

Project Title: Secure Mailbox

**Submitted By**

**Md. Mazharul Islam**

ID: 171-35-1798

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

**Supervised By**

**Ms. Afsana Begum**

Assistant Professor

Department of Software Engineering

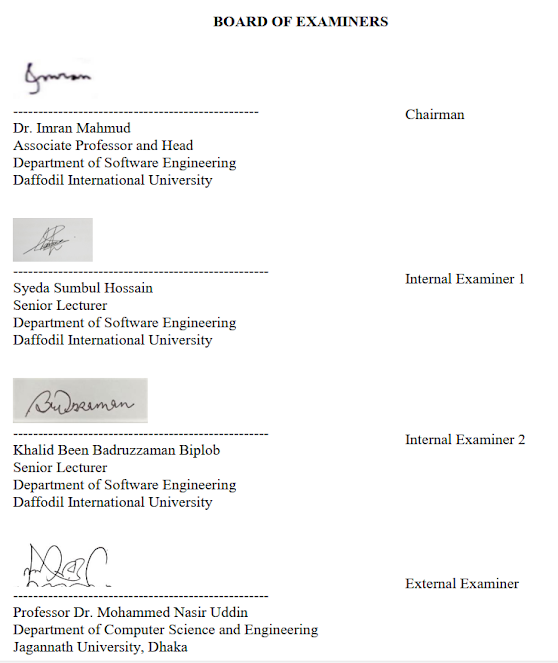
Faculty of Science and Information Technology

Daffodil International University

This project report has been submitted in partial fulfillment of the Requirement for the degree of Bachelor of Science in Software Engineering.

## APPROVAL

This project titled on “**Secure Mailbox**”, submitted by **Md. Mazharul Islam,** **ID:** **171- 35-1798** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to it is style and contents.

****

## DECLARATION

It hereby declares that this project has been done by me (**Md. Mazharul Islam**) under the supervision of **Ms. Afsana Begum**, Assistant Professor, Department of Software Engineering and Daffodil International University. I declare that this project is my original work for the degree of BSc in Software Engineering and that neither the whole work nor any part has been submitted for another degree in this university or anywhere.



#### Md. Mazharul Islam

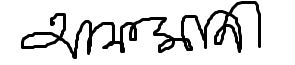
ID: 171-35-1798

Batch: 22th

Department Of Software Engineering

Faculty of Science and Information Technology Daffodil International University

Certified by:



#### Ms. Afsana Begum

Assistant Professor

Department Of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

## ACKNOWLEDGEMENT

I would first like to thank the almighty Allah for allowing us to accomplish this BSc study successfully. We are really thankful for the abundant blessings the Almighty Allah has bequeathed upon us, not only during our study life but in life long.

I would also like to express my sincere gratitude to my **Supervisor, Ms. Afsana Begum**, for the continuous support in my undergraduate project. Her guidance helped me throughout this project and also in writing the documentation.

I would like to thank my family and my parents for trusting in me and my decisions and supporting me spiritually throughout my life.

Last but definitely not least, I would also like to thank some of my friends, for their inspiration, motivation which encourages me a lot to finish this project.

## Table of Contents

Contents

[APPROVAL ii](#_bookmark0)

[BOARD OF EXAMINERS ii](#_bookmark1)

[DECLARATION iii](#_bookmark2)

[ACKNOWLEDGEMENT iv](#_bookmark3)

[Table of Contents v](#_bookmark4)

[List of Figures ix](#_bookmark5)

[Chapter 1 1](#_bookmark6)

[Introduction 1](#_bookmark7)

* 1. [Project Overview 1](#_bookmark8)
  2. [Project Purpose 1](#_bookmark9)
  3. [Project Goal 1](#_bookmark10)
  4. [Benefits and Beneficiaries 2](#_bookmark11)
  5. [Stakeholders 2](#_bookmark12)

1.5.1 [Admin 2](#_bookmark12)

[1.5.2 User 2](#_bookmark13)

[Chapter 2 3](#_bookmark14)

[Software Requirement Specification (SRS) 3](#_bookmark15)

* 1. [Business requirements 3](#_bookmark16)
  2. [SDLC model 4](#_bookmark17)
  3. [Functional requirements 5](#_bookmark19)
     1. Create [User account 6](#_bookmark20)
     2. [Admin Id Verification 7](#_bookmark21)
     3. [Show User List 7](#_bookmark22)
     4. [Create Notification 7](#_bookmark23)
     5. [Create Mail 8](#_bookmark24)
     6. [Encrypt/Decrypt Mail 8](#_bookmark25)
     7. [Live Chat Between User 8](#_bookmark26)
     8. [View Mail List 9](#_bookmark27)
     9. [View Notification List 9](#_bookmark28)
  4. [Non-Functional requirements 9](#_bookmark29)
     1. [Performance 10](#_bookmark30)
     2. [User sensitive data will dissolve from others 10](#_bookmark31)
     3. [Reliability and Availability 10](#_bookmark32)
     4. [Hardware 11](#_bookmark33)
     5. [Response time 11](#_bookmark34)
     6. [Fault Tolerance 11](#_bookmark35)
     7. [Maintenance 12](#_bookmark36)

[Chapter 3 13](#_bookmark38)

[System Analysis 13](#_bookmark39)

* 1. [Use Case Diagram 13](#_bookmark40)
     1. [User 13](#_bookmark41)
     2. [Admin 14](#_bookmark43)
  2. [Use Case Description 15](#_bookmark45)
     1. [Registration with token based authentication for User 15](#_bookmark46)
     2. [Login for Registered Admin 16](#_bookmark47)
     3. [Check User List 17](#_bookmark48)
     4. [Create Notification 18](#_bookmark49)
     5. [Log Out of All Registered Admin 21](#_bookmark52)
     6. [Registration with token based authentication for User 22](#_bookmark53)
     7. [Login for Registered User 23](#_bookmark54)
     8. [Create mail 25](#_bookmark55)
     9. [Encrypt/Decrypt File 26](#_bookmark56)
     10. [Live Chat 27](#_bookmark57)
     11. [View Mail List 28](#_bookmark58)
     12. View Notification List……………………………………………………………….29
     13. View user List…………………………………………………..…………………..30
     14. [Log Out for Registered Patient 29](#_bookmark59)
  3. [Activity Diagram 31](#_bookmark60)
     1. [User Activity Diagram 31](#_bookmark61)
        1. [Admin Activity Diagram 31](#_bookmark62)
     2. [User 36](#_bookmark68)
  4. [Sequence Diagram 39](#_bookmark72)
     1. [Admin 39](#_bookmark73)
     2. [User 42](#_bookmark77)

[Chapter 4 47](#_bookmark83)

[System Design Specification (SDS) 47](#_bookmark84)

* 1. [Development Tools and Technologies 47](#_bookmark85)
     1. [API Development Technology 47](#_bookmark86)
     2. [API Development Tools 49](#_bookmark87)
  2. [Class Diagram 50](#_bookmark88)
  3. [ER Diagram 51](#_bookmark90)

[Chapter 5 52](#_bookmark92)

[System Test 52](#_bookmark93)

* 1. [Testing Features 52](#_bookmark94)
     1. [User Features 52](#_bookmark95)

[5.1.1 Admin Features 53](#_bookmark96)

* 1. [Test Cases 54](#_bookmark97)
     1. [Account Registration (Admin) 54](#_bookmark98)
     2. [Account Registration (User) 55](#_bookmark99)
     3. [Login (Admin) 56](#_bookmark100)
     4. [Login (User) 56](#_bookmark101)
     5. [View User List 57](#_bookmark102)
     6. [Create Notification 58](#_bookmark103)
     7. [Create mail 59](#_bookmark104)
     8. [Encrypt File 59](#_bookmark105)
     9. [Decrypt File 60](#_bookmark106)
     10. [View Mail List 61](#_bookmark107)
     11. [Check Notification 62](#_bookmark108)
     12. [Live Chat 63](#_bookmark109)
     13. [View Notification List 63](#_bookmark110)

[Chapter 6 65](#_bookmark111)

6.1: [User Manual 65](#_bookmark112)

[6.1.1: Admin profile](#_bookmark116) Login [69](#_bookmark116)

[6.1.2: Che](#_bookmark118)ck user list [67](#_bookmark118)

[6.1.3: Create Not](#_bookmark120)ification [68](#_bookmark120)

6.1.4: User Registration….........................................................................................69

6.1.5: User Login…...................................................................................................70

6.1.6: Check Mail List...………………………………………………….......………..….71

6.1.7: User create mail……………………………………………………………………72

6.1.8: Encrypt File……………………………………………………………………….......73

6.1.9: Decrypt File…………………………………………………………………..............74

6.1.10: Live Chat……………………………………………………………………….........75

6.1.11: Check Notification List………………………………………………………..........76

[Chapter 7……………………………………………………………………………………...…77](#_bookmark113)

* 1. [Github Link……………………………………………………………………………......78](#_bookmark115)
  2. [Obstacles & Achievements………………………………………………………...…...78](#_bookmark127)
  3. [Future Scope…………………………………………………………………………......78](#_bookmark129)

References………………………………........................................................................80

Plagiarism Report………………………………..............................................................81

### 

### List of Figures

[Figure 2. 1: SDLC model (Agile)](#_bookmark18)……………………………………………………………….. [13](#_bookmark18)

[Figure 3. 1: Use Case Diagram for](#_bookmark42) User [22](#_bookmark42)

[Figure 3. 1: Use Case Diagram for Admin](#_bookmark44) [22](#_bookmark44)

[Figure 3. 2: Activity Diagram for User](#_bookmark63) [37](#_bookmark63)

[Figure 3. 3: Activity Diagram for Admin](#_bookmark64) [38](#_bookmark64)

[Figure 3. 4: Sequence diagram for user Re](#_bookmark65)gistration [39](#_bookmark65)

[Figure 3. 5: Sequence diagram for](#_bookmark66) user Login [40](#_bookmark66)

[Figure 3. 6: Sequence diagram for creat](#_bookmark67)e mail [41](#_bookmark67)

[Figure 3. 7: Sequence diagram for Live Chat](#_bookmark69) [42](#_bookmark69)

[Figure 3. 8: Sequence diagram for encrypt](#_bookmark70) file [43](#_bookmark70)

[Figure 3. 9: Sequence diagram for decrypt file 4](#_bookmark71)4

[Figure 3. 10: Sequence Diagram for Admin Login](#_bookmark74) [45](#_bookmark74)

[Figure 3. 11: Sequence Diagram for Create Notification](#_bookmark75) [46](#_bookmark75)

[Figure 3. 12: Admin view user list](#_bookmark76) [47](#_bookmark76)

[Figure 4. 1: Class diagram](#_bookmark89) [53](#_bookmark89)

[Figure 4. 2: ER diagram](#_bookmark91) [54](#_bookmark91)

[Figure 6. 1: Admin profile](#_bookmark116) Login [69](#_bookmark116)

[Figure 6. 2: Che](#_bookmark118)ck user list [67](#_bookmark118)

[Figure 6. 3: Create Not](#_bookmark120)ification [68](#_bookmark120)

Figure 6.4: User Registration……………………………………………………………………..69

Figure 6.5: User Login…................................................................................................................70

Figure 6.6: Check Mail List...…………………………………………………………………….71

Figure 6.7: User create mail………………………………………………………………………72

Figure 6.8: Encrypt File…………………………………………………………………………..73

Figure 6.9: Decrypt File…………………………………………………………………………..74

Figure 6.10: Live Chat………………………………………………………………………...….75

Figure 6.11: Check Notification List…………………………………………………………......76

## Chapter 1 Introduction

### Project Overview

Due to the corona pandemic we have been locked ourselves for too long in home. Therefore, we could not go to the office on time whenever any emergency that’s related with official or private documents. Therefore, a question, how we can get/send our documents safely? To solve this problem, we need a virtual environment to get/send documents, where all are documents can encrypt/decrypt by user and they can send all documents with each other.

### Project Purpose

The main purpose of “**Secure Mailbox**” is to make an automation system which will be helpful for the current society. It will be a platform where user can encrypt and decrypt any file easily. User can make an email to send another. Receivers can easily receive a mail and also decrypt a file by this platform. Users can easily create communication with each other using by email and live-chat features.

### Project Goal

The goal of this project is user can encrypt their files using file encryption and decryption features. They can communicate each other with live chat using live chat features. When user get an encrypted files which are encrypt by using this system then, user can decrypt the encrypted files using this platform. Finally user can send mail using this application.

### Benefits and Beneficiaries

From the application all users get benefits. User can encrypt/decrypt all type documents using this platform. User can send/get their sensitive data using this system between few minutes. So that, user can save their time and secure their documents privacy. User can easily communicate with each other and they get more benefits from other applications.

