Software Requirements Specification

NFT

Not For Terrorists Secure Messaging App

James Bunting

Jonathon Herrera

Jorge Gasca-Cervantes

Justin De Luna

Ricardo Lesmes

March 26, 2017

Table of Contents

[List of Figures 3](#_Toc478328716)

[List of Tables 4](#_Toc478328717)

[SRS 5](#_Toc478328718)

[Section 1: Introduction 5](#_Toc478328719)

[1.1 Purpose of this SRS 5](#_Toc478328720)

[1.2 Scope 5](#_Toc478328721)

[1.3 Glossary, Acronyms, Abbreviations, Notational conventions 5](#_Toc478328722)

[1.4 Overview 6](#_Toc478328723)

[Section 2 7](#_Toc478328724)

[2.1 Use Case Diagram 7](#_Toc478328725)

[2.2 Brief Description of Use Case Diagram 8](#_Toc478328726)

[Section 3 8](#_Toc478328727)

[3.1 Use Case Tables 8](#_Toc478328728)

[Section 4 13](#_Toc478328729)

[4.1. UML Class Diagram 13](#_Toc478328730)

[4.2 UML Class Diagram Description 14](#_Toc478328731)

[4.3 UML Sequence Diagrams 14](#_Toc478328732)

[4.4 UML State Machine Diagrams 22](#_Toc478328733)

[4.5 Prototype Interfaces 24](#_Toc478328734)

[Section 5 28](#_Toc478328735)

[5.1 Scrum Backlog 28](#_Toc478328736)

[5.2 Non-Functional Requirements 28](#_Toc478328737)

# List of Figures

[Figure 1: 2.1.1 Use Case Diagram 7](#_Toc478328738)

[Figure 2: 4.3.1 UML Sequence Diagram UC-0001 Add Account 14](#_Toc478328739)

[Figure 3: 4.3.2 UML Sequence Diagram UC-0002 Lock/Unlock Account 15](#_Toc478328740)

[Figure 4: 4.3.2 UML Sequence Diagram UC-0003 Delete Account 16](#_Toc478328741)

[Figure 5: 4.3.4 UML Sequence Diagram UC-0004 Login 17](#_Toc478328742)

[Figure 6: 4.3.5 UML Sequence Diagram UC-0005 Send Message 18](#_Toc478328743)

[Figure 7: 4.3.6 UML Sequence Diagram UC-0006 Add Contact 19](#_Toc478328744)

[Figure 8: 4.3.7 UML Sequence Diagram UC-0007 View Message 20](#_Toc478328745)

[Figure 9: 4.3.3 UML Sequence Diagram UC-0008 Delete message/conversation 21](#_Toc478328746)

[Figure 10: 4.4.1 UML State Machine Diagram: Message 22](#_Toc478328747)

[Figure 11: 4.4.2 UML State Machine Diagram: Server Request 23](#_Toc478328748)

[Figure 12: 4.5.1 App Login Screen 24](file:///C:\Users\justi_000\Documents\Justins%20Docs\School\Spring2017\SoftwareEngineering\SEProject1-master\SEProject1-master\NFTFinalSRS.docx#_Toc478328749)

[Figure 13: 4.5.2 App Contacts List 25](#_Toc478328750)

[Figure 14: 4.5.3 App Message Unlock Pattern Screen 26](#_Toc478328751)

[Figure 15: 4.5.4 App Read Message Screen 27](#_Toc478328752)

# List of Tables

[Table 1: UC-0001 8](#_Toc478328753)

[Table 2: UC-0002 9](#_Toc478328754)

[Table 3: UC-0003 9](#_Toc478328755)

[Table 4: UC-0004 10](#_Toc478328756)

[Table 5: UC-0005 11](#_Toc478328757)

[Table 6: UC-0006 11](#_Toc478328758)

[Table 7: UC-0007 11](#_Toc478328759)

[Table 8: UC-0008 12](#_Toc478328760)

[Table 9: Scrum Backlog 28](#_Toc478328761)

[Table 10: Non-Functional Requirements 28](#_Toc478328762)

# SRS

## Section 1: Introduction

### 1.1 Purpose of this SRS

The purpose of this document is to explain in detail the requirements for the “Not for terrorists secure messaging app”. This document will also explain the purpose of the aforementioned system. In conclusion, this document will act as a proposal to customers as well as a reference for the development team.

### 1.2 Scope

The purpose of this application is to be able to send highly secure messages to those with the same app. Messages will be deleted at least 5 minutes after being read, and will automatically delete after 8 hours. Customers will be able to add encryption to their messages, as well as ten-point security patterns. The makers of this app will not have a way to view these messages, and will have minimal access to the back-end. Passwords will be only be kept on the developers servers, under standard encryption. Accounts can only be created by admins; usernames and passwords will be randomized and be at least 10 characters.

This application needs data-service or Wi-Fi to send messages.

