touch.inc

Wi-Fi Communication Platform
Software Architecture Document

Version 1.0

Contents

Software Architecture Document

1. Introduction

This section will describe about this Software Architecture Document and how to use it.

1.1 Purpose

The purpose of this document is to provide a detailed architecture design of the Wi-Fi communication system. The expected audience is Developer and Clients who wish to take it.

Developer can use this as a reference for the development and will provide the guidance to the expected outcome of this project. Clients can use this document to understand the system for their needs.

1.2 Scope

The scope of this software architecture document is to depict the architecture of WI-FI communication system created by Dushan soft.

1.3 Definitions, Acronyms, and Abbreviations

These short words have the given meanings in this document.

Term	Description
UI	User interface
OS	Operating system (here it is Windows XP or above)
User	Person who is using this system
PC	Personal Computer, a computer which runs a windows operating system

1.4 References

Tools used to create diagrams.

[1] Microsoft “Visio” Internet: https://products.office.com/en/visio/flowchart-software. 2017 [Mar. 17, 2017]

[2] Cinergix Pty. Ltd. “Creately” Internet: https://creately.com. 2017 [Mar. 17, 2017]

[3] Microsoft “Word” Internet: https://products.office.com/en/word. 2017 [Mar. 17, 2017]

1.5 Overview

Rest of this document is organized as follows.

Section 2: Architectural Representation

Describes what software architecture is for the current system, and how it is represented

Section 3: Architectural Goals and Constraints

Describes the software requirements, objectives that have some significant impact on the architecture and constraints of the system.

Section 4: Use-Case View

Describes the functional requirements with a significant impact on the architecture

Section 5: Logical View

Describes the architecturally significant parts of the design model

Section 6: Process View

Describes the system's decomposition into lightweight processes

Section 7: Deployment View

Describes how the system will be deployed. Will contain the Deployment Model

Section 8: Implementation View

Describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model

Section 9: Data View

Describes the persistent data storage perspective of the system

Section 10: Size and Performance

Describes the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints

Section 11: Quality

Describe how the software architecture contributes to all capabilities of the system

Section 12: References

Referenced materials used to create this document

2. Architectural Representation

Architecture representation has organized in following manner.

View	Audience	Area	Related artifacts
Use case view	All the stakeholders	Scenarios and use cases of functionality in the system	Use-Case Model, Use-Case documents
Logical view	Designer	Design’s object model and important use case realizations of functional requirements	Design model
Process view	Developer	describes concurrency and synchronization aspects	-
Deployment view	Deployment manager	Software and hardware mapping and the system’s distributed aspects	Deployment model.
Implementation view	Developer	Layers and the subsystems of the application	Implementation model, components

Data view	Designer	Significant persistent elements in the data model	Data model.

3. Architectural Goals and Constraints

Main focused requirements and areas in the system that have significant impact on architectural design are follows

- Availability of the system is 100%
- Every user is identified by his MAC address.
- User profile data is changed with the instance of the usage because IP address is changed time to time.

4. Use-Case View

Main use cases in this software have been described in this section.

4.1 Use-Case Realizations

4.1.1 Use case diagram

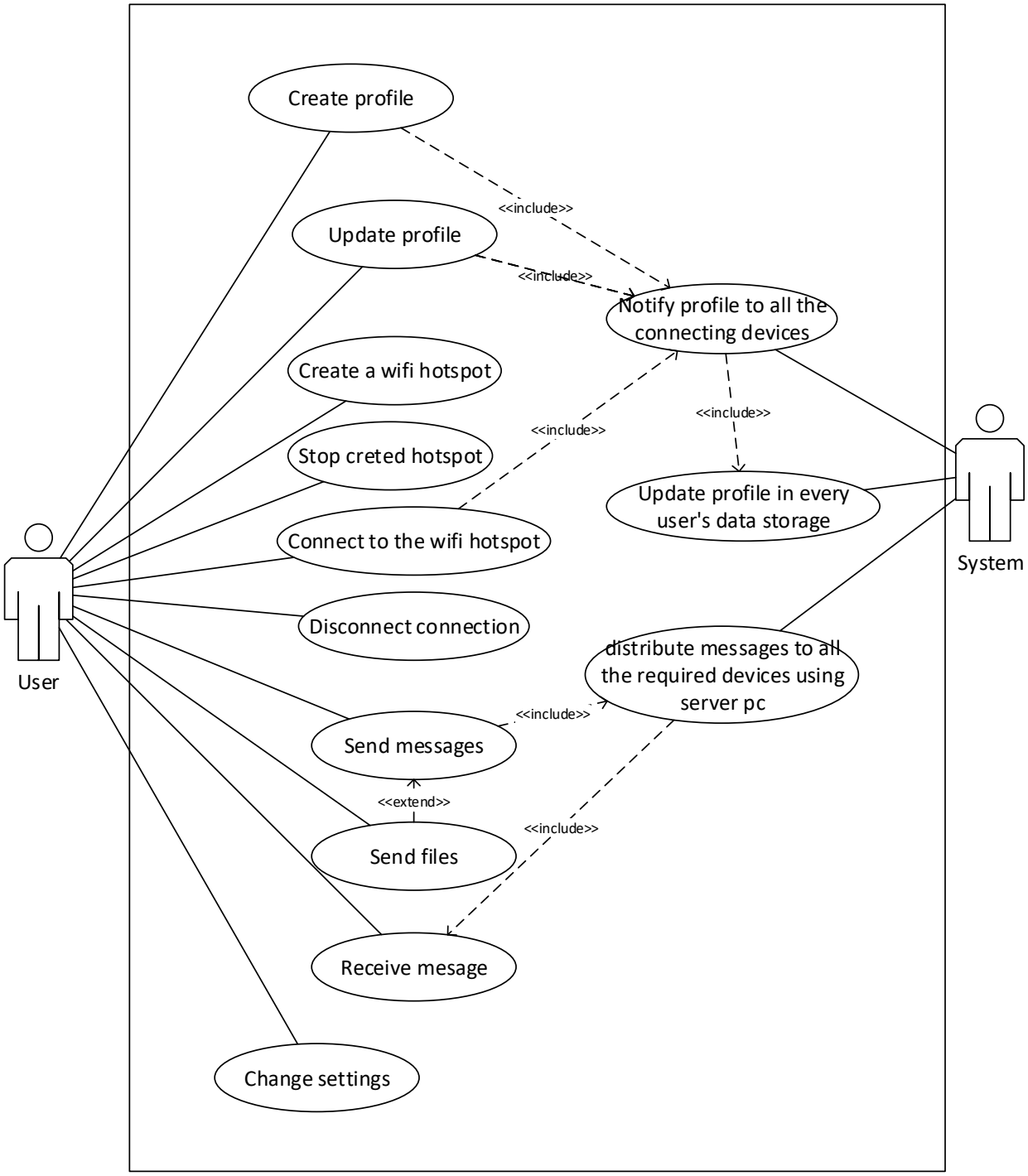

Figure 1-Use case diagram

4.1.2 Use case scenarios

4.1.2.1 Create profile

Use case name	Create profile
Actor	User
Description	Newly added user add his/her profile picture and his/her name to is profile and save it.
preconditions	N/A
Main flow	User add his details(profile picture and name) and save the changes Then the system updates the user profile data in the data storage.(it is not using a sql database, that will be a serialized data object) Then if the device has connected to a network system will notify all the connected devices about profile data
Successful end/post condition	Profile added confirmation
Fail end/post condition	N/A
Extensions	N/A

4.1.2.2 Update profile

Use case name	Update profile
Actor	User
Description	Profile added user edit his/her profile picture and/or his/her name to is profile and save it.
preconditions	N/A
Main flow	User edit his details(profile picture and name) and save the changes Then the system updates the user profile data in the data storage.(it is not using a sql database, that will be a serialized data object) Then if the device has connected to a network system will notify all the connected devices about profile data
Successful end/post condition	Profile edit confirmation
Fail end/post condition	N/A
Extensions	N/A

4.1.2.3 Create Wi-Fi hotspot

Use case name	Create Wi-Fi hotspot
Actor	User
Description	User press the button for creating Wi-Fi hotspot
preconditions	N/A
Main flow	When user tries to create a hotspot system check for connected networks and give a warning about it. If the user proceed to create the hotspot system will create it.
Successful end/post condition	Hotspot confirmation with hotspot name and password.
Fail end/post condition	If password length is not greater than 6 characters.
Extensions	N/A

