer
2. The User decides to pause the Timer. The User then has 2 options, either to cancel the timer or resume it and let the Timer to ring.
3. The User also can directly cancel the Timer without resuming the Timer.

6.3.8 Class Diagram for Tracking Time



This is a Class Diagram for Tracking Time. There are 5 main Classes (Clock App, Alarm, Stop-watch, World Clock and Timer)

- All the classes are connected to Clock App Class, because the User could use all the Time tracking applications only via the Clock App.
- The Alarm Class and World Clock Class are connected to Clock App via association relation, wherein the User can have 0 or more alarms and World clock time zone that's why the cardinality of these classes is 0..*
- The Stopwatch Class and Timer Class are connected with Clock App with generalization (inheritance), where the Stopwatch inherits methods and attributes like – end_time and get_end_time , also same with the Timer Class, where it inherits attributes and methods like – start_time, end_time , get_end_time.

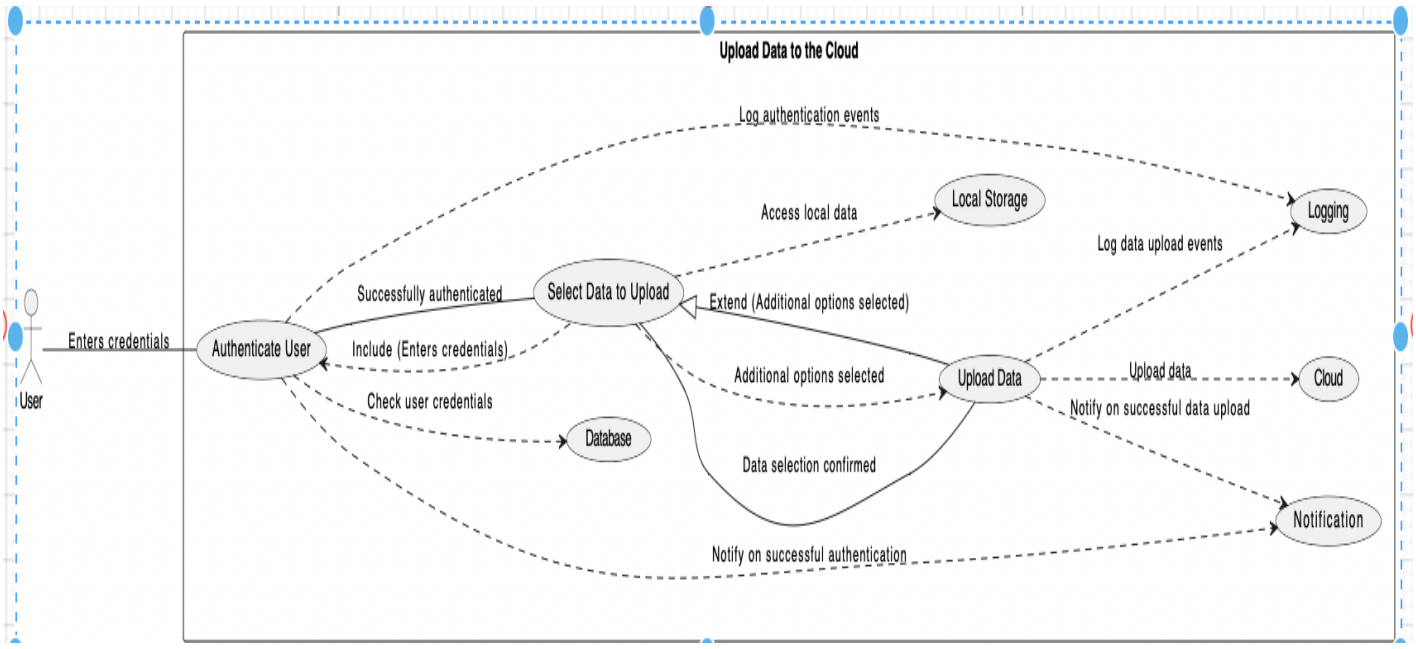
No sequence Diagram was made, because these Classes have no sequence of occurrence.

Zhiyu Li

7. Upload data to the cloud

Through the close integration with the cloud, the smart watch realizes automatic data backup and synchronization, and provides users with a comprehensive, safe and convenient personal information management solution. From the real-time feedback of health monitoring data, to the cloud synchronization of photo albums and information, to the backup of mobile phone information, this combination not only ensures the security of data, but also brings users an unprecedented convenient experience. Through advanced encryption techniques and regular system updates, the combination of smartwatches and the cloud provides efficient data transfer and seamless synchronization across devices while protecting user privacy. With the continuous progress of technology, we expect that the interactive functions between smart watches and the cloud will be richer and more perfect, bringing more convenience and value to users.

