

# **SWEA Final Report**

Software Engineering – Analysis  
WiSe 2023/24

**Dozent: Peter Thoma**  
**Tutor: Jan Wagner**

**Topic: Smart watch project**  
**„Posture Piece“**

**Group 2 – Team 4**

Changmin Lee (1516177)  
Eric Jacob (1409493)  
Jeongmin Ahn (1522888)  
Yoonsun Lee (1519516)  
Soojin Jung (1524022)

# Table of Content

<b>1.</b>	<b>ATTENDANCE.....</b>	<b>7</b>
<b>2.</b>	<b>MEETING POINT &amp; PROTOCOL.....</b>	<b>8</b>
2.1.	MEETING PROTOCOL.....	8
2.2.	SCRUM .....	9
<b>3.</b>	<b>DECLARATION OR AUTHORSHIP .....</b>	<b>10</b>
<b>4.</b>	<b>INTRODUCTION.....</b>	<b>11</b>
<b>5.</b>	<b>OVERVIEW .....</b>	<b>12</b>
<b>6.</b>	<b>PRODUCT BACKLOG.....</b>	<b>13</b>
<b>7.</b>	<b>CUSTOMER REGISTRATION, LOGIN .....</b>	<b>15</b>
7.1.	REGISTRATION .....	16
7.1.1.	<i>Basic information.....</i>	17
7.1.2.	<i>Body Information.....</i>	19
7.1.3.	<i>Account Information.....</i>	22
7.2.	LOGIN .....	25
7.3.	ENTER ACCOUNT .....	27
7.4.	DELETE ACCOUNT .....	28
<b>8.</b>	<b>DASHBOARD.....</b>	<b>29</b>
8.1.	CUSTOMIZE DASHBOARD.....	29
8.2.	EDIT DASHBOARD.....	32
8.3.	CONTRACTOR DASHBOARD UPDATES.....	35
8.4.	HEARTRATE GRAPH WIDGET .....	37
8.5.	STEP COUNT WIDGET .....	41
<b>9.</b>	<b>HEART RATE .....</b>	<b>46</b>
9.1.	CHECK HEART RATE.....	46
9.2.	PERSONALIZED HEART RATE RANGE.....	48
9.3.	VISUALIZE HEART RATE.....	50
9.4.	HEART RATE REAL TIME ALERT .....	52
9.5.	SAVE THE HEART RATE DATA.....	55
<b>10.</b>	<b>STEPS .....</b>	<b>57</b>
10.1.	WORKOUT DETECTION .....	57
10.2.	STROLL DETECTION.....	60
10.3.	FALL DETECTION.....	62
10.4.	SAVE THE DAILY USER'S WALKING STEPS DATA.....	63
10.5.	SET THE STANDARD OF AVERAGE OF WEEKLY STEPS.....	65
10.6.	FINAL WEEKLY SUMMARY OF STEPS.....	68
<b>11.</b>	<b>TURTLENECK MODE .....</b>	<b>69</b>
11.1.	TURTLENECK MODE BUTTON ACTIVATION .....	69
11.2.	START TURTLENECK MODE.....	71
11.3.	TURTLENECK DETECTION.....	73
11.4.	VISUALIZE REAL-TIME NECK ANGLE .....	75
11.5.	TURTLENECK WARNING NOTIFICATION .....	76
11.6.	ACTIVITY DIAGRAM .....	77
11.7.	SEQUENCE DIAGRAM.....	78
<b>12.</b>	<b>FEEDBACK FROM EXPERT .....</b>	<b>79</b>

12.1. CALCULATE THE SCORE .....	79
12.2. MAKE THE WEEKLY REPORT FROM EXPERT .....	81
12.3. VISUALIZE THE FEEDBACK REPORT .....	83
12.4. SEND THE ALARM TO USER.....	85
12.5. REVIEW FOR THE FEEDBACK .....	87
<b>13. OTHER USEFUL FUNCTIONS .....</b>	<b>91</b>
13.1. SLEEP ANALYSIS .....	91
13.2. PROLONGED SITTING (SEDENTARY) DETECTION .....	95
<b>14. PAYMENT &amp; SUBSCRIPTION .....</b>	<b>97</b>
14.1. SUBSCRIPTION.....	97
14.2. PRICING AND PROMOTIONAL STRATEGY.....	99
14.3. PAYMENT INTEGRATION .....	100
14.4. CUSTOMER SUPPORT FOR PRO USERS .....	102
<b>15. USABILITY.....</b>	<b>104</b>
15.1. APP LANGUAGE.....	104
15.2. NOTIFICATION SETTINGS .....	105
15.3. THEME SETTINGS .....	107
15.4. NETWORK USAGE SETTINGS .....	108
15.5. HELP & INFORMATION .....	109
15.6. ABNORMAL SHUTDOWN .....	111
15.7. FONT SIZE SETTINGS.....	112
<b>16. REAL-TIME DATA COLLECTION AND LOGGING .....</b>	<b>114</b>
16.1. ACTIVATE SENSORS .....	114
16.2. DEFINITION DATA TYPE FOR EACH SENSOR & COLLECTING DATA.....	117
16.3. PAUSE OR FINISH COLLECTING DATA.....	121
<b>17. COMPATIBILITY .....</b>	<b>123</b>
17.1. DATA SYNCHRONIZATION, BACKUP.....	123
17.2. AUTHENTICATION AND SECURITY.....	125
17.3. USE MBAAS (MOBILE BACKEND AS A SERVICE).....	127
17.4. COST AND SCALABILITY CONSIDERATIONS .....	129
17.5. USER EXPERIENCE CONSISTENCY .....	130
17.6. BATTERY MANAGING .....	132
17.7. SEQUENCE DIAGRAM .....	133
<b>18. SECURITY .....</b>	<b>135</b>
18.1. ACCESS CONTROL .....	135
18.2. ENCRYPTION .....	137
<b>19. UML CLASS DIAGRAM.....</b>	<b>138</b>
<b>20. UI/UX PROTOTYPE .....</b>	<b>139</b>
<b>21. SUMMARY .....</b>	<b>141</b>

Table 1 Attendance list .....	7
Table 2 Meeting protocol.....	9
Table 3 Backlog .....	13
Table 4 Backlog_members .....	14

Image 1 Overview .....	12
Image 2 UI/UX Start checking heart rate .....	47

Image 3 UI/UX Visualize heart rate .....	51
Image 4 UI/UX Heart rate real time alert .....	54
Image 5 Steps UIUX prototype.....	68
Image 6 Plan for real-time reflection screen.....	75
Image 7 Turtleneck warning notification.....	76
Image 8 UI/UX Visualize the feedback report .....	84
Image 9 UI/UX Visualize the feedback report .....	85
Image 10 UI/UX Review for feedback .....	90
Image 11 Activate sensors .....	115
Image 12 Definition data type for each sensor .....	118
Image 13 Pause of finish collecting data .....	122
Image 14 App Icon prototype .....	139
Image 15 UI/UX prototype .....	139
Image 16 UI/UX Payment (1).....	139
Image 17 UI/UX Payment (2).....	139
Image 18 UI/UX Feedback(1) .....	140
Image 19 UI/UX Feedback(2) .....	140
Image 20 UI/UX Feedback(3) .....	140
Image 21 UI/UX Entire flow .....	141
 Description of Usecase 1 Basic Information .....	19
Description of Usecase 2 Body information .....	22
Description of Usecase 3 Account of registration .....	24
Description of Usecase 4 Login .....	26
Description of Usecase 5 Customize dashboard.....	30
Description of Usecase 6 Edit dashboard .....	33
Description of Usecase 7 Contractor updates .....	36
Description of Usecase 8 Heart rate graph .....	38
Description of Usecase 9 Step count graph .....	42
Description of Usecase 10 Start checking heart rate .....	48
Description of Usecase 11 Personalised heart rate range .....	50
Description of Usecase 12 Heart rate real time alert .....	53
Description of Usecase 13 Workout detection.....	58
Description of Usecase 14 Turtleneck mode button activation .....	70
Description of Usecase 15 Start turtleneck mode .....	72
Description of Usecase 16 Turtleneck detection .....	74
Description of Usecase 17 Show the total score of weekly health .....	81
Description of Usecase 18 Sleep analysis.....	92
Description of Usecase 19 Payment integeration .....	102
Description of Usecase 20 Data synchronization and Backup .....	124
Description of Usecase 21 Authentication and security .....	127
Description of Usecase 22 Use MBaaS .....	129
Description of Usecase 23 Access Control.....	136
 Usecase diagram 1 Basic information of Registration.....	19
Usecase diagram 2 Body information of Registration.....	21
Usecase diagram 3 Account of Registration .....	24
Usecase diagram 4 Login.....	25

Usecase diagram 5 Customize dashboard.....	30
Usecase diagram 6 Edit dashboard .....	34
Usecase diagram 7 Contractor updates .....	37
Usecase diagram 8 Heart rate graph .....	39
Usecase diagram 9 Step count graph .....	43
Usecase diagram 10 Start checking heart rate .....	47
Usecase diagram 11 Personalised heart rate range .....	49
Usecase diagram 12 Heart rate real time alert .....	53
Usecase diagram 13 Workout detection .....	60
Usecase diagram 14 Daily step data .....	64
Usecase diagram 15 ‘Turtleneck mode’ button activation .....	70
Usecase diagram 16 Start Turtleneck mode.....	72
Usecase diagram 17 Turtleneck detection .....	74
Usecase diagram 18 Show the total score of weekly health .....	80
Usecase diagram 19 Review for feedback .....	88
Usecase diagram 20 Sleep analysis.....	92
Usecase diagram 21 Prolonged sitting detection .....	96
Usecase diagram 22 Payment integration .....	101
Usecase diagram 23 App language .....	105
Usecase diagram 24 Notification settings.....	106
Usecase diagram 25 Theme settings .....	107
Usecase diagram 26 Network usage settings .....	109
Usecase diagram 27 Help&Information .....	110
Usecase diagram 28 Abnormal shutdown.....	111
Usecase diagram 29 Fontsize settings.....	113
Usecase diagram 30 Activate sensors .....	116
Usecase diagram 31 Collecting data .....	120
Usecase diagram 32 Pause of finish collecting data .....	122
Usecase diagram 33 Data synchronization and Backup .....	124
Usecase diagram 34 Authentication and security .....	126
Usecase diagram 35 Use MBaaS .....	128
Usecase diagram 36 Access control.....	136
 Sequence diagram 1 Registration.....	16
Sequence diagram 2 Login.....	27
Sequence diagram 3 Customize dashboard.....	31
Sequence diagram 4 Edit dashboard .....	35
Sequence diagram 5 Heart rate graph .....	41
Sequence diagram 6 Step count graph .....	45
Sequence diagram 7 Visualize heart rate .....	51
Sequence diagram 8 Heart rate real time alert .....	54
Sequence diagram 9 Stroll detection.....	61
Sequence diagram 10 Daily steps data.....	65
Sequence diagram 11 Turtleneck mode .....	78
Sequence diagram 12 Give the feedback from expert .....	82
Sequence diagram 13 Review for feedback .....	89
Sequence diagram 14 Sleep analysis .....	94
Sequence diagram 15 Prolonged sitting detection .....	96
Sequence diagram 16 Subscription.....	99

Sequence diagram 17 Abnormal shutd .....	112
Sequence diagram 18 Compatibility(1) .....	133
Sequence diagram 19 Compatibility(2) .....	134
Activity diagram 1 Registration/Login .....	15
Activity diagram 2 Customize dashboard.....	32
Activity diagram 3 Edit dashboard .....	34
Activity diagram 4 Heart rate graph .....	40
Activity diagram 5 Step count graph .....	44
Activity diagram 6 Workout detection .....	59
Activity diagram 7 Fall detection .....	63
Activity diagram 8 the average of weekly steps .....	67
Activity diagram 9 Turtleneck mode .....	77
Activity diagram 10 Send the alarm to user.....	87
Activity diagram 11 Subscription .....	98
Activity diagram 12 Activate sensors .....	117
Activity diagram 13 Collecting data.....	121
Activity diagram 14 Data synchronization and Backup .....	125
Class diagram 1 .....	138

# 1. Attendance

## 1. Attendance list

	1	2	3	4	5	6	7	8	9	10	11	12
Name	31.10.23	07.11.23	14.11.23	05.12.23	12.12.23	14.12.23	16.12.23	18.12.23	19.12.23	09.01.24	12.01.24	
Changmin	33%	25%	25%	0%	25%	20%	25%	25%	20%	25%	25%	
Jeongmin	33%	25%	25%	0%	25%	20%	25%	25%	20%	25%	25%	
Yoonsun	25%	25%	50%	0%	20%	25%	25%	20%	0%	25%	0%	
Soojin	33%	25%	25%	0%	25%	20%	25%	20%	0%	25%	25%	
Eric	0%	20%	0%	25%	25%	0%	0%	20%	20%	25%	0%	

	13	14	15	16	17	18	19	20	21	22	23	24
Date	16.01.24	17.01.24	22.01.24	23.01.24	25.01.24	30.01.24	02.02.24	05.02.24	06.02.24	07.02.24	08.02.24	09.02.24
Changmin	25%	20%	25%	25%	25%	25%	25%	25%	20%	20%	25%	22%
Jeongmin	25%	20%	25%	0%	0%	0%	25%	25%	20%	20%	25%	22%
Yoonsun	25%	20%	25%	25%	25%	25%	25%	25%	20%	20%	25%	22%
Soojin	25%	20%	25%	25%	25%	25%	25%	25%	20%	20%	25%	22%
Eric	0%	20%	0%	25%	25%	25%	0%	0%	20%	20%	0%	12%

100% = Done

Table 1 Attendance list

## 2. Meeting Point & Protocol

Meeting Protocol and Scrums are created on starting from Tuesday (Class date), detailing what we should have done and how much we have accomplished.

### 2.1. Meeting Protocol

Date	Task	Place
24.10.23	Worksheet 1	Frankfurt UAS
31.10.23	Worksheet 2 / Snowcards	Frankfurt UAS
07.11.23	Worksheet 3 / Task distribution	Frankfurt UAS
14.11.23	Worksheet 4 / UCD	Frankfurt UAS
05.12.23	Worksheet 5 / Activity Diagrams	Frankfurt UAS
12.12.23	Worksheet 6 / Class Diagrams	Frankfurt UAS
14.12.23	Work on the presentation	Frankfurt UAS
16.12.23	Presentation Powerpoint	Frankfurt UAS
18.12.23	Work on the presentation	Google Meet
19.12.23	Work on the presentation	Frankfurt UAS
09.01.24	Diagrams Powerpoint	Frankfurt UAS
12.01.24	Work on the report	Frankfurt UAS
16.01.24	Work on the report	Frankfurt UAS
17.01.24	Work on the report	Frankfurt UAS
22.01.24	Work on the presentation	Frankfurt UAS
23.01.24	Work on the presentation	Frankfurt UAS
25.01.24	Work on the report	Frankfurt UAS
30.01.24	Work on the report	Frankfurt UAS

02.02.24	Work on the report	Frankfurt UAS
05.02.24	Work on the report	Frankfurt UAS
06.02.24	Work on the presentation	Frankfurt UAS
07.02.24	Work on the report	Frankfurt UAS
08.02.24	Work on the report	Frankfurt UAS
09.02.24	Work on the report	Frankfurt UAS

Table 2 Meeting protocol

## 2.2. Scrum

Scrum is an agile team collaboration framework commonly used in software development and other industries.

Scrum prescribes for teams to break work into goals to be completed within time-boxed iterations, called *sprints*. Each sprint is no longer than one month and commonly lasts two weeks. The scrum team assesses progress in time-boxed, stand-up-meetings of up to 15 minutes, called daily scrums. At the end of the sprint, the team holds two further meetings: one sprint review to demonstrate the work for stakeholders and solicit feedback, and one internal sprint-retrospective.

Scrum's approach to product development involves bringing decision-making authority to an operational level. Unlike a sequential approach to product development, scrum is an iterative and incremental framework for product development. Scrum allows for continuous feedback and flexibility, requiring teams to self-organize by encouraging physical co-location or close online collaboration, and mandating frequent communication among all team members. The flexible and semi-unplanned approach of scrum is based in part on the notion of requirements volatility, that stakeholders will change their requirements as the project evolves.

([https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)#See\\_also](https://en.wikipedia.org/wiki/Scrum_(software_development)#See_also))

### 3. Declaration or Authorship

We hereby certify that the project report we are submitting is entirely our own original work except where otherwise indicated. We did not submit this work anywhere else before. We are aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Frankfurt am Main, date: 09.02.2024

Signature:

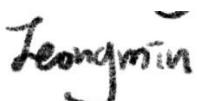
Changmin Lee (1516177)



Eric Jacob (1409493)



Jeongmin Ahn (1522888)



Yoonsun Lee (1519516)



Sooin Jung (1524022)



## **4. Introduction**

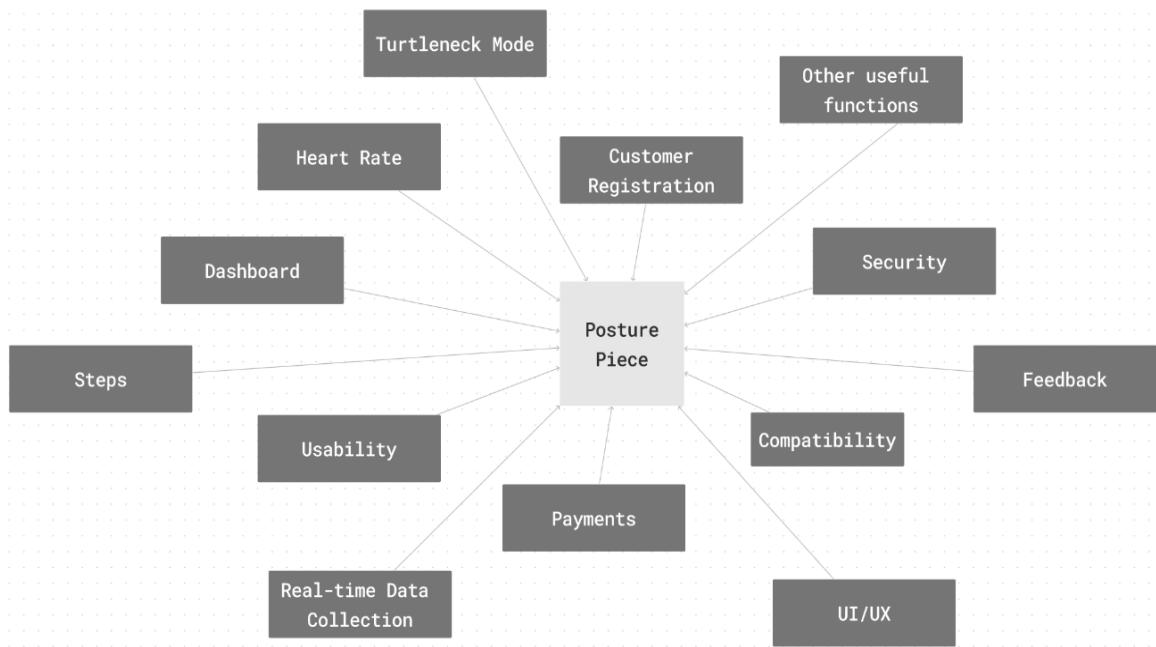
These days, people's interest in health has grown, especially since the COVID-19 era, and the number of people sitting in the wrong position for long periods of time at home or in the office is increasing due to the increase in remote working and hybrid working methods.

In everyday life, correcting ingrained improper walking or sitting postures is challenging as individuals may not be consciously aware of these habits. Even when awareness is present, without consistent efforts to address and modify them, habitual postures are unlikely to change. Through this software project, we aim to facilitate continuous self-awareness regarding users' physical condition and posture. Our goal is to provide ongoing feedback, moving beyond mere diagnosis to offer prescriptive guidance based on diagnostic results.

This approach empowers users to actively engage in posture improvement, recognizing that momentary awareness is fleeting, and sustained efforts are essential for breaking entrenched habits and contributing to overall health consciousness.

## 5. Overview

These are the whole requirements for implementing the 'Posturepiece' software system, divided into 13 major headings.



*Image 1 Overview*

## 6. Product Backlog

The Product Backlog is a prioritized list of tasks needed for software development, including features, bug fixes, and enhancements. It evolves over time based on stakeholder feedback and serves as a roadmap for the development team.

(<https://www.atlassian.com/agile/product-management/product-backlog>)

ID	Actor	Title	Business value	Week
1	Customer, Contractor	Customer registration, login	high	6
2	Customer	Dashboard	high	8
3	Customer	Heart rate	high	11
4	Customer	Steps	high	9
5	Customer	Turtleneck mode	high	11
6	Customer	Feedback from expert	high	11
7	Customer	Other useful functions	medium	13
8	Customer, Contractor	Payment & Subscription	high	10
9	Customer	Usability	high	8
10	Contractor	Real-time data collection and logging	high	10
11	Contractor	Compatibility	high	8
12	Contractor	Security	high	11

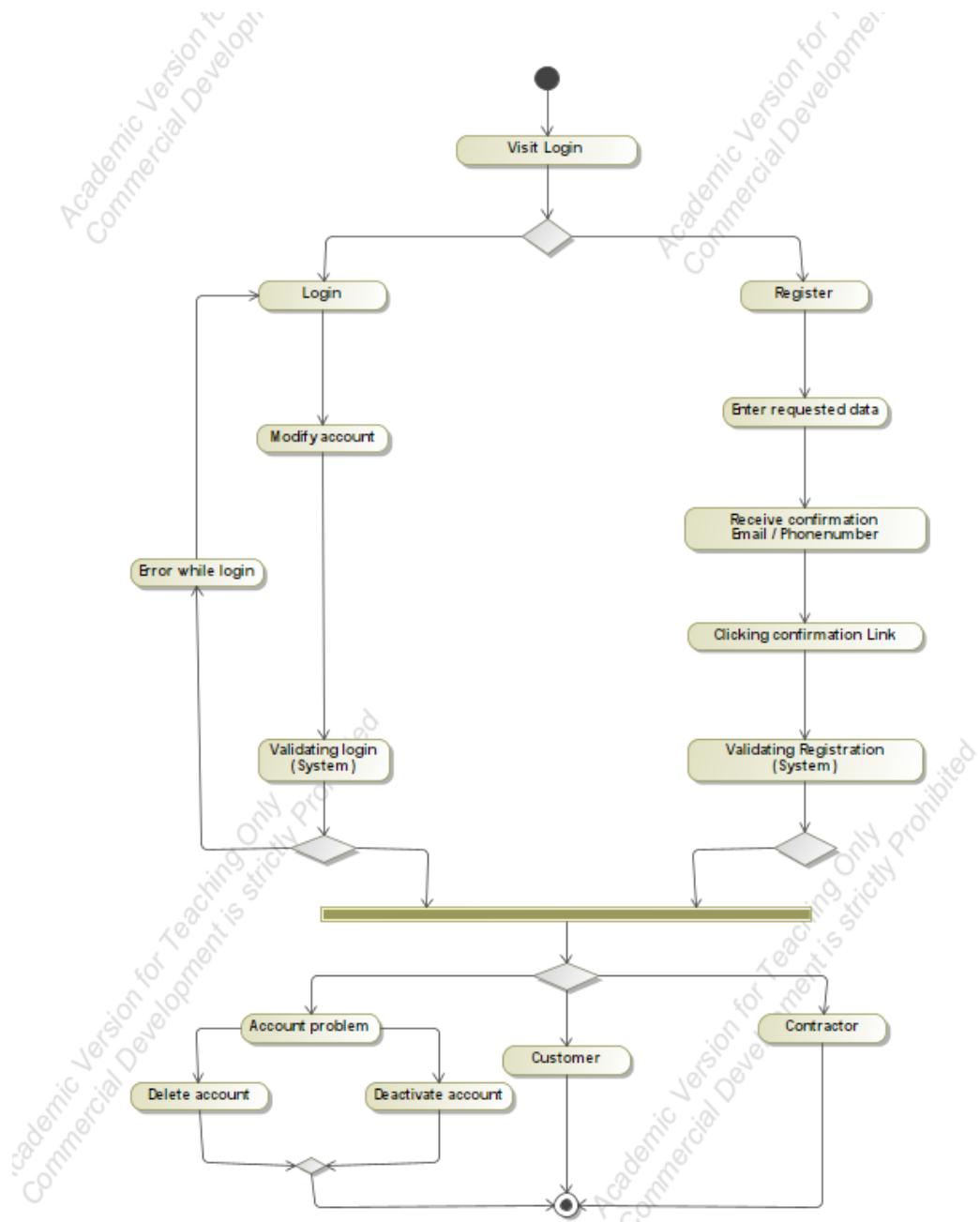
Table 3 Backlog

Tasks	%	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	
Introduction																					Soojin(100%)	
Attendance list																					Yoonsun(100%)	
Scrum																					Soojin(100%)	
Product Backlog																					Yoonsun(100%)	
Processes		Changmin(21%)		Jeongmin(21%)			Eric(16%)			Yoonsun(21%)			Soojin(21%)									
Snowcard		Changmin(20%)		Jeongmin(20%)			Eric(20%)			Yoonsun(20%)			Soojin(20%)									
Requirement 1. Customer registration																					Jeongmin(100%)	
Requirement 2. Dashboard																					Eric(100%)	
Requirement 3. Heart rate																					Soojin(100%)	
Requirement 4. Steps																					Jeongmin(100%)	
Requirement 5. Turtleneck mode																					Yoonsun(100%)	
Requirement 6. Feedback from expert																					Soojin(100%)	
Requirement 7. Other useful functions							Eric(50%)														Yoonsun(50%)	
Requirement 8. Payment & Subscription																					Changmin(100%)	
Requirement 9. Usability																					Yoonsun(100%)	
Requirement 10. Real-time Data Collection and Logging																					Soojin(100%)	
Requirement 11. Compatibility																					Changmin(100%)	
Requirement 12. Security																					Changmin(100%)	
UML Class diagram							Changmin(30%)			Jeongmin(30%)			Yoonsun(30%)								Soojin(20%)	
UI/UX Prototype								Changmin(45%)					Yoonsun(30%)									Jeongmin(25%)
Summary																					Jeongmin(100%)	



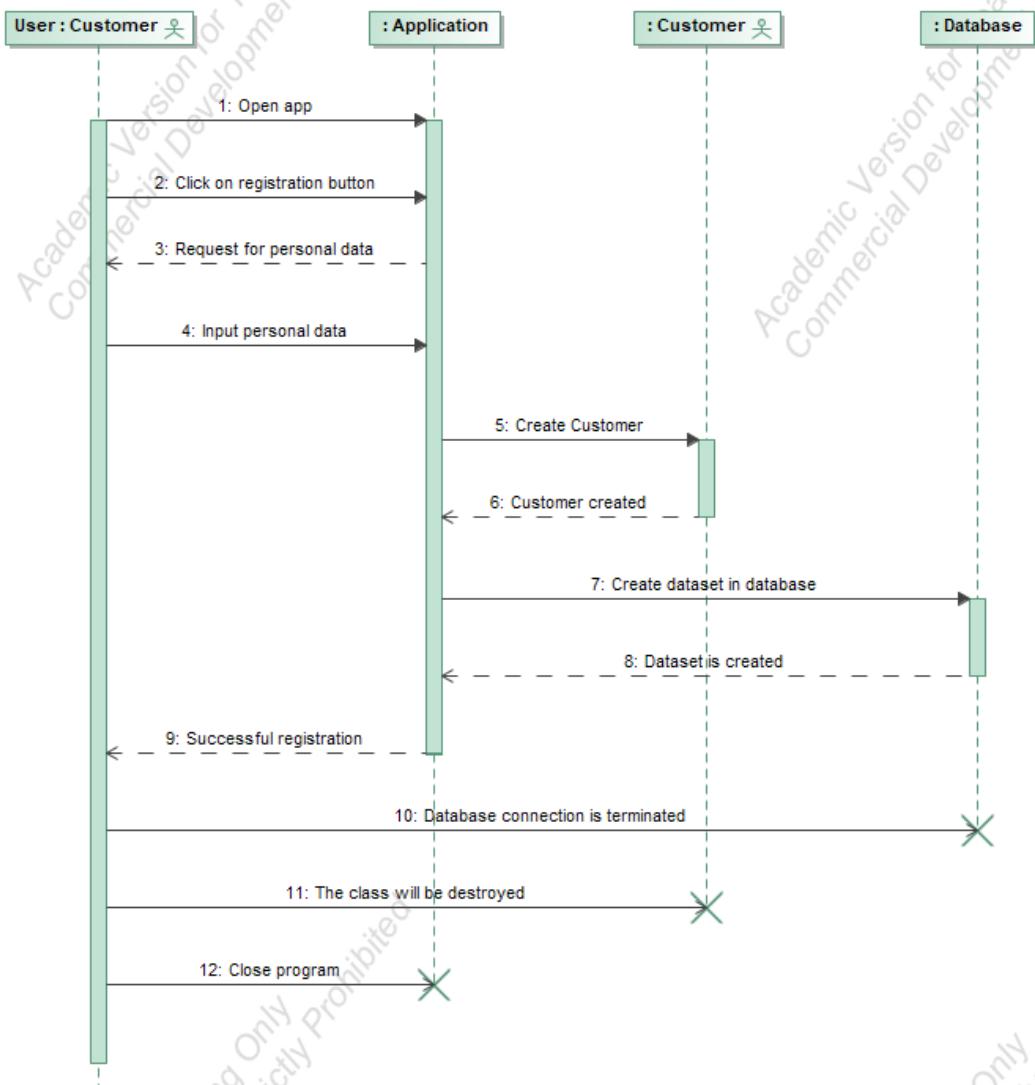
Table 4 Backlog\_members

## 7. Customer registration, login



Activity diagram 1 Registration/Login

## 7.1. Registration



Sequence diagram 1 Registration

During the registration process, the user's personal data is collected and stored securely in the database. This information is essential for creating a unique user profile. The access data, including a username and password, is then generated and stored, allowing the user to log in at any time.

For a successful registration, the following information is typically required:

Name  
Surname

Birthday  
Email  
Phone number  
Bank account  
Address  
Zip  
City  
Language  
Profile Picture  
Height  
Weight  
Gender  
Age  
Interactive surveys  
ID  
Passwords  
Privacy Policy Agreements

(1-6) Name/Surname/Birthday/Email/Phone number/Bank account are mandatory basic information.

(7-11) Address/Zip/ City/Language/Profile Picture are optional basic information.

(12-16) Height/Weight/Gender/Age/Interactive surveys are the body information.

(17-19) ID/Passwords/Privacy Policy Agreements are account information.

These details collectively contribute to a comprehensive user profile, ensuring a secure and personalized experience within the software service.

### 7.1.1. Basic information

When a user registers and logs in to utilize the service, contractor can collect personal information such as name and date of birth for unique identification. This allows the service to identify each user and provide a personalized

experience. Additionally, the collection of basic information, including name and date of birth, is necessary for service delivery, customization, addressing security concerns, and compliance with legal regulations and authorization requests.

And then, Phone number and Email are required for personal identity verification. In case of unsuccessful login attempts, users can resolve the issue through personal identity verification.

Additionally, Bank account is connected to payment system. When entering language ability, the contractor can enter any number of languages.

### **Certain criteria must be met when entering data.**

Name/Surname must be at least 3 characters long.

Birthday format is DD/MM/YYYY.

Phone number format is that the customer has to input customer's personal number along with the country code.

The Bank account must be verified including account number and bank name.

The Address (including City) is compared with the database and a corresponding zip code is suggested.

At least one language must be selected for use.



## Customer Access Management of requirements

### #01.Basic Information of Registration

**Requirement Type:** functional

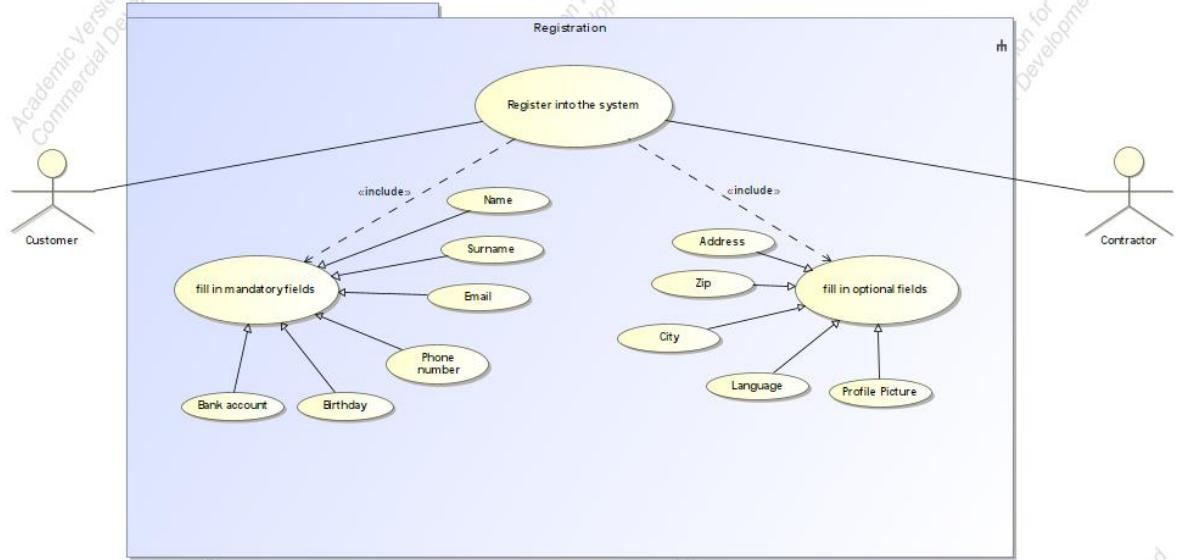
**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### **Description:**

1. Mandatory Information : Name, Surname, Birthday, Email, Phone number and Bank account
  - Email / Phone-number for identity verification.
  - Bank account for payments
2. Optional Information : Address, Zip, City, Language, Profile Picture



Usecase diagram 1 Basic information of Registration

<b>Name</b>	Basic information of registration
<b>ID</b>	8.1.1
<b>Description</b>	The contractor collects personal basic information which allows the service to identify each user.
<b>Trigger</b>	The customer wants to log in and opens the app or web.
<b>Actors</b>	Customer, Contractor
<b>Pre-conditions</b>	App installed or web site, connected to the internet.
<b>Post-conditions</b>	The customer is successfully connect to the system.
<b>Basic Flow</b>	
<b>Description</b>	The mandatory fields must be filled out.
<b>Actions</b>	<ol style="list-style-type: none"> <li>1 Name/Surname/Birthday are essential elements.</li> <li>2 Email and Phone number are requested for identity verification purposes.</li> <li>3 Bank account information is essential at the purchase step.</li> </ol>
<b>Alternative Flow</b>	
<b>Description</b>	All optional fields have been filled out.
<b>Actions</b>	<ol style="list-style-type: none"> <li>1 Utilizing location data based on the address enables on the selected language.</li> <li>2 Detailed service diagnostics are provided based on the selected language.</li> </ol>
<b>Alternative Flow</b>	
<b>Description</b>	Not all optional fields have been filled out.
<b>Actions</b>	<ol style="list-style-type: none"> <li>1 Without the address-based location data, more accurate data collection is not possible.</li> <li>2 General service diagnostics are provided in the default language, English.</li> </ol>

Description of Usecase 1 Basic Information

## 7.1.2. Body Information

Body information refers to the user's health status and specific body measurement-related data. It aids users in tracking and managing their health status, while applications leverage this information to generate personalized

advice and reports. By collecting such data, users can maintain better health and receive assistance in achieving their health goals.

Additionally, to enhance the accuracy of the service, we proactively collect input values from users for data that cannot be measured by a smartwatch, such as weight and height. Users provide information such as height, weight, gender, age, and respond to a brief survey about their health status. This enables us to understand if users have any pre-existing medical conditions. Users also receive notifications to periodically update their health information.

**Certain criteria must be met when entering data.**

Height and Weight must be input in appropriate units (e.g., centimeters for height, kilograms for weight).

Gender should be selected from the provided options (e.g., Male, Female, Other).

Age should be entered as a numeric value.

Interactive surveys about the user's health status, including any pre-existing medical conditions, must be completed.

## Customer Access Management of requirements

### #02.Body Information of Registration

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

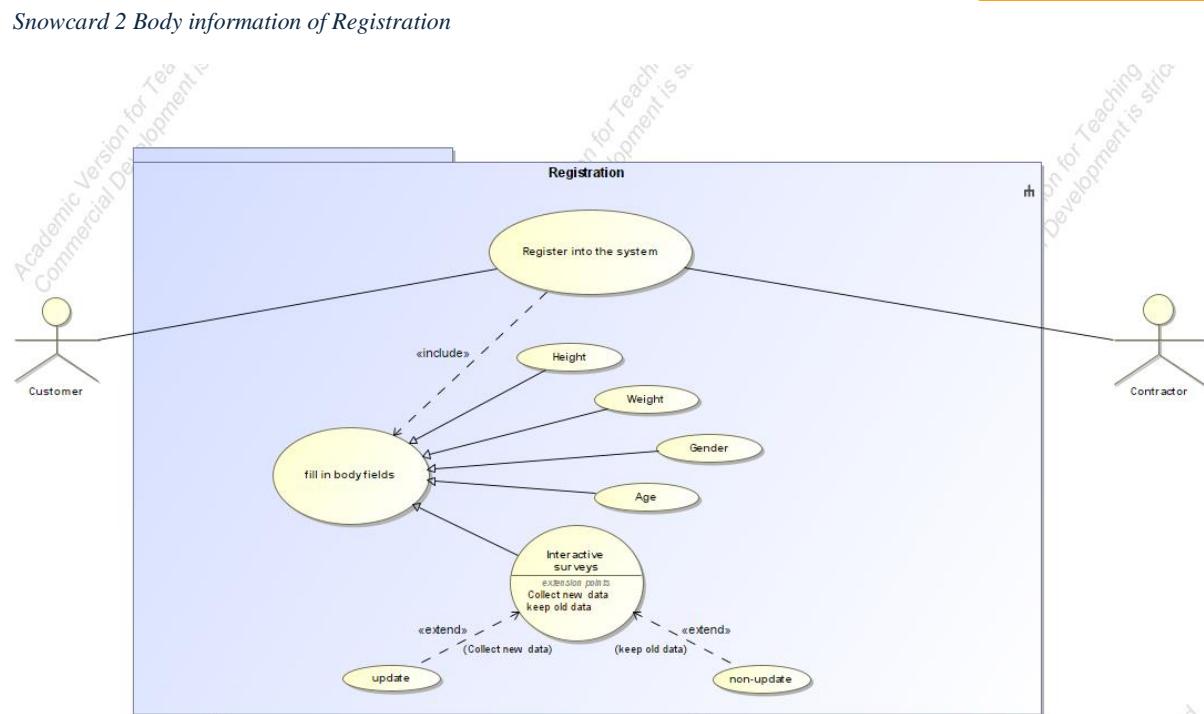
#### Description:

If customers provide detailed health information in advance, we can utilize the data more accurately. The basic body information is pre-input into the database.

- Height
- Weight
- Gender
- Age
- Interactive surveys : Existing Health Conditions

Subsequently, customers receive notifications to periodically update their health information.