4.1.2.4 Stop created hotspot

Use case name	Stop created hotspot
Actor	User
Description	When a user has created a hotspot and the conversation is over the user press stop hotspot button.
preconditions	Should have created a hotspot by himself/herself.
Main flow	User press stop hotspot button. Then system warn about the conversation termination. If user proceed to go forward system will stop the hotspot
Successful end/post condition	Hotspot stop confirmation
Fail end/post condition	N/A
Extensions	N/A

4.1.2.5 Send messages

Use case name	Send messages
Actor	User
Description	When a user has connected to a network and started a chat he can type a text and press send button.
preconditions	Should have connected to a network and select a receiver from contacts list.
Main flow	User select a chat group or a another user. Type text or select a file for send. User press send button. Then system distribute the message to required users. Then system save the messages for in data files.
Successful end/post condition	Sent confirmation
Fail end/post condition	If the connection goes down while a message being sent sending will be fail.
Extensions	Send files

4.1.2.6 Receive messages

Use case name	Receive messages
Actor	System
Description	When a someone send a message, the receiver pc will get the message and notify the user.
preconditions	Should have connected to a network.
Main flow	Someone send a message. Receiver pc will get the message and notify the user. System will save the massage for further usage in data files.
Successful end/post condition	N/A
Fail end/post condition	N/A
Extensions	N/A

4.1.2.7 Change settings

Use case name	Change settings
Actor	User
Description	When a user needs to change notification settings or advanced settings he/she go to settings and change the required settings and press save.
preconditions	N/A
Main flow	User change notification settings or advanced settings User save the changed settings. System save the configuration settings in the config file.
Successful end/post condition	N/A
Fail end/post condition	N/A
Extensions	N/A

5. Logical View

This section describes the architecturally significant parts of the design model

5.1 Overview

The system has been divided to 3 layers as View, Control and Modal. Those layers have its own classes. Those classes are shown in the class diagram and the layered architecture has described in implementation view section.

5.2 Architecturally Significant Design Packages

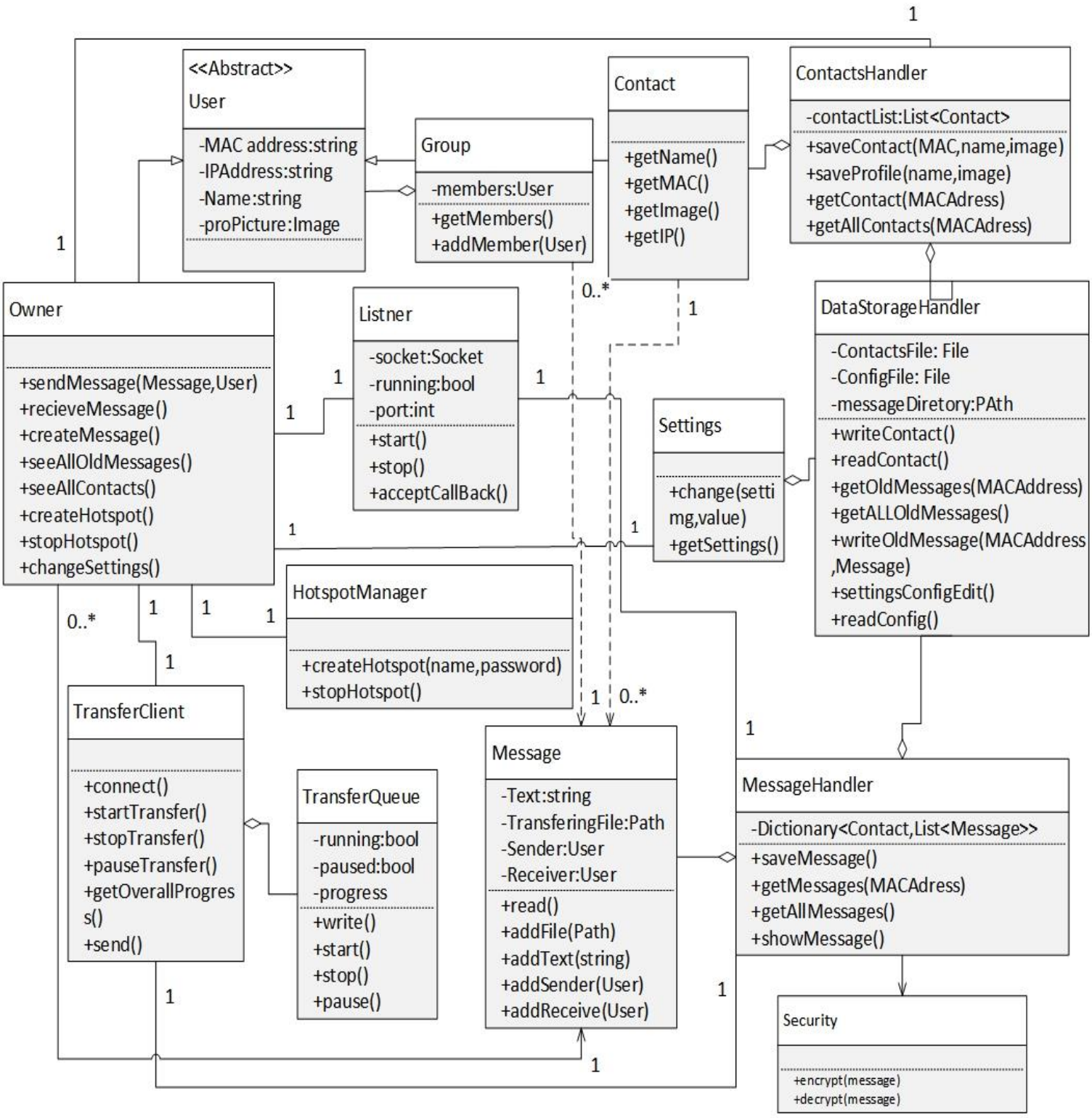

Figure 2-UML Class diagram

6. Process View

In this section it shows the processes in this system using activity diagrams and sequence diagrams.