### Stakeholders

In this application there are three types of stakeholder (Project Owner, Admin, User).There are two type of user use this application (Secure Mailbox). User and admin are main user in our application.

#### Admin

Admin play a very major role in the system. They do a large number of activities, after login the system. They can create system update notification in this system. Easily can update their profile information. They can support all user, if user need any technical support.

### User

User play another very major role in the system. They do a large number of activities, after login the system. They can easily find specific user based on user name. Easily can contact another users and they can send mail each other. Easily they can use this platform.

## Chapter 2

## Software Requirement Specification (SRS)

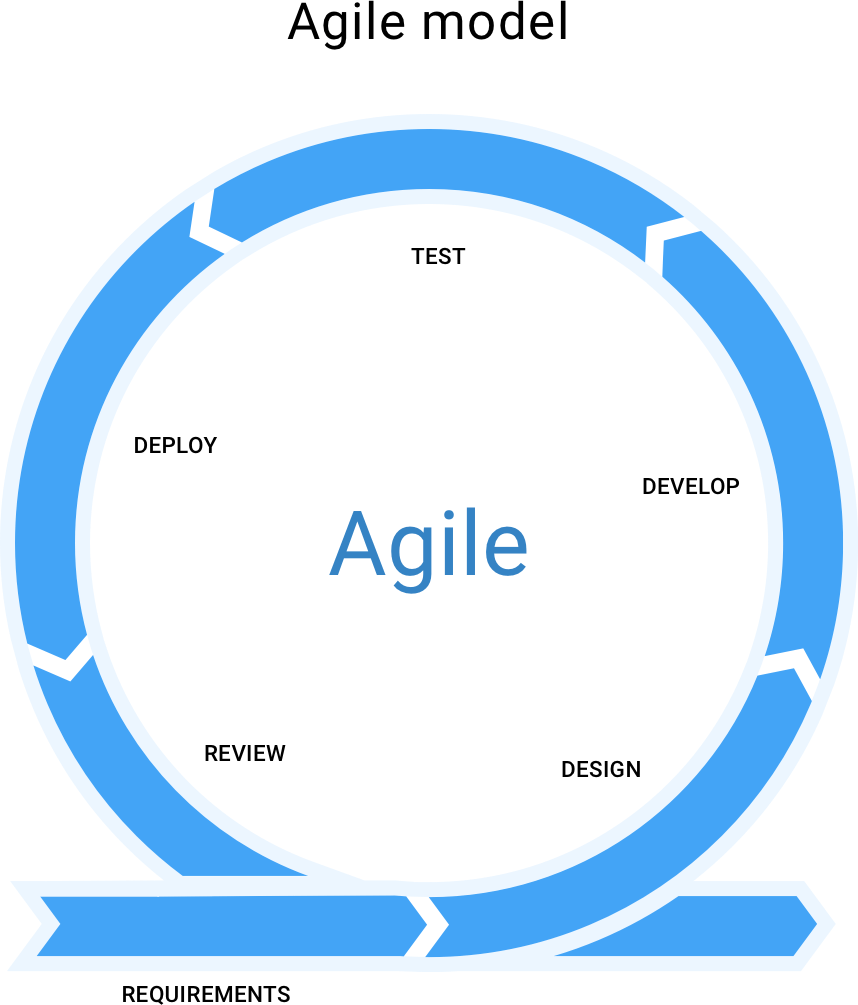
In this chapter we specified about the business requirements of this platform, which SDLC we use, functional and non-functional requirements.

### Business requirements

|  |  |  |
| --- | --- | --- |
| **Feature** | **Definition** | **Requirement shopping** |
| Individual account | For Admin and user they need to email to create an account. |  |
| Compose Mail | User can create a mail to send another user after their login. |  |
| Encrypt/Decrypt File | User can encrypt/decrypt their file and they can save their file after encrypt/decrypt. |  |
| Live chat | User can communicate using live chat features. |  |
| Check Inbox/sending mail list | User can check their sending list and inbox mail. |  |
| Notify about system update/ technical support | Admin can create a notification and admin can provide technical support to user. |  |

### SDLC model

In this project we used “Agile” model to complete the application development. We used agile model because, we needed to communicate with the client in every process. After completing each modules we made sure if the module is okay or not. Based on the client review we change/correct some of our functionality. In this project, we have not any specific requirements. So, we used agile model to complete the project.



*Figure 2. 1: SDLC model (Agile)*

### Functional requirements

Functional requirements refer to the function which are obviously need for the system. They are mandatory contain the application. There is no system avoid the functional requirements. Here, we are going to discuss about functional requirements for this project.

|  |  |
| --- | --- |
| **FR ID** | **Description** |
| FR 01 | Create an User account |

|  |  |
| --- | --- |
| FR 02 | User Id verification |
| FR 03 | Show User List |
| FR 04 | Create Notification |
| FR 05 | Create Mail |
| FR 06 | Encrypt and Decrypt File |
| FR 07 | Live Chat between User to User |
| FR 08 | View Mail List(Send and Receive) |
| FR 09 | View Notification List |

### User account

|  |  |
| --- | --- |
| **FR 01** | Create an user account |
| **Description** | Unregistered users must complete their registration before accessing system functionality. |
| **Stakeholder** | Admin, User |

* + 1. **User Id Verification**

|  |  |
| --- | --- |
| **FR 02** | User Id verification |
| **Description** | System verify user account |
| **Stakeholder** | User |

* + 1. **Show User List**

|  |  |
| --- | --- |
| **FR 03** | Show User List |
| **Description** | Admin can check user list. |
| **Stakeholder** | Admin, User |

* + 1. **Create Notification**

|  |  |
| --- | --- |
| **FR 04** | Create Notification |
| **Description** | Admin can create notification to send system update or user support. |
| **Stakeholder** | Admin |

* + 1. **Create Mail**

|  |  |
| --- | --- |
| **FR 05** | Create Mail |
| **Description** | By using an online create mail method user can create a mail to send another user. |
| **Stakeholder** | User |

* + 1. **Encrypt and Decrypt File**

|  |  |
| --- | --- |
| **FR 06** | Encrypt and Decrypt file |
| **Description** | User can encrypt/decrypt any extensional file through the encryption/decryption feature. |
| **Stakeholder** | User |

* + 1. **Live chat between user**

|  |  |
| --- | --- |
| **FR 07** | Live Chat between user |
| **Description** | All user can use live chat feature to communicate with real-time medium. |
| **Stakeholder** | User |

* + 1. **View Mail List(Send/Receive)**

|  |  |
| --- | --- |
| **FR 08** | View mail List |
| **Description** | User can check mail list, Which mail they are receive and send by them. |
| **Stakeholder** | User |

* + 1. **View Notification List**

|  |  |
| --- | --- |
| **FR 09** | View Notification List |
| **Description** | In this feature user can receive notification from admin and they can check the notification list anytime. |
| **Stakeholder** | User |

* 1. **Non-Functional requirements**

|  |  |
| --- | --- |
| **NFR ID** | **Description** |
| NFR 01 | Performance |
| NFR 02 | User sensitive data will disappear from others |
| NFR 03 | The system must be available 24x7 |
| NFR 04 | Hardware |
| NFR 05 | Response time |
| NFR 06 | Fault Tolerance |
| NFR 07 | Maintenance |

* + 1. **Performance**

|  |  |
| --- | --- |
| **NFR 01** | Request and response will build within a second |
| **Description** | This is a Web Based Application and the responses are fetch from API. |
| **Stakeholder** | Admin, User |

* + 1. **User sensitive data will disappear from others**

|  |  |
| --- | --- |
| **NFR 02** | User sensitive data will dissolve from others |
| **Description** | System will be designed as per requirement. So that, It’s not harm for user. |
| **Stakeholder** | Admin, User |

* + 1. **Reliability and Availability**

|  |  |
| --- | --- |
| **NFR 03** | The system must be available 24x7 |
| **Description** | The system will available 24 hours always. |
|  | Expect the system update/upgrade. |
| **Stakeholder** | N/A |

* + 1. **Hardware**

|  |  |
| --- | --- |
| **NFR 04** | Hardware |
| **Description** | User can reach this application from Mobile and Pc. |
| **Stakeholder** | N/A |

* + 1. **Response time**

|  |  |
| --- | --- |
| **NFR 05** | Response time |
| **Description** | As much as fast the internet speed. |
| **Stakeholder** | N/A |

* + 1. **Fault Tolerance**

|  |  |
| --- | --- |
| **NFR 06** | Always updated the system |
| **Description** | If any problem occurs, then admin will create notification to send all user about system update |
| **Stakeholder** | Admin |

* + 1. **Maintenance**

|  |  |
| --- | --- |
| **NFR 07** | The system helps to update any Information in any time |
| **Description** | The admin can changes or updates any information in any situation from database. |
| **Stakeholder** | Admin |

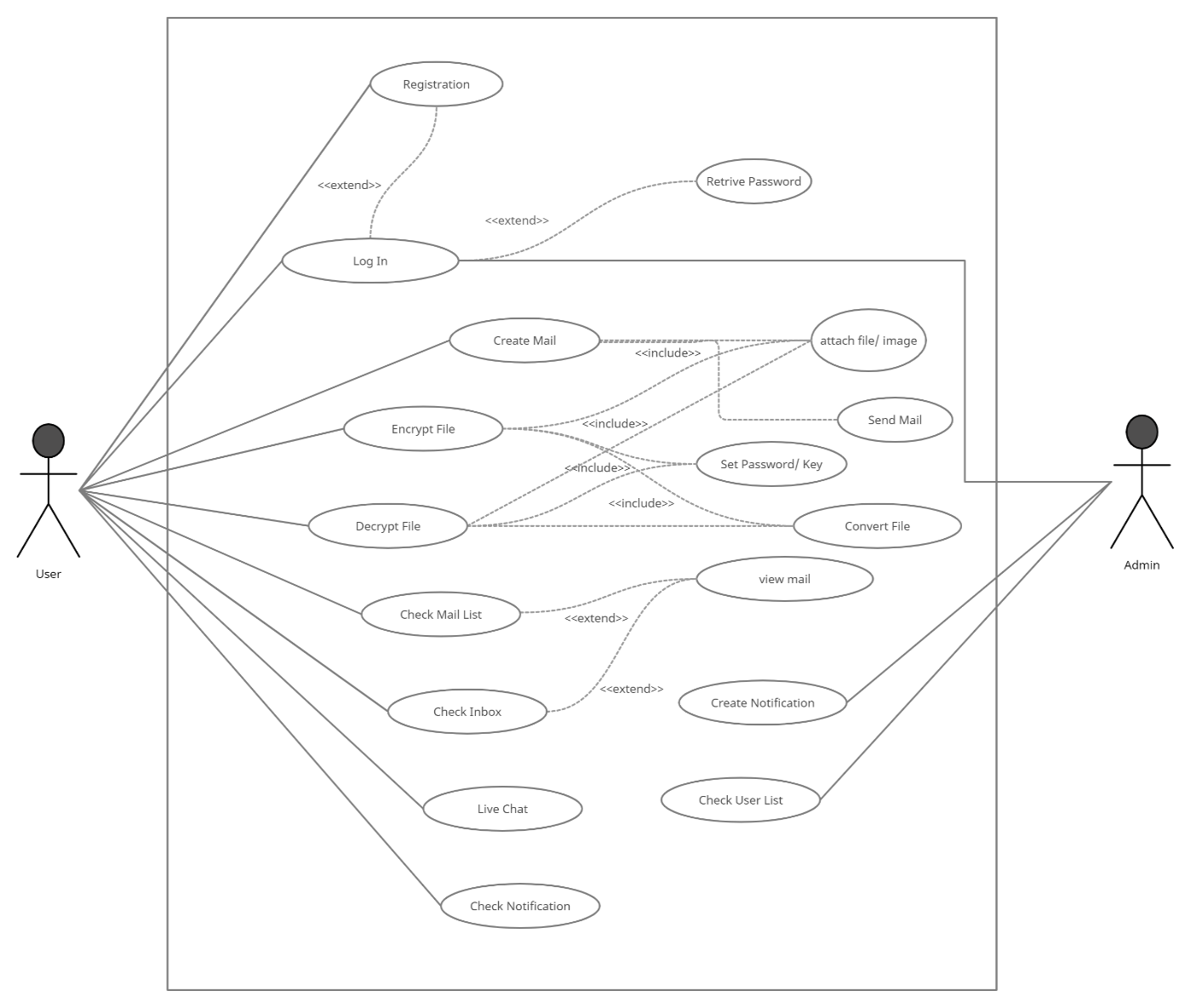
## Chapter 3

**System Analysis**

In this chapter we showed the use case diagrams, use case description, activity diagrams, sequence diagrams of the Online Doctor Chamber.

