[CS350] 2021 Fall Team Project
Project #2. Software Requirements Specification (SRS)

Smart Tracker

Team 12
20160509 이정재
20180155 김준범
20180265 박윤정
Submission Date: 2021-10-17

Table of Contents

I. Introduction

Smart Tracker is the product we introduce to prevent users from losing their important properties. This idea was suggested for the people who lose their possessions easily and are forgetful, or people who possess important belongings, which every person could likely be part of. The minimal Smart Tracker could be easily attached to the important, and the users would be able to keep an eye on their belongings anytime by checking the application connected to the tracker.

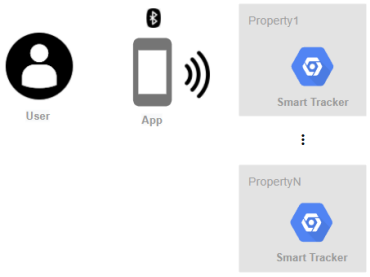
The most simple use case of the tracker would be checking the location of the tracker using GPS or buzzer. By choosing the tracker to utilize by using the connected application, users would know where the object they are looking for is by sounds or visualized location from the tracker attached to it. In addition to these two use cases, turning on or off the tracker, giving warnings as the tracker gets further or closer, checking the battery and charging the battery would be important use cases our team will be aware of.

This paper covers the main ideas of the software requirements to implement Smart Tracker. We especially focused on six types of management: communication management for the connection between the smartphone application and the tracker, battery management to use the tracker for a long time, tracking management for the implementation of location of the tracker, device management for the users to manage the tracker device itself, usability management helping users to easily get familiar to using the tracker, and security management for the privacy issues.

II. Overall Description

A. Product Perspective

The main idea and context of the product is to make our properties trackable. The origin of this was a small tracking device that only made a sound when the user got close. So if the user gets near the lost object the device makes a sound and the user is able to find where it is by listening to it. However, when we lose something it may not always be in the connection range which makes this kind of tracker useless. Therefore to help users in these kinds of situations, we combined the tracker with GPS and created many more features such as showing on a map. This smart tracker system is mainly composed of a smartphone app and tracker device.



<Figure A.1> Major components of the overall system

B. Product Features

This product provides a solution for people who frequently forget their belongings. After attaching a tracker device to their belongings, users can receive alerts from a smartphone application when the distance between the smartphone and tracker device gets further away. They can find their belongings by GPS position of the attached tracker devices, or by the ringing sound from the tracker devices.
Users can register their designated tracker devices in their smartphone applications, and manage them from the applications, such as turning on/off certain tracker devices, or checking the remaining battery.

C. Operating Environment

Hardware Platform: Mobile
Supporting OS: iOS(7 or more), AOS(5 or more)

- iOS gets no longer updated after 6 years since it was released, and 2021 is the year after 6 years passed since iOS version 6 was released.
- AOS version 5.20 is a long term supported version.

Bluetooth communication should be available on a mobile device.

