Software Design Description

(IEEE 1016-2009)

V1.0

NoNET

Prepared by FixIT

Gulsah SABIRSIZ
Gulnaz SHAIDOLDA
Ceyda TOSUN

METU - Department of Computer Engineering
CENG 492 Senior Design Project
Fall 2015-2016

TABLE OF CONTENTS

1. Overview
    1.1. Scope
    1.2. Purpose
    1.3.Intended Audience

2. Definitions

3. Design Viewpoints
    3.1 Introduction
    3.2 Context Viewpoint
        3.2.1 Context Diagram
        3.2.2 Use Case Diagram
            3.2.2.1 User Register with Phone Number
            3.2.2.2. Discover Devices
            3.2.2.3 Pick User from the Contact List
            3.2.2.4. Select a Device
            3.2.2.5. Write Message
            3.2.2.6. Send Messages to a Distant Peer
            3.2.2.7. Send Messages to a Near Peer
            3.2.2.8. Broadcast Messages
            3.2.2.9. Reply to the message
            3.2.2.10. Send Location
            3.2.2.11. Turn on Battery Save Mode
            3.2.2.12. Turn Off Battery Save Mode
            3.2.2.13. Make Visible
            3.2.2.14. Make Invisible
    3.3 Composition Viewpoint.
        3.3.1 Component Diagram
    3.4 Logical Viewpoint
        3.4.1 Class Diagram
    3.5 Interaction Viewpoint.
        3.5.1 Sequence Diagram.
    3.6 State Dynamics Viewpoint
        3.6.1 State chart Diagram
    3.7 Interface Viewpoint
        3.7.1 Register Page User Interface
        3.7.2 Main Page User Interface

1. Overview

1.1 Scope

This document contains the software design description of NoNET – An Emergency Communication system. It is prepared according to the “IEEE Standard for Information Technology– Systems Design – Software Design Descriptions – IEEE 1016 – 2009”. In this document, the design views are shown by using UML diagrams to provide an understanding of system architecture.

1.1 Purpose

This document is prepared to describe architecture and functional structure of NoNET – Emergency Communication Application. The main aim of the document is to give a general description for what the design elements are and how they interact with each other, what the system components and behaviors are and how they have a significance on the system operation.

1.1 Intended Audience

The intended audience of this document is software development team. The team can use this document to review and implement the project.

2. Definitions

UML	Unified Modeling Language
IEEE	Institute of Electrical and Electronics Engineering
SDD	System Design Document
Flooding	Broadcasting a message to all neighbors , then neighbors recursively broadcasts message so that all devices in the network gets the message
Ack	Ack messages sent to acknowledge that certain message is reached successfully
Adhoc network	A decentralized type of wireless network .It does not rely on a pre existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing

3. Design Viewpoints

3.1 Introduction

In this design documentation, below main design viewpoints will be explained.

- Context viewpoint
- Composition viewpoint
- Logical viewpoint
- Interaction viewpoint
- State dynamics viewpoint
- Interface viewpoint

We will use UML diagrams to make these viewpoints more understandable.

3.2 Context Viewpoint

The Context view of a system defines the relationships, dependencies, and interactions between the system and its environment - the people, systems, and external entities with which it interacts.

Design concerns: System scope and responsibilities, identity of external entities and services and data used, identity and responsibilities of external interfaces, other external interdependencies

Context diagram and use case diagrams are given for context viewpoint of the system.

3.2.1 Context Diagram

Context diagram represents the user outside the system and that could interact with that system. It shows the interactions and relations in the system.

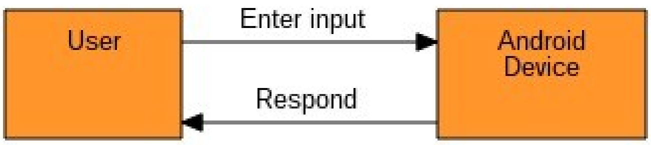


Figure 1- Context Diagram

3.2.2 Use Case Diagram

Use case diagram is used to capture the dynamic aspect of a system.