### Use Case Diagram

In these diagrams we showed the overall process of the doctor and patient including login, registration, appointment, scheduling, prescription, communication etc. We have showed both of the diagrams here and the use case description is down below.

****

*Figure 3. 1: Use Case Diagram*

### Use Case Description

#### Registration with token based authentication for User

|  |  |  |
| --- | --- | --- |
| **Use Case** | Registration with token based authentication. | |
| **Goal** | User will be able to login into the system. | |
| **Predictions** | The user has to fill up the registration form. | |
| **Success end condition** | The user gets access to the login page. | |
| **Failed end condition** | Has to fill up the registration for again with correct information. | |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | The system is ready for login. | |
| **Description** | **Step** | **Action** |
| 1 | Fill up the registration form properly. |
| 2 | Users will be able to login into the system. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Registration form may not fill up properly. |
| 2 | System may take time to load. |
| **Quality Requirements** | **Step** | **Requirement** |

|  |  |  |
| --- | --- | --- |
|  | 1 | Has to fill up each and every detail. |

#### Login for Registered User

|  |  |  |
| --- | --- | --- |
| **Use Case** | Login for Registered User | |
| **Goal** | User will get access to the system. | |
| **Predictions** | The user can login with email and password. | |
| **Success end condition** | The user gets access to the system. | |
| **Failed end condition** | Has to get the authorization again to have accessing credentials. | |
| **Primary actors: Secondary actors:** | User and Admin  Admin | |
| **Trigger** | System is accessed by the user. | |
| **Description** | **Step** | **Action** |
| 1 | Input email and password. |
| 2 | System gives access. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Username or password can be forgotten. |
| 2 | Unauthorized person may try to login. |

|  |  |  |
| --- | --- | --- |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | Two step authentication. |
| 2 | Account recovery. |

* + 1. **Check User List**

|  |  |  |
| --- | --- | --- |
| **Use Case** | Check User List | |
| **Goal** | Admin will be able to check the user list. | |
| **Predictions** | By checking the user list admin can send notify each user or admin can give support to user. | |
| **Success end condition** | The user send notify to all users about system update. | |
| **Failed end condition** | The user not receive the notification. | |
| **Primary actors: Secondary actors:** | Admin  Admin | |
| **Trigger** | Admin notify to user. | |
| **Description** | **Step** | **Action** |
| 1 | Admin will check the user list. |
| 2 | Admin send notification to the user. |

|  |  |  |
| --- | --- | --- |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | User may not receive the notification. |
| 2 |  |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | Admin can support to user serially. |

* + 1. **Create Notification**

|  |  |  |
| --- | --- | --- |
| **Use Case** | Create Notification | |
| **Goal** | Admin can create notification to notify all user. | |
| **Predictions** | By checking the user list admin notify to all. | |
| **Success end condition** | The admin send notification successfully and user get notify. | |
| **Failed end condition** | The user may not receive notification. | |
| **Primary actors: Secondary actors:** | Admin  Super Admin | |
| **Trigger** | Admin notify users. | |
| **Description** | **Step** | **Action** |

|  |  |  |
| --- | --- | --- |
|  | 1 | Admin will check user list. |
| 2 | Admin will notify users. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | User may not receive the notification. |
| 2 |  |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | Admin has to maintain support by list |

* + 1. **Create Mail**

|  |  |
| --- | --- |
| **Use Case** | Create mail |
| **Goal** | User Create a mail to send another user. |
| **Predictions** | The user may login to create an email. |
| **Success end condition** | The user send mail to destination through the mail address and another user get the mail. |
| **Failed end condition** | User does create again the mail. |
| **Primary actors: Secondary actors:** | User  Admin |

|  |  |  |
| --- | --- | --- |
| **Trigger** | A mail send by the user to user. | |
| **Description** | **Step** | **Action** |
| 1 | Input the required information for create a mail. |
| 2 | Send it to the user. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | System may crash because of the internet. |
| 2 | User may send before fulfil requirements. |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | The mail has to contain the detailed information of the user. |

* + 1. **Encrypt/Decrypt Mail**

|  |  |
| --- | --- |
| **Use Case** | Encrypt/Decrypt mail |
| **Goal** | User will be able to encrypt/decrypt any file |
| **Predictions** | The user need to attach any file. |
| **Success end condition** | User provide security key to encrypt/decrypt their file. |
| **Failed end condition** | If user does not enter security key |

|  |  |  |
| --- | --- | --- |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | A file encrypt/decrypt by user. | |
| **Description** | **Step** | **Action** |
| 1 | Input the required file and key. |
| 2 | Save the file in the device. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | System may crash because of the internet. |
| 2 | User may mistake without attach file or security key. |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | The file has to contain the detailed information of the user. |

* + 1. **Log Out for Registered Admin**

|  |  |
| --- | --- |
| **Use Case** | Log Out for Registered Admin |
| **Goal** | Admin can log out of their account |
| **Predictions** | Admin is logged into the system |

|  |  |  |
| --- | --- | --- |
| **Success end condition** | Admin logs out of their dashboard | |
| **Failed end condition** | Cannot access system | |
| **Primary actors: Secondary actors:** | Admin  Admin | |
| **Trigger** | Admin click on the “Logout” button | |
| **Description** | **Step** | **Action** |
| 1 | Admin clicks on the “Logout” button |
| 2 | Member gets log out of the system |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Cannot access the system |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | N/A |

* + 1. **Registration with token based authentication for User**

|  |  |
| --- | --- |
| **Use Case** | Registration with token based authentication |
| **Goal** | User will be able to login into the system. |
| **Predictions** | The user has to fill up the registration form. |

|  |  |  |
| --- | --- | --- |
| **Success end condition** | The user gets access to the login page. | |
| **Failed end condition** | Has to fill up the registration for again with correct information. | |
| **Primary actors: Secondary actors:** | User  User | |
| **Trigger** | The system is ready for login. | |
| **Description** | **Step** | **Action** |
| 1 | Fill up the registration form properly. |
| 2 | User will be able to login into the system. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Registration form may not fill up properly. |
| 2 | System may take time to load. |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | Has to fill up each and every detail. |

* + 1. **Login for Registered User**

|  |  |
| --- | --- |
| **Use Case** | Login for Registered User |
| **Goal** | User will get access to the system. |

|  |  |  |
| --- | --- | --- |
| **Predictions** | The user has to have the username and password. | |
| **Success end condition** | The user gets access to the system. | |
| **Failed end condition** | Has to get the authorization again to have accessing credentials. | |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | System is accessed by the user. | |
| **Description** | **Step** | **Action** |
| 1 | Input email and password. |
| 2 | System gives access. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Username or password can be forgotten. |
| 2 | Unauthorized person may try to login. |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | Two step authentication. |
| 2 | Account recovery. |

* + 1. **Live Chat between user**

|  |  |  |
| --- | --- | --- |
| **Use Case** | Live chat between user | |
| **Goal** | User will be able to contact with them through the live chat. | |
| **Predictions** | By searching the user mail they can contact with each-others | |
| **Success end condition** | The user send message to another user. | |
| **Failed end condition** | If user not select another user | |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | User send message to others. | |
| **Description** | **Step** | **Action** |
| 1 | User will check the user list. |
| 2 | User send the message. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | User may not view expected user information |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | N/A |

* + 1. **View mail list**

|  |  |  |
| --- | --- | --- |
| **Use Case** | View mail list | |
| **Goal** | User will be able to view their mail list Send/Receive mail List) | |
| **Predictions** | User successfully view the mail list. | |
| **Success end condition** | User check mail list successfully. | |
| **Failed end condition** | If Internet connection will lost. | |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | View mail list. | |
| **Description** | **Step** | **Action** |
| 1 | User will check inbox to check mail. |
| 2 | User will check send mail list. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Users may not find the mail list if they not receive or send mail. |
| 2 |  |
| **Quality Requirements** | **Step** | **Requirement** |

|  |  |  |
| --- | --- | --- |
|  | 1 | The user can view the mail list if user send any mail to another or receive. |

* + 1. **View notification list**

|  |  |  |
| --- | --- | --- |
| **Use Case** | View notification list | |
| **Goal** | User will be able to check notification by the notification list which are through from Admin. | |
| **Predictions** | The user get notification by the notification list from admin. | |
| **Success end condition** | User get reach all notification. | |
| **Failed end condition** | If the internet speed is too much slow or it’s will disconnect. | |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | The user can see the notification. | |
| **Description** | **Step** | **Action** |
| 1 | After see the notification list user know the update information. |
| 2 |  |
| **Alternative Flows** | **Step** | **Branching Action** |

|  |  |  |
| --- | --- | --- |
|  | 1 | System may crash because of the internet. |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | The user may get notify from admin. |

* + 1. **Log Out for Registered User**

|  |  |  |
| --- | --- | --- |
| **Use Case** | Log Out for Registered User | |
| **Goal** | User can log out of their account | |
| **Predictions** | User is logged into the system | |
| **Success end condition** | User logs out of their dashboard | |
| **Failed end condition** | Cannot access system | |
| **Primary actors: Secondary actors:** | User  Admin | |
| **Trigger** | Patient click on the “Logout” button | |
| **Description** | **Step** | **Action** |
| 1 | User clicks on the “Logout” button |
| 2 | User gets log out of the system |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1 | Cannot access the system |

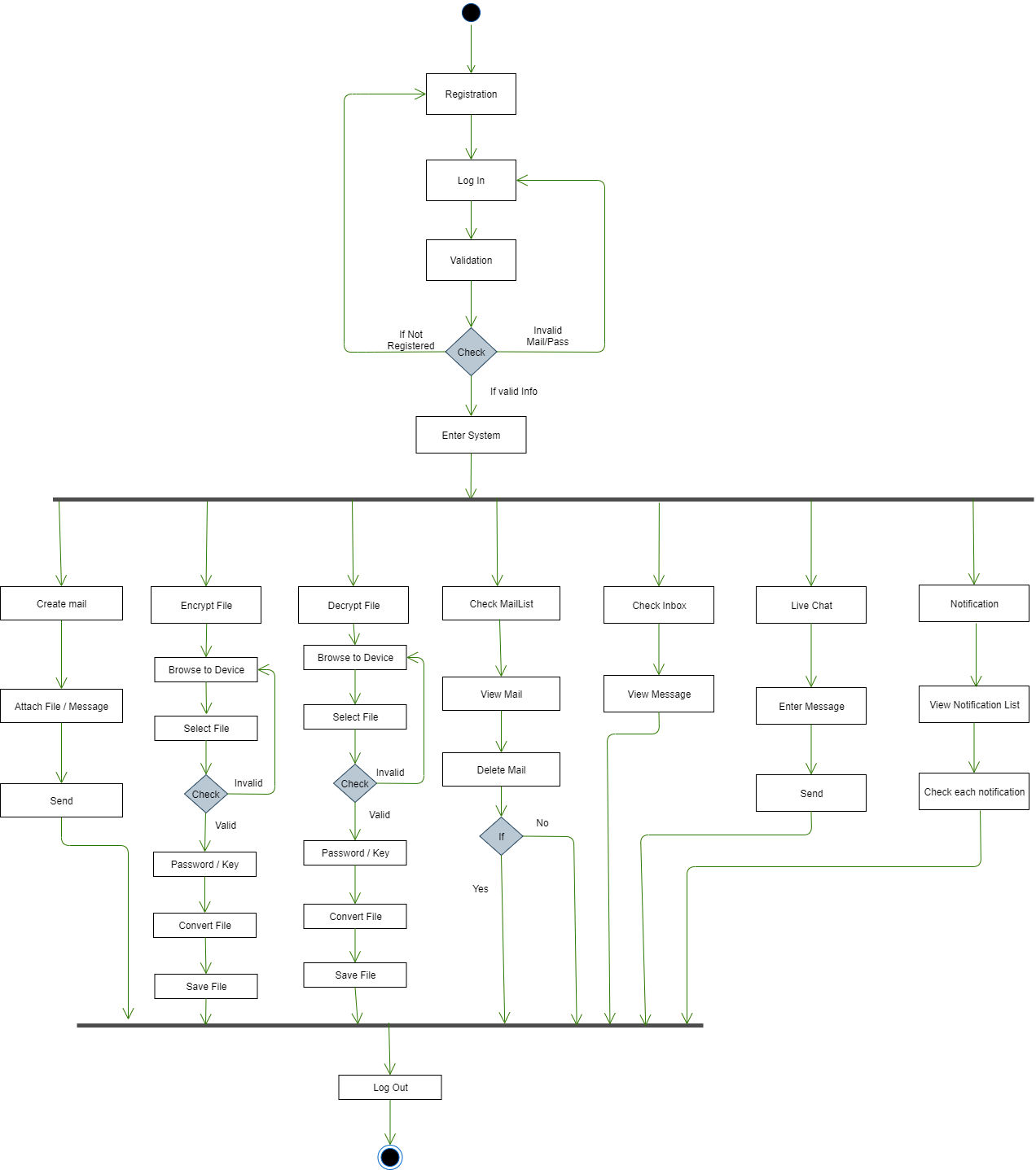
|  |  |  |
| --- | --- | --- |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | N/A |

* 1. **Activity Diagram**

#### User

* + - 1. **Activity Diagram for User**

This figure represents the all feature of user of the Secure Mailbox.