Seite 3



Use case diagram 2 Body information of Registration

<b>Name</b>	Body information of registration
<b>ID</b>	8.1.2
<b>Description</b>	Body information refers to the user's health status and specific body measurement-related data.
<b>Trigger</b>	It aids users in tracking and managing their health status.
<b>Actors</b>	Customer, Contractor
<b>Pre-conditions</b>	The input of basic information is completed.
<b>Post-conditions</b>	It leverage this information to generate personalized advice and reports.
<b>Basic Flow</b>	<p><b>Description</b></p> <p>Customized diagnosis for the user is required.</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> Periodic pop-up are displayed to gather input for basic responses.</li> <li><b>2</b> The button to accept body information input is pressed.</li> <li><b>3</b> Inputs such as height, weight, etc are collected.</li> <li><b>4</b> It can help the user achieve their health goals.</li> </ul>
<b>Alternative Flow</b>	<p><b>Description</b></p> <p>Customized diagnosis for the user is required.</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> Periodic pop-up are displayed to gather input for basic responses.</li> <li><b>2</b> Input values for data that cannot be measured by the</li> <li><b>3</b> Respond to a brief survey about health status.</li> <li><b>4</b> Understand the user's existing medical conditions.</li> </ul>
<b>Alternative Flow</b>	<p><b>Description</b></p> <p>Customized diagnosis for the user may be not required.</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> Periodic pop-up are displayed to gather input for optional responses.</li> <li><b>2</b> Unable to collect input for optional responses.</li> <li><b>3</b> Show users diagnostic results that are not entirely precise.</li> </ul>

*Description of Usecase 2 Body information*

### 7.1.3. Account Information

The customer has personal ID/Passwords. Additionally agreeing to the privacy policy is required for a successful account creation.

**Privacy Policy Agreements:** To successfully create an account, the customer must carefully read and agree to the privacy policy. This ensures that the user is aware of how their personal information will be handled and establishes a transparent relationship between the user and the service provider.

**Certain criteria must be met when entering data.**

**ID**

Minimum of 7 digits

Consist of Number

Consist of Lowercase letter

1. Passwords
  - A. Minimum of 12 digits
  - B. Consist of Capital letter
  - C. Consist of Lowercase letter
  - D. Consist of Number
  - E. No country specific letter
2. Privacy Policy Agreements
  - A. Yes
  - B. No



## Customer Access Management of requirements

### #03.Account of Registration

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / high

**User Dissatisfaction** low / medium / high

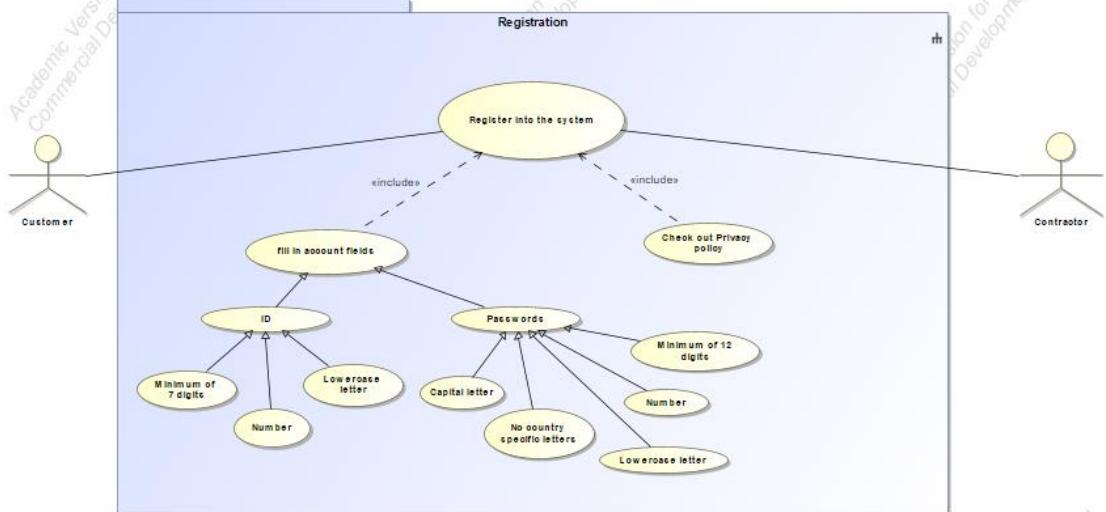
#### Description:

The customer has personal ID / Passwords. Before creating an account, there are several conditions as follows. Additionally agreeing to the privacy policy is required for a successful account creation.

- ID : Minimum of 7 digits / Consist of Number and Lowercase letter
- Passwords : Minimum of 12 digits / Consist of Capital Letter, Lowercase Letter and Number / No country specific letter
- Privacy Policy : Agreeing to the utilization of personal information and receiving notifications for changing the password every 3 months in the future

Seite 4

Snowcard 3 Account of Registration



Usecase diagram 3 Account of Registration

<b>Name</b>	Account of registration
<b>ID</b>	8.1.3
<b>Description</b>	The customer has to retain personal ID/Passwords.
<b>Trigger</b>	The customer wants to log in and starts our service.
<b>Actors</b>	Customer, Contractor
<b>Pre-conditions</b>	Agreeing to the privacy policy is required for a successful account creation
<b>Post-conditions</b>	The customer is successfully connect to the system.
<b>Basic Flow</b>	
<b>Description</b>	The user is prompted to check for consent to the privacy policy.
<b>Actions</b>	
1	ID/Passwords are generated according to each condition.
2	A pop-up indicating successful account creation is displayed on the screen.
<b>Alternative Flow</b>	A
<b>Description</b>	The user is prompted to check for consent to the privacy policy.
<b>Actions</b>	
1	ID/Passwords are not generated according to each condition.
2	A pop-up indicating that the account creation was not successful and prompting to recreate according to the conditions is displayed on the screen.
<b>Alternative Flow</b>	B
<b>Description</b>	The user fails to check the checkbox for consent to the privacy policy.
<b>Actions</b>	
1	ID/Passwords are not generated according to each condition.
2	A pop-up instructing the user to agree to the privacy policy is displayed on the screen.

Description of Usecase 3 Account of registration

## 7.2. Login

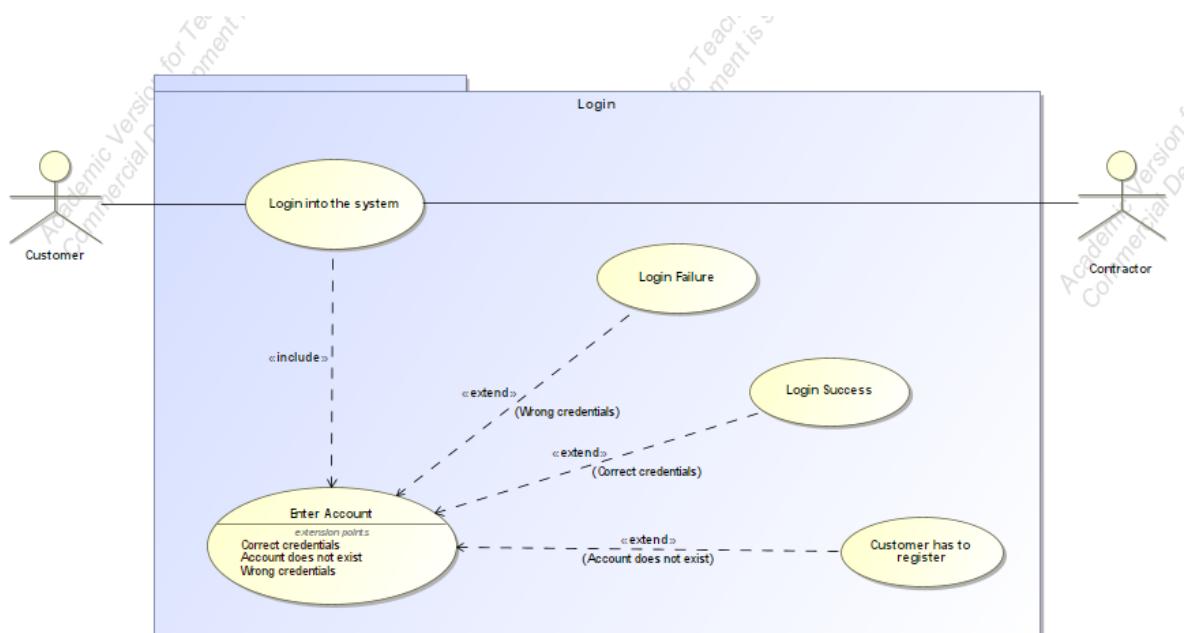
To access the system, the customer is required to log in using the credentials chosen during the registration process. The login involves entering either the assigned registered ID, along with the corresponding password.

### Successful Login

In the event that the customer accurately inputs the correct combination of ID and password, the system grants successful login access. The customer is then directed to the personalized dashboard or main interface.

### Login Retry

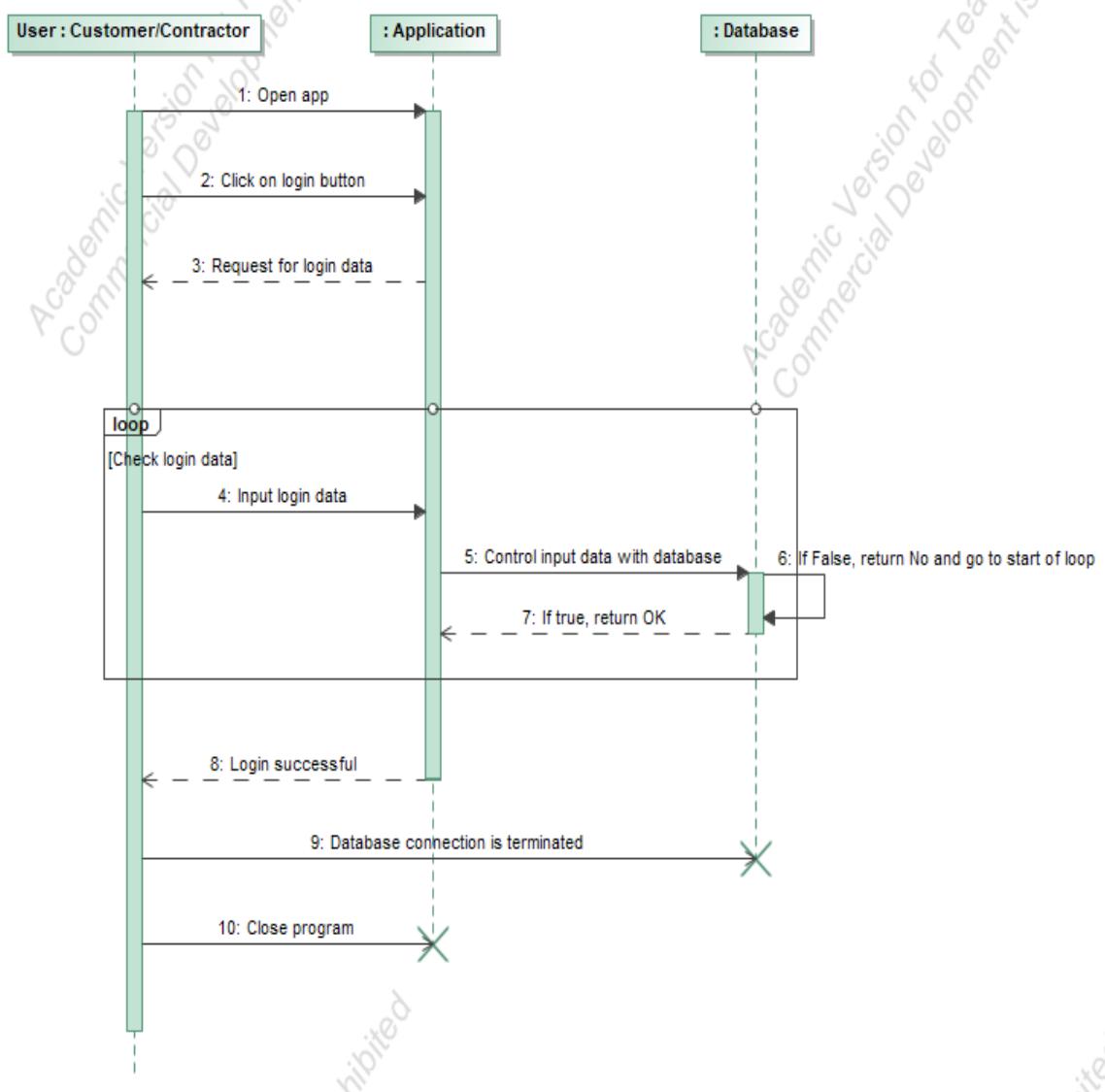
If the customer enters incorrect credentials, the system prompts a login retry. The customer is encouraged to verify and re-enter ID along with the correct password. After several unsuccessful attempts, additional security measures such as account lockout or a password reset process may be triggered to ensure account security.



Use case diagram 4 Login

<b>Name</b>	Login
<b>ID</b>	8.2
<b>Description</b>	The user is required to login in using the credentials chosen during the registration process.
<b>Trigger</b>	The user wants to access the system.
<b>Actors</b>	Customer, Contractor
<b>Pre-conditions</b>	The user has to register.
<b>Post-conditions</b>	The user can login into the system.
<b>Basic Flow</b>	<p><b>Description</b></p> <p>The user accesses the login form.</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> They enter the correct ID and Passwords.</li> <li><b>2</b> They successfully access the system.</li> <li><b>3</b> The user can access their personal dashboard.</li> </ul>
<b>Alternative Flow</b>	<p>A</p> <p>The user accesses the login form.</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> They fail to enter the correct ID and Passwords.</li> <li><b>2</b> The system sends a notification to retry login.</li> <li><b>3</b> The user is encouraged to enter the ID/Passwords correctly.</li> </ul>
<b>Alternative Flow</b>	<p>B</p> <p>The user accesses the login form.</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> They repeatedly fail to enter the correct ID and Passwords.</li> <li><b>2</b> Additional security measures are provided.</li> <li><b>3</b> Account lockout or password reset procedures are initiated.</li> </ul>

*Description of Usecase 4 Login*



Sequence diagram 2 Login

### 7.3. Enter Account

## Customer Access Management of requirements

### #04.Enter Account of Login

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### Description:

The customer / contractor has to login with ID and password to use our services. In the case of unsuccessful login, there are two main issues.

- 1) ID-related problem
  - Case : Forget customer's own ID
  - Solution : To confirm the individual's identity through personal information like Email/Phone number, and then find the ID.
- 1) Password-related problem
  - Case : Forget customer's own password or enter the wrong password more than 5 times consecutively
  - Solution : To reset the password under the condition that the same password cannot be reused after confirming the customer's identity.

Seite 5

*Snowcard 4 Enter Account of Login*

## 7.4. Delete Account

## Customer Access Management of requirements

### #05.Delete Account

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### Description:

If the customer does not want to use the platform anymore

- Customer delete all the data which is related to the customer should be deleted.
- Customer can not login to the system anymore.

If the customer wants to temporarily pause their service usage for a certain period

- Customer can deactivate their account.

Seite 6

*Snowcard 5 Delete account*

## 8. Dashboard

### 8.1. Customize Dashboard

The Software requirement “Customize Dashboard “is a function which will allow the user of the smartwatch to freely choose a layout for his dashboard.

These layouts will be built and uploaded by the supplier of the software.  
The user also has the option to build his own Dashboard layout.  
Using this function the user will be able to choose the Background picture or color, the placement of the widgets in the Dashboard and how many should be shown and small details like font or the size of the widgets.

#### # Customize Dashboard



**Requirement Type:** functional

**For Whom?** customer

**User Satisfaction:** high

**User Dissatisfaction:** low

**Description:**

User is able to choose a prefixed design for the dashboard or create a layout himself.

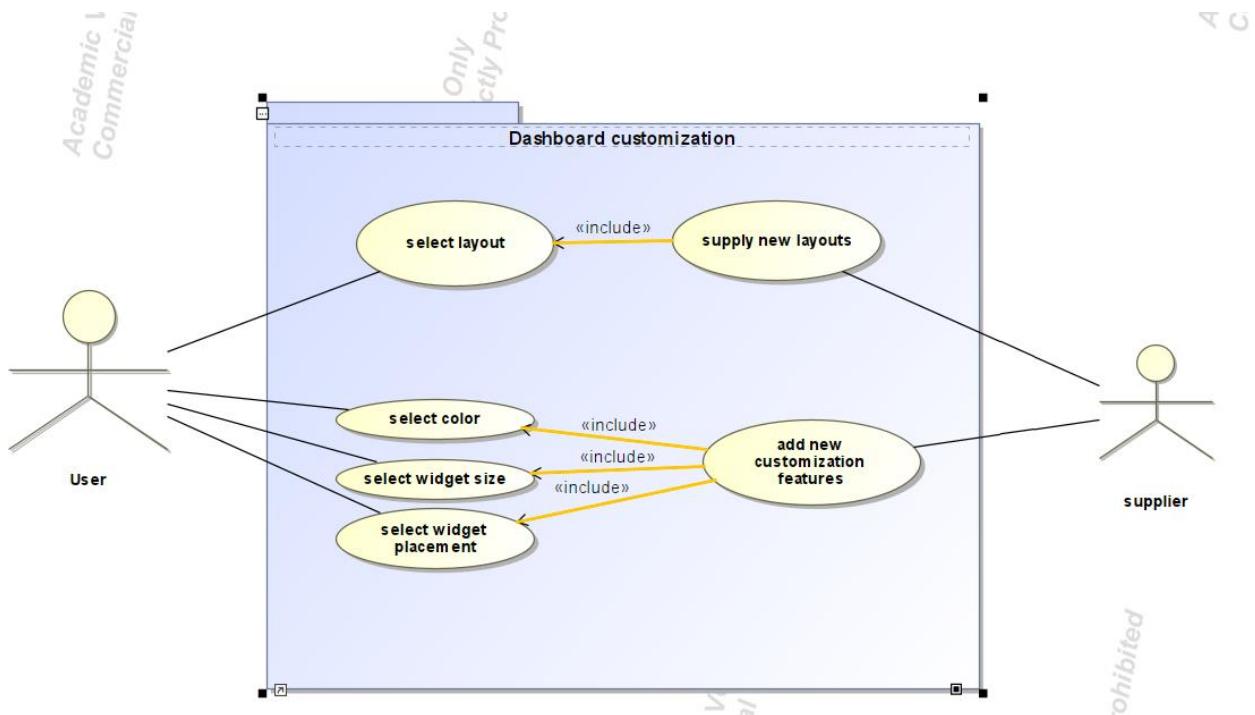
Software Engineering Analysis

Snowcard 6 Customize dashboard

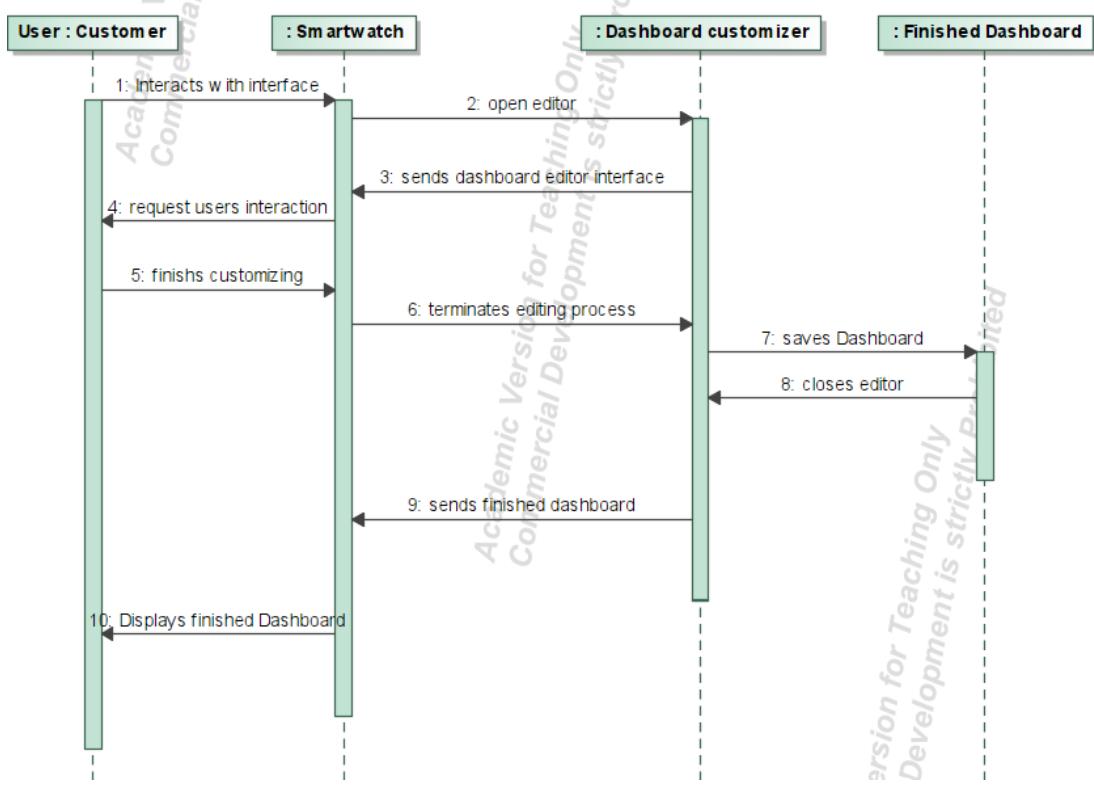


<b>Name</b>	Customize Dashboard
<b>ID</b>	1
<b>Description</b>	User selects a prefixed design for the Dashboard or creates a own layout
<b>Trigger</b>	User selects the edit function
<b>Actors</b>	User, Contractor
<b>Pre-conditions</b>	/
<b>Post-conditions</b>	/
<b>Basic Flow</b>	
<b>Description</b>	User selects a layout from the presets.
<b>Actions</b>	
<b>1</b>	User browses available layouts
<b>2</b>	Selects Layout
<b>Alternative Flow</b>	A
<b>Description</b>	Wants to customize own Layout
<b>Actions</b>	
<b>1</b>	Selects background color
<b>2</b>	Selects widget sizes
<b>3</b>	Selects widget placements

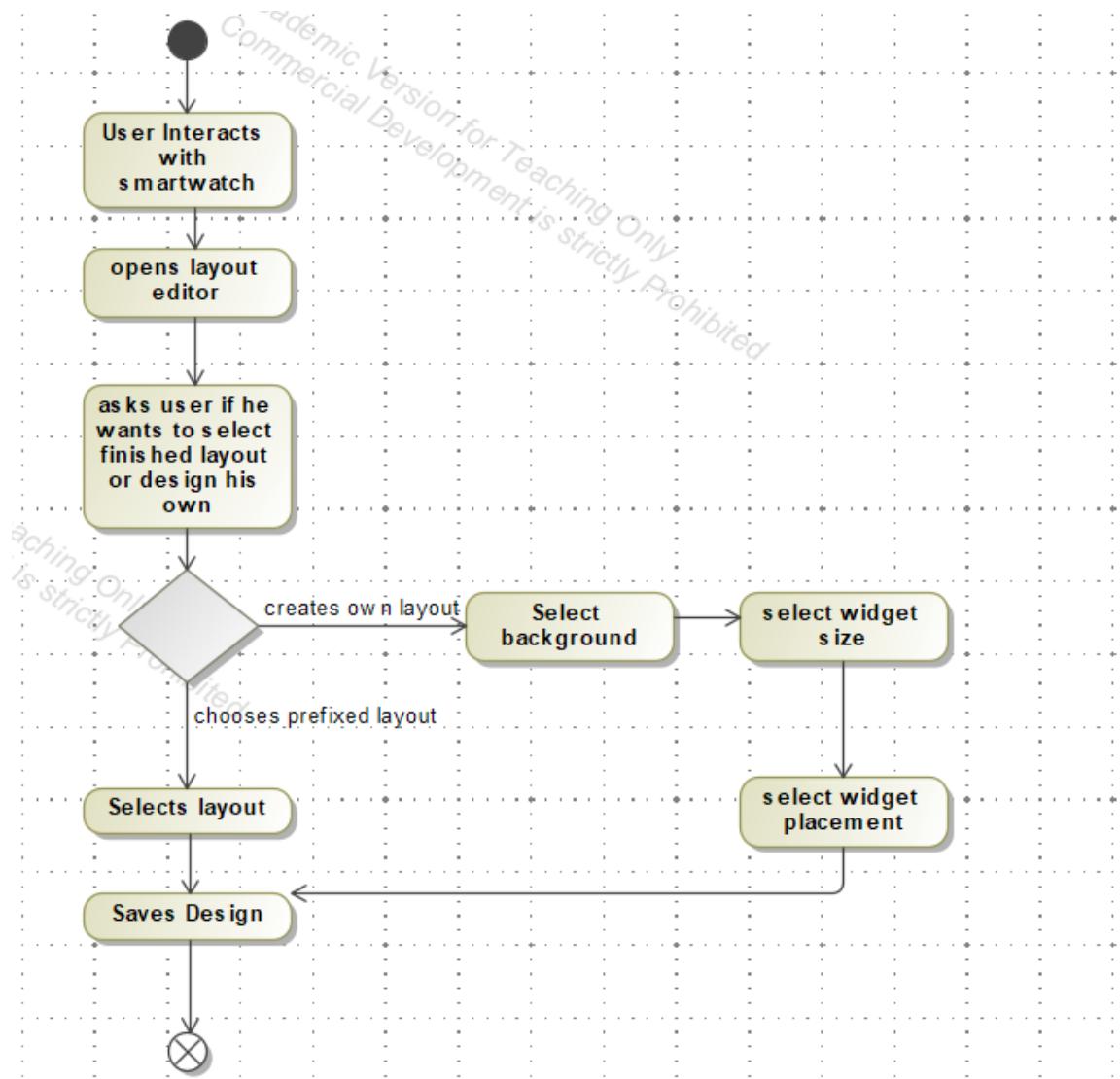
*Description of Usecase 5 Customize dashboard*



*Usecase diagram 5 Customize dashboard*



Sequence diagram 3 Customize dashboard



*Activity diagram 2 Customize dashboard*

## 8.2. Edit Dashboard

The requirement Edit Dashboard allows the user of the software to select and remove widgets on the Dashboard. The user can choose the number of widgets that are most important to him and always have them Displayed. The number of widgets on the Dashboard are determined by the layout which is selected in the previous requirement.

## # edit app data on the dashboard

**Requirement Type:** functional

**For Whom?** customer

**User Satisfaction:** high

**User Dissatisfaction:** low

**Description:**

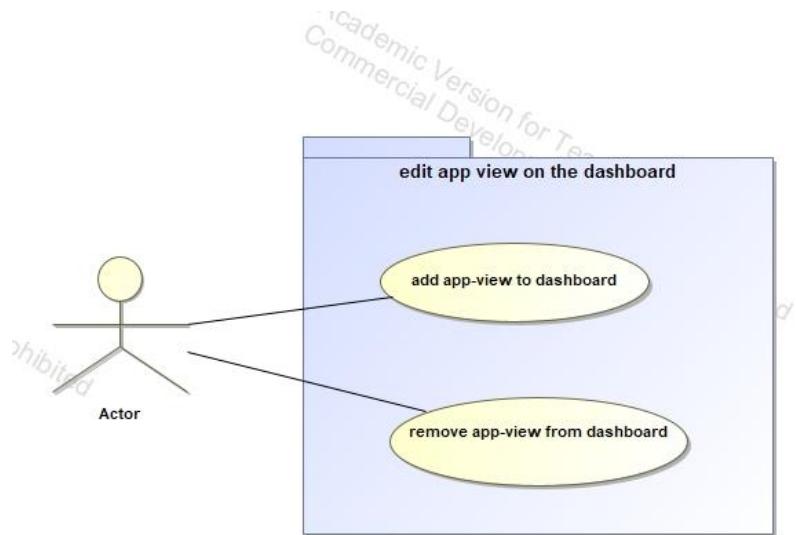
The user is able to add and remove up to three apps to the dashboard which always show a summary of ~~realtime~~ data.

Software Engineering Analysis

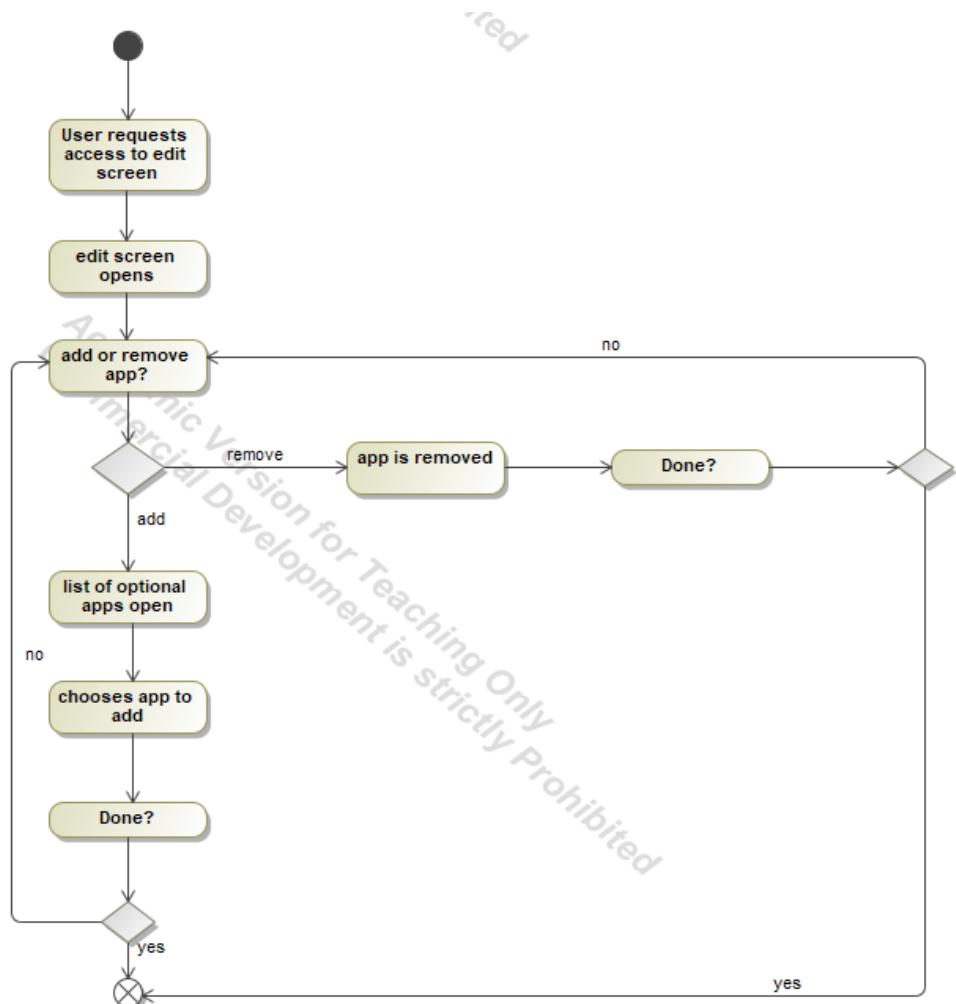
Snowcard 7 Edit dashboard

<b>Name</b>	Edit Dashboard
<b>ID</b>	2
<b>Description</b>	User adds and removes apps on the Dashboard
<b>Trigger</b>	User activates edit mode
<b>Actors</b>	user
<b>Pre-conditions</b>	
<b>Post-conditions</b>	
<b>Basic Flow</b>	
<b>Description</b>	User chooses the widgets he wants on the Dashboard
<b>Actions</b>	
<b>1</b>	User removes widgets he doesn't want from the Dashboard
<b>2</b>	User adds wanted apps
<b>Alternative Flow</b>	/
<b>Description</b>	
<b>Actions</b>	
<b>1</b>	
<b>2</b>	

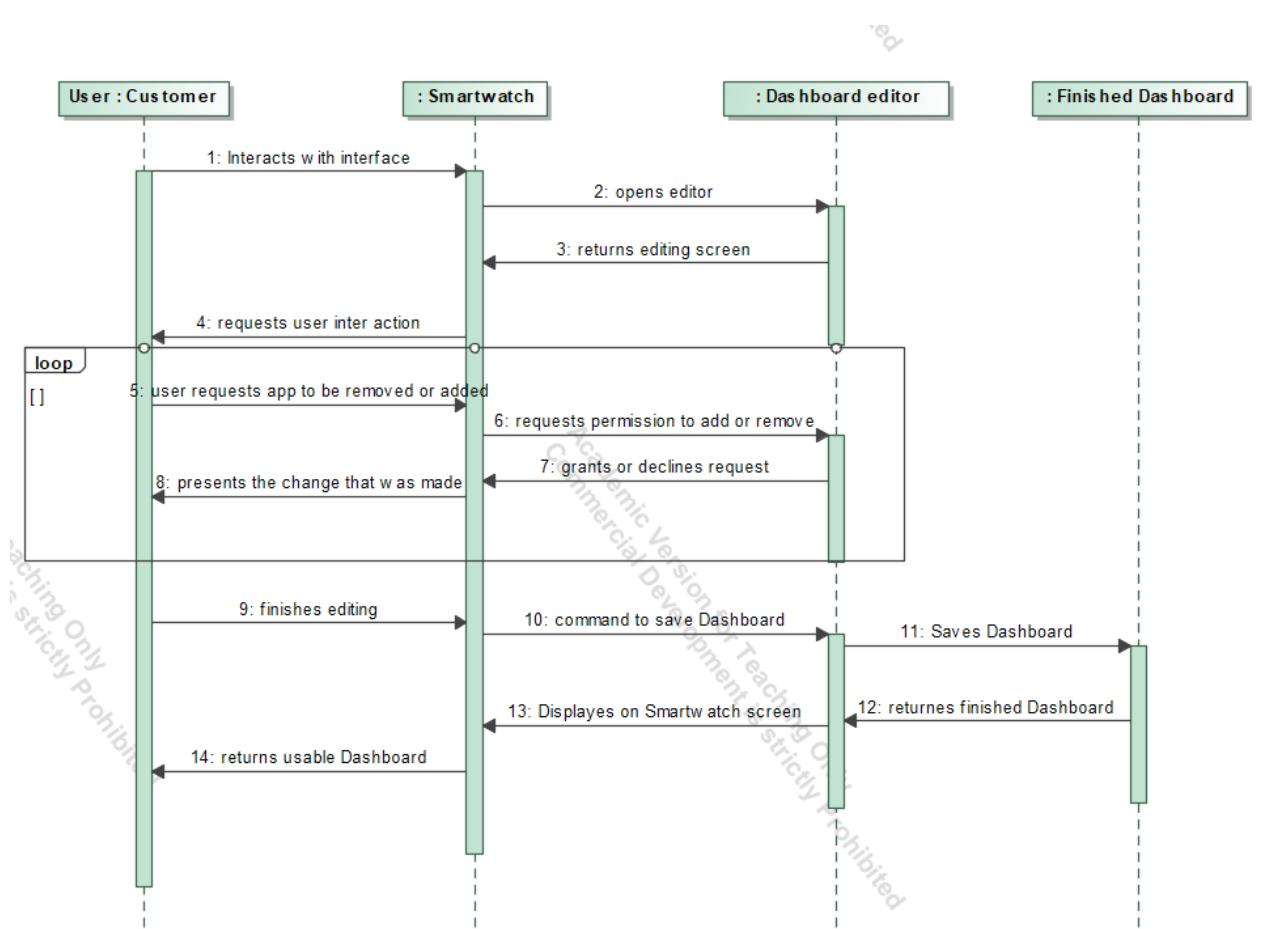
Description of Usecase 6 Edit dashboard



Usecase diagram 6 Edit dashboard



Activity diagram 3 Edit dashboard



Sequence diagram 4 Edit dashboard

### 8.3. Contractor Dashboard Updates

The requirement contractor updates make the Software developer constantly react to user feedback and add features to the dashboard to make the customization process better and easier. These Updates include new layouts, new widget features and functions.

## #3 Contractor updates

**Requirement Type:** non-functional

**For Whom?** contractor

**User Satisfaction:** high

**User Dissatisfaction:** low

**Description:**

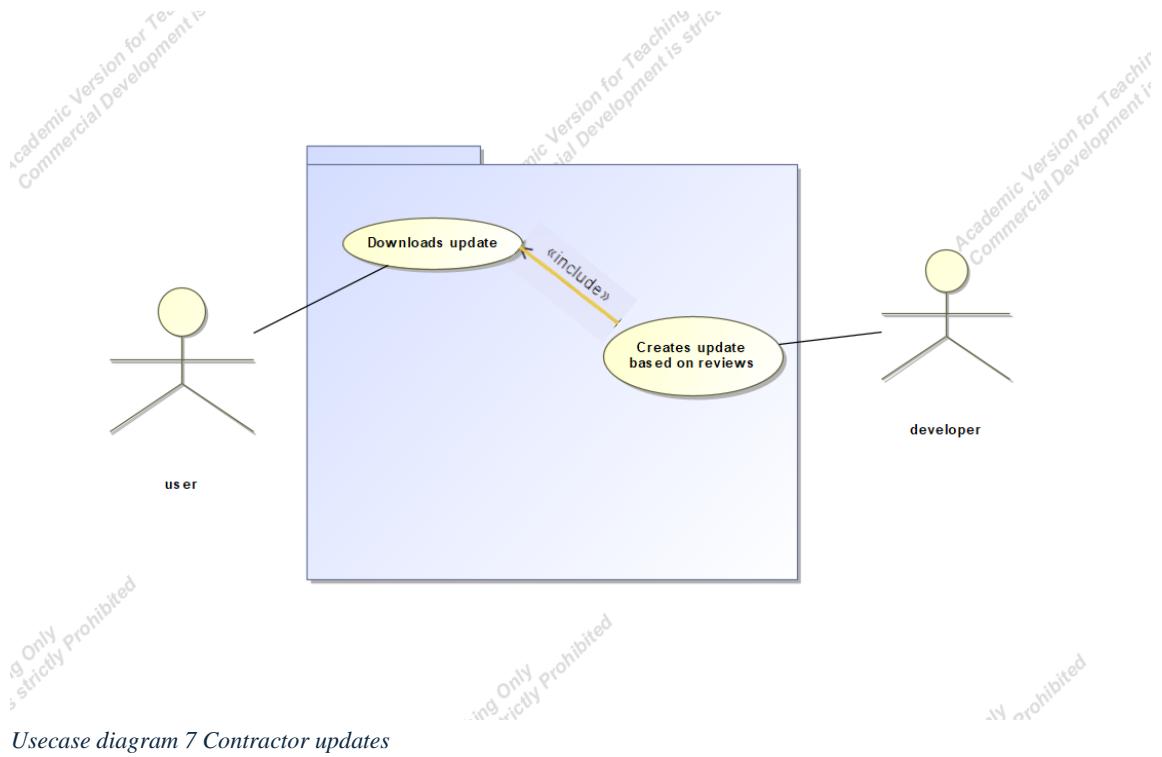
Automatic updates from contractor for the Dashboard to improve user satisfaction.

Software Engineering Analysis

### *Snowcard 8 Contractor updates*

Name	Contractor updates								
ID	3								
Description	Contractor supplies updates for a better user experience								
Trigger									
Actors	Contractor								
Pre-conditions									
Post-conditions									
Basic Flow	<table border="1"> <tr> <td>Description</td><td>Contractor sends notification to user to update the software</td></tr> <tr> <td>Actions</td><td></td></tr> <tr> <td>1</td><td>User gets update notification</td></tr> <tr> <td>2</td><td>User decides if he wants to do the update or decline</td></tr> </table>	Description	Contractor sends notification to user to update the software	Actions		1	User gets update notification	2	User decides if he wants to do the update or decline
Description	Contractor sends notification to user to update the software								
Actions									
1	User gets update notification								
2	User decides if he wants to do the update or decline								

*Description of Usecase 7 Contractor updates*



Usecase diagram 7 Contractor updates

## 8.4. Heartrate Graph Widget

The "Heart rate graph" requirement pertains to the functionality of displaying a graphical representation of a user's heart rate data over a period of time. This feature allows users to visualize their heart rate trends, monitor their cardiovascular health, and track changes in their heart rate patterns. By presenting the data in a graph format, users can easily interpret their heart rate variations and identify any anomalies or irregularities. This widget is an extension of the app Heart rate Monitor.