6.1 Activity diagrams

6.1.1 Create profile

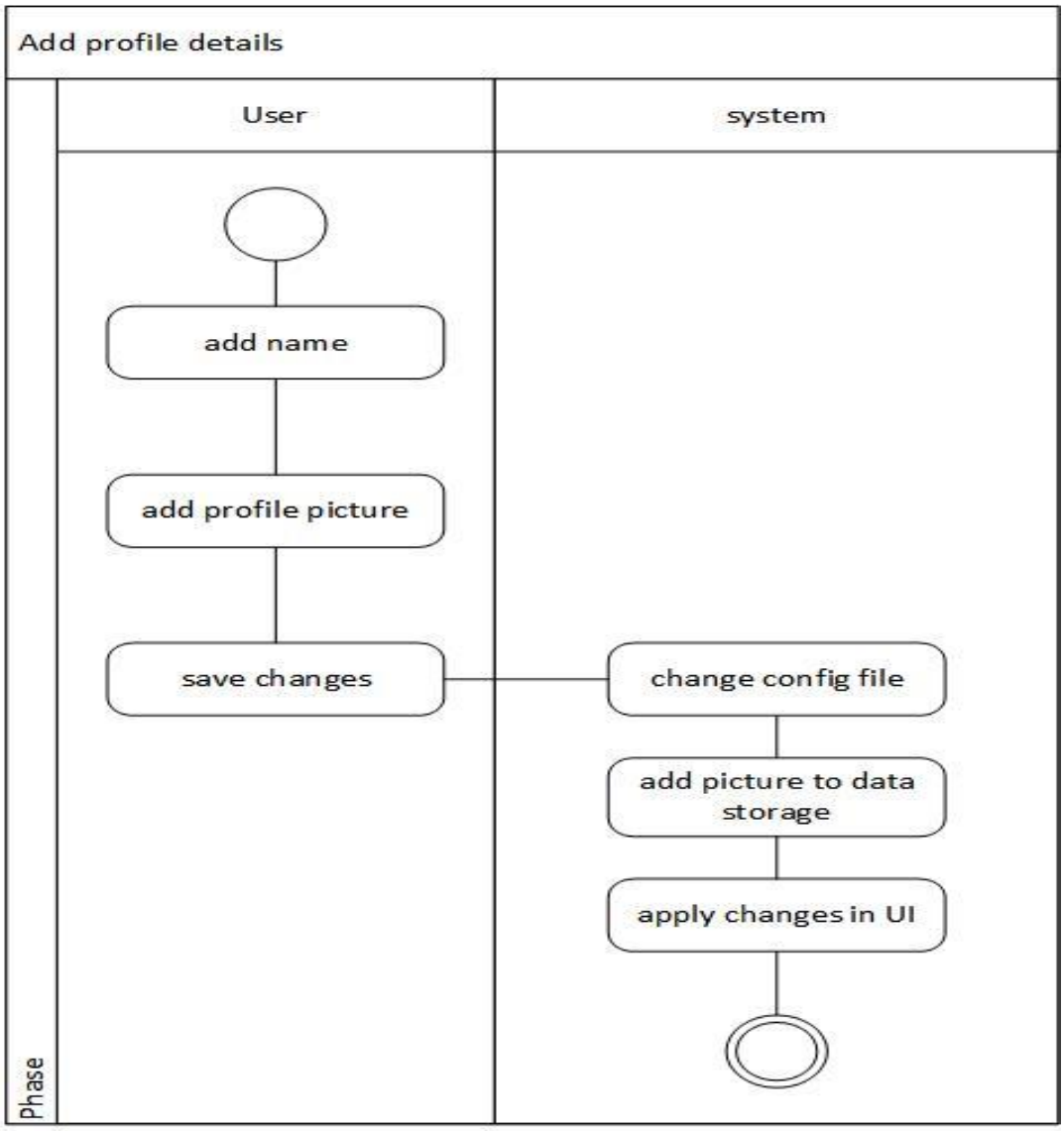

Figure 3-Add profile details activity daigram

6.1.2 Edit profile

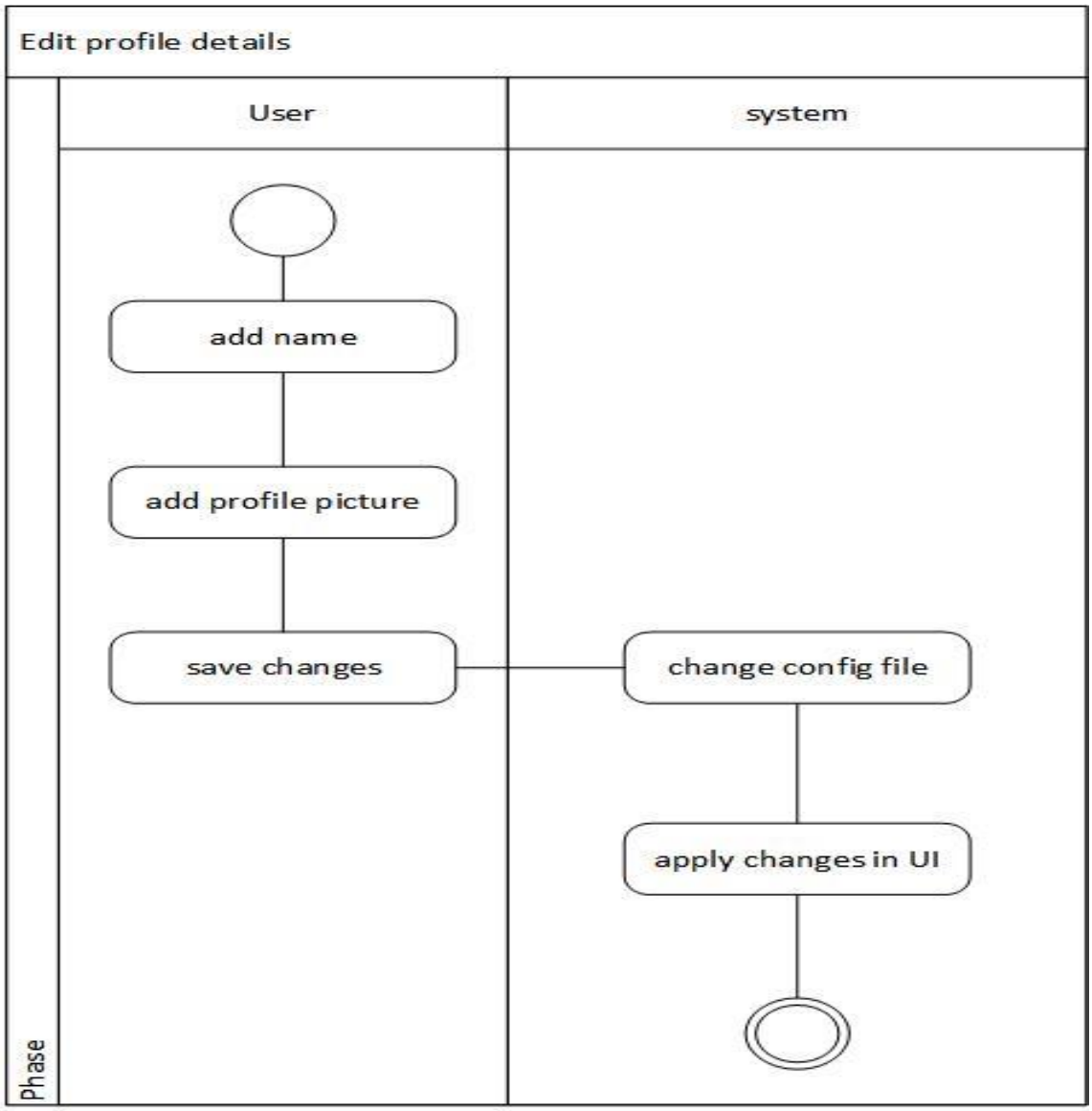

Figure 4-Edit profile details activity diagram

6.1.3 Create hotspot

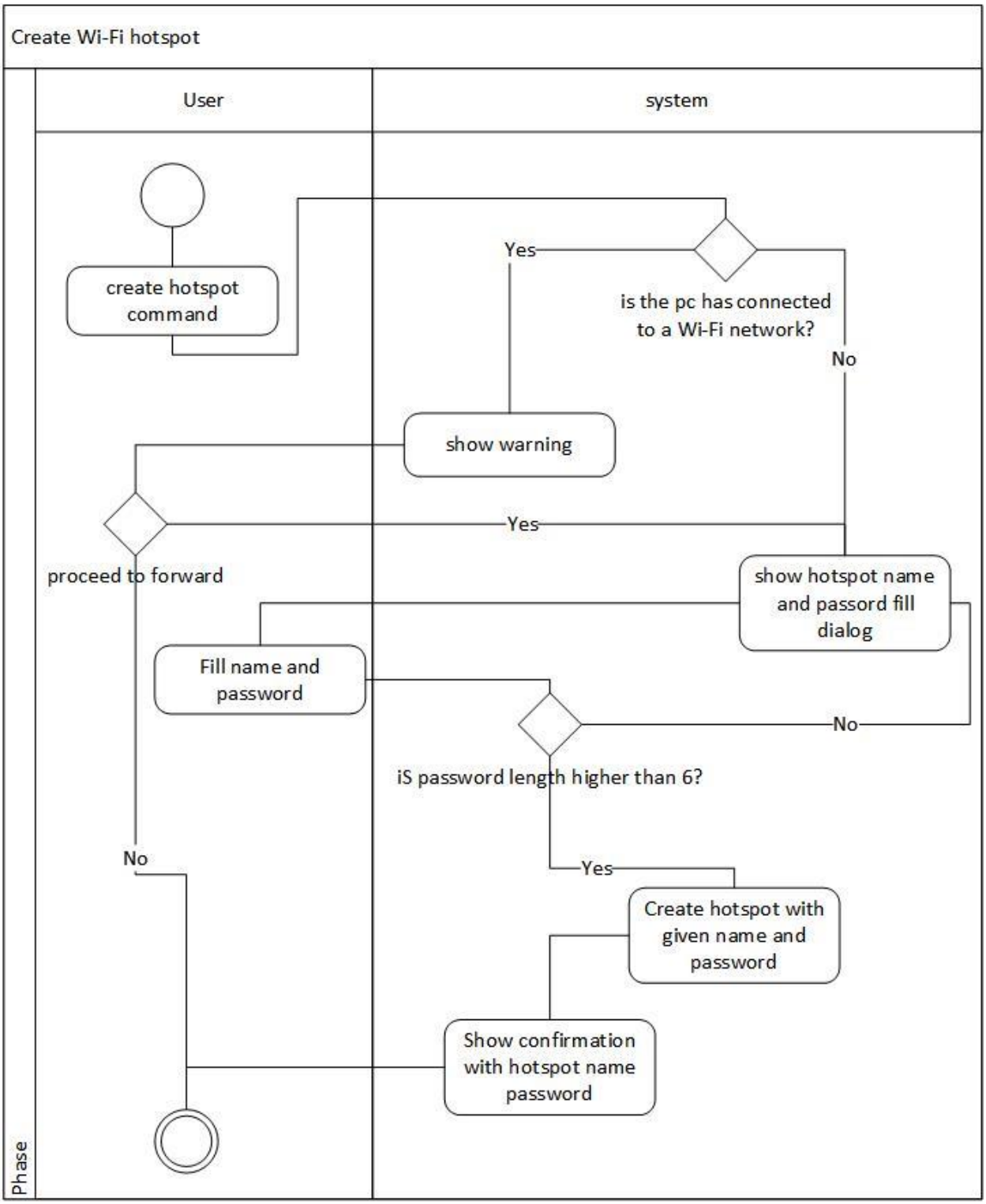

Figure 5-Create hotspot activity diagram

6.1.4 Stop hotspot

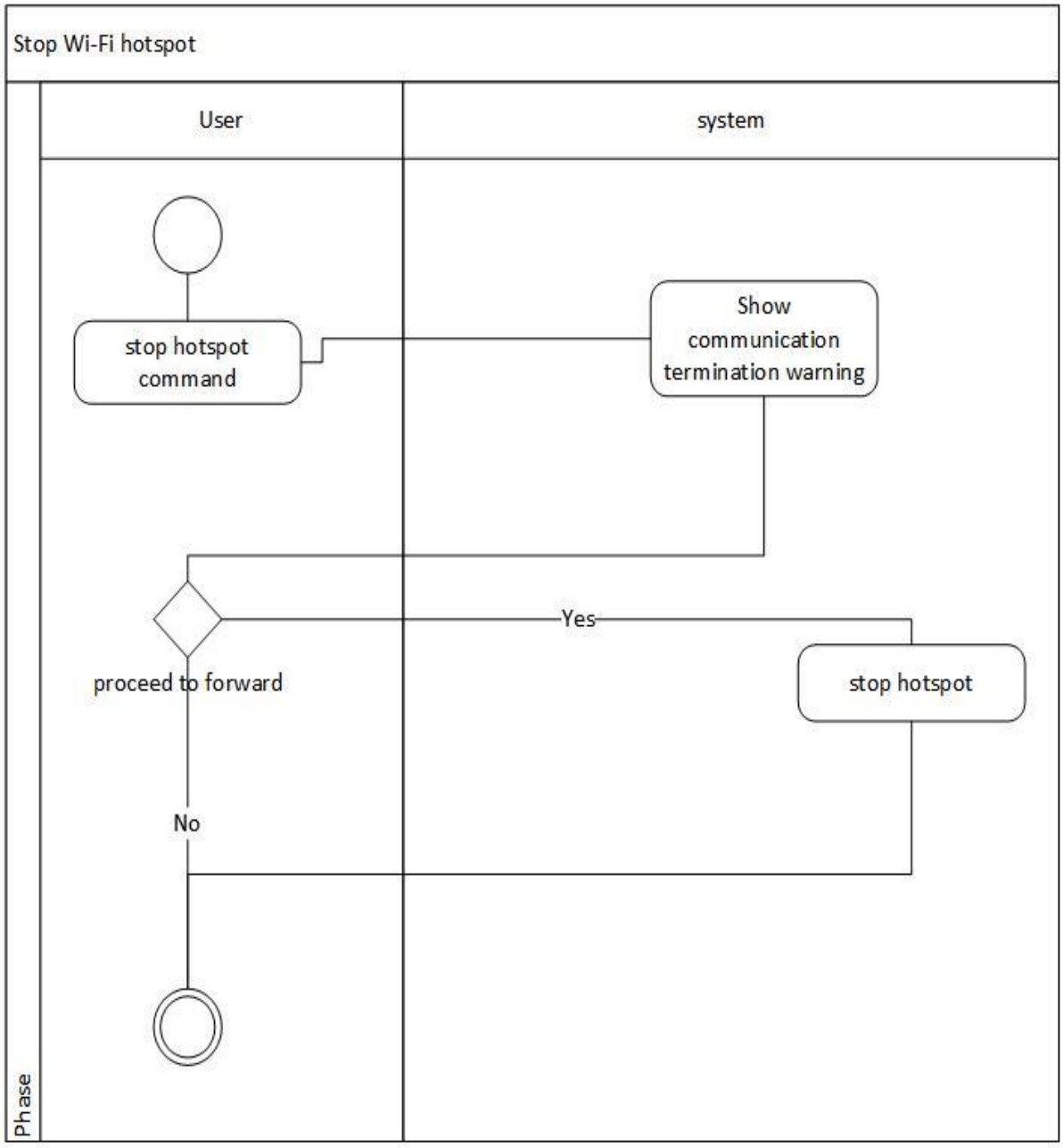

Figure 6-Stop hotspot activity diagram

6.1.5 Change settings

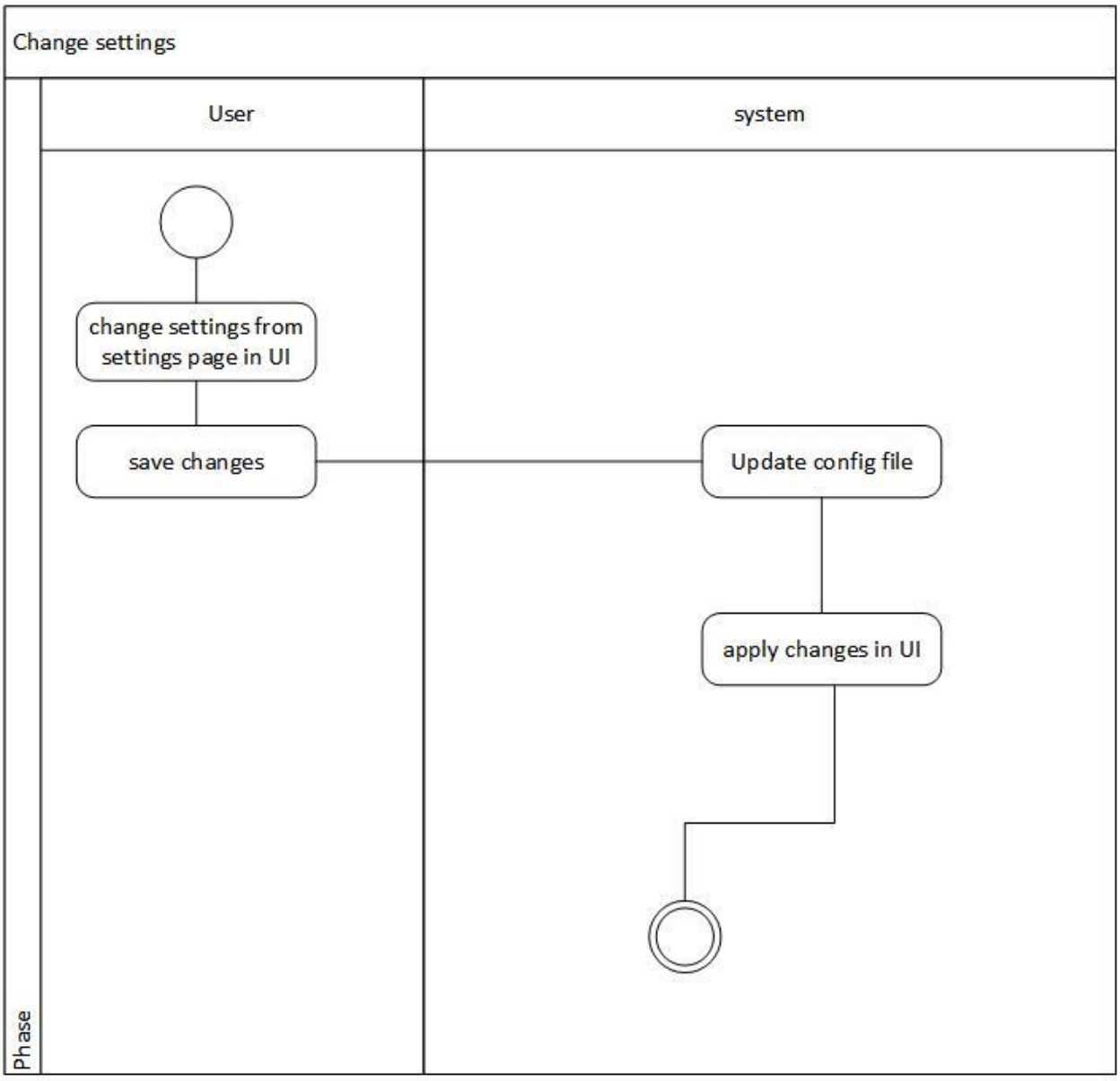

Figure 7-Change settings activity diagram

6.1.6 Send message

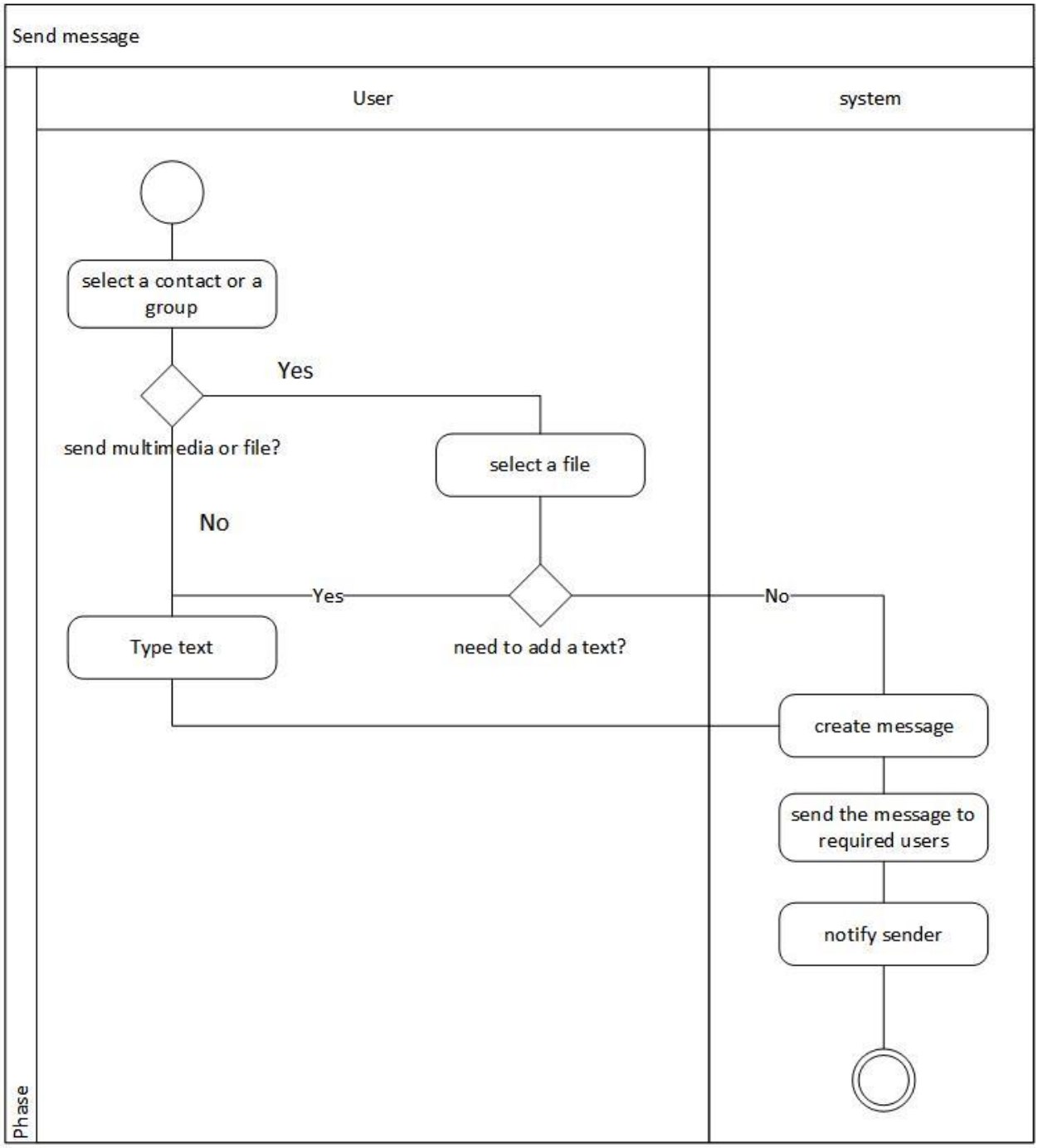

Figure 8-Send message activity diagram

6.1.7 Receive message