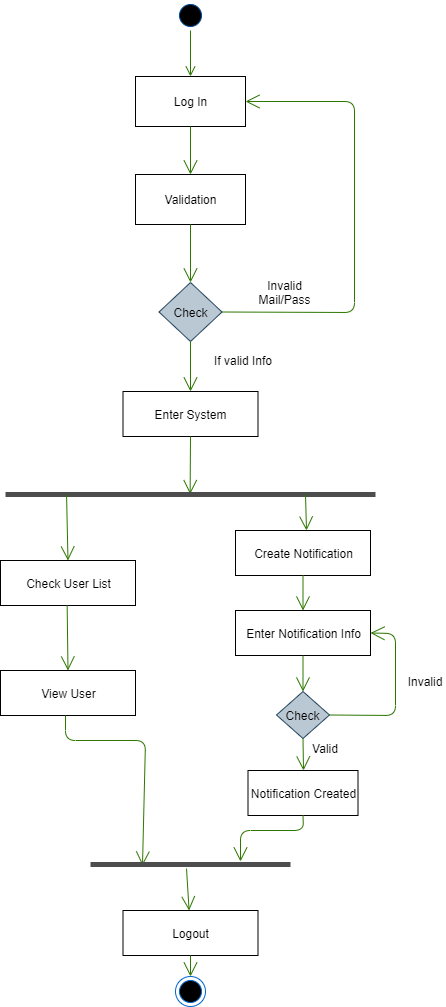


*Figure 3. 2: Activity Diagram for User*

#### Admin

#### Activity Diagram for Admin

This figure represents the all feature of admin part of the Secure Mailbox.

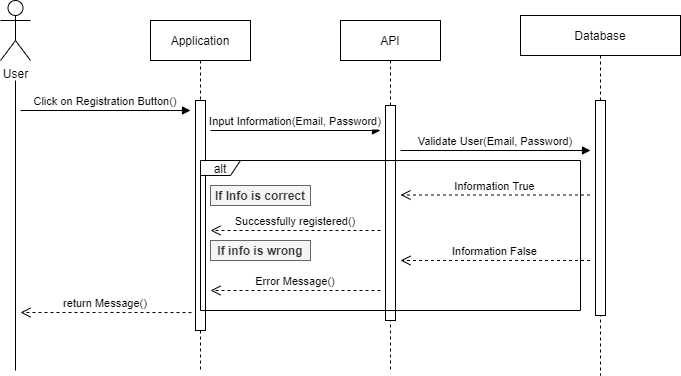


*Figure 3. 3: Activity Diagram for Admin*

### Sequence Diagram

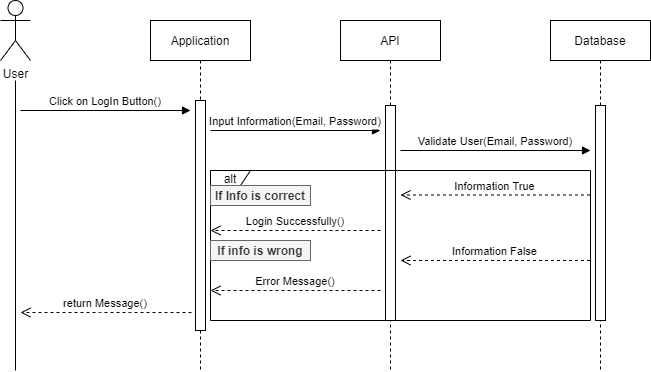
#### User

* + - 1. **Sequence Diagram for User Registration**

****

*Figure 3.4*: Sequence Diagram for Registration

#### Sequence Diagram for User Login

****

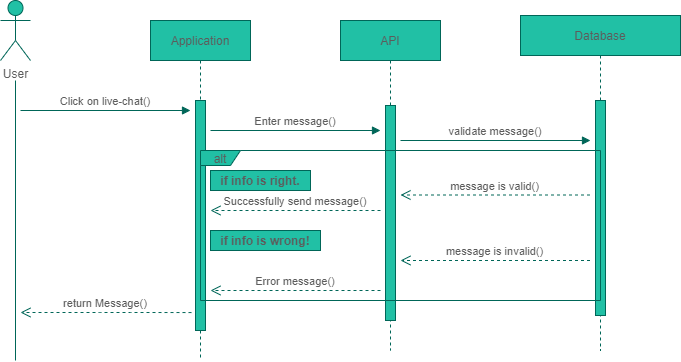
*Figure 3.5*: Sequence Diagram for Login

#### Create Mail

#### 

*Figure 3.6*: Create mail

* + - 1. **Live Chat between user**

****

*Figure 3.7*: Live Chat

#### Sequence Diagram for user file encryption

#### 

*Figure 3.8*: Sequence Diagram for user file encryption

#### Sequence Diagram for user file decryption

#### 

*Figure 3.9*: Decrypt file

#### Admin

#### Admin Login Sequence Diagram

#### 

*Figure 3.10*: Admin Login

#### Create Notification

#### 

*Figure 3. 11:* Create and send notification

#### View user list

#### 

*Figure 3. 12:* View User List

## Chapter 4

## System Design Specification (SDS)

In this chapter we talked about the stack that we used for making Secure Mailbox, tools, class diagram, and entity-relationship diagram.

### Development Tools and Technologies

#### API Development Technology

* + - 1. **JavaScript (ES6+)**

**JavaScript** is a scripting or programming language that allows you to implement complex features on web applications. **ES6** stands for ECMAScript 6. ECMAScript was created to standardize JavaScript. It is a major enhancement to the JavaScript language.

#### Node.js

**Node.js** is an open source server environment. Node.js runs on various platforms like Linux, Mac OS, and Windows etc. It is a JavaScript runtime built on Chrome is V8 JavaScript engine.

#### Express.js

**Express** is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications **APIs**.

#### Swagger

**Swagger** allows you to describe the structure of your **APIs** so that machines can read them. Swagger **doc** does this by asking your API to return a **JSON** that contains a detailed

Description of your entire API.

#### Babel

Babel is an open source JavaScript Trans compiler that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript that can be run by older JavaScript engines.

#### JWT

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self- contained way for security transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed

#### Jest

Jest is a JavaScript testing framework designed to ensure correctness of any JavaScript codebase.

#### No-demon

No-demon is a utility that will monitor for any changes in your source and automatically restart your server. Perfect for development.

#### MongoDB

MongoDB is the most popular database for modern applications. It is a cross-platform document-oriented database program. Classifier as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

#### API Development Tools

* + - 1. **Visual Studio Code**

Visual Studio Code is an open source editor made by Microsoft for Linux, Mac OS, and Windows. Many features include support for debugging, snippets, syntax highlighting, code refactoring, intelligent code completion, Git etc.

#### Postman

Postman is a great tool, when trying to make and test **RESTful APIs.** It offers a sleek user interface with which to make HTML requests, without the hassle of writing a bunch of code just to test an APIs functionality.

#### Compass

Compass is the official GUI for MongoDB. It helps users to make clever decisions about the data-structure, querying, indexing and many more actions developer can perform on the database.

#### Version Control (Github)

Github is an open source repository hosting service, sort of like a cloud for code. It offers the distributed version control and source code management functionality of **Git**.

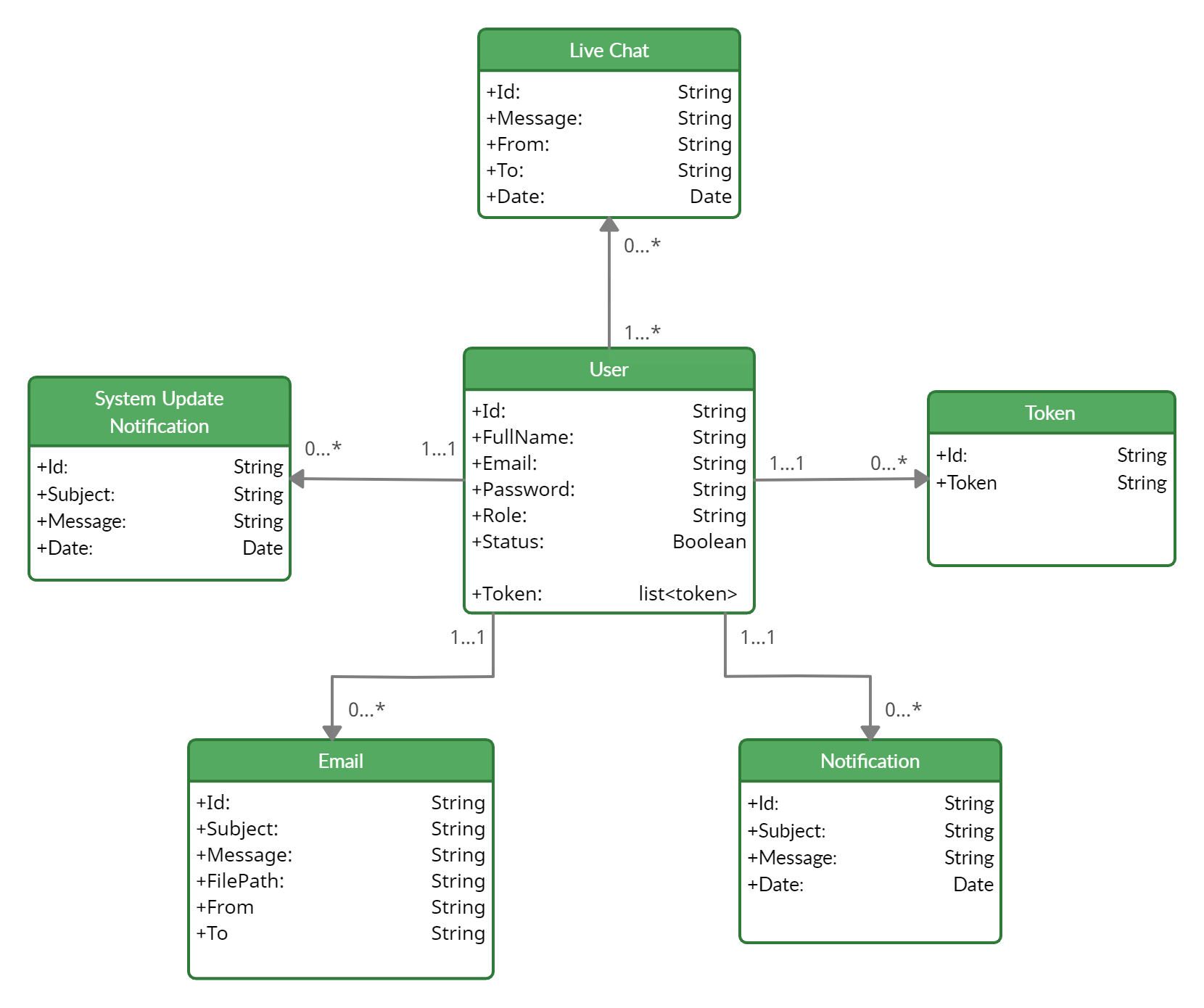
#### Github CI

Github provides a Node.js workflow that will work for most Node.js projects. It is a workflow to build and test Node.js projects. If continuous integration (CI) tests pass then you may want to deploy your code. Developer can check their code and generate a report.