7.1 Use case



Name	Upload Data to Cloud Storage
ID	UD-0001
Description	The contractor logs in to the system. This is a necessary step before the contractor can use any function of the system
Trigger	User initiates the upload process.
Actors	Contractor (Cloud Storage System) , Customer (User)
Pre-conditions	User has a stable network connection;User is authenticated and authorized to access the cloud storage.
Post-conditions	Data is successfully uploaded to the cloud storage;User receives confirmation of successful upload.
Basic Flow	
Description	If the upload process encounters an error or fails.
Actions	
1	User selects files for upload.
2	User logs into the cloud storage system.
3	System securely uploads the selected data to the cloud.
	System confirms successful upload to the user.
Alternative Flow	A (Upload Failure)
Description	The contractor does not have an account yet (email is unknown)
Actions	
1	System displays an error message to the user.
2	User can choose to retry the upload or cancel the process.
Alternative Flow	B(Authentication Failure)
Description	If the user's credentials are incorrect or not authorized.
Actions	
1	System prompts the user to re-enter credentials.
2	If unsuccessful, system denies access and informs the user of the issue.
3	User is given the option to resolve the authentication problem or contact support.

The use case aims to implement a process whereby users securely upload data to the cloud storage system through the application. During this process, users have the ability to select and transfer files to the cloud storage system while ensuring secure authentication. The primary objective is to provide authorized users with a reliable, user-friendly, and secure method for managing their data. Preconditions include a stable network connection, successful authentication, and proper authorization for accessing the cloud storage system. Upon successful completion of the upload, users will receive confirmation from the system. The primary objective is to provide users with a user-friendly interface for securely managing their data.

Now, with cloud storage, you can view, back up, and even share your precious data anytime and anywhere without worrying about losing or forgetting it.

And the benefits of cloud storage go far beyond that. It's highly extensible and flexible, so no matter how much storage you need, you can get it. More importantly, through cloud storage, we can easily synchronize and share data, which greatly improves the efficiency of team collaboration.

Not only that, the combination of smart watches and the cloud has also brought us an unprecedented convenient experience. For example, my watch can interface seamlessly with a health APP and automatically upload exercise data to the cloud. This way, I can not only check my sports data at any time, but also clearly see my progress and growth.

Of course, uploads to the cloud require some attention to detail. First of all, it's important to choose a reliable service provider to ensure that your data is safe. In addition, when uploading and downloading data, it is better to use a stable Wi-Fi environment to avoid unnecessary trouble.

7.2 Snow card

#<4>: Upload data to the cloud

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Requirement Type: Functional
For Whom? Contractor, Customer
User Satisfaction: High
User Dissatisfaction: Medium

Description:

As a user, I want the capability to seamlessly upload my data to a cloud-based storage system. This functionality is vital for ensuring data accessibility across multiple devices and serves as a secure backup solution. The upload process should guarantee data integrity and security while facilitating ease of access and retrieval from any location. High user satisfaction will be achieved by providing a reliable and efficient cloud upload feature, thereby ensuring data safety and accessibility.

Software Engineering Analysis

Key aspects that must be considered when developing a cloud storage system are highlighted. The interactivity analysis highlights the importance of user interface design and the need to ensure that users can easily upload data through an intuitive and easy-to-use interface. The input/output analysis details how the user selects and uploads data and how the system ensures the accessibility and safe backup of the data. The analysis of the interface design emphasizes the friendly and intuitive design principles, and fully takes into account the different user operation habits. The testing and verification part involves specific testing strategies, including unit testing, performance testing and security testing, to ensure that the system functions normally, stably and safely.

When developing a cloud storage system, a holistic approach encompassing various stages of the software engineering process is essential. Beginning with requirement analysis, understanding user and stakeholder needs regarding storage capacity, accessibility, data security, scalability, and performance is crucial. Subsequently, system architecture design should focus on determining the overall structure, including storage mechanisms, data distribution strategies, redundancy measures, and scalability options, while selecting appropriate technologies and frameworks for implementation.

