Locust – Tracker for Safety
Analysis Model

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:

Celeste, Jimuel Jr.
Hernandez, Manolo
Puato, Ricardo III

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-2020

Unique Reference:

The documents are stored in the https://github.com/manologhernandez/SOS/tree/master/02-Requirements%20Engineering/Project%20Deliverables referenced with 1-Locust-Analysis-Model.pdf

Purpose:

To provide a base model of what the software functions, features and constraints are.

Audience:

The target audience of the system are young adults, specifically high school and college students who usually go out at night and value their safety and security.

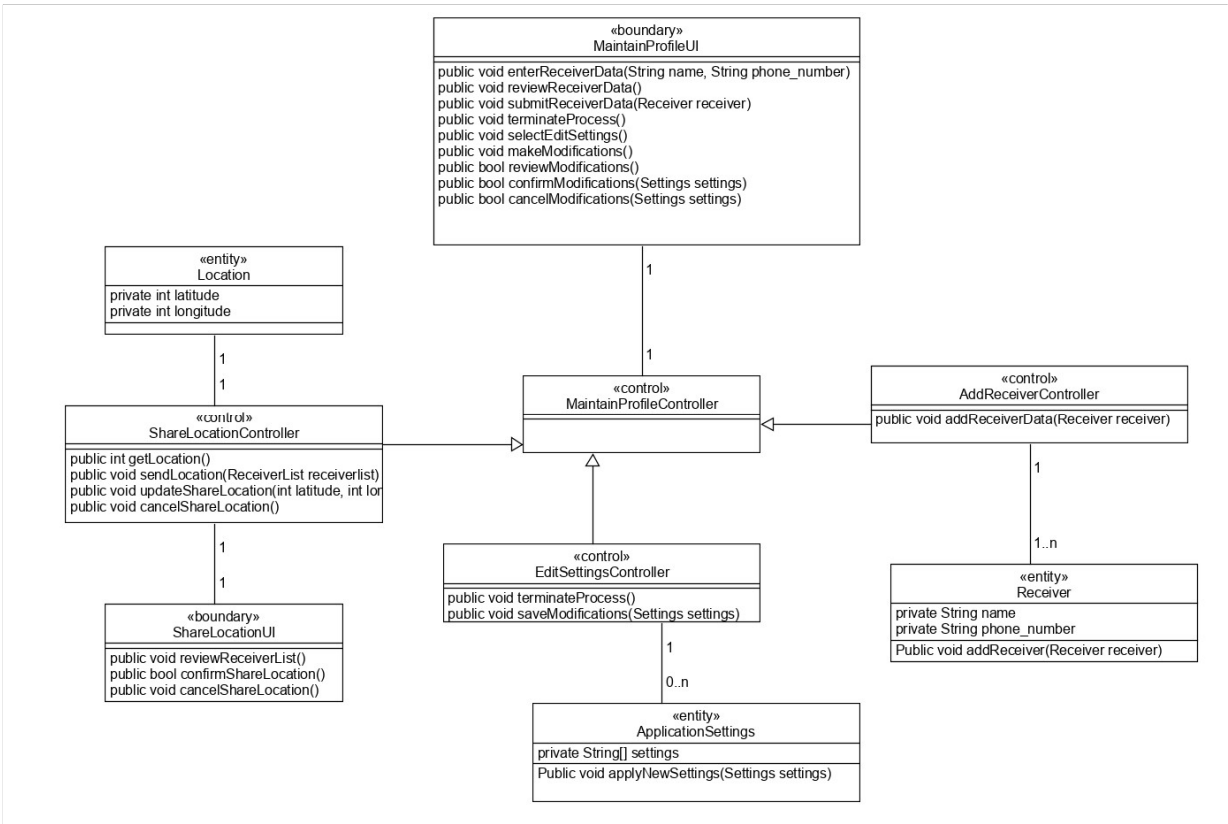
Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Modification*
10/01/2019	Jimuel Celeste Jr.	1.0	Initial Document; Added identified analysis classes; Added Behavioral Model for use-case 1.1 Add Receiver; Added identified responsibilities of based on Behavioral Model for use case 1.1
10/03/2019	Manolo Hernandez	2.0	Added Behavioral Model for use-case 1.4 Edit Settings. Added class descriptions and responsibilities based on behavioral model for use case 1.4
10/04/2019	Ricardo Puato	3.0	Added Behavioral Model for use-case 2.0 Share Location. Added class descriptions and responsibilities based on behavioral model for use-case 2.0
10/04/2019	Manolo Hernandez	4.0	Added Analysis Model for the Locust System

***System Name*:** Locust – Tracker for Safety

***Description*:** Locust – Tracker for Safety is a location sharing Android application. It would let the user to automate the process of location sharing.