## #Heart rate Graph

<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	customer
<b>User Satisfaction:</b>	high

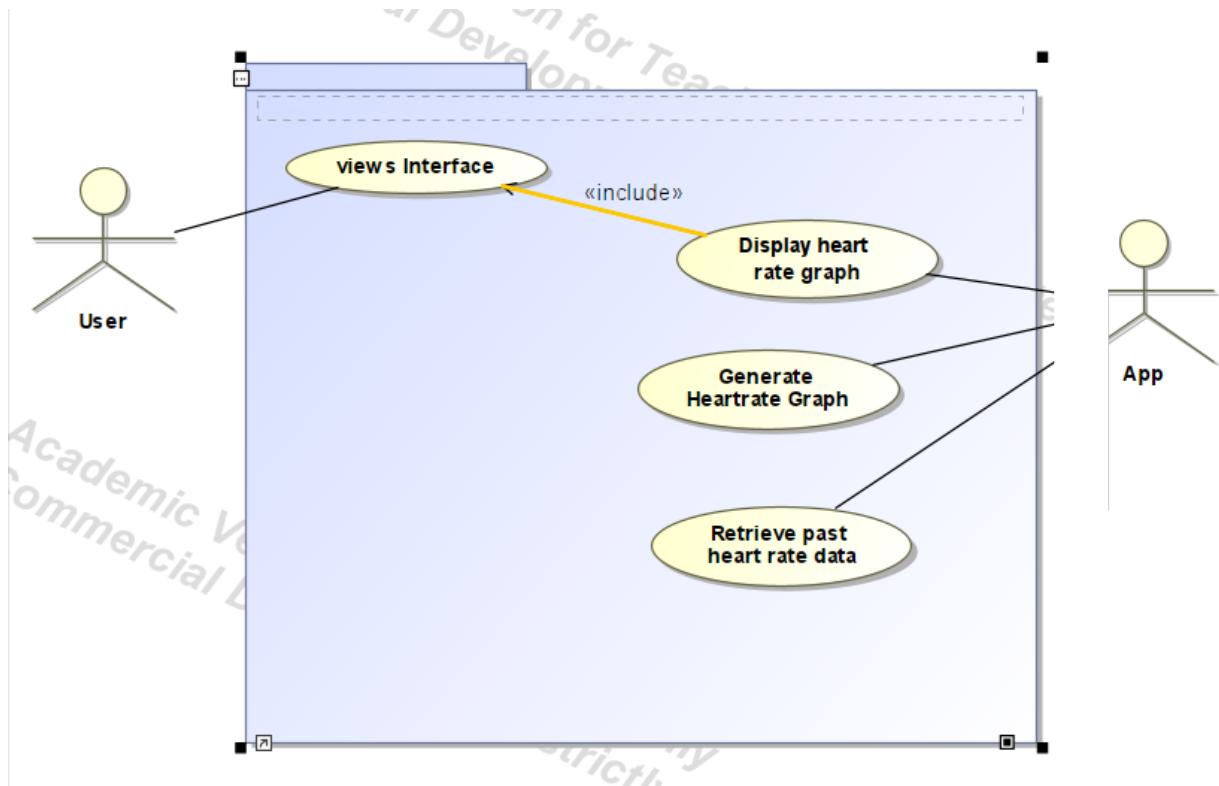
### Description:

This is a widget which the user can select to be constantly displayed on his Dashboard. It collects Heart rate data which was previously collected by the heart rate monitor in form of a graph.

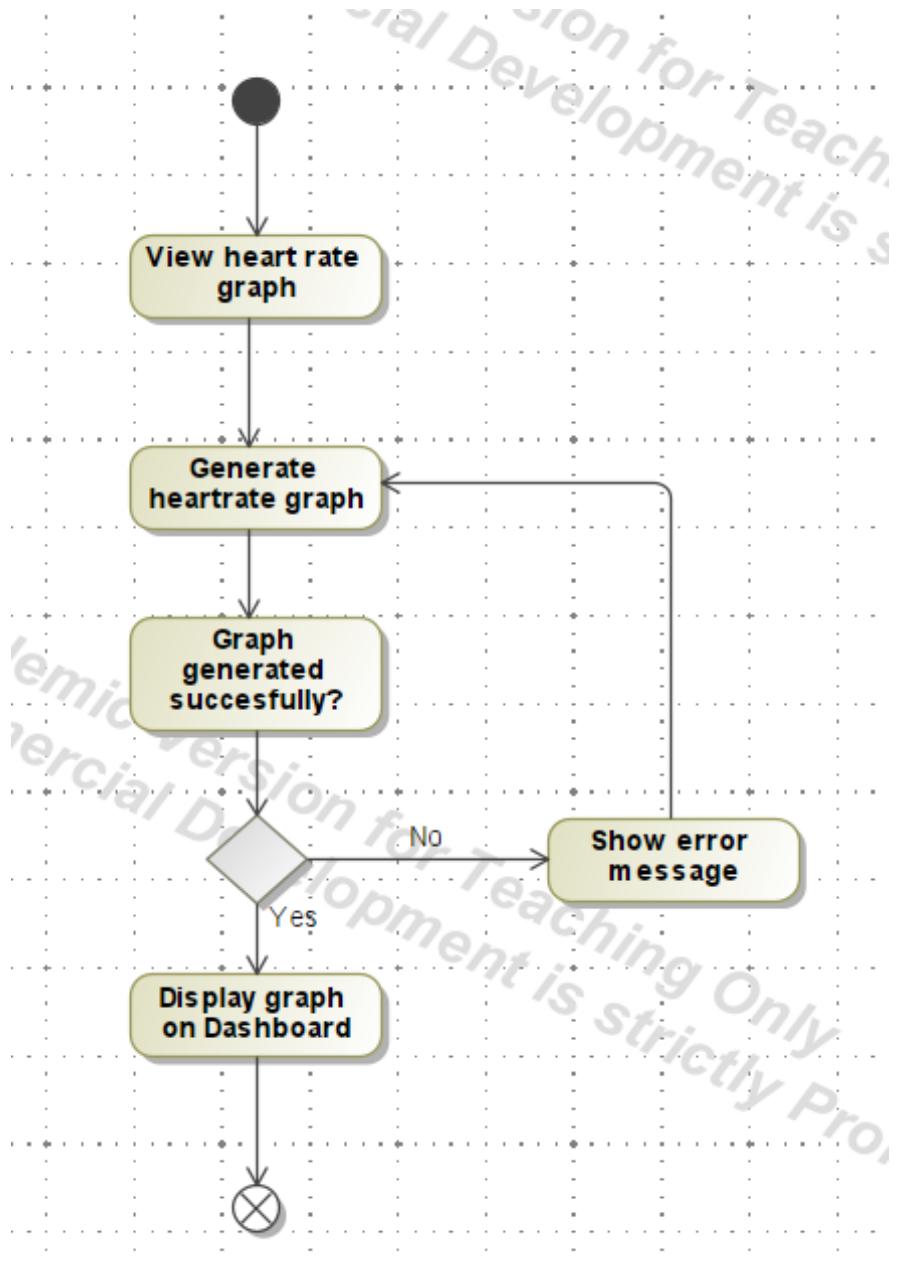
*Snowcard 9 Heart rate graph*

<b>Name</b>	Heartrate history graph
<b>ID</b>	4
<b>Description</b>	This is a Dashboard widget which constantly presents the Heartrate history over a certain period of time
<b>Trigger</b>	User views dashboard
<b>Actors</b>	user
<b>Pre-conditions</b>	A implemented Heartrate detection feature and a data base that collects the data.
<b>Post-conditions</b>	
<b>Basic Flow</b>	
<b>Description</b>	The widget constantly displays heartrate progression over a certain period of time
<b>Actions</b>	
<b>1</b>	User selects widget to be displayed on dashboard
<b>2</b>	Real-time data about heartrate history is gathered and displayed

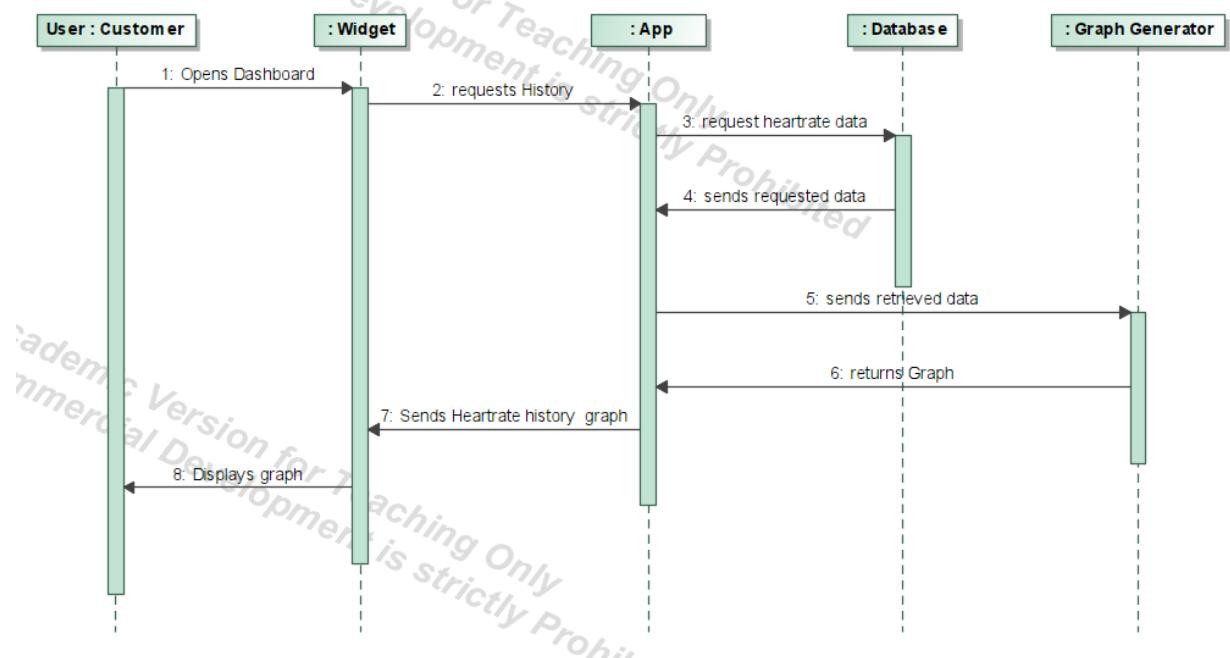
*Description of Usecase 8 Heart rate graph*



Use case diagram 8 Heart rate graph



Activity diagram 4 Heart rate graph



Sequence diagram 5 Heart rate graph

## 8.5. Step count Widget

A step count graph widget is a visual element typically found in health and fitness applications or smartwatch dashboards. It displays a graphical representation of a user's step count data over a specific period, such as a day, week, or month. The widget visually depicts the user's physical activity level, showcasing trends and patterns in their step count over time. This allows users to track their daily activity progress, set fitness goals, and monitor their overall physical activity levels.

## #Step count graph

**Requirement Type:** non-functional

**For Whom?** customer

**User Satisfaction:** high

### Description:

This is a widget which the user can select to be constantly displayed on his Dashboard.

It collects step count data which was previously collected by the step count application in form of a graph.

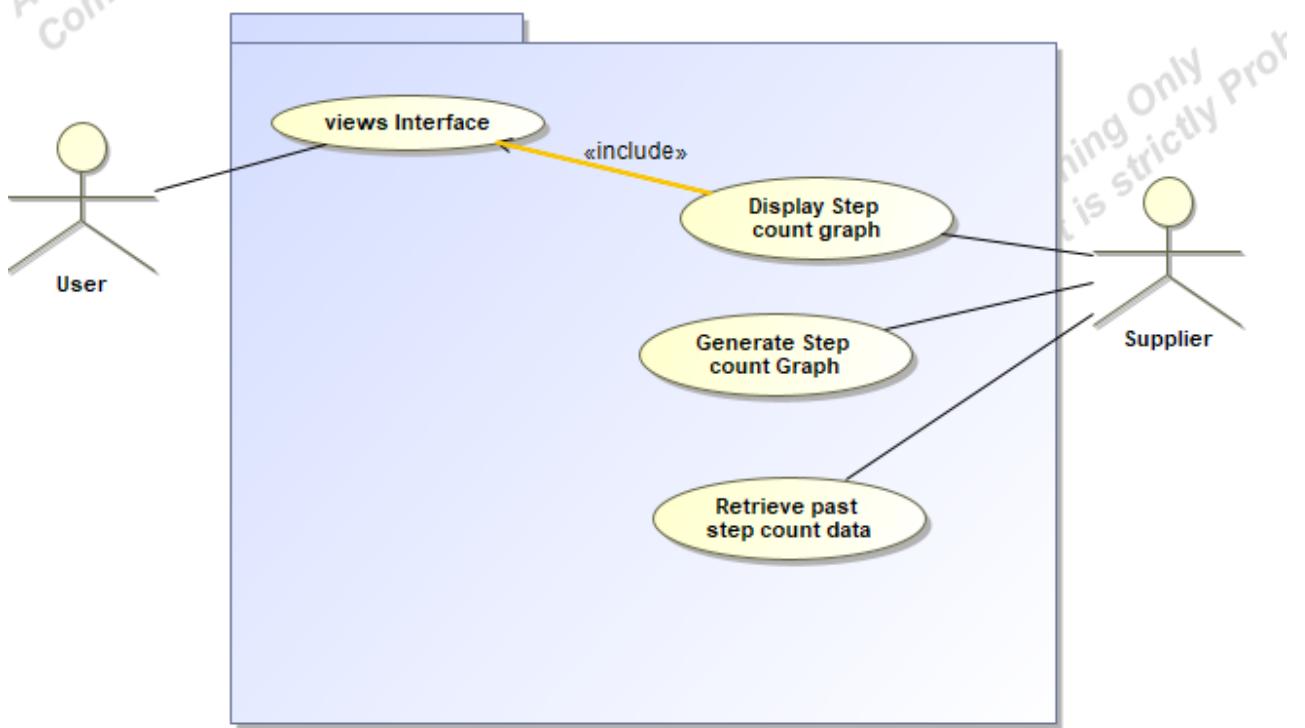
Software Engineering Analysis

*Snowcard 10 Step count graph*

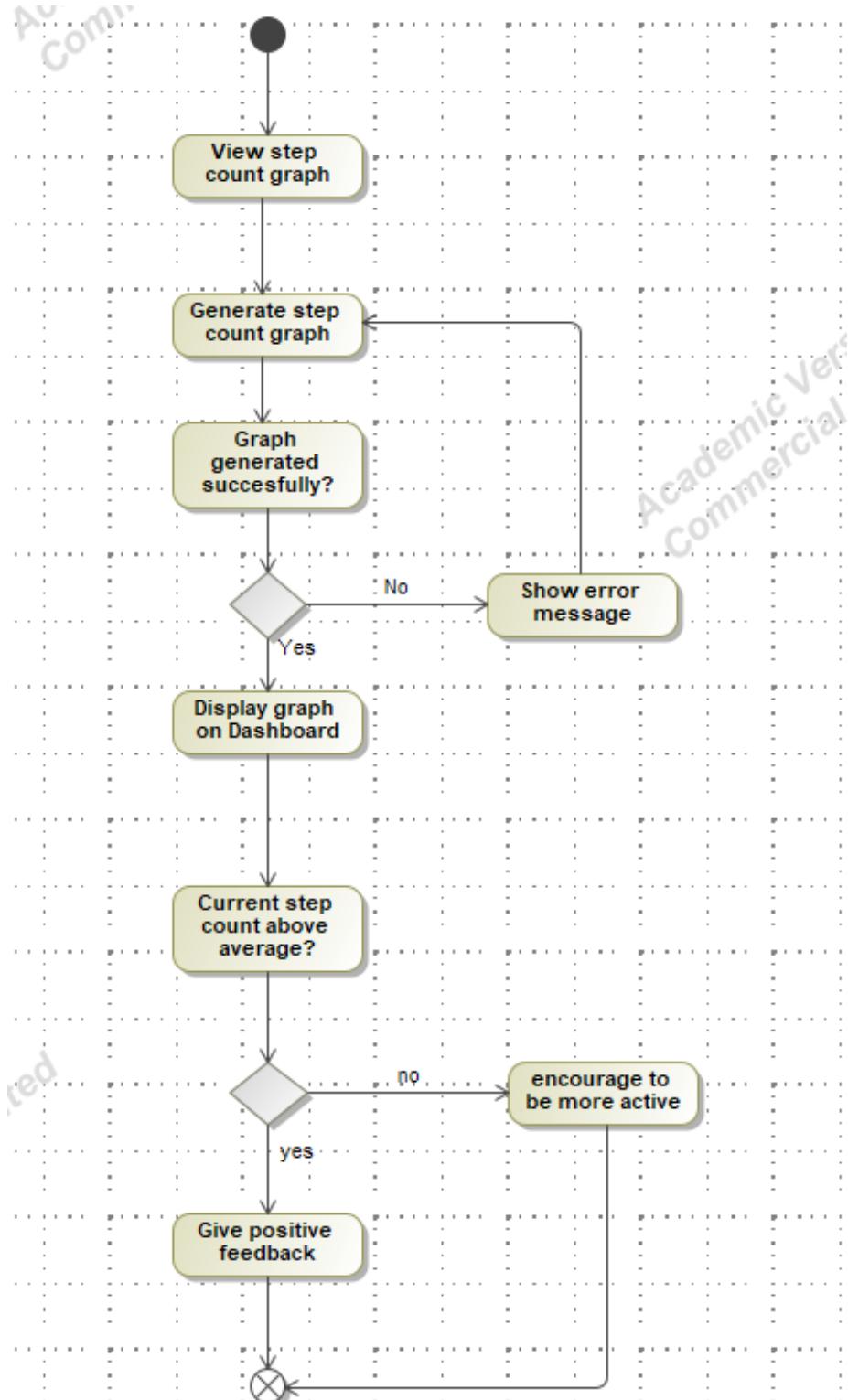


<b>Name</b>	Heartrate history graph
<b>ID</b>	4
<b>Description</b>	This is a Dashboard widget which constantly presents the steps taken over a certain period of time
<b>Trigger</b>	User views dashboard
<b>Actors</b>	user
<b>Pre-conditions</b>	A implemented Step detection feature and a data base that collects the data.
<b>Post-conditions</b>	
<b>Basic Flow</b>	
<b>Description</b>	The widget constantly displays step count progression over a certain period of time
<b>Actions</b>	
<b>1</b>	User selects widget to be displayed on dashboard
<b>2</b>	Real-time data about heartrate history is gathered and displayed

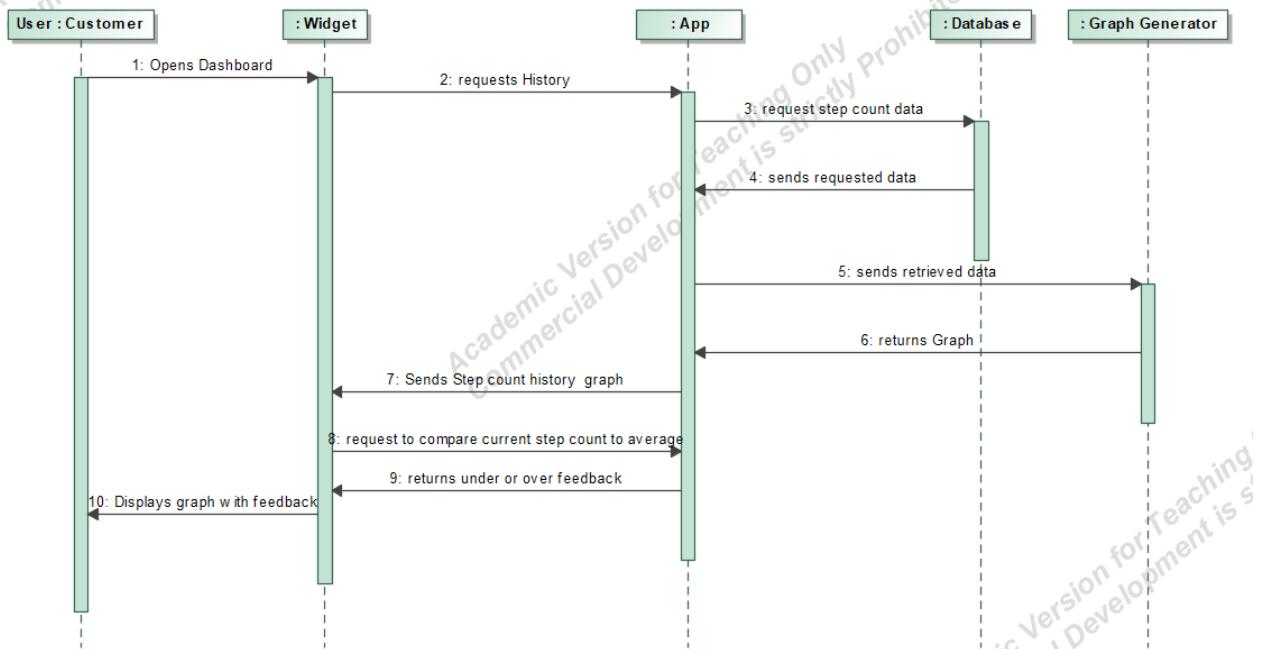
*Description of Usecase 9 Step count graph*



Usecase diagram 9 Step count graph



Activity diagram 5 Step count graph



Sequence diagram 6 Step count graph

## 9. Heart rate

Heart Rate is one of the core functionalities of Posture Piece, which measures the user's heart rate in real-time, calculates personalized health heart rate ranges based on the user's input personal information, and sends alert notifications to the smartwatch if the user's heart rate exceeds the specified range. This feature goes beyond the conventional one-dimensional information provision of existing smartwatches, which merely display measurement results, by building personalized data for users based on measurement data and providing real-time alert services. This provides a user-friendly feeling of the application continually accompanying the user. Furthermore, it offers an essential feature that helps users by providing weekly reports and future expert feedback.

### 9.1. Check Heart Rate

This is the first step in the functionality. When you click on the part of the smartwatch's main screen where the heart rate is displayed, it transitions to the next screen where the user can check their own heart rate and average heart rate.

#### #1: Start checking heart rate

**Requirement Type:** Functional

**For Whom?** User

**User Satisfaction:** Medium

**User Dissatisfaction:** Medium

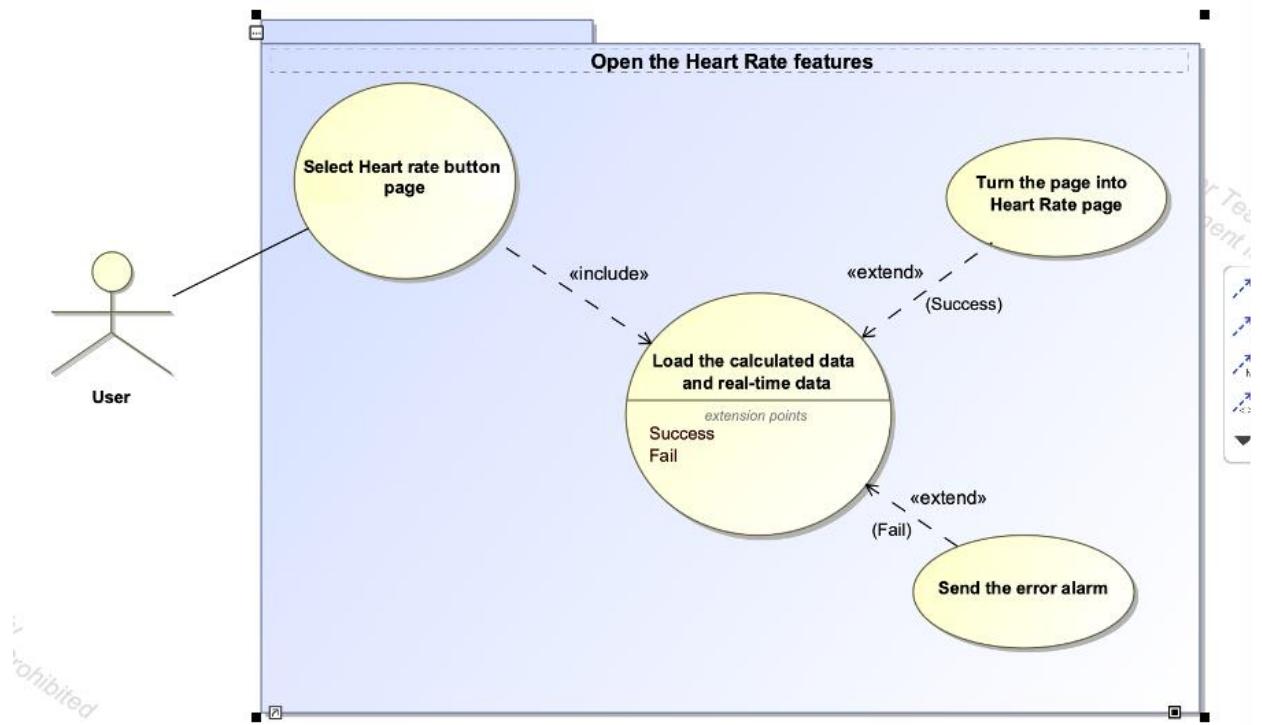
**Description:**

When the users click on the heart rate button on the main page, users will be directed to a page where you can check users' own heart rate and the average heart rate.

*Snowcard 11 Start checking heart rate*



Image 2 UI/UX Start checking heart rate



Usecase diagram 10 Start checking heart rate

<b>Name</b>	Activate the Heart rate features								
<b>ID</b>	9								
<b>Description</b>	When user clicked the main page's 'Heart rate' button, the page will call the 'Heart rate' page								
<b>Trigger</b>	Systems detected user's touched on the button								
<b>Actors</b>	Costumer, Contractor								
<b>Pre-conditions</b>	The main page is on the smart watch								
<b>Post-conditions</b>	Load the user's health data								
<b>Basic Flow</b>	<table border="1"> <tr> <td><b>Description</b></td><td>User click the main page's button</td></tr> <tr> <td><b>Actions</b></td><td></td></tr> <tr> <td><b>1</b></td><td>The user's touch send signal to system</td></tr> <tr> <td><b>2</b></td><td>System prepared for loading the user's data</td></tr> </table>	<b>Description</b>	User click the main page's button	<b>Actions</b>		<b>1</b>	The user's touch send signal to system	<b>2</b>	System prepared for loading the user's data
<b>Description</b>	User click the main page's button								
<b>Actions</b>									
<b>1</b>	The user's touch send signal to system								
<b>2</b>	System prepared for loading the user's data								
<b>Alternative Flow</b>	<table border="1"> <tr> <td><b>Description</b></td><td>A</td></tr> <tr> <td><b>Actions</b></td><td>If system load the user's data</td></tr> <tr> <td><b>1</b></td><td>Show the user's data on the screen</td></tr> <tr> <td><b>2</b></td><td>Enter to the Heart rage page</td></tr> </table>	<b>Description</b>	A	<b>Actions</b>	If system load the user's data	<b>1</b>	Show the user's data on the screen	<b>2</b>	Enter to the Heart rage page
<b>Description</b>	A								
<b>Actions</b>	If system load the user's data								
<b>1</b>	Show the user's data on the screen								
<b>2</b>	Enter to the Heart rage page								
<b>Alternative Flow</b>	<table border="1"> <tr> <td><b>Description</b></td><td>B</td></tr> <tr> <td><b>Actions</b></td><td>If system can not upload the user's data</td></tr> <tr> <td><b>1</b></td><td>Show the alarm to the screen 'Can't upload your health data'</td></tr> </table>	<b>Description</b>	B	<b>Actions</b>	If system can not upload the user's data	<b>1</b>	Show the alarm to the screen 'Can't upload your health data'		
<b>Description</b>	B								
<b>Actions</b>	If system can not upload the user's data								
<b>1</b>	Show the alarm to the screen 'Can't upload your health data'								

*Description of Usecase 10 Start checking heart rate*

## 9.2. Personalized Heart Rate Range

Based on the personal information entered by the user when first signing up for the app, we calculate the user's customized normal heart rate range. This serves as an important criterion for later measuring normal and abnormal heart rates based on real-time heart rate measurements.

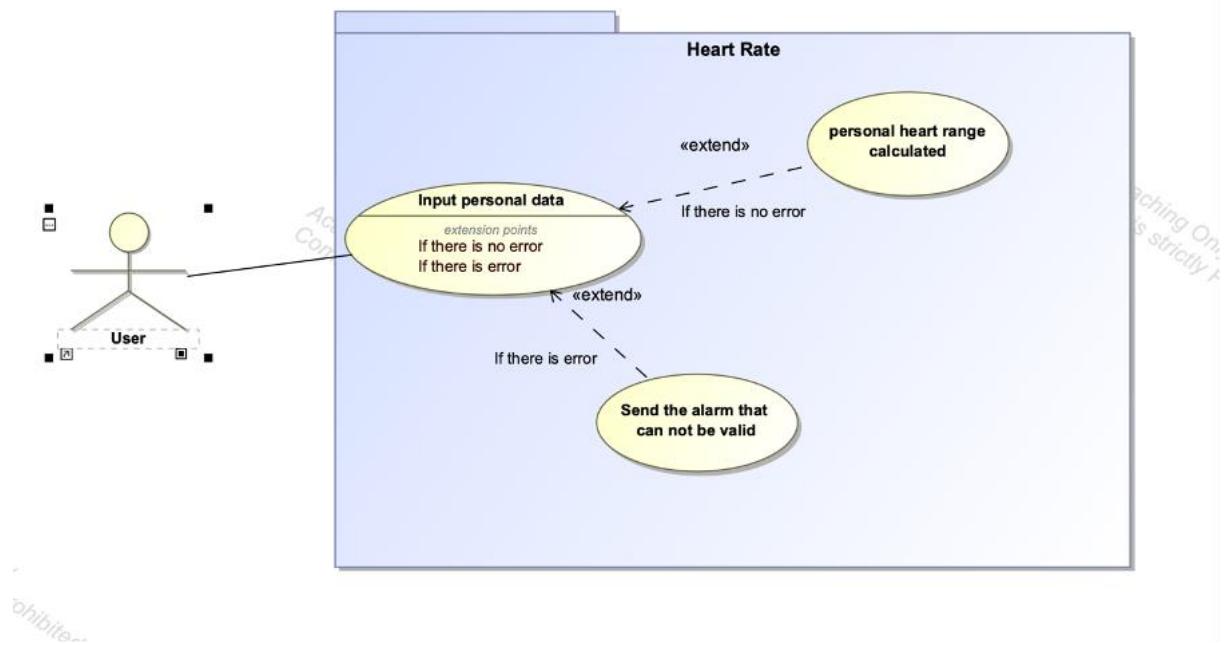
## #1: Standard of abnormal heart rate

<b>Requirement Type:</b>	Functional
<b>For Whom?</b>	Management
<b>User Satisfaction:</b>	medium
<b>User Dissatisfaction:</b>	medium

### Description:

- Based on the user input information (such as age, gender, etc.), establish criteria for determining whether the user's detected heart rate is abnormal or normal by calculating the average heart rate.

Snowcard 12 Personalised heart rate range



Use case diagram 11 Personalised heart rate range

<b>Name</b>	Personalized Heart rate page								
<b>ID</b>	9								
<b>Description</b>	Set the personal normal Heart rate range with user's basic information								
<b>Trigger</b>	When user wants to make user's own heart range								
<b>Actors</b>	Customer, System								
<b>Pre-conditions</b>	User input their basic information on the application								
<b>Post-conditions</b>	The result of personalised normal heart rate range comes out								
<b>Basic Flow</b>	<table border="0"> <tr> <td><b>Description</b></td><td>User input his/her information on the application</td></tr> <tr> <td><b>Actions</b></td><td></td></tr> <tr> <td>1</td><td>Server sent that information to the system</td></tr> <tr> <td>2</td><td>System calculated the normal heart range with that information</td></tr> </table>	<b>Description</b>	User input his/her information on the application	<b>Actions</b>		1	Server sent that information to the system	2	System calculated the normal heart range with that information
<b>Description</b>	User input his/her information on the application								
<b>Actions</b>									
1	Server sent that information to the system								
2	System calculated the normal heart range with that information								
<b>Alternative Flow</b>	<table border="0"> <tr> <td><b>Description</b></td><td>A</td></tr> <tr> <td><b>Actions</b></td><td>If information has the error</td></tr> <tr> <td>1</td><td>Send the alarm to user that informations are not valid</td></tr> </table>	<b>Description</b>	A	<b>Actions</b>	If information has the error	1	Send the alarm to user that informations are not valid		
<b>Description</b>	A								
<b>Actions</b>	If information has the error								
1	Send the alarm to user that informations are not valid								

*Description of Usecase 11 Personalised heart rate range*

### 9.3. Visualize Heart Rate

Based on the calculated data, the user's real-time heart rate and normal range are displayed on the smartwatch screen in numerical form, allowing the user to easily monitor them at a glance. While the real-time heart rate continuously changes, the user's normal heart rate remains constant unless their personal information is updated.

## #2: Visualize the heart rate

**Requirement Type:** Functional

**For Whom?** User

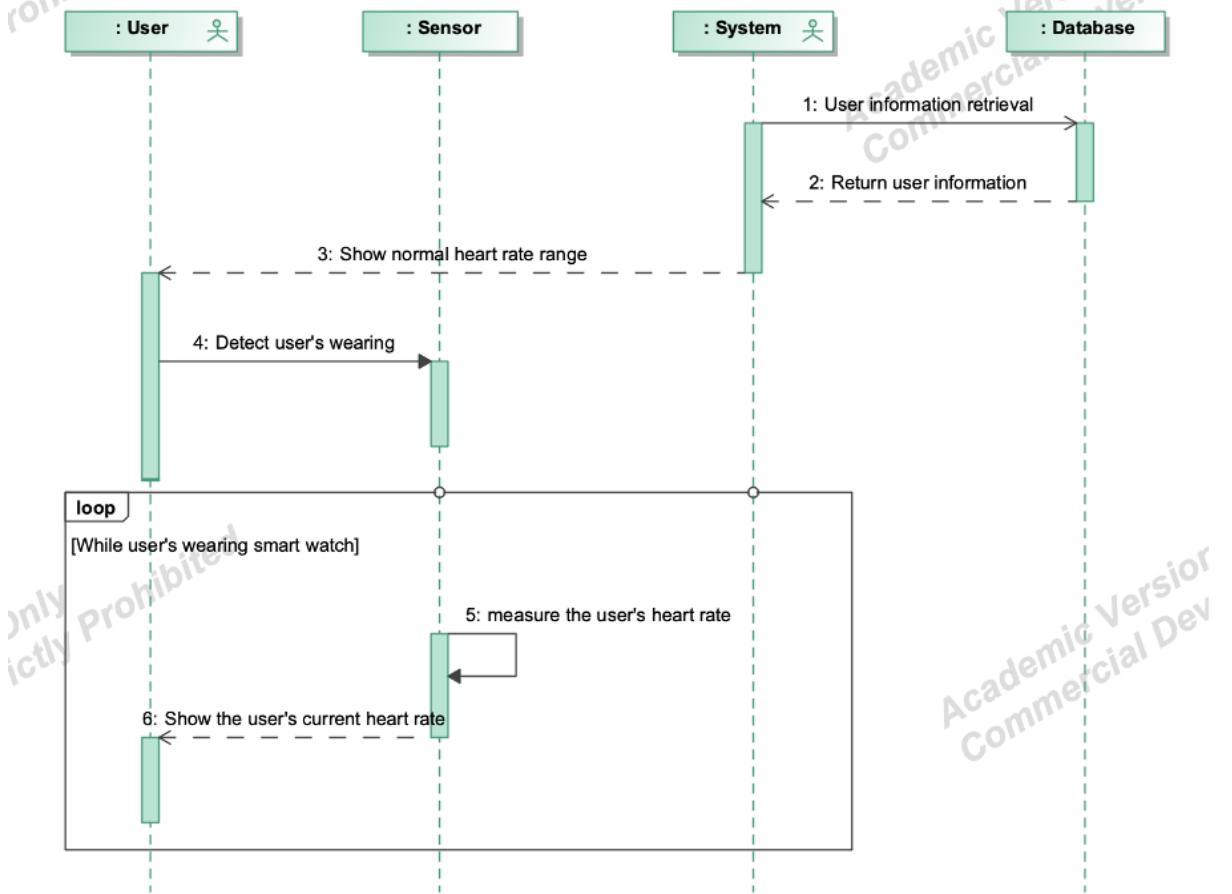
**User Satisfaction:** High

**User Dissatisfaction:** Low

**Description:**

The user can easily see their own heart rate and the average heart rate range displayed on the screen for quick reference. The numbers on the screen change in real-time as data is updated through sensors.

*Snowcard 13 Visualize the heart rate*



Sequence diagram 7 Visualize heart rate

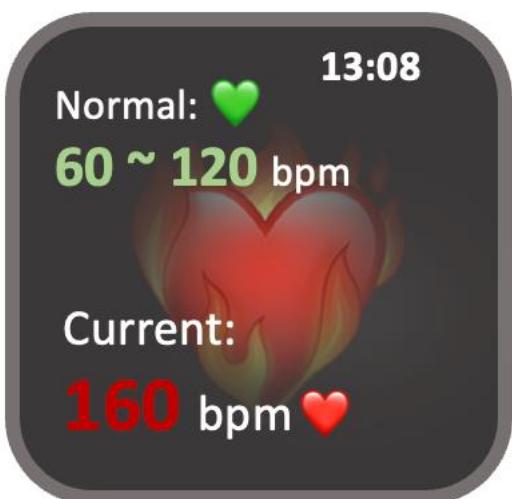


Image 3 UI/UX Visualize heart rate

## 9.4. Heart Rate real time alert

When the user's real-time heart rate measured in real-time deviates from their normal range, a warning push notification is triggered. This feature prompts the user to regularly check their health status and provides simple guidance to assist the user in making improvements. The alarm persists for up to 3 minutes, disappearing immediately once the user's heart rate returns to the normal range.