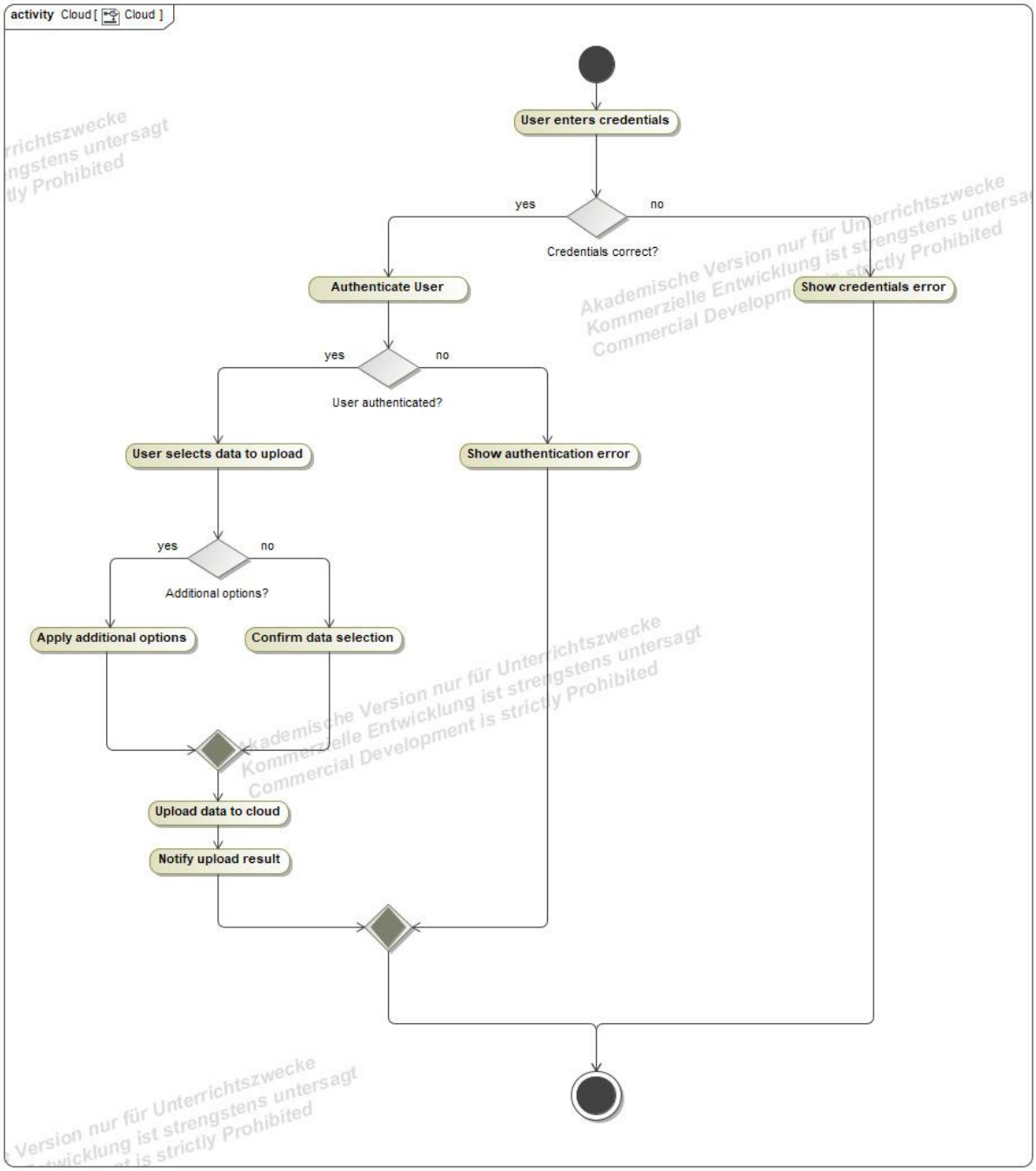
User interface design plays a pivotal role in ensuring a positive user experience. The interface should be intuitive, responsive, and visually appealing, with features such as drag-and-drop functionality, file preview options, and customizable settings. An efficient and user-friendly data upload mechanism should be developed, incorporating interfaces for file selection, drag-and-drop functionality, progress indicators, and error handling to ensure smooth data transfer.

Data accessibility and safe backup are paramount, requiring robust storage and retrieval mechanisms, data encryption, replication across multiple servers or data centers, and regular backups to prevent data loss. Performance optimization involves rigorous testing under various load conditions to identify and address bottlenecks or performance issues.

Security measures are critical, necessitating the implementation of encryption protocols, access control mechanisms, authentication, and intrusion detection systems to safeguard user data. Scalability and resource management should be addressed by designing systems to scale seamlessly and implementing dynamic resource allocation and load balancing strategies.

