

**Bachelor of Computer Science**

**(Digital System Security)**

CSCI321 – Final Year Project

SOFTWARE REQUIREMENT

SPECIFICATION

**Project Particulars**

|  |  |
| --- | --- |
| Supervisor | Dr Ta Nguyen Binh Duong |
| Project Group | SS18/1F |
| Project Title | Two Factor Authentication |

**Project Team’s Particulars**

|  |  |  |
| --- | --- | --- |
| Student Number | Student Name | Email Address |
| 5363536 | Koh Hong Wei | hwkoh003@mymail.sim.edu.sg |
| 5710923 | Chua Han Ming Adler | hmachua002@mymail.sim.edu.sg |
| 5711356 | Ong Wei Hao | whong012@mymail.sim.edu.sg |

****

**A secure file locking application**

**Document Control**

Title: Project Proposal

Document Name: FYP\_ProjectProposal

**Distribution List**

|  |  |  |
| --- | --- | --- |
| Name | Title/Role | Where |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Record of Revision**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Revision Date | Description | Section Affected | Changes Made By | Version after Revision | Notes |
| 16 March | 1st Draft of SRS | All | Team |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

[1. Introduction 7](#_Toc508982098)

[1.1. Background 7](#_Toc508982099)

[1.2. Project Scope 7](#_Toc508982100)

[1.3 Intended Audience and Reading Suggestions 7](#_Toc508982101)

[2. Product Overview Description 7](#_Toc508982102)

[2.1 Product Perspective 7](#_Toc508982103)

[2.2 Product Features 7](#_Toc508982104)

[2.2.1 Locking File 7](#_Toc508982105)

[2.2.2 Unlocking file 7](#_Toc508982106)

[2.2.3 Secure OTP generation 7](#_Toc508982107)

[2.2.4 View File 8](#_Toc508982108)

[2.2.5 View History 8](#_Toc508982109)

[2.3 User Classes and Characteristics 8](#_Toc508982110)

[2.3.1 Physical Actors 8](#_Toc508982111)

[2.3.2 System Actors 8](#_Toc508982112)

[2.4 Operating Environment 8](#_Toc508982113)

[2.5 Implementation Constraints 8](#_Toc508982114)

[2.6 User Documentation 8](#_Toc508982115)

[2.7 Assumptions and Dependencies 8](#_Toc508982116)

[3. System Features 8](#_Toc508982117)

[Main Use Case 9](#_Toc508982118)

[1.1 Create Account 9](#_Toc508982119)

[Use Case 9](#_Toc508982120)

[Activity Diagram 10](#_Toc508982121)

[Sequence Diagram 10](#_Toc508982122)

[1.2 Account Recovery 10](#_Toc508982123)

[Use Case 10](#_Toc508982124)

[Activity Diagram 11](#_Toc508982125)

[Sequence Diagram 12](#_Toc508982126)

[1.3 Login 12](#_Toc508982127)

[Use Case 12](#_Toc508982128)

[Activity Diagram 13](#_Toc508982129)

[Sequence Diagram 14](#_Toc508982130)

[1.4 Change Password 14](#_Toc508982131)

[Use Case 14](#_Toc508982132)

[Activity Diagram 15](#_Toc508982133)

[Sequence Diagram 15](#_Toc508982134)

[1.5 Lock 15](#_Toc508982135)

[Use Case 15](#_Toc508982136)

[Activity Diagram 16](#_Toc508982137)

[Sequence Diagram 17](#_Toc508982138)

[1.6 Unlock 17](#_Toc508982139)

[Use Case 17](#_Toc508982140)

[Activity Diagram 18](#_Toc508982141)

[Sequence Diagram 19](#_Toc508982142)

[1.7 View File 19](#_Toc508982143)

[Use Case 19](#_Toc508982144)

[Activity Diagram 20](#_Toc508982145)

[Sequence Diagram 21](#_Toc508982146)

[1.8 View History 21](#_Toc508982147)

[Use Case 21](#_Toc508982148)

[Activity Diagram 22](#_Toc508982149)

[Sequence Diagram 23](#_Toc508982150)

[2.1 Delete User 23](#_Toc508982151)

[Use Case 23](#_Toc508982152)

[Activity Diagram 24](#_Toc508982153)

[Sequence Diagram 24](#_Toc508982154)

[2.2 Account Recovery 25](#_Toc508982155)

[Use Case 25](#_Toc508982156)

[Activity Diagram 25](#_Toc508982157)

[Sequence Diagram 26](#_Toc508982158)

[3.1 Login 27](#_Toc508982159)

[Use Case 27](#_Toc508982160)

[Activity Diagram 27](#_Toc508982161)

[Sequence Diagram 28](#_Toc508982162)

[3.2 Lock/ Unlock 29](#_Toc508982163)

[Use Case 29](#_Toc508982164)

[Activity Diagram 29](#_Toc508982165)

[Sequence Diagram 30](#_Toc508982166)

[3.3 Change Password 31](#_Toc508982167)

[Use Case 31](#_Toc508982168)

[Activity Diagram 32](#_Toc508982169)

[Sequence Diagram 32](#_Toc508982170)

[4. External Interface Requirements 33](#_Toc508982171)

[4.1 Hardware Interfaces 33](#_Toc508982172)

[4.2 Software Interfaces 33](#_Toc508982173)

[5. Non-functional Requirements 34](#_Toc508982174)

[5.1 Performance Requirements 34](#_Toc508982175)

[5.2 Safety Requirements 34](#_Toc508982176)

[5.3 Security Requirements 34](#_Toc508982177)

[5.4 Software Quality Attributes 34](#_Toc508982178)

[References 34](#_Toc508982179)

[Appendix A: Glossary 34](#_Toc508982180)

# 1. Introduction

## 1.1. Background

The purpose of this document is to provide a detailed overview of our software product. This includes the description of our project’s target audience, system features, user interface and both hardware and software requirements.

## 1.2. Project Scope

This project aims to provide users with a way to enhance security on file management on desktop applications. We will be including a two factor authentication mechanism to secure our files from unauthorised usage.

## 1.3 Intended Audience and Reading Suggestions

# 2. Product Overview Description

## 2.1 Product Perspective

Go2FA is an application running on both Windows(PC) and Mobile(Android) platform written in JAVA. It is intended for providing a secure method of protecting a user’s file or folder in their local PC with our Two Factor Authentication. The system uses a secured cryptographic algorithm to identify a user action within the application. This system uses what the user has and what the user knows as a way of identification. User has a username and password with will be created upon installing the application on either PC or mobile. User has mobile phone with them all the time, so we implemented QRCode scanner capability in our application to further add into the security of the application function.

## 2.2 Product Features

### 2.2.1 Locking File

The user has the option to lock file located in their local PC with our cryptographic algorithm.

### 2.2.2 Unlocking file

The user has the option to unlock file located in their local PC with our cryptographic algorithm.

### 2.2.3 Secure OTP generation

The system provides QRCode upon the locking or unlocking action done by User, which will need the camera functionality of the user’s mobile by using the application on the device. The QRCode will then issue an OTP after scanning which can be entered to recognise legitimacy of user.

### 2.2.4 View File

The user has the option to view files that are locked using our application just by simply entering the date which the file has been locked or the file name.

### 2.2.5 View History

The user has the option to view past login history and actions that he has done using our application just by simply entering the date he wants to identify.

## 2.3 User Classes and Characteristics

### 2.3.1 Physical Actors

**Window/Mobile Users:** The user who uses the system and make use of the services provided by the application.

### 2.3.2 System Actors

#### 2.3.2.1 SQL Database

## 2.4 Operating Environment

This system operates in both Windows and Android Operating System.

## 2.5 Implementation Constraints

The main constraint of this program is the support for files and folders in local directories and files locked can only be unlocked by the same PC with contains the directory.

## 2.6 User Documentation

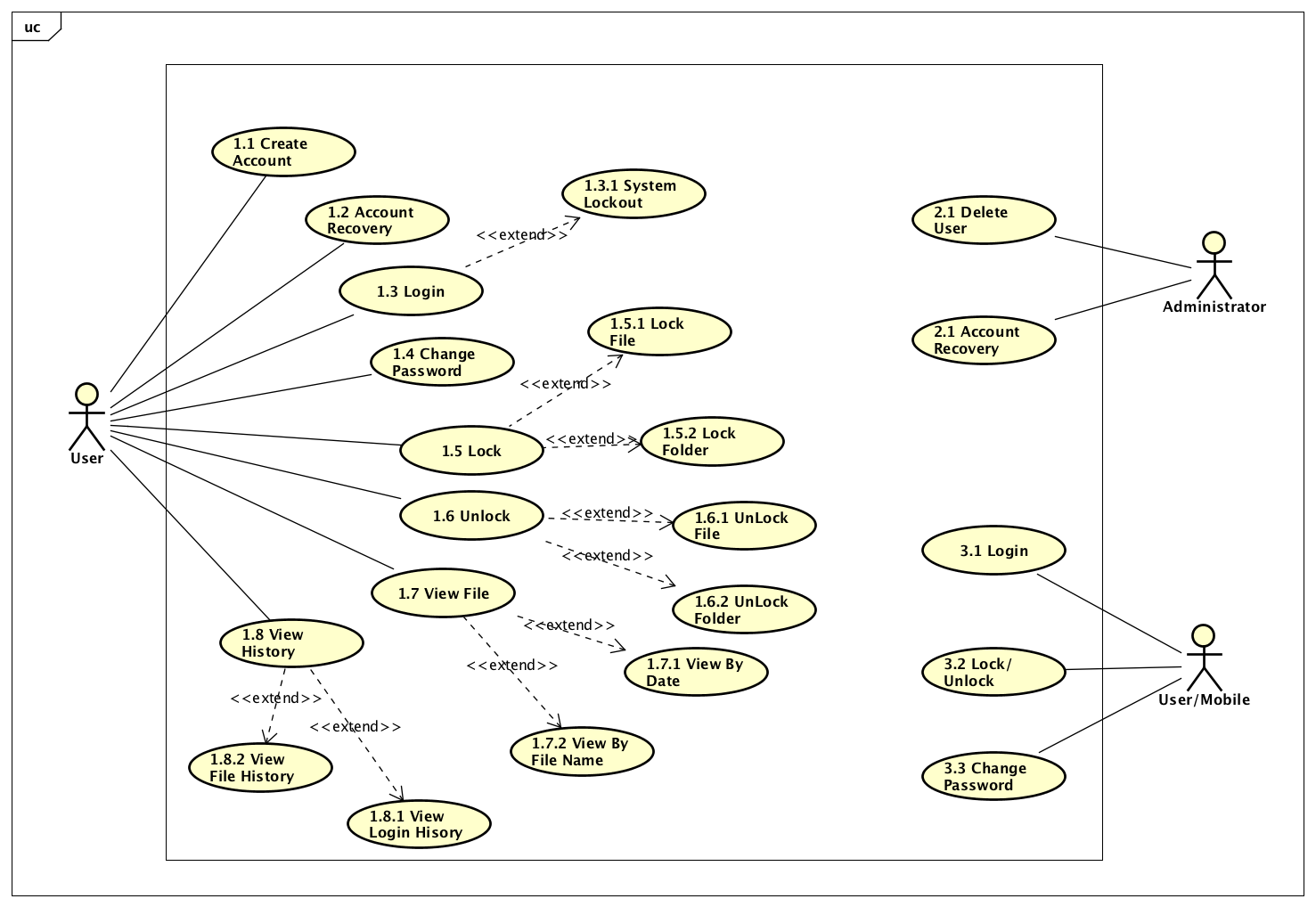
The user can use the User Manual to understand more of the interfaces of the system.

## 2.7 Assumptions and Dependencies

* The user is liable for keeping his log in credentials secured and confidential.
* The user is liable for the safekeeping of his copies of document outside of our system secured.
* The user is liable for the security and breach of his local PC.