### #2: Heart Rate real time alert

**Requirement Type:** Functional

**For Whom?** Users

**User Satisfaction:** big

**User Dissatisfaction:** low

**Description:**

- If the user's heart rate data collected through the sensor falls below or exceeds the predetermined range, a real-time alert is triggered to alert the user consciously.
- This feature is only for Pro mode

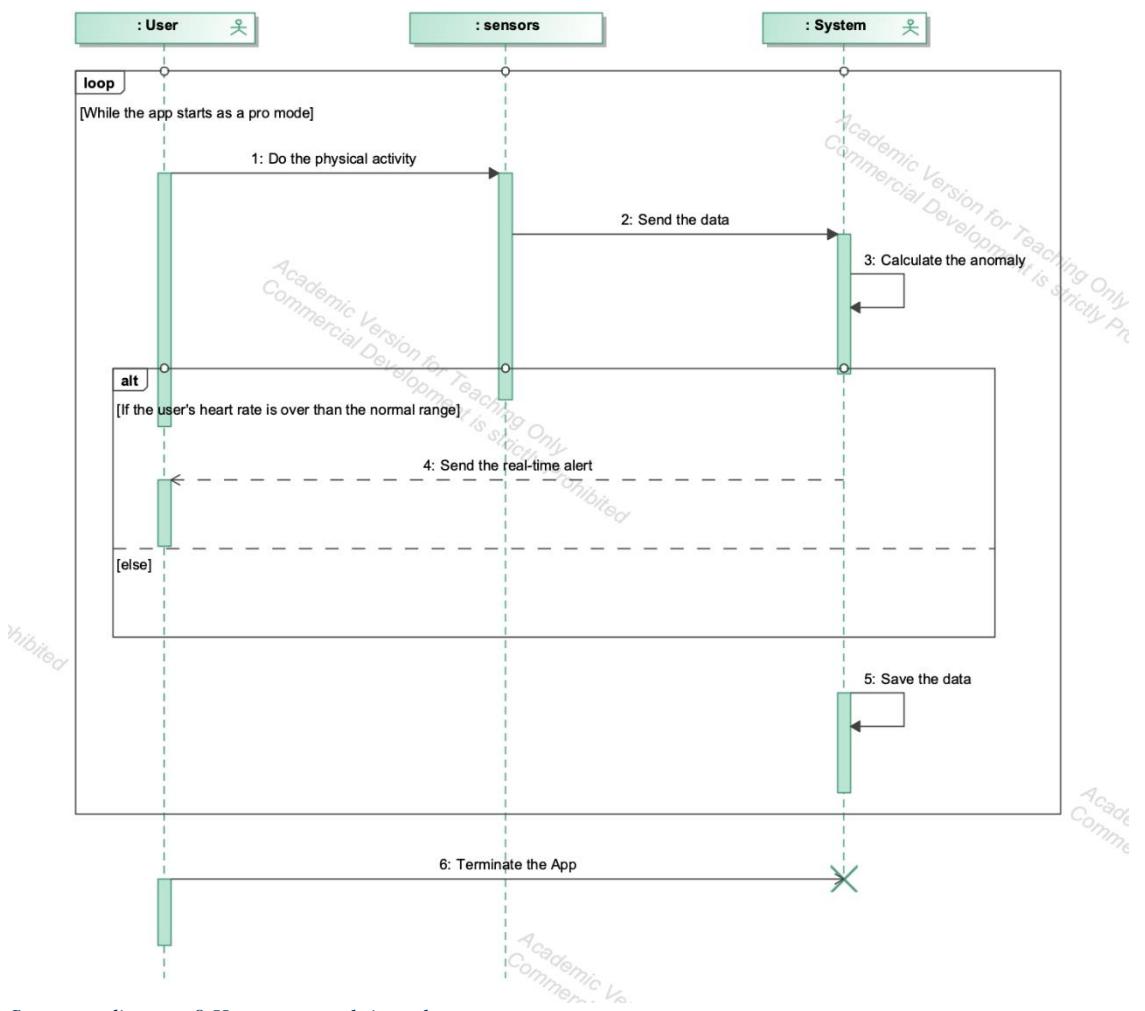
*Snowcard 14 Heart rate real time alert*



Usecase diagram 12 Heart rate real time alert

Name	Heart real time alert								
ID	9								
Description	The system sends real-time alert about user's heart statement to the smart watch								
Trigger	When user's real-time heart rate is over than the normal range								
Actors	Customer, System								
Pre-conditions	The user's normal heart range, as well as real-time monitoring of the user's heart rate, are being measured.								
Post-conditions	Warning alert came to the user's smart watch								
Basic Flow	<table border="1"> <thead> <tr> <th>Description</th> <th></th> </tr> </thead> <tbody> <tr> <td>Actions</td> <td></td></tr> <tr> <td>1</td> <td>The sensor detected the user's heart rate</td></tr> <tr> <td>2</td> <td>The system compares the user's real-time heart rate with the normal range.</td></tr> </tbody> </table>	Description		Actions		1	The sensor detected the user's heart rate	2	The system compares the user's real-time heart rate with the normal range.
Description									
Actions									
1	The sensor detected the user's heart rate								
2	The system compares the user's real-time heart rate with the normal range.								
Alternative Flow	<table border="1"> <thead> <tr> <th>Description</th> <th></th> </tr> </thead> <tbody> <tr> <td>Actions</td> <td></td></tr> <tr> <td>1</td> <td>If the user's heart rate is in normal range Keep detecting the user's heart rate</td></tr> </tbody> </table>	Description		Actions		1	If the user's heart rate is in normal range Keep detecting the user's heart rate		
Description									
Actions									
1	If the user's heart rate is in normal range Keep detecting the user's heart rate								
Alternative Flow	<table border="1"> <thead> <tr> <th>Description</th> <th></th> </tr> </thead> <tbody> <tr> <td>Actions</td> <td></td></tr> <tr> <td>1</td> <td>If the user's heart rate in out normal range Send the warning alert to the user's smart watch</td></tr> </tbody> </table>	Description		Actions		1	If the user's heart rate in out normal range Send the warning alert to the user's smart watch		
Description									
Actions									
1	If the user's heart rate in out normal range Send the warning alert to the user's smart watch								

Description of Usecase 12 Heart rate real time alert



Sequence diagram 8 Heart rate real time alert



Image 4 UI/UX Heart rate real time alert

## 9.5. Save the heart rate data

All measured data and alerts are stored for later connection to the dashboard, enabling the generation of weekly graphs and reports. This information becomes crucial for obtaining feedback from experts. Heart rate data can be measured in real-time, and the heart rate report, in terms of graphs on a weekly basis, can be viewed on the smartwatch dashboard, while weekly reports and feedback from experts can be accessed on the user's smartphone.

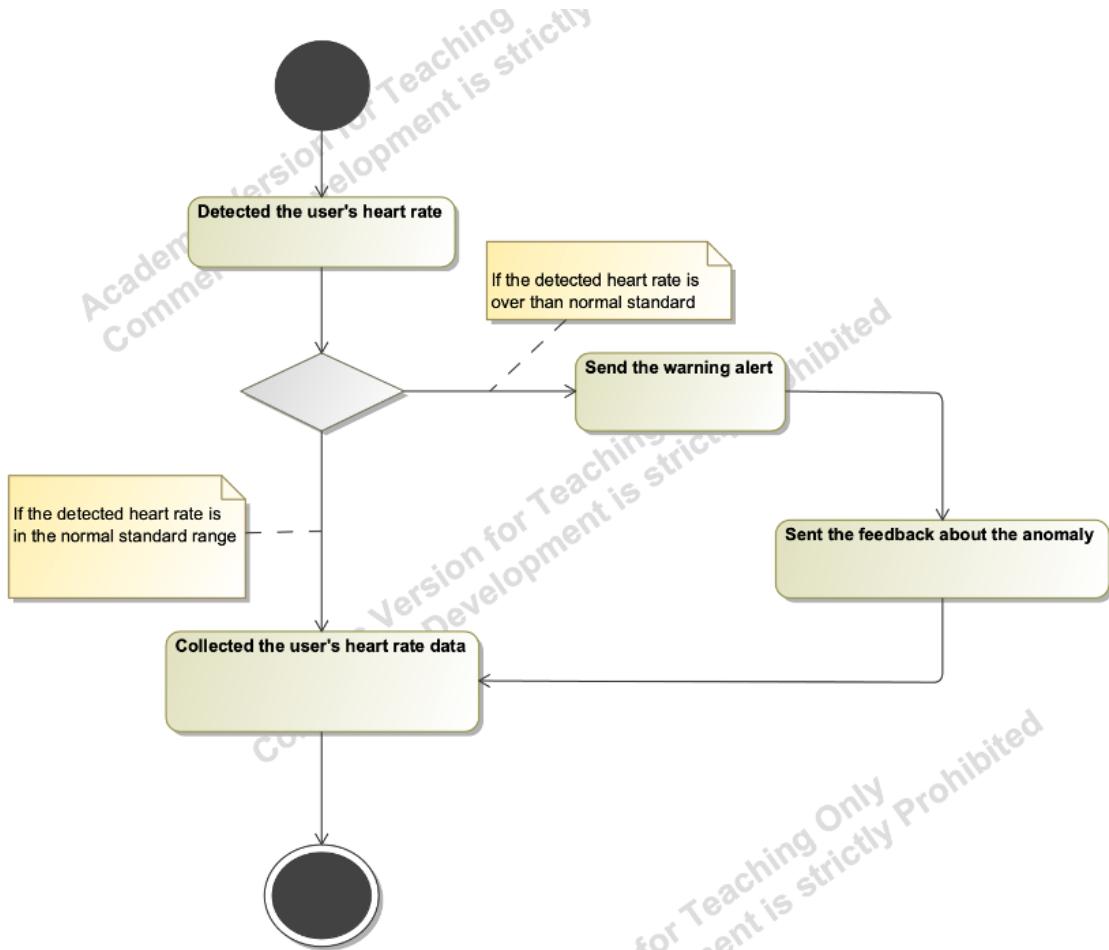


### #3: Save heart rate data

<b>Requirement Type:</b>	Functional
<b>For Whom?</b>	User, Management
<b>User Satisfaction:</b>	Medium
<b>User Dissatisfaction:</b>	Medium
<b>Description:</b>	

- Store the data of compared users against the benchmark range to enable viewing the accumulated heart rate graph at a glance later on

*Snowcard 5 Save heart rate data*



Activity diagram 1 Save heart rate data

# **10. Steps**

Steps collects data by tracking the user's walking behavior while wearing the watch, enabling diagnostics based on this information. It not only gathers step counts but also captures characteristic data such as stride length, speed, and distance traveled during walking.

When the user switches the watch to exercise mode, the data collection becomes even more precise. With accurate data, it becomes possible to detect stride patterns effectively and even detect instances of stumbling, thereby aiding in the cultivation of healthy exercise habits.

## **10.1. Workout detection**

Detecting when the user is exercising is a crucial step in data collection. This allows for more precise data collection and serves as an indicator of how often the user engages in walking exercises daily.

When users start walking for exercise purposes, there are two options. If walking for exercise, users can manually change the watch's status to "Exercise" before walking. This is considered a manual change. On the other hand, if users walk for more than 10 minutes for exercise purposes, the watch detects it first. When prompted, selecting "Yes" changes the user's status to "Exercise." Choosing "No" maintains the user's previous status.

## Steps

### #01.Workout detection

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### Description:

If the user starts walking for the purpose, there are 2 options.

1) Manual Change

Change the user's current status to <Exercise> using the watch directly before walking

2) Automatic Change

When walking for more than 10 minutes, the watch detects this. After that, a notification is posted to the user to see if the user is exercising.

YES : Change user's status to <Exercise>

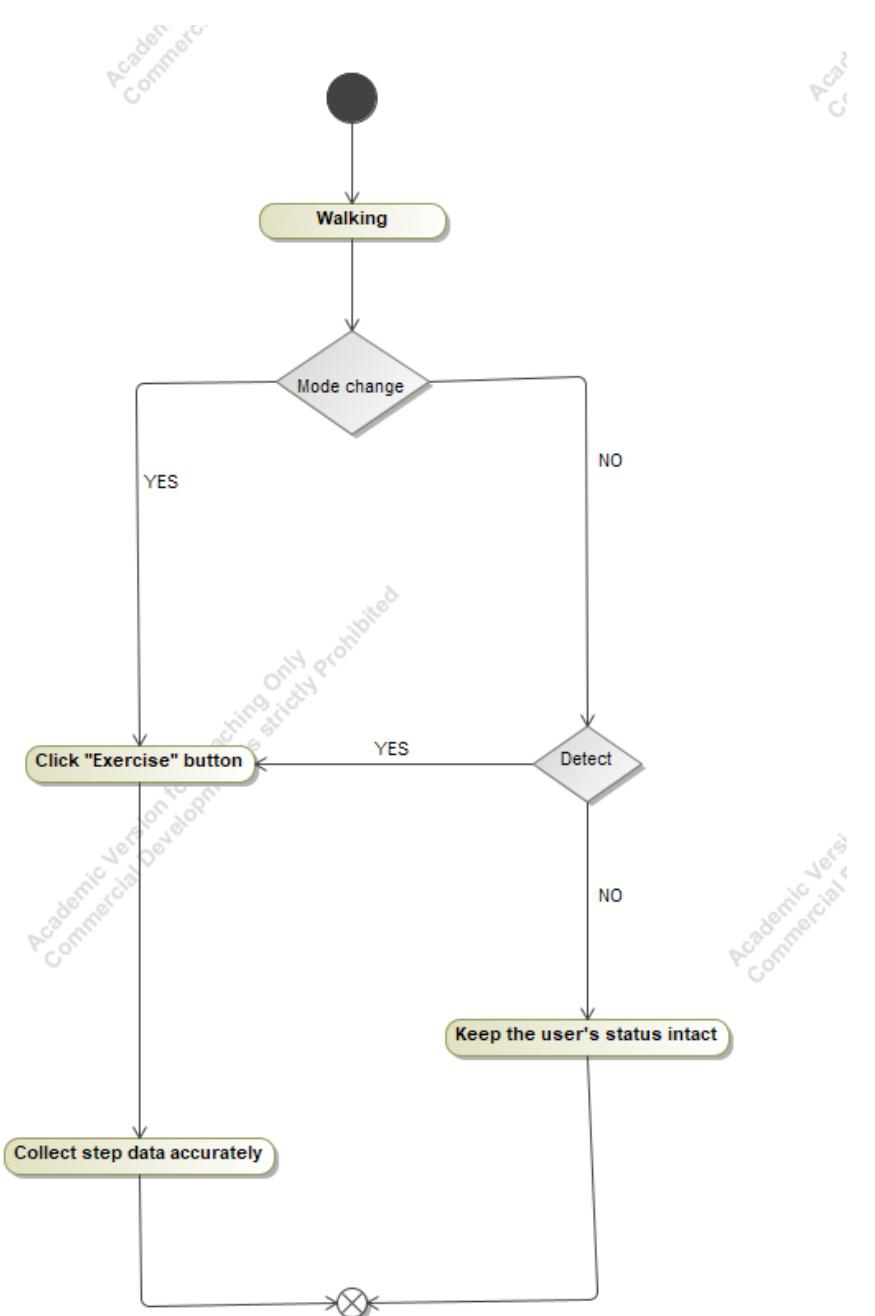
NO : Keep the user's status intact => <Just walking>

Seite 3

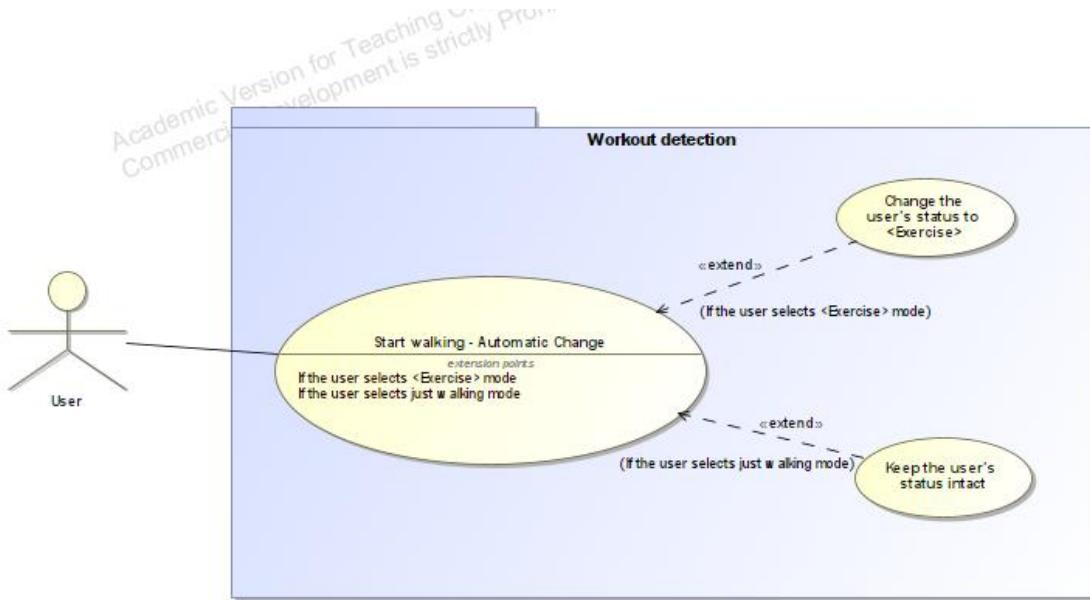
#### Snowcard 15 Workout detection

<b>Name</b>	Workout detection
<b>ID</b>	11.1
<b>Description</b>	An indicator of how often the user engages in walking exercises daily
<b>Trigger</b>	The user starts walking
<b>Actors</b>	Customer
<b>Pre-conditions</b>	The user wears the smartwatch
<b>Post-conditions</b>	This allows for more precise data collection
<b>Basic Flow</b>	
<b>Description</b>	The user's steps are detected while the user is walking for exercise purposes.
<b>Actions</b>	
1	The <Exercise> button is pressed.
2	The status is changed to <Exercise> mode.
3	It can collect more detailed step data compared to regular walking.
<b>Alternative Flow</b>	A
<b>Description</b>	The user's steps are detected while the user is just walking for more than 10 minutes.
<b>Actions</b>	
1	A notification is posted to the user to see if the user is exercising.
2	The <Exercise> button is pressed.
3	The status is changed to <Exercise> mode.
4	It can collect more detailed step data compared to regular walking.
<b>Alternative Flow</b>	B
<b>Description</b>	The user's steps are detected while the user is just walking for more than 10 minutes.
<b>Actions</b>	
1	A notification is posted to the user to see if the user is exercising.
2	The <Exercise> button is not pressed.
3	The watch status is not changed.

#### Description of Usecase 13 Workout detection



Activity diagram 6 Workout detection



*Usecase diagram 13 Workout detection*

## 10.2. Stroll detection

Stroll detection assists users in walking by helping them achieve an appropriate stride. In this context, the appropriate stride is determined by subtracting 100 from the user's height. If there is a significant difference from this value, guidance is provided to the user on the correct walking technique.

## Steps

### #02.Stroll detection

**Requirement Type:** functional

**For Whom?** contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### Description:

Guide the user to have a suitable stride.

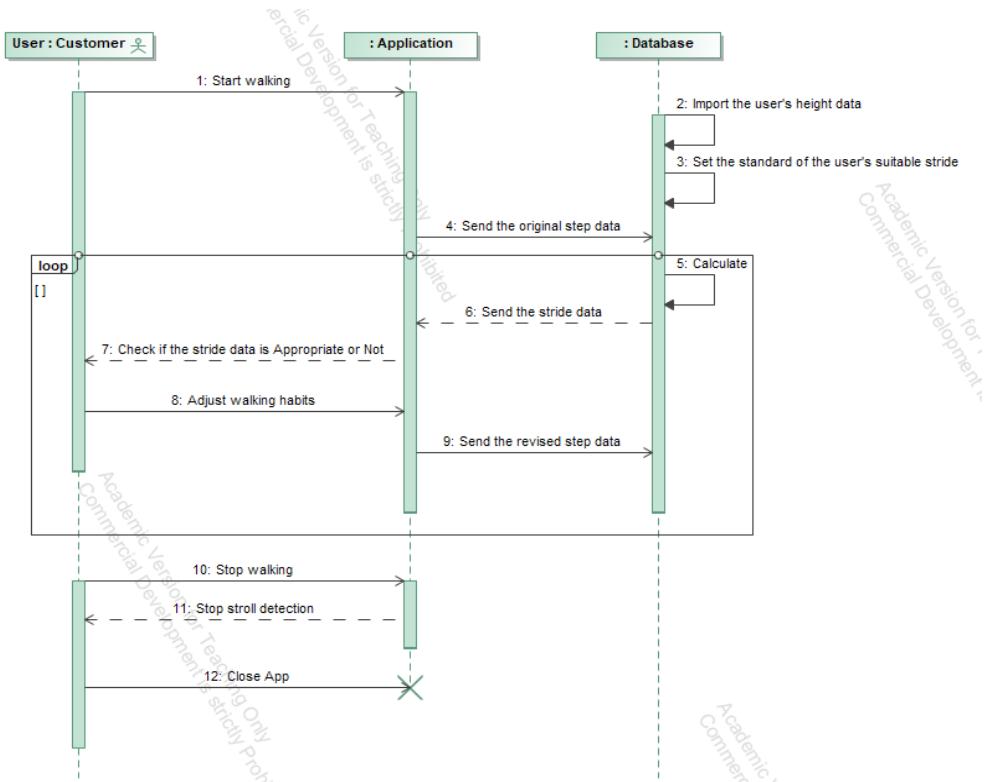
Shows the user's step state so that the user can walk correctly. if there is a significant difference in value based on Suitable stride : **Suitable stride = the user's height - 100**.

Thereafter, the user's individual step feature data may be collected.

Seite 4



*Snowcard 16 Stroll detection*



*Sequence diagram 9 Stroll detection*

## 10.3. Fall detection

Fall detection is implemented based on real-time monitoring of the watch's movements. This involves monitoring the acceleration and angle detected by the watch's sensors, and activating when they deviate from the normal range. The criteria for normal range include consistent speed and patterns, excluding outlier values. When a user's falling motion is detected, the watch provides notifications to the user through vibration and alerts.



### Steps

#### #03. Fall detection

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

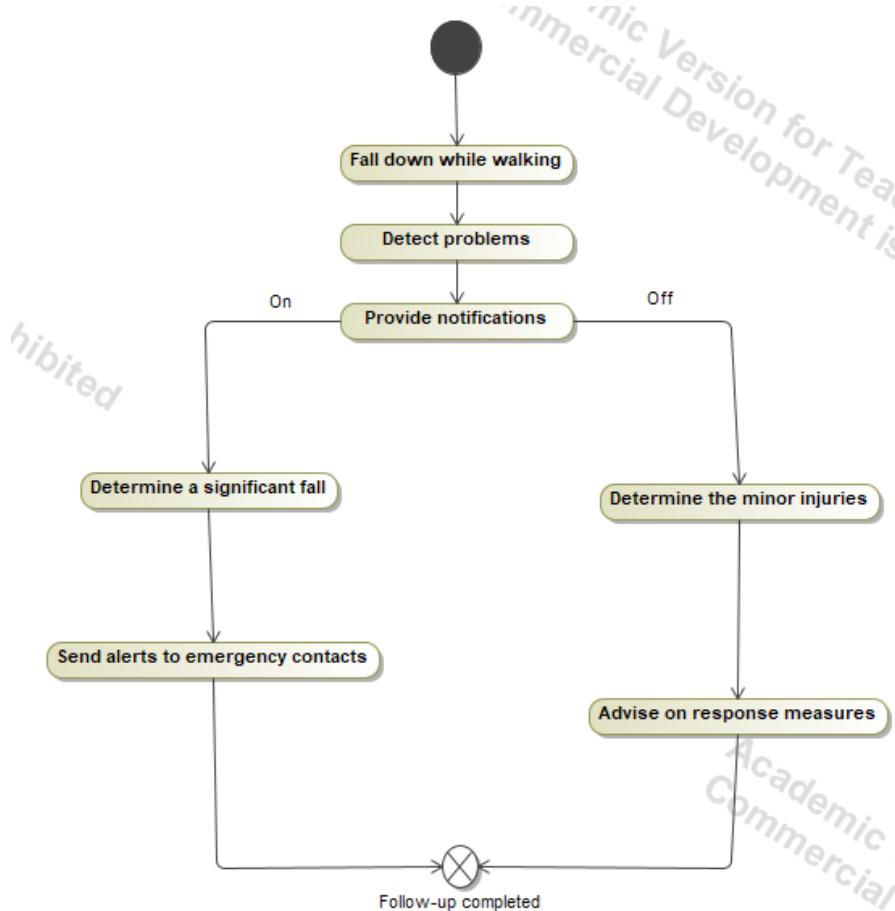
**User Dissatisfaction** low / medium / high

#### Description:

- The smartwatch monitors in real-time through the movement of the watch if the acceleration or angle detected by the watch's sensor deviates from the normal range.
- The watch detects the user's falling motion
  - ON : It determines a significant fall and sends alerts to emergency contacts.
  - OFF : It alerts the user of minor injuries or advises on response measures.

Seite 5

*Snowcard 17 Fall detection*



Activity diagram 7 Fall detection

## 10.4. Save the daily user's walking steps data

The user's daily step data can be collected when wearing smartwatch. The smartwatch walking sensor gathers acceleration and other sensor data to measure the user's step count. This data can be utilized to quantify the user's activity by calculating characteristics such as distance traveled and speed.

When the collected daily step count reaches the user's personal threshold, congratulatory notifications are displayed to provide motivation for exercise. However, this "threshold" may refer to either the user's average step count from the previous week or the activity goal set by the user. In cases where the user does not reach the goal, exercise is encouraged, and personalized support is offered.

## Steps

### #04. Save the daily user's walking steps data

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

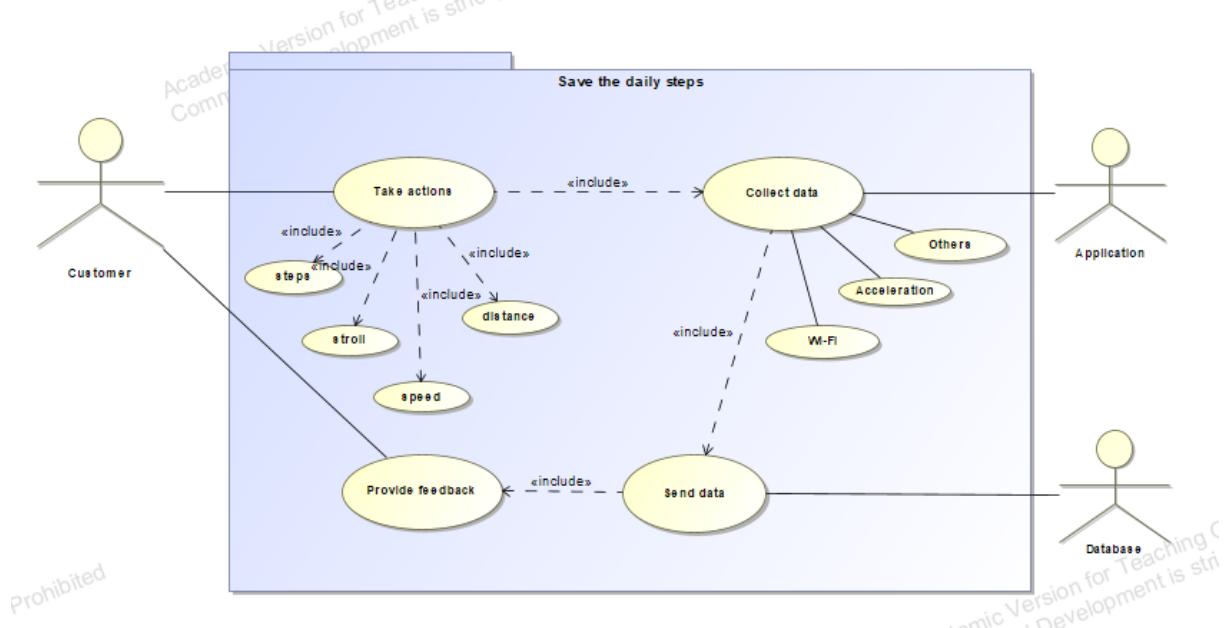
**User Dissatisfaction** low / medium / high

#### Description:

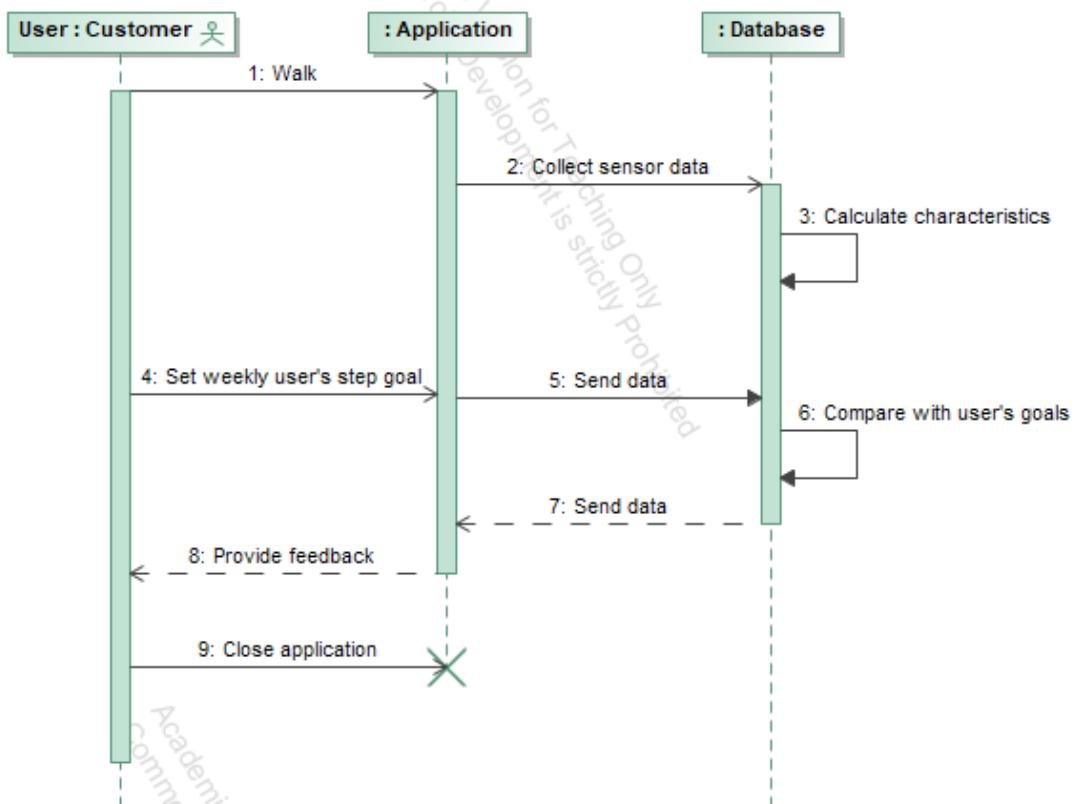
- The smartwatch walking sensor collects acceleration and other sensor data to measure the number of steps of the user.  
=> Characteristics such as travel distance, speed, etc. are calculated to quantify user activity
- The number of daily steps reached a certain level for the user (per day compared to the average number of steps last week / target for custom activities)  
=> Provides personalized support such as congratulatory notifications and exercise recommendations

Seite 6

Snowcard 18 the daily step data



Use case diagram 14 Daily step data



Sequence diagram 10 Daily steps data

## 10.5. Set the standard of average of weekly steps

The average number of steps per week is calculated by dividing the total number of steps per day by the number of days in the week. This is then compared to the weekly step goal set by the user to analyze performance. Subsequently, notifications are provided to the user to support goal achievement.

Weekly average steps are automatically updated to provide the latest information, enabling users to maintain healthy activity habits in real-time.

## Steps

### #05. Set the standard of average of weekly steps

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

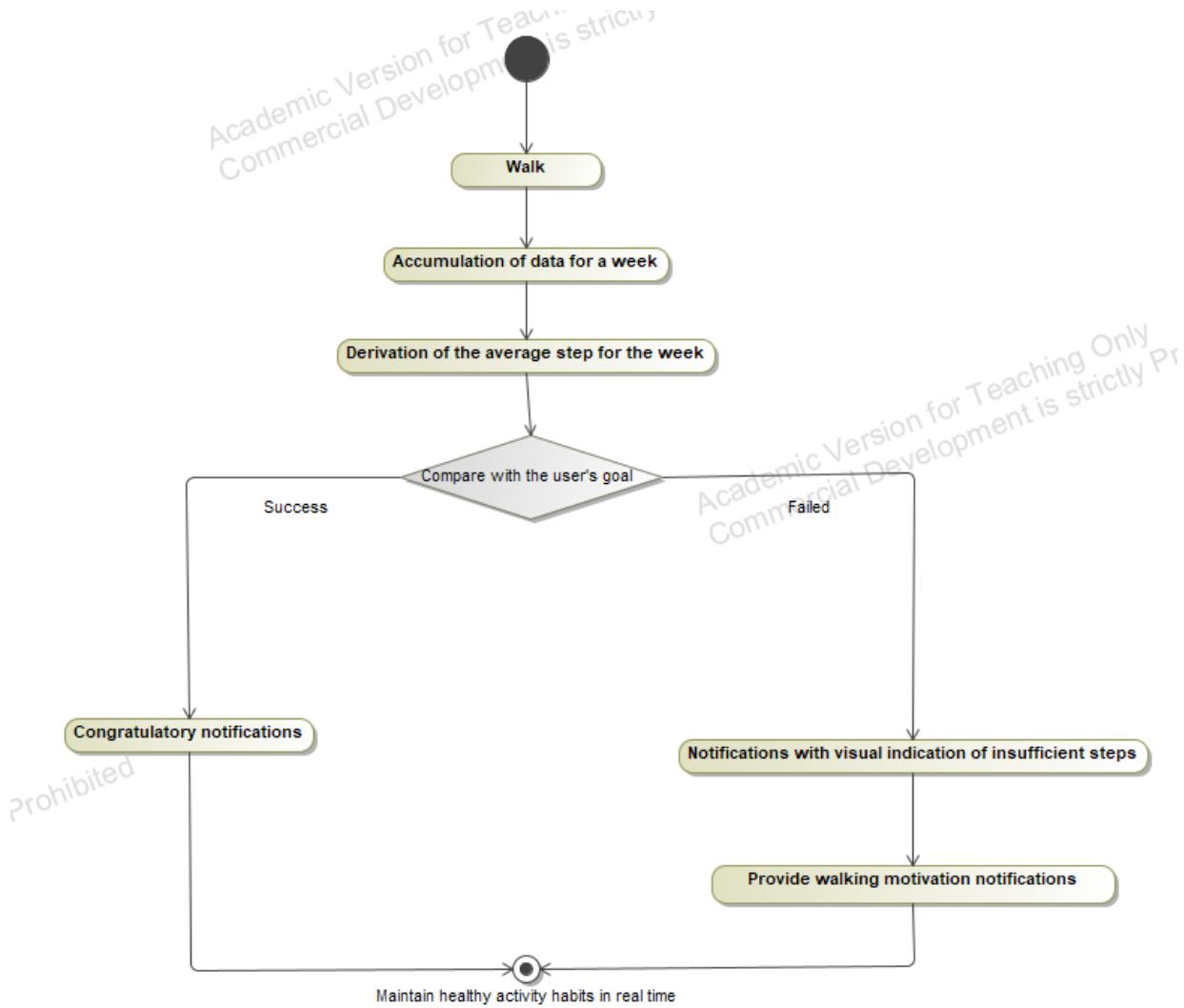
**User Dissatisfaction** low / medium / high

#### Description:

- The average number of steps per week is calculated by dividing the total number of steps per day by the number of days per week.
- Analyze user-set average weekly step goal versus the average number of steps per week.
  - Success : Provides "Signal of Achievement" with congratulatory notifications.
  - Failed : Promote notifications with visual identification of insufficient steps.
- Average weekly steps are automatically updated to keep users up to date

Seite 7

Snowcard 19 the average of weekly steps



Activity diagram 8 the average of weekly steps

## 10.6. Final weekly summary of steps

Steps



### #06.Final weekly summary of steps

**Requirement Type:** functional

**For Whom?** customer, contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### Description:

- Visual comparison of each step in a week using a bar graph
- Show the report of weekly user's steps count compared to last week's steps
- Summary of maximum, minimum and abnormal strides for the week

Seite 8

Snowcard 20 Weekly summary of steps



Below is the UI/UX corresponding to the step phase.



Image 5 Steps UIUX prototype

# 11. Turtleneck mode

Turtleneck mode is one of the key features emphasized in our software. As it relies on sensors in the wireless earphones to detect the tilt of the neck, this functionality is activated only when the user's wireless earphones are connected to the device. This functionality will be available only to customers who have subscribed to the Premium.

If the tilt of the user's neck detected by the sensor exceeds the threshold of 20 degrees, it is recognized as "turtleneck" posture, and a warning notification is sent along with a message prompting the user to straighten their neck. Additionally, the user can monitor their current neck tilt level in real-time on the smart watch screen.

## 11.1. Turtleneck mode button activation

Since Turtleneck can only be measured when wireless earphones are connected, the button will be inactive when the earphones are not connected. If the user attempts to press the inactivated button, the system displays the message that this is only available when earphones are connected.



### Turtleneck mode

#### #01. 'Turtleneck mode' button activation

**Requirement Type:** functional

**For Whom?** Customer

**User Satisfaction** low / medium / high

**User Dissatisfaction** low / medium / high

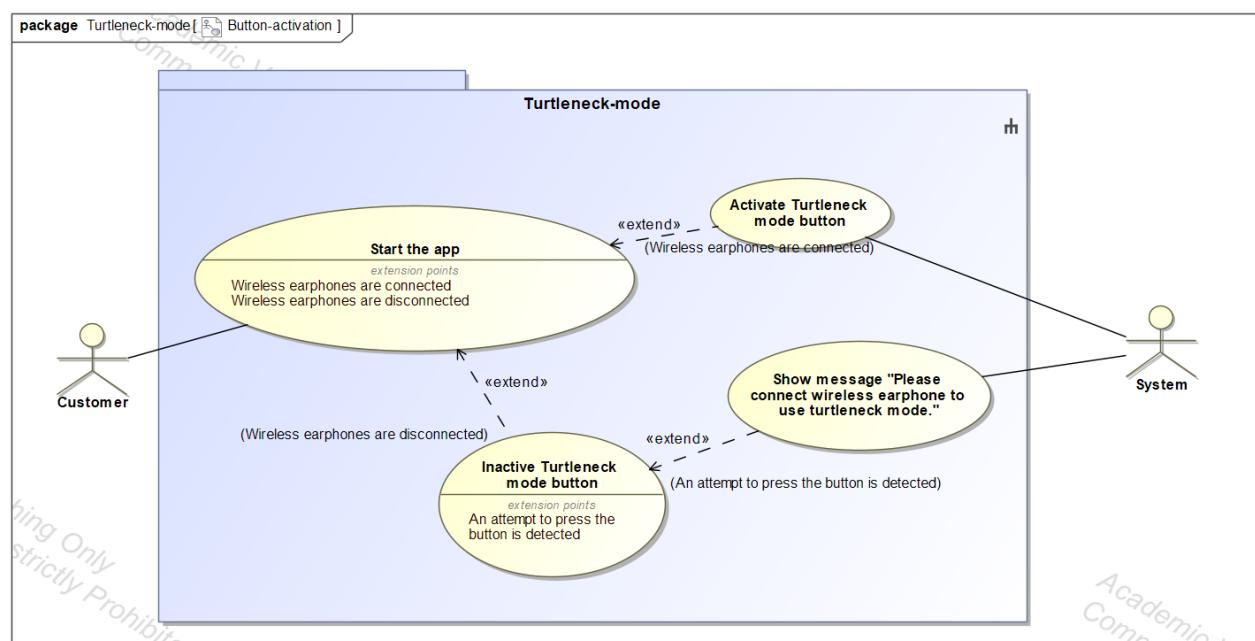
**Description:**

'Turtleneck mode' button is in the Main page.

1. If the wireless earphones are connected
  - Activate button
  
2. If the wireless earphones are disconnected
  - Inactivate button
  - If the customer try to press the button, show message  
"Connect wireless earphone to use Turtleneck mode"

Name		Turtleneck mode button activation
ID		
Description		The button is activated when the wireless earphones are connected to the device
Trigger		The customer want to use Turtleneck mode
Actors		System
Pre-conditions		The customer is subscribing Pro mode, and has own wireless earphones
Post-conditions		The customer can use Turtleneck mode
Basic Flow		
	Description	The customer start the app
	Actions	
	1	Check if the wireless earphones are connected
	2	
	3	
Alternative Flow		A
	Description	If the wireless earphones are connected
	Actions	
	1	Activate Turtleneck mode button
	2	The customer can press the button
	3	
Alternative Flow		Bluetooth is turned on, App is installed
	Description	If the wireless earphones are not connected
	Actions	
	1	Inactivate Turtleneck mode button
	2	The customer cannot press the button
	3	Show alert window if the customer try to press inactivated button

Description of Usecase 14 Turtleneck mode button activation



Usecase diagram 15 'Turtleneck mode' button activation

## 11.2. Start Turtleneck mode

After the wireless earphones are connected and the turtleneck button is activated, clicking the button will turn on the turtleneck mode, leading to a transition to the turtleneck page and initiating sensor measurements.