### 1.3 Glossary, Acronyms, Abbreviations, Notational conventions

* + 1. Glossary

|  |  |
| --- | --- |
| Term | Definitions |
| Actor | A user that will interact with the system |
| Logged in/on | When a user has provided the correct credentials and is currently allowed to navigate the system |
| NFT(Not for Terrorists Secure Messaging App) | An android application that will be developed according to the requirements described in this SRS |
| System Requirements Specification | A document that lays out the functions and constraints of a system to be developed |
| Encryption | A method of making data hard to read and decode |
| User | A person who will be interacting with the portal |
| Android Application | A software application that runs on an android device |

* + 1. Acronyms-

SRS - System Requirements Specifications

UC - Use case

UML - Unified Model Language

* + 1. Abbreviations-

Etc. - etcetera

N/A - Not applicable

App - application

* + 1. Notational conventions-

Section titles will be Cambria light and size 16, titles will be Cambria and size 13, and subtitles will be Cambria and size 11. Use case tables are labeled “UC-“, followed by the case number.

### 1.4 Overview

The document consists of a UML case diagram of all identifiable use cases, with detailed descriptions of each. Furthermore, each use case will have a corresponding sequence diagram. The next section gives a detailed UML class diagram, showing all actors, relations, etc. A brief description will also be provided.

Next will be two state machines for use cases of interest. Functional and non-functional requirements will be documented. Use case stories will be next, followed lastly by the appendix, containing minutes from the stakeholder meetings.

## Section 2

### 2.1 Use Case Diagram

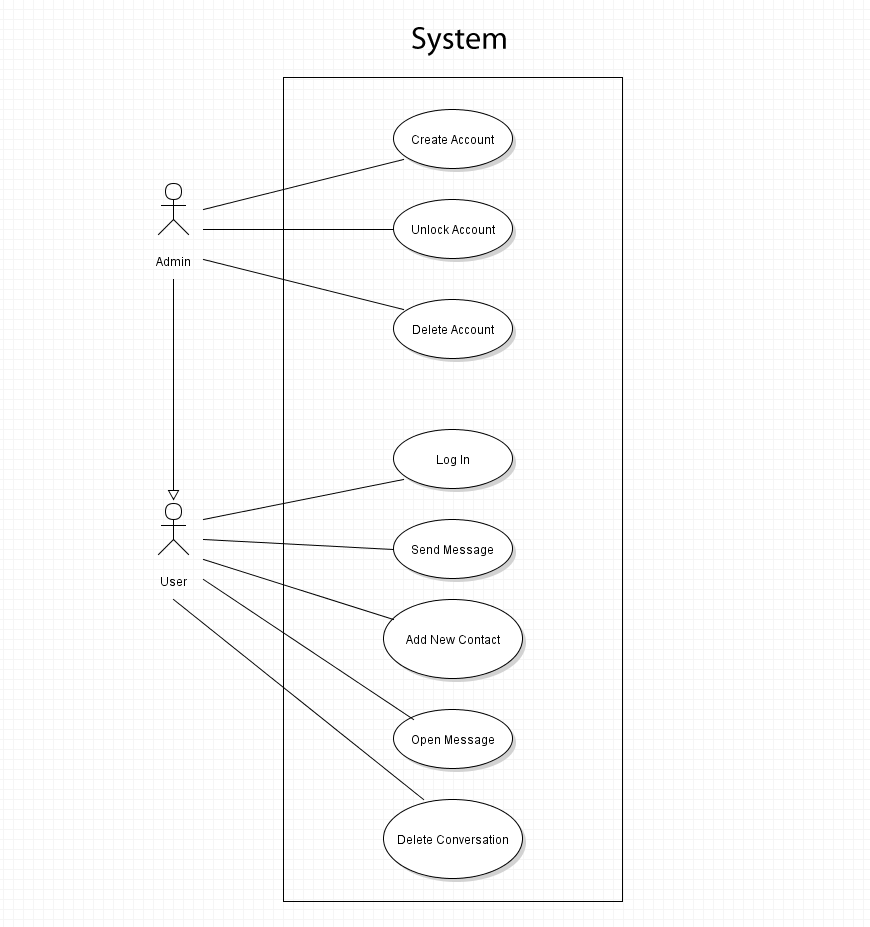


Figure 1: 2.1.1 Use Case Diagram

### 2.2 Brief Description of Use Case Diagram

The Use case diagram displays the 2 actors, the admin and the user and the cases they would be involved in. The list of use cases is as follows: Admins: UC-0001 Create Account; UC- 0002-Lock/Unlock Account; UC- 0003-Delete Account; All Users: UC-0004-Login; UC-0005-Send Message; UC- 0006-Add new Contact; UC- 0007-View Message; UC- 0008-Delete Conversation.