# 3. System Features

## Main Use Case

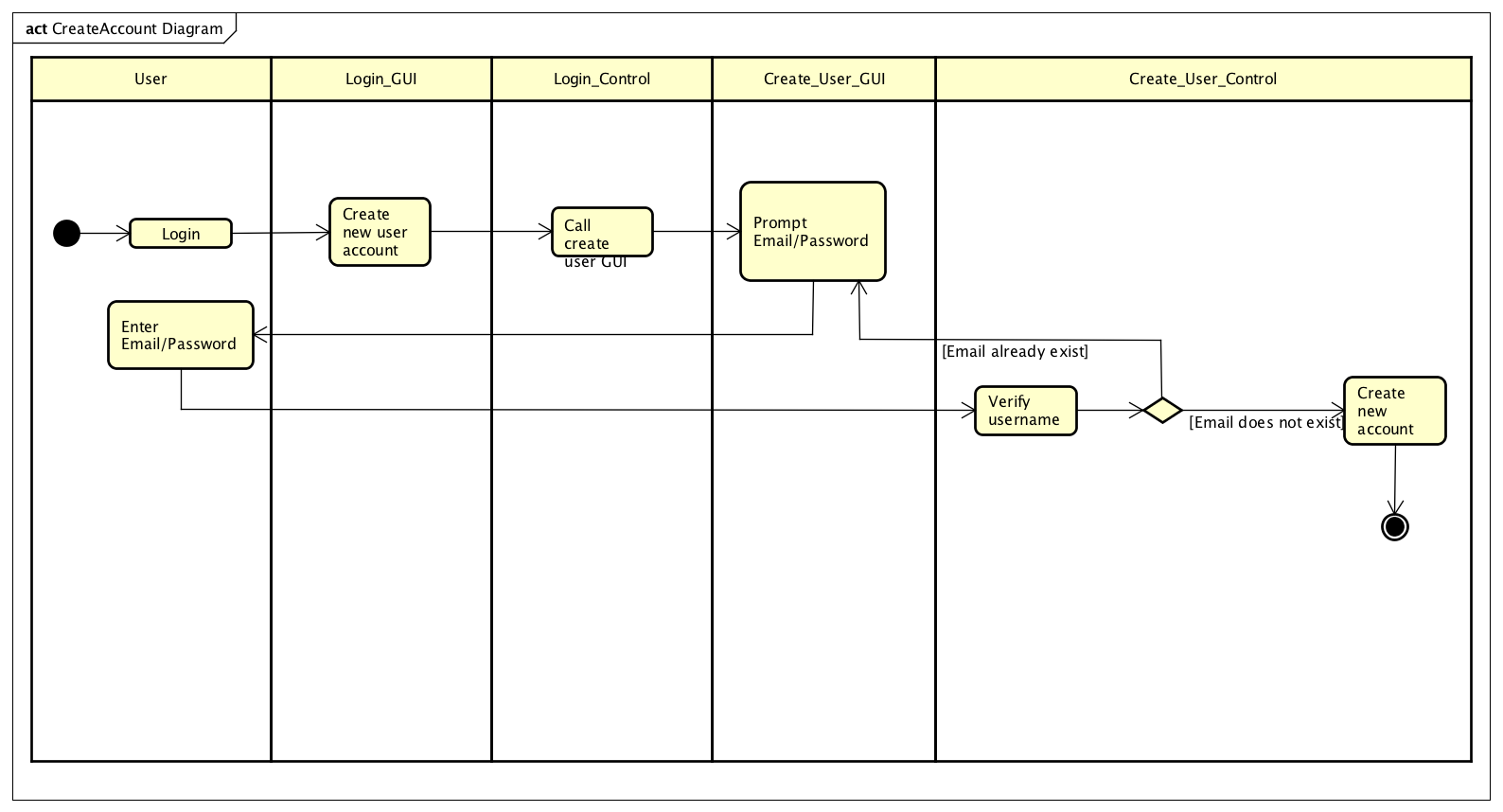


## Create Account

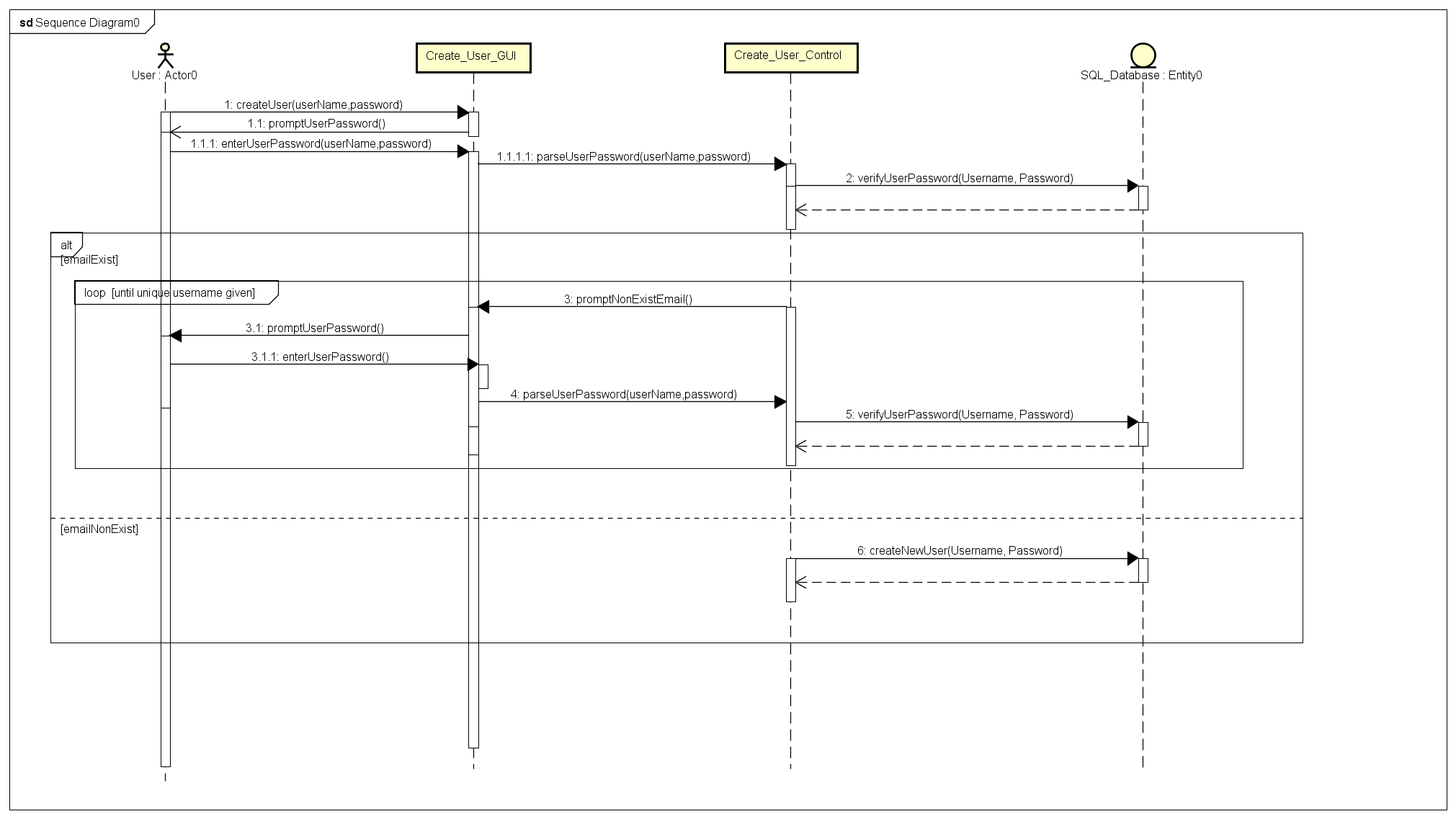
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.1 |
| Name | Create Account |
| Description | To allow user to create new account after he download the application for the first time |
| Actor(s) | User |
| Precondition | A user has installed the application and wish to create a new account |
| Main Scenario | Step 1: System prints out a sign-up form with fields to ask for user email address and desired password  Step 2: User submits form and system does validation  Step 2(alternate): If data is invalid, an error message will be displayed, and user is sent back to form for re-entry of right credentials  Step 3: User password will be hashed  Step 4: System updates the SQL Database  Step 5: User is redirected to log in screen |

### Activity Diagram



### Sequence Diagram

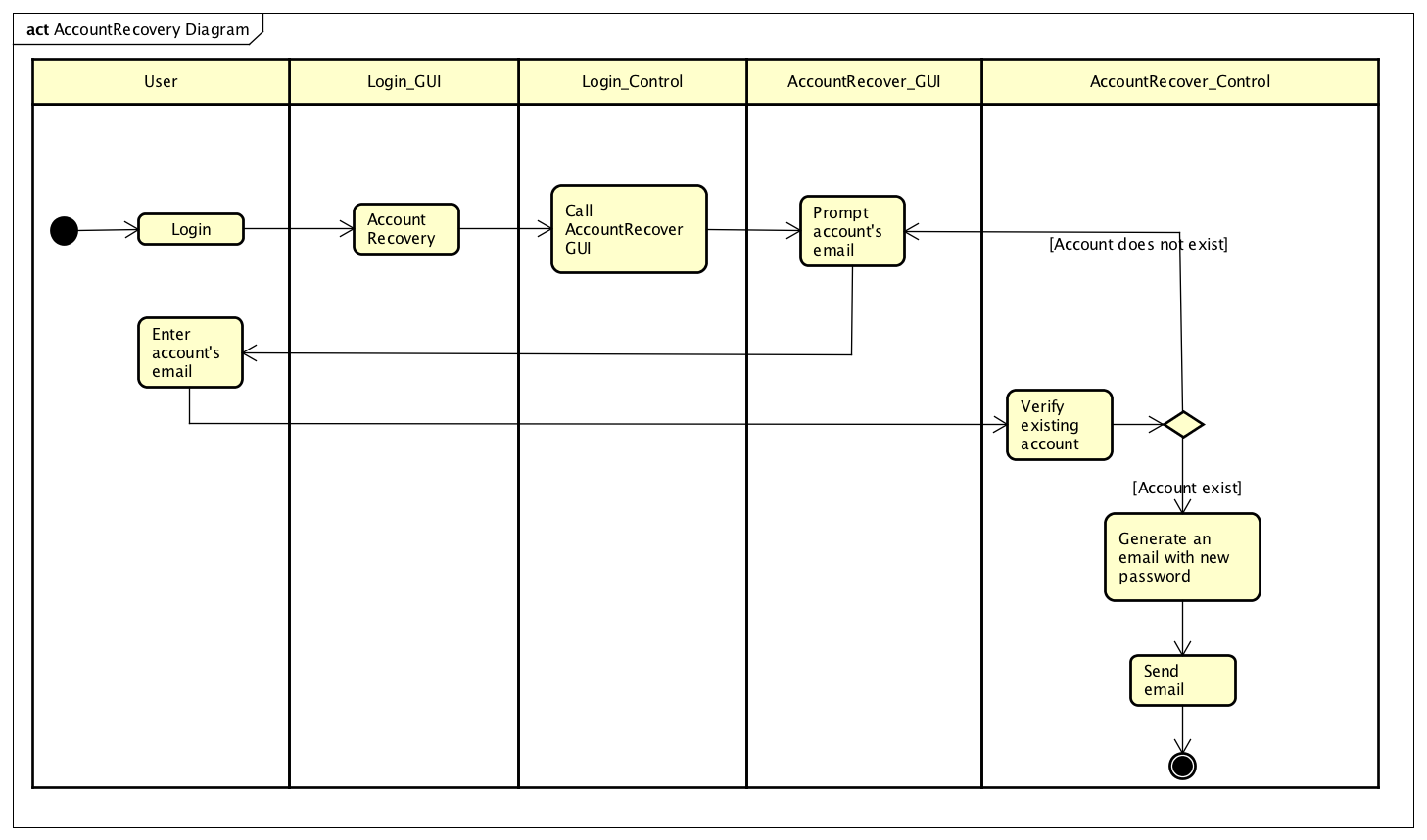


## Account Recovery

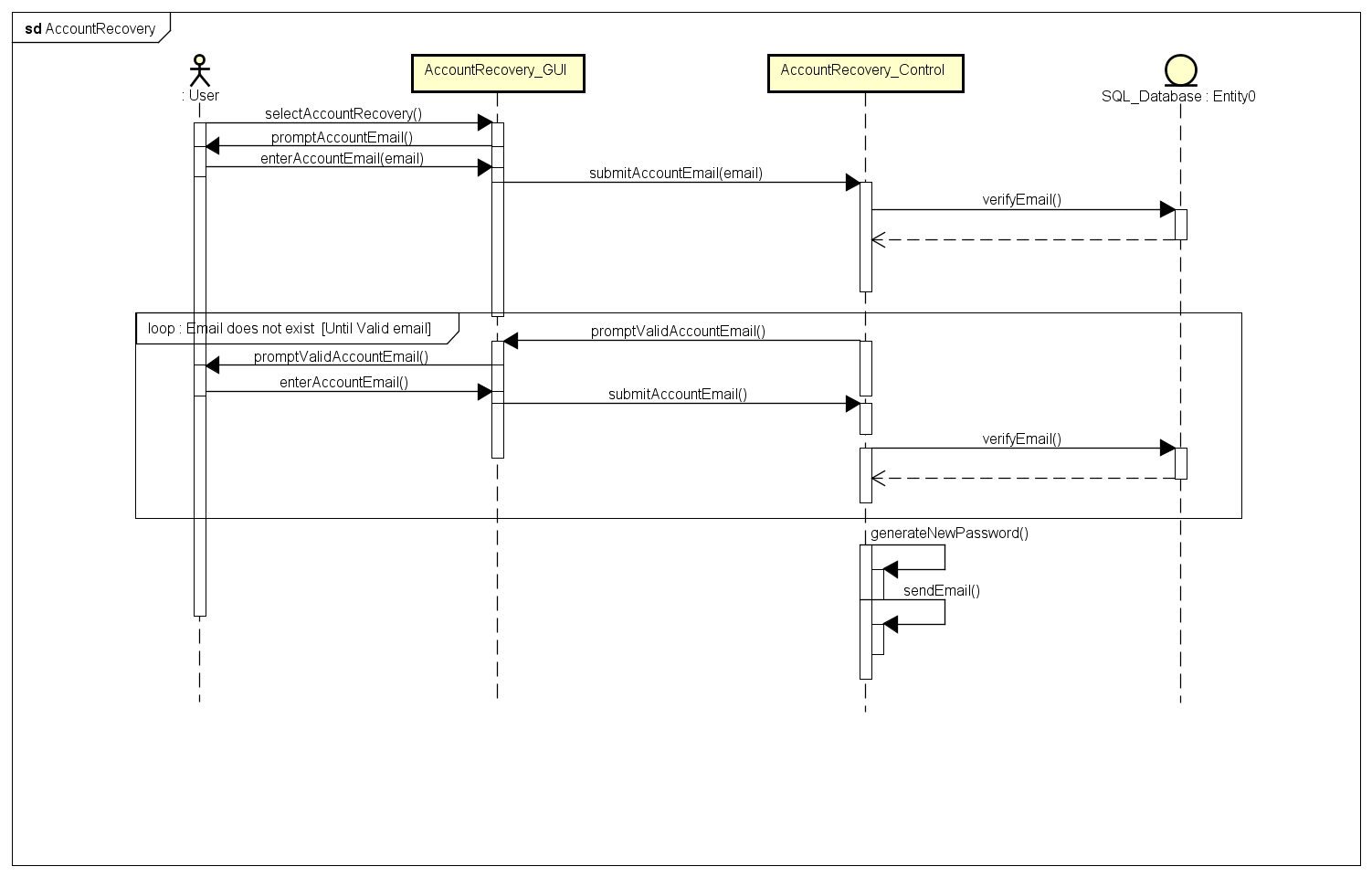
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.2 |
| Name | Account Recovery |
| Description | To allow user to recover their lost or forgotten account |
| Actor(s) | User |
| Precondition | The user has a registered account and forgotten password |
| Main Scenario | Step 1: User selects account recovery on Log in screen  Step 2: System prints out a form to prompt for User account email  Step 3: User fills up the email and system do validation  Step 3(alternate): If email is invalid, an error message will be displayed, and user is sent back to form for re-entry of right credentials  Step 4: System will generate new password  Step 5: Email with new generated password will be sent to User email address. |