### Turtleneck mode

#### #02. Start ‘Turtleneck mode’

**Requirement Type:** functional

**For Whom?** Customer

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / **medium** / high

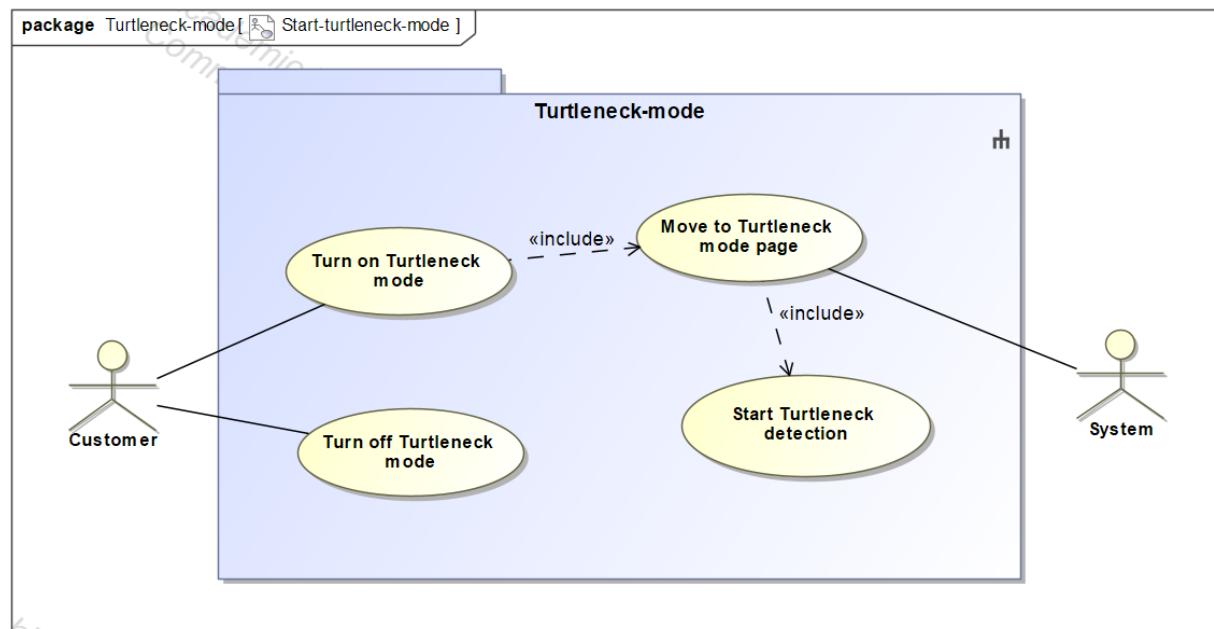
**Description:**

If the “On” button is pressed, move to “Turtleneck mode” page and start the detection.



Name		Start turtleneck mode
ID		
Description		The customer turn on the turtleneck mode
Trigger		The customer want to use turtleneck mode
Actors		Customer
Pre-conditions		The turtleneck mode button is connected and the button is activated
Post-conditions		Ready to start turtleneck detection
Basic Flow		
	Description	The customer has activated turtleneck mode button
	Actions	
	1	The customer press the button
	2	
	3	
Alternative Flow		A
	Description	If the turtleneck mode is on
	Actions	
	1	Move to Turtleneck mode page
	2	Start turtleneck detection
	3	
Alternative Flow		B
	Description	If the turtleneck mode is off
	Actions	
	1	Stop the detection
	2	Move to Turtleneck mode page
	3	

Description of Usecase 15 Start turtleneck mode



Usecase diagram 16 Start Turtleneck mode

## 11.3. Turtleneck detection

The Turtleneck threshold is set at 20 degrees. The user's initial state is 'Normal', and if the user's neck angle exceeds 20 degrees, it is considered as 'turtleneck' posture, triggering a warning notification along with a message to stretch the neck. When the user's neck angle returns to 20 degrees or below, it reverts to the 'Normal' state.

Measurement is halted if the connection between the device and wireless earphones is disconnected or if the Turtleneck mode is terminated by the user.



### Turtleneck mode

#### #03. Turtleneck detection

**Requirement Type:** functional

**For Whom?** Customer, Management

**User Satisfaction** low / medium / high

**User Dissatisfaction** low / medium / high

##### Description:

The earphone's sensor measures the degree of tilt of the neck in real-time to detect forward head posture(turtleneck). The threshold for turtleneck is set at **20 degrees**

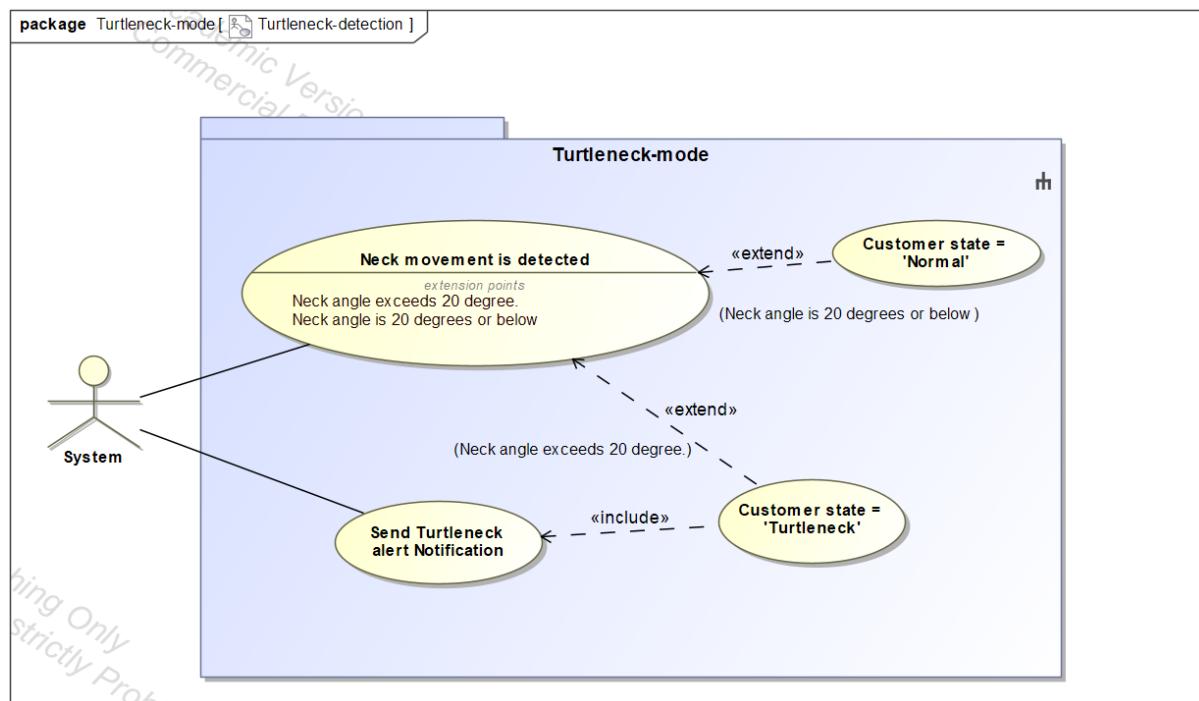
- 'Normal' state : when the neck angle is 20 degrees or below.
- 'Turtleneck' state : when the neck angle exceeds 20 degrees.

These criteria determine the color change of the solid line on the screen, the activation of the corresponding message, and the overall system feedback for the user's neck posture.



Name		Turtleneck detection
ID		
Description		The sensor of the wireless earphones detect the customer's neck state
Trigger		The turtleneck mode is turned on
Actors		Sensor, System
Pre-conditions		The wireless earphones are connected and the turtleneck mode is turned on
Post-conditions		The customer's neck state is detected
Basic Flow		
	Description	The sensor of the wireless earphones detect the customer's neck angle
	Actions	
	1	Detection of the customer's neck angle
	2	The threshold for turtleneck is set at 20 degrees
	3	
Alternative Flow		A
	Description	If the neck angle exceeds the threshold
	Actions	
	1	The customer state = 'turtleneck'
	2	
	3	
Alternative Flow		B
	Description	If the neck angle is below the threshold
	Actions	
	1	The customer state = 'normal'
	2	
	3	

Description of Usecase 16 Turtleneck detection



Usecase diagram 17 Turtleneck detection

## 11.4. Visualize real-time neck angle

The real-time reflection screen is a visual display showing the tilt status of the user's neck. This screen is displayed when the turtleneck mode is activated. Users can monitor the current angle of their neck in real-time on this screen.

If the neck angle exceeds the safe range, a warning is displayed to the user along with a message encouraging neck stretching. Additionally, this screen can provide visual guidance to assist users in adjusting their neck posture.



Turtleneck mode

### #04. Visualize real -time neck angle

**Requirement Type:** functional

**For Whom?** Customer

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** **low** / medium / high

**Description:**

The user interface is designed to display the reference angle of 20 degrees for turtleneck as a dashed line on the screen. Simultaneously, the current neck angle is represented by a solid line, with real-time updates reflected on the screen.

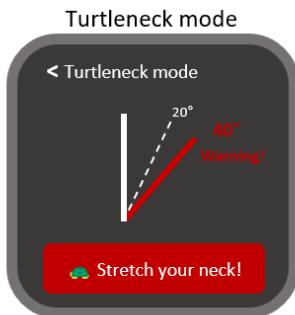
The **solid line** changes its color to **green** when in a 'normal' state and **red** when in a 'turtleneck' state. Additionally, an 'Warning!' message is triggered.

At the bottom of the screen, a message box conveys the current status.

If the posture is 'normal,' a **green box** displays the message "**Good job. Keep it up.**" In the case of a 'forward head posture,' a **red box** appears with the message "**Tuck in your neck!**"

Seite 14 Software Engineering Analysis

*Snowcard 24 Visualize real-time neck angle*



*Image 6 Plan for real-time reflection screen*

## 11.5. Turtleneck warning notification

If turtleneck posture is detected while the real-time monitoring screen is not open, a pop-up warning notification is sent to the user.



### Turtleneck detection

#### #05. Turtleneck warning notification

**Requirement Type:** functional

**For Whom?** Customer, Contractor

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** **low** / medium / high

##### Description:

Upon detecting Turtleneck, a push notification is sent to the user as a warning message "Turtleneck detected. Tuck in your neck!".

Given that the user is wearing earphones, an audible alert accompanies the notification and a voice message saying "Tuck in your neck!" is broadcasted.

*\* Users have the option to customize these alerts, including push and voice notifications, in the Settings -> Notification Settings.*

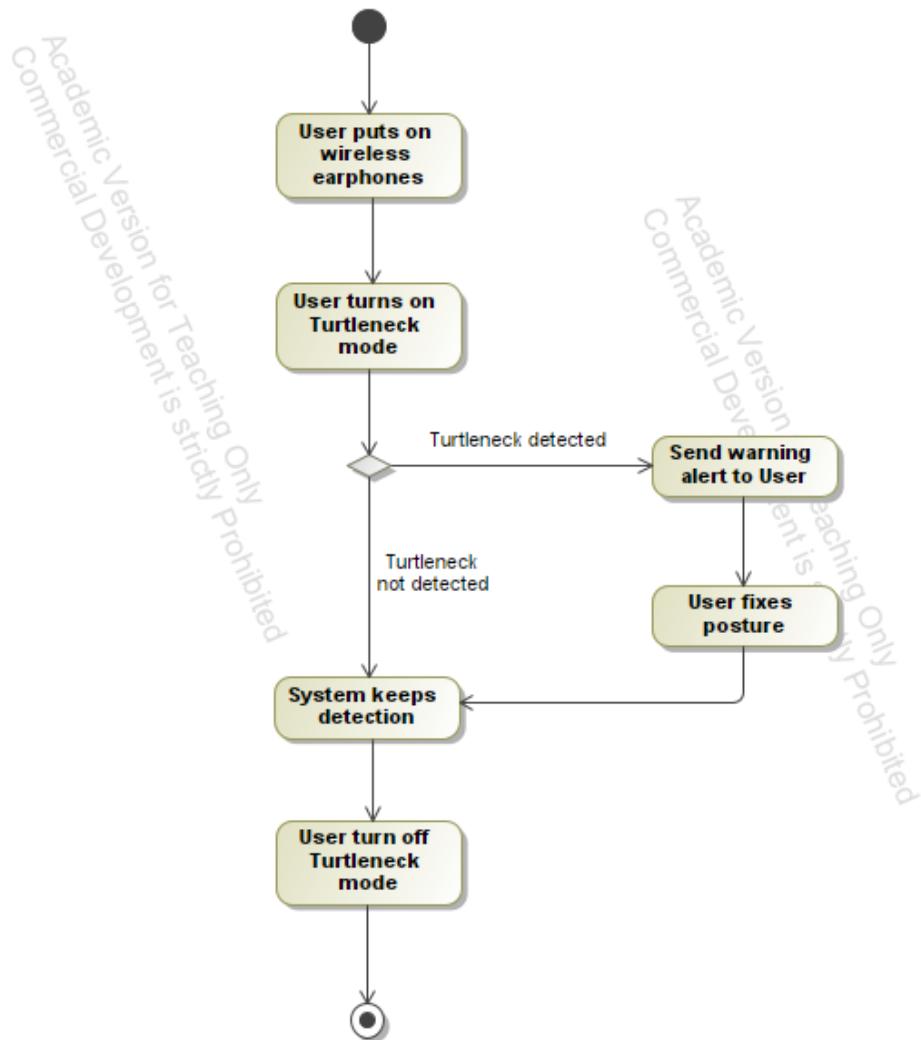
Seite 15 Software Engineering-Analysis

Snowcard 25 Turtleneck warning notification



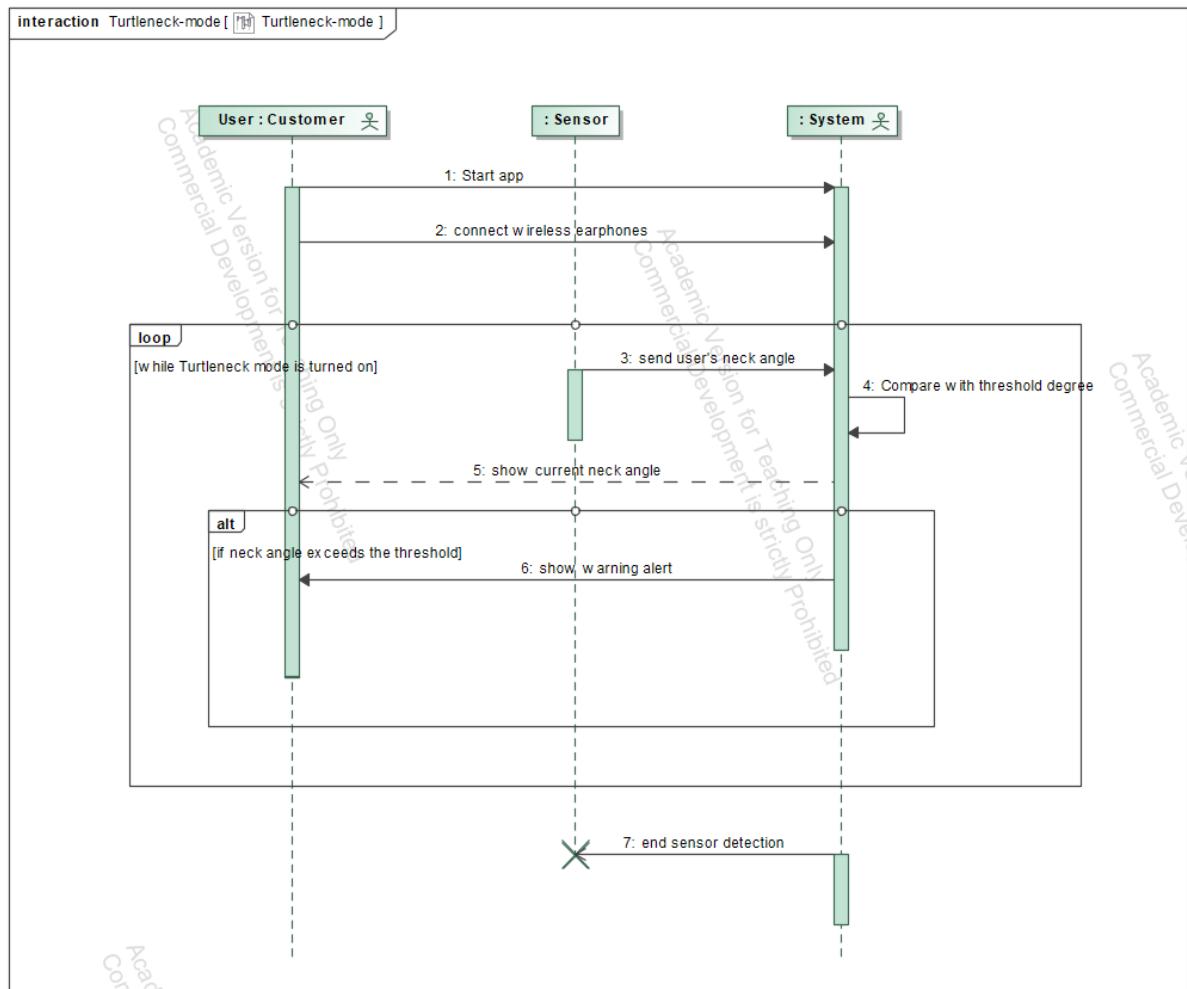
Image 7 Turtleneck warning notification

## 11.6. Activity diagram



Activity diagram 9 Turtleneck mode

## 11.7. Sequence diagram



## **12. Feedback from expert**

Based on all the health data and results derived from the smartwatch, weekly reports are generated, providing users with a visually easy-to-understand health score out of 100. Additionally, expert feedback is provided to assist users in improving their health status. This feature is not merely about diagnosis but also about accompanying users through the improvement process, making it a core function of the program. Furthermore, users can evaluate the feedback they receive, and these evaluations are accumulated and reflected in the next feedback, enabling them to receive personalized feedback tailored to their needs.

### **12.1. Calculate the score**

We calculate the final health score by aggregating all the user's weekly comprehensive health data. This is to enable users to easily assess their health status numerically, allowing them to intuitively understand whether they have improved or worsened compared to the previous week. All data relevant to the user, including heart rate data, steps data, turtle neck data, and any personal information provided by the user, are utilized in computing the health score.

## #2: show the total score of health this week

**Requirement Type:** Functional

**For Whom?** User

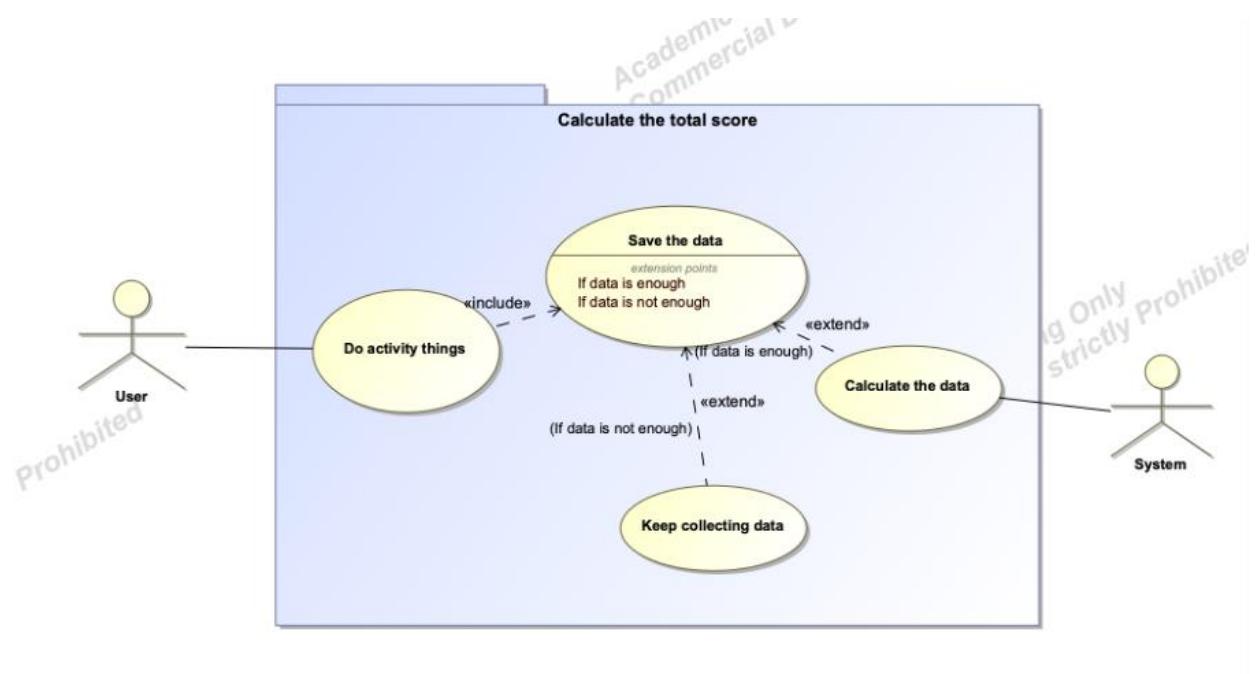
**User Satisfaction:** High

**User Dissatisfaction:** Low

### Description:

- Calculate the weekly heart rate, posture, and turtle neck to present a comprehensive health score for the week, scored out of 100 points, making it easy for the user to understand the results.
- This allows users to continuously assess their health status based on scores, making it easy to see improvements or deteriorations in results each week.

*Snowcard 26 Show the total score of weekly health*



*Use case diagram 18 Show the total score of weekly health*

<b>Name</b>	Calculate the total score
<b>ID</b>	13
<b>Description</b>	Calculate the user's weekly total health score
<b>Trigger</b>	User did the activity things for week
<b>Actors</b>	Customer, System
<b>Pre-conditions</b>	The user's weekly data has to be collected
<b>Post-conditions</b>	The health score is shown up on the screen
<b>Basic Flow</b>	<p><b>Description</b> Calculate the User's health score</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> Collect weekly user data on steps taken, heart rate, and turtle neck posture.</li> <li><b>2</b> Calculate the health score based on each data point.</li> <li><b>3</b> Send the health score along with a weekly report to the user.</li> </ul>
<b>Alternative Flow</b>	<p>A</p> <p>If there is a failure or omission in data collection</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li><b>1</b> The system displays a warning to the user and guides them to recollect the data.</li> <li><b>2</b> If some data is inaccurate or anomalous, the system notifies the user and provides a revised health score.</li> </ul>

*Description of Usecase 17 Show the total score of weekly health*

## 12.2. Make the weekly report from expert

To derive meaningful results from all user data, sufficient time is needed; hence, it is sent to the user's smartphone once a week along with the weekly report. Additionally, the data collected over the week is transmitted to relevant experts who evaluate the user's weekly data and generate feedback data. This feedback is then sent back to our software system and stored in the database.

## #2 give the feedback from expert

**Requirement Type:** Nonfunctional

**For Whom?** User

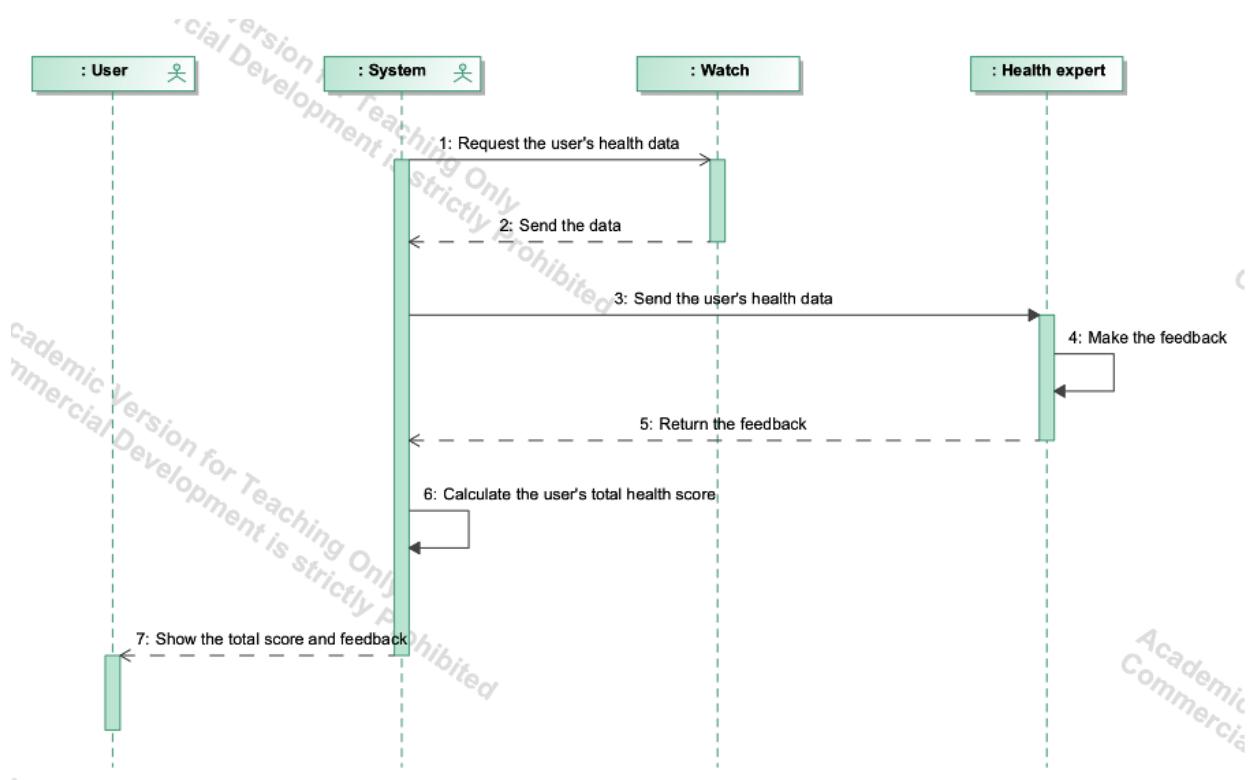
**User Satisfaction:** High

**User Dissatisfaction:** Low

**Description:**

- We provide users with feedback on weekly reports using either experts or AI. This feedback includes specific insights into what areas the user struggled with and how they can improve. This feedback data accumulates over time and continues to influence future diagnostic reports.

Snowcard 27 Give the feedback from expert



Sequence diagram 12 Give the feedback from expert

## 12.3. Visualize the feedback report

As our software program is targeting users of various age groups, we utilize user-friendly UI/UX to visualize user feedback in a way that is easy to read for any user. This is done to consider the convenience of users.

UI APPLIED SCIENCE

### #3: visualize the feedback report

<b>Requirement Type:</b>	Functional
<b>For Whom?</b>	User
<b>User Satisfaction:</b>	High
<b>User Dissatisfaction:</b>	Low

#### Description:

System visualizes the comprehensive diagnosis result of heart rate, step (posture), and turtle neck on the screen for the user to confirm.

*Snowcard 28 Visualize the feedback report*

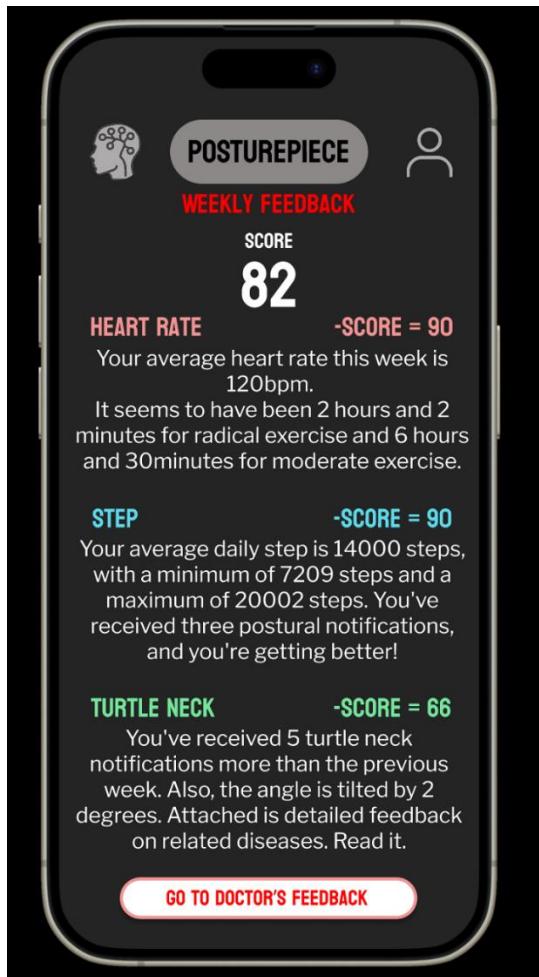


Image 8 UI/UX Visualize the feedback report

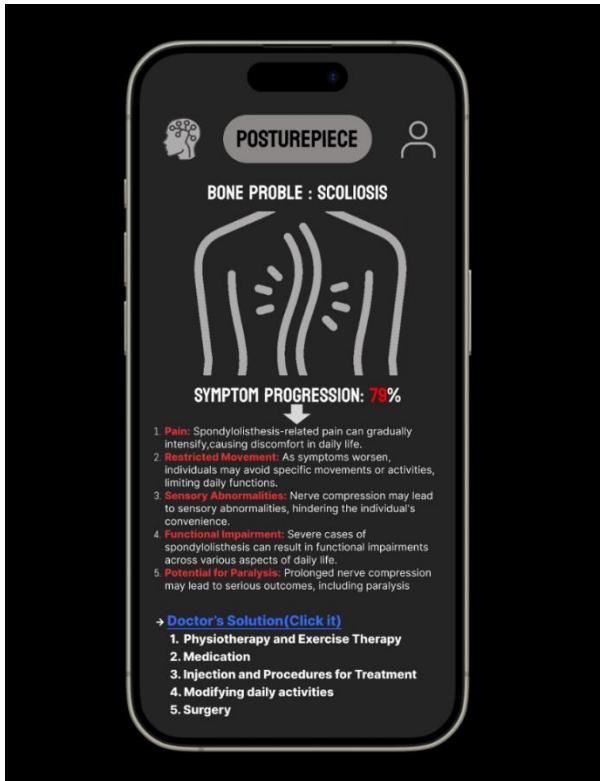


Image 9 UI/UX Visualize the feedback report

## 12.4. Send the alarm to user

We send an alert to the user's smartwatch to notify them of the arrival of expert feedback. As expert feedback is a core feature that sets our software system apart from others, we ensure that users can always access it. The alert disappears as soon as the user checks the feedback, and it also persists for up to 3 minutes, allowing users to delete the alert before then.

## #4: send the alarm to user

**Requirement Type:** Functional

**For Whom?** User

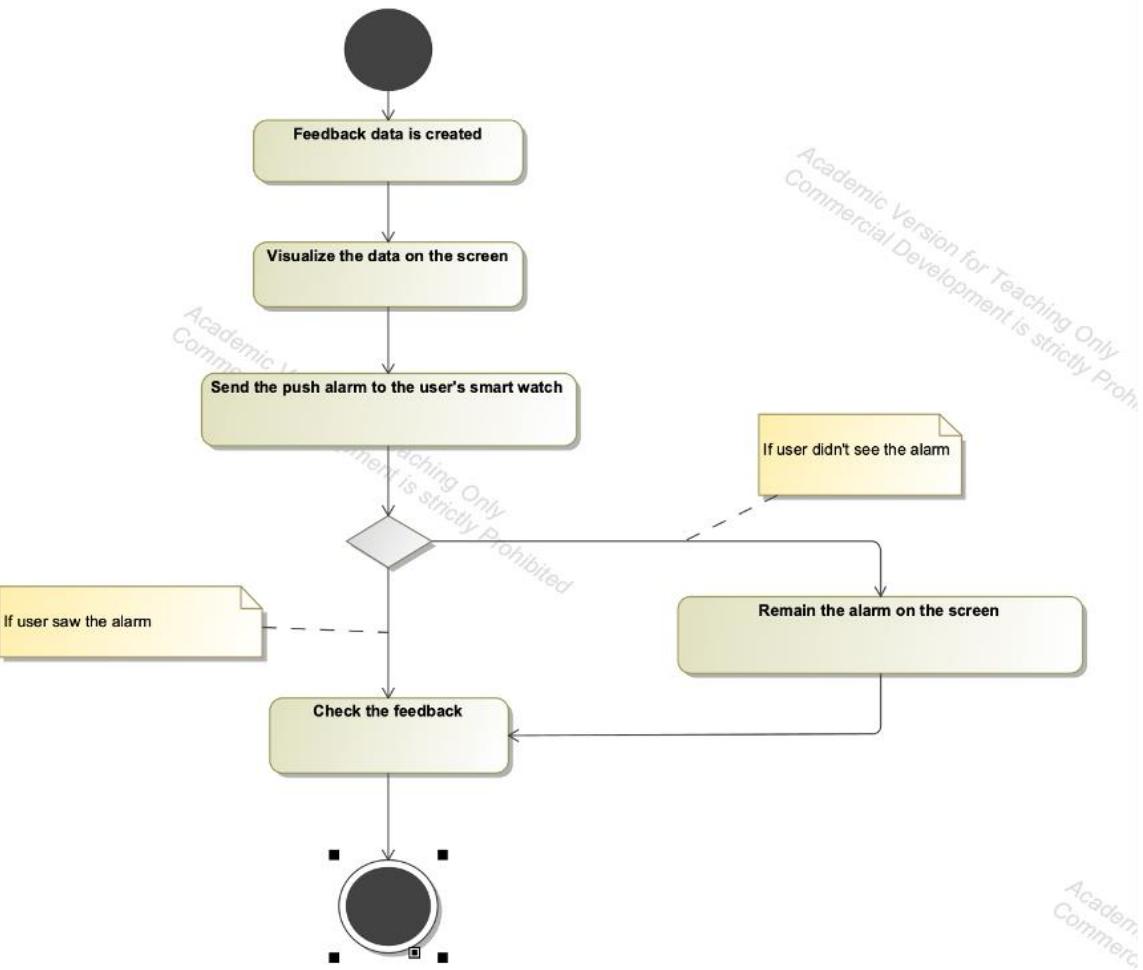
**User Satisfaction:** High

**User Dissatisfaction:** Low

**Description:**

We display an alert on the user's smartwatch to notify that the weekly doctor report has arrived on the user's smartphone.

*Snowcard 29 Send the alarm to user*



Activity diagram 10 Send the alarm to user

## 12.5. Review for the feedback

Users have ability to evaluate the feedback they receive. They can provide feedback indicating dissatisfaction, such as when the feedback is not detailed enough or contains incorrect information. Conversely, they can also provide supplementary evaluations for aspects they are satisfied with. The evaluations submitted by users accumulate and are reflected in subsequent feedback, allowing users to receive increasingly personalized feedback over time.

## #3: review for feedback

**Requirement Type:** Nonfunctional

**For Whom?** User

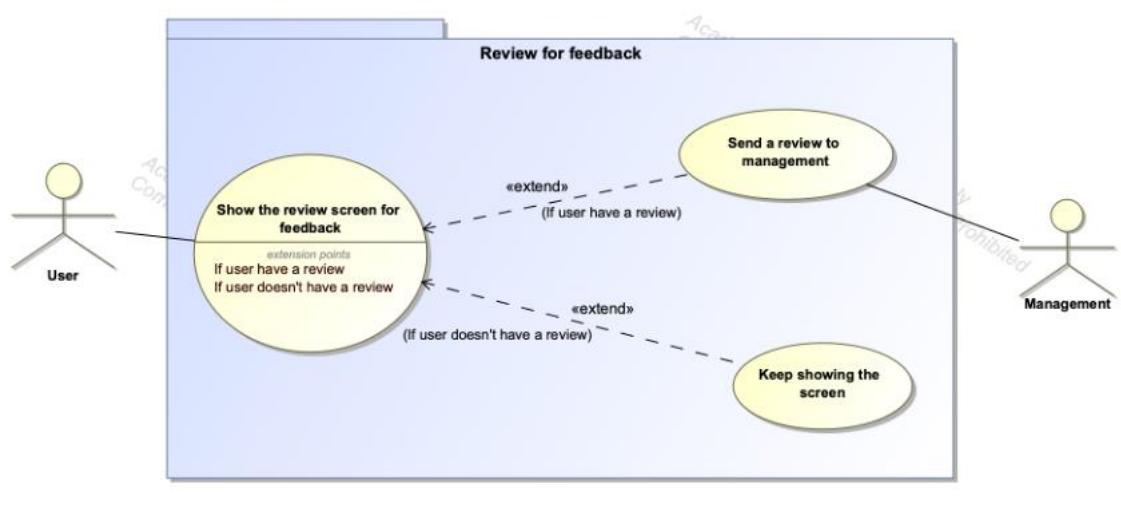
**User Satisfaction:** High

**User Dissatisfaction:** Low

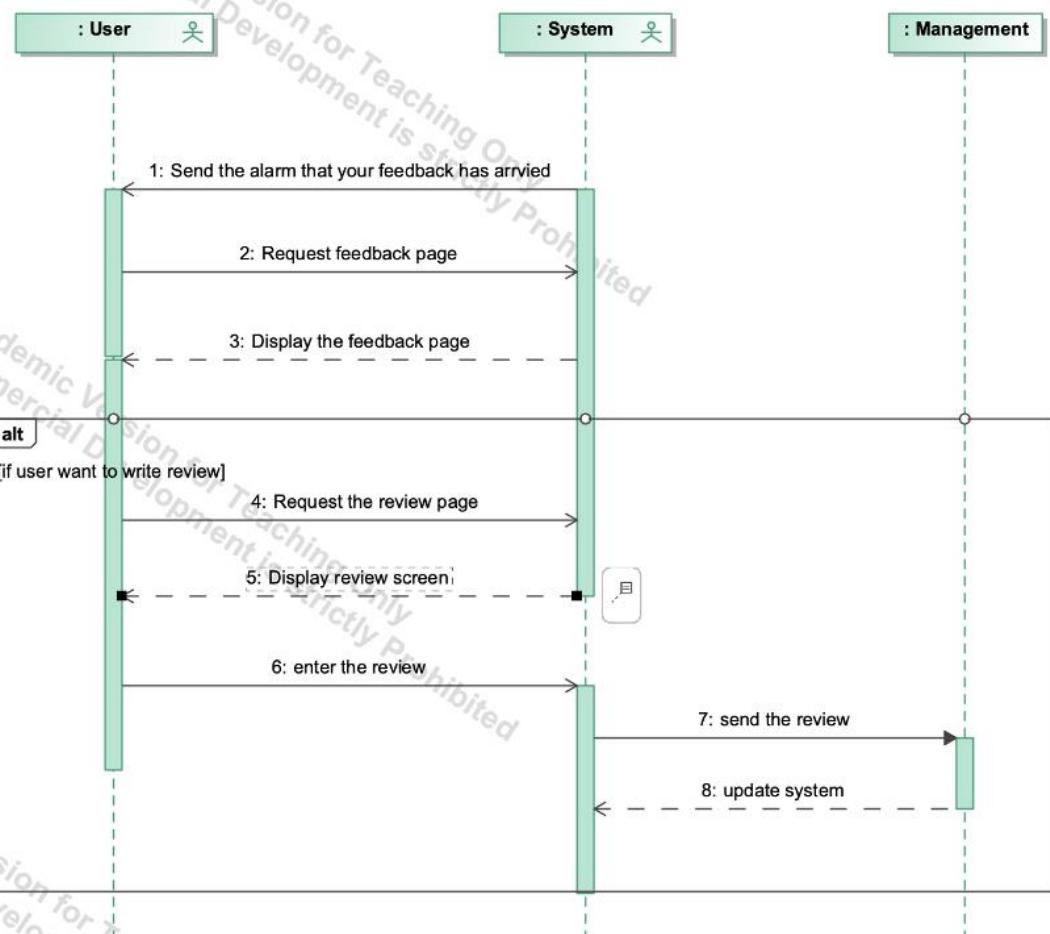
**Description:**

This feature allows users to indicate whether the feedback was helpful and provide insights on areas that need improvement.