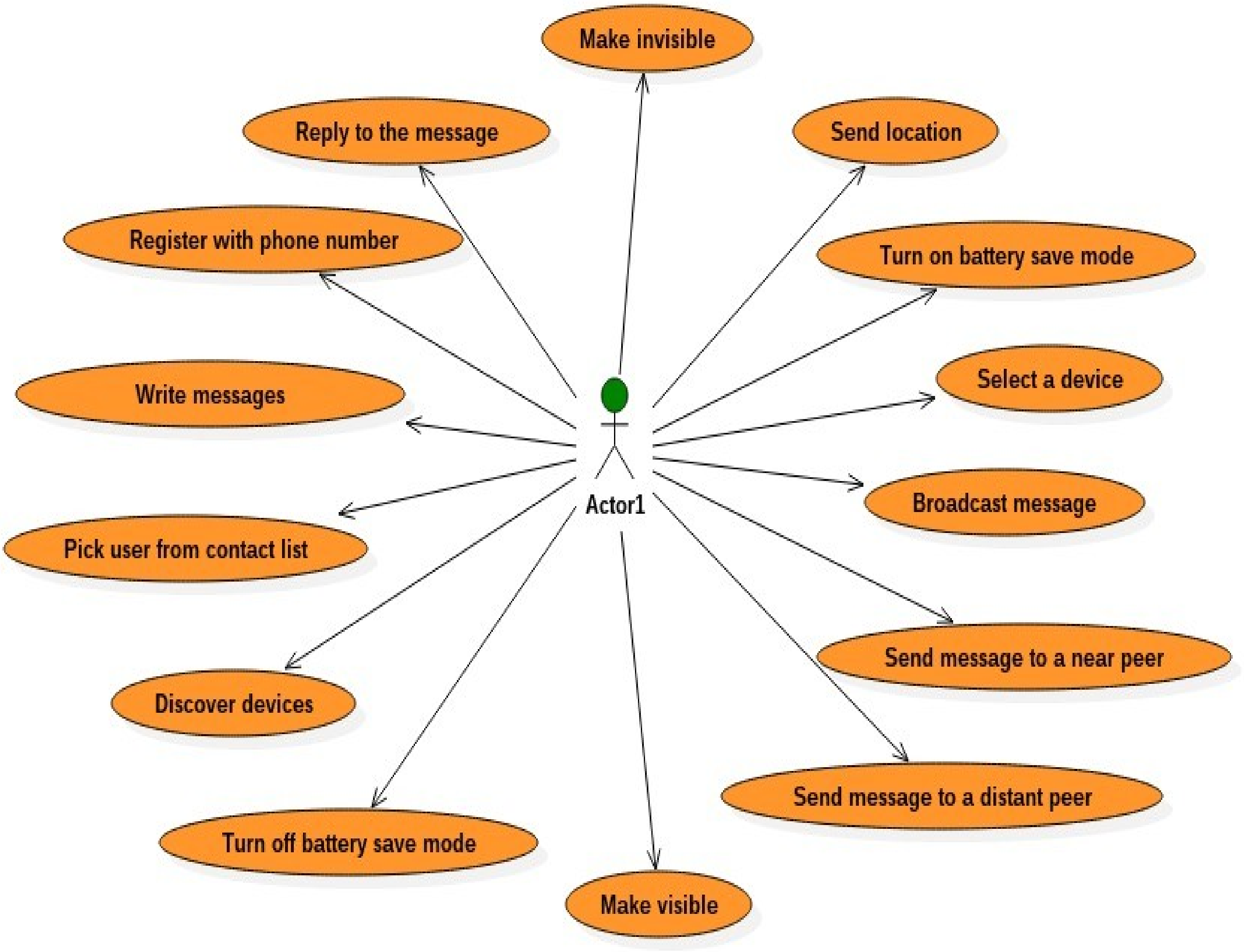

Make invisible
Reply to the message
Send location
Register with phone number
Turn on battery save mode
Write messages
Select a device
Actor1
Broadcast message
Pick user from contact list
Send message to a near peer
Discover devices
Send message to a distant peer
Turn off battery save mode
Make visible

Figure2 – Use Case Diagram

Use Case	Description
Register with phone number	Ordinary person has to register with phone number in order to use the application.
Discover Device	Ordinary person lists the information of devices and device users.
Pick user from the contact list	Ordinary person can select receiver from own contact list.
Select a device	Select devices and send message.
Write message	Ordinary person types a short message.
Send message to distant peer	Ordinary person can send message to a distant specific peer.
Send message to near peer	Ordinary person can send message to a near peer.
Broadcast message	Ordinary person can broadcast message to surrounding users.
Reply to the message	Ordinary person can reply to message.
Send Location	Ordinary person sends his location within the message.
Turn on battery save mode	Ordinary person can save battery using `guc modunu ac` button.
Turn off battery save mode	Ordinary person can attend Ad-Hoc network.
Make visible	Ordinary person can discovery nearby devices immediately using `gorunur ol` button.
Make invisible	Ordinary person can go back to the invisible mode.

3.2.2.1 User Register with Phone Number

Use Case Number	1
Use Case Name	Register with phone number
Summary	Ordinary person needs to register to the system
Actor	Ordinary person
Trigger	`Tamam` button
Precondition	Downloaded the NoNET application.
Scenario	• After user download the NoNET, s/he must register to

	system in order to use application.. • After filling required field with information, s/he can register to the system with `tamam` button. • `Tamam` button in the register page redirects user to the main page. • After user register the system, s/he can start to use the application.
Postcondition	After user registers the system, s/he can start to use the application.

3.2.2.2. Discover Devices

Use Case Number	2
Use Case Name	Discover Devices
Summary	Discover other devices.
Actor	Ordinary person
Trigger	Clicking to the NoNet Application
Precondition	Initiate the NoNET application **.**
Scenario	• When application is started, it discovers the near devices without clicking any button.
Postcondition	After application discovers devices, it communicates and sends messages them.