D. Assumption and Dependencies

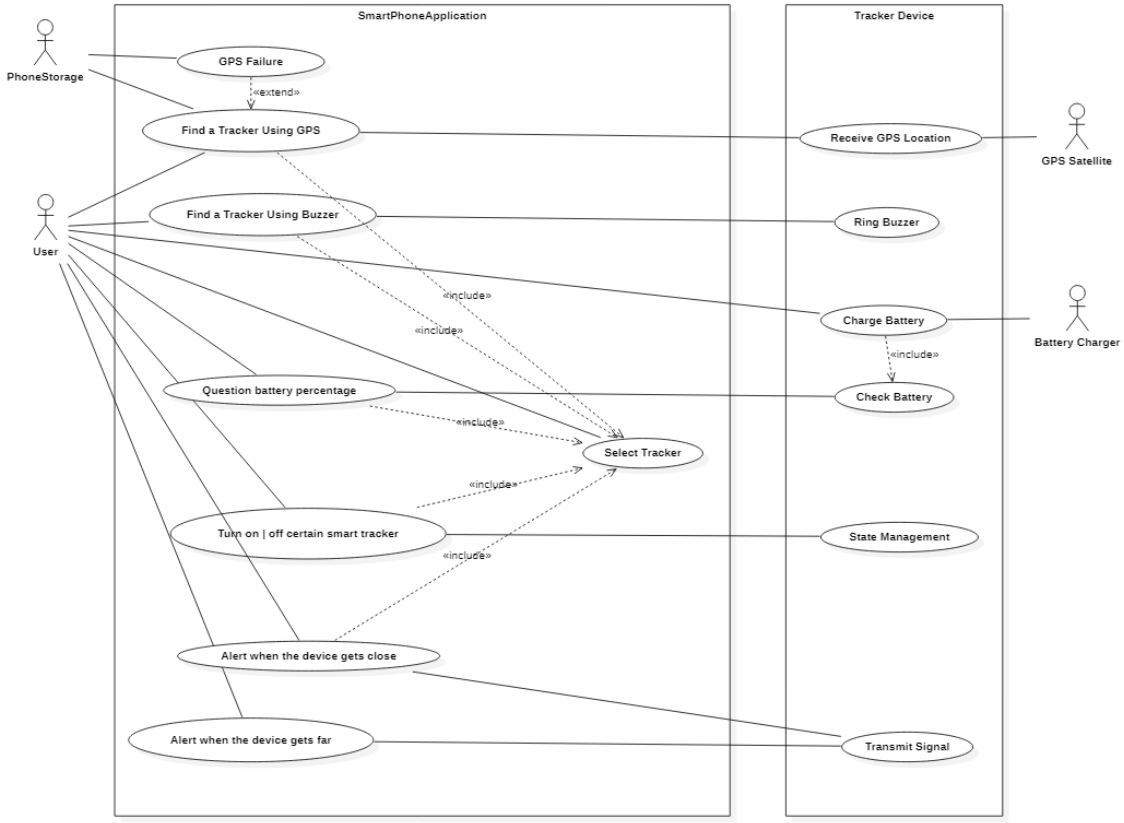
- Bluetooth connection will be available
- GPS will be usable and precise enough
- The battery will last long enough for the connection to be meaningful
- The battery charger will meet the size of the tracker and be compatible to enable the charging function
- Power sensor will be precise to alert the users of the tracker in particular distance
- Power sensor will be able to cover a big range of distance
- There should be enough memory for the application to save information (tracker information, GPS location)
- The materials should be affordable
- The application would be compatible on mobile os (android, iOS)

III. System Features

A. Functional Requirements

Hierarchy	Requirement ID	Requirement Description	Actors	Priority
Communication Management	A_1	Bluetooth connection between tracker and smartphone application.	None	Critical
Communication Management	A_2	A smartphone application should always run in the background and should receive a strength signal from the tracker device for distance measurement.	None	Critical
Battery Management	B_1	Users can charge the device by connecting the charging port with the C-type charger.	User, Battery Charger	Critical
Tracking Management	C_1	Users can make the tracker's buzzer ring for exact positioning.	User	Trivial
Tracking Management	C_2	Users can use the smartphone application to find the exact position of the tracker.	User, GPS satellite	Critical
Tracking Management	C_3	Smartphone application saves the most recent position of the tracker periodically	Smartphone storage	Critical
Device Management	D_1	Users can check the battery status of the trackers using the smartphone application.	User	Trivial
Device Management	D_2	Users can use smartphone application to turn on/off devices	User	Critical
Usability Management	E_1	Users can easily attach or remove the tracker from their property by utilizing tag or ring form.	User	Trivial
Security Management	F_1	Tracker device remembers the first paired smart phone application and only communicates with it.	User	Critical

B. Use Case Diagram & Descriptions



<Figure B.1> Overall Use Case Diagram

Use case name	Select Tracker
Related Requirements	Requirement C.1., Requirement C.2., Requirement C.3., Requirement D.1., Requirement D.2.
Goal in Context	User chooses which tracker to utilize using the smartphone application.
Preconditions	Tracker must be in bluetooth connection range and have enough power.
Successful End Condition	Users operate features on a selected tracker or get information on its status.
Failed End Condition	The connection of the tracker and smartphone does not occur and the user gets an error message
Primary Actors	User
Secondary Actors	None
Trigger	User asks the smartphone application to select a certain tracker.

Main Flow	Step	Action
	1 2 3 4	User selects which tracker they will utilize on the smartphone application. Smartphone application sends a bluetooth signal to the tracker and requests connection. Tracker receives the signal and sends a response back. The smartphone application informs the users that the connection has been established.
Extension		

Use case name	Find Tracker Using GPS	
Related Requirements	Requirement C.2.	
Goal in Context	User receives the precise location of the tracker using GPS.	
Preconditions		
Successful End Condition	User receives information about the location of the tracker on the map.	
Failed End Condition	Tracker device does not send the tracker’s GPS information.	
Primary Actors	User	
Secondary Actors	GPS Satellite, Phone Storage	
Trigger	User asks the smartphone application to load a map that shows where the tracker is.	
Main Flow	Step	Action
	1 2 3 4 5 6 7	User selects which tracker they will utilize. User chooses to see where the tracker is on a map. Smartphone application sends a bluetooth signal to the tracker and requests its location information. After the tracker receives the bluetooth signal from the smartphone, it communicates with the GPS satellite to get its location. Tracker sends a bluetooth signal which contains information about its location. Smartphone application receives the bluetooth signal from the tracker and saves it in smartphone storage. Map which contains the tracker's location is created.
Extension	3.1 3.2	Smartphone application does not receive any bluetooth signal from the tracker Smartphone application utilizes the most recently saved

		location of the tracker to draw a map.