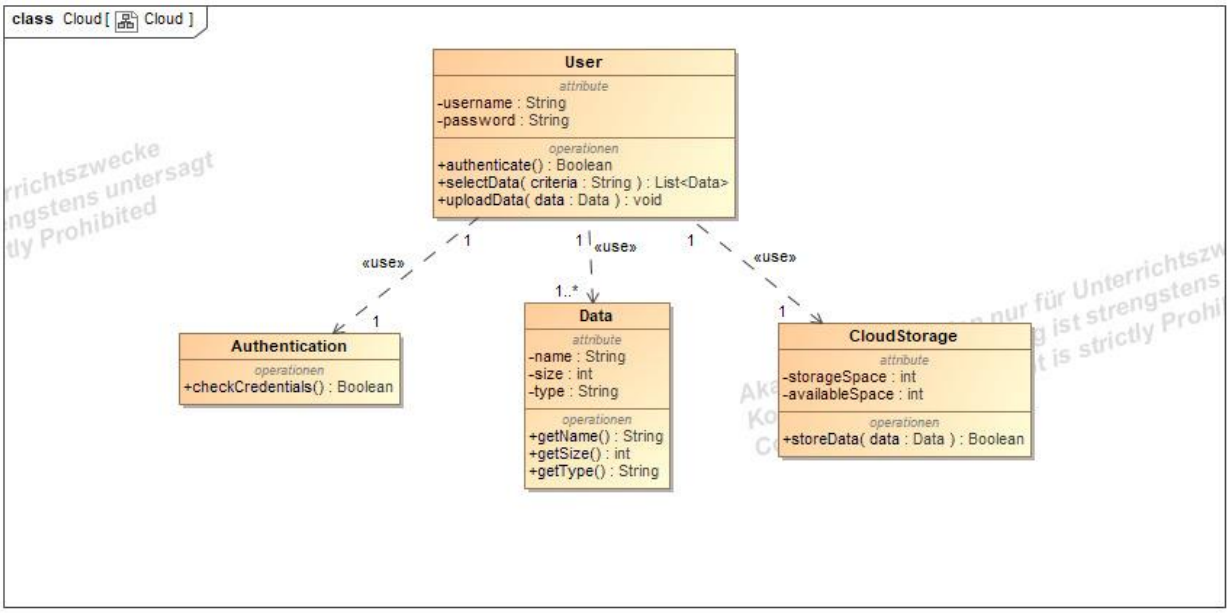
Comprehensive testing and verification are essential to validate functionality, performance, and security. This includes unit testing, integration testing, system testing, performance testing, and security testing. Additionally, soliciting user feedback for iterative improvement ensures that the cloud storage system remains aligned with user needs and preferences over time.

7.3 Activity diagram



This Activity diagram depicts the core process of uploading user data to the cloud, highlighting the key steps such as authentication, data selection and revalidation, and data uploading. To ensure the integrity, confidentiality and availability of data, strict authentication mechanism, data validation rules and encryption technology are needed. In addition, the optimization of user feedback and experience is equally important to enhance the usability of the system and user satisfaction.