3.2.2.3 Pick User from the Contact List

Use Case Number	3
Use Case Name	Pick user from the contact list
Summary	User can select sender from own contact list
Actor	Ordinary person
Trigger	Clicking to phone contact picture
Precondition	Initiate the NoNET application **.**
Scenario	• When user select receiver from own contact list, s/he can send message this peer.
Postcondition	User send messages.

3.2.2.4. Select a Device

Use Case Number	4
Use Case Name	Select a Device
Summary	Select a device before sending message
Actor	Ordinary person
Trigger	Clicking to the receiver's phone number.
Precondition	User discovers to the near devices or select from the phone contact.
Scenario	• User select the receiver before sending message to a certain person.
Postcondition	User sends message.

3.2.2.5. Write Message

Use Case Number	5
Use Case Name	Write Message
Summary	Type a text message
Actor	Ordinary person
Trigger	Send
Precondition	Select receivers.
Scenario	• User writes a message by clicking on the text field. • Also, message content includes time of sent message and information about the user’s situation.
Postcondition	User sends messages.

3.2.2.6. Send Messages to a Distant Peer

Use Case Number	6
Use Case Name	Send Messages to a Distant Peer
Summary	User sends messages to a specific peer.
Actor	Ordinary person

Trigger	Send
Precondition	Selected to the peer and found the correct path to send messages to the peer.
Scenario	- User sends messages to a specific peer. - S/he sends messages to her/his family members to inform about their situations or to receive information from them.
Postcondition	Receive information from the distant peer.

3.2.2.7. Send Messages to a Near Peer

Use Case Number	7
Use Case Name	Send Messages to a Near Peer.
Summary	User sends a short message.
Actor	Ordinary person
Trigger	Send
Precondition	Selected device
Scenario	- User sends a short message to ask help to a person near to her/him.
Postcondition	The receiver can reply this message.

3.2.2.8. Broadcast Messages

Use Case Number	8
Use Case Name	Broadcast Messages
Summary	User broadcasts message to all people in the Ad-hoc network.
Actor	Ordinary person
Trigger	Send
Precondition	Discovered devices or select receiver.
Scenario	- User broadcasts a message to the surrounding people to ask help in emergency situations.
Postcondition	People will learn the location of people who is under the debris.

3.2.2.9. Reply to the message

Use Case Number	9
Use Case Name	Reply to the message
Summary	User can reply message.
Actor	Ordinary person
Trigger	Clicking to the coming message
Precondition	Receive a message
Scenario	• User want to reply incoming message as immediate as possible.
Postcondition	Sender take a message.

3.2.2.10. Send Location

Use Case Number	10
Use Case Name	Send Location
Summary	User sends the location of her/him.
Actor	Ordinary person
Trigger	Location Button
Precondition	Discovered devices.
Scenario	• User of our application sends the location of her/his attached to the message to the other people. • User clicks location button and location of her/him will be send.
Postcondition	People who come to the rescue learns her/his location.

3.2.2.11. Turn on Battery Save Mode

Use Case Number	11
Use Case Name	Turn on Battery save mode
Summary	User want to save device's battery.
Actor	Ordinary person
Precondition	User’s device’s battery should be lower than 30%.

Scenario	• When the user may be under debris and waiting for help, s/he can reject to transferring the received messages. Because, her/his device’s battery may be low.
Postcondition	User be rescued before end of the battery.

3.2.2.12. Turn Off Battery Save Mode

Use Case Number	12
Use Case Name	Turn off battery save mode
Actor	Ordinary person
Precondition	User’s device’s battery should be higher than 30%. User want to help other people using NoNET application.
Scenario	• When the user may be under debris and waiting for help, s/he can forward to the received messages. So, user can help himself/herself and other people.
Postcondition	User attend to the Ad-hoc network.

3.2.2.13. Make Visible

Use Case Number	13
Use Case Name	Make Visible
Summary	Ordinary person can discovery nearby devices immediately using `gorunur ol` button.
Actor	Ordinary person
Trigger	Visible Button
Precondition	User want to contact with specific peer immediately.
Scenario	• User contact with surrounding relatives as immediate as possible.
Postcondition	User see surrounding people.

3.2.2.14. Make Invisible

Use Case Number	14
Use Case Name	Make invisible
Summary	Ordinary person can go back to the invisible mode
Actor	Ordinary person
Trigger	Invisible Button
Precondition	User contact with surrounding people. Then, user want to save battery status.
Scenario	• User contact with surrounding relatives as immediate as possible. Then, user must save battery state.
Postcondition	User can conserve battery. Therefore, user's device used long time.

3.3 Composition Viewpoint

This section provides information about NoNET’s components and their relations with each other.

Design Concern: is the composition and modular assembly of systems in terms of subsystems and components.

3.3.1 Component Diagram

Component diagram of relevant system is given below.