### Activity Diagram



### Sequence Diagram



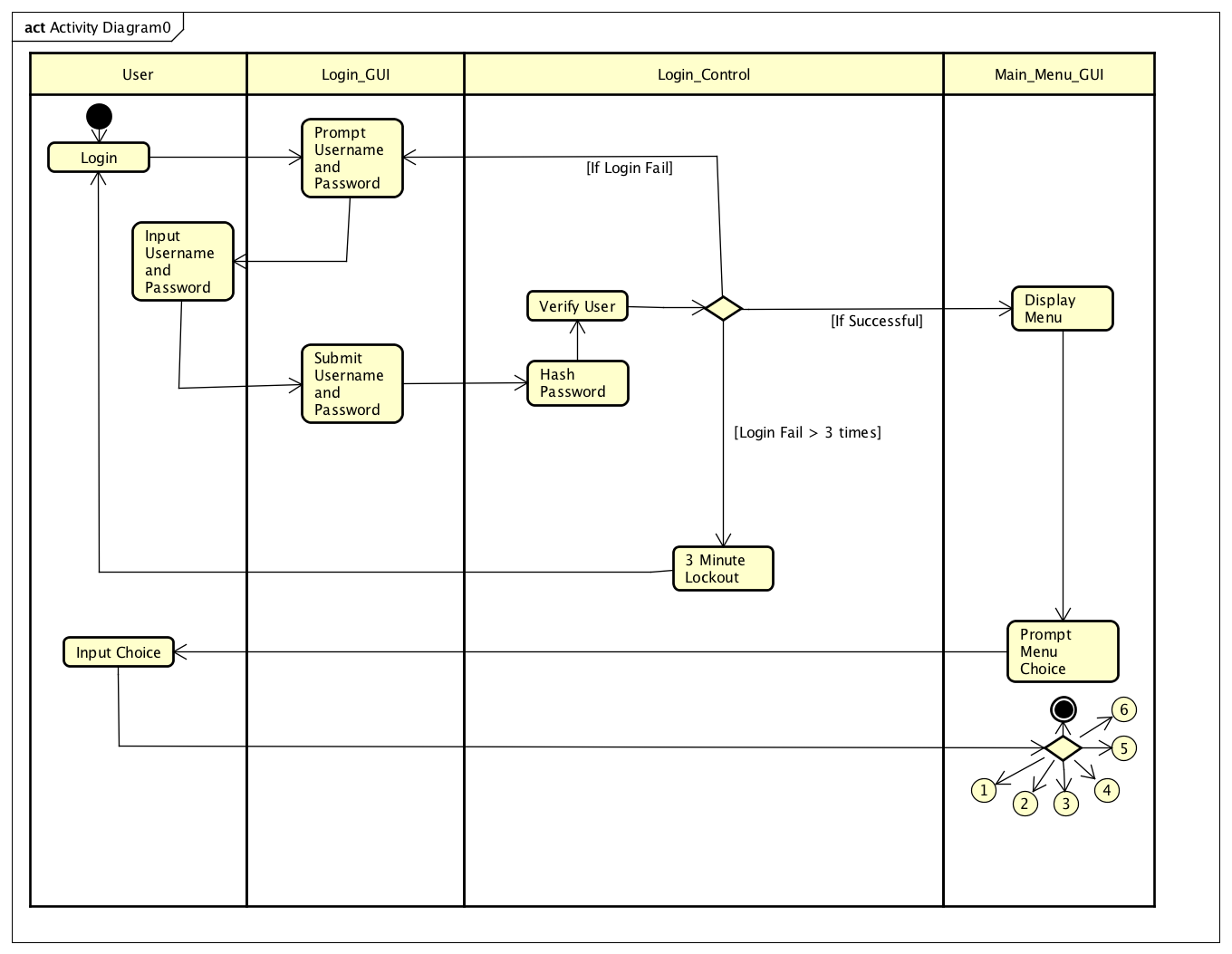
## Login

### Use Case

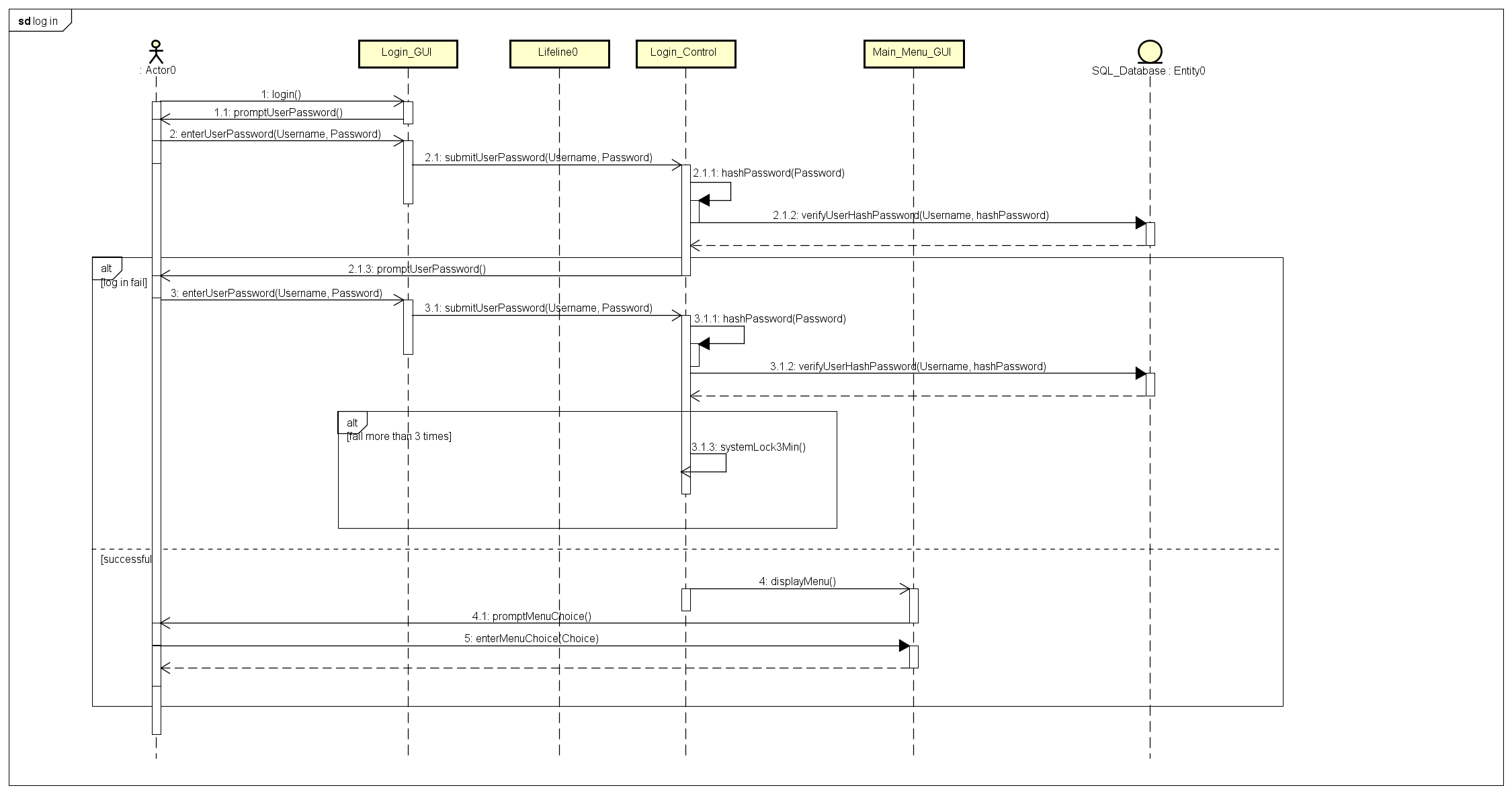
|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.3 |
| Name | Login |
| Description | To allow User to log in to the system |
| Actor(s) | User |
| Precondition | The user has a registered account |
| Main Scenario | Step 1: User enters their log in credentials in the log in screen  Step 2: System hash the User password and validates with SQL Database  Step 2(alternate): If log in credentials are wrong, an error message will display, and User will be prompt to enter credentials again  Step 2(alternate): If log in credentials is wrong for 3 times, system will be lock from User for 3 minutes **via U/C 1.3.1**  Step 3: System will display Main Menu  Step 4: User will enter choice from Main Menu |

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.3.1 |
| Name | System Lockout |
| Description | Lock User out after several failed log in attempts |
| Actor(s) | User |
| Precondition | The user has 3 failed log in attempts |
| Main Scenario | Step 1: System will lock the user out of the system for 3 minutes  Step 2: System will resume after 3 minutes. |

### Activity Diagram



### Sequence Diagram

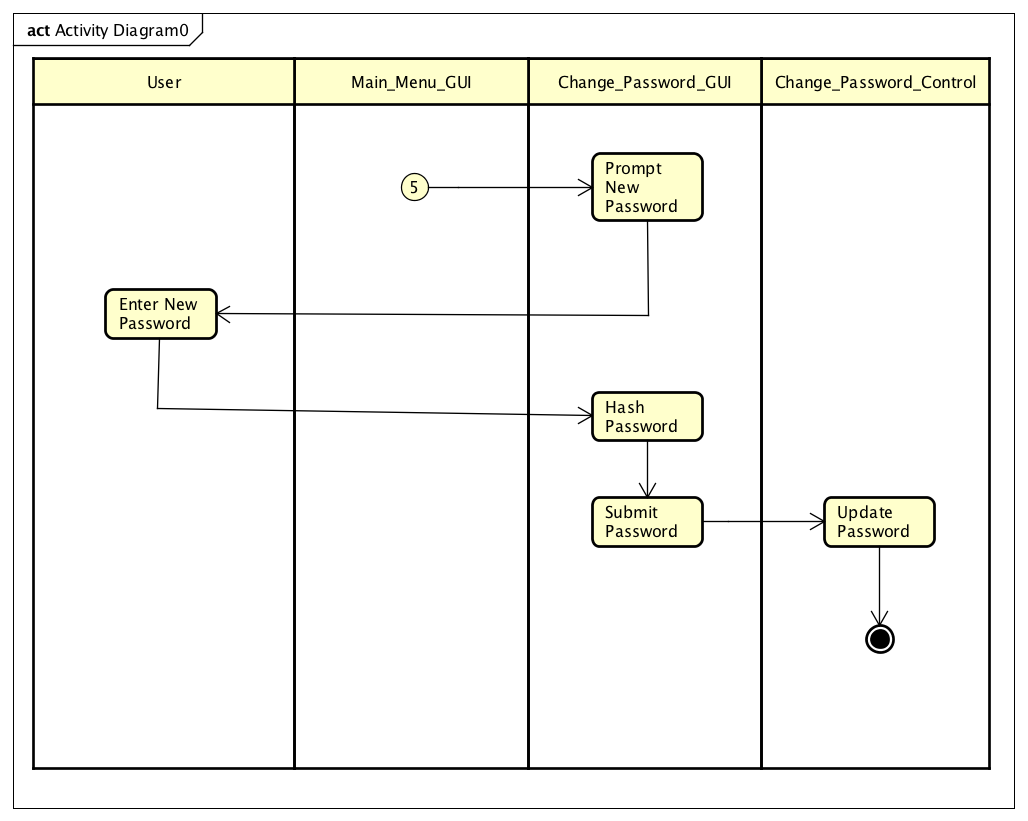


## Change Password

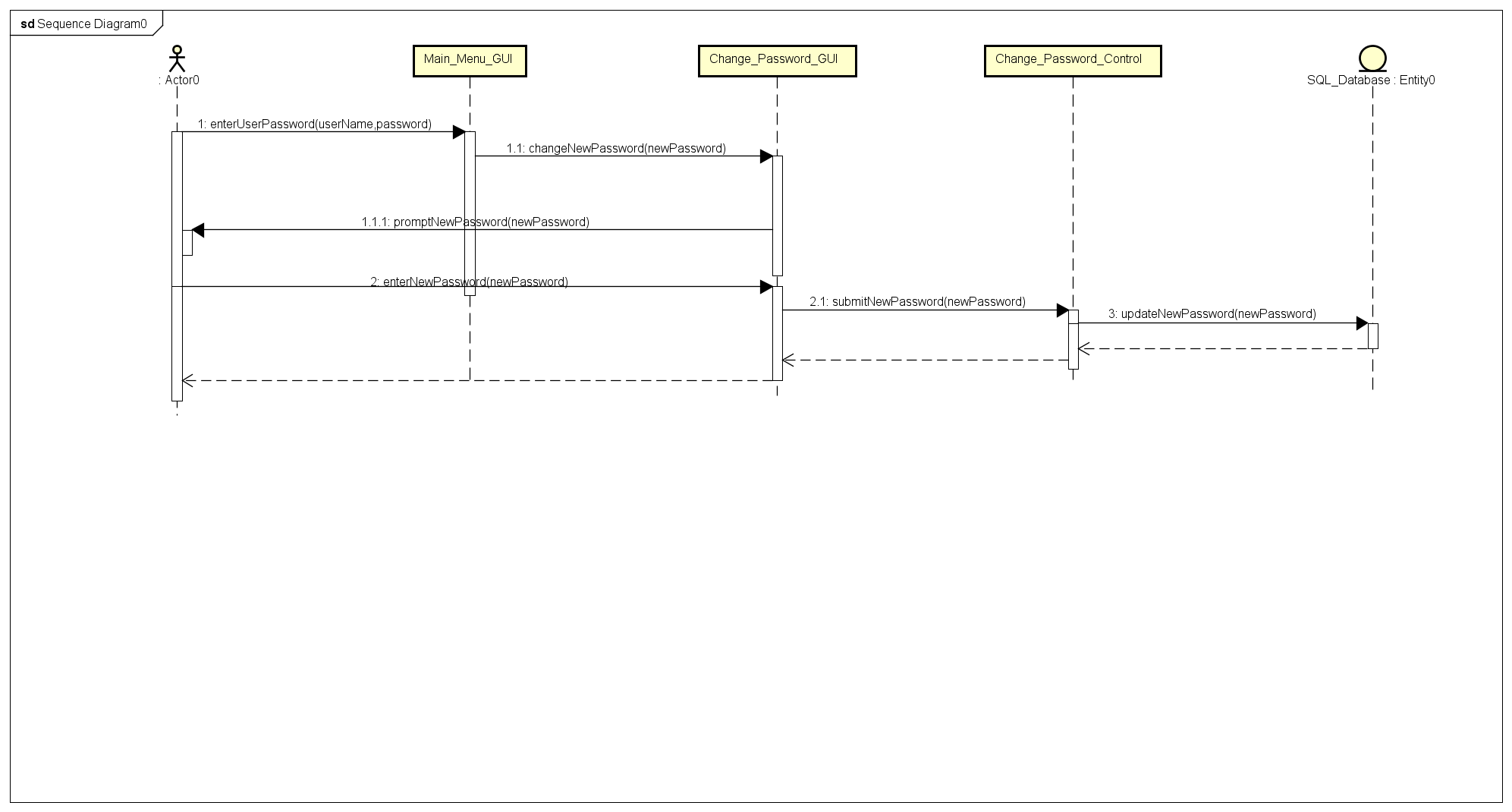
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.4 |
| Name | Change Password |
| Description | To allow User to change their password |
| Actor(s) | User |
| Precondition | The user has a registered account and wish to update the current password |
| Main Scenario | Step 1: User enters username and password at Log in screen  Step 2: User selects change password at the main menu  Step 3: System displays a form for user to fill up the new password they desire  Step 4: User enters the new desired password and system do validation  Step 5: System will update the SQL Database with the new password |