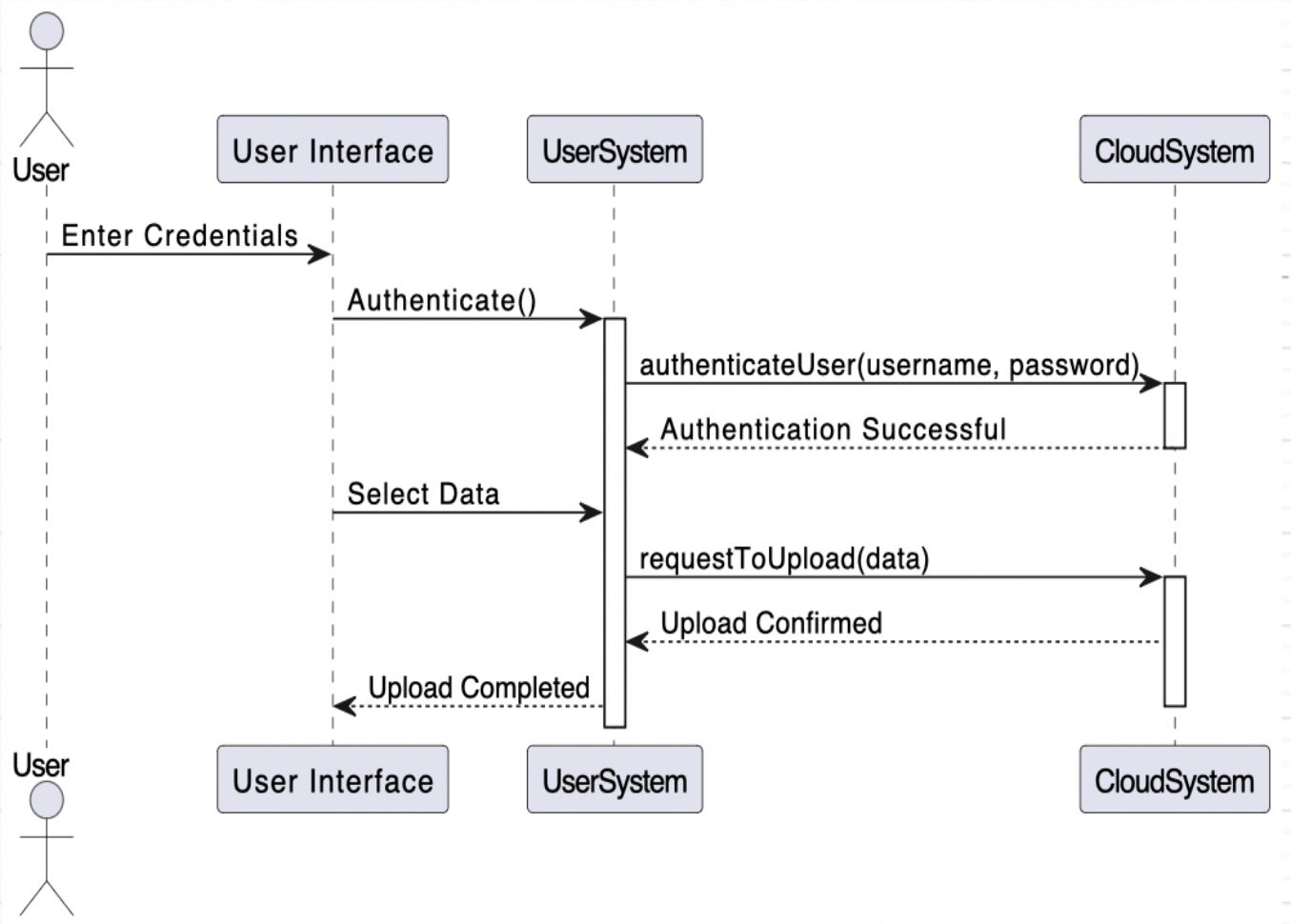
7.4 Class diagram



In the class diagram of our cloud data upload system, the key classes include user interface class, data upload class, cloud storage class, and test and verification class. The user interface class is the core of the interaction between the system and the user, providing an intuitive and easy to use upload interface. The data upload class is responsible for handling the data selected by the user and ensuring safe and correct uploading to the cloud storage. The cloud storage class manages the storage and access of data in the cloud, and works with the data upload class to achieve secure data storage and access. The testing and verification class is responsible for ensuring that the system functions properly, performs well, and is safe and reliable, verifying the integrity and security of the uploaded data through various testing strategies. These key classes work synergistically to achieve a user-friendly upload interface, secure data upload and storage, and guarantee of high quality and reliability of the system.

In the class diagram of our cloud data upload system, key classes include the user interface class, data upload class, cloud storage class, and test and verification class. The user interface class serves as the core of user-system interaction, offering an intuitive upload interface. The data upload class manages selected data, ensuring safe upload to cloud storage. The cloud storage class oversees data storage and access, collaborating with the data upload class for secure operations. The test and verification class validates system functionality, performance, and security, ensuring data integrity. These classes work in synergy to deliver a user-friendly interface, secure data handling, and reliable system performance.

7.5 Sequence diagram



7.6. Smart Watch - upload data to the cloud

Health detection information is automatically synchronized to the cloud

7.6.1 Overall UI design

a. Real-time monitoring and feedback: Smartwatches continuously collect user health data through built-in health detection functions such as heart rate monitoring, step counting, and sleep quality analysis.

b. Automatic upload and storage: The data is not only displayed on the watch, but also automatically synced to the user's cloud account.
c. Anytime and anywhere: users can view, track and analyze their health data at any time on the mobile APP, so as to better understand their physical condition and sports performance.
d. Health analysis and advice: Based on the data from the cloud, the APP can provide personalized health analysis and advice to help users improve their living habits and adjust exercise plans.

With real-time monitoring and feedback, smartwatches play a crucial role in tracking users' health data.

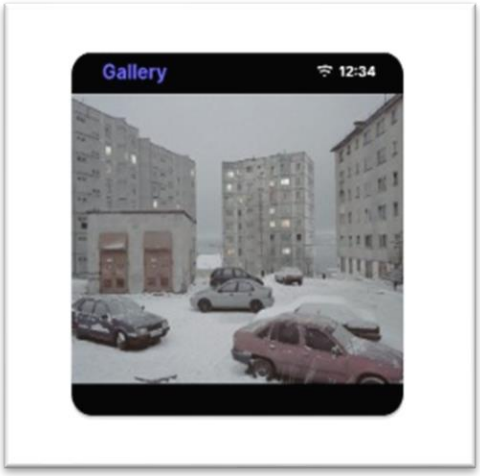
These advanced devices utilize built-in functions like heart rate monitoring, step counting, and sleep quality analysis to constantly collect information on the wearer's well-being.