## Section 3

### 

### 3.1 Use Case Tables

For Admins

|  |  |
| --- | --- |
| Use Case Name | Create Account |
| Use Case ID | UC-0001 |
| Description | Creating an account for another User |
| Actors | Admin, System |
| Pre-Conditions | Admin is logged into an Administration account, user who is receiving new profile has passed company security checks. |
| Post-Conditions | Admin has created an account |
| Normal Use Scenario | 1. Admin clicks on ‘Create Account’ button in the toolbar 2. System generates a random 10-15 digit number ID. 3. System checks to see if number matches an existing account\* 4. System generates a random password using numbers letters and special characters 5. System checks if password matches an existing account\* 6. System returns ID and password to Admin. 7. System asks Admin to confirm creation. 8. Admin confirms 9. System creates account with generated ID and password |
| Alternative Use Cases | 3a. System finds a match  3b. System generates new ID  5a. System finds a match  5b. System generates new password |

Table 1: UC-0001

|  |  |
| --- | --- |
| Use Case Name | Lock/Unlock Account |
| Use Case ID | UC-0002 |
| Description | Admin locks/unlocks account |
| Actors | System, Admin |
| Pre-Conditions | Admin is logged in and has a list of user accounts. |
| Post-Conditions | Account is locked/unlocked. |
| Normal Use Scenario | 1. Admin opens app 2. Admin types in ID and password 3. System checks account data for a match\* 4. App unlocks and shows received messages 5. Admin navigates to users page. 6. Admin selects user to edit. 7. Admin selects lock/unlock account. |
| Alternative Use Cases | N/A |

Table 2: UC-0002

|  |  |
| --- | --- |
| Use Case Name | Delete Account |
| Use Case ID | UC-0003 |
| Description | Admin deletes an account |
| Actors | System, Admin |
| Pre-Conditions | User has an account; Admin has list of user accounts. |
| Post-Conditions | Account is deleted |
| Normal Use Scenario | 1. Admin opens app 2. Admin types in ID and password 3. System checks account data for a match\* 4. App unlocks 5. Admin navigates to users page 6. Admin selects user to edit. 7. Admin selects delete 8. Admin confirms deletion of account |
| Alternative Use Cases | 3a. User incorrectly enters data  3b User incorrectly enters data 3 times  3c. System locks the app  3d. User must contact Admin outside of app to unlock |

Table 3: UC-0003

User Use Cases

\*These cases are also usable by Admins

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case ID | UC-0004 |
| Description | User logs into their account |
| Actors | User, System, Admin |
| Pre-Conditions | User has an account; User has downloaded the app onto their mobile device |
| Post-Conditions | User is logged into account |
| Normal Use Scenario | 1. User opens app 2. User types in ID and password 3. System checks account data for a match\* 4. App unlocks and shows received messages |
| Alternative Use Cases | 3a. User incorrectly enters data  3b User incorrectly enters data 3 times  3c. System locks the app  3d. User must contact Admin outside of app to unlock |

Table 4: UC-0004

|  |  |
| --- | --- |
| Use Case Name | Send Message |
| Use Case ID | UC-0005 |
| Description | User sends a message to another User |
| Actors | User, System |
| Pre-Conditions | Both Users have accounts and are logged in. Both Users must have an agreed upon encryption code and or security pattern |
| Post-Conditions | A message is sent to another user |
| Normal Use Scenario | 1. User types in recipient ID or opens chat with desired recipient 2. User types a message into the message bar 3. User chooses time after message is read that it should be deleted. 4. User chooses whether to add encryption 5. User chooses whether to add pattern 6. User hits send button 7. System sends message to recipient ID |
| Alternative Use Cases | N/A |

Table 5: UC-0005

|  |  |
| --- | --- |
| Use Case Name | Add new contact |
| Use Case ID | UC-0006 |
| Description | User adds a new contact |
| Actors | User, System |
| Pre-Conditions | User is logged in and on contact screen |
| Post-Conditions | New Contact is Added |
| Normal Use Scenario | 1. User selects add contact 2. User enters contact data 3. User confirms contact 4. User exits created contact |
| Alternative Use Cases | 3a. User cancels contact creation |

Table 6: UC-0006

|  |  |
| --- | --- |
| Use Case Name | View message |
| Use Case ID | UC-0007 |
| Description | User views message |
| Actors | User, System |
| Pre-Conditions | User is on contacts screen |
| Post-Conditions | User can view message |
| Normal Use Scenario | 1. User selects a contact with an unviewed message 2. User enters encryption key 3. User views message 4. User deletes message 5. User returns to contacts screen |
| Alternative Use Cases | 2a. User enters pattern  2b. User enters pattern and key  3a. Message is deleted due to user failing to enter the correct code  4a. User exits without deleting message |

Table 7: UC-0007

|  |  |
| --- | --- |
| Use Case Name | Delete conversation |
| Use Case ID | UC-0008 |
| Description | User deletes conversation |
| Actors | User, System |
| Pre-Conditions | User is on contacts and has had a conversation |
| Post-Conditions | Conversation is deleted |
| Normal Use Scenario | 1. User selects a contact 2. User presses delete conversation |
| Alternative Use Cases | N/A |

Table 8: UC-0008

## 

## Section 4

### C:\Users\justi_000\Documents\Justins Docs\School\Spring2017\SoftwareEngineering\SEProject1-master\SEProject1-master\UMLsecureapp.PNG4.1. UML Class Diagram

Figure 4.1.1

Figure 1: 4.1.1 UML Class Diagram

### 4.2 UML Class Diagram Description

The UML Diagram contains 5 classes: User, Message, Conversation, Admin, and Main. There is interaction between the server and the user, admin, and main classes, which are all connected to an interface as well. User stores the username and passwords. Message stores the data the message consists of. Admin contains the ability to create, lock, unlock, and delete accounts.