*Snowcard 30 Review for feedback*



*Use case diagram 19 Review for feedback*



Sequence diagram 13 Review for feedback

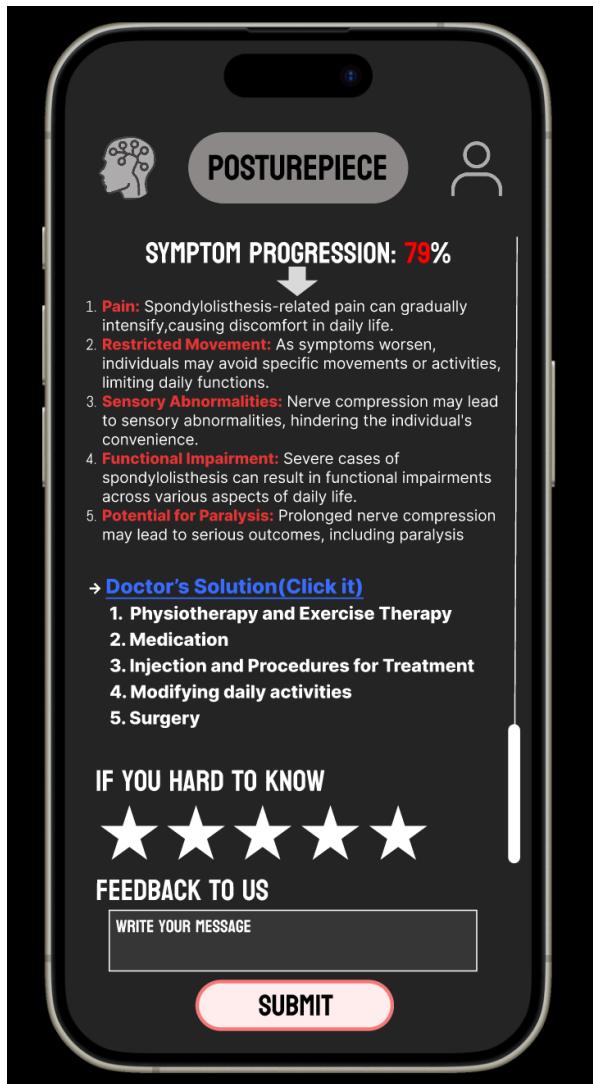


Image 10 UI/UX Review for feedback

# 13. Other useful functions

## 13.1. Sleep analysis

The sleep analysis feature tracks and analyzes sleep patterns, duration, and quality to provide users with insights into their sleep habits. It helps users understand their sleep patterns, identify sleep disturbances, and improve overall sleep quality for better health and well-being.



### #sleep analysis

**Requirement Type:** non-functional

**For Whom?** customer

**User Satisfaction:** high

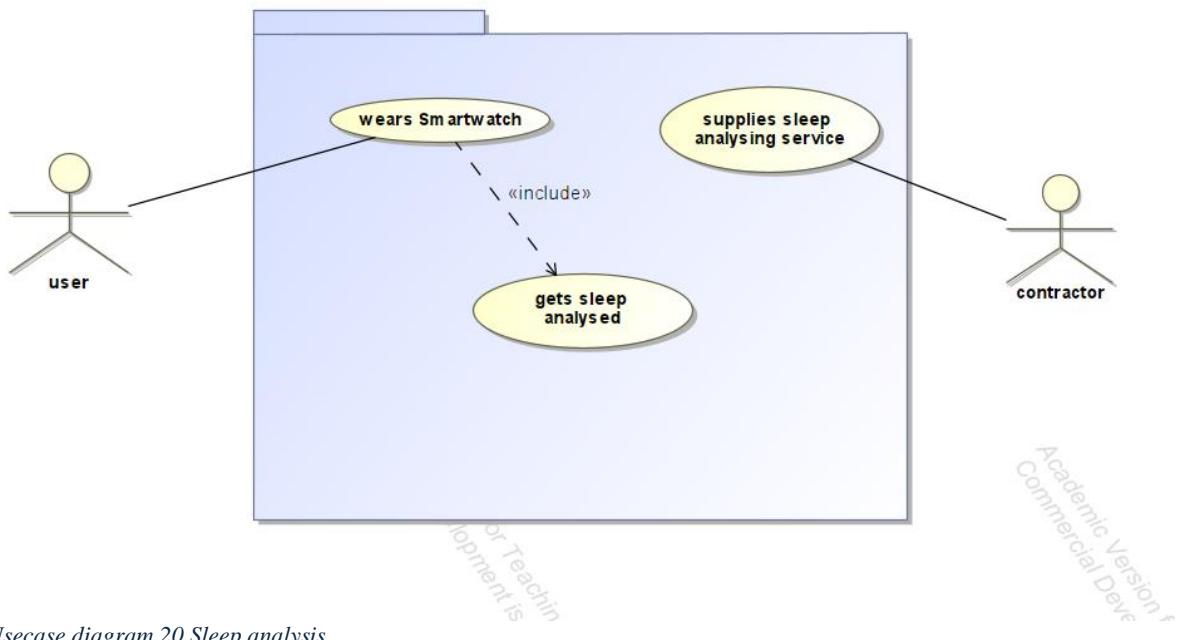
#### Description:

This is a app which records data on the sleep of the user wearing the smartwatch.

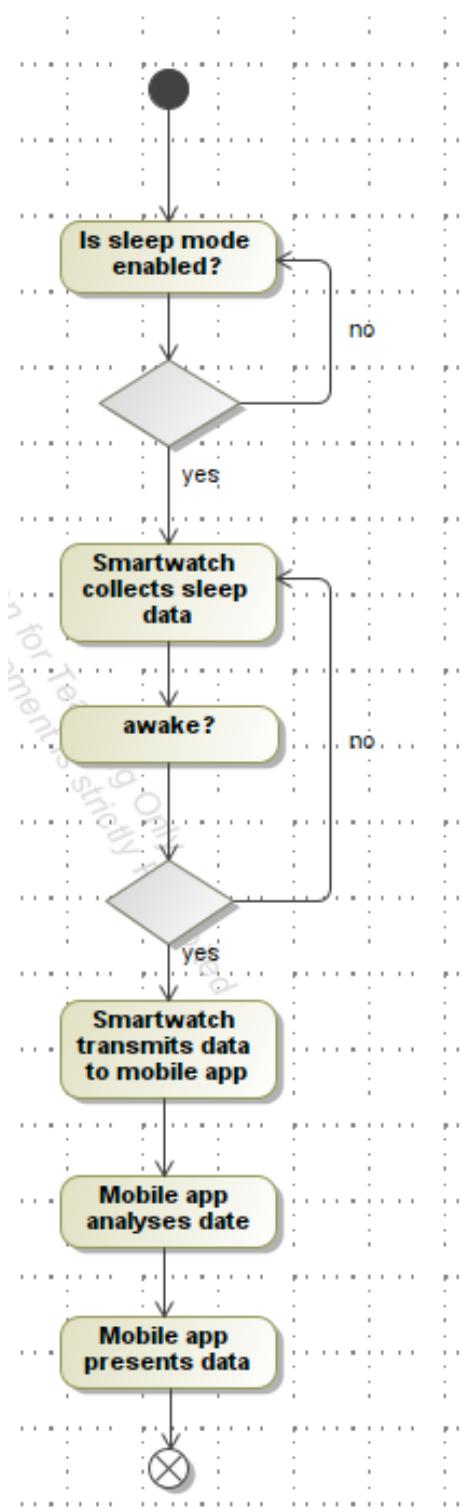


<b>Name</b>	Sleep analysis
<b>ID</b>	6
<b>Description</b>	Analyses Users sleep
<b>Trigger</b>	User falls asleep
<b>Actors</b>	User falls asleep
<b>Pre-conditions</b>	User wears watch while sleep
<b>Post-conditions</b>	
<b>Basic Flow</b>	
<b>Description</b>	The app records the sleep of the user and gives feedback
<b>Actions</b>	
<b>1</b>	User puts on watch before sleep
<b>2</b>	watch analyses sleep

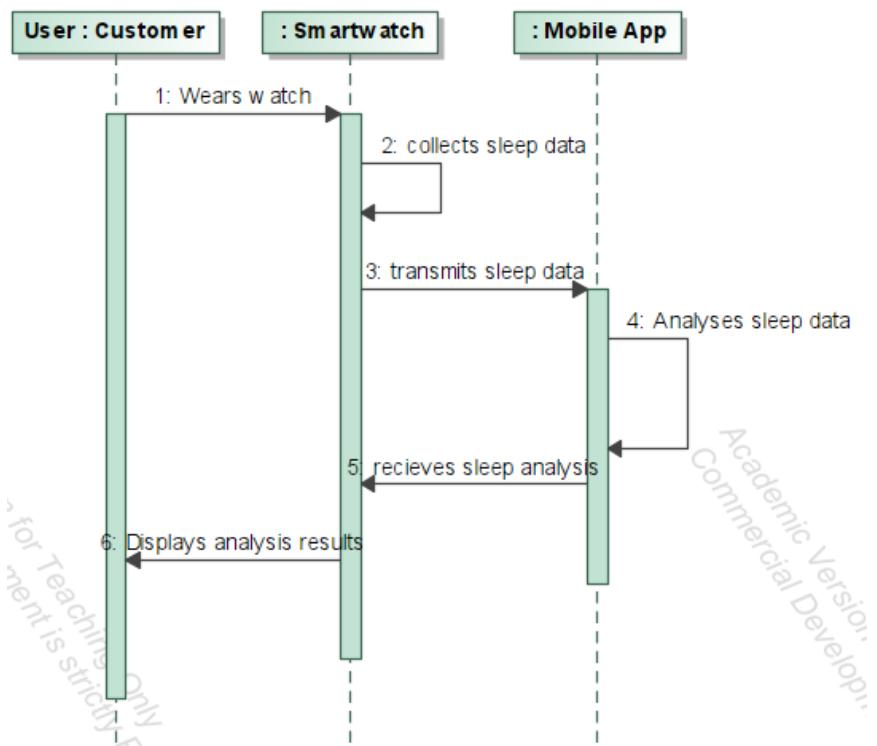
*Description of Usecase 18 Sleep analysis*



*Use case diagram 20 Sleep analysis*



Active diagram 10 Sleep analysis



Sequence diagram 14 Sleep analysis

## 13.2. Prolonged sitting (sedentary) detection

Prolonged sitting (sedentary) detection is a feature that prompts users to engage in stretching or walking if no movement is detected for more than 1 hour. If no movement is detected for 1 hour, indicating a sedentary state, a warning notification is sent. When movement is detected, the user's status is reset to active, and the 1-hour countdown restarts. Sedentary state and sleep state are distinguished based on heart rate, and notifications are not sent during sleep.



### Other useful functions

#### #04. Prolonged sitting detection

**Requirement Type:** functional

**For Whom?** Customer, Contractor

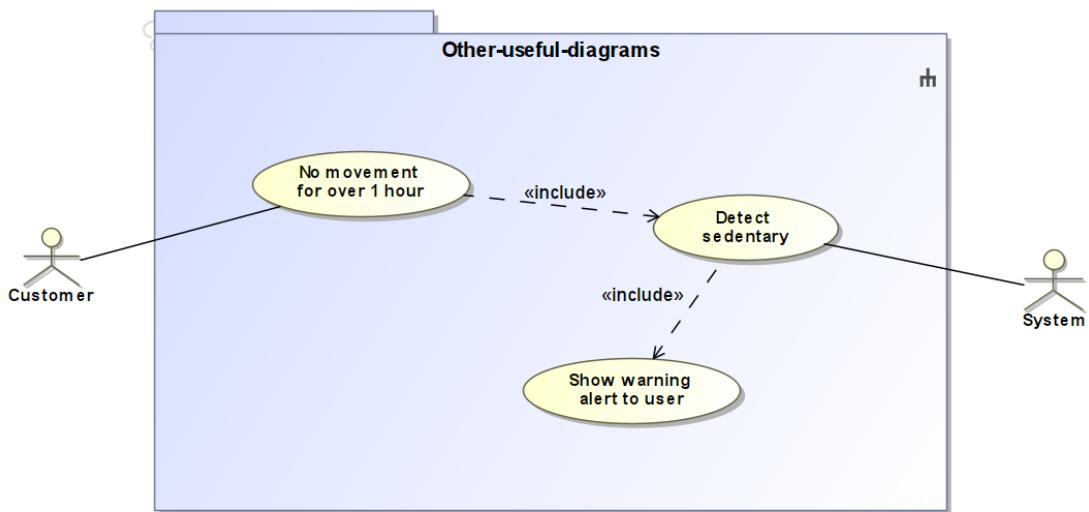
**User Satisfaction** low / medium / high

**User Dissatisfaction** low / medium / high

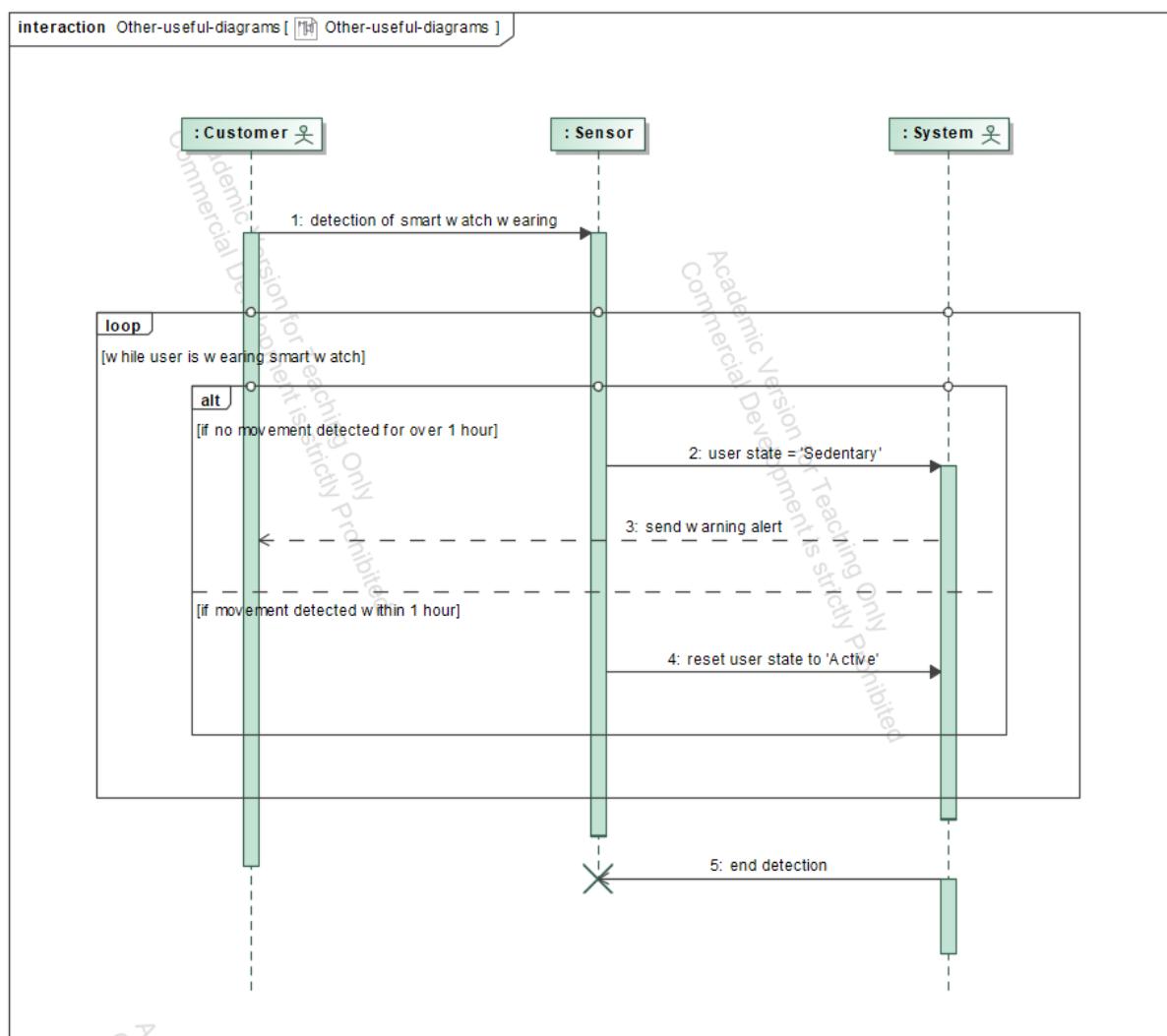
##### Description:

While wearing a smartwatch, if no movement is detected for over **an hour**, it is considered as '**sedentary' status**', and a warning alarm is sent to encourage stretching.

The states of 'sedentary' and 'sleep' are distinguished based on heart rate, and the alarm is not sent during sleep state.



Usecase diagram 21 Prolonged sitting detection



Sequence diagram 15 Prolonged sitting detection

# 14. Payment & Subscription

## 14.1. Subscription

The subscription system can make money at the company level, and customers can use all the performance of the application. There are two types of subscriptions, and the standard option is to use in-app graphs, be notified of the results of the analysis, and receive feedback. It is available for free.

In addition to all things in the standard, the Pro option has the Turtle Neck Analysis option, which alerts consumers to correct their posture when walking or acting in real time. Additionally, a free month promotion was held so that consumers could recognize its performance.



FRANKFURT  
UNIVERSITY  
OF APPLIED SCIENCES

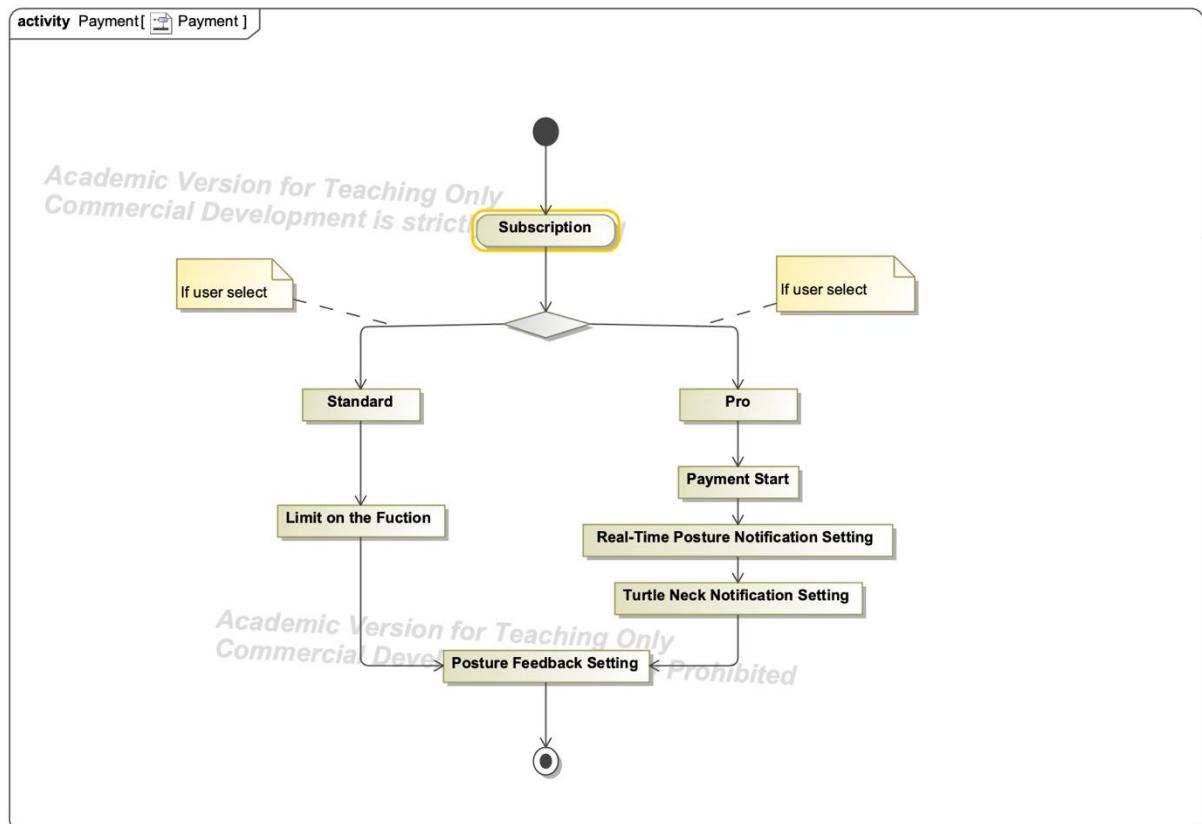
### #15-1: Subscription

<b>Requirement Type:</b>	functional
<b>For Whom?</b>	customer
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

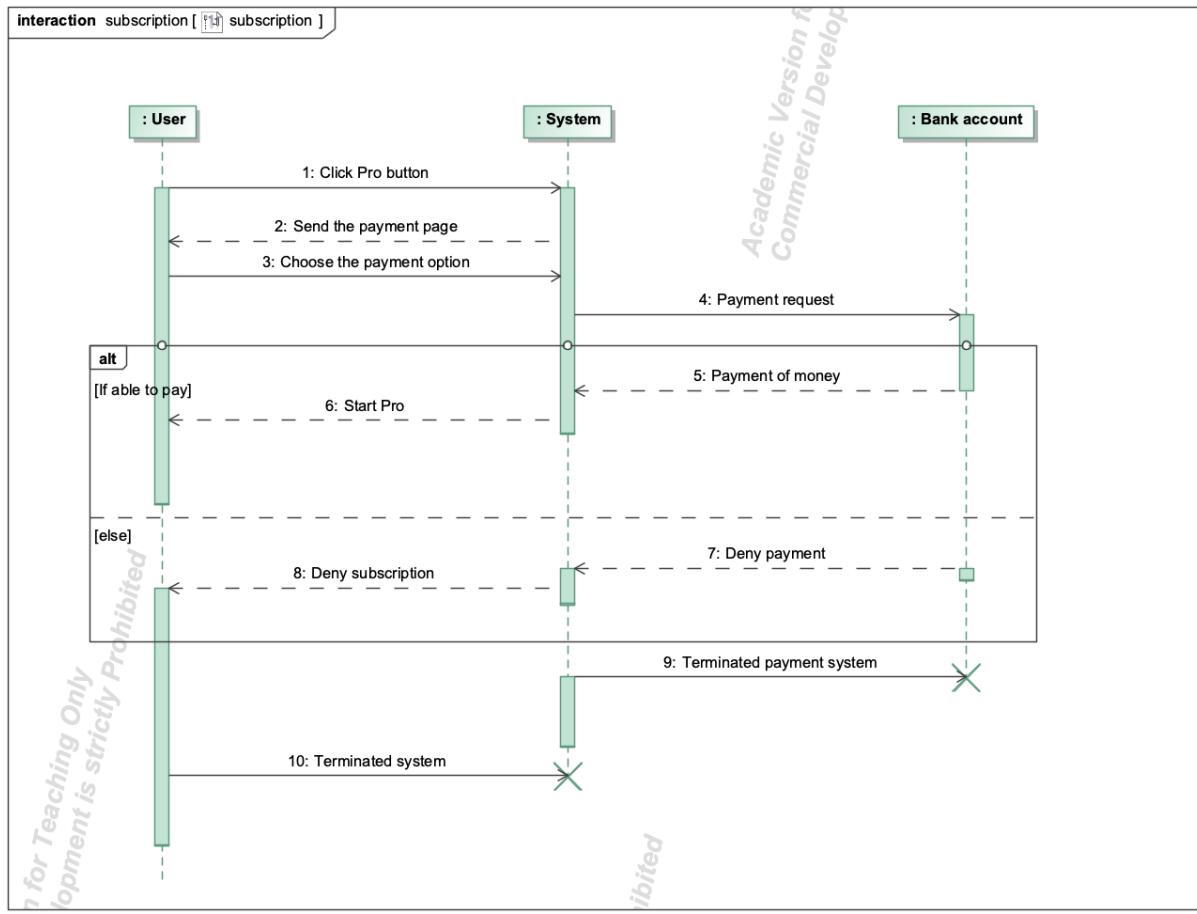
**Description:**  
If customers want to use deeper than standard mode, they need to subscribe it.  
We have two option.

Basic option = Apps analyze them, and when enough data is gathered, they draw conclusions. No notifications or real-time analysis of posture corrections are available in between. Free options. And can use graph such as number of steps.

Pro option = It analyzes in real time and sends an alarm by analyzing posture errors while walking. You can also receive detailed information about the conclusion and information about the turtle neck.



Activity diagram 11 Subscription



Sequence diagram 16 Subscription

## 14.2. Pricing and Promotional Strategy

Customers can consider whether to subscribe or not by allowing them to experience a month for free. It also measures inexpensive prices and teaches bad attitudes and habits, regardless of age or gender.

## #15-2: Pricing and Promotional Strategy

**Requirement Type:** non-functional

**For Whom?** customer

**User Satisfaction:** high

**User Dissatisfaction:** low

**Description:**

It holds a free subscription event for the first month, attracting many customers by inducing many people to use the program.

Software Engineering Analysis

Snowcard 33 Pricing and promotional strategy

### 14.3. Payment Integration

By supporting various payment systems, customers do not have to register their cards cumbersomely, but also support fast payment

## #15-3: Payment Integration

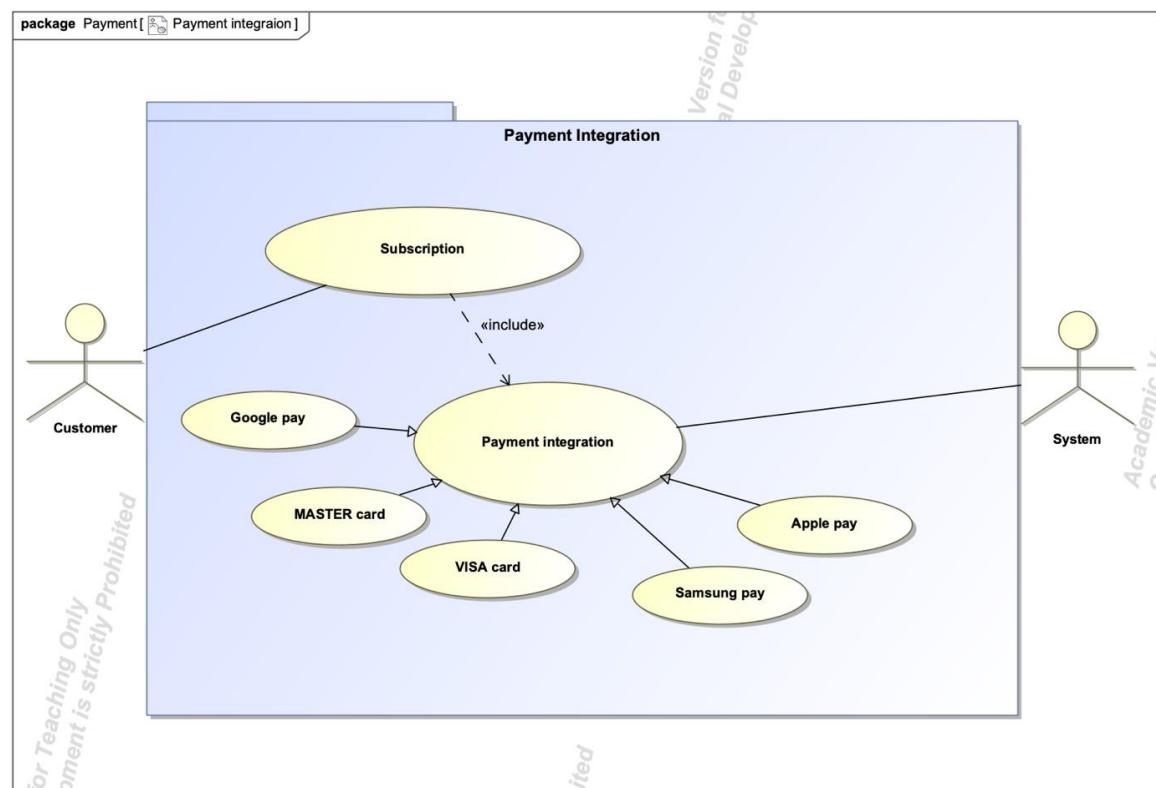
<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	Customer/contractor
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

### Description:

It supports payment of visas and master cards, and additionally supports Google Pay, Apple Pay, and Samsung Pay, allowing subscription services to be made smoothly.

Software Engineering Analysis

Showcard 34 Payment integration



Usecase diagram 22 Payment integration

Name		Payment integration
ID		
Description		The process of payment process
Trigger		The customer want to subscription
Actors		User
Pre-conditions		The customer click Pro button
Post-conditions		Subscription will be start
Basic Flow		
	Description	Go to the payment page
	Actions	
	1	Move to the payment page
	2	Choose the payment option
Alternative Flow		A
	Description	The subscription button is pressed
	Actions	
	1	If Visa card is selected
	2	Enter the card information.
	3	Payment progress
Alternative Flow		B
	Description	The subscription button is pressed
	Actions	
	1	If Master card is selected
	2	Enter the card information.
	3	Payment progress
Alternative Flow		C
	Description	The subscription button is pressed
	Actions	
	1	If Samsung pay is selected
	2	show the samsung pay
	3	Payment progress
Alternative Flow		D
	Description	The subscription button is pressed
	Actions	
	1	If Google pay is selected
	2	show the google pay
	3	Payment progress
Alternative Flow		E
	Description	The subscription button is pressed
	Actions	
	1	If Apple pay is selected
	2	Show the apply pay.
	3	Payment progress

*Description of Usecase 19 Payment integration*

## 14.4. Customer Support for Pro Users

It operates a customer center that will help pro users use subscription-related questions, refunds, additional changes in the app, and requests more quickly and safely.



## #15-4: Customer Support for Premium Users

<b>Requirement Type:</b>	<b>non-functional</b>
<b>For Whom?</b>	Customer/contractor
<b>User Satisfaction:</b>	<b>high</b>
<b>User Dissatisfaction:</b>	<b>low</b>

### Description:

In order to prevent inconvenience to customers who have made subscriptions and payments, we will take measures for inconvenience by operating quick refunds, cancellation of payments, and customer support centers.

Software Engineering Analysis

*Snowcard 35 Support for Premium users*

# 15. Usability

Usability is the requirement to make it easier for users to use the app. All usability-related tabs are included in the 'Settings' tab.

In the 'Settings' tab, user can easily change things like the app language, notification settings, theme settings, and network usage settings according to their preferences. In the Help tab, user can also contact the administrator directly with questions, check information about the app, and more. Usability also includes the ability to notify users that an error has occurred when the app is running.

## 15.1. App language

The app language setting allows user to set the app to the user's preferred language, and all text displayed within the app will change to that language. The default setting is English, with options for German and Korean.

### Usability



#### #01. App Language

**Requirement Type:** functional

**For Whom?** customer

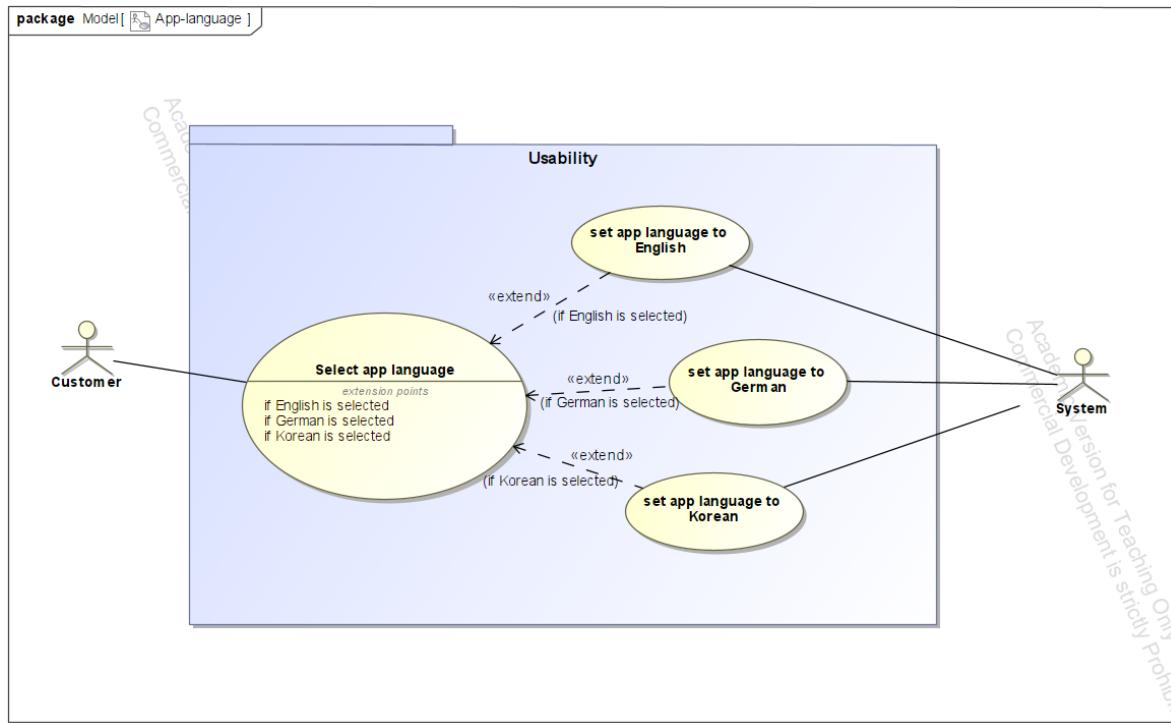
**User Satisfaction** low / medium / **high**

**User Dissatisfaction** **low** / medium / high

**Description:**

Customers can set up the app in the language of their choice. We support English, German, and Korean.





Usecase diagram 23 App language

## 15.2. Notification settings

Notification settings allow users to customize how they receive notifications for the app. This includes the ability to turn off push notifications, choose their own notification sound, and turn vibration off or on.

## Usability

### #02. Notifications settings

**Requirement Type:** functional

**For Whom?** customer

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** **low** / medium / high

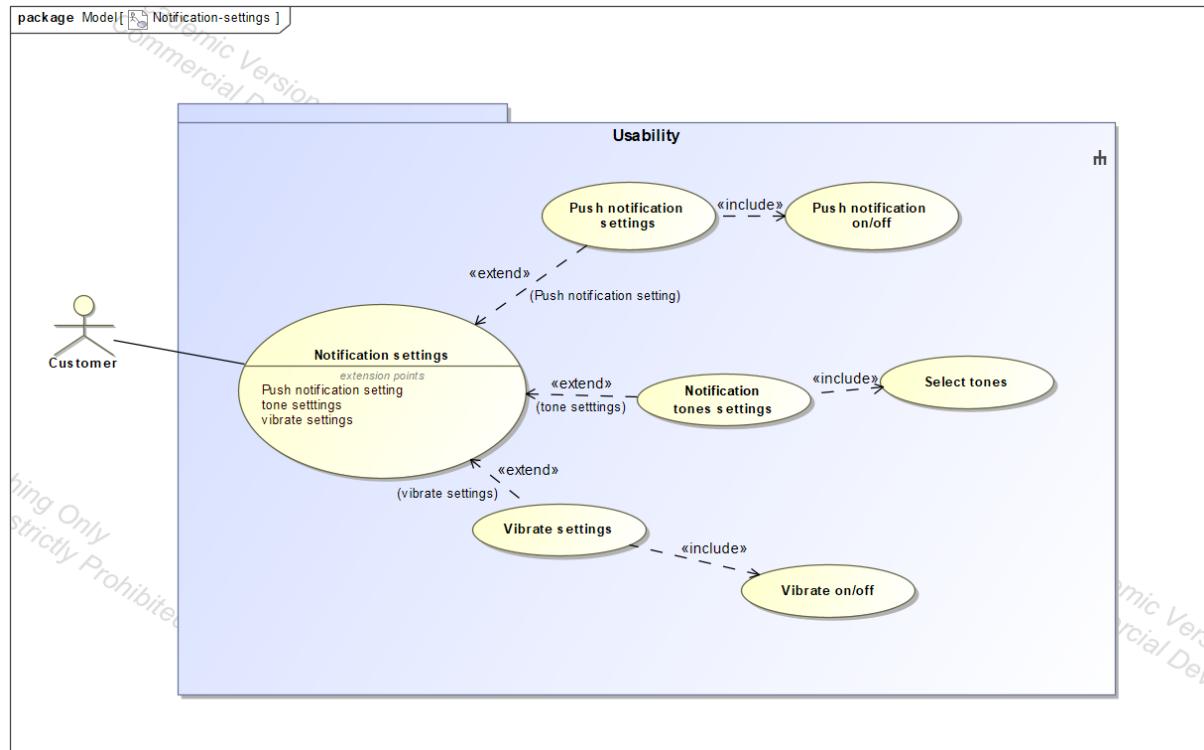
#### Description:

Customers can set whether to allow push notifications, as well as the tone and vibration.

- Push notification (on/off)
- Notification Tone (on/off)
- Vibrate (on/off)

Seite 3 Software Engineering-Analysis

Snowcard 37 Notification settings



Usecase diagram 24 Notification settings

## 15.3. Theme settings

Theme settings allow users to choose the theme of their app. Light mode is the default and can be changed to dark mode.

### Usability



#### #03. Theme settings

**Requirement Type:** functional

**For Whom?** customer

**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

##### Description:

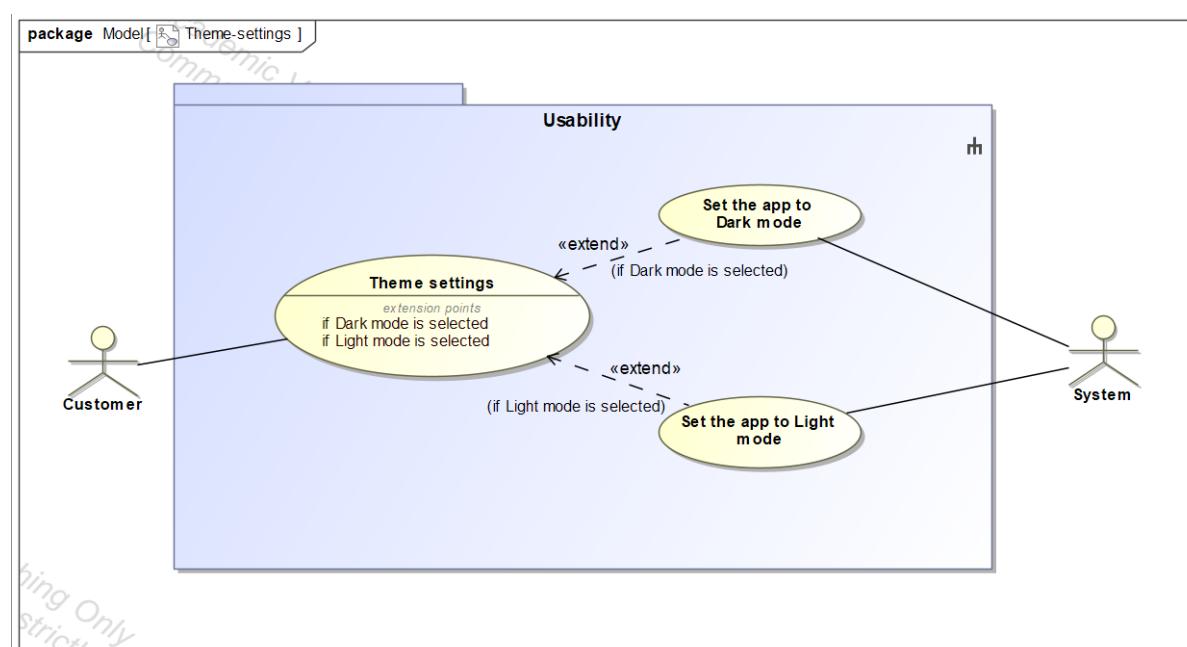
Customers can choose to have the app's theme mode.