Analysis Model



Boundary Classes:

Class Name	**Description**
MaintainProfileUI	This is the interface of the user to the application whenever the user wants to maintain her/his profile. *Responsibilities:* public void enterReceiverData(String name, String phone_number) public void reviewReceiverData() public void submitReceiverData(Receiver receiver) public void terminateProcess() public void selectEditSettings() public void makeModifications() public bool reviewModifications() public bool confirmModifications(Settings settings) public bool cancelModifications(Settings settings)
ShareLocationUI	This is a interface shown to the user whenever he/she wants to share location. *Responsibilities:* public void reviewReceiverList() public bool confirmShareLocation() public void cancelShareLocation()

Control Classes:

Class Name	**Description**
MaintainProfileController	This is the control that maintain user's profile. It is considered as an abstract class.
AddReceiverController	This is the control that adds a receiver to the application. It extends MaintainProfileController. *Responsibilities:* public void addReceiverData(Receiver receiver)
EditSettingsController	This is the control that edits the settings of the application. It extends the MaintainProfileController. *Responsibilities:* public void terminateProcess() public void saveModifications(Settings settings)
ShareLocationController	This is the control that lets the user review the list of receivers and then shares the user's location. *Responsibilities:* public int getLocation() public void sendLocation(ReceiverList receiverlist) public void updateShareLocation(int latitude, int longitude) public void cancelShareLocation()

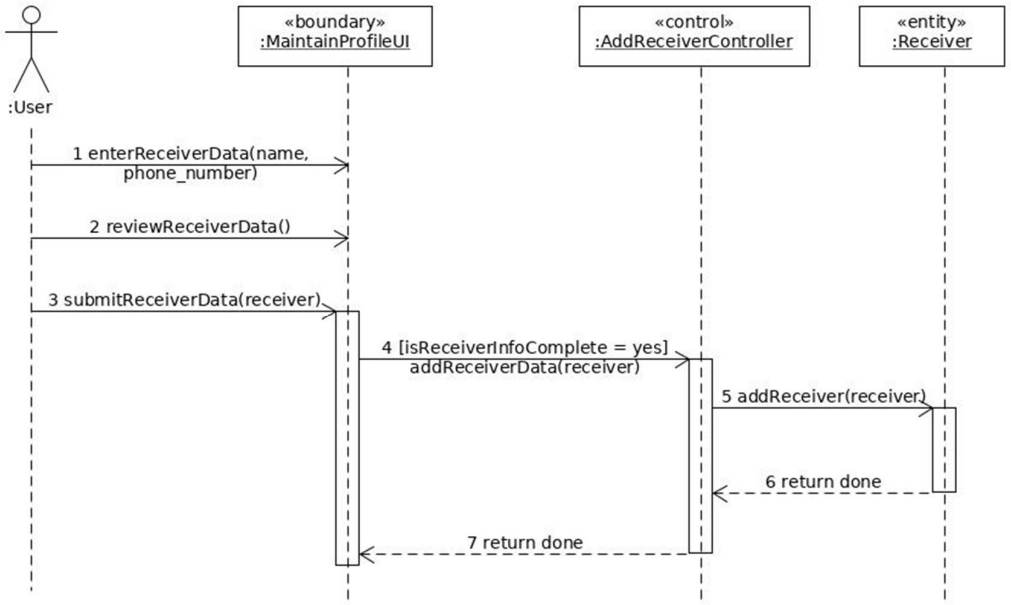
Entity Classes:

Class Name	**Description**
Receiver	This entity class contains the data about the receiver. *Attributes:* private String name private String phone_number *Responsibilities* Public void addReceiver(Receiver receiver)
ApplicationSettings	This entity class contains the user's application settings. *Attributes:* private String[] settings *Responsibilities* Public void applyNewSettings(Settings settings)
Location	This entity class contains the data about the user's location. *Attributes:* private int latitude private int longitude