Not only does this data appear on the watch's display, but it is also automatically uploaded and stored in the user's cloud account.

This allows for convenient access to the information anytime, anywhere, through a mobile application. The mobile app not only enables users to view and track their health data in detail but also provides personalized health analysis and advice based on the collected information.

This feedback loop offers valuable insights into lifestyle habits and exercise patterns, ultimately aiding individuals in improving their well-being through meaningful adjustments to their daily routine.

7.6.2 The watch syncs with the cloud of your phone's photo album



a. Quick access and sharing: When users want to view photos or videos on the watch, they can easily sync photos and videos from their phone albums to the watch through the cloud link.
b. Convenience: This feature provides additional convenience to the user, who does not need to manually transfer photos between the watch and the phone, but only needs a single synchronization operation to complete.
c. Secure transmission: During the synchronization process, data should be protected using advanced encryption techniques to ensure user privacy and data security.

Our system prioritizes user convenience and security through a multifaceted approach. Quick access and sharing are facilitated by effortless synchronization of photos and videos from phone albums to the watch via the cloud link. This eliminates manual transfers, offering users unparalleled convenience with a single synchronization operation.

Moreover, our system goes beyond mere convenience by prioritizing data security. During the synchronization process, advanced encryption techniques are implemented to ensure the utmost privacy and security of user data. This commitment to secure transmission underscores our dedication to maintaining user trust and confidence.

7.6.3 Mobile phone information cloud backup and watch read

Message Center: Through the cloud connection, the watch can display various notifications and messages from the phone, such as text messages, social media notifications, emails, etc.

View and reply: Users can view and reply to these messages directly on the watch, providing a convenient mobile experience.

Information protection: In consideration of privacy and security, the watch only displays a summary or tag of the message and does not show the full message content. The user needs to use the mobile phone to view or reply to the complete message.

Cloud backup: Backing up phone information to the cloud means that even if the phone is lost or damaged, users can recover their information from the cloud, ensuring data security.

7.6.4 Security Considerations

Encrypted transmission: All data transmitted through the cloud should be protected with advanced encryption technology to ensure the security of data during transmission.

Access control: Users should be able to control what data can be uploaded to the cloud and can set different levels of privacy and access rights.

Regular updates and patches: In order to deal with potential security threats, watches and cloud services should be regularly updated and patched to ensure the security of the system.

Multifactor authentication: For advanced user accounts, consider introducing multifactor authentication to increase the security of the account.

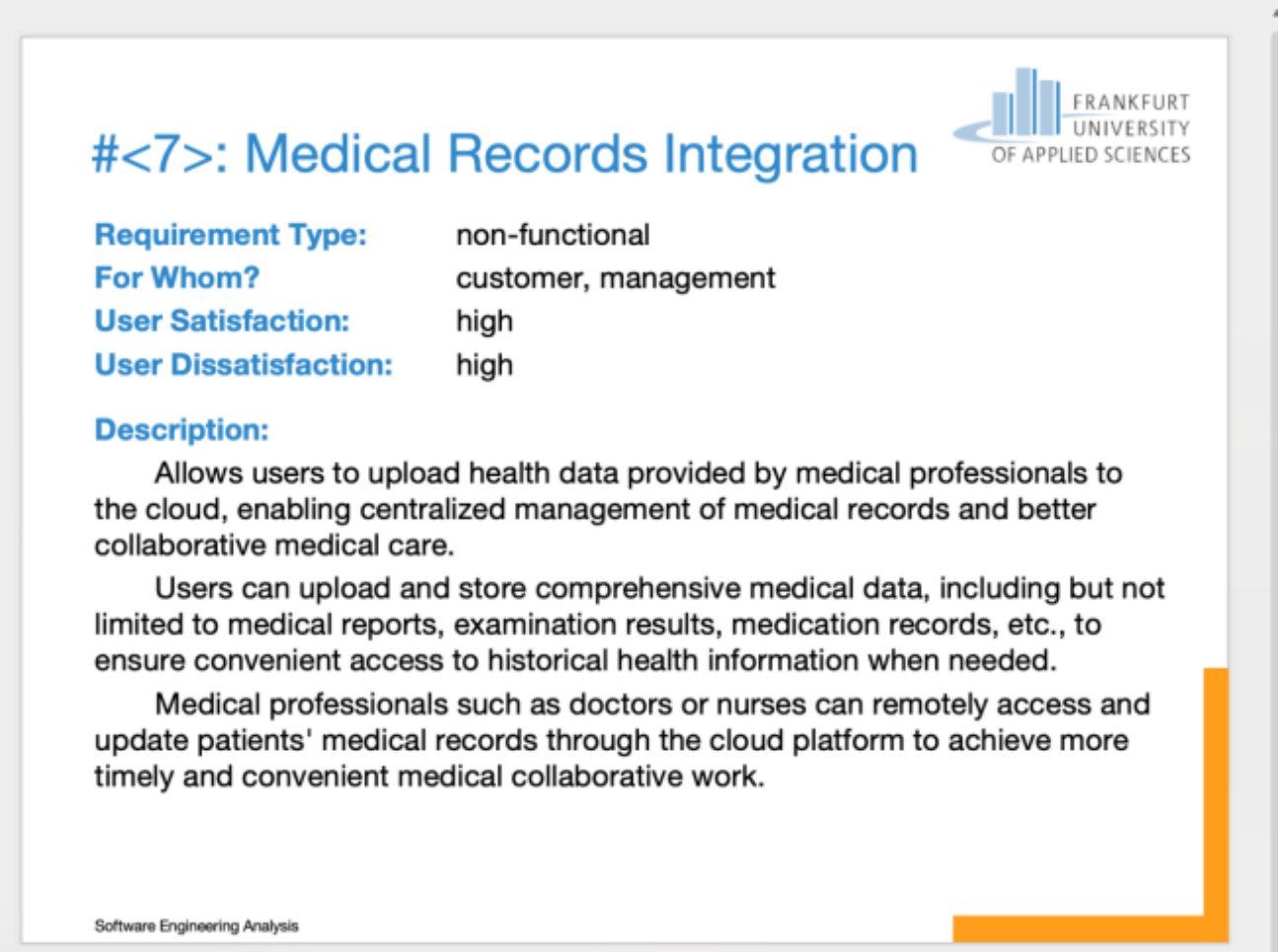
Security considerations are paramount in the development of any software system, and our cloud-based watch service is no exception. In 7.6.4, we outline several key measures to ensure the integrity and confidentiality of user data, as well as the overall security of the system.