Use case name	Find Tracker Using Buzzer	
Related Requirements	Requirement C.1., Requirement C.2.	
Goal in Context	User receives the location of the tracker by making tracker buzz	
Preconditions	None	
Successful End Condition	The buzzer inside the tracker rings.	
Failed End Condition	The connection of the tracker and smartphone does not occur and the user gets an error message.	
Primary Actors	User	
Secondary Actors	None	
Trigger	User asks the smartphone application to make the tracker's buzzer ring.	
Main Flow	Step	Action
	1 2 3 4	User selects which tracker they will utilize. User chooses to make the tracker's buzzer ring. Smartphone application sends a bluetooth signal to the tracker and requests its buzzer to ring. After the tracker receives the bluetooth signal from the smartphone, it makes the buzzer ring for a certain period.
Extension		

Use case name	Alert when the device and smartphone get further away
Related Requirements	Requirement A.1., Requirement A.2.
Goal in Context	Smartphone application sends alert to user when the signal from the device gets weaker, expecting the user might lose the target belonging.
Preconditions	The device should be attached to the target belonging.
Successful End Condition	User notices that he/she left his/her target belongings behind.
Failed End Condition	The user cannot receive an alert from the smartphone application.

Primary Actors	User	
Secondary Actors	None	
Trigger	The power of the signal between the device and smartphone is under a certain threshold.	
Main Flow	Step	Action
	1 2 3	The user brings the smartphone and gets further away from the target belonging. The smartphone application detects the signal from the smart tracker's transmitter gets weaker so that it falls below a certain threshold. The smartphone application alerts the user in a set way (notification, vibration or alarm sound)
Extension		

Use case name	Turn on/off certain smart trackers' functions.	
Related Requirements	Requirement D.2.	
Goal in Context	Users should be able to manage multiple devices. Users can choose which device to activate to prevent unnecessary alerts from not using devices.	
Preconditions	Smart tracker devices should be properly registered on the smartphone application.	
Successful End Condition	The smartphone application will not alert for smart tracker devices which are turned off, but still keep tracking on all devices. The smartphone application marks the target device as "turned off" in its device list. The target device changes its state to standby mode.	
Failed End Condition	The smartphone application should not mark the target device as "turned off" and alert if the distance from the target device is over threshold. The target device should not change its state.	
Primary Actors	User	
Secondary Actors	None	
Trigger	The user executes the smartphone application and requests to turn on/off a certain smart tracker device.	
Main Flow	Step	Action
	1	The user uses a smartphone application to request to turn on/off a certain device

	2 3 4 5	The smartphone application sends a request to the target smart tracker device. The target tracker device sets its state to “turned on” / ”turned off” The smartphone application receives response from the target tracker device through bluetooth communication. The smartphone application shows the user the target device is “turned on” / ”turned off”.
Extension	2.1 2.2	The smartphone application does not receive any message from the target device. The smartphone application shows a connection loss error message to the user.

Use case name	Check the remaining battery	
Related Requirements	Requirement D.1.	
Goal in Context	User requests the battery leftover percentage from the smartphone application.	
Preconditions	The tracker should be connected to the user’s smartphone application.	
Successful End Condition	The smartphone application sends the current percentage of the remaining battery successfully to the user.	
Failed End Condition	The application for the current battery is rejected.	
Primary Actors	User	
Secondary Actors	None	
Trigger	The user asks for the remaining battery of the tracker.	
Main Flow	Step	Action
	1 2 3 4	The user selects the tracker to check the battery. The application demands the tracker to check the remaining battery The application receives the percentage from the tracker. The user earns the numerical value.
Extensions	2.1 2.2	The application fails to get connected to the tracker. The application sends an error message back to the user.