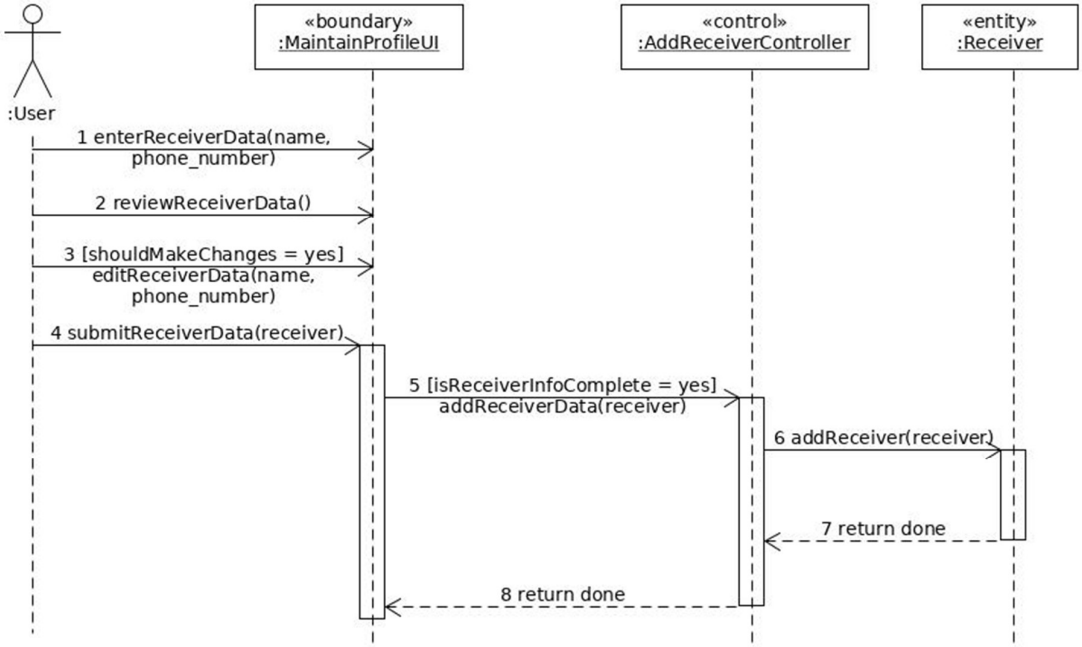
***Use-Case Name*:** **1.1 Add Receiver**

Description: *The user would add receivers. Personal information—name and contact details—of the receiver would be asked. These information would be used for location sharing.*