#### Heroku

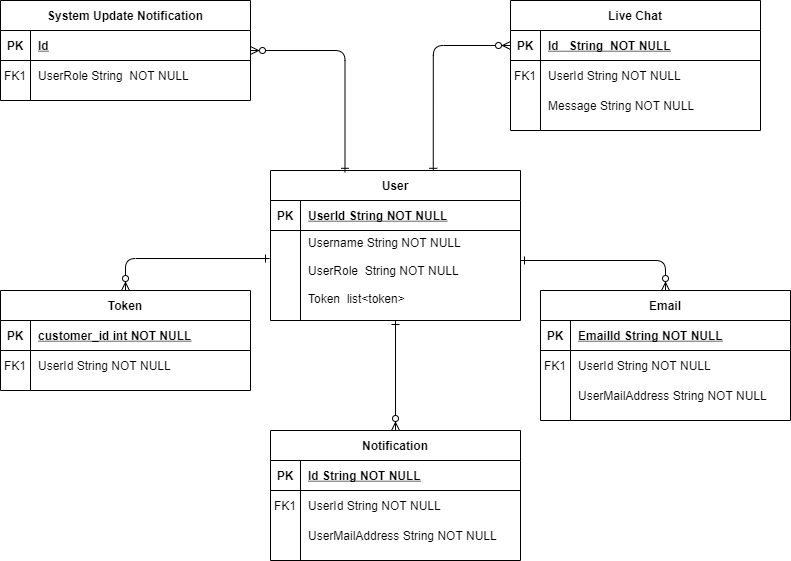
Heroku is a platform of a service that enables for developers to build, run and operate applications entirely in the cloud. We are deploying our API on this.

### Class Diagram

****

*Figure 4. 1:* Class diagram

### ER Diagram

****

*Figure 4. 2:* ER diagram

### Testing Features

Priority Level

* + 1. Low priority
    2. Medium priority
    3. High priority

#### 5.1.1 Admin Features

**Chapter 5 System Test**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Priority** | **Description** |
| Registration | 3 | System must validate new registration |
| Log In | 3 | System must authentication login information |
| Create Notification | 3 | Admin need to create Notification |
| View User List | 2 | Admin need to view user list |

#### 5.1.1 User Features

|  |  |  |
| --- | --- | --- |
| **Feature** | **Priority** | **Description** |
| Registration | 3 | System must validate new registration |
| Log In | 3 | System must authentication login information |
| Create Mail | 3 | User can create mail |
| Encrypt/Decrypt File | 3 | User can encrypt/decrypt file |
| Live Chat | 3 | User can communicate with real-time Live chat feature. |
| View Mail List(Send) | 2 | User can view mail list. |
| View mail List(Receive) | 2 | User can view mail list which are receive they are. |
| View Notification List | 2 | User can view notification list. |
| View User List | 2 | User can view user list to send message. |

### Test Cases

#### Account Registration (Admin)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**: 5.2.1 | | | **Test Case Name:** Account Registration (Admin) | | |
| **System:** Secure mailbox | | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | | **Design Date:** | | |
| **Executed by:** | | | **Execution Date:** | | |
| **Description:** Admin must register to gain access to the create notification. To register an account, must be provided an emailand password. | | | | | |
| **Pre-condition:**   * Email address must be unique. * Passwords must be between 3 to 30 characters. | | | | | |
| **Step** | **Email** | **Password** | **Response** | **Pass/Fail** | **Comment** |
| 1 | admin@securemail.com | admin123 | Registration successfully | Pass |  |
| 2 | admin1@securemail.com | admin123 | Registration successfully | Pass |  |
| 3 | admin2@securemail.com |  | Enter Your Password | Fail |  |
| **Post-condition:** A new user account has been registered successfully | | | | | |

#### Account Registration (User)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**: 5.2.2 | | | **Test Case Name:** Account Registration (User) | | |
| **System:** Secure Mailbox | | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | | **Design Date:** | | |
| **Executed by:** | | | **Execution Date:** | | |
| **Description:** User must register to gain access to the create schedule. To register an account, must be provided a valid email and password. | | | | | |
| **Pre-condition:**   * Email address must be unique. * Passwords must be between 3 to 30 characters. | | | | | |
| **Step** | **Email address** | **Password** | **Response** | **Pass/Fail** | **Comment** |
| 1 | mazharul@securemail.com | mazharul123 | Registration successfully | Pass | Figure 6.4 |
| 2 | mehedi@securemail.com | mehedi123 | Registration successfully | Pass | Figure 6.4 |
| 3 | user2@securemail.com | Tanjina1122 | Enter correct mail / password | Fail |  |
| **Post-condition:** A new user account has been registered successfully | | | | | |

#### Login (Admin)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**: 5.2.3 | | | **Test Case Name:** Login (Admin) | | |
| **System:** Secure Mailbox | | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | | **Design Date:** | | |
| **Executed by:** | | | **Execution Date:** | | |
| **Description:** Admin must login to gain access to the created notification. The system will check for authentication and authorization | | | | | |
| **Pre-condition:**   * Admin must be use valid email and password | | | | | |
| **Step** | **Email address** | **Password** | **Response** | **Pass/Fail** | **Comment** |
| 1 | admin1@securemail.com | 123 | Login successfully | Pass | Figure 6.1 |
| 2 | admin1@securemail.com | 1456d | Incorrect credential | Fail |  |
| **Post-condition:** Admin will successfully log into the system | | | | | |

#### Login (User)

|  |  |
| --- | --- |
| **Test Case**: 5.2.4 | **Test Case Name:** Login (User) |
| **System:** Secure mailbox | **Subsystem:** |
| **Designed by:** Mazharul Islam | **Design Date:** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Executed by:** | | | **Execution Date:** | | |
| **Description:** User must login to gain access. The system will check for authentication and authorization | | | | | |
| **Pre-condition:**   * user must input valid email and password | | | | | |
| **Step** | **Email address** | **Password** | **Response** | **Pass/Fail** | **Comment** |
| 1 | user1@securemail.com | 123 | Login successfully | Pass | Figure 6.5 |
| 2 | user1@securemail.com | 675656cfdcfgg | Incorrect credential | Fail |  |
| **Post-condition:** User will successfully log into the syste | | | | m |  |

#### View User List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**: 5.2.5 | | **Test Case Name:** View User List | | |
| **System:** Secure Mailbox | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Registered admin can view the user list | | | | |
| **Pre-condition:**   * Database has at least one user information to display | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Admin Click on User List | System will display User list | Pass |  |
| **Post-condition:** Display all old or new user | | | | |

#### Create Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**: 5.2.6 | | **Test Case Name:** Create Notification | | |
| **System:** Secure Mailbox | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Admin can create notification | | | | |
| **Pre-condition:**   * Must be logged in | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | Fill up all the required boxes and clicks on ‘Send’ button | System Send the notification | Pass |  |
| 2 | Does not fill up all the required boxes and  clicks on ‘Send’ button | All required boxes must be  filled | Fail |  |
| **Post-condition:** Notification Created Successfully. | | | | |

#### Create Mail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**: 5.2.7 | | **Test Case Name:** Create Mail | | |
| **System:** Secure Mailbox | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Registered user can create his/her mail | | | | |
| **Pre-condition:**   * Must be logged in | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | Fill up all the required boxes and clicks on ‘Send’ button | Send Successfully | Pass |  |
| 2 | Does not fill up all the required boxes and  clicks on ‘Send’ button | All required boxes must be  filled | Fail |  |
| **Post-condition:** Your mail send Successfully! | | | | |

#### Encrypt File

|  |  |
| --- | --- |
| **Test Case**: 5.2.8 | **Test Case Name:** Encrypt File |
| **System:** Secure Mailbox | **Subsystem:** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Registered user can encrypt their file after login the system. | | | | |
| **Pre-condition:**   * Must be logged in | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | Fill up all the required boxes and clicks on ‘encrypt’ button | Your File is encrypted | Pass |  |
| 2 | Does not fill up all the required boxes and  clicks on ‘encrypt’ button | All required boxes must be  filled | Fail |  |
| **Post-condition:** Successfully your file is encrypted! | | | | |

#### Decrypt File

|  |  |
| --- | --- |
| **Test Case**: 5.2.9 | **Test Case Name:** Decrypt File |
| **System:** Secure Mailbox | **Subsystem:** |
| **Designed by:** Mazharul Islam | **Design Date:** |
| **Executed by:** | **Execution Date:** |
| **Description:** Registered user can encrypt their file. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-condition:**  Database has at least one data to display | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | Attach file and enter a secret key then click on “Decrypt” | Decrypted Successfully | Pass |  |
| **Post-condition:** Your file is decrypted successfully! | | | | |

#### View system notification List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**: 5.2.10 | | **Test Case Name:** View Notification List | | |
| **System:** Secure Mailbox | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Registered user can check their notification list. | | | | |
| **Pre-condition:**  Database has at least one notification to display. | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | User click on Notification List to view list info. | System will display the notification list. | Pass |  |

**Post-condition:** Display all notification

#### Live Chat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**: 5.2.11 | | **Test Case Name:** Live Chat | | |
| **System:** Secure Mailbox | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Registered user should send message each other. | | | | |
| **Pre-condition:**   * User must be logged in * User select another to send text | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | Fill up all the required boxes and clicks on ‘Send’ button | Sent Successfully | Pass |  |
| 2 | Does not fill up all the required boxes and  clicks on ‘Send’ button | All required boxes must be  filled | Fail |  |
| **Post-condition:** Message sent successfully. | | | | |

#### Check message notification

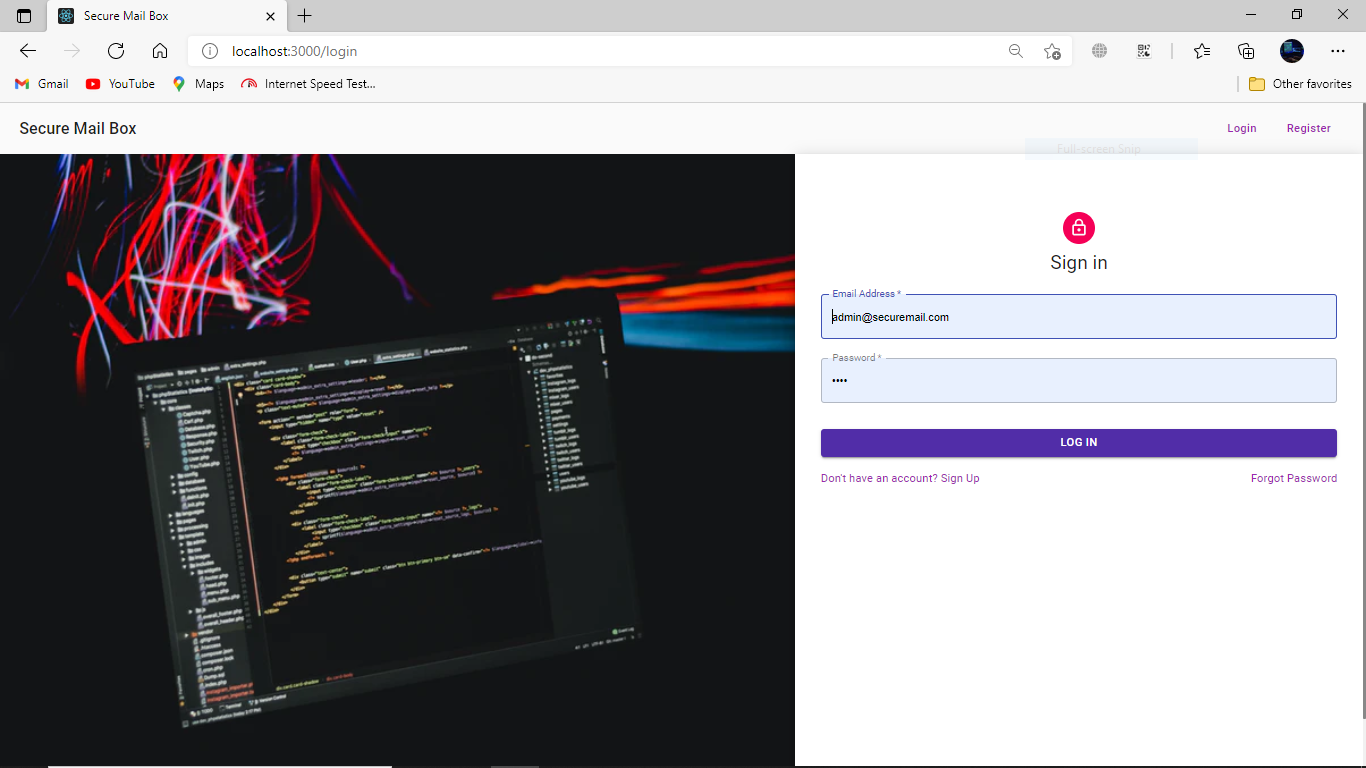
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**: 5.2.12 | | Test Case Name: Check message notification | | |
| **System:** Secure Mailbox | | **Subsystem:** | | |
| **Designed by:** Mazharul Islam | | **Design Date:** | | |
| **Executed by:** | | **Execution Date:** | | |
| **Description:** Registered user can check notification list after their login. | | | | |
| **Pre-condition:**   * Must be logged in * Must have at least a data | | | | |
| **Step** | **Action** | **Response** | **Pass/Fail** | **Comment** |
| 1 | Click on notification icon | Show notification | Pass |  |
| **Post-condition:** Show notification list. | | | | |