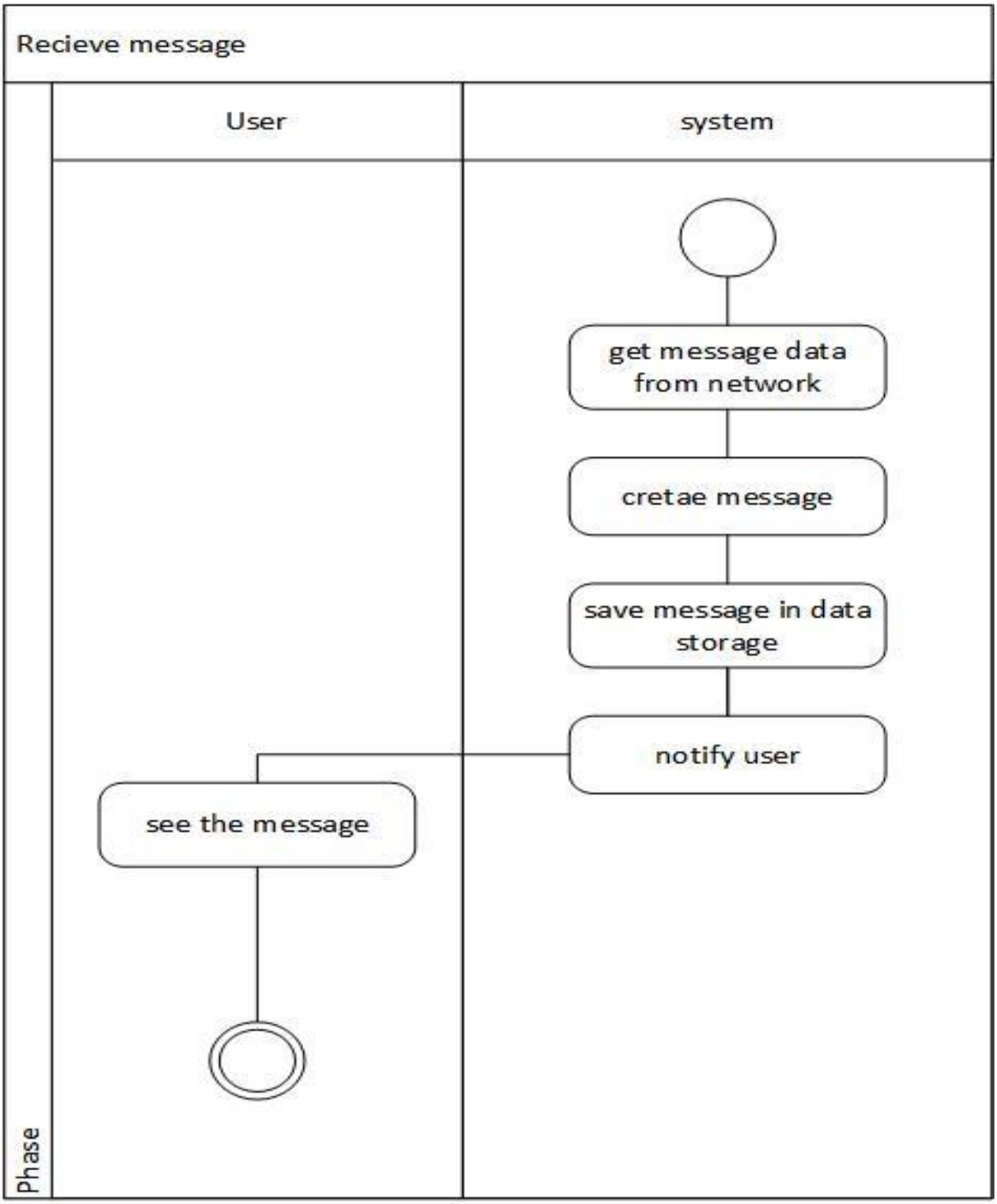

Figure 9-Receive message activity diagram

6.2 Sequence diagrams

6.2.1 Create profile

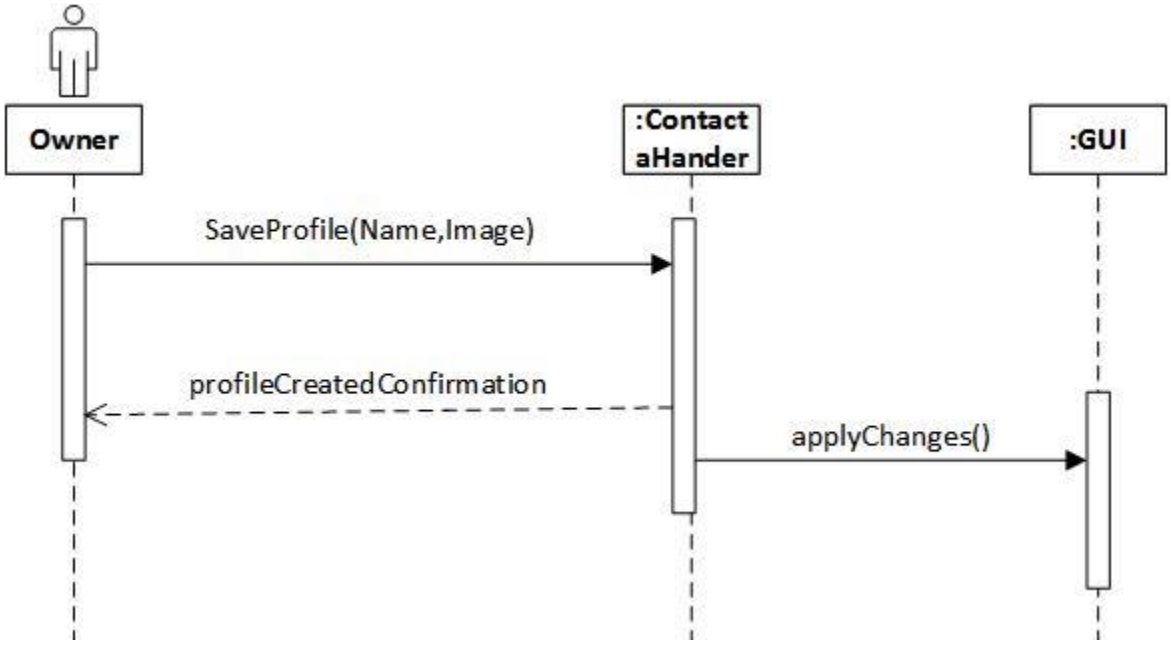

Figure 10-Create profile sequence diagram

6.2.2 Edit profile

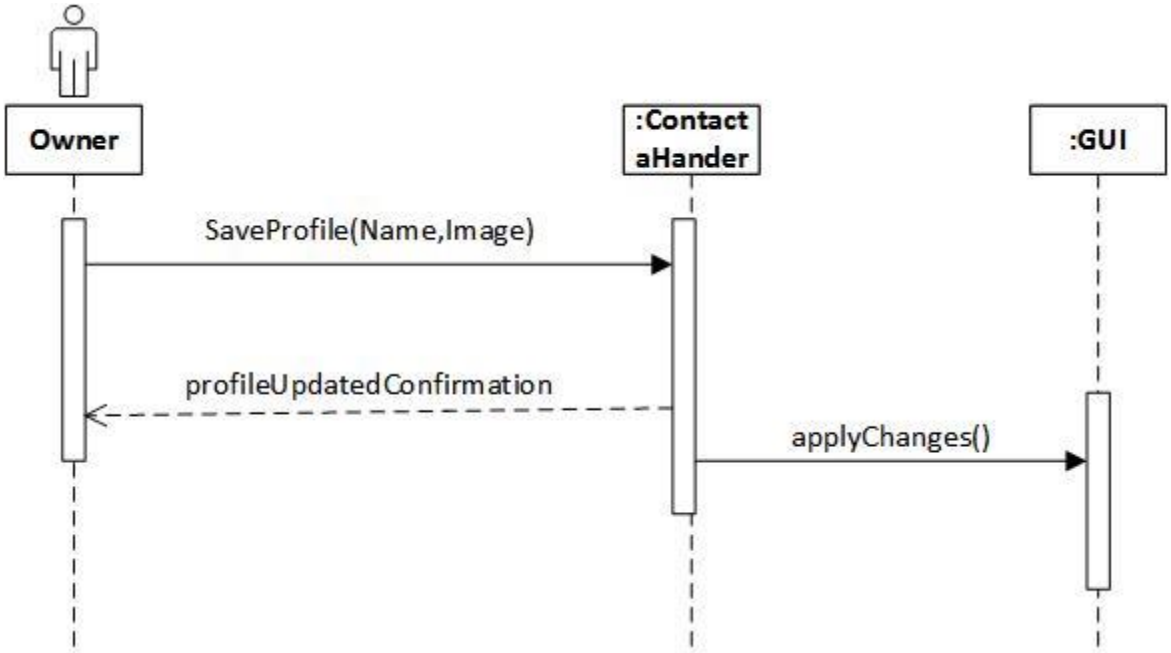

Figure 11-Edit profile sequence diagram

6.2.3 Create hotspot

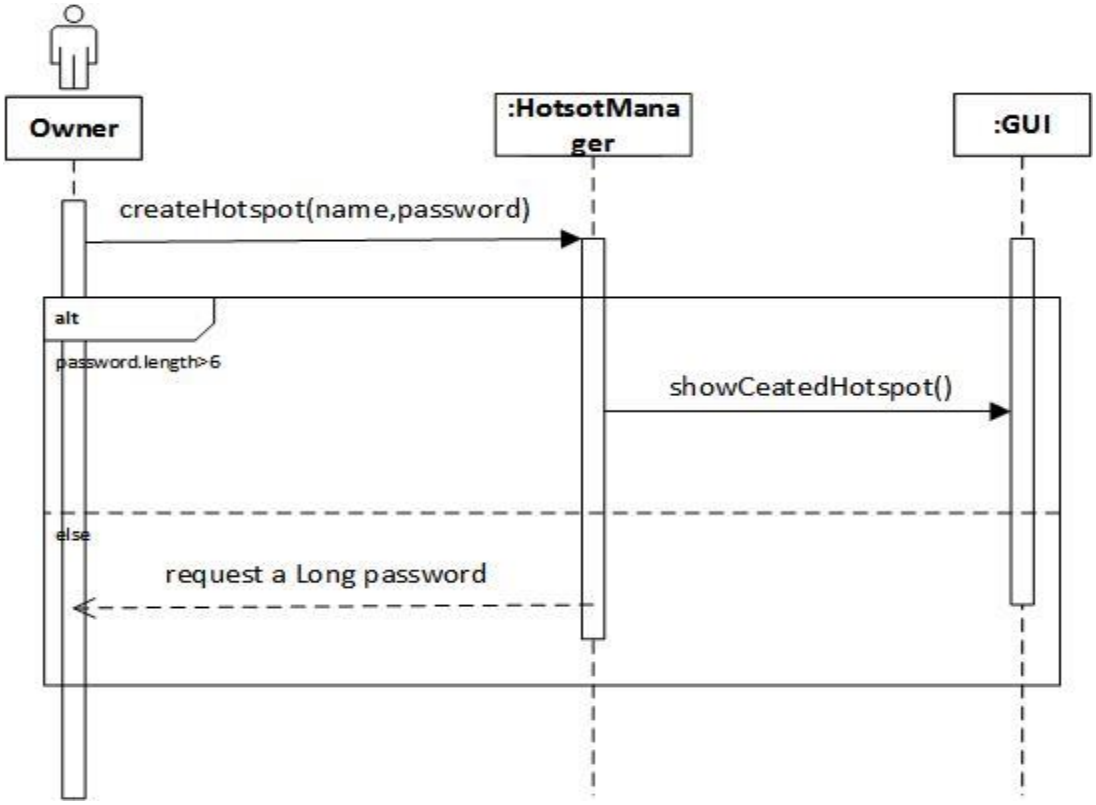

Figure 12-Create hotspot sequence diagram

6.2.4 Stop hotspot

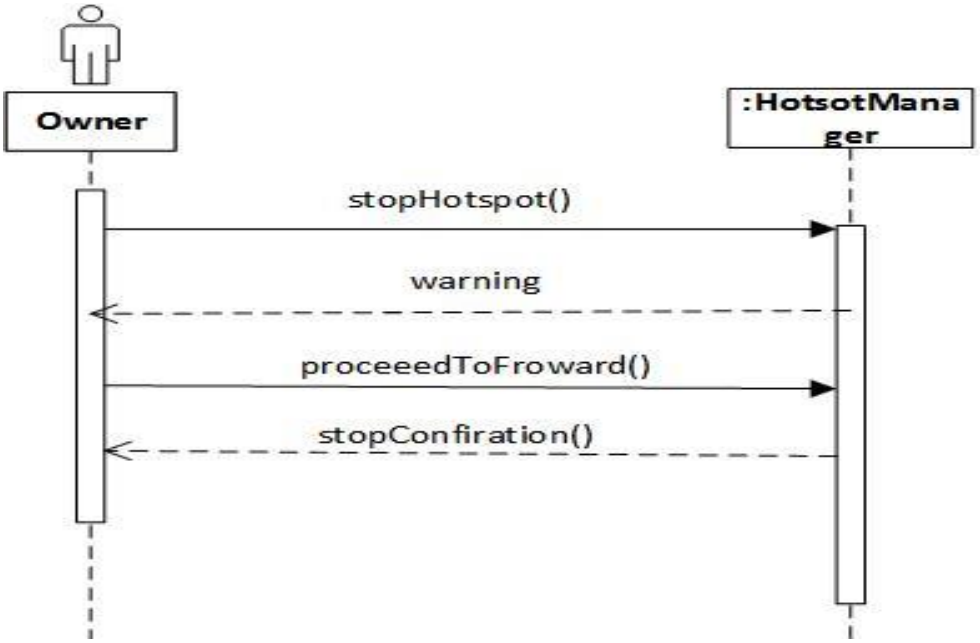

Figure 13-Stop hotspot sequence diagram

6.2.5 Change settings

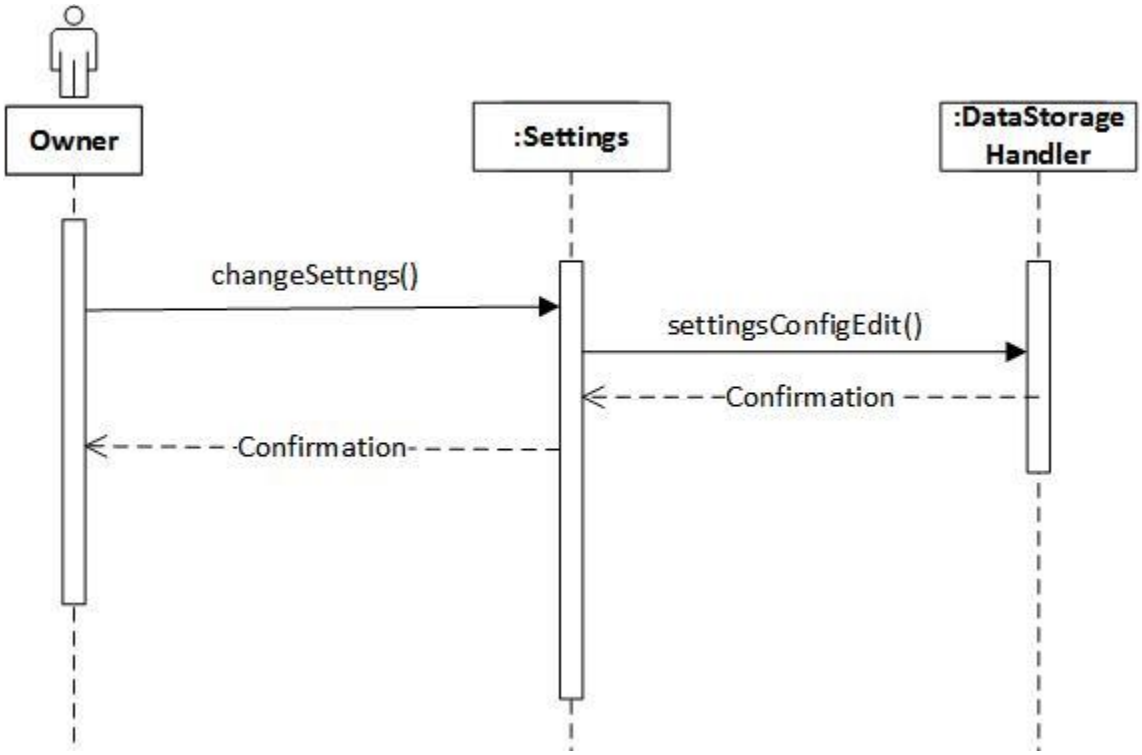

Figure 14-Change settings sequence diagram

6.2.6 Send message