- Light mode(default)
- Dark mode

Seite 4

Software Engineering-Analyse

Snowcard 38 Theme settings



Usecase diagram 25 Theme settings

## 15.4. Network usage settings

Network usage settings allow users to conserve mobile data by only allowing the app to run on Wi-Fi, or to always run with an "Always" option that allows mobile data usage.



### Usability

#### #04. Network usage settings

**Requirement Type:** functional

**For Whom?** customer

**User Satisfaction** low / medium / **high**

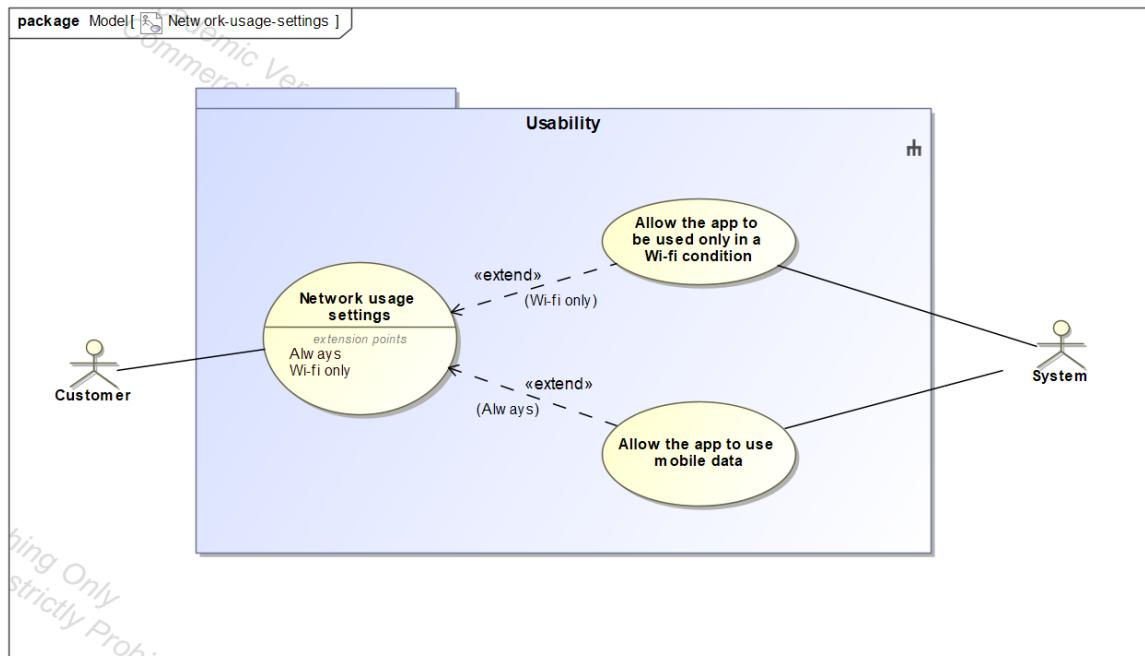
**User Dissatisfaction** **low** / medium / high

##### Description:

Customers can set the network environment in which they want to use the app.

- Always : Allow the app to use mobile data.
- Wi-fi only : Allow the app to be used only in a Wi-Fi environment.





Use case diagram 26 Network usage settings

## 15.5. Help & Information

Help & Information is a feature that provides users with information about the app. "Contact us" allows users to contact the administrator directly about the app, and "Terms and privacy policy" provides information about the use of personal information. "App info" provides a brief introduction to the app and version information.

## Usability

### #05. Help & Information

**Requirement Type:** functional

**For Whom?** Customer, Contractor

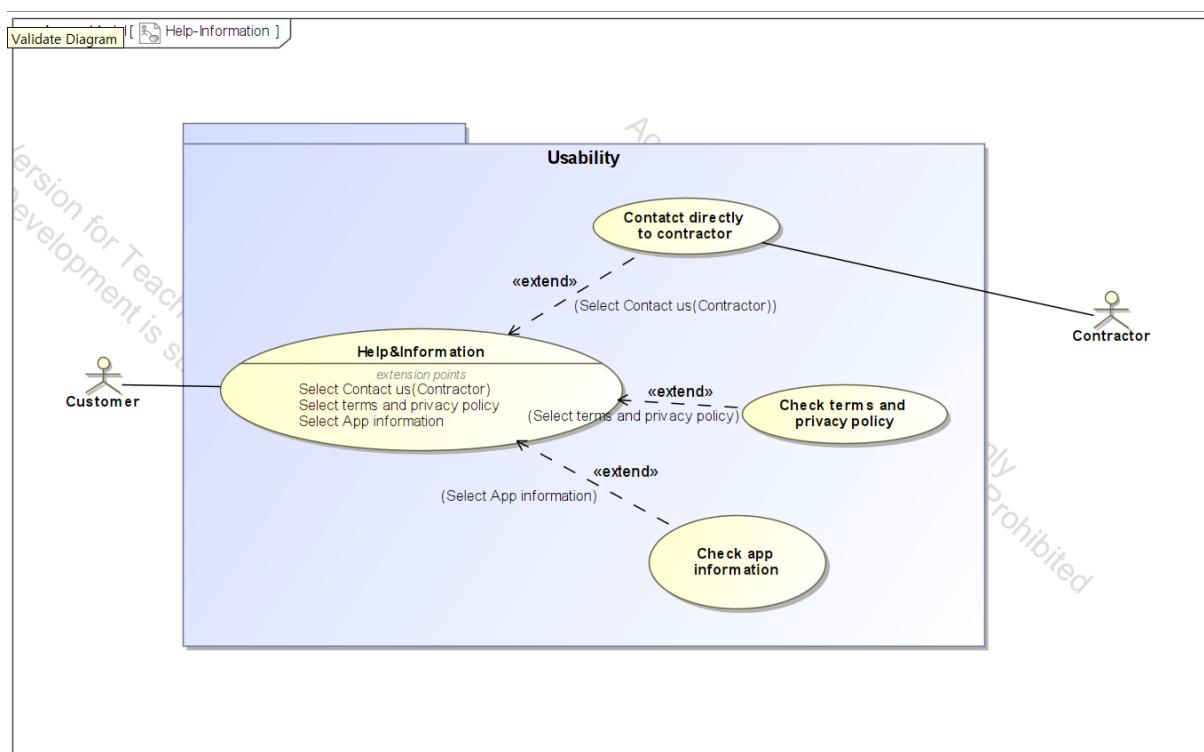
**User Satisfaction** low / medium / **high**

**User Dissatisfaction** low / medium / high

#### Description:

Customers can ask questions about the app, check the privacy policy, app version, and more.

- Contact us : If the customer need help, they can contact an administrator directly.
- Terms and Privacy Policy
- App info : Description and version information about the app.



Usecase diagram 27 Help&Information

## 15.6. Abnormal shutdown

If a user encounters an error while using the app, the system terminates the app and displays an error window to notify the user.



### Usability

#### #06. Abnormal shutdown

**Requirement Type:** non-functional

**For Whom?** Customer, Contractor

**User Satisfaction** low / medium / high

**User Dissatisfaction** low / medium / high

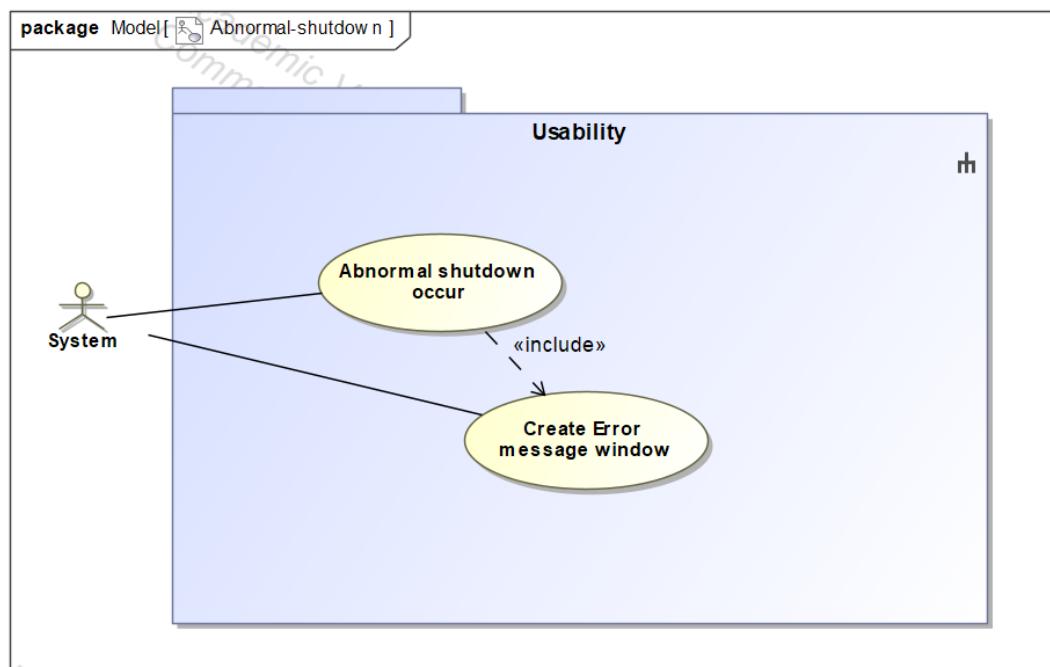
**Description:**

If an error occurs during use, stop the app from running and bring up the Abnormal shutdown window.

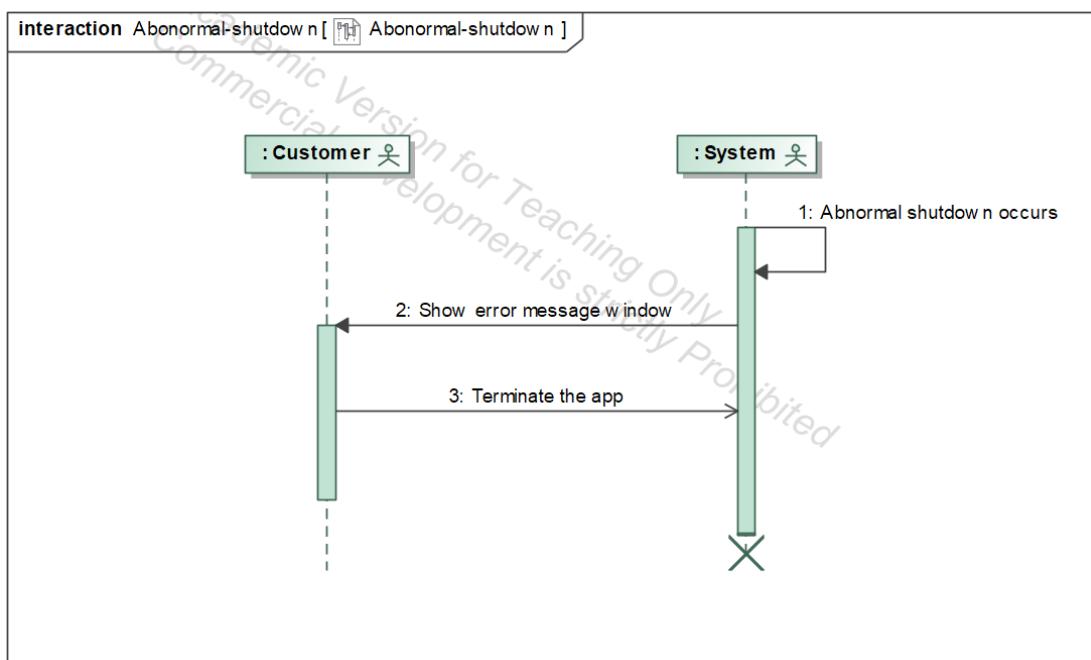
Seite 7

Software Engineering-Analyse

Snowcard 41 Abnormal shutdown



Use case diagram 28 Abnormal shutdown



Sequence diagram 17 Abnormal shutdown

## 15.7. Font size settings

The app allows users to choose from three font sizes: small, medium, and large, according to their preferences.

## Usability

### #07. Font size settings

**Requirement Type:** non-functional

**For Whom?** Customer, Contractor

**User Satisfaction** low / medium / **high**

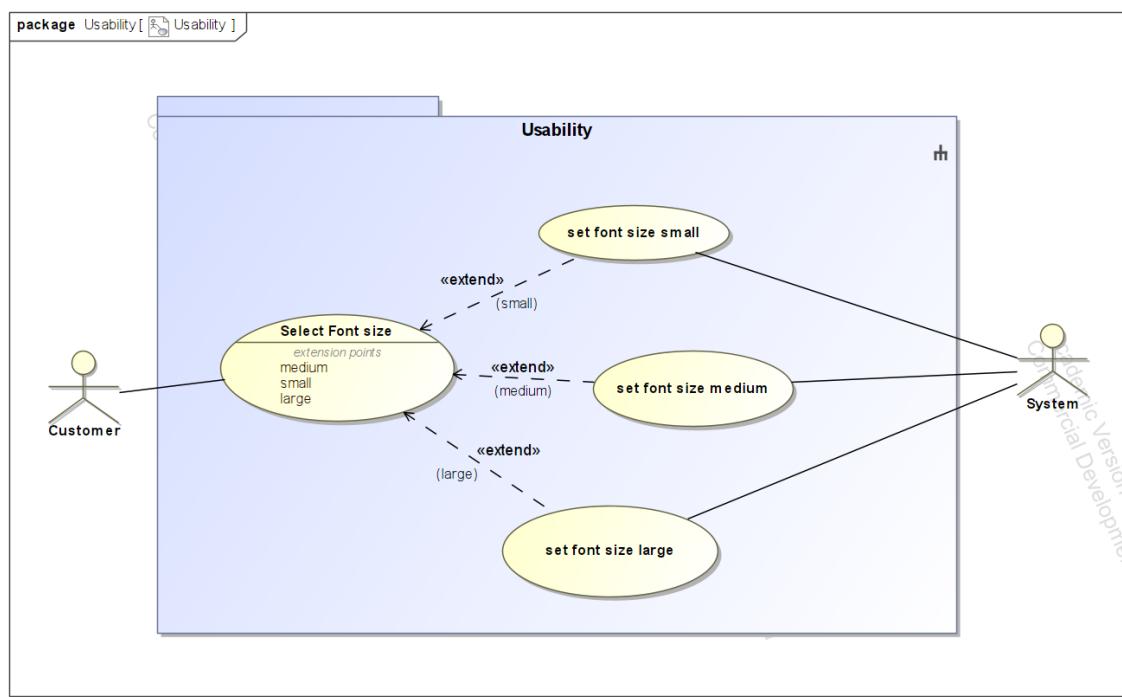
**User Dissatisfaction** low / **medium** / high

**Description:**

The user can choose from small, medium, or large font sizes within the app.

Seite 8 Software Engineering-Analysis

Snowcard 42 Font size settings



Usecase diagram 29 Fontsize settings

# 16. Real-time Data Collection and Logging

This is the entire process of data collection, starting from when the user wears the Apple Watch, triggering the commencement of data collection. The sensors collect and store data based on the user's movements, and data collection ceases when the user stops moving or no longer wears the Apple Watch. This requirement encompasses the recording of data from the beginning to the end of this process. In this section, I aimed to create requirements from the perspectives of the user, sensors, and the system, and based on these, I have developed snow cards and diagrams.

In terms of the progress status, I have completed all diagrams and snow cards except for the class diagram in the data storage section. In the next phase, I'm planning to focus on how the stored data is recorded. This will involve crafting more detailed requirements and creating snow cards and diagrams. Additionally, I intend to work on requirements for the data processing section, addressing aspects beyond just data-related considerations.

## 16.1. Activate sensors

The Sensor will continue unless any specific anomaly is detected while being worn - as incomplete data collection leads to inaccuracies, **the Sensor can not be manually stopped.**

## Collecting

- The sensor will continue unless any specific anomaly is detected while being worn - as incomplete data collection leads to inaccuracies, **the sensor cannot be manually stopped.**
  - User: wear the apple watch
  - System: detect whether user wears the apple watch with thermometer sensor or all the other sensors
    - if the system determines that the user is wearing an Apple watch..
      - System: start to activate sensors and features for measuring data

*Image 11 Activate sensors*



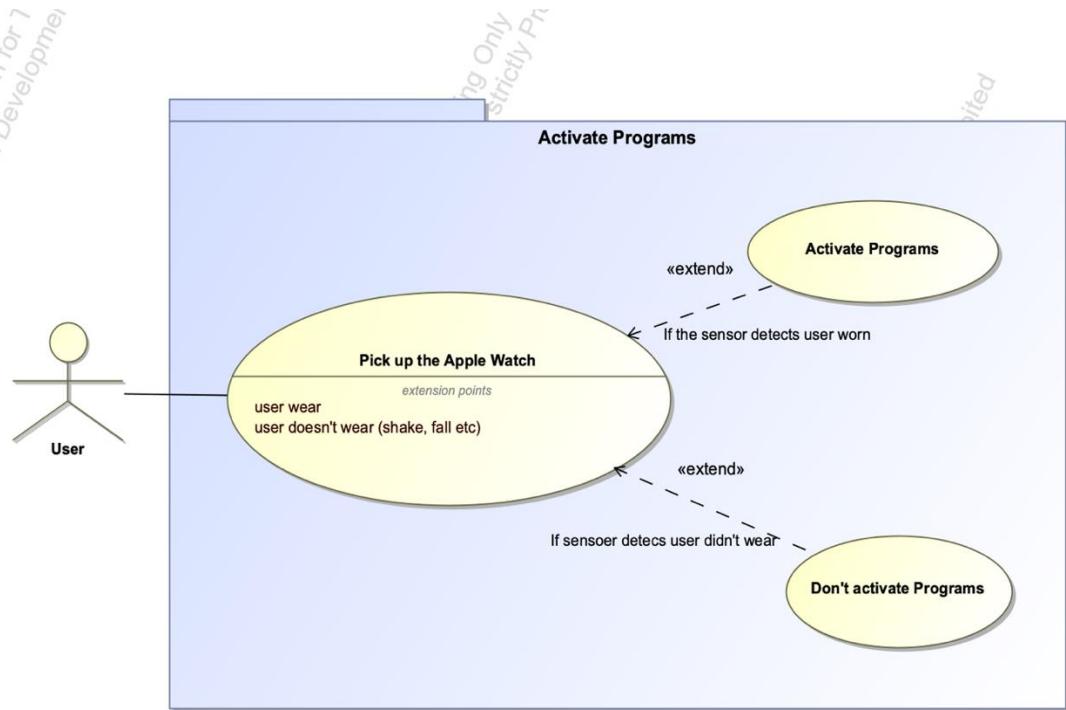
## #1: Activate Sensors

<b>Requirement Type:</b>	functional
<b>For Whom?</b>	User, management
<b>User Satisfaction:</b>	medium
<b>User Dissatisfaction:</b>	medium

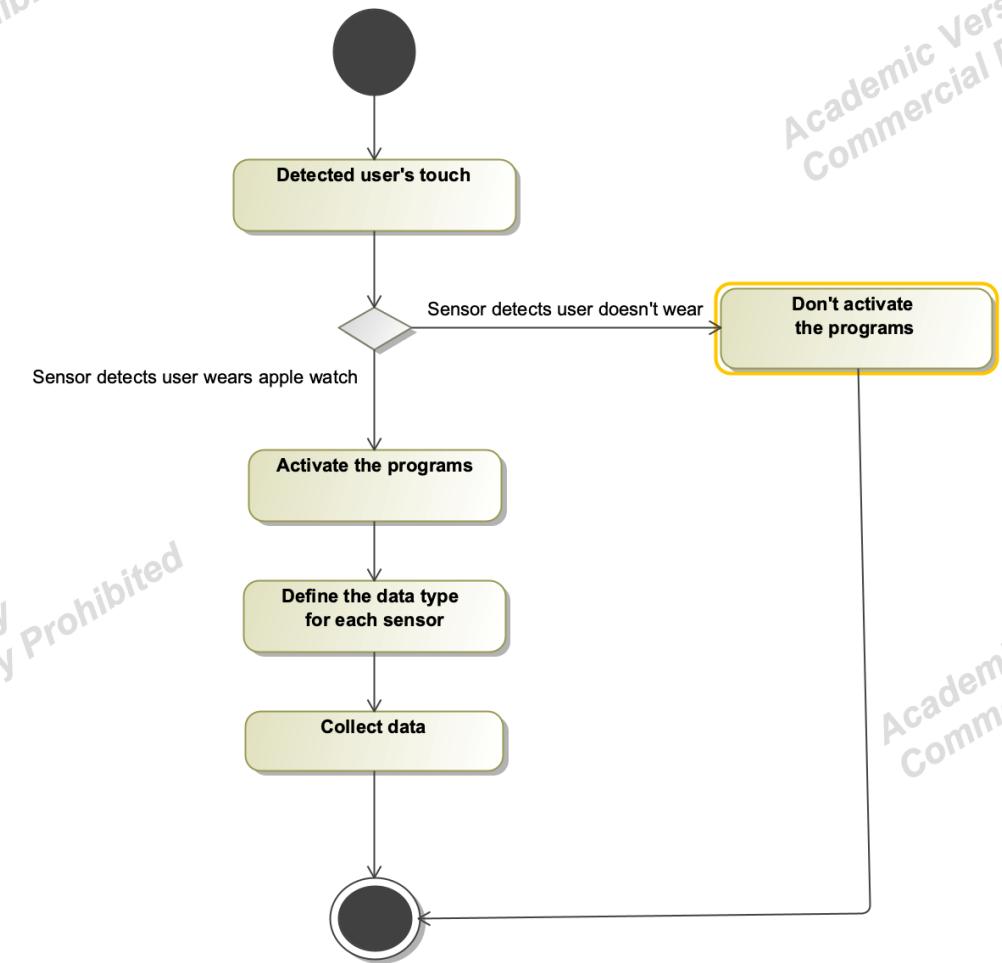
### Description:

- User: wear the apple watch
- System: detect whether user wears the apple watch with thermometer sensor or all the other sensors
  - If the system determines that the user is wearing an Apple watch..
    - System: start to activate sensors and features for measuring data automatically

*Snowcard 43 Activate sensors*



Use case diagram 30 Activate sensors



*Activity diagram 12 Activate sensors*

## 16.2. Definition data type for each sensor & Collecting data

The definition of sensor-specific data collection refers to a clear and specific description of the information gathered by a particular sensor within an environment or system.

The process of collecting data through sensors and the definition of sensor-specific data collection requirements are strongly interrelated, and I have created a diagram depicting their connection.

- Sensor Utilization - System
  - Kind of Sensor
    - Accelerometers
      - Accelerometers measure the acceleration when the user walks or runs, allowing the recognition of changes in walking speed and stride. This information is utilized to understand the user's walking patterns and activity levels.
    - Gyroscopes
      - Gyroscopes identify movements, rotations, and shifts in the user's center of gravity. This functionality aids in accurately tracking and analyzing the user's posture and movements.
    - Thermometer
      - The thermometer measures physiological signals such as heart rate, body temperature, and blood pressure. This data is employed to monitor the user's health status and understand the body's response during physical activities.
    - GPS
      - GPS tracks the user's current location and measures distances for stride calculations. This is valuable for recording exercise routes and measuring overall distance traveled.
  - Definition data collection for each sensor - System
    - Accelerometers: Detect User's walking speed change and stride
    - Gyroscopes: Detect User's posture, rotation change, center of gravity shift
    - Thermometer: Detect Heart Rate, Temperature, Blood Pressure
    - GPS: location, Distance measurement for stride calculation
  - Collecting Data
    - User: Remaining physically active for accurate measurements
    - System: Continuously monitoring sensor data for accurate information, ensuring data integrity and security

*Image 12 Definition data type for each sensor*

## #2: Definition data type for each sensor

<b>Requirement Type:</b>	functional
<b>For Whom?</b>	Management
<b>User Satisfaction:</b>	medium
<b>User Dissatisfaction:</b>	medium

### Description:

- Accelerometers: Detect User's walking speed change and stride
- Gyroscopes: Detect User's posture, rotation change, center of gravity shift
- Thermometer: Detect Heart Rate, Temperature, Blood Pressure
- GPS: location, Distance measurement for stride calculation

*Snowcard 44 Definition of data type for each sensor*

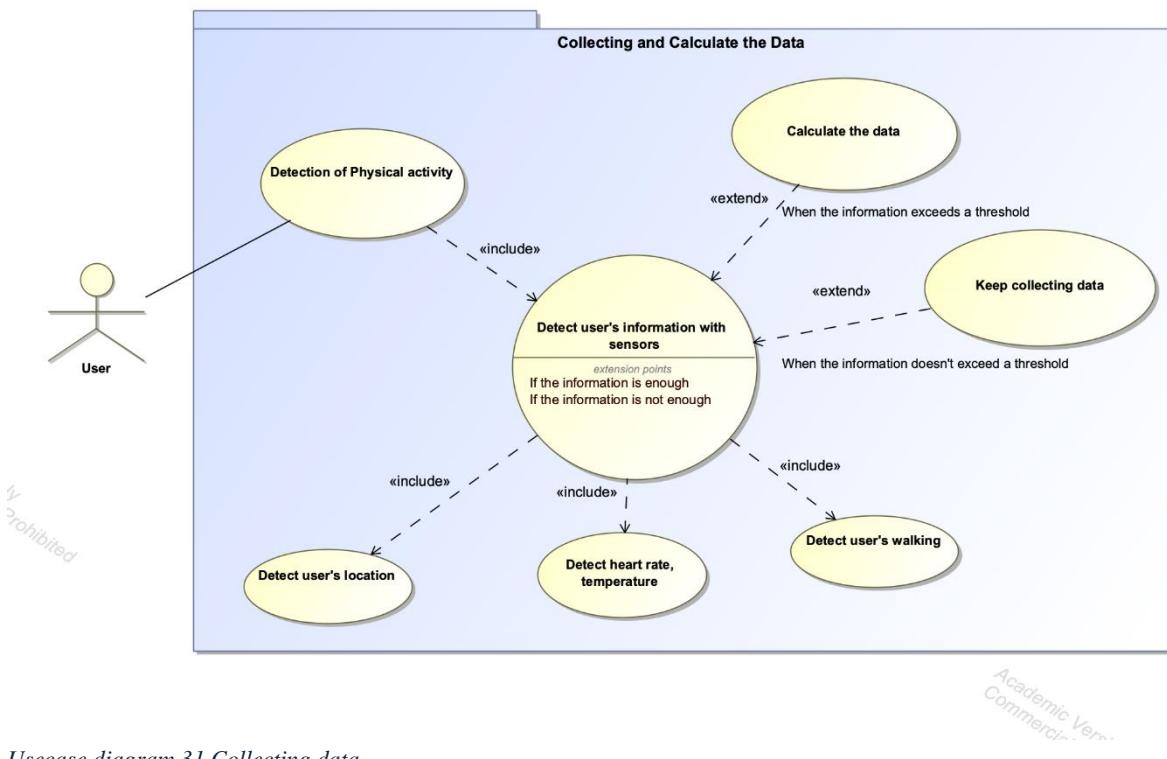
## #3: Collecting Data

<b>Requirement Type:</b>	functional
<b>For Whom?</b>	User, management
<b>User Satisfaction:</b>	medium
<b>User Dissatisfaction:</b>	medium

### Description:

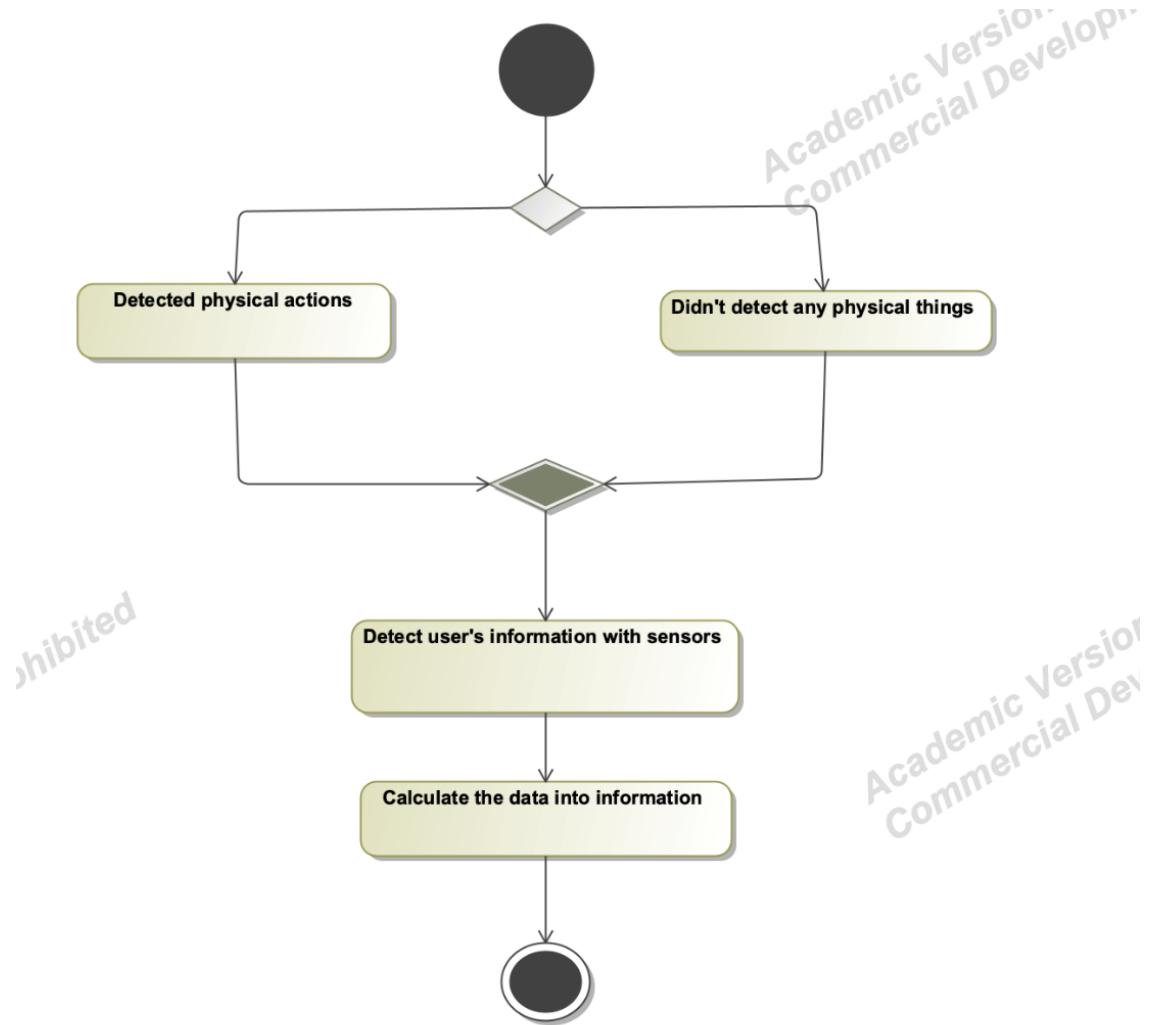
- User: Remaining physically active for accurate measurements ex)walking or running
- System: Continuously monitoring sensor data for accurate information and ensuring data integrity and security

*Snowcard 45 Collecting data*



Use case diagram 31 Collecting data

*Academic Version  
Commercial Version*



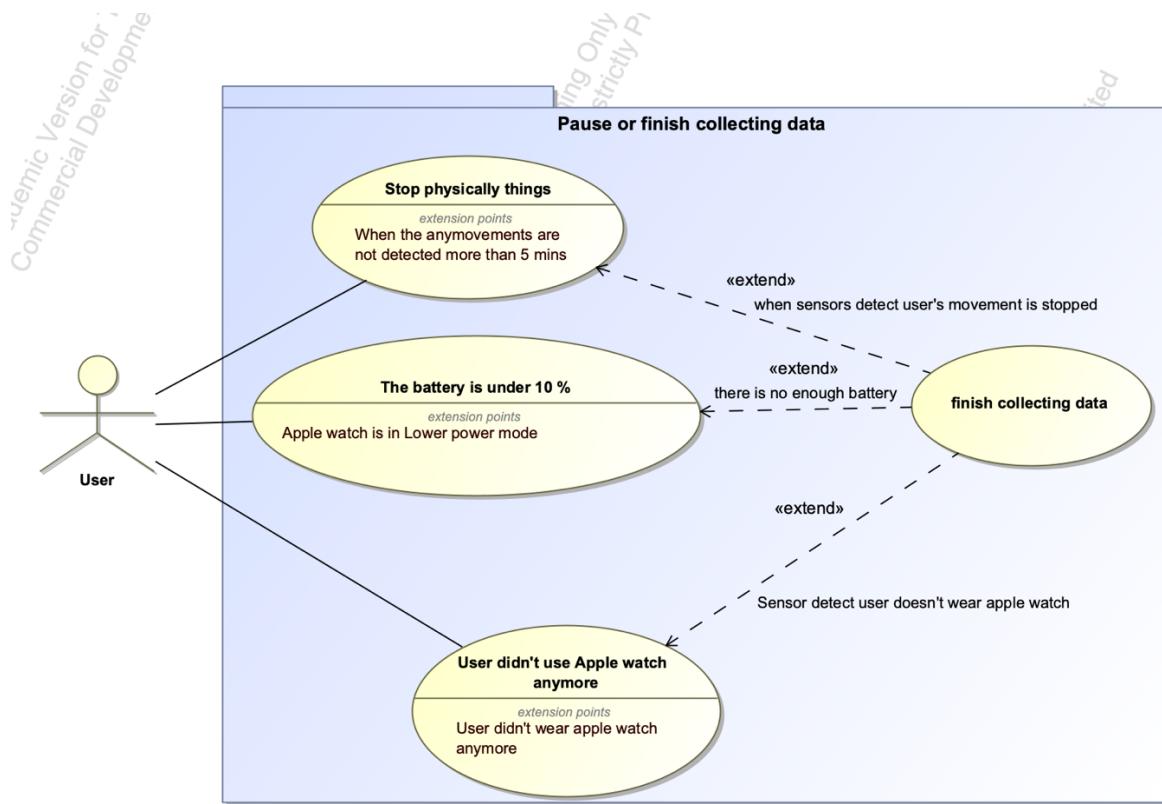
*Activity diagram 13 Collecting data*

### 16.3. Pause or finish collecting data

As the sensors of the Apple Watch are constantly operational, I have delineated situations where data collection through the sensors ceases, considering perspectives from the user, sensor, and system.

- When the any movements are not detected more than 5mins
- When the battery is under 10% or in Low-power mode
- When the anomaly things are detected
- If User doesn't use Apple watch anymore - detect with sensor (thermometer)
  - Automatically
    - System: Stop measuring automatically
    - Web: Displaying an alert to inquire whether to continue or not
    - User: Press Yes or No
      - Yes
        - System: Restart measuring
      - No
        - System: Stop measuring

*Image 13 Pause or finish collecting data*



*Use case diagram 32 Pause or finish collecting data*

# 17. Compatibility

## 17.1. Data Synchronization, Backup

Synchronization and backup maintain data consistency between apps and the cloud through real-time or periodic data synchronization. It also introduces a secure backup mechanism to prevent data loss and secure critical information. Automating synchronization allows you to quickly combine and analyze existing and current information.



### #: Data Synchronization & Back up

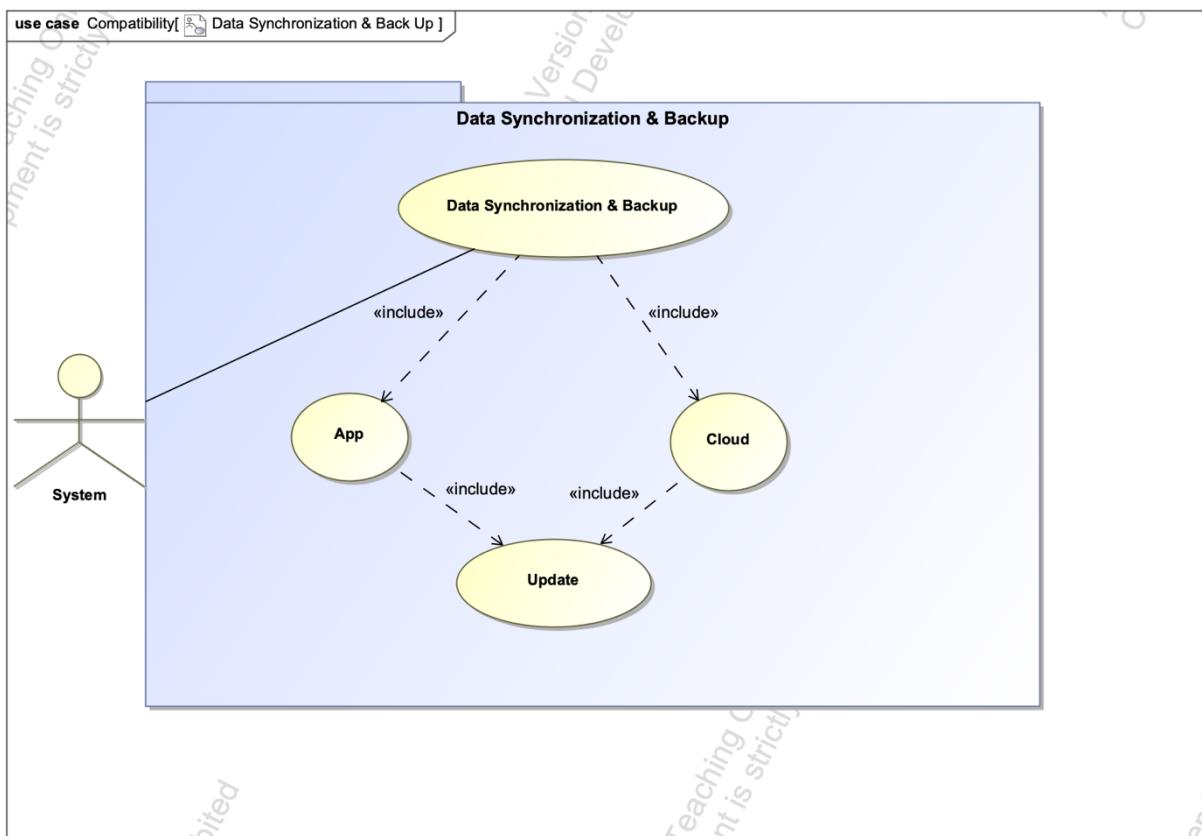
<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	customer/contractor/management
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

#### Description:

It helps maintain App and Cloud system. The customer don't have to update the information whenever they want. And the contractor don't have to renew cloud and app.