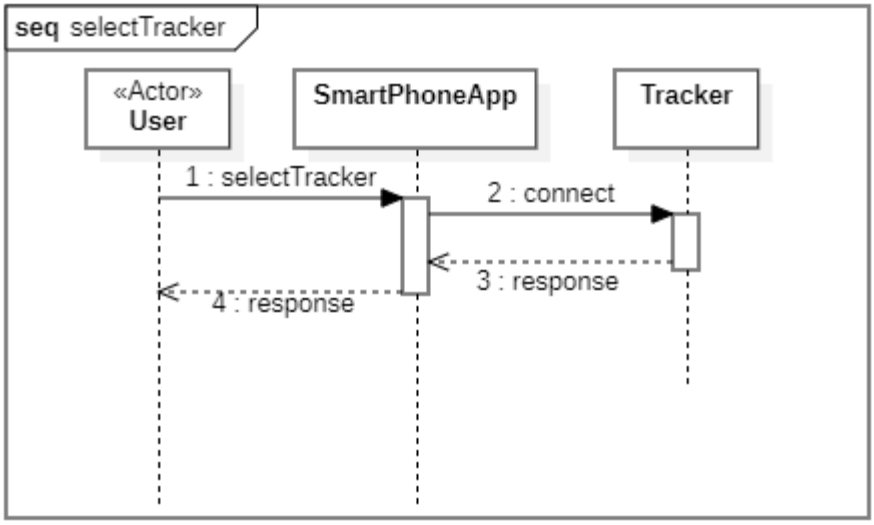
Use case name	Charging the battery
Related Requirements	Requirement B.1.

Goal in Context	User charges the battery of the tracker.	
Preconditions		
Successful End Condition	The tracker gets fully charged.	
Failed End Condition	The battery percentage of the tracker does not change after charging.	
Primary Actors	User, Battery Charger	
Secondary Actors	None	
Trigger	The user connects the charger to the charging port of the tracker.	
Main Flow	Step	Action
	1 2 3 4	User connects the charger to the charging port of the tracker. The charger charges the embedded battery of the tracker. After the battery is charged 100%, the BCS gives a signal to the transmitter. The transmitter gives a signal to the smartphone application that the charging is done.
Extensions	2.1 2.2 2.3	The connection between the charger and the tracker is unstable. The charger sends an error signal to the smartphone application. The application shows an error message to the user.

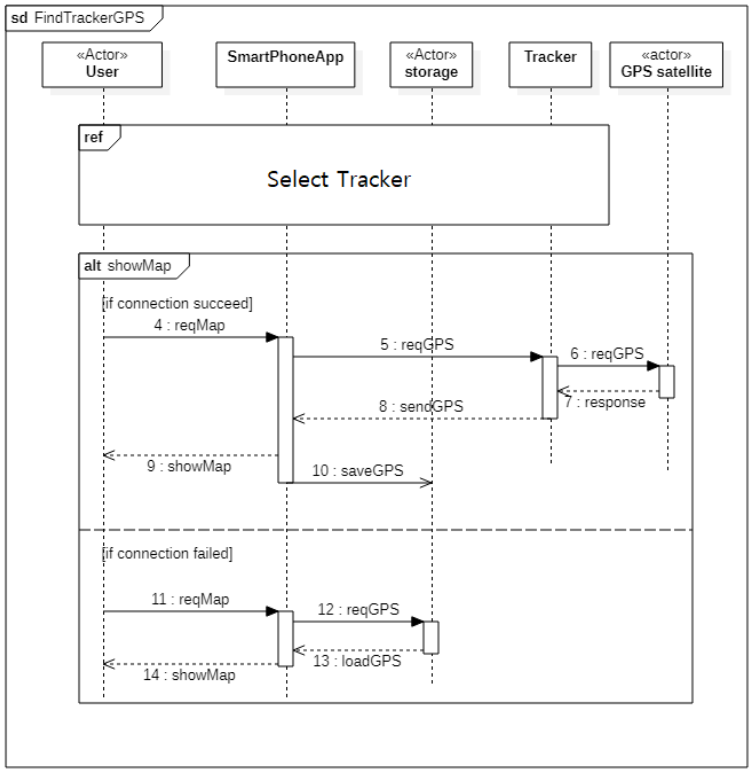
Use case name	Give signal when the tracker gets close
Related Requirements	Requirement A.2.
Goal in Context	When the tracker gets closer than a certain distance, the smartphone application gives a signal.
Preconditions	The smartphone application should be connected to the tracker.
Successful End Condition	Signal appears in the application.
Failed End Condition	The signal does not appear even though the tracker is right next to the user.
Primary Actors	User
Secondary Actors	None

Trigger	The tracker gets closer to the user (and his/her smartphone)	
Main Flow	1 2 3 4 5	The user selects the tracker to keep track of. The tracker gets close. Power sensor detects the power stronger than the chosen limit at a certain point. The tracker sends a signal to the smartphone application via bluetooth device. The smartphone application notifies the user that the tracker got close. (Buzzer)
Extensions	1.1	If the connection fails the application shows the error message to the user.