### Activity Diagram



### Sequence Diagram

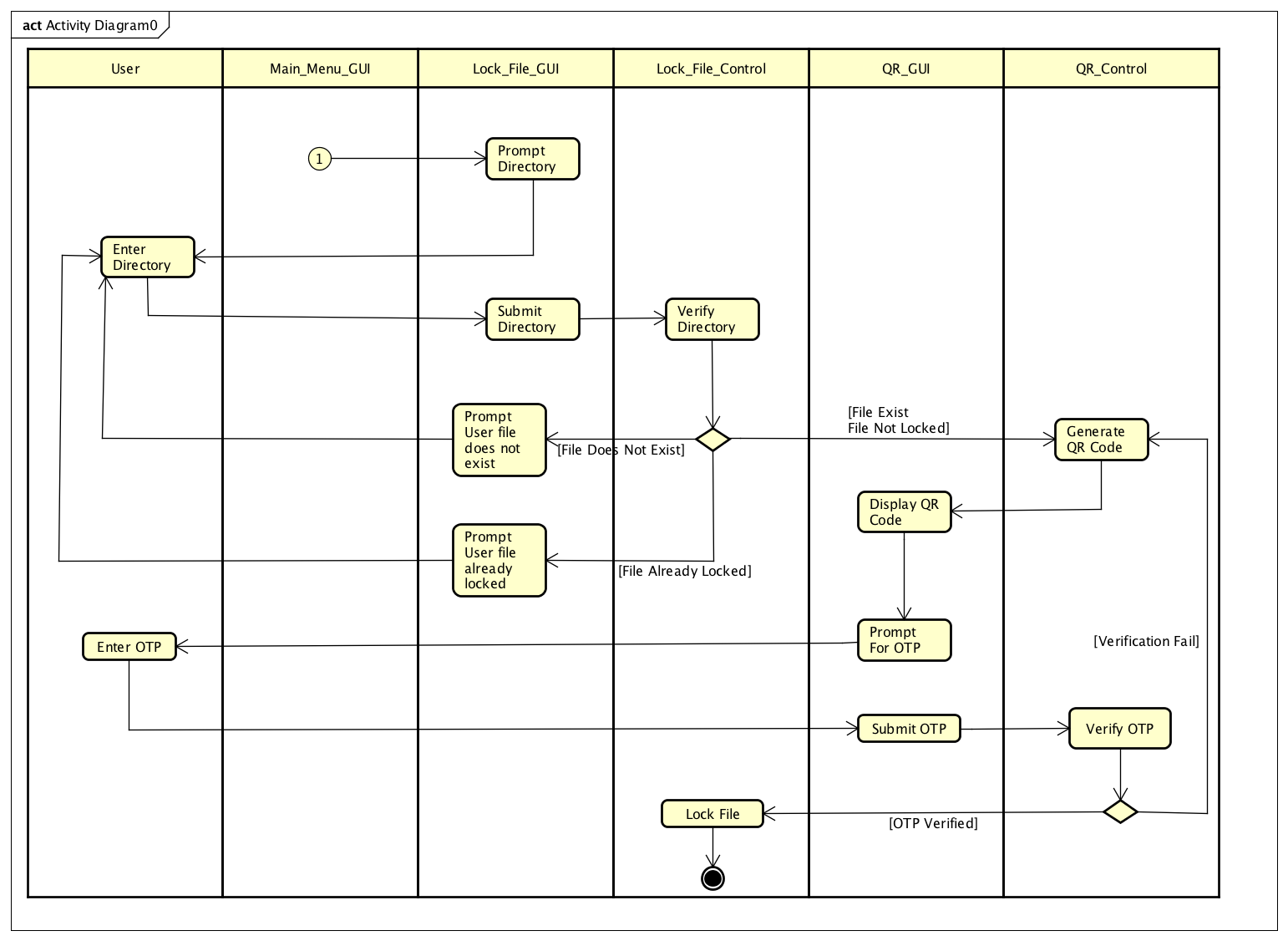


## Lock

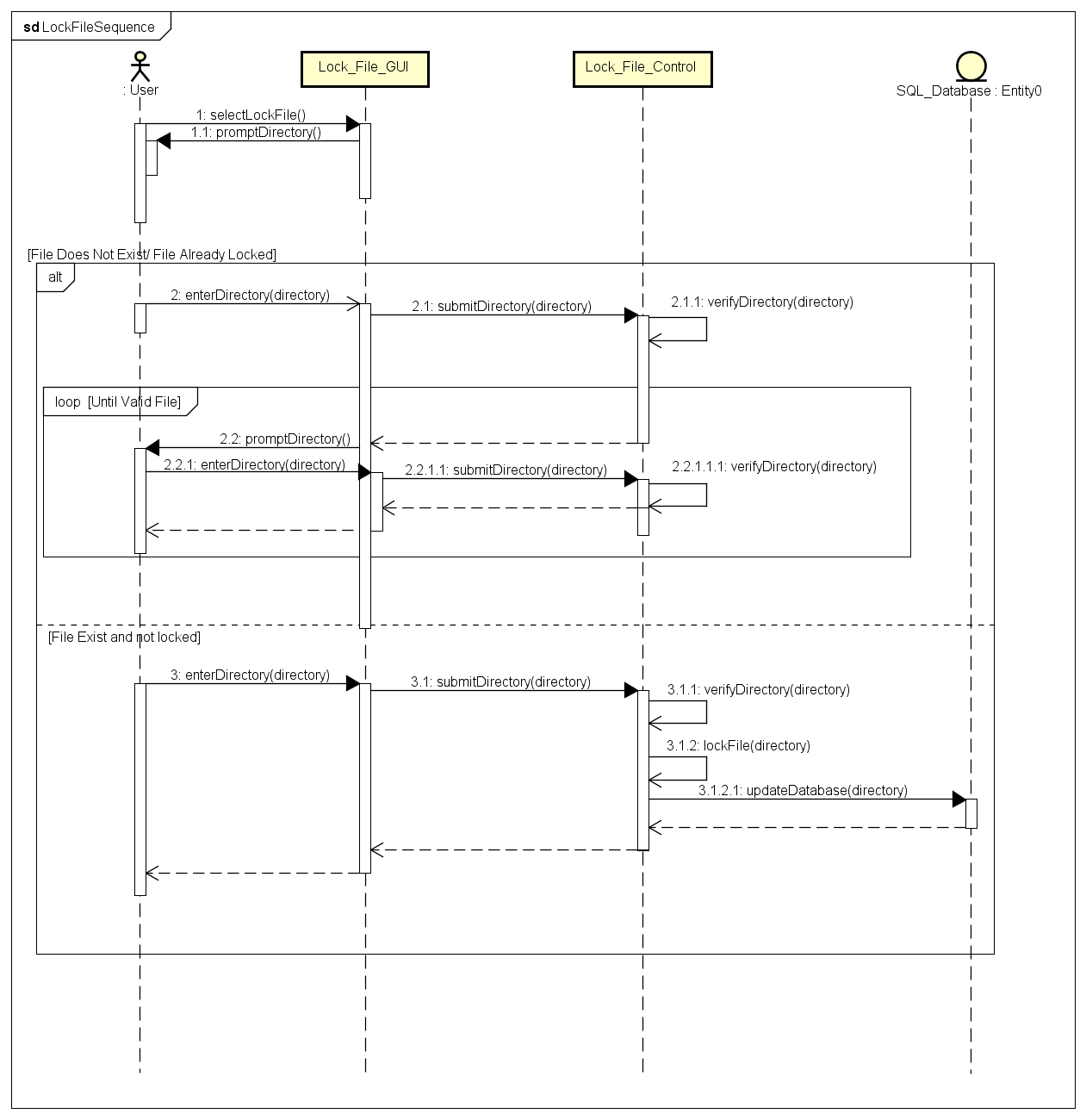
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.5 |
| Name | Lock |
| Description | To allow user to lock their folders or files using system cryptography feature |
| Actor(s) | User |
| Precondition | The user has a registered account and has a folder or file that they want to lock |
| Main Scenario | Step 1: User selects Lock file option in the Main Menu.  Step 2: System displays a form to fill up which directory the User has the folder or file  Step 3: User enters the directory and system validates  Step 3(alternate): If directory is invalid, an error message will display, and User will be redirected to form to fill up directory.  Step 3(alternate): If the file or folder in directory is already locked, an error message will display, and User will be redirected to form to fill up the correct directory  Step 4: The system locks the file or folder of the User directory with cryptography feature |