Software Engineering Analysis

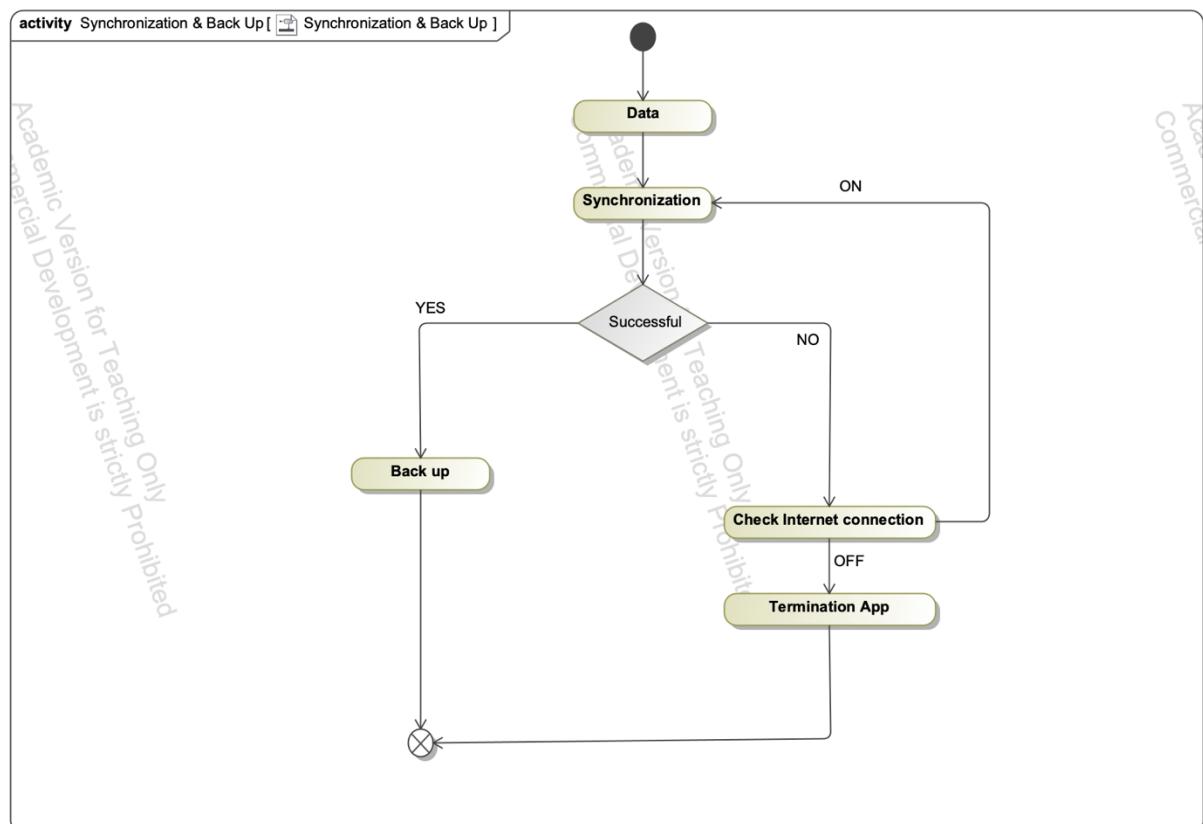
*Snowcard 46 Data synchronization and Backup*



Usecase diagram 33 Data synchronization and Backup

Name		Data Synchronization & Back up
ID		
Description		This is important part of managing data
Trigger		The user connect App.
Actors		System
Pre-conditions		Internet connection
Post-conditions		User data will be synchronization
Basic Flow		
	Description	The user may check that his or her information has been updated and accumulated.
	Actions	
	1	The authentication was granted
	2	Synchronization and back up will be start automatically.
Alternative Flow		A
	Description	Network stopped during data transfer
	Actions	
	1	Synchronization and back up will be delayed.
	2	proceed again when the network connection is stable
Alternative Flow		
	Description	
	Actions	
	1	
	2	

Description of Usecase 20 Data synchronization and Backup



Activity diagram 14 Data synchronization and Backup

## 17.2. Authentication and Security

Enhance user and data security by implementing secure authentication and authorization mechanisms. Secure communication is guaranteed using security authentication protocols such as JWT. JWT allows you to authenticate users and verify that the request is valid between servers, users, and apps

## #: Authentication and Security

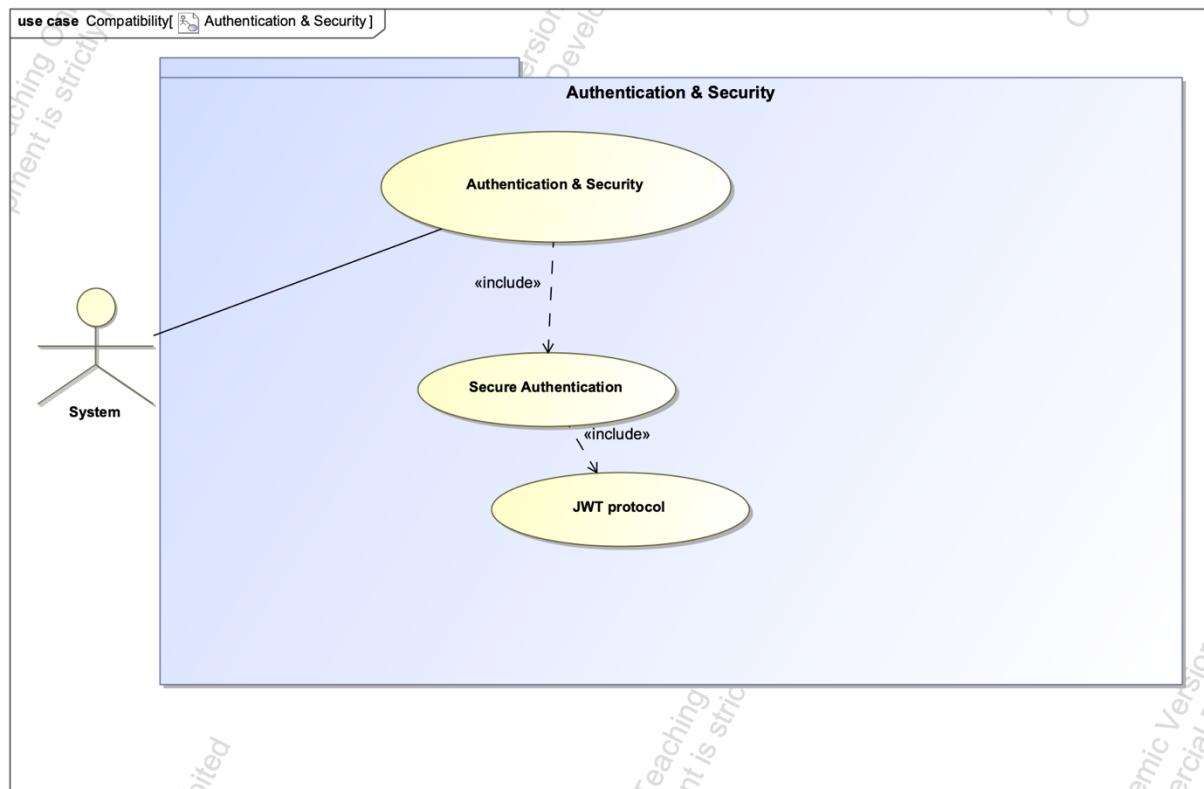
<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	customer/management
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

### Description:

This feature enhances user and data security by implementing secure authentication and authorization mechanisms. Additionally, it guarantees safety protocols such as JWT.

Software Engineering Analysis

Snowcard 47 Authentication and security



Usecase diagram 34 Authentication and security

<b>Name</b>		Authentication and security
<b>ID</b>		
<b>Description</b>		This is important part of connecting app
<b>Trigger</b>		The user turn on the App.
<b>Actors</b>		System
<b>Pre-conditions</b>		Internet connection
<b>Post-conditions</b>		Authentication will be granted
<b>Basic Flow</b>		
	<b>Description</b>	The user get authentication
	<b>Actions</b>	
	1	App started
	2	Permissions are granted when user information matches.
<b>Alternative Flow</b>		A
	<b>Description</b>	The user can not get authentication
	<b>Actions</b>	
	1	Permissions are not granted when user information not matches.
	2	The customer need to log in
<b>Alternative Flow</b>		
	<b>Description</b>	
	<b>Actions</b>	
	1	
	2	

*Description of Usecase 21 Authentication and security*

### 17.3. Use MBaaS (Mobile Backend as a Service)

MBaaS improves development productivity and manages backend services efficiently. It also shortens development time by receiving basic functions such as user authentication, data storage, and push notifications from the cloud. It is also easy to access external services or data using APIs and provides an interface.

## #: Use MBaaS

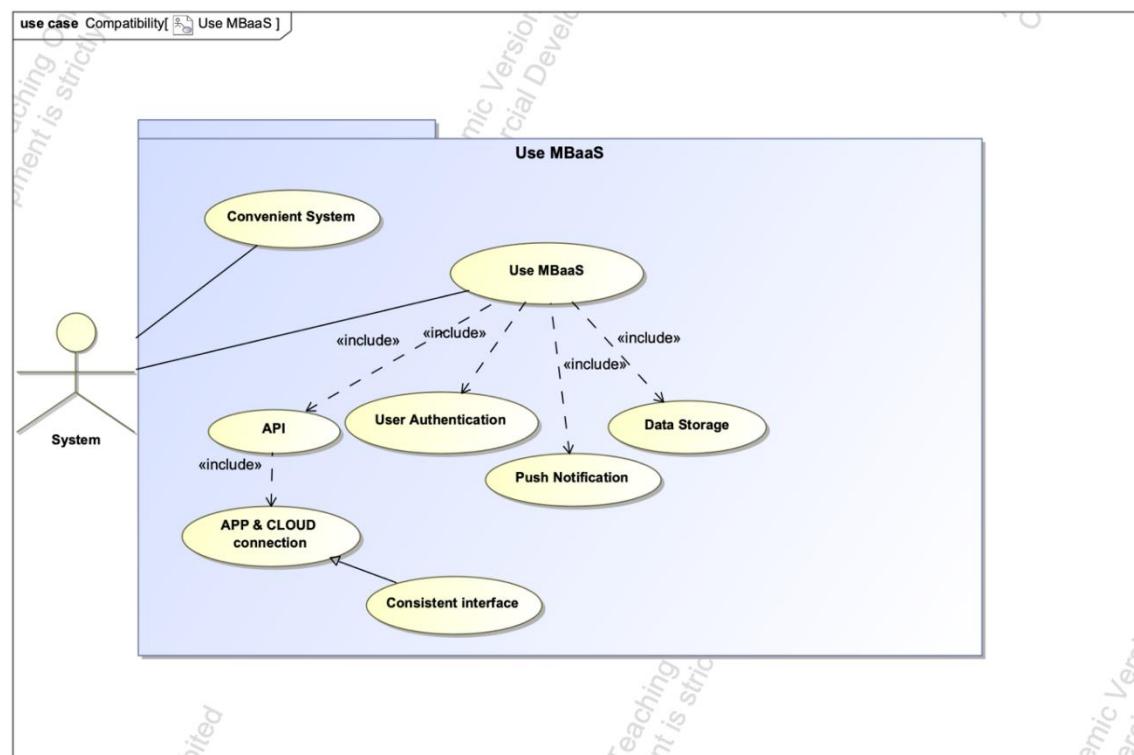
<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	customer/contractor/management
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

### Description:

It utilizes **MBaaS** to improve development productivity and manage backend services efficiently. It also accelerates development time by receiving basic features from the cloud, such as user authentication, data storage, and push notifications. So the customer can get a convenient service.

Software Engineering Analysis

Snowcard 48 Use MBaaS



Usecase diagram 35 Use MBaaS

<b>Name</b>	Use MBaaS
<b>ID</b>	18
<b>Description</b>	It make it Intuitive and convenient
<b>Trigger</b>	The customer connects to the App
<b>Actors</b>	System
<b>Pre-conditions</b>	Normal operation of the server
<b>Post-conditions</b>	Save data, Push notification, Analysis data
<b>Basic Flow</b>	<p><b>Description</b> The app detects a connection</p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li>1 API check their cloud</li> <li>2 Backend check their data storage</li> <li>3 Send related notification</li> <li>4 Show consistent interface</li> </ul>
<b>Alternative Flow</b>	<p><b>Description</b></p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li>1</li> <li>2</li> </ul>

*Description of Usecase 22 Use MBaaS*

## 17.4. Cost and Scalability Considerations

Carefully plan and budget the costs of using cloud services. Understand the cost model of cloud providers and design a structure that can effectively expand resources when needed. If the cost of using cloud services decreases, the cost that customers have to pay will also be further reduced.

## #: Cost and Scalability Considerations

<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	contractor/management
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

### Description:

It will help for reducing cost. It also understand the provider's model and make sure how can the contractor get resources effectively.

Software Engineering Analysis

Snowcard 49 Cost and scalability



## 17.5. User Experience Consistency

It implements a responsive design that provides a consistent user experience in response to various screen sizes and resolutions. This allows users to feel the same way no matter what device they connect to. It also maintains a consistent UI/UX design to maintain a comfortable UI/UX design.

## #: User Experience Consistency

<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	customer/management
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

### Description:

Maintain a consistent UI/UX design on each platform for a comfortable experience for users.

Software Engineering Analysis

*Snowcard 50 User experience consistency*



## 17.6. Battery Managing

Introduce energy-efficient algorithms to minimize battery consumption. Use apps or watch's apps to schedule so that you don't consume too much battery. The above algorithm will help you schedule, and you can increase the usage time of your device.



### #: Battery Managing

<b>Requirement Type:</b>	non-functional
<b>For Whom?</b>	customer/contractor/management
<b>User Satisfaction:</b>	high
<b>User Dissatisfaction:</b>	low

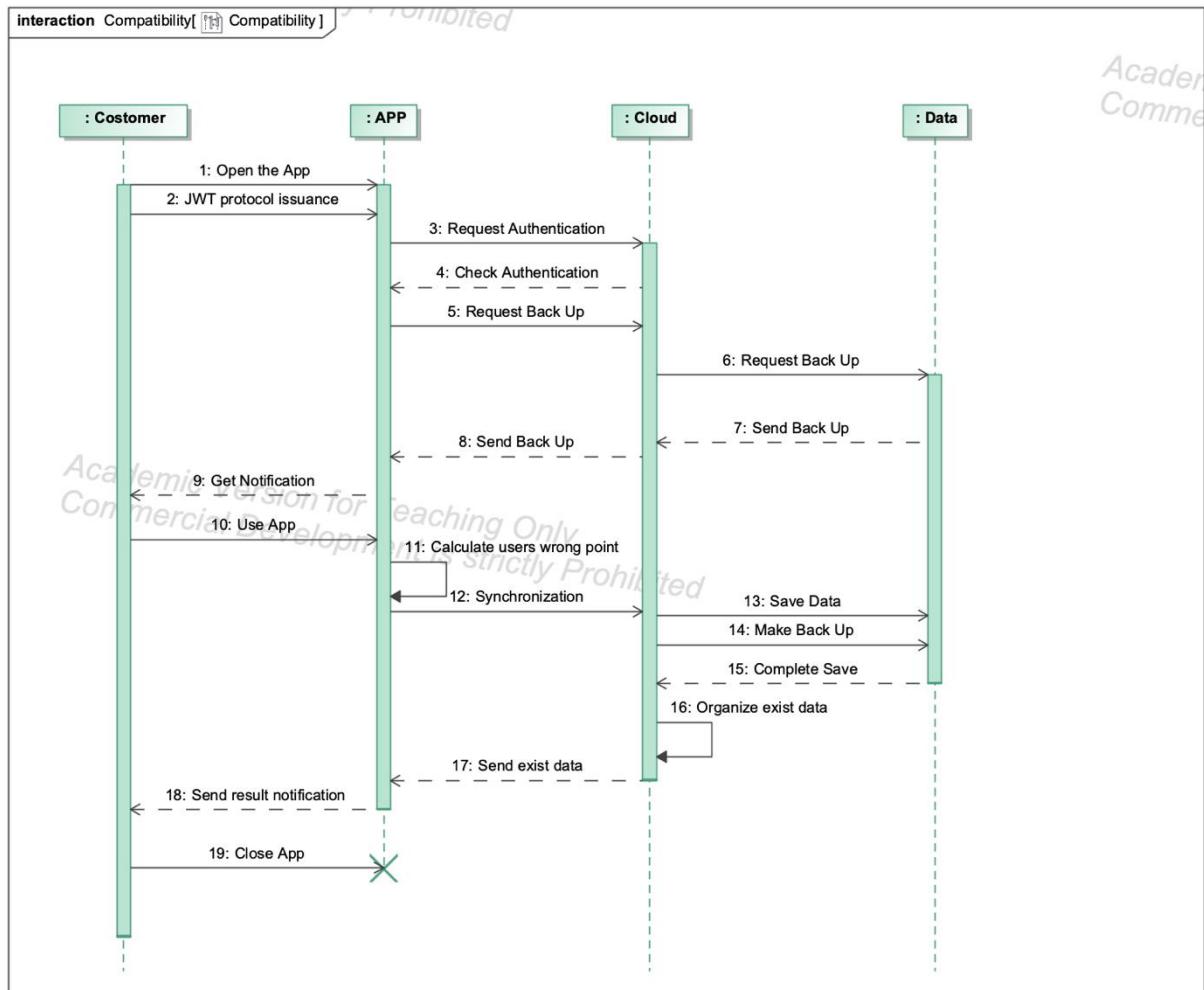
#### Description:

It introduces energy-efficient algorithms to minimize battery consumption. And it performs careful battery management and optimization for battery-intensive tasks. The customer increases device usage by minimizing battery consumption. And improve the user experience with clear information about battery status. The contractor can enhance efficiency to make the app's battery consumption competitive.

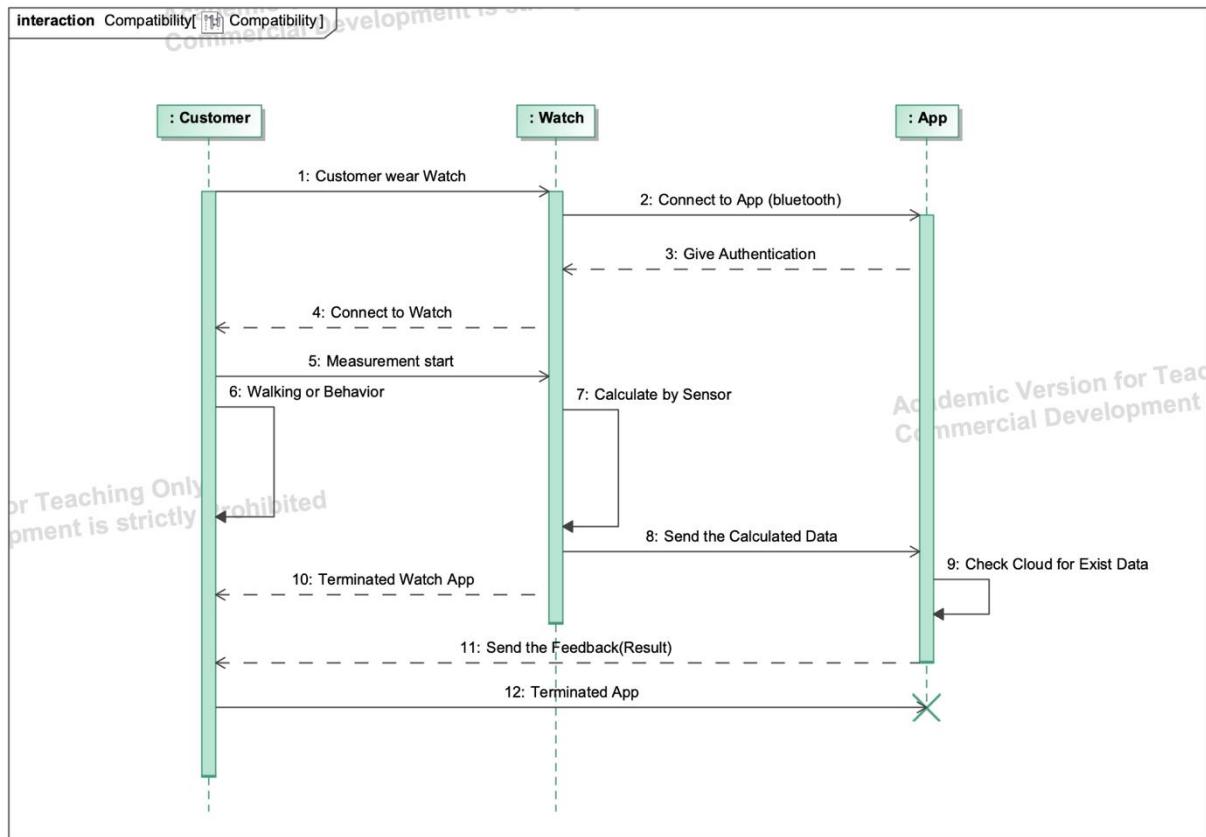
Software Engineering Analysis

*Snowcard 51 Battery managment*

## 17.7. Sequence Diagram



Sequence diagram 18 Compatibility(I)



Sequence diagram 19 Compatibility(2)

# 18. Security

## 18.1. Access control

It prevents and keeps safe the leakage of personal information that may occur when a user accesses an app, and modifies unauthorized data, and transmits other people's data to feedback sent.

  
FRANKFURT  
UNIVERSITY  
OF APPLIED SCIENCES

### #2-1: Access Control

<b>Requirement Type:</b>	<b>non-functional</b>
<b>For Whom?</b>	Customer
<b>User Satisfaction:</b>	<b>high</b>
<b>User Dissatisfaction:</b>	<b>low</b>

**Description:**  
Access Control is a security mechanism that controls and manages access to resources (such as files, data, and programs) within an information system.

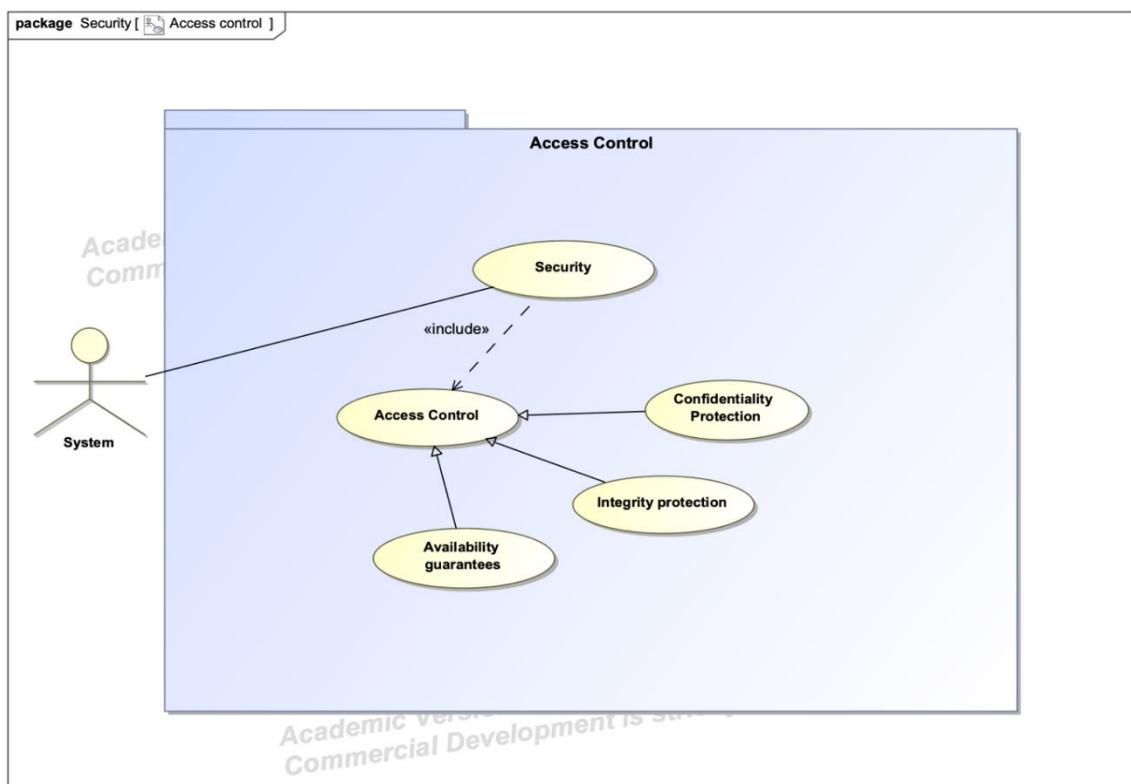
Confidentiality Protection: Protects confidentiality by preventing unauthorized access to sensitive information.

Integrity protection: Protects the integrity of your data by preventing sensitive information from being modified without permission.

Availability guarantees: Ensures the availability of the system by allowing authorized users access to the resources they need.

Software Engineering Analysis

Snowcard 52 Access control



Usecase diagram 36 Access control

Name		Access control
ID		19
Description		Protect user's information
Trigger		
Actors		System
Pre-conditions		Internet connection
Post-conditions		Protect user's data
Basic Flow		
	Description	The user receive data from the system or send it to the system
	Actions	
	1	The user data changed
	2	Data is downloaded or sent
Alternative Flow	A	
	Description	Confidential Protection
	Actions	
	1	An unknown user or unauthorized user accesses the data.
	2	Block the connection itself.
Alternative Flow	B	
	Description	Integrity protection
	Actions	
	1	Request for modification of information.
	2	The user's personal information is overseas and the right to modify is restricted.
Alternative Flow	C	
	Description	Availability guarantees
	Actions	
	1	The server sends a weekly summary.
	2	Verify that the recipients match.

Description of Usecase 23 Access Control

## 18.2. Encryption

Encryption enables data to be encrypted and unreadable, and only when the correct key is used, thereby protecting data by building a stronger and more robust cryptosystem.



### #2-2: Encryption

**Requirement Type:** non-functional

**For Whom?** Customer

**User Satisfaction:** high

**User Dissatisfaction:** low

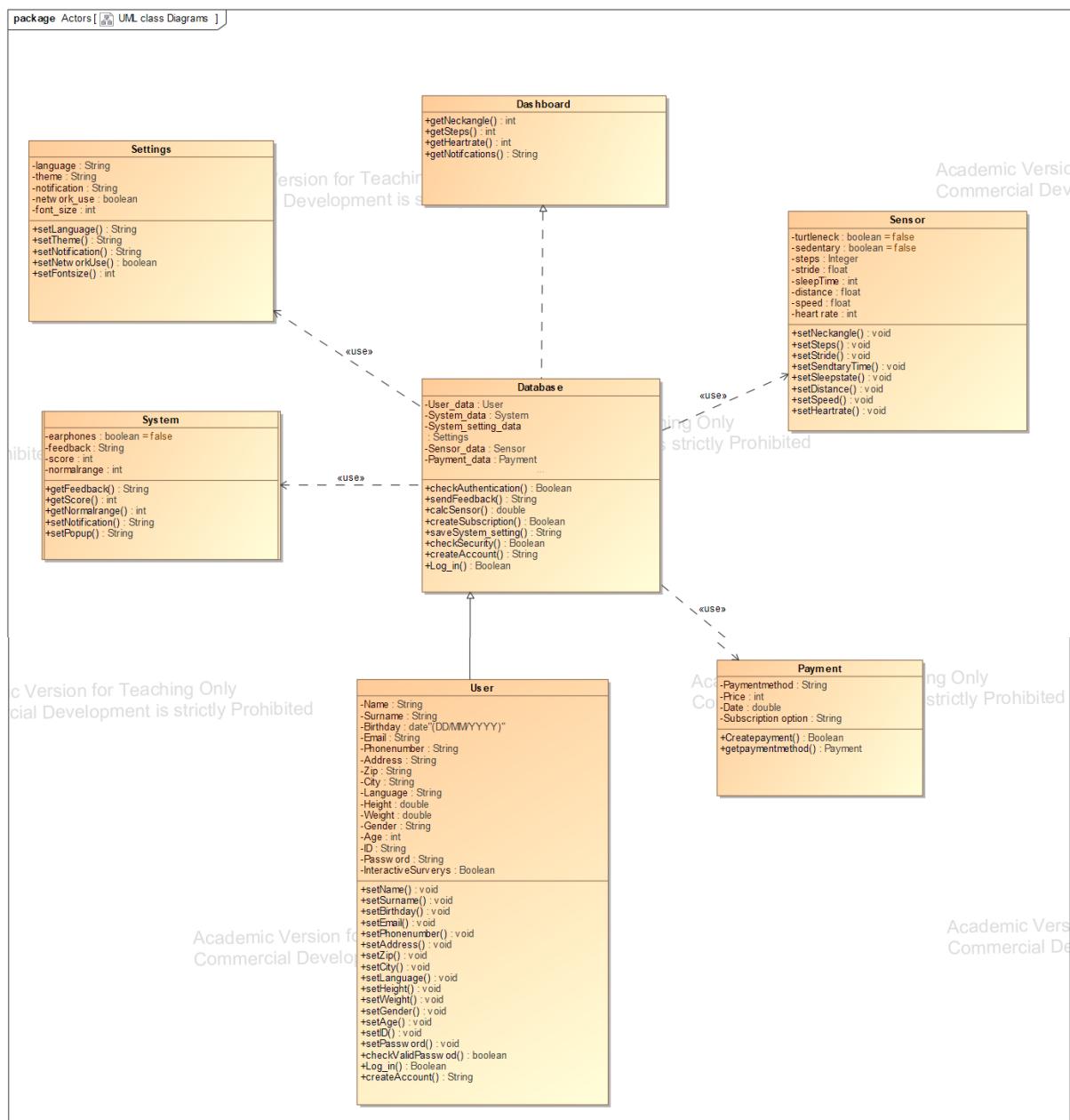
#### Description:

Encryption is the process of encrypting data to enhance security. Encryption protects data by converting it into an illegible form and allows it to be decrypted only with the correct key. It is mainly used in storing or transmitting data, and plays an important role in protecting the confidentiality and integrity of data. All transmitted data is encrypted and protected. This process completely protects all transmitted data.

Software Engineering Analysis

Snowcard 53 Encryption

# 19. UML Class Diagram



Class diagram 1

## 20. UI/UX prototype

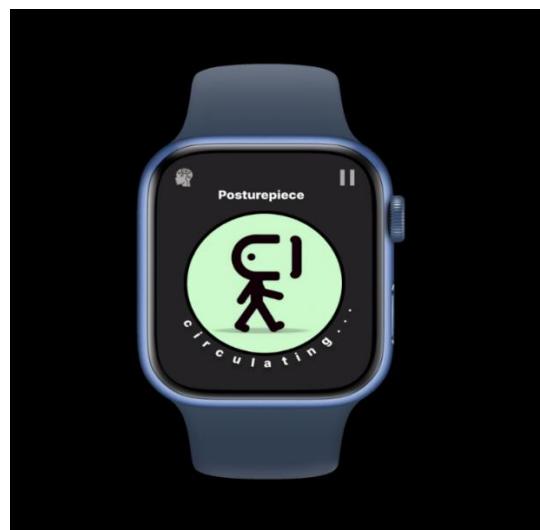


Image 14 App Icon prototype  
Image 15 UI/UX prototype

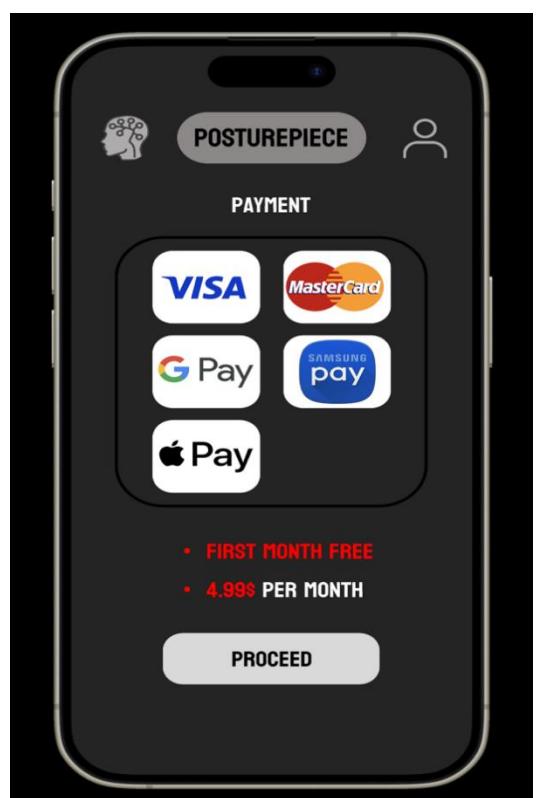
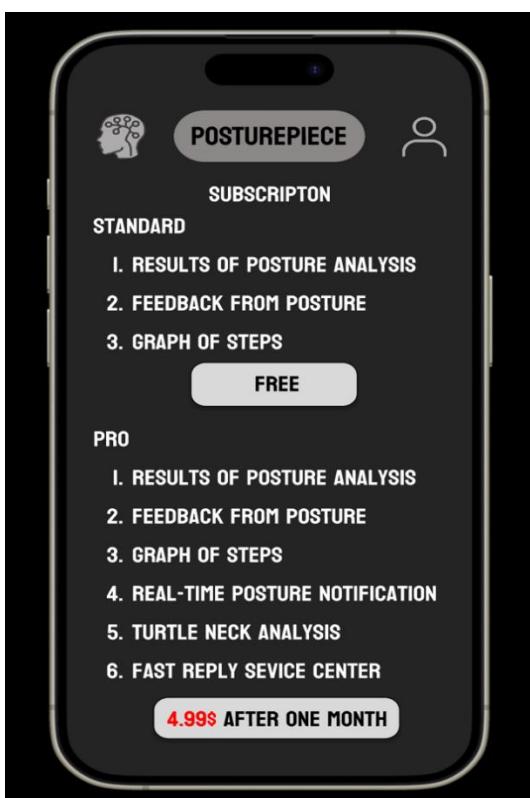


Image 16 UI/UX Payment (1)  
Image 17 UI/UX Payment (2)

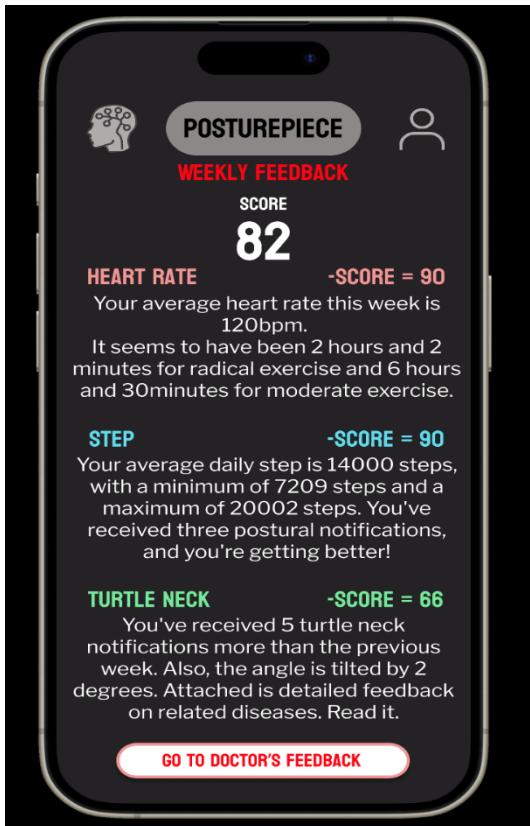


Image 18 UI/UX Feedback(1)



Image 19 UI/UX Feedback(2)

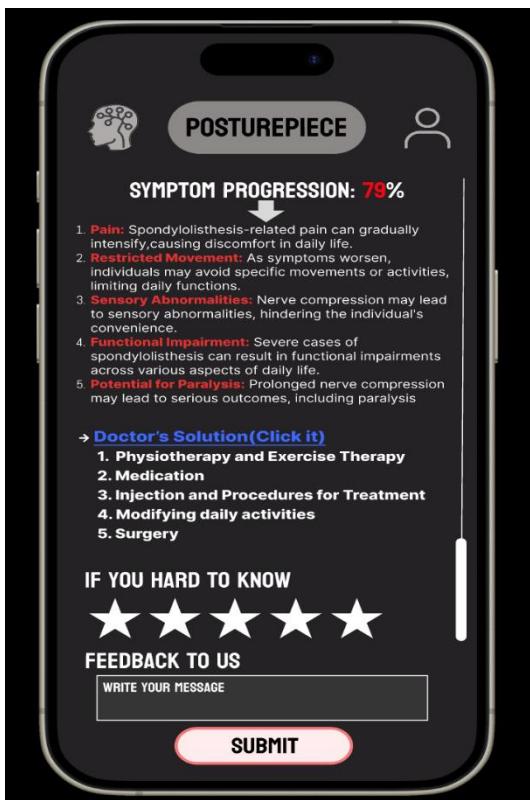


Image 20 UI/UX Feedback(3)

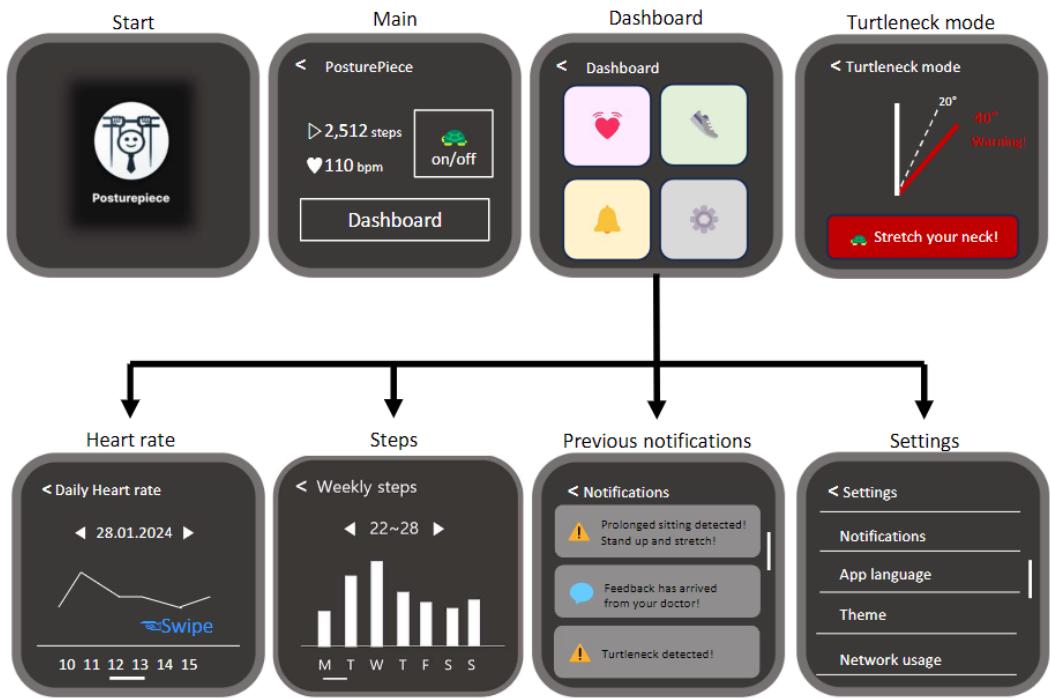


Image 21 UI/UX Entire flow

## 21. Summary

Our software is developed to promote users' healthy lifestyle habits. It analyzes users' heart rate, step count, and spinal angle captured by smartwatches to identify posture issues and provide personalized feedback. Through this, users can better understand and improve their walking habits and overall health status. Key features of our software include customer registration/login, dashboard functionality, heart rate monitoring, step tracking, neck posture mode, expert feedback, other useful features, payment/subscription options, usability improvements, real-time data collection and logging, compatibility assurance, and UI/UX prototype development.

Our goal is to foster healthy habits through personalized feedback, leading to positive changes in users' daily lives and enhancing their overall health and well-being. Additionally, we plan to enhance the user-customized posture training aspect of our software by providing tailored training plans and improving posture analysis accuracy. These efforts aim to advance our software for actual release, ensuring user satisfaction and effective posture improvement.