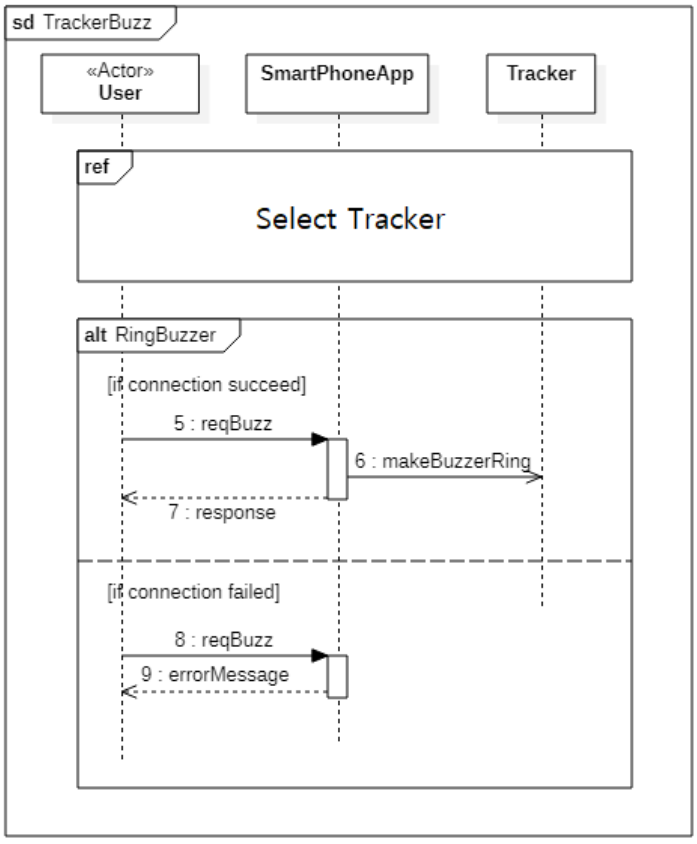
C. Sequence Diagrams



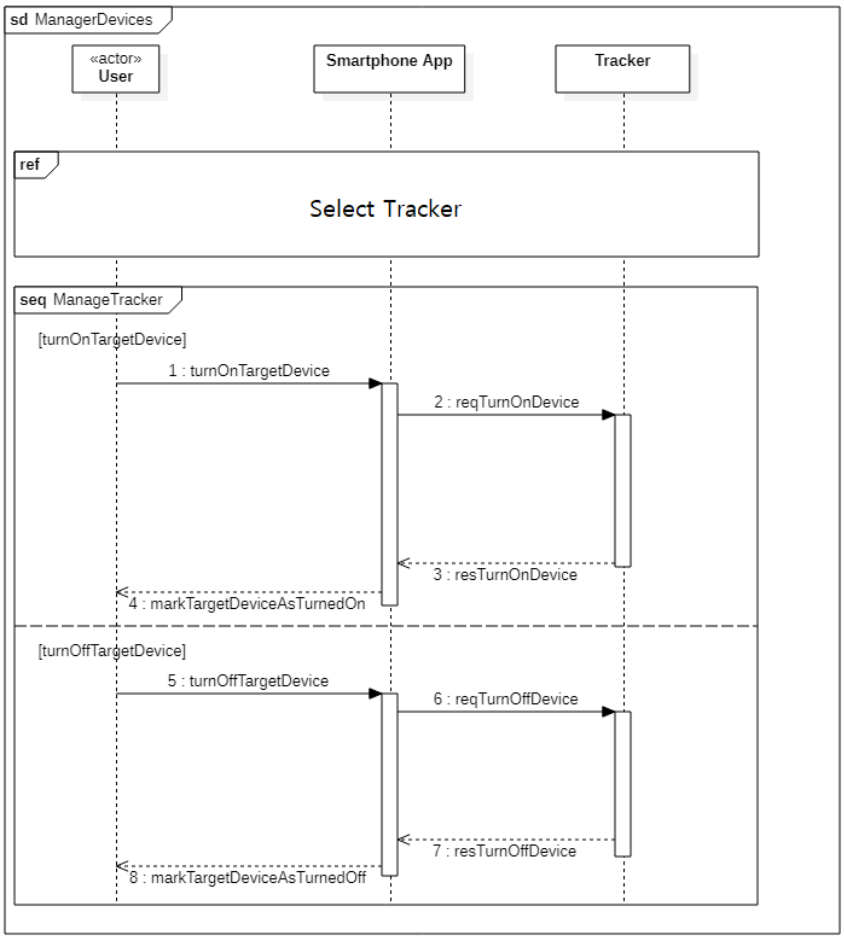
<Figure C.1> Sequence diagram of ‘Select tracker’ use case