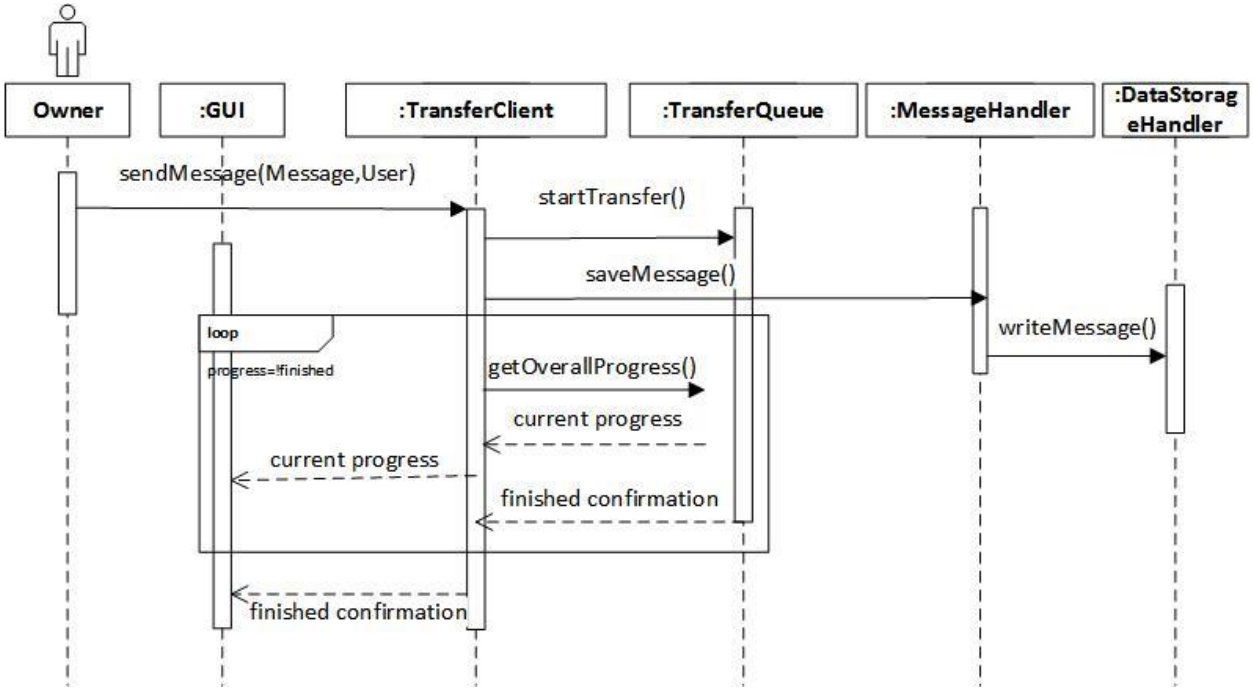

Figure 15-Send message sequence diagram

6.2.7 *Receive message*

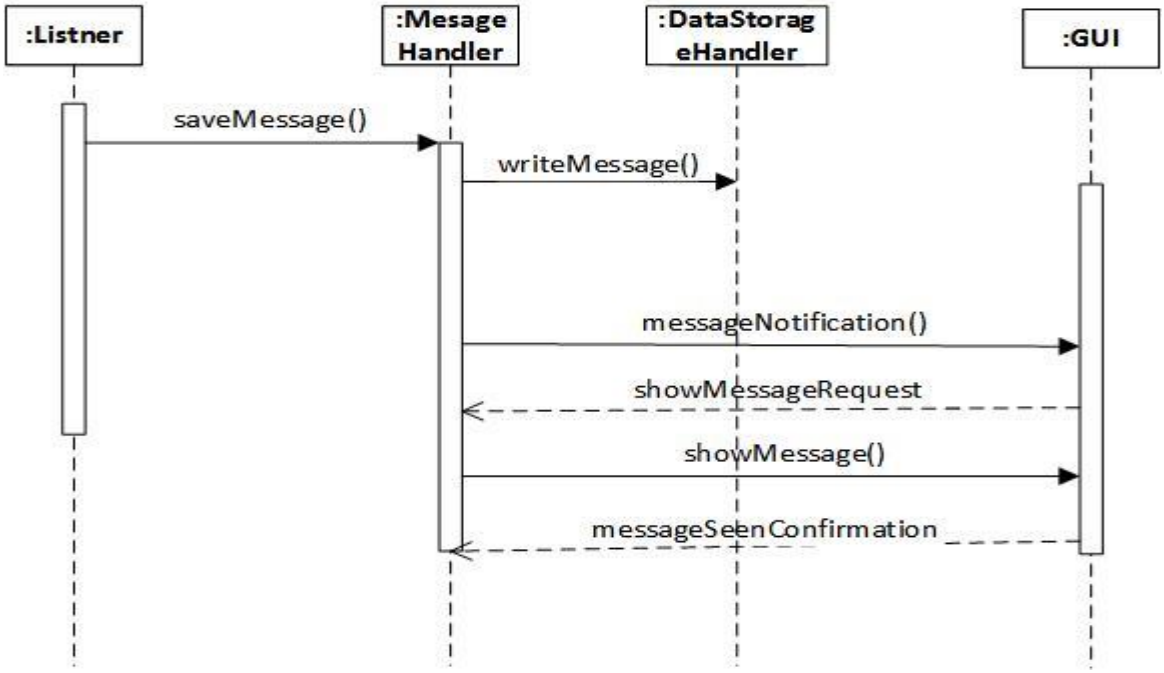

Figure 16-Receive message sequence diagram

7. Deployment View

The application is using socket programming. Using IP address with the dedicated port. Here it is expected to use port 7171 as default and user will be able to change it if needed

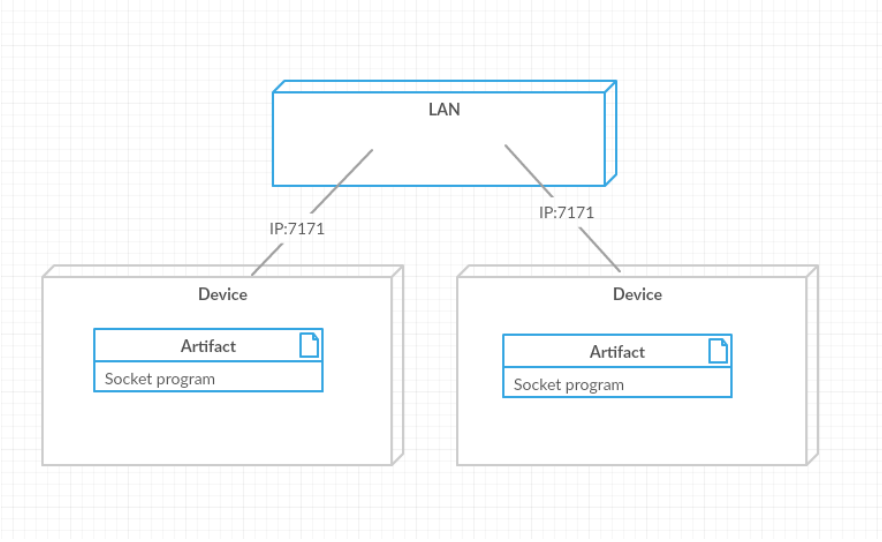

Figure 17-Deployment diagram

8. Implementation View

This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.

8.1 Overview

This system is following MVC architecture. Model, View and Control. That way the implementation is divided so that understanding and the modification become well organized.