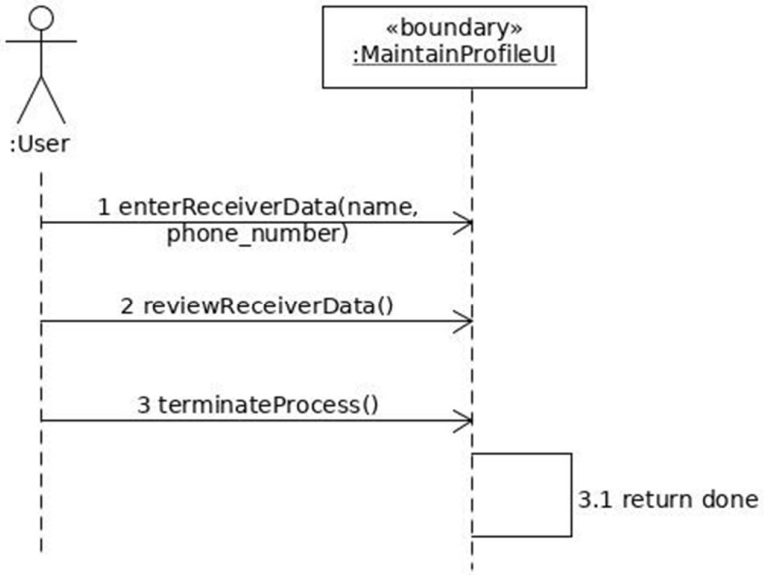
Scenario 1: Basic Flow



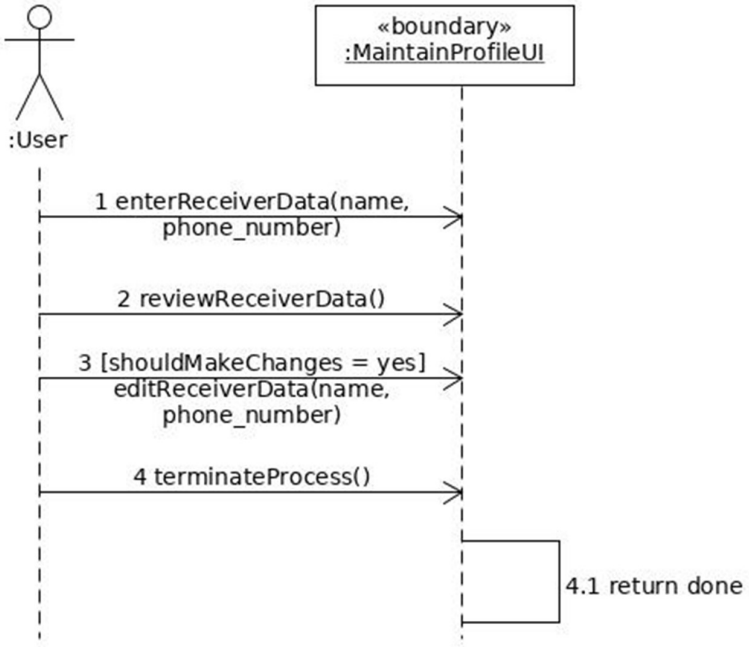
Scenario 2: User edits inputted information before finalizing.



Scenario 3: User cancels adding a new receiver after reviewing input.



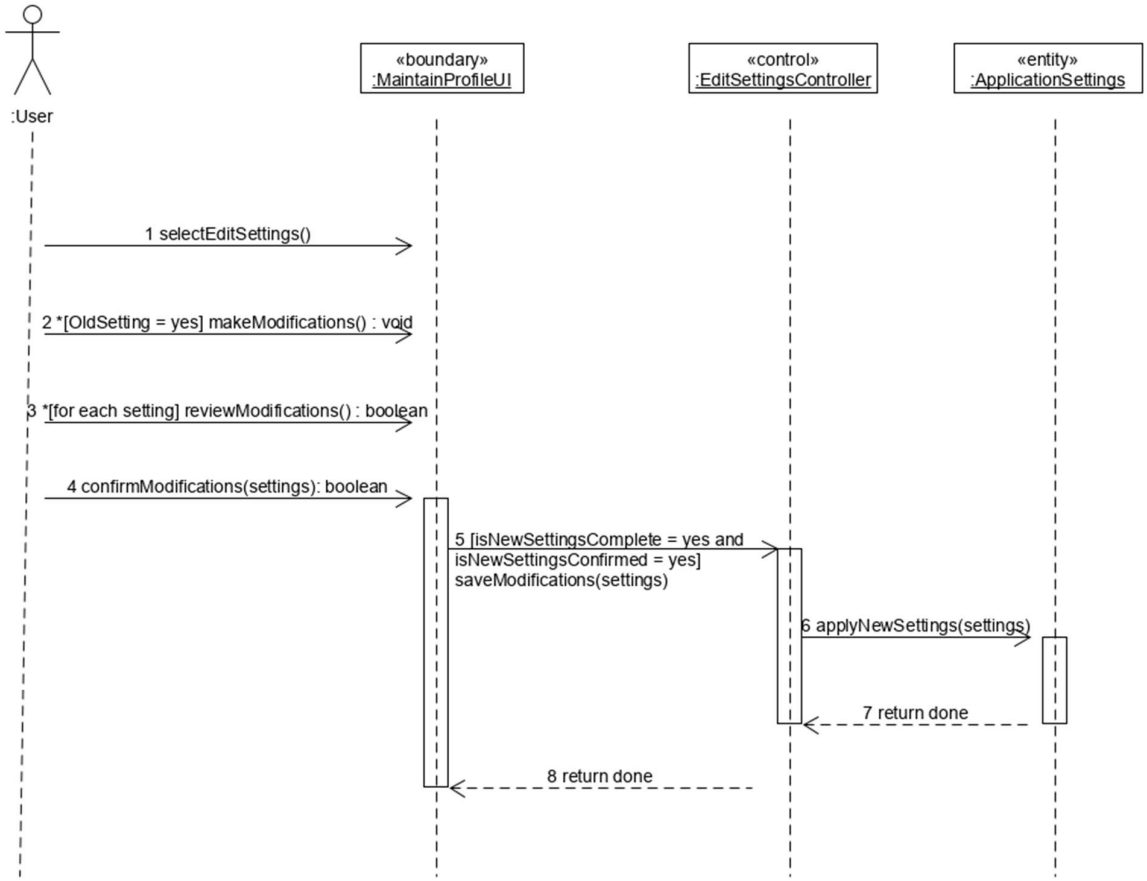
Scenario 4: User cancels adding a new receiver after editing data.