### Activity Diagram



### Sequence Diagram

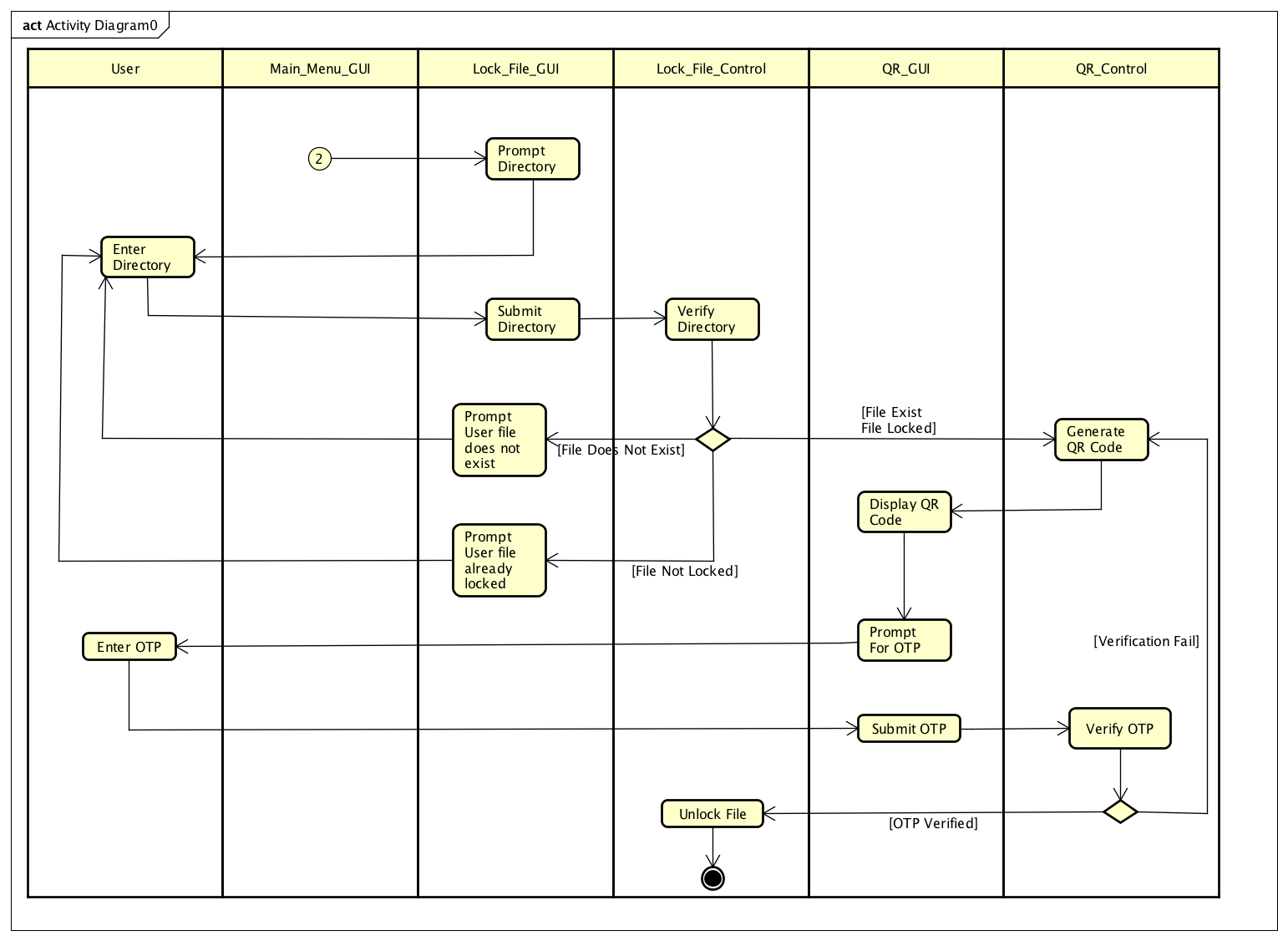


## Unlock

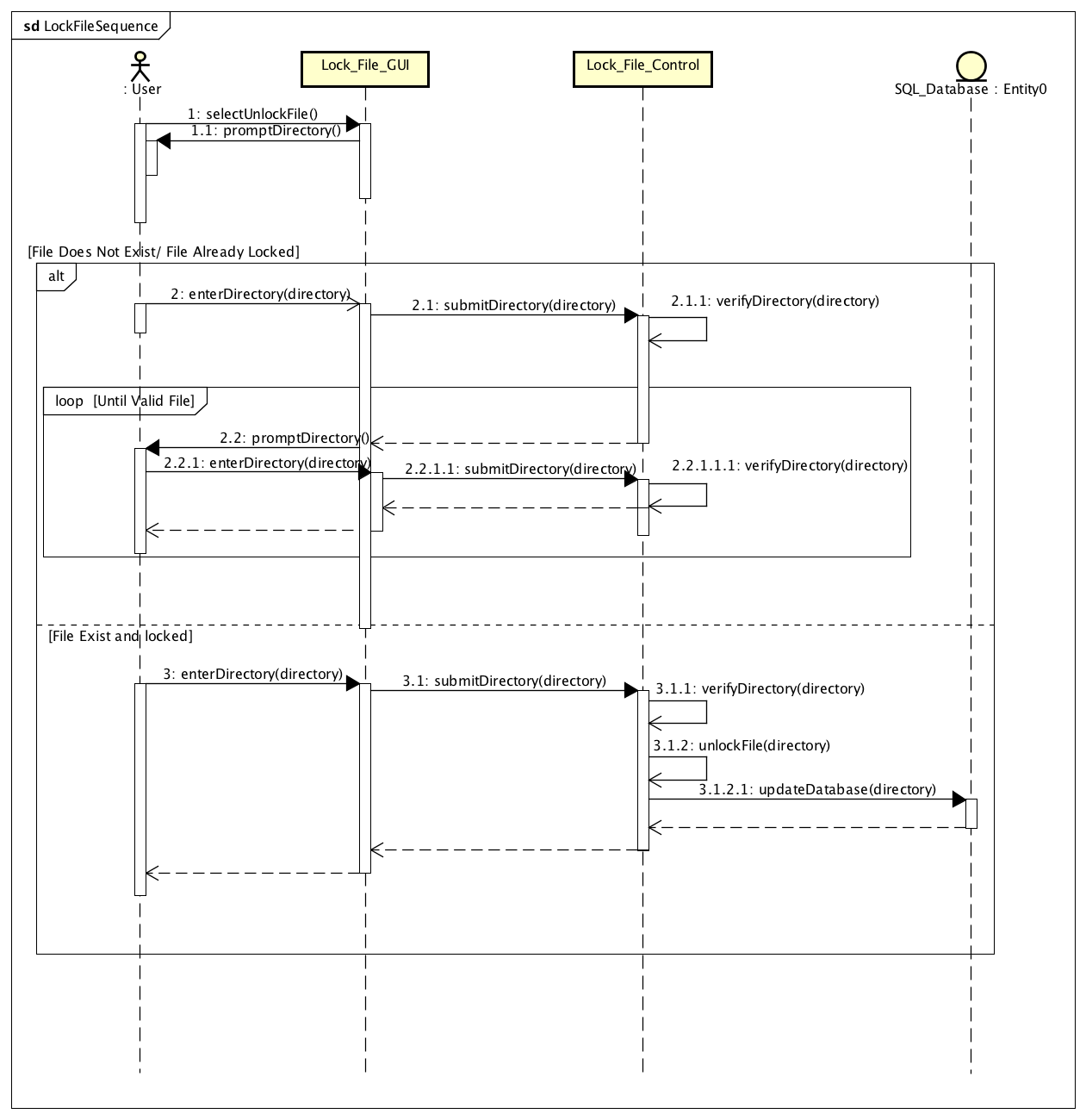
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.6 |
| Name | Unlock |
| Description | To allow user to unlock their folders or files using system cryptography feature |
| Actor(s) | User |
| Precondition | The user has a registered account and logged into the system |
| Main Scenario | Step 1: User selects Unlock file option in the Main Menu.  Step 2: System displays a form to fill up which directory the User has the folder or file  Step 3: User enters the directory and system validates  Step 3(alternate): If directory is invalid, an error message will display, and User will be redirected to form to fill up directory.  Step 3(alternate): If the file or folder in directory is not locked, an error message will display, and User will be redirected to form to fill up the correct directory  Step 4: The system unlocks the file or folder of the User directory with cryptography feature |

### Activity Diagram



### Sequence Diagram

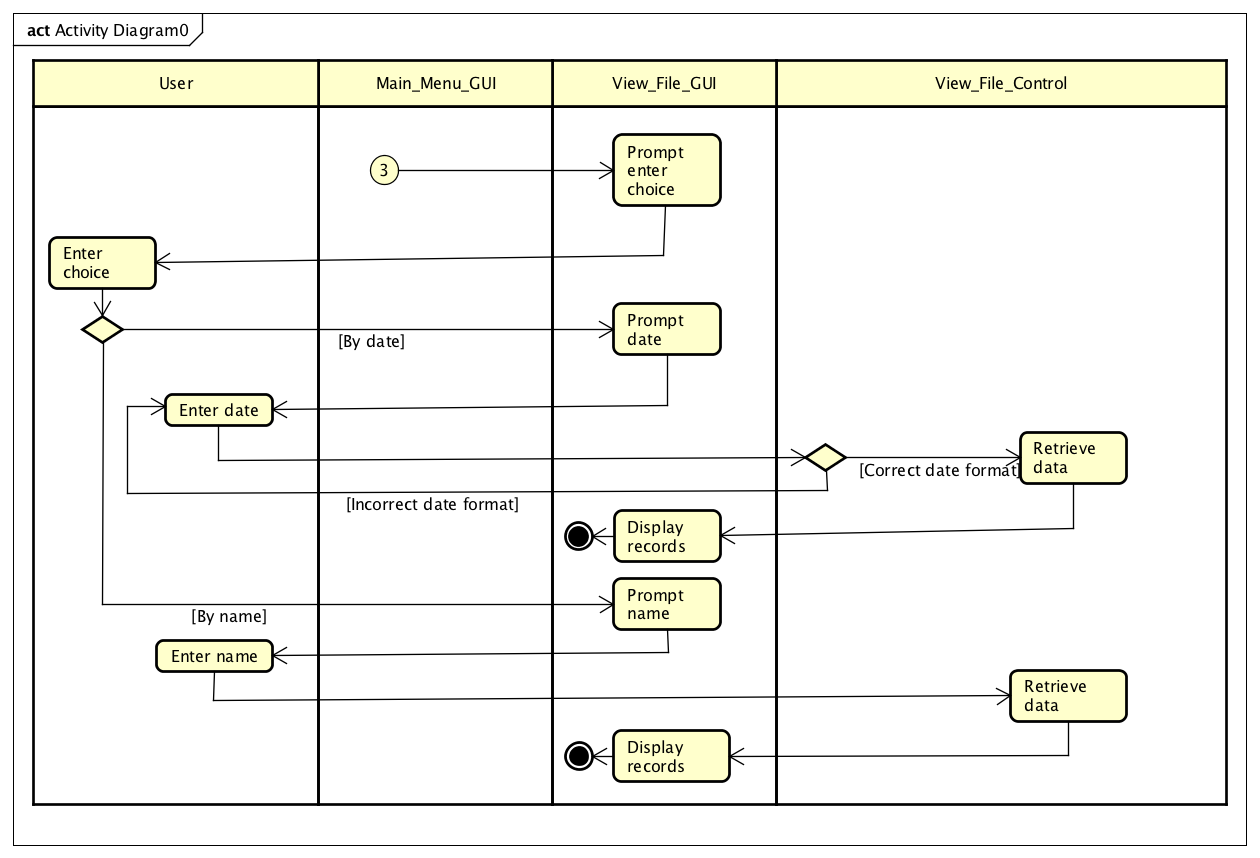


## View File

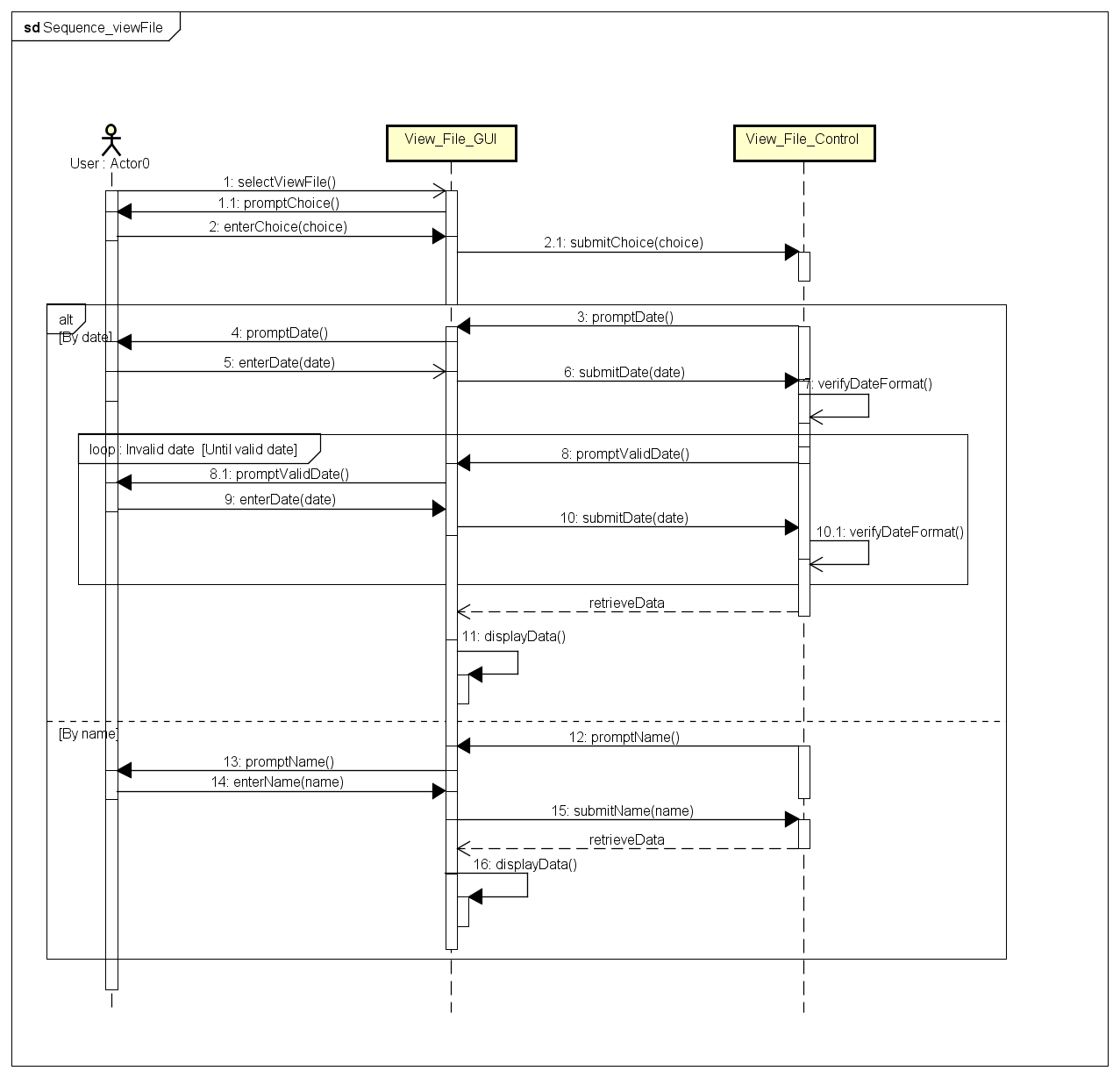
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.7 |
| Name | View File |
| Description | To allow user to view the locked files in the system |
| Actor(s) | User |
| Precondition | The user has a registered account and logged into the system |
| Main Scenario | Step 1: User selects View File option in the Main Menu  Step 2: User selects a choice in the view file menu  Step 3: System will display a prompt to User to choose from viewbydate or viewbyname  Step 4(viewbydate): User enters a date from where he locked a file and system validates **via U/C 1.7.1**  Step 4(viewbyname): User enters a date from where he locked a file and system validates **via U/C 1.7.2**  Step 4(alternate): if the date/name the User entered does not contain any locked file, the system will display an error message and prompt the User to enter a valid date  Step 5: When date/name is validated, system will retrieve the data and display the data to the User. |