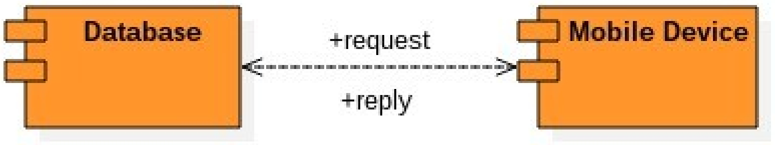


Figure 3 – Component Diagram

3.4 Logical Viewpoint

In this viewpoint, all classes that are going to be implemented for the NoNET will be explained. Relationships and hierarchies between classes are explained after the detailed explanation of classes.

Design Concerns: Logical viewpoints’ design concerns are static structure like classes,interfaces,and their relationships and reusing of types and implementations.

3.4.1 Class Diagram

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. NoNET has 4 class diagrams and they are explained below.

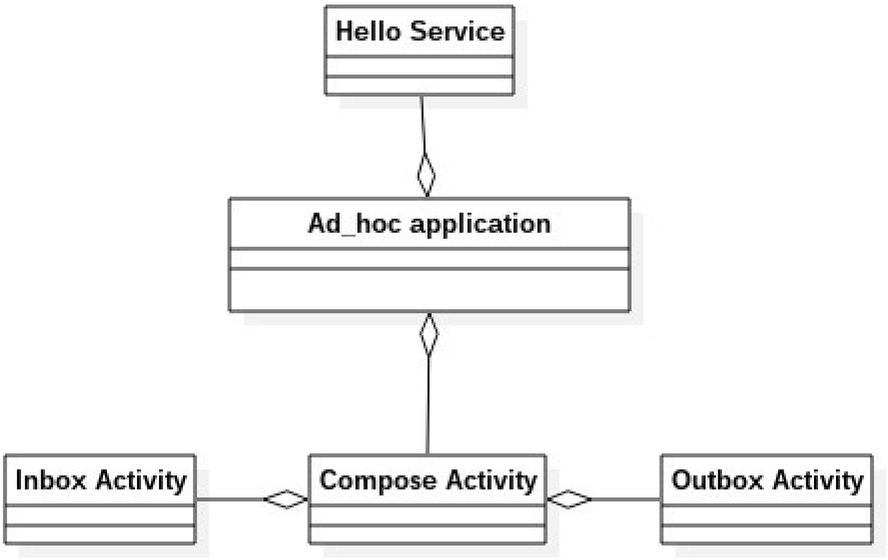


Figure 4– Application Layer Class Diagram

Hello service: Hello service periodically broadcasts a control datagram.
Ad hoc application: It enables objects to communicate with each other.
Compose Activity: It is the main activity of the application. It has inbox, outbox, location, send, broadcast, contact list buttons.
Inbox Activity: Received messages will be saved here.
Outbox Activity: Sent messages will be located here.

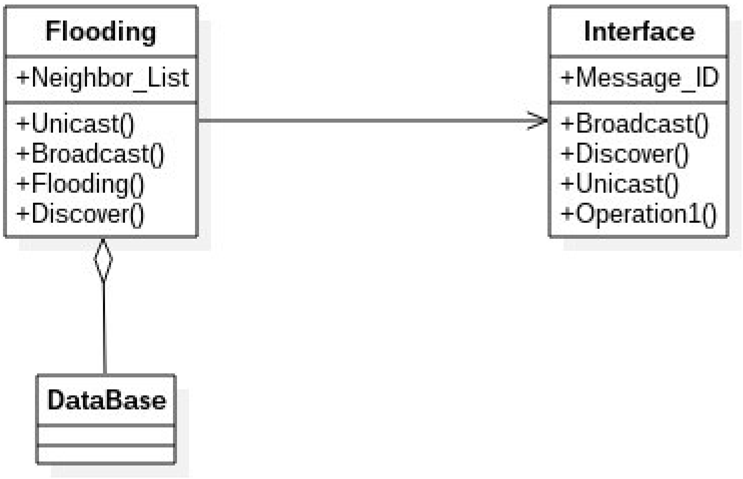


Figure 5– Ad hoc Network Layer Class Diagram

Flooding: It implements the flooding algorithm. It implements unicast and broadcast methods in the ad hoc network.

Database: It has Datagram table and Routing table. In this tables, we hold information about datagrams.

Interface: It used to discover, send and receive message in the ad hoc network.