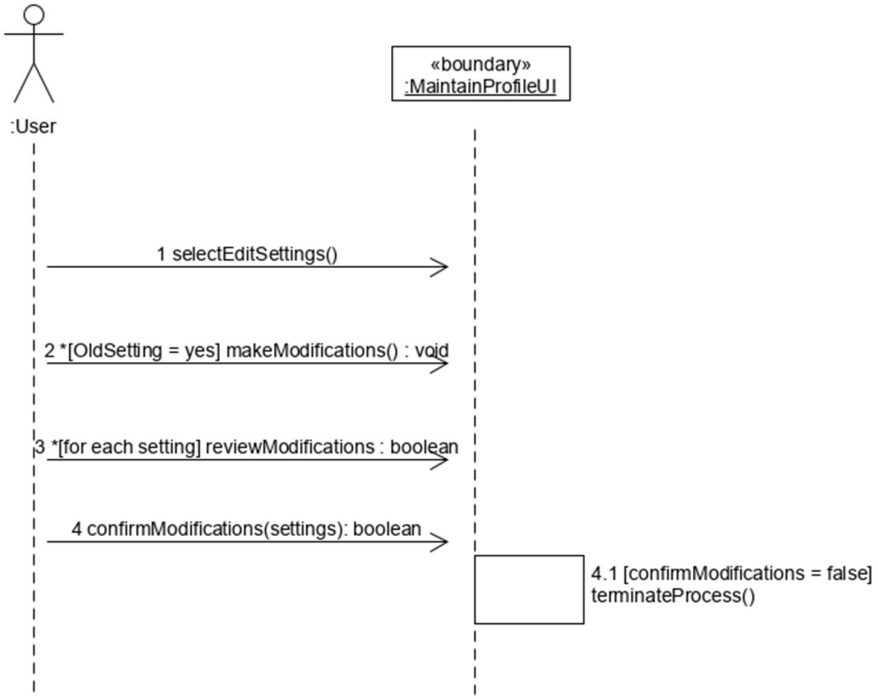
***Use-Case Name*:** **1.4 Edit Settings**

Description: The user will edit the settings of the application. Editing the settings means customizing how the user would want the app to function, and whether to allow, modify or restrict some functionalities. These modifications include the frequency of location sharing, which platforms the user would want their location to be shared, adding a passcode/lock, and more.

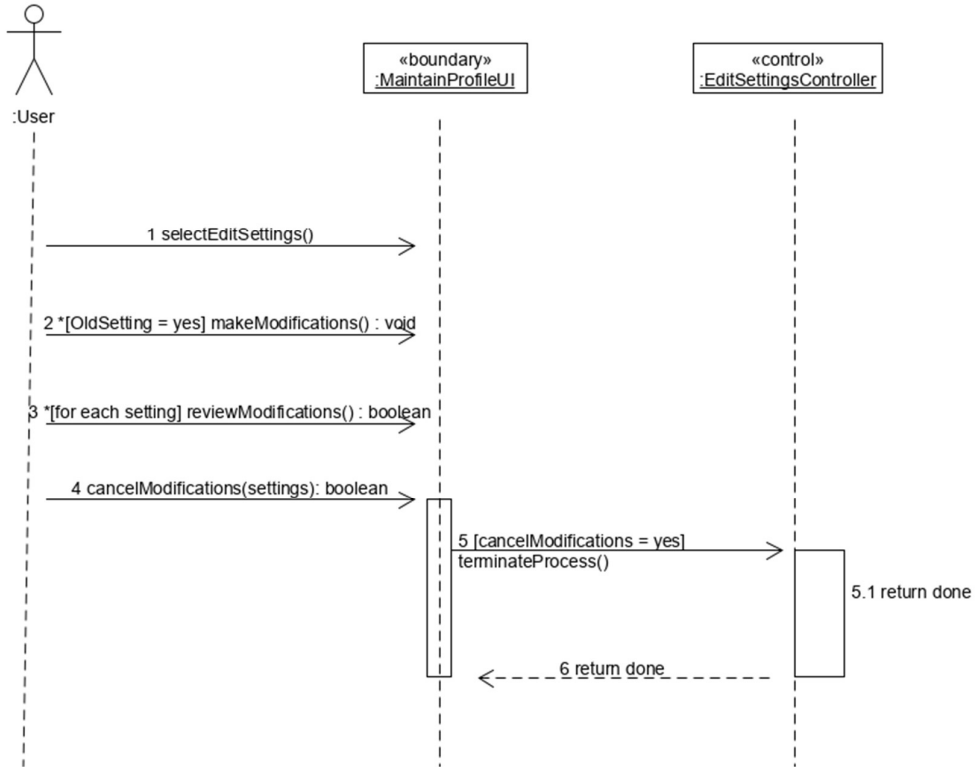
Scenario 1: Basic Flow: User modifies their setting



Scenario 2: User makes modifications to their settings but does not save.