Encrypted transmission serves as the foundation of our security strategy. By employing advanced encryption technology, all data transmitted through the cloud is shielded from unauthorized access or interception, guaranteeing the security of user data during transmission.

By integrating these security considerations into our development process, we prioritize the protection of user data and the overall security of our cloud-based watch service. Through a combination of encryption, access control, regular updates, and multifactor authentication, we strive to create a secure and trustworthy platform that users can rely on with confidence.

7.7. Medical Records Integration

7.7.1. Snow card



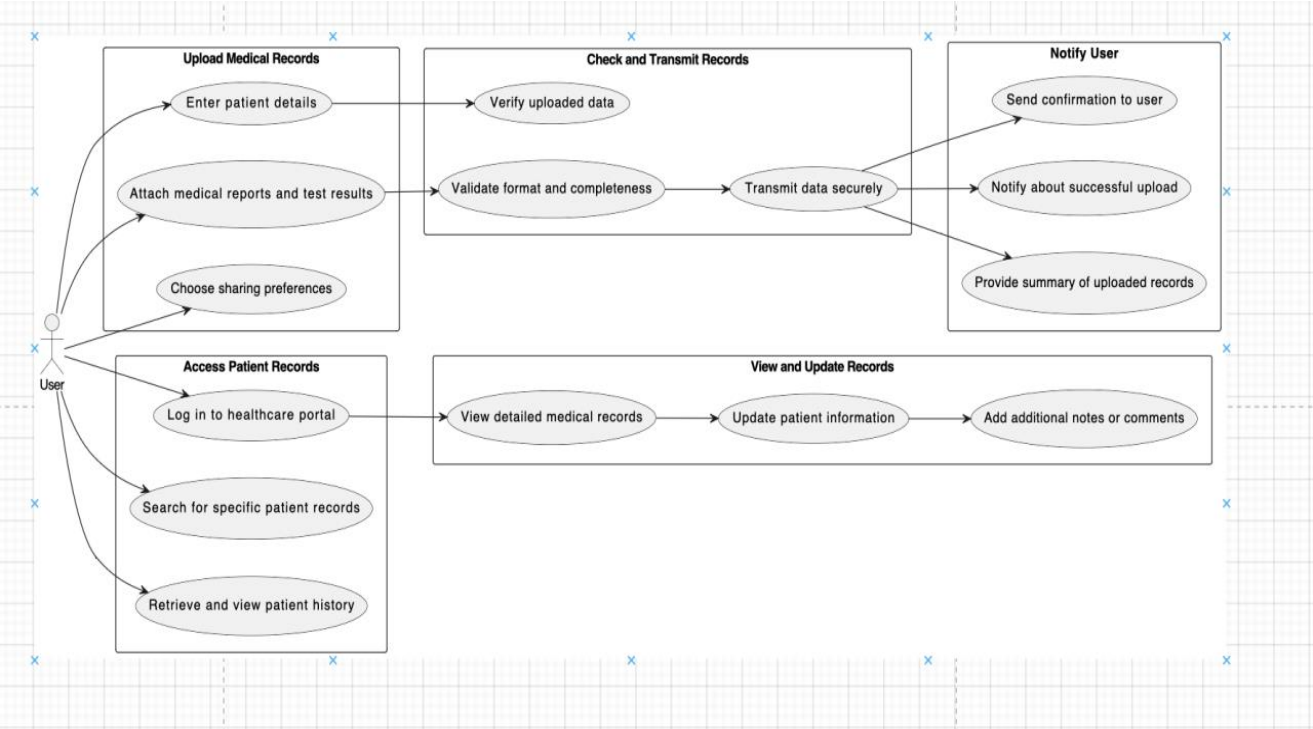
The cloud platform provides highly secure data transmission and storage mechanisms and uses encryption technology to ensure the privacy security of medical data uploaded by users in the process of transmission and storage.