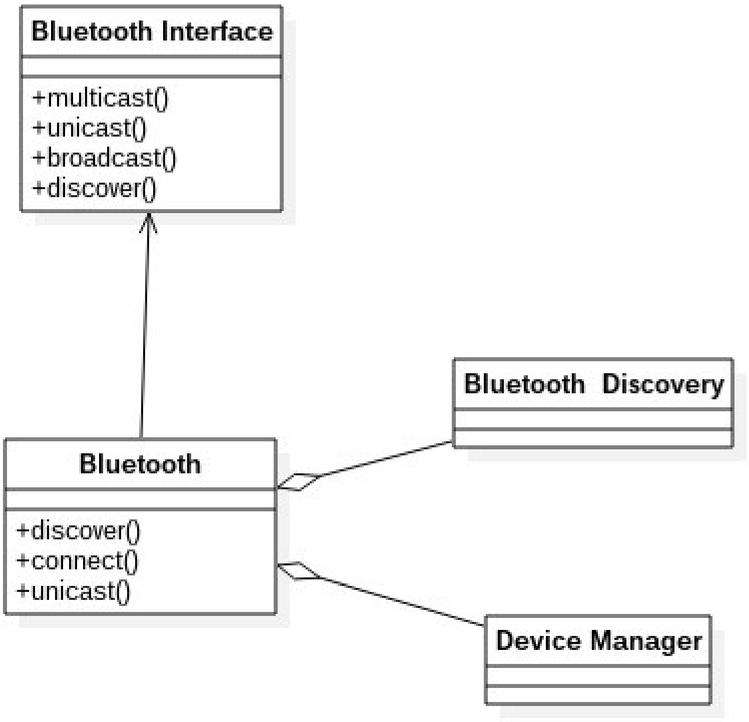


Figure 6– Local Network Layer Class Diagram

Bluetooth interface: It used to discover, send and receive message from nearby devices.
Bluetooth Discovery: It discovers the neighborhood devices.
Device manager: It holds last discovery list of the devices.
Bluetooth: It broadcasts, unicasts and multicasts message to nearby devices.

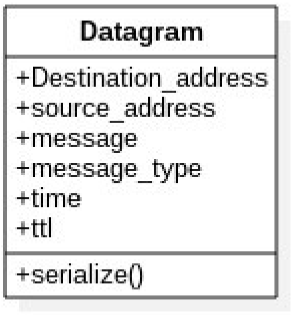


Figure 8 – Datagram Class Diagram

Datagram: It holds the message information. Unique identifier of a message is destination address, source address and the time message is created.

3.5 Interaction Viewpoint

Interaction viewpoint is provided through sequence diagrams to explain the main functionalities of modules of the project.

Design Concerns: The aim of this view is showing the flow of application running system. NoNET project has several work flows and some main sequences were shown here to illustrate functionalities of the project.

3.5.1 Sequence Diagram

A Sequence diagram shows how objects operate with one another and in what order. It shows object interactions arranged in time sequence. In our application, we have discovery, compose message, receive message, send message sequence diagrams.

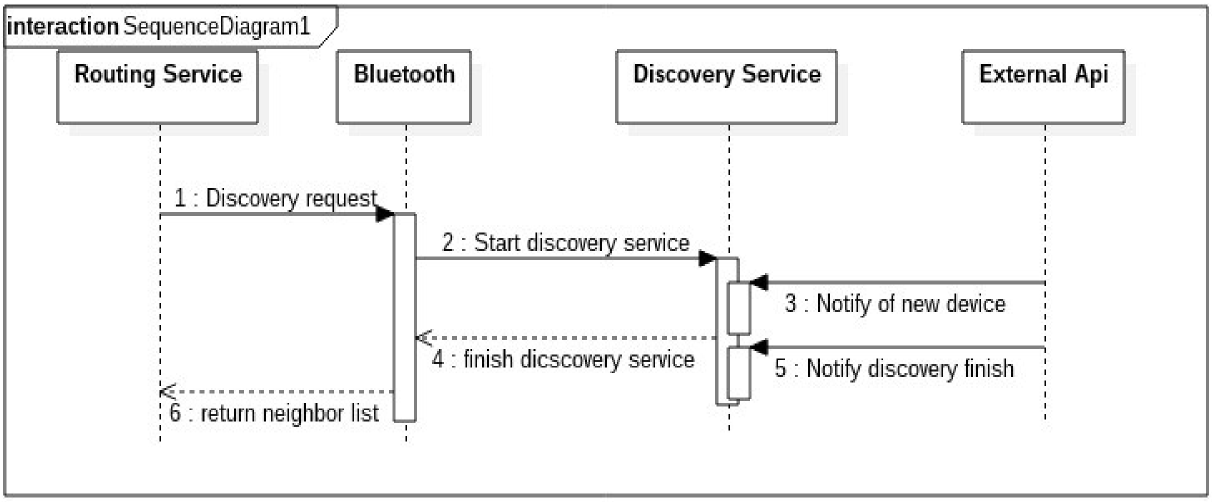


Figure 9 – Discovery Flow Sequence Diagram

The following flow describes the process of discovering nearby devices. The flow starts when the routing service decides to issue a discovery request.