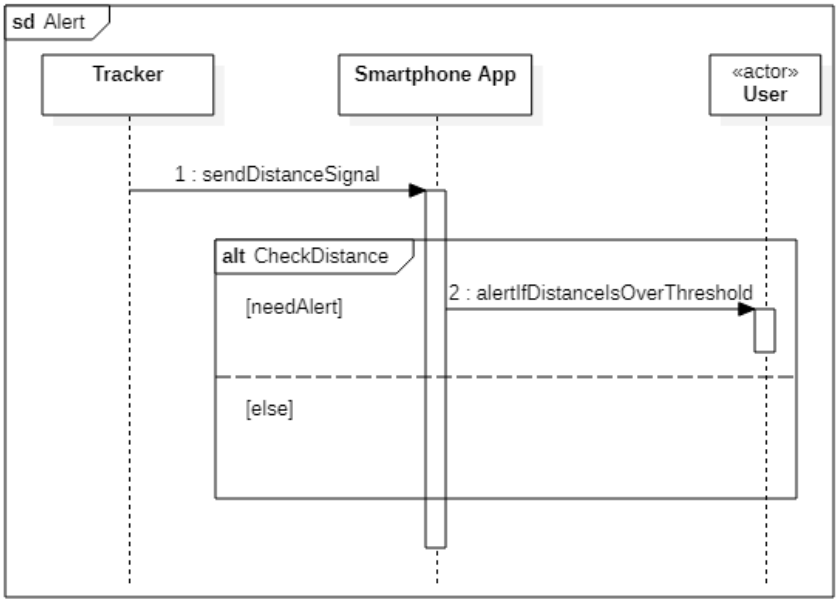
<Figure C.2> Sequence diagram of 'Find tracker using GPS' use case



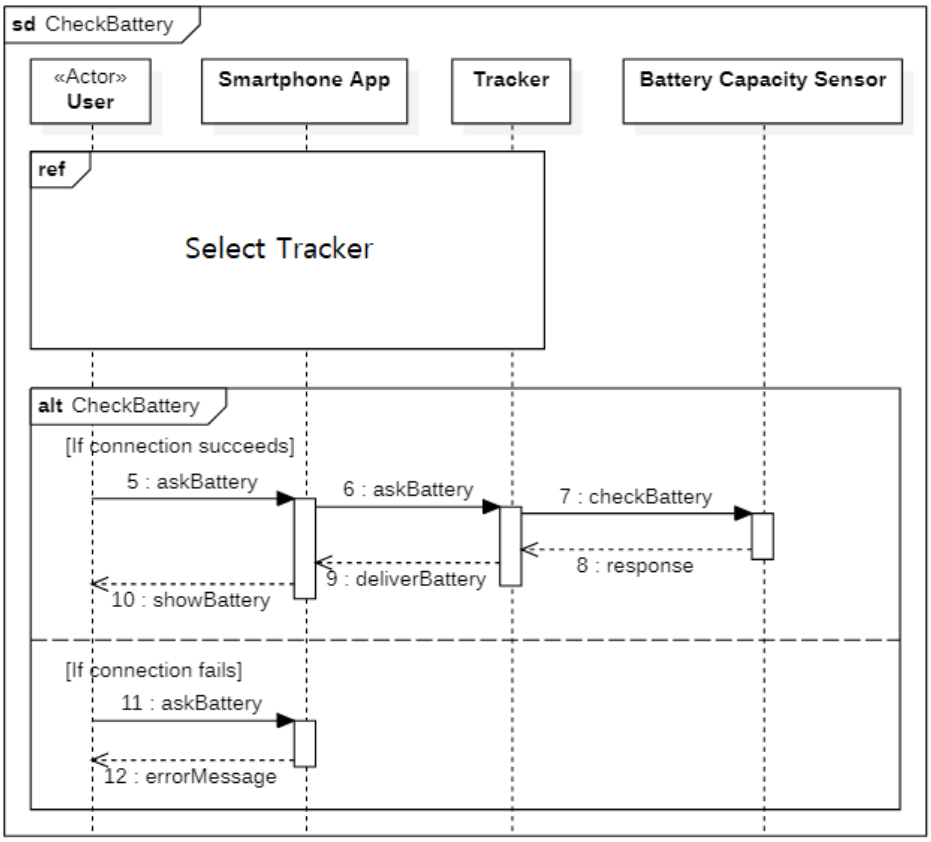
<Figure C.3> Sequence diagram of 'Find tracker using buzzer' use case