### 4.3 UML Sequence Diagrams

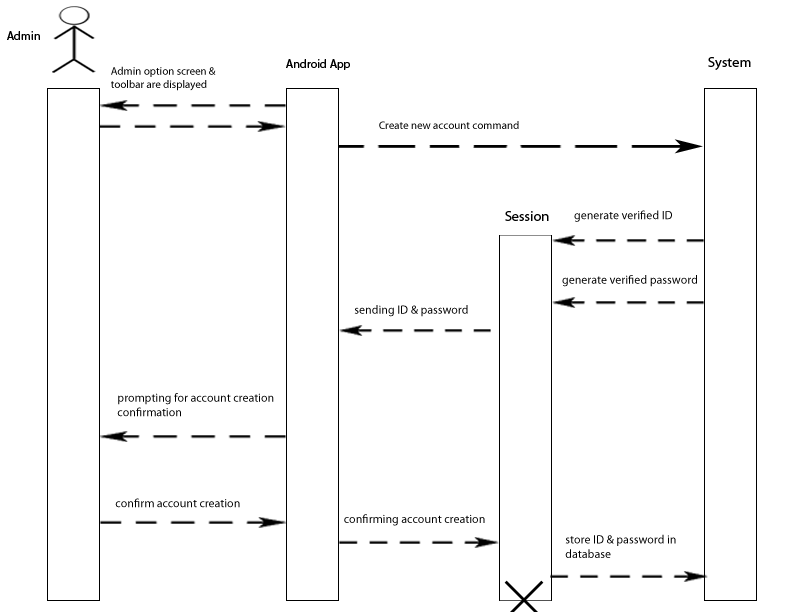


Figure 2: 4.3.1 UML Sequence Diagram UC-0001 Add Account

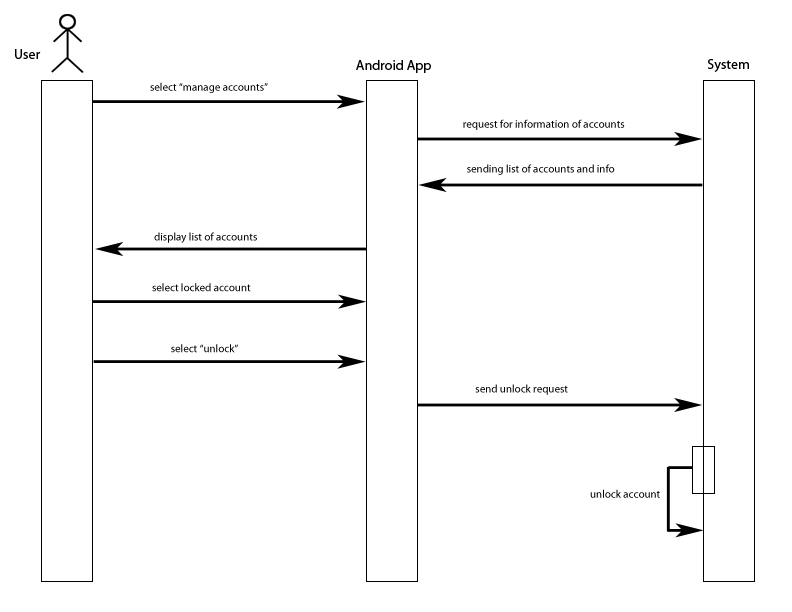


Figure 3: 4.3.2 UML Sequence Diagram UC-0002 Lock/Unlock Account

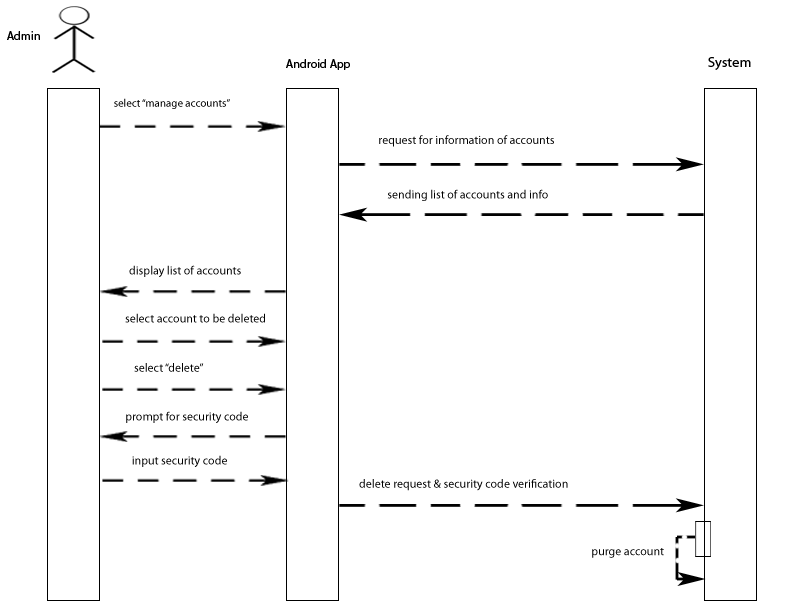


Figure 4: 4.3.2 UML Sequence Diagram UC-0003 Delete Account

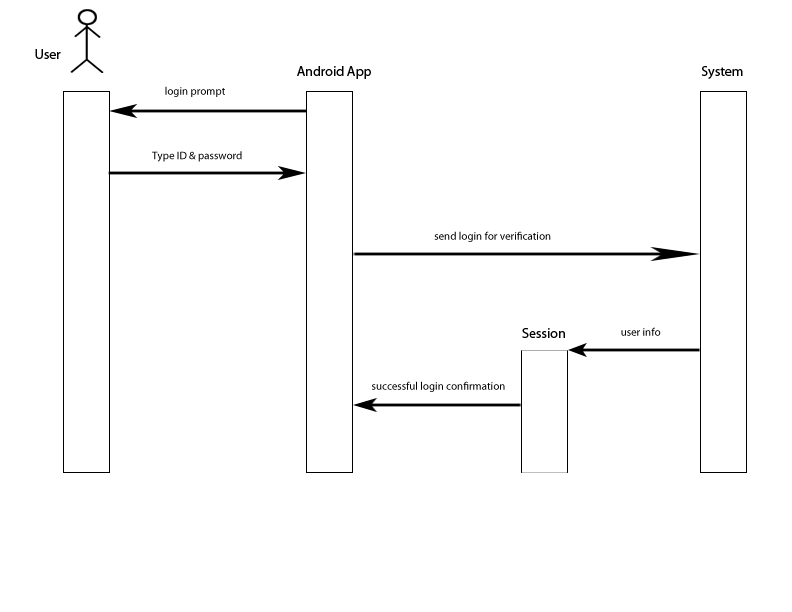


Figure 5: 4.3.4 UML Sequence Diagram UC-0004 Login

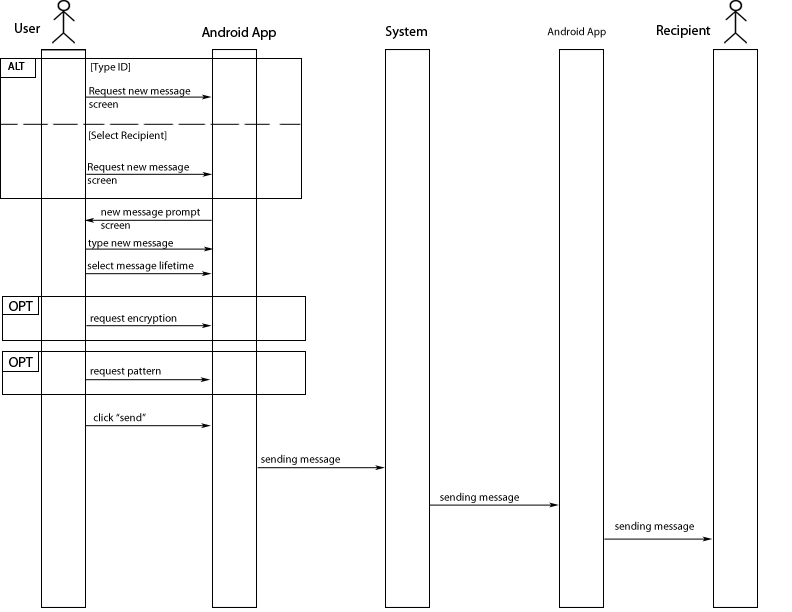