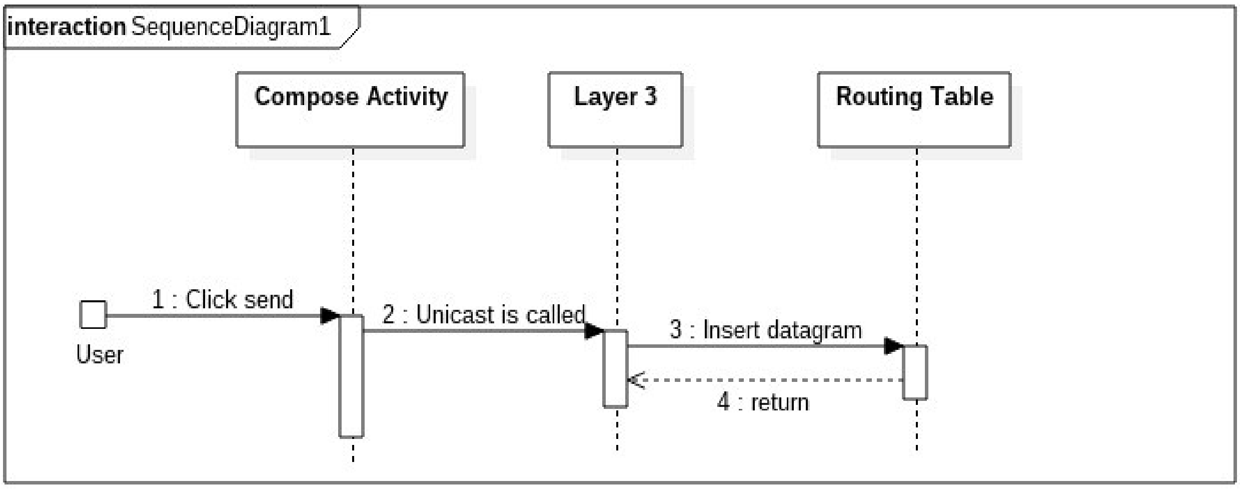


Figure 10 – Compose Message Sequence Diagram

The compose flow is responsible for handling user requests for sending a message. The flow starts with the user choosing destinations, and writing a message in the compose activity. The flow ends with the datagram being inserted into the datagram table.

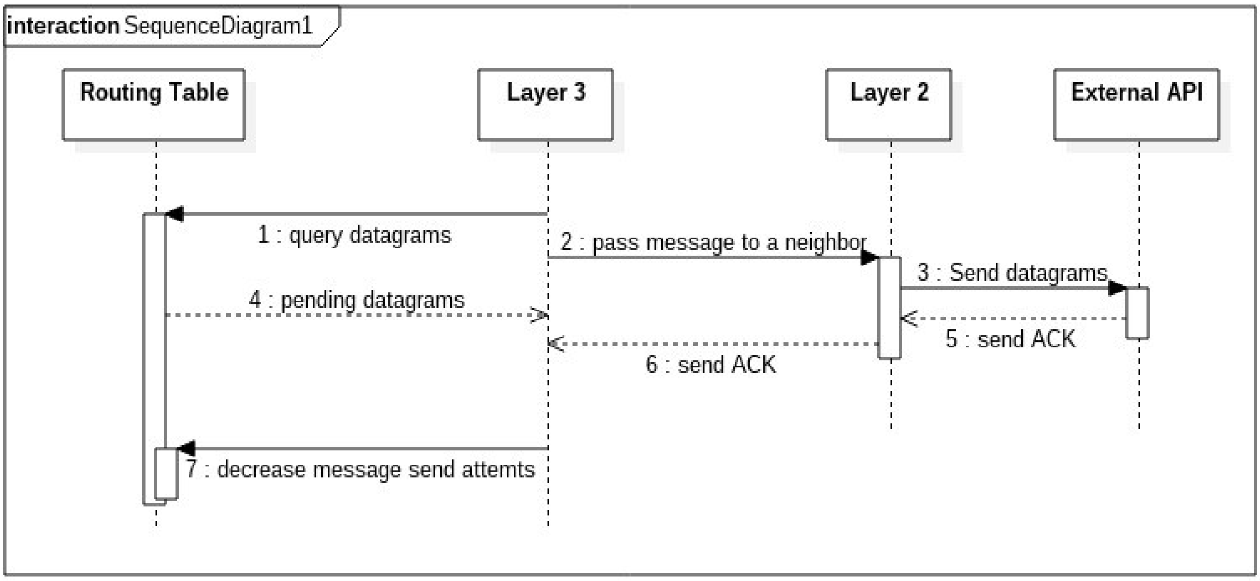


Figure 11 – Send Message Sequence Diagram

The Send Message flow is responsible for sending the datagram messages.