**Chapter 6**

**User Manual**

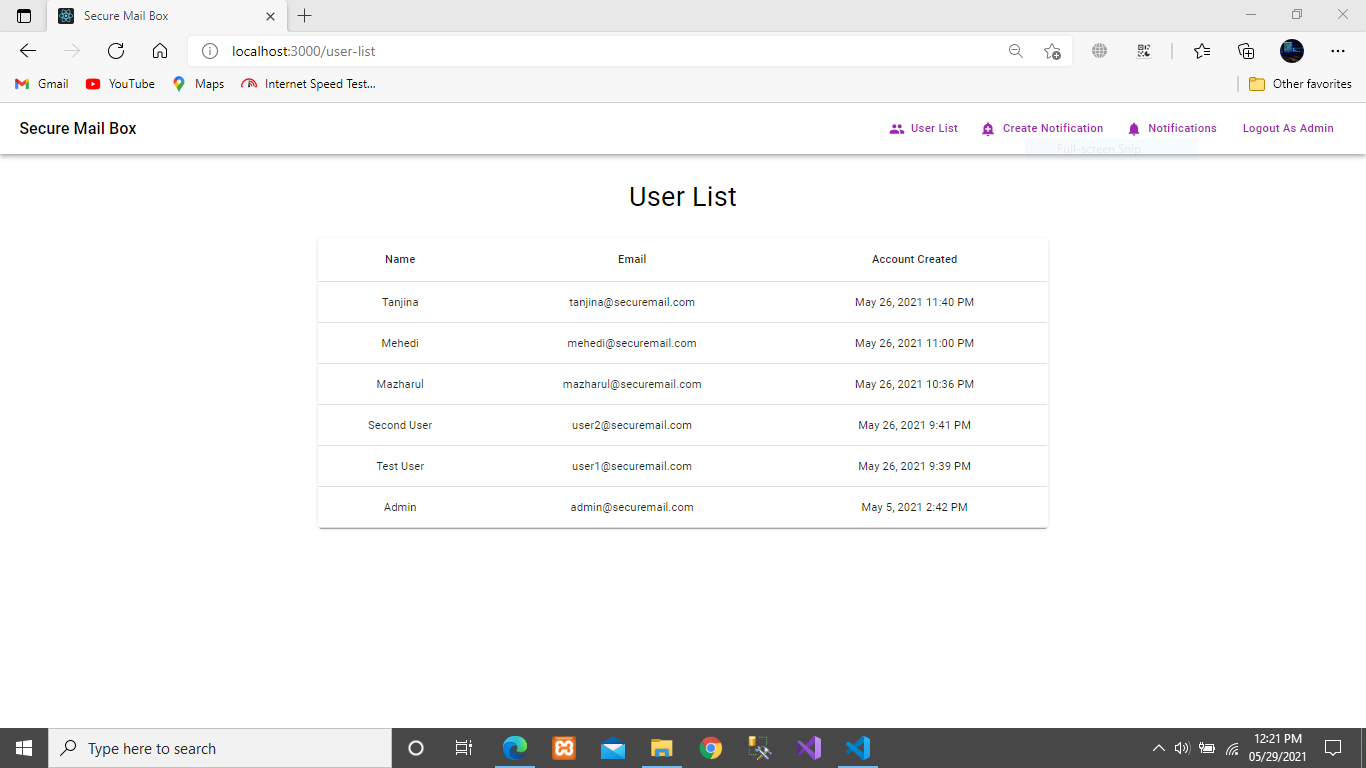
**6.1 Admin**

**6.1.1 Login (Admin)**

****

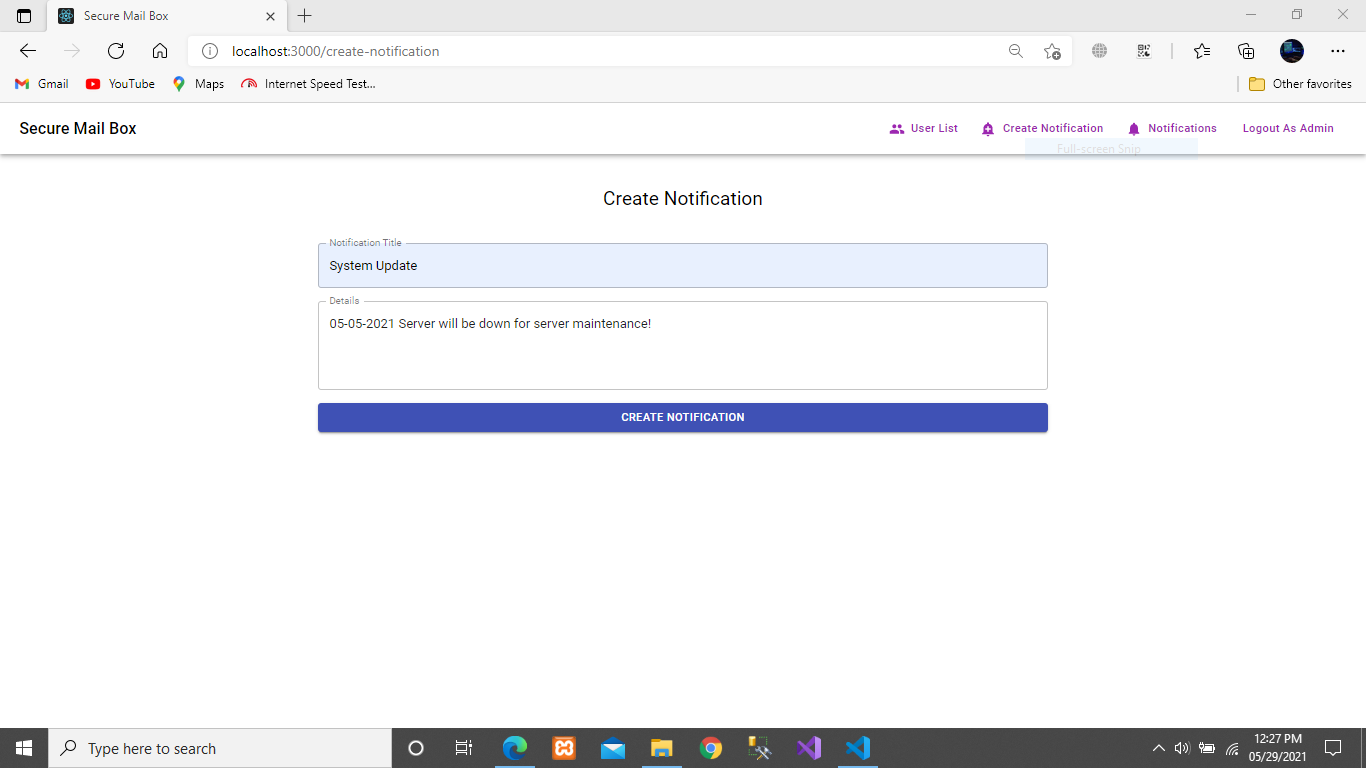
**Figure 6.1: Login**

**6.1.2 Check User List**

****

**Figure 6.2: User List**

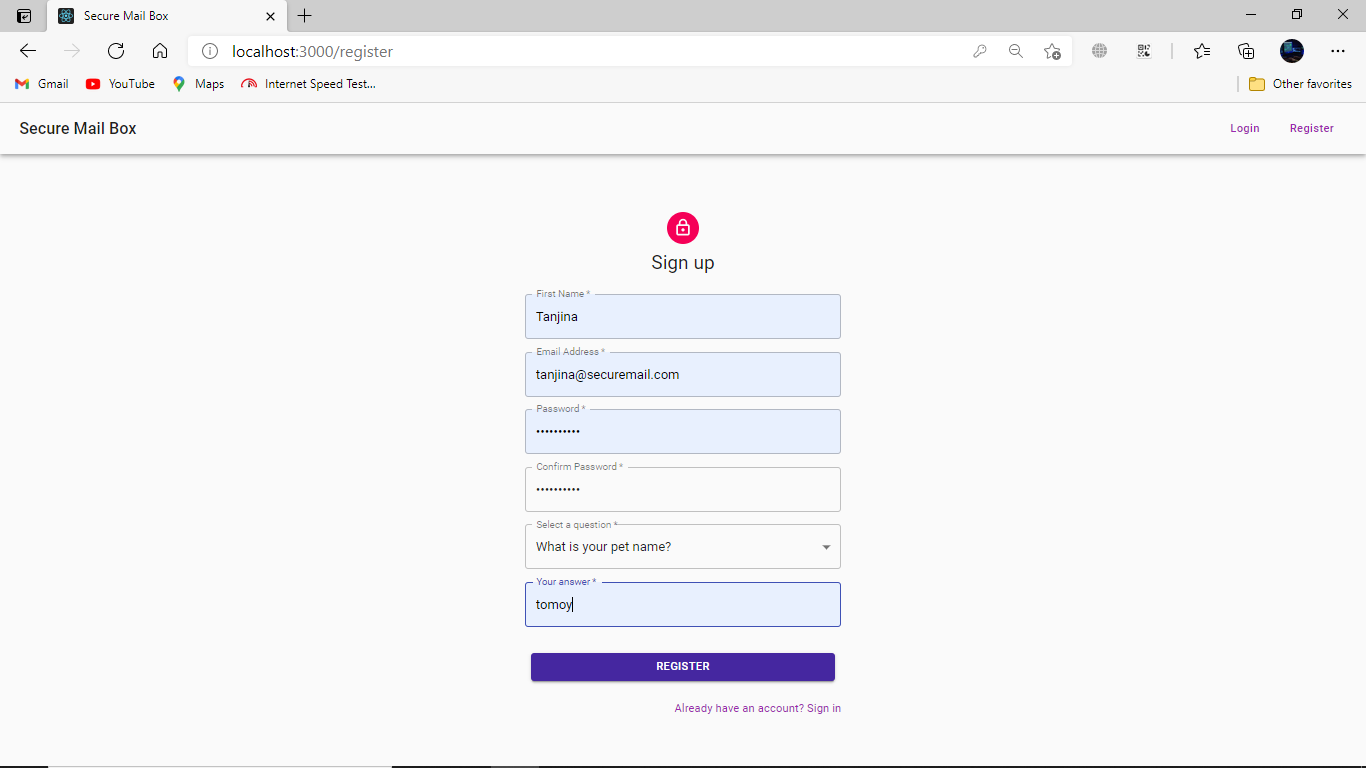
**6.1.3 Create Notification**

****

**Figure 6.3: Create Notification**

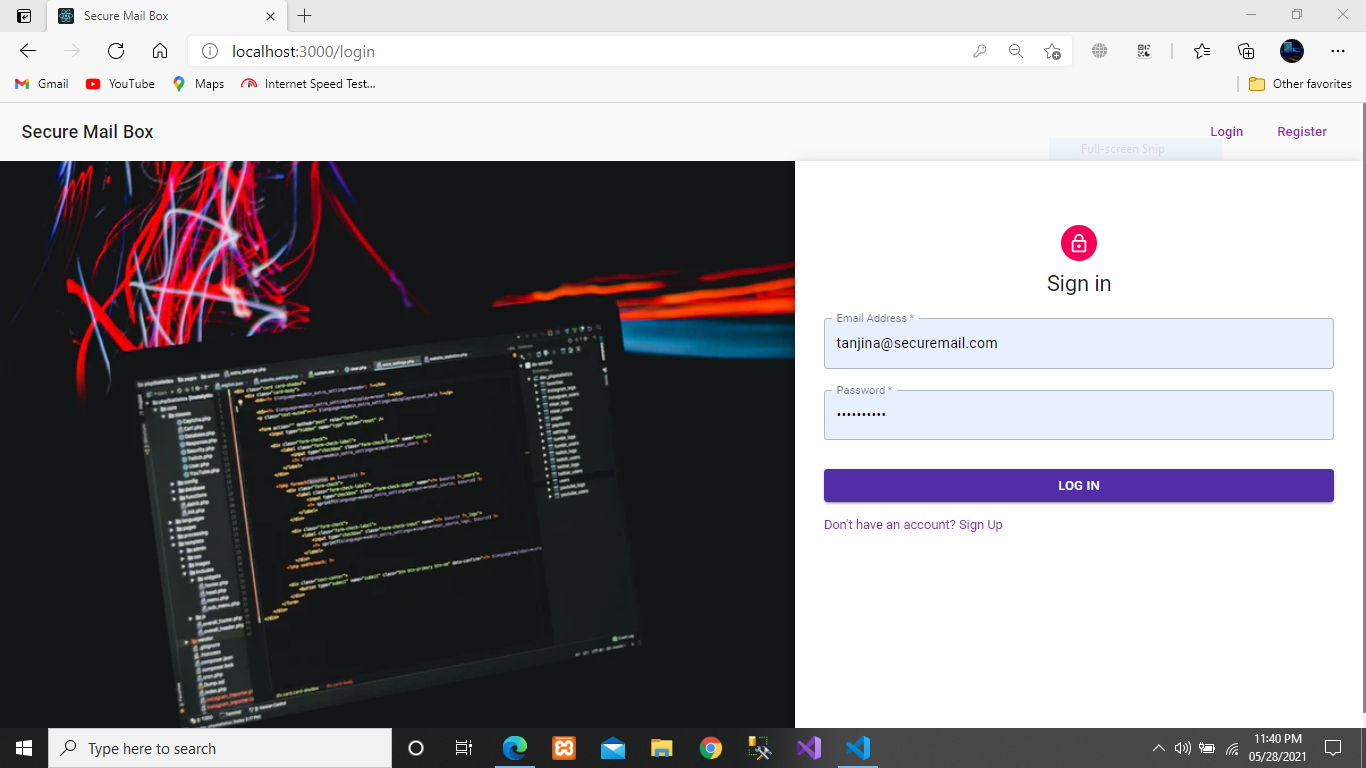
**6.2 User**

**6.2.1 User Registration**

****

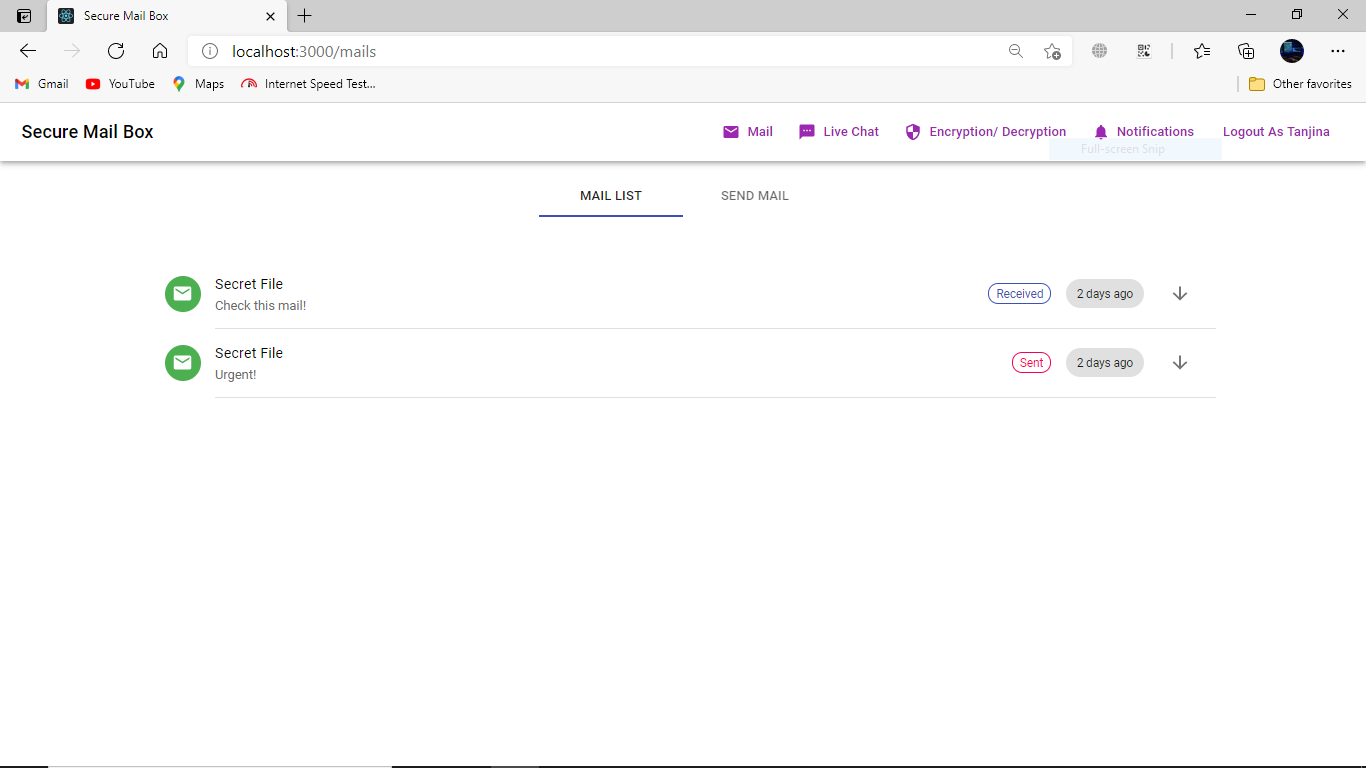
**Figure 6.4: User Registration**

**6.2.2 Login (User)**

****

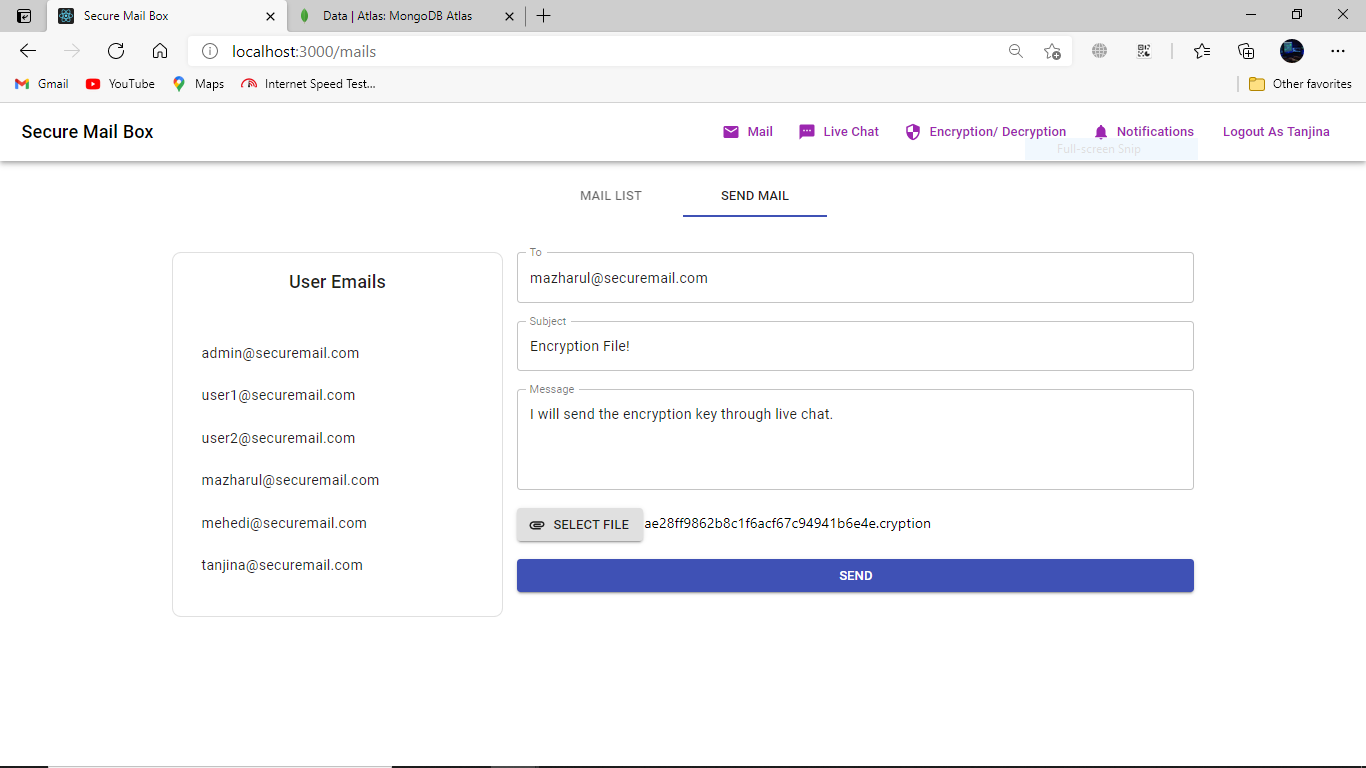
**Figure 6.5: Login**

**6.2.3 User Check Mail List**

****

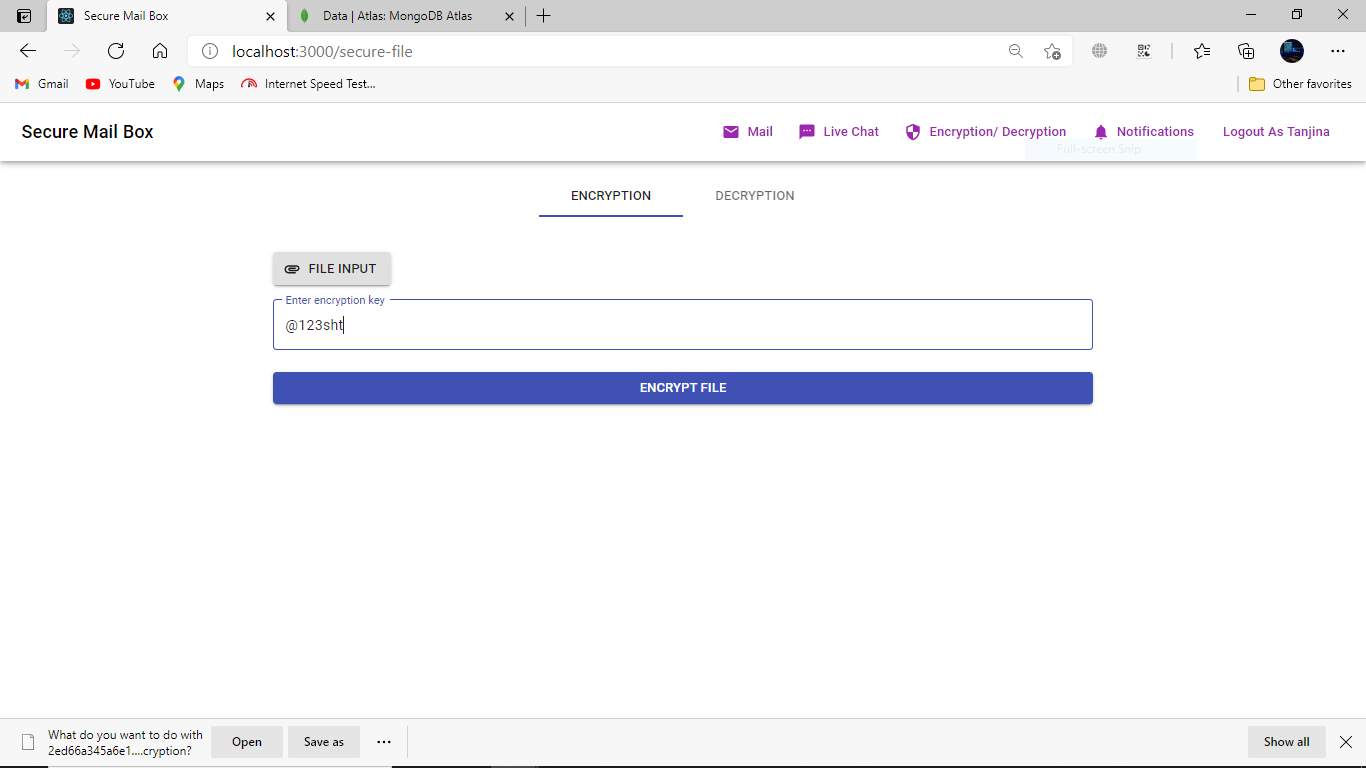
**Figure 6.6 Mail List**

**6.2.4 User Create Mail**

****

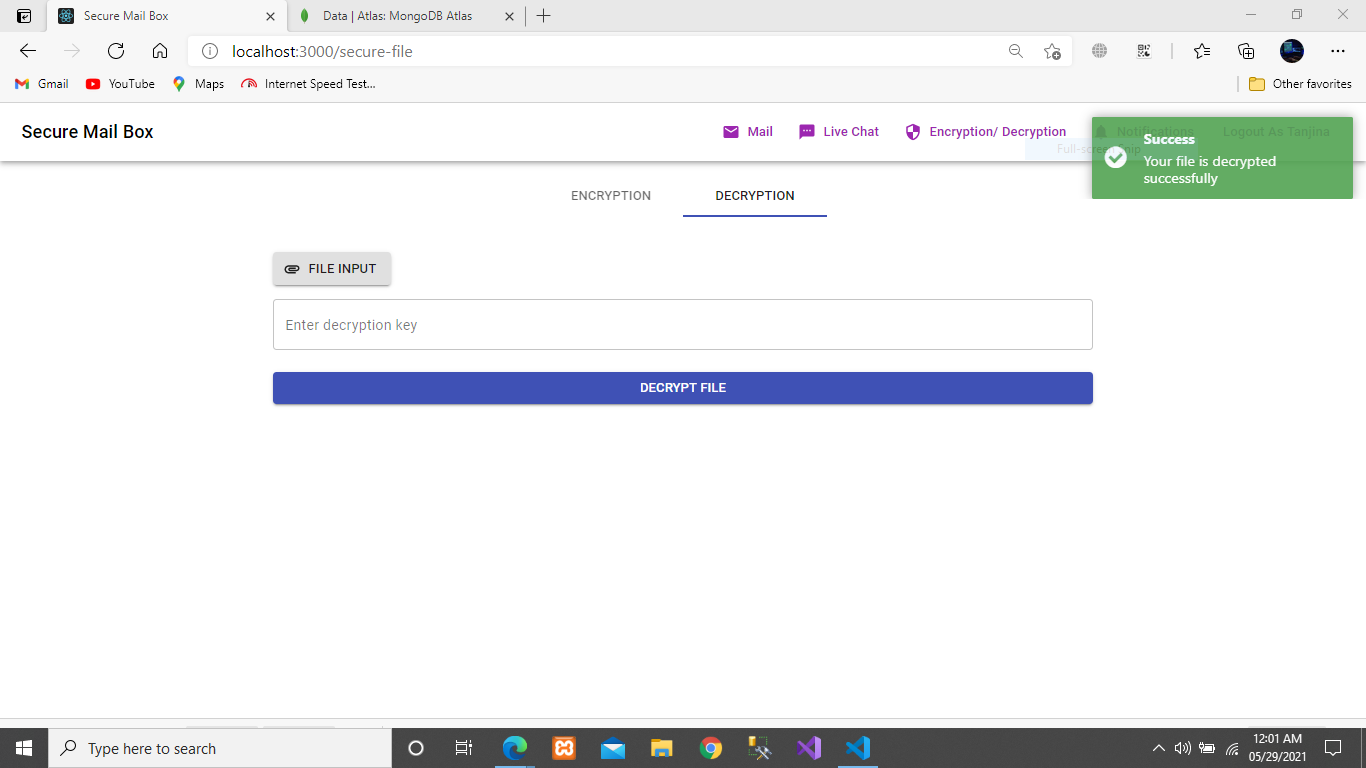
**Figure 6.7 Create Mail**

**6.2.5 Encrypt File**

****

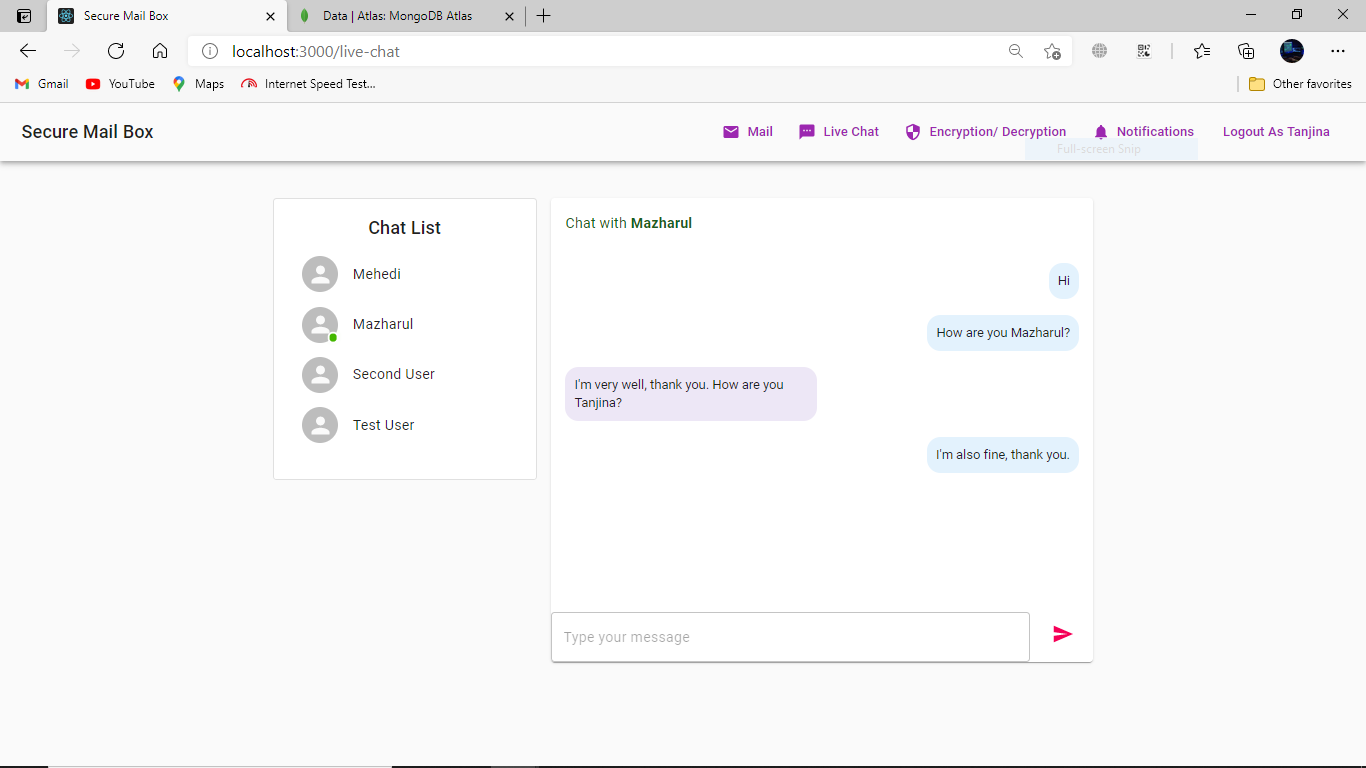
**Figure 6.8: Encrypt File**

**6.2.6 Decrypt File**

****

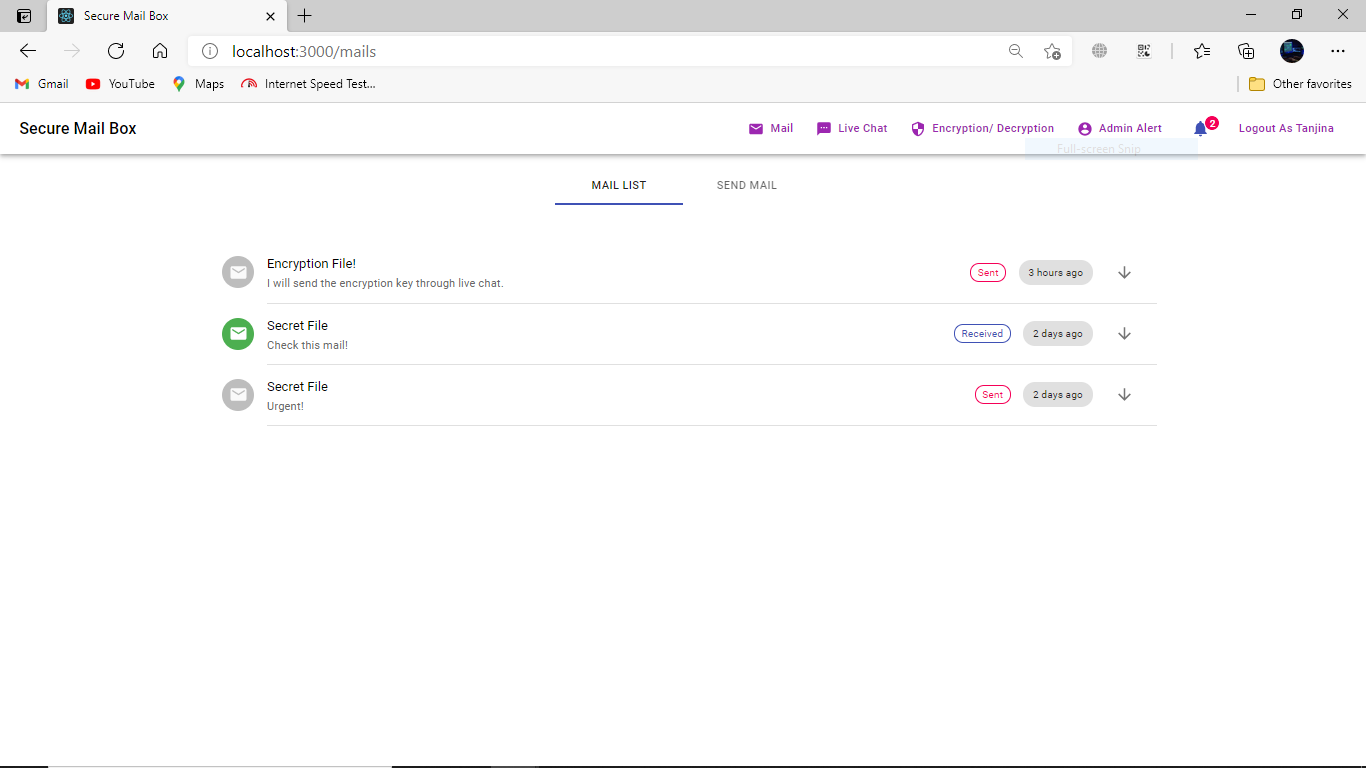