### Activity Diagram



### Sequence Diagram

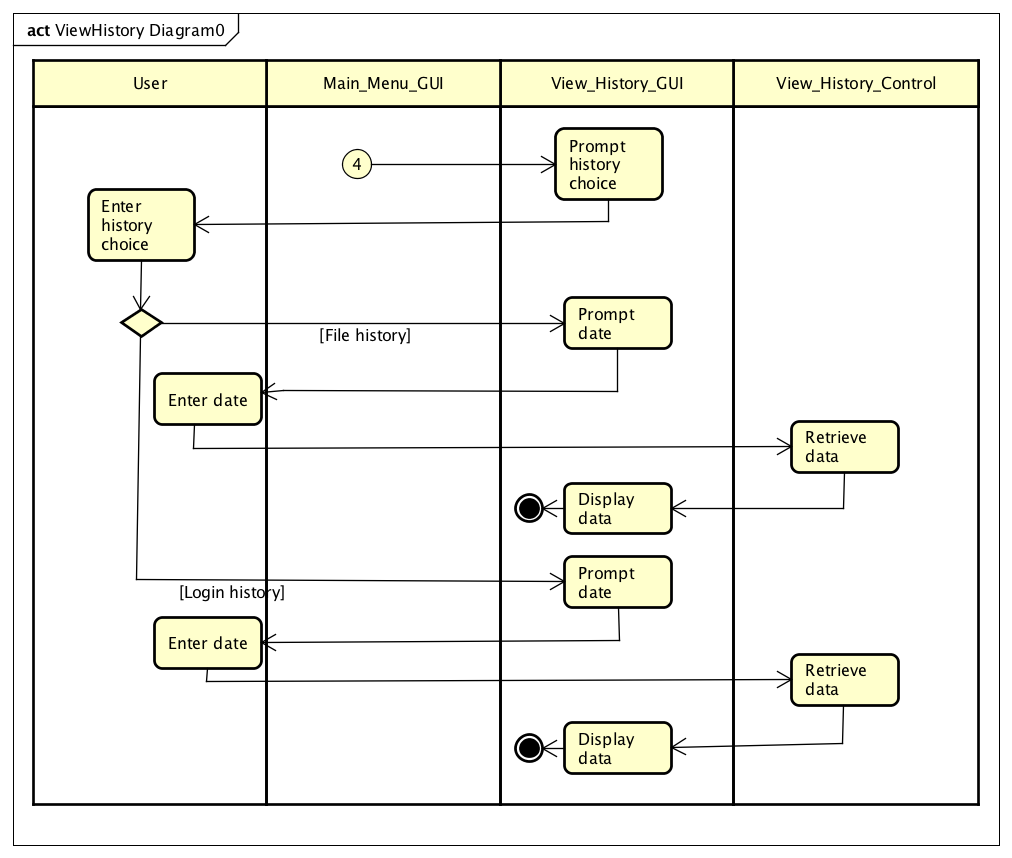


## View History

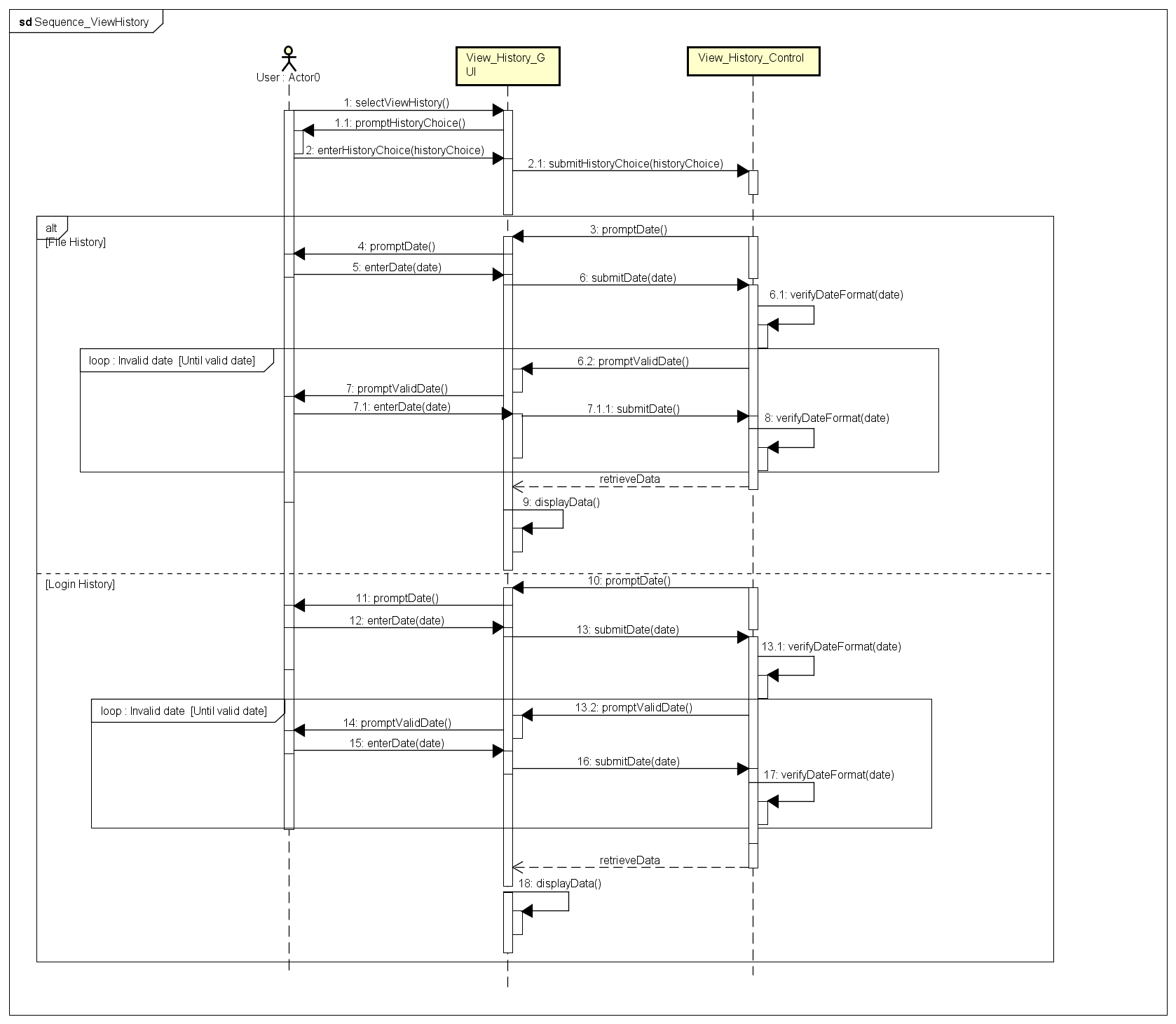
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 1.8 |
| Name | View History |
| Description | To allow user to view history of log in and file |
| Actor(s) | User |
| Precondition | The user has a registered account and logged into the system |
| Main Scenario | Step 1: User selects View History option in the Main Menu  Step 2: User selects a choice in the view history menu  Step 3: System will display a prompt to User to choose from viewfilehistory or viewloginhistory  Step 4(viewfilehistory): User enters a date from where he locked a file **via U/C 1.8.2**  Step 4(viewloginhistory): User enters a date from where he logged in **via U/C 1.8.1**  Step 5: System will retrieve the data and display the data to the User from the date User input. |

### Activity Diagram



### Sequence Diagram

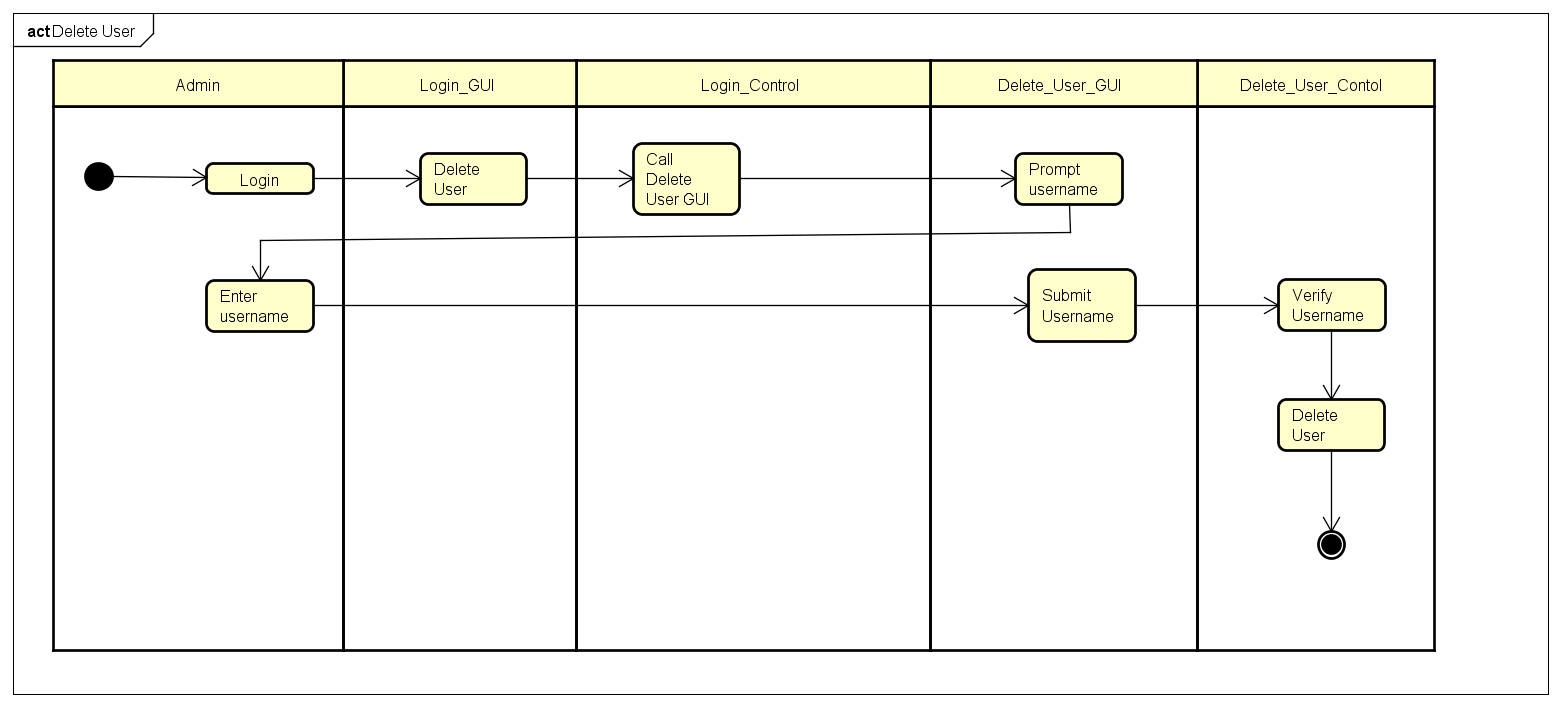


## 2.1 Delete User

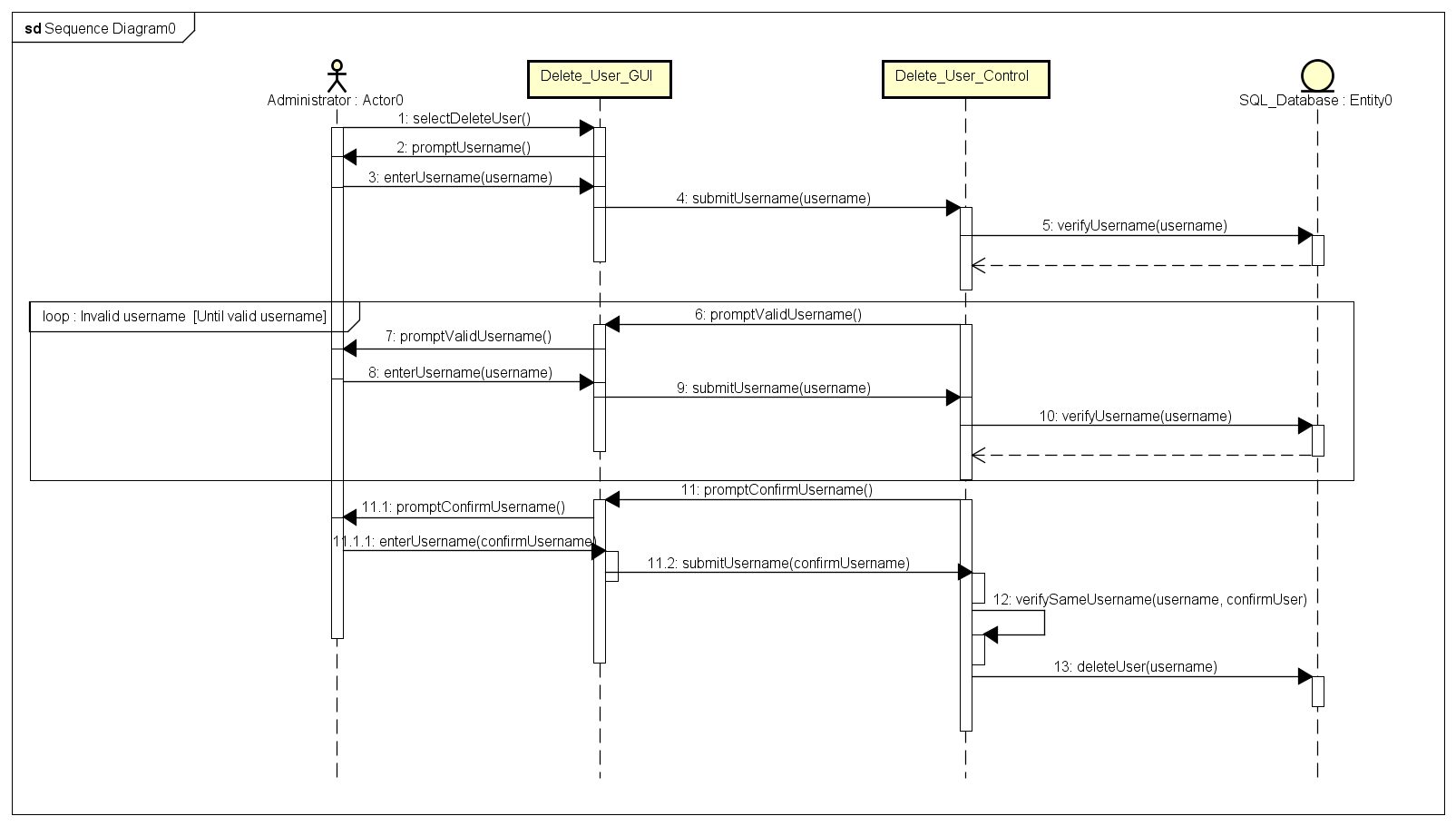
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 2.1 |
| Name | Delete User |
| Description | To allow the administrator to delete users from database |
| Actor(s) | Administrator |
| Precondition | The administrator wishes to delete off unused accounts |
| Main Scenario | Step 1: Administrator selects delete user on Main Menu  Step 2: System prompts to enter username of the account that is to be deleted  Step 3: Administrator fills up form and system validate  Step 3(alternate): If username is invalid, an error message will be displayed, and Administrator is sent back to form for re-entry of right credentials  Step 4: System will prompt to confirm username  Step 5: Administrator re-enter username of the account to be deleted  Step 6: System validates  Step 7: System deletes username from database |