Users have flexible access control, which can set the sharing permissions of medical records, ensure that only authorized medical professionals can view specific medical information, and protect user privacy.

The cloud platform implements robust security measures to safeguard the confidentiality and integrity of medical data during transmission and storage. Through advanced encryption technology, all medical data uploaded by users undergoes secure transmission and storage, ensuring the privacy and security of sensitive medical information.

Moreover, users are afforded flexible access control capabilities, allowing them to define and manage sharing permissions for their medical records. This granular control ensures that only authorized medical professionals can access specific medical information, thereby preserving user privacy and maintaining strict confidentiality. By empowering users with these access control features, the cloud platform enhances user privacy and security, while facilitating seamless collaboration and communication among healthcare professionals.

7.7.2. Use case


Upload Medical Records
Enter patient details
Attach medical reports and test results
Choose sharing preferences
Check and Transmit Records
Verify uploaded data
Validate format and completeness
Transmit data securely
Notify User
Send confirmation to user
Notify about successful upload
Provide summary of uploaded records
User
Access Patient Records
Log in to healthcare portal
Search for specific patient records
Retrieve and view patient history
View and Update Records
View detailed medical records
Update patient information
Add additional notes or comments

New feature of our smartwatch - Medical Records Integration. With this functionality, we aim to enhance the user's health management experience and provide healthcare professionals with a more convenient data access pathway.

Users can effortlessly complete the medical records upload in a few simple steps. First, users input detailed patient information using the smartwatch and attach medical reports and test results. To ensure users have full control, we provide options to choose their data sharing preferences.

Subsequently, our system conducts data verification and transmission. The system ensures the uploaded data's correct format and completeness, securely transmitting it to our cloud storage system. Not only do users receive a confirmation message post-upload, but they can also view a summary of their uploaded records anytime on the smartwatch, ensuring everything is well-organized.

Users can log in to the healthcare portal, search for specific patient records, and view the complete health history. This feature not only empowers users to manage their health effectively but also provides healthcare professionals with an efficient data management platform.

On the healthcare portal, users can scrutinize detailed medical records, including test results and diagnostic information. More importantly, users can update personal information and even add additional notes or comments, creating a more comprehensive and detailed health profile.

We delved into the intricacies of our smartwatch application's use case, specifically focusing on the secure and user-friendly process of uploading data to the cloud storage system. This feature is designed with the utmost priority on providing authorized users with a reliable method for managing their data, ensuring secure authentication and a seamless experience.

As we ventured into the realm of cloud storage benefits, we explored its extensibility and flexibility, highlighting the seamless synchronization and sharing of data that significantly amplifies team collaboration efficiency. The integration of smartwatches with the cloud introduces an unprecedented level of convenience, exemplified by the automatic upload of exercise data. This synergy allows users not only to track their sports data but also witness their progress and growth firsthand.

Taking a step further, we introduced a groundbreaking feature - Medical Records Integration. This functionality aims to revolutionize the user's health management experience. Users can effortlessly upload detailed medical records, ensuring data accuracy and completeness. The healthcare portal provides a comprehensive platform for users and professionals alike, allowing seamless data access, updates to personal information, and detailed health profile management.

In conclusion, our smartwatch application doesn't just offer data uploads to the cloud; it's a holistic approach to health management and collaborative data sharing. We're not only ensuring the safety and accessibility of your data but also paving the way for a future where health information is seamlessly integrated and managed.

Through these functions and considerations, the integration of smartwatches with the cloud provides users with a comprehensive, secure and convenient personal information management solution, which further enhances the practicality and user experience of smart devices.