8.2 Layers

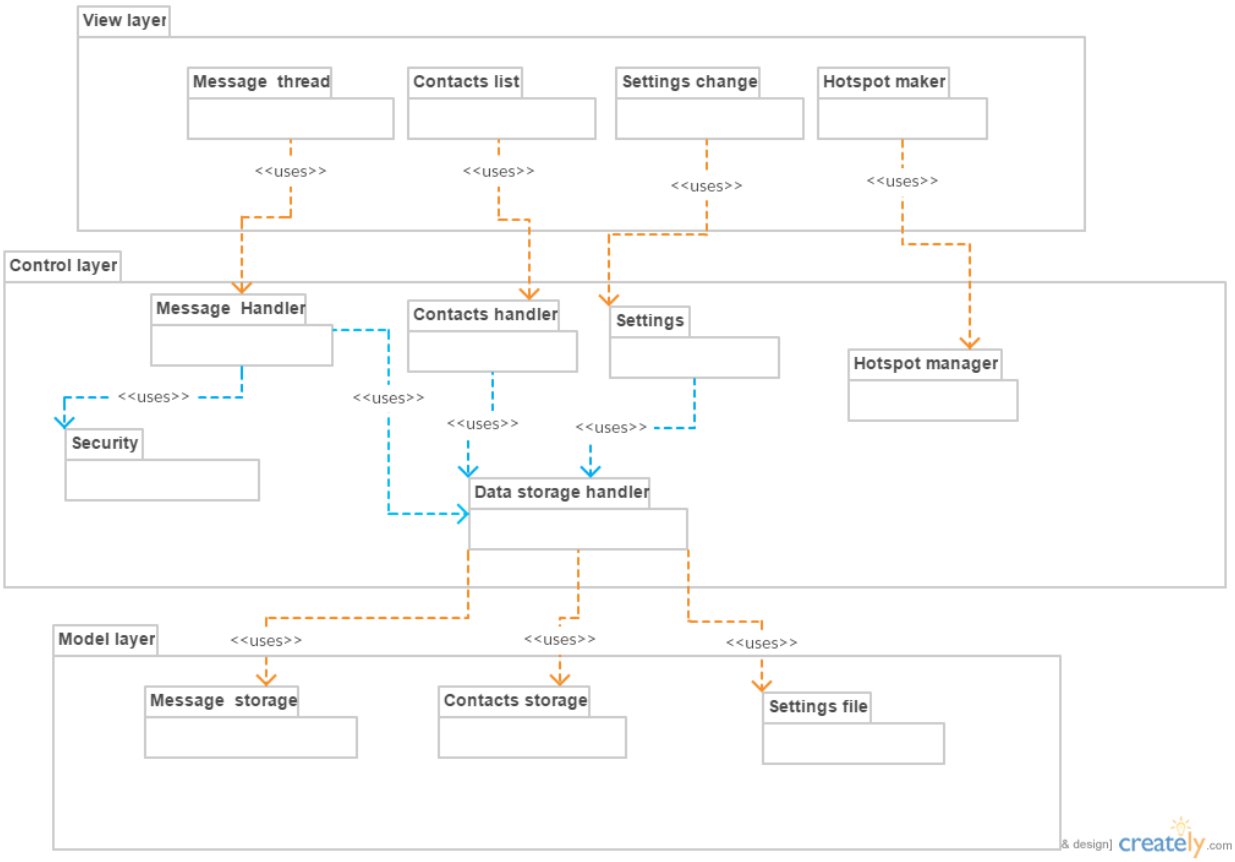

Figure 18-Package diagram

Main view components are handled by controlled by control layer components and those components are use data storage in the Model layer to do the tasks and create the views.

All the data handling activities are done using “Data handler” class. It has the required data storage with data manipulation functions. Then the other controller package class needed to use “Data handler” and use its function. This is useful if further implementation of data handling is change to another way like using databases.

9. Data View (optional)

This system uses persistent data storage for contacts saving, message saving and to save settings. It is not using sql databases. It uses serialized data objects as persistent data. And settings are saved in settings.config file.

10. Size and Performance

Target constraints:

Maximum of 3seconds response time is expected.

With human performance it is not needed a high transaction frequency. 5 transactions(message sending and receiving) per second is expected.

Expected size of the application

Hard disk space <= 100mb,
Ram usage at text message chat<=150mb.
Ram when file sharing <= 400mb.

11. Quality

11.1 Extensibility

Application should be able to easily extensible. It should be a main concern in development because this application expected to add more features in future.

11.2 Reliability

It should provide real time message receiving and sending function without delays or packet losses.

11.3 Portability

The application should be portable so it should run without getting too much installing time.

11.4 Security

this application should use suitable encryption method for security. The data it communicates should only be able to read by the required person.

12. References

[1] Object Management Group, Inc. “Welcome To OMG's BPM, UML, & CORBA Training Listings!” Internet: http://www.omg.org/gettingstarted/training.htm . 2017 [Mar. 17, 2017]