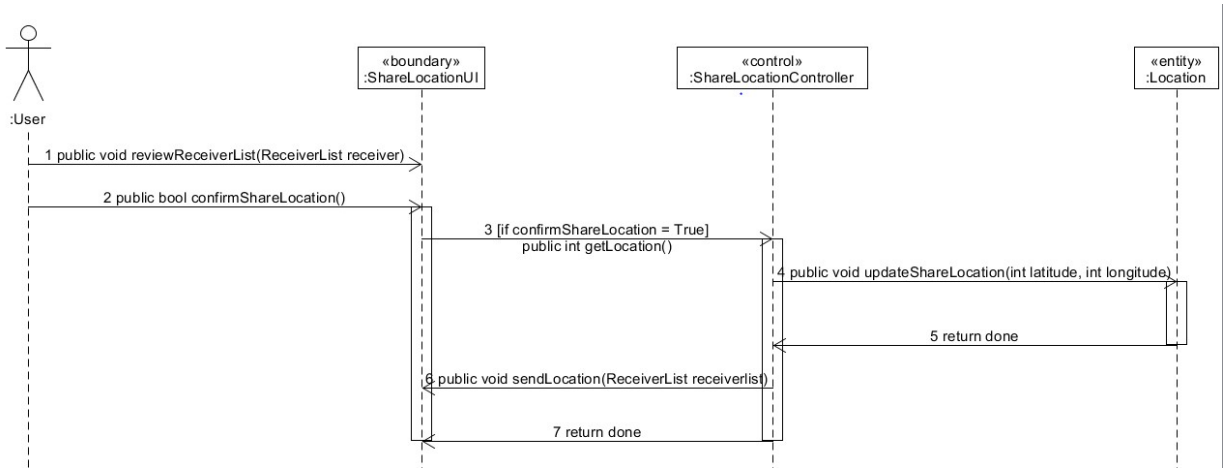
Scenario 3: User makes modifications to their setting but decides to cancel their changes instead.



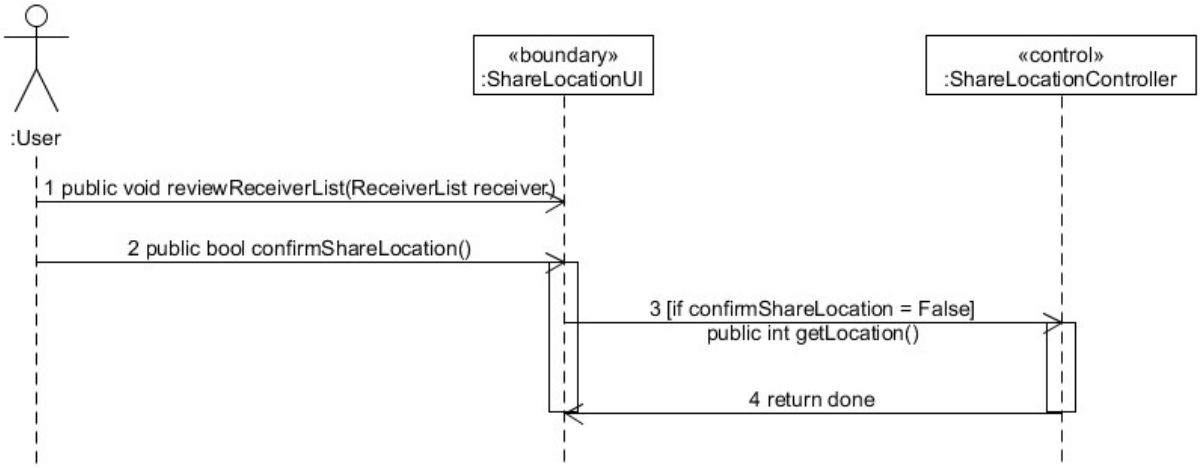
***Use-Case Name*:** **2.0 Share Location**

Description: The User selects to share his/her location. The user will review the trusted receivers. A confirmation will be asked then location is sent.

Scenario#1: Basic Flow



Scenario#2: User doesn't confirm to share location.



Scenario#3: User cancels to share location