### Activity Diagram



### Sequence Diagram

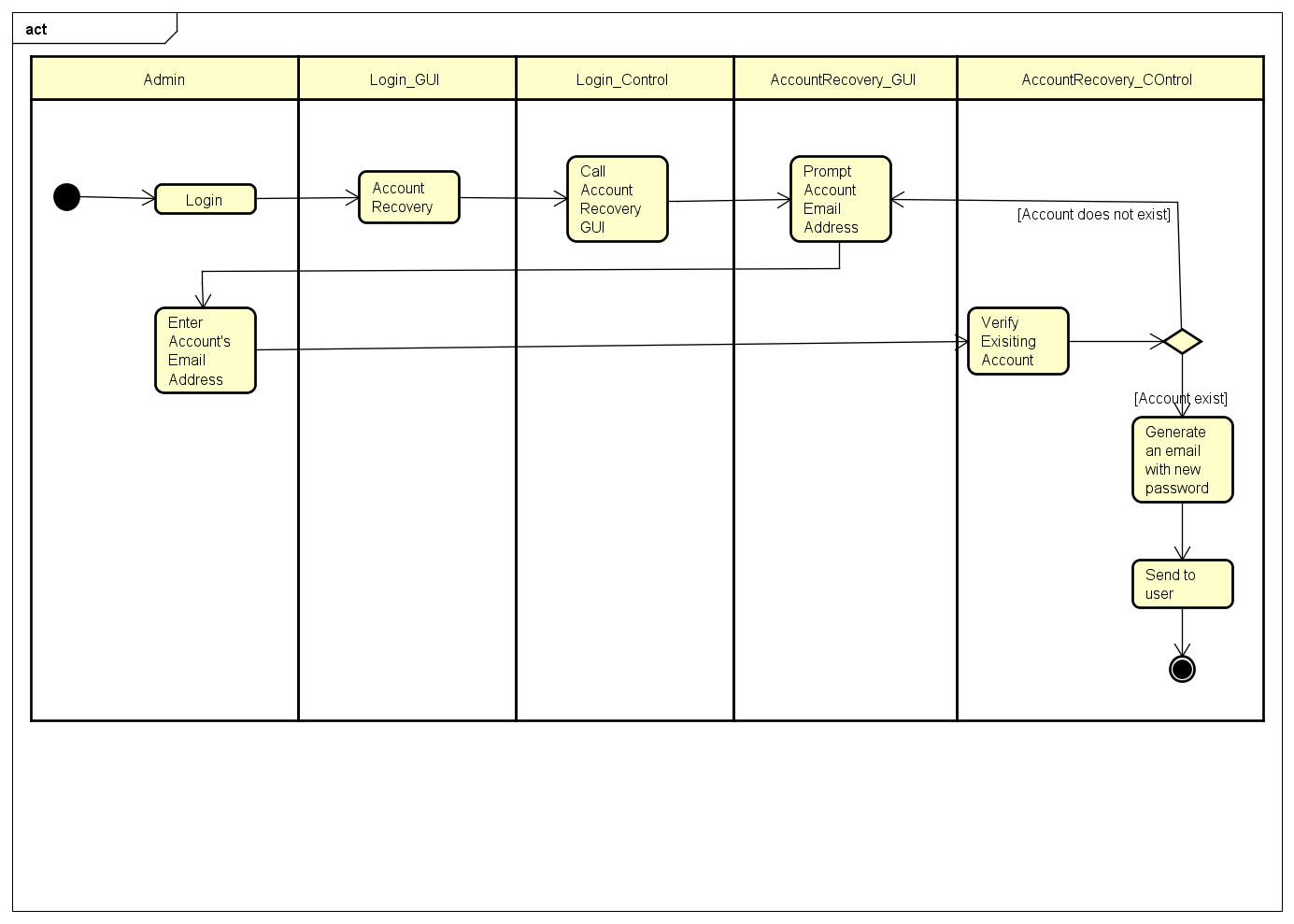


## 2.2 Account Recovery

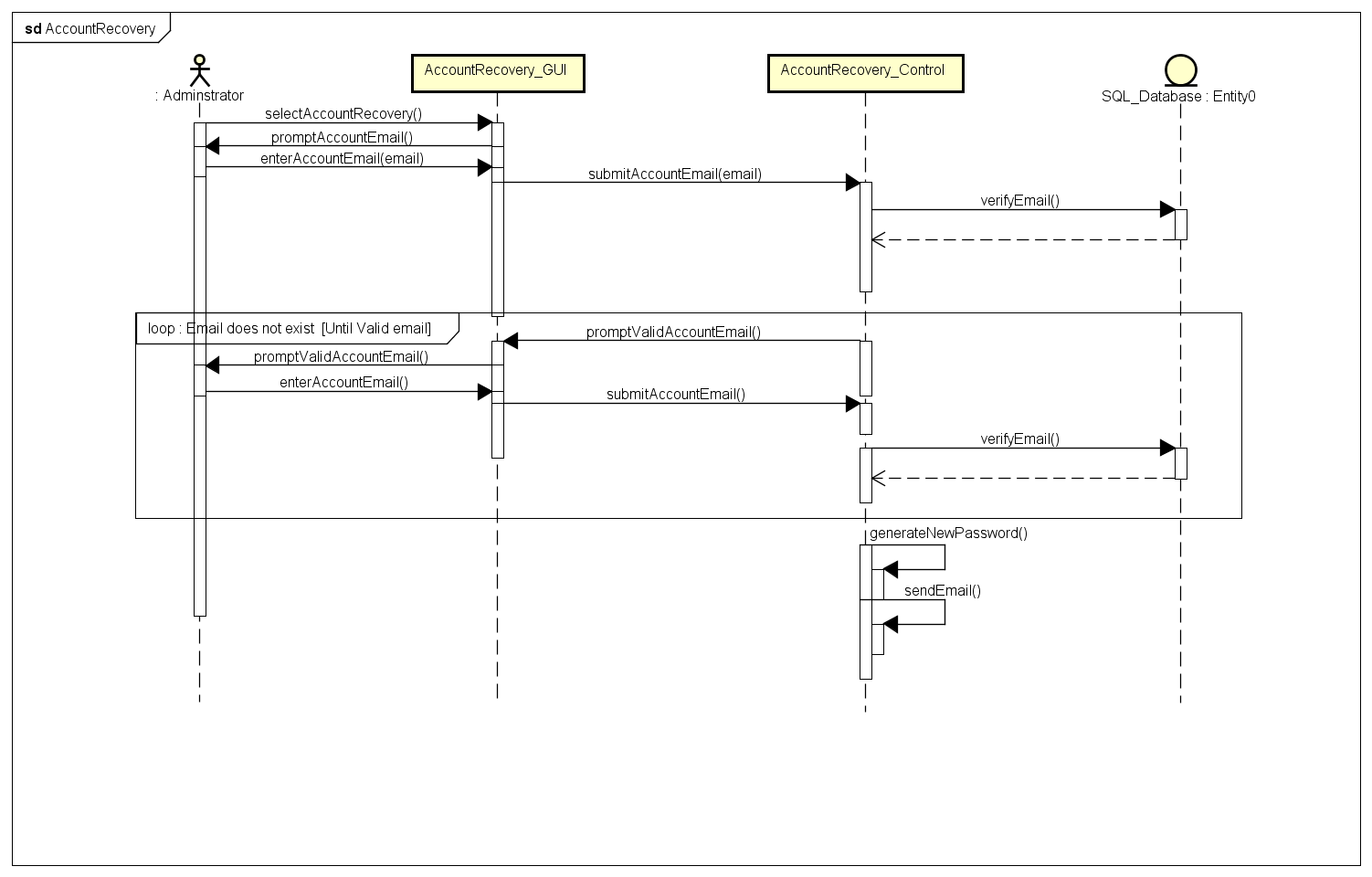
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 2.2 |
| Name | Account Recovery |
| Description | To allow Administrator to recover their lost or forgotten account |
| Actor(s) | User |
| Precondition | The Administrator has registered account and forgotten password |
| Main Scenario | Step 1: Administrator selects account recovery on Log in screen  Step 2: System prints out a form to prompt for Administrator account email  Step 3: Administrator fills up the email and system do validation  Step 3(alternate): If email is invalid, an error message will be displayed, and Administrator is sent back to form for re-entry of right credentials  Step 4: System will generate new password  Step 5: Email with new generated password will be sent to Administrator email address. |

### Activity Diagram



### Sequence Diagram

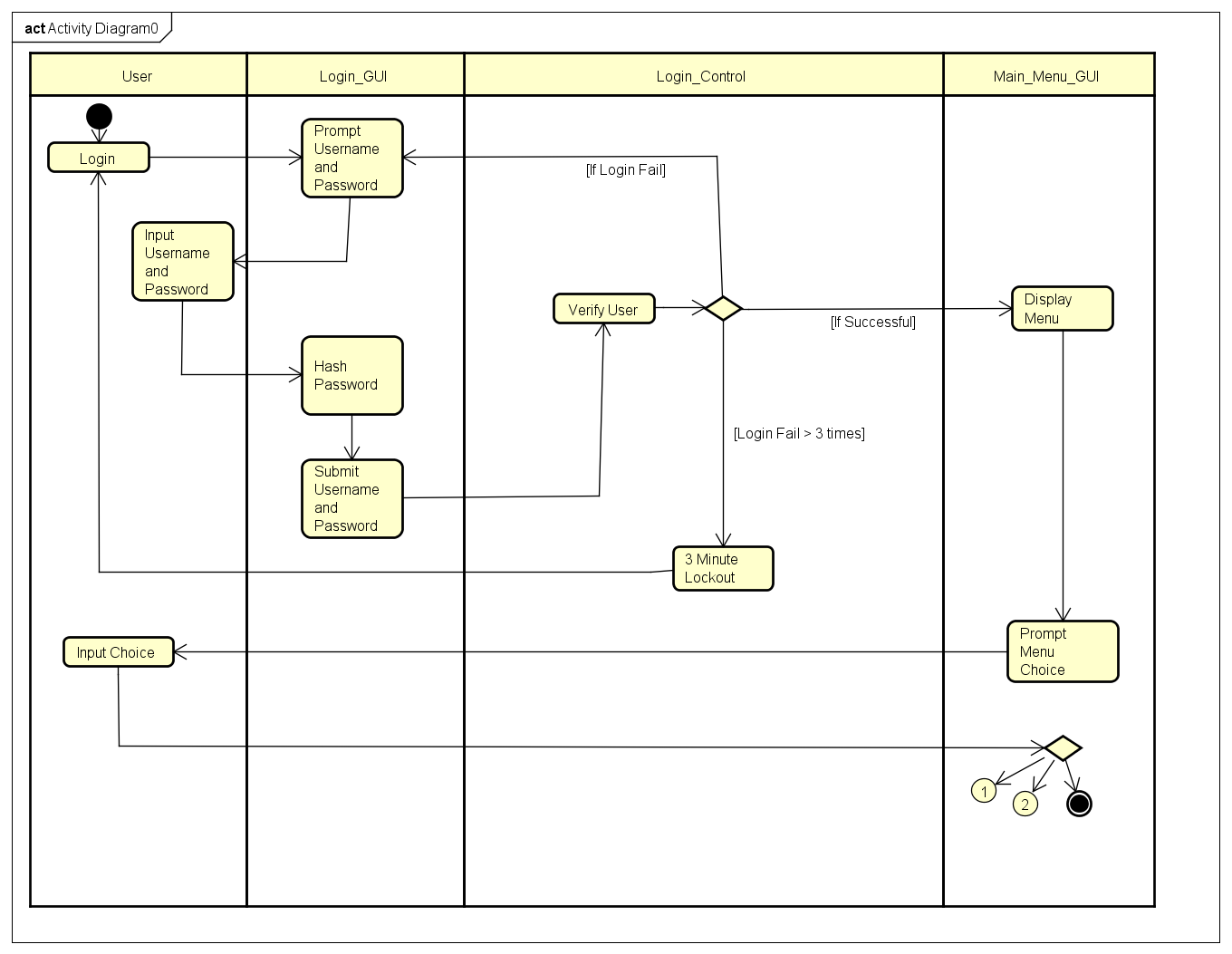


## 3.1 Login

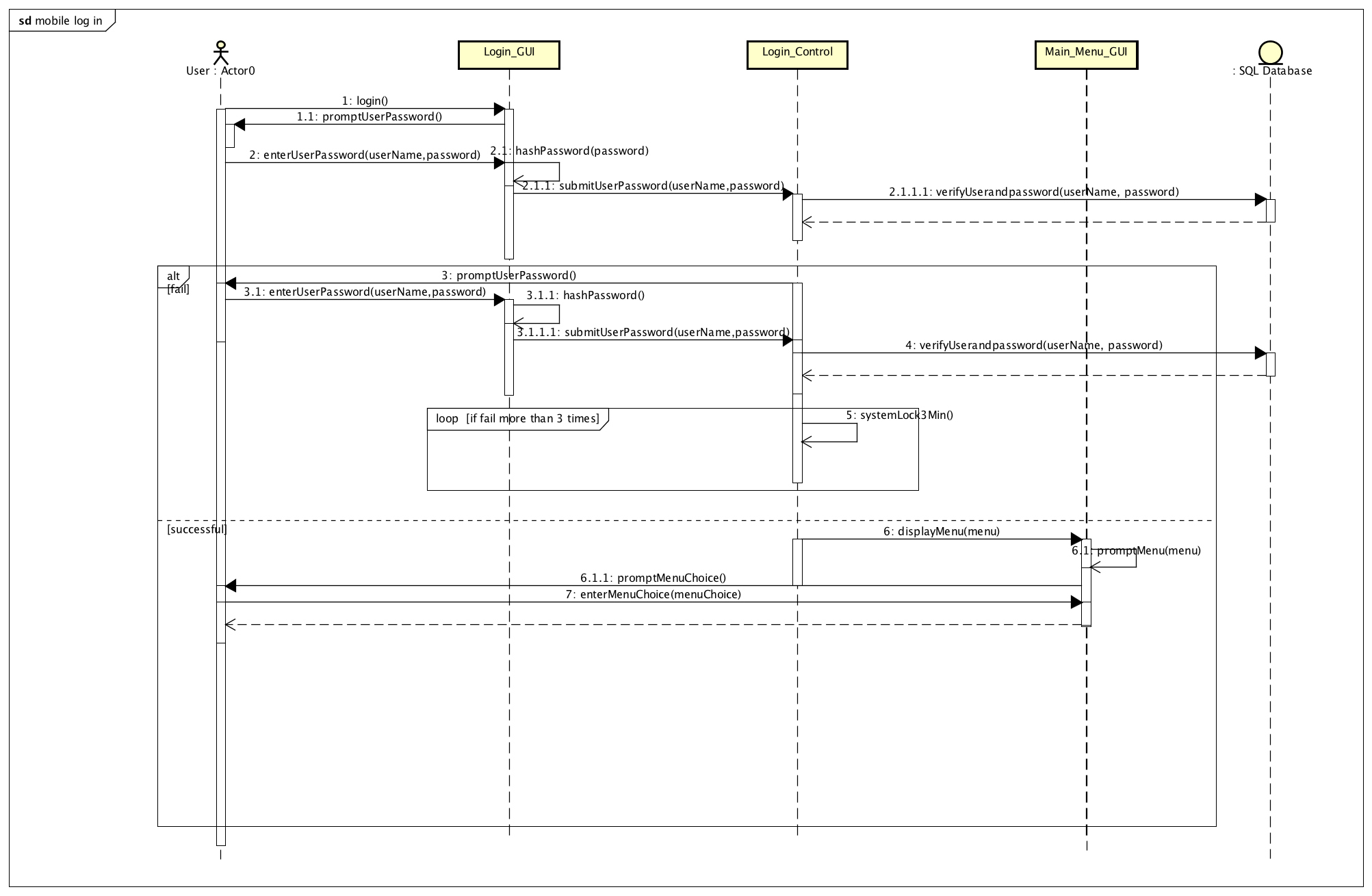
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 3.1 |
| Name | Login |
| Description | To allow User to log in to the system |
| Actor(s) | User |
| Precondition | User wants to use the system |
| Main Scenario | Step 1: User enters their log in credentials in the log in screen  Step 2: System hash the User password and validates with SQL Database  Step 2(alternate): If log in credentials is wrong, an error message will display, and User will be prompt to enter credentials again  Step 2(alternate): If log in credentials is wrong for 3 times, system will be lock from User for 3 minutes  Step 3: System will display Main Menu  Step 4: User will enter choice from Main Menu |