Figure 6: 4.3.5 UML Sequence Diagram UC-0005 Send Message

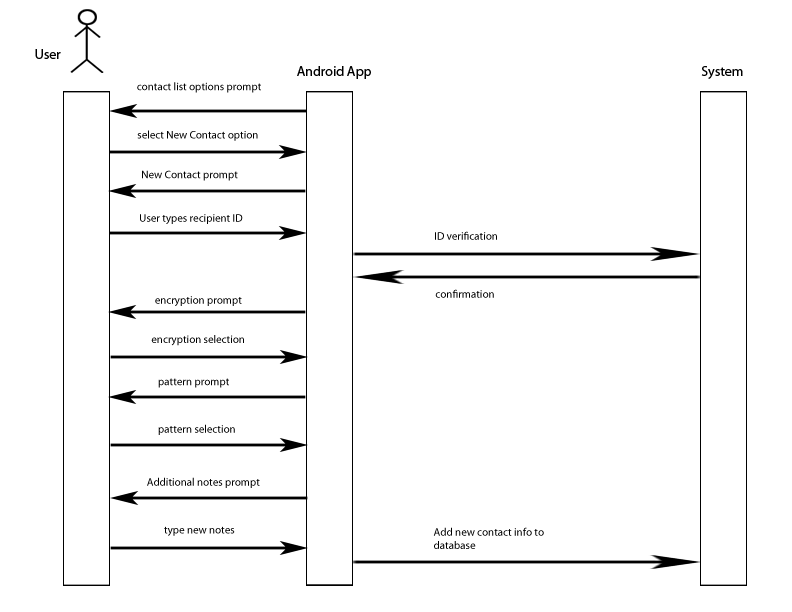


Figure 7: 4.3.6 UML Sequence Diagram UC-0006 Add Contact

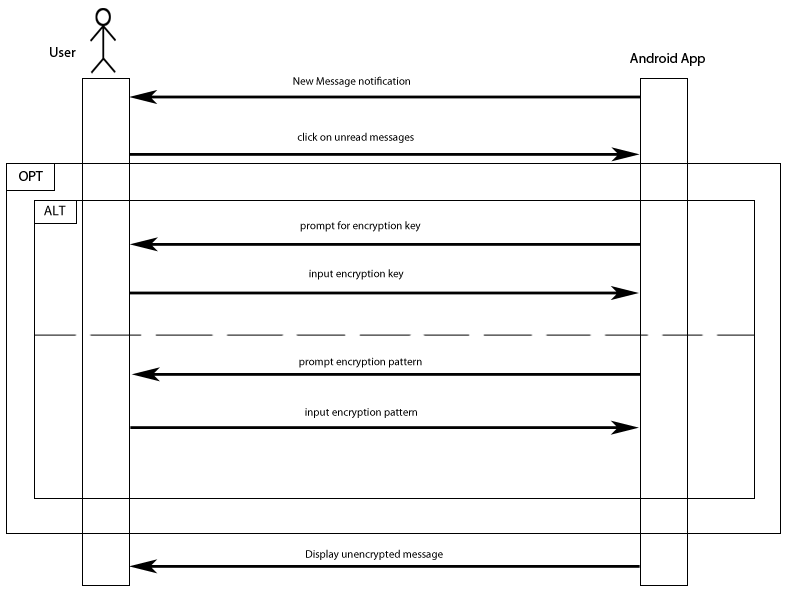


Figure 8: 4.3.7 UML Sequence Diagram UC-0007 View Message

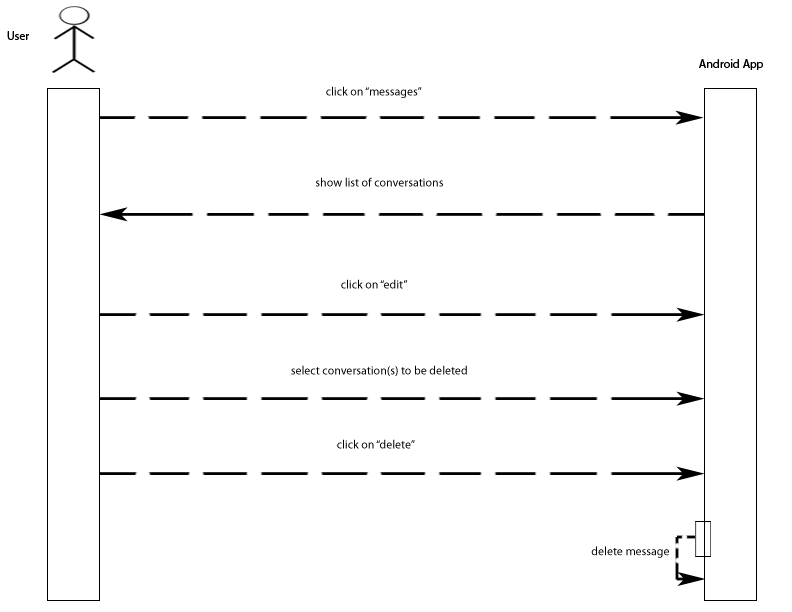


Figure 9: 4.3.3 UML Sequence Diagram UC-0008 Delete message/conversation

### 4.4 UML State Machine Diagrams

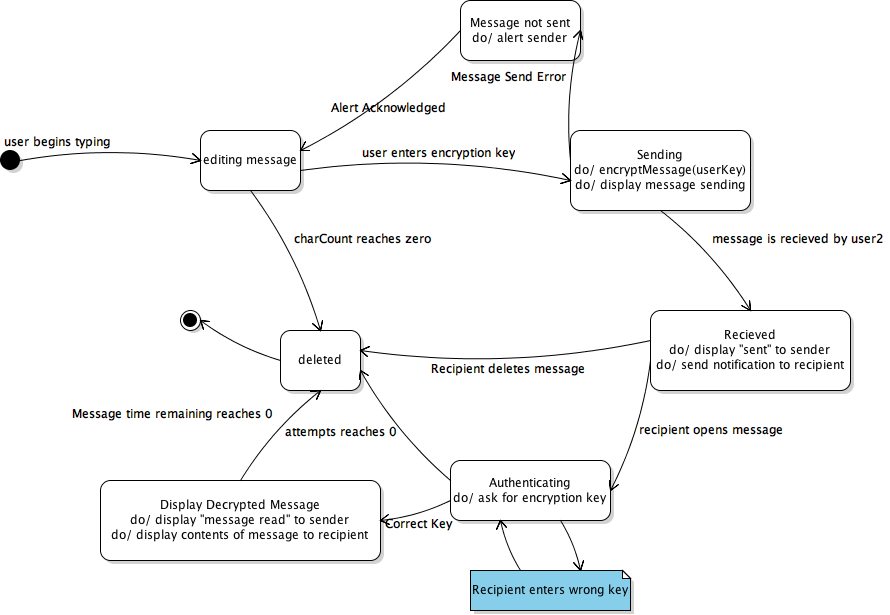


Figure 10: 4.4.1 UML State Machine Diagram: Message

Message State Machine Diagram Description: This state machine runs through the life cycle of a message. It is triggered by the user typing and flows into an editing state, encryption selection state, sending state, received state, authentication state, display state, and deletion. If the message fails to reach receiver an alert is sent to sender. The authentication state is repeatable on wrong entry until the attempts left reaches 0, which deletes the message.