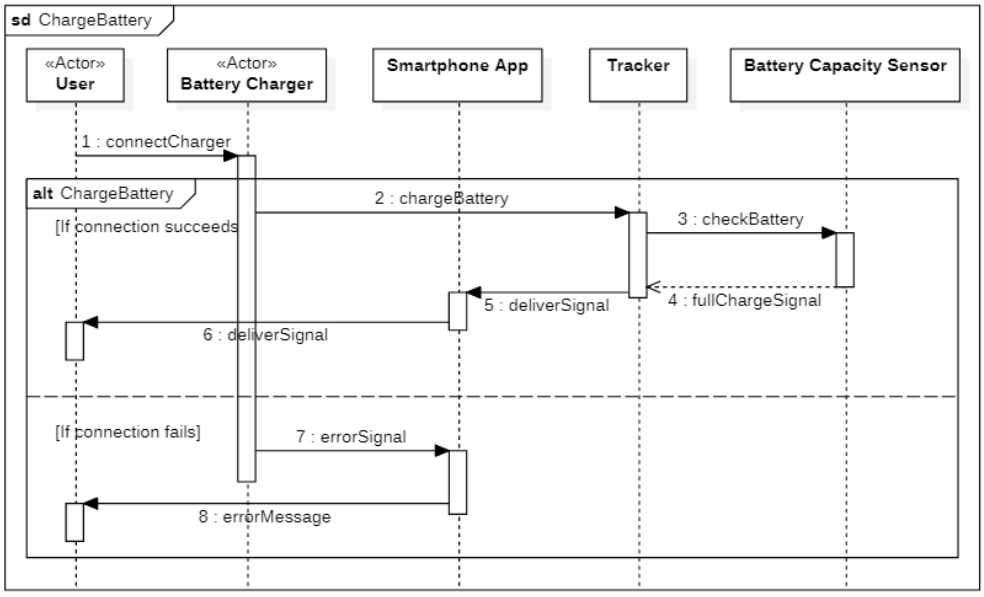
<Figure C.4> Sequence diagram of ‘Turn on/off certain smart trackers’ functions’ use case



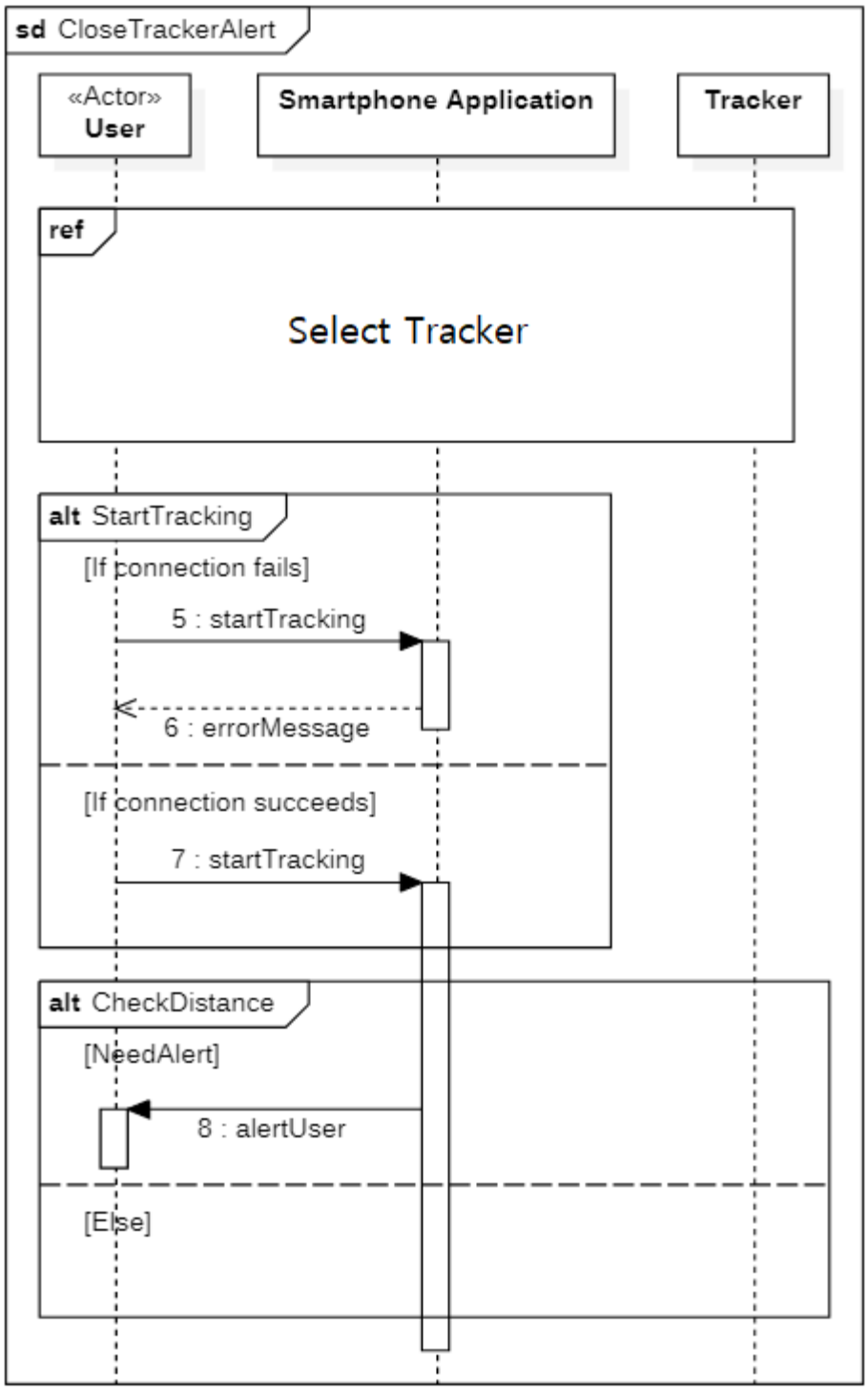
<Figure C.5> Sequence diagram of ‘Alert when the device and smartphone get further away’ use case