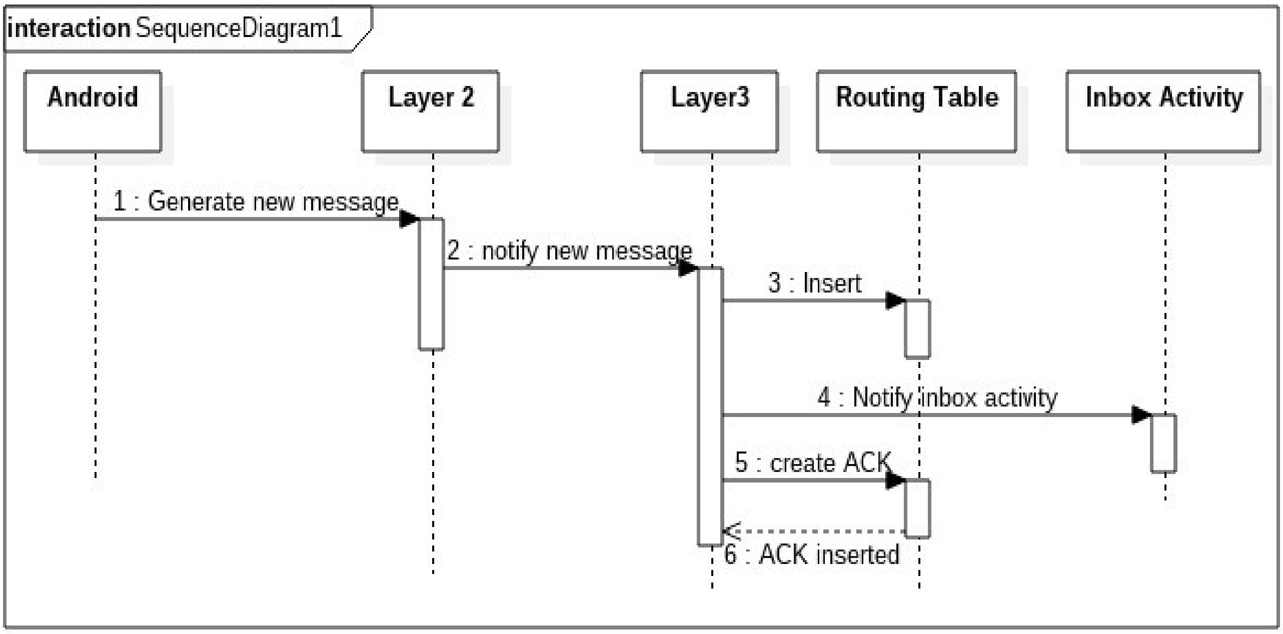


Figure 12 – Receive Message Sequence Diagram

Receive message flow starts with an indication from the Android OS of new received data. It ends with notifying Inbox activity.

3.6 State Dynamics Viewpoint

State viewpoint deals with behavior of the system when some particular action happened in the program flow. It shows how the application reacts to that action. This viewpoint is shown with the help of UML state chart diagram.

Design Concerns: States and reaction of those states to events.