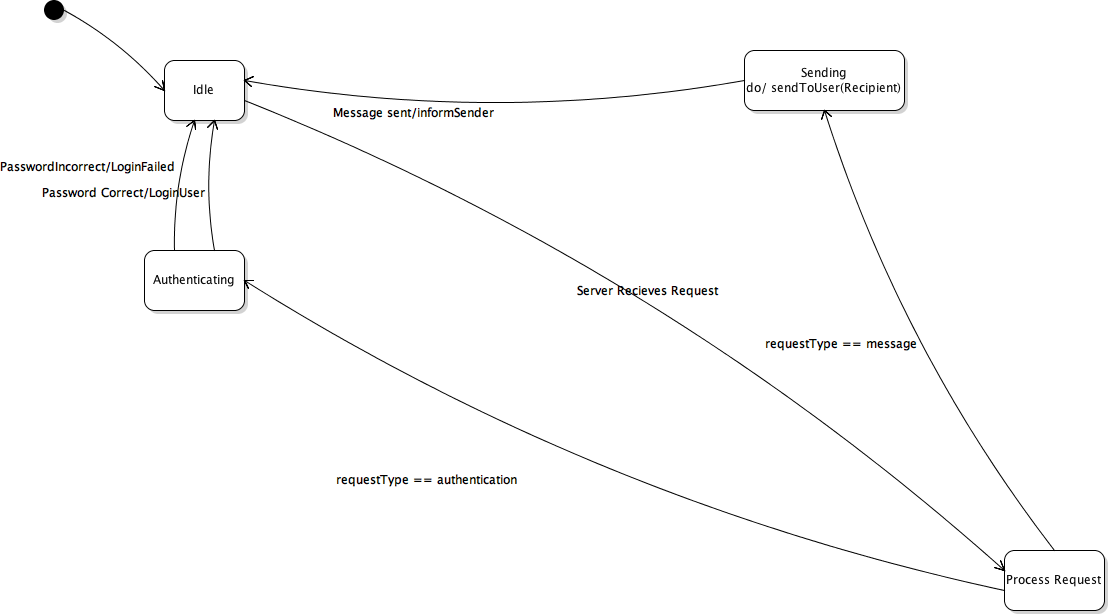


Figure 11: 4.4.2 UML State Machine Diagram: Server Request

Server Request State Machine Diagram Description: The server request state machine displays the states of the server on login and message. The app begins at idle and sends a request to the server, which processes the request. If the request is a message it sends it to receiver and notifies the sender of its status, and if the request is for authentication it checks for a match and responds to user with login if correct.

### 4.5 Prototype Interfaces

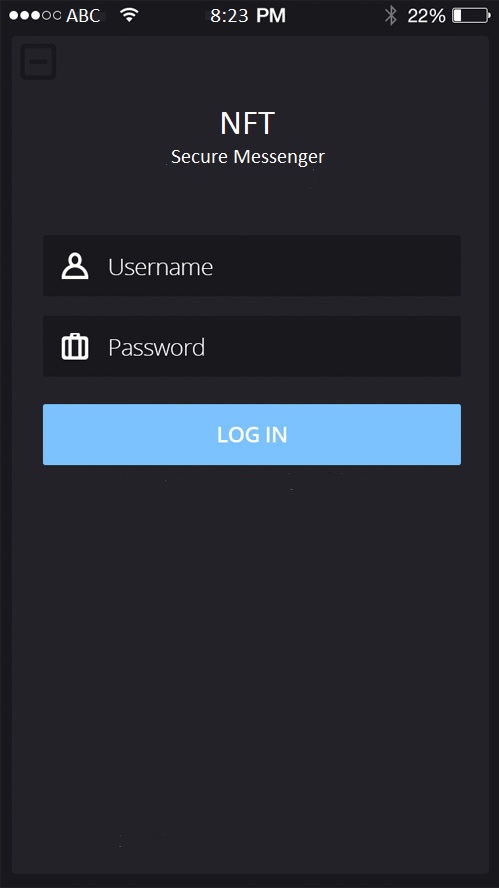


Figure 12: 4.5.1 App Login Screen

App Login Screen Description: Purpose of this page is to log in to the app. Contains a username text entry field widget, a password text entry field widget, and a confirm button labeled “Log In” which takes you to the contacts list page.