<Figure C.6> Sequence diagram of ‘Check the remaining battery ’ use case



<Figure C.7> Sequence diagram of ‘Charging the battery’ use case



<Figure C.8> Sequence diagram of ‘Give signal when the tracker gets close’ use case

IV. Preliminary User Manual

The users of Smart Tracker are those who often lose something important, or those who have something very important that they don't want to lose. In that sense, anyone can be the user of the Smart Tracker. Also, Smart Tracker is very easy to use. What the user has to do is just simply put the tracker on the object. Then, users can track the device whenever they want by only interacting with smartphone applications.

For instance, let's assume a user has placed the smart tracker in his wallet. He is at home, getting ready to go out but he can't find his wallet. Then he can select the corresponding tracker from his smartphone application and ask its location. Then, the smartphone application will show a map that points the location of the tracker or make the tracker ring so the user can hear it. Now he gets his wallet and goes outside, but he drops his wallet while walking in the street. As soon as the tracker gets out of the connection range, the smartphone application warns the user about it by push-alarm, vibration, and sound.

Not only Smart Tracker can warn users when tracker is out of range but also can notify users when trackes comes into the range. Such a case would be when the user has put the tracker in his baggage, and is waiting for it at the airport. Smart Tracker's power is supplied by an embedded battery which is rechargeable. Therefore, users can check the remaining battery of the trackers and charge those who are at low state.

V. Non-functional Requirements (Quality Attribute)

1. User Interface and Human Factors

Expected User Types

- Users who frequently forget their belongings
- Users who use smartphones and usually bring their smartphones
- Users don't have to be familiar with smartphone apps, but should have no restriction for installing smartphone apps.

Available Input/Output Devices

- Smartphone device (application should be installed)
- Tracker device

Requirements

- Tracker device attaching process should be simple so users can easily attach/detach tracker devices to their belongings.
- GUIs should be intuitive so that they are easily understandable for users.
- All critical errors should be notified to users well, so that users can recognize and deal with errors (try again).

2. Documentation

- Documentation for tracker device attaching/detaching process.
- Smartphone application installation guide should be given so that users can easily find and install the designated application from the app store.
- A documentation for registering their designated tracker device in their smartphone application.

- A list of the system's functions (alert, device management, GPS) and their processes.

3. Hardware Considerations

Tracker Device Hardware Requirements

- Size should be small enough so that they will not border users when they are attached to belongings.
- Batteries should last at least several hours when they are fully charged.
- Devices should not be easily broken.

Smartphone Application Hardware Requirements

- Bluetooth communication.
- Enough storage for the designated smartphone application.
- Enough storage for storing the most recent GPS position of tracker devices.

4. Performance Characteristics

- Bluetooth communication should be available at distance

5. Error Handling and Extreme Conditions

- All fatal errors and their suggested handling procedure should be notified to a user in a proper language through a smartphone application message so that the user can easily figure out how to deal with errors or try again.

Expected Error Situations

- Bluetooth connection lost

6. System Interfacing

User - App interface

- Graphic user interface

App - Tracker device interface

- Bluetooth communication

Tracker device - GPS interface

- GPS communication

7. Quality Issues

- Should be compatible with major smartphone OSs (iOS, AOS)
- Periodically update tracker devices' state in a acceptable period

8. System Modifications

- Smartphone applications can be modified in GUI or efficiency updates
- The system can be further modified to deal with an unstable communication environment.

9. Physical Environment

- The system should work in an environment where wireless connection is stable.

10. Security Issues

- Smartphone applications should not track or manage unregistered tracker devices.
- Tracker devices should not receive messages from unregistered smartphone applications.
- The GPS positions of tracker devices should be stored periodically for the case of a tracker device becoming disconnected.

11. Resources and Management Issues

- Mobile application development skills.
- Mobile GPS api application skills.
- Hardware engineer

VI. Acknowledgement

Introduction - 박윤정
Overall Description - 김준범, 박윤정, 이정재
System Features - 김준범, 박윤정, 이정재
Preliminary User Manual - 김준범
Non-functional Requirements - 이정재
Editing - 김준범, 박윤정, 이정재