**Figure 6.9: Decrypt File**

**6.2.7 Live Chat**

****

**Figure 6.10: Live Chat**

**6.2.8 Notification List**

****

**Figure 6.11: Check Notification List**

## Chapter 7

### Github Link

### 

Github Link: https://github.com/MAZHARULISLAM11981/SecureMailbox

### Obstacles & Achievements:

When I was started this project implementation in dot net core framework, it was very challenging to me. Then I think about another framework to implement this project, which framework can be easy to implement this project? Then I decided I will implement this project individually Frontend and Backend. That’s why I was learned JavaScript (React.js and Node.js) this project development experience was really surprising to me. Now, I’m very confident to me to develop any kind of project using this project experience. Now, I know how to overcome the obstacles in development life.

### Future Scope: In the future it can be include video call feature and some advance feature below given-

* + - Full system will be encrypted
    - Add Encryption on Transport Layer
    - Each mail can encrypt before send
    - User may to have paying to use advance feature

## References

IEEE Explore:

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7406040&queryText=image%20aes&newsearch=true>

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5734951&queryText=image%20aes&newsearch=true>

Abhinav Asthana, A. K. (2014). *Postman*. Retrieved from Postman The collaboration platform: https:[//www.postman.com/](http://www.postman.com/)

Microsoft Corporation, R. D. (2009, May 27). *Node.js*. Retrieved from node.js org: https://nodejs.org/en/ MongoDB Inc., N. (2009, February 11). *Mongo DB*. Retrieved from Mongo DB the popular database:

https:[//www.mongodb.com/](http://www.mongodb.com/)

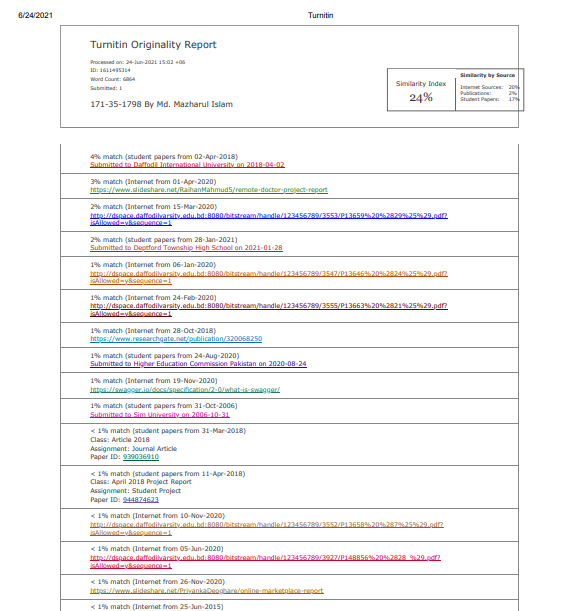
Smart Bear Software, O.-s. s. (n.d.). *Swagger*. Retrieved from Swagger API documentation: https://swagger.io/

Strong loop. (2010, November 16). *Express.js*. Retrieved from Express JS: <https://expressjs.com/>

Wikipedia: [Advanced Encryption Standard process - Wikipedia](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard_process)

Development Guideline: [Cryptographic Standards and Guidelines | CSRC (nist.gov)](https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development)

**Plagiarism Report**

****