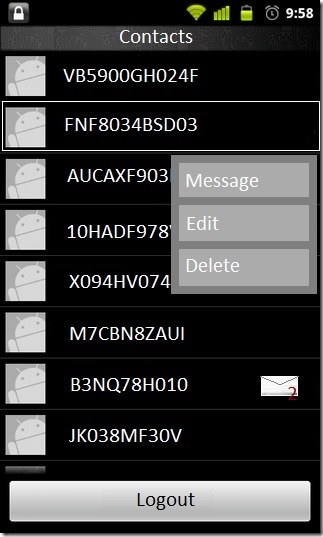


Figure 13: 4.5.2 App Contacts List

App Contacts List description: This page contains a list of contacts. Each contact name can be tapped to open a menu bar with options to message, edit the contact, or delete the contact. At the bottom there is a button to logout of the app.

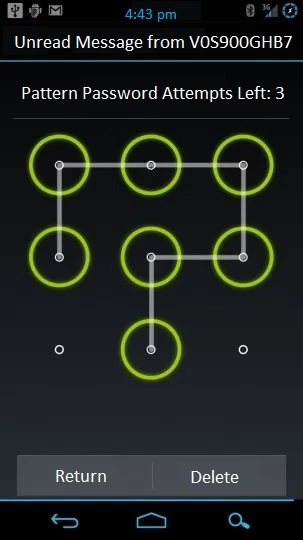


Figure 14: 4.5.3 App Message Unlock Pattern Screen

App Message Unlock Pattern screen: This screen allows you to enter a pattern or code, depending on the encryption methods in use. The pattern widget allows you to drag a pattern across the screen. The return button takes you back to the contacts screen. The Delete button deletes the message after a confirmation window pops up.

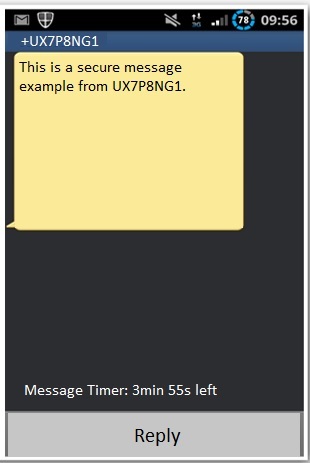


Figure 15: 4.5.4 App Read Message Screen

App Read Message Screen Description: This screen allows you to view the message after a successful un-encryption. The Reply button allows you to send a message back.

## Section 5

### 5.1 Scrum Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Story | Estimation | Priority |
| 1 | As an admin I want to create an Admin account | 2 | 1 |
| 9 | As an admin I want to view a list of accounts and awaiting accounts | 1 | 2 |
| 10 | As an admin I want to create a user account | 1 | 3 |
| 5 | As an admin I want to unlock an account | 1 | 11 |
| 6 | As an admin I want to delete an account | 1 | 12 |
| 2 | As an authorized user I want to log in | 1 | 4 |
| 7 | As an authorized user I want to log out | 1 | 5 |
| 4 | As an authorized user I want to add a contact | 1 | 7 |
| 3 | As an authorized user I want to send a message | 5 | 6 |
| 11 | As an authorized user I want to add a security measure | 2 | 9 |
| 8 | As an authorized user I want to open a message to read | 1 | 8 |
| 12 | As an authorized user I want to select a de-encryption type | 3 | 10 |
|  | Total | 20 |  |

Table 9: Scrum Backlog

### 5.2 Non-Functional Requirements

|  |  |
| --- | --- |
| 1 | Key for messages should not be kept, saved, or sent(sender and receiver should have set up the key previously outside of the app) |
| 2 | Only the Encrypted message should be sent |
| 3 | Pattern should be mapped to an encryption |
| 4 | Patter should be at least 10 points long |
| 5 | 3 chances to unlock |
| 6 | On 3rd failure- delete message |
| 7 | Send failure message to original sender (no security options) |
| 8 | If 2 encryption options are selected, the message should be encrypted 2 times |
| 9 | Messages should not be stored on any server or anywhere else |
| 10 | No back end if possible(otherwise only for use in transferring message) |
| 11 | User Must have Security clearance for admin to create account |
| 12 | Username must be 10-15 digits long |
| 13 | Password must be at least 16 characters long and contain at least 1 lowercase letter, 1 uppercase letter, 1 number, and 1 special character, and should be unique |
| 14 | Only the encrypted version of a password should be kept on the server |

Table 10: Non-Functional Requirements

Appendix

Meeting with Client

Date: Wednesday, March 8, 2017, 5:20 p.m.

Length: 35 minutes

Information Elicited:

The company we are making the app for deals with private messages which contain sensitive information. We are free to explore which encryption scheme we will employ. The keys would be persistent. A server for keeping track of locations will be easiest. Concern over screenshots is there, but low priority.

On failed login attempts, the account will lock requiring an admin to unlock it. The max time for a message to remain unopened on the device will be 24 hours. Layout and design is up to us. Admin accounts should be able to create, delete, lock, and unlock accounts. Tracker status may be a possibility to display things like the delivered, read, sent statuses.

The app notifications should be regular push notifications. The app will have a normal icon. Usernames need not be randomized. Optional contacts list and group messaging.