3.6.1 State Chart Diagram

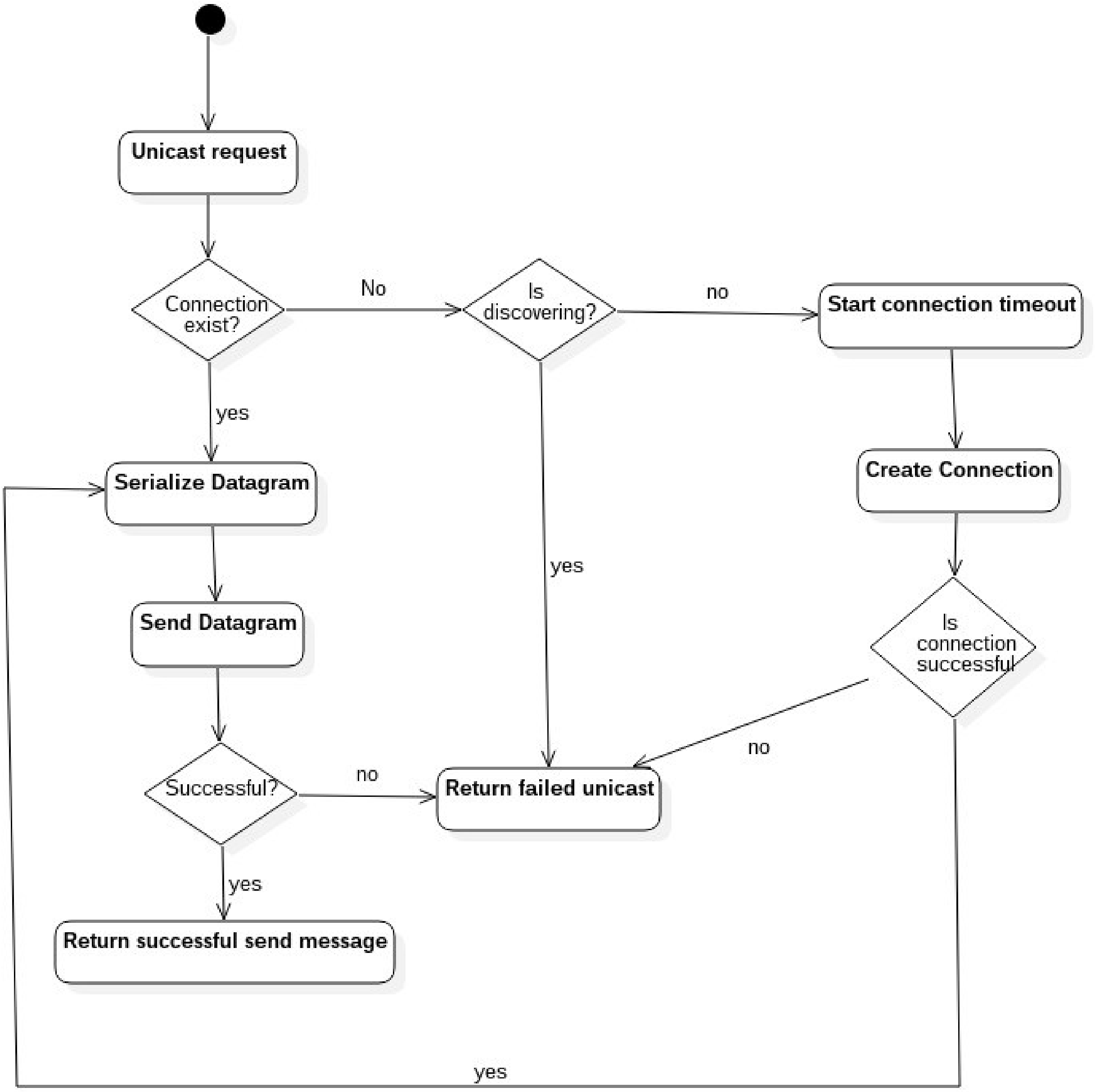


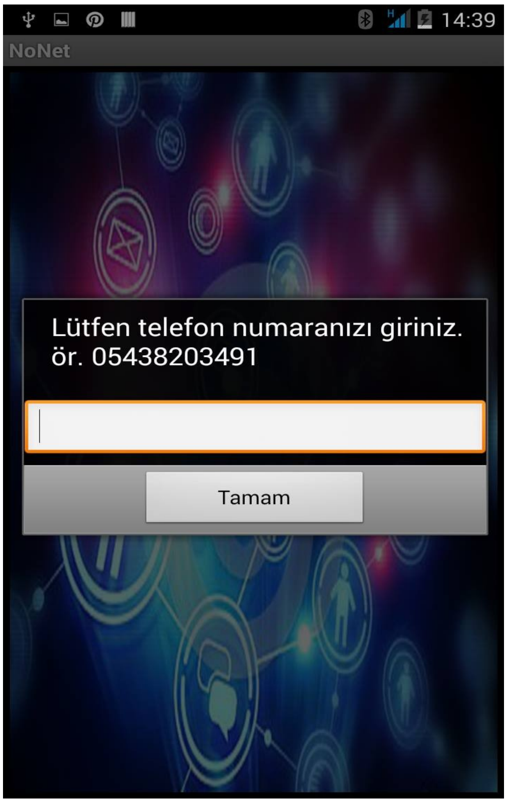
Figure 13 – Send Message State Chart Diagram

3.7 Interface Viewpoint

In this section user interface features are provided.

Design Concerns: Design concerns of interface viewpoint includes service definition and service access of the system.

3.7.1 Register page



Lütfen telefon numaranızı giriniz.
ör. 05438203491

Figure 14 – Register page user interface

This is a registration page of NoNET. After installing the application, user can enter his/her phone number in the box so that s/he will be registered to the system.

3.7.2 Main page

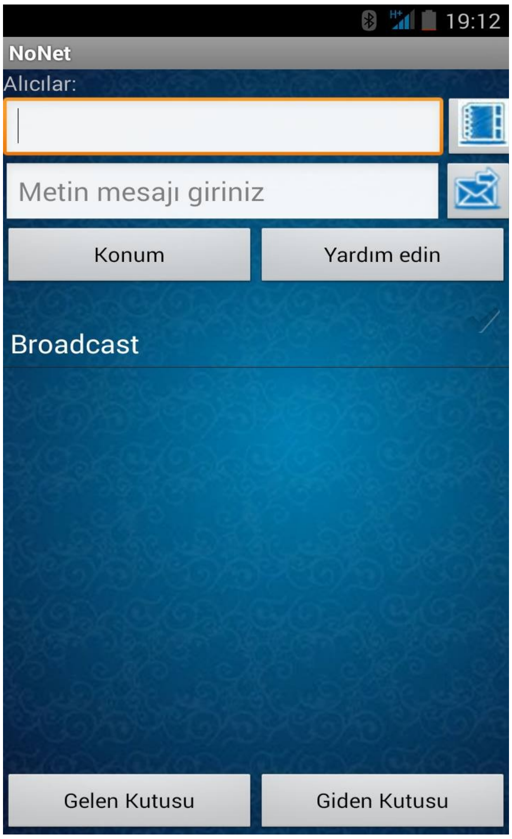


Figure 15 – Main page user interface

This is the main page of NoNET. User can write message here. User can access to the inbox and outbox of the application. Moreover s/he can broadcast and unicast message to other users. Also, accessing to the contact list is possible.