### Activity Diagram



### Sequence Diagram

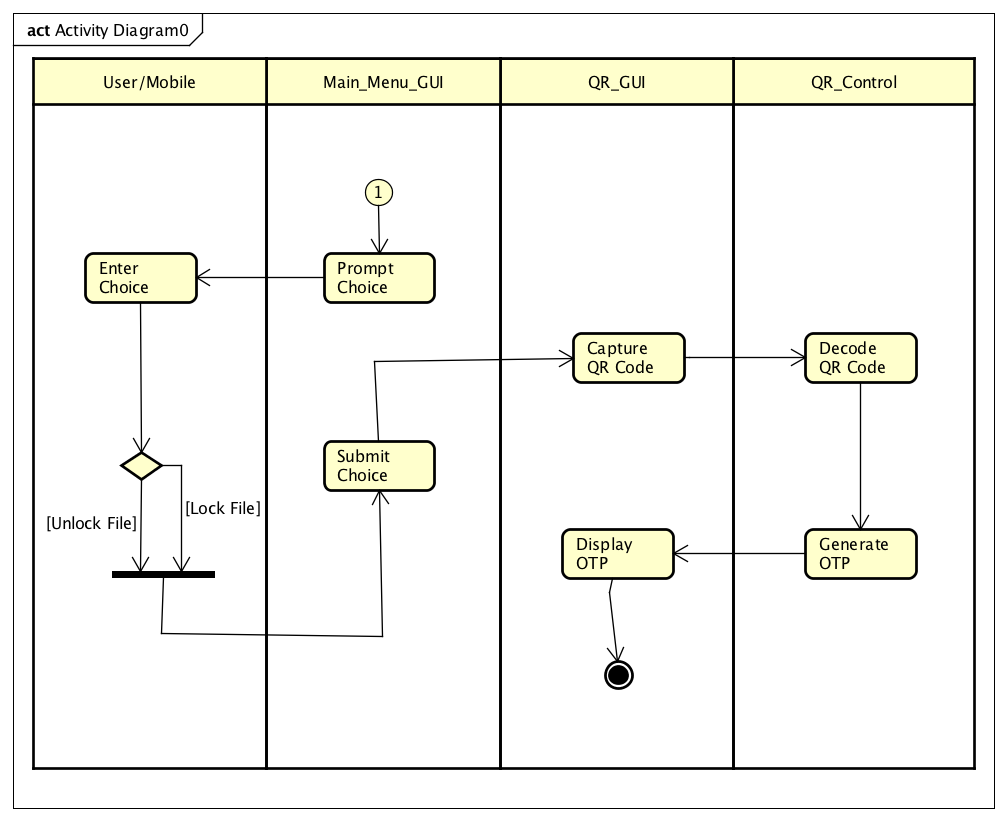


## 3.2 Lock/ Unlock

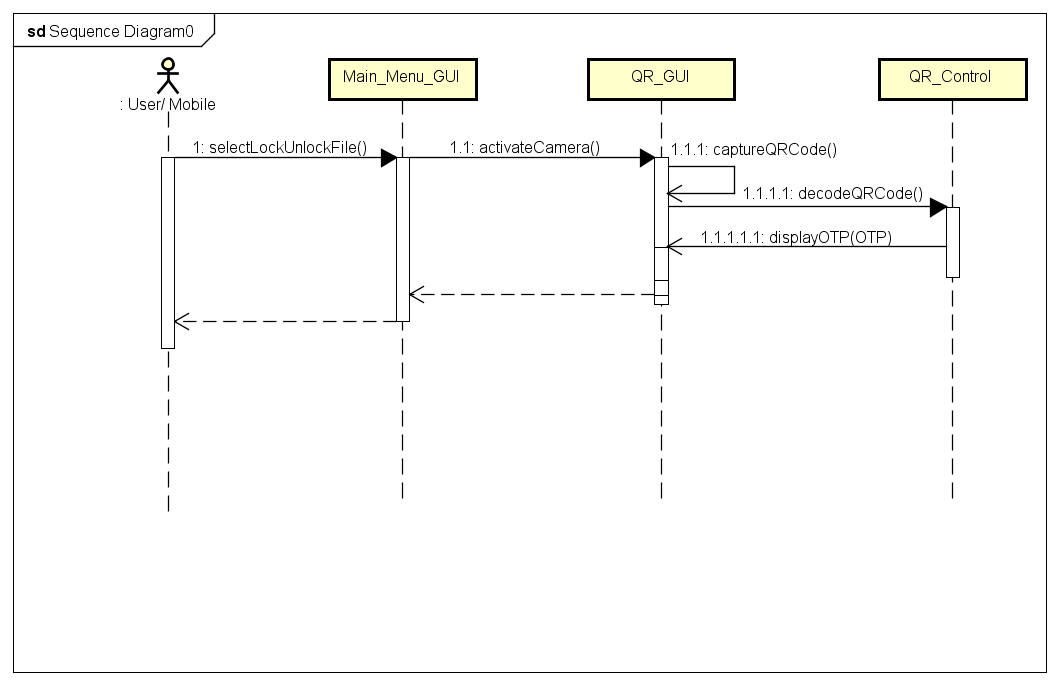
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 3.2 |
| Name | Lock/Unlock |
| Description | To allow user to lock and unlock their folders or files using system cryptography feature |
| Actor(s) | User |
| Precondition | User has a folder or file that they want to lock or unlock |
| Main Scenario | Step 1: User selects LockUnlock file option in the Main Menu.  Step 2: System activates camera of the mobile phone  Step 3: User captures the QRCODE displayed on the webpage of the app  Step 4: System will decode the QRCODE  Step 5: System displays the hidden OTP in the QRCODE for the user to enter into the system to lock/unlock their files` |

### Activity Diagram



### Sequence Diagram

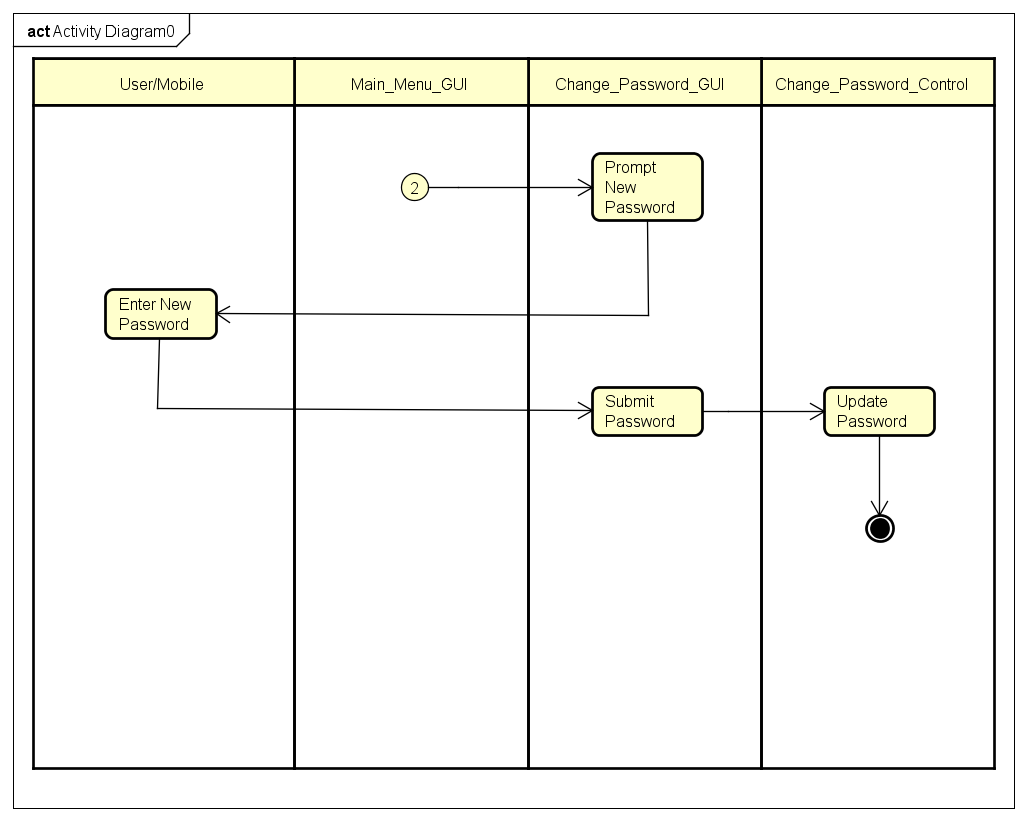


## 3.3 Change Password

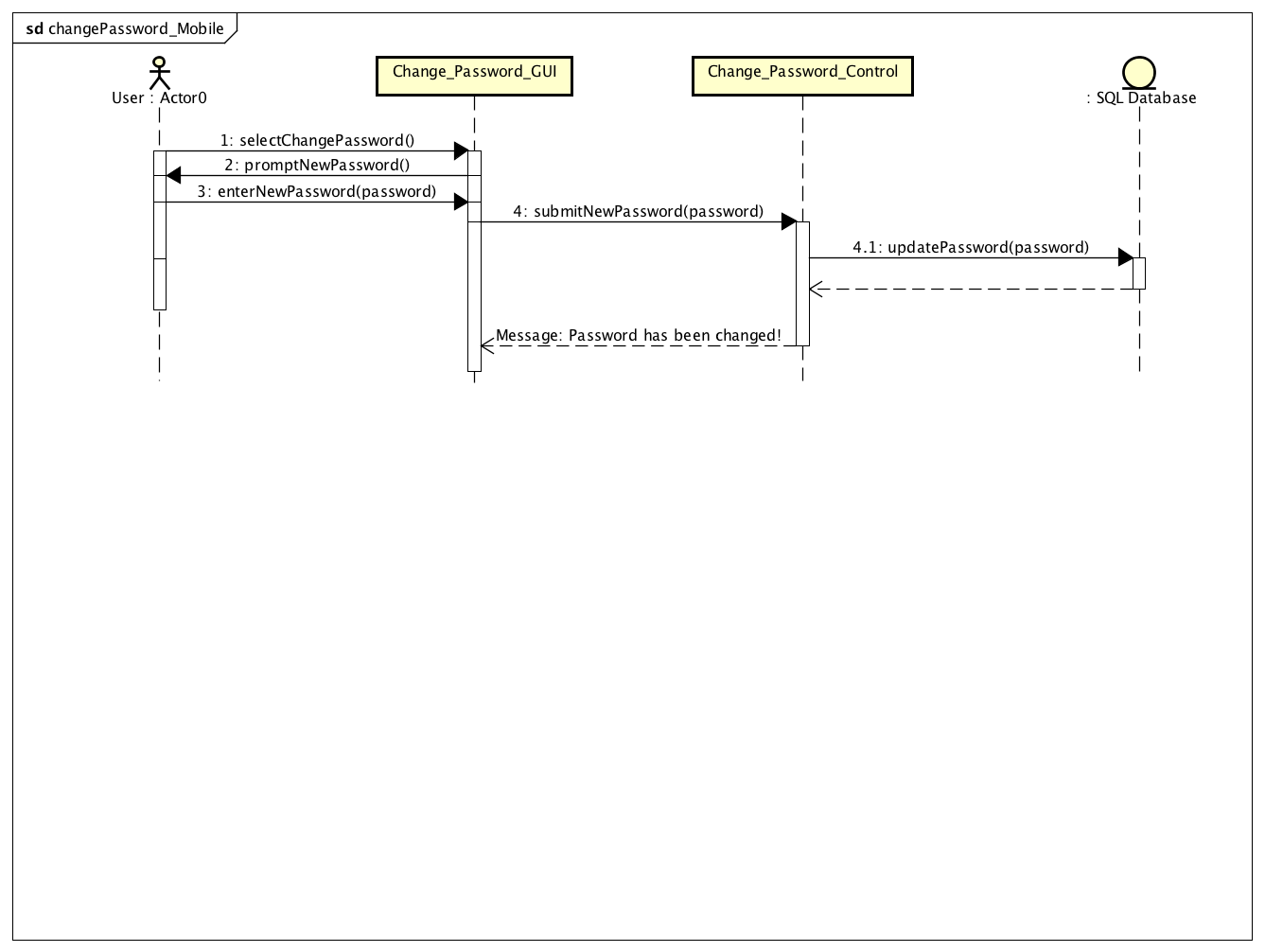
### Use Case

|  |  |
| --- | --- |
| Use Case Textual Description | |
| UC-ID | 3.3 |
| Name | Change Password |
| Description | To allow User to change their password |
| Actor(s) | User |
| Precondition | User wants to change existing password to a new one |
| Main Scenario | Step 1: User enters username and password at Log in screen  Step 2: User selects change password at the main menu  Step 3: System displays a form for user to fill up the new password they desire  Step 4: User enters the new desired password and system do validation  Step 5: System will update the SQL Database with the new password |

### Activity Diagram



### Sequence Diagram



# 4. External Interface Requirements

## 4.1 Hardware Interfaces

The hardware mainly consists of the PC which the user use to lock and unlock their files and a mobile phone with android operating system. Mobile phone must be equipped with a working camera.

## 4.2 Software Interfaces

The user will install the application on their mobile device from Android Play Store and PC application will be installed from the website of go2FA.

## 5. Non-functional Requirements

## 5.1 Performance Requirements

Upon log in, user credentials will be stored locally to minimise communication overhead with the central database.

## 5.2 Safety Requirements

**Changing Password**

The user can change the password by providing the current password and the new password will be updated to the central database.

**Password Recovery**

If the user forgets his password, the system will generate a new password upon request, and send via email to the user

**Multiple Logins**

When a user is log in, other instances of logging in with the same username will not be able to successfully log in.

## 5.3 Security Requirements

## 5.4 Software Quality Attributes

# References

# Appendix A: Glossary