



## **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

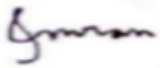
Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

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 its style and contents.

### BOARD OF EXAMINERS



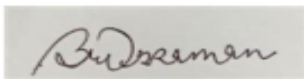
-----  
Dr. Imran Mahmud  
Associate Professor and Head  
Department of Software Engineering  
Daffodil International University

Chairman



-----  
Syeda Sumbul Hossain  
Senior Lecturer  
Department of Software Engineering  
Daffodil International University

Internal Examiner 1



-----  
Khalid Been Badruzzaman Biplob  
Senior Lecturer  
Department of Software Engineering  
Daffodil International University

Internal Examiner 2



-----  
Professor Dr. Mohammed Nasir Uddin  
Department of Computer Science and Engineering  
Jagannath University, Dhaka

External Examiner

## 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

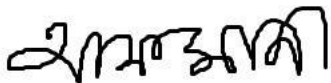
Batch: 22th

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
BOARD OF EXAMINERS.....	ii
DECLARATION.....	iii
ACKNOWLEDGEMENT.....	iv
Table of Contents.....	v
List of Figures.....	ix
Chapter 1.....	1
Introduction.....	1
1.1 Project Overview.....	1
1.2 Project Purpose.....	1
1.3 Project Goal.....	1
1.4 Benefits and Beneficiaries.....	2
1.5 Stakeholders.....	2
1.5.1 Admin.....	2
1.5.2 User.....	2
Chapter 2.....	3
Software Requirement Specification (SRS).....	3
2.1 Business requirements.....	3
2.2 SDLC model.....	4
2.3 Functional requirements.....	5
2.3.1 Create User account.....	6
2.3.2 Admin Id Verification.....	7
2.3.3 Show User List.....	7
2.3.4 Create Notification.....	7
2.3.5 Create Mail.....	8
2.3.6 Encrypt/Decrypt Mail.....	8
2.3.7 Live Chat Between User.....	8
2.3.8 View Mail List.....	9
2.3.9 View Notification List.....	9
2.4 Non-Functional requirements.....	9

2.4.1 Performance .....	10
2.4.2 User sensitive data will dissolve from others .....	10
2.4.3 Reliability and Availability .....	10
2.4.4 Hardware .....	11
2.4.5 Response time .....	11
2.4.6 Fault Tolerance .....	11
2.4.7 Maintenance .....	12
Chapter 3 .....	13
System Analysis .....	13
3.1 Use Case Diagram .....	13
3.1.1 User .....	13
3.1.2 Admin .....	14
3.2 Use Case Description .....	15
3.2.1 Registration with token based authentication for User .....	15
3.2.2 Login for Registered Admin .....	16
3.2.3 Check User List .....	17
3.2.4 Create Notification .....	18
3.2.5 Log Out of All Registered Admin .....	21
3.2.6 Registration with token based authentication for User .....	22
3.2.7 Login for Registered User .....	23
3.2.8 Create mail .....	25
3.2.9 Encrypt/Decrypt File .....	26
3.2.10 Live Chat .....	27
3.2.11 View Mail List .....	28
3.2.12 View Notification List .....	29
3.2.13 View user List .....	30
3.2.14 Log Out for Registered Patient .....	29
3.3 Activity Diagram .....	31
3.3.1 User Activity Diagram .....	31
3.3.1.1 Admin Activity Diagram .....	31
3.3.2 User .....	36
3.4 Sequence Diagram .....	39
3.4.1 Admin .....	39
3.4.2 User .....	42

Chapter 4 .....	47
System Design Specification (SDS) .....	47
4.1 Development Tools and Technologies .....	47
4.1.1 API Development Technology .....	47
4.1.2 API Development Tools .....	49
4.2 Class Diagram .....	50
4.3 ER Diagram .....	51
Chapter 5 .....	52
System Test .....	52
5.1 Testing Features .....	52
5.1.1 User Features .....	52
5.1.1 Admin Features .....	53
5.2 Test Cases .....	54
5.2.1 Account Registration (Admin) .....	54
5.2.2 Account Registration (User) .....	55
5.2.3 Login (Admin) .....	56
5.2.4 Login (User) .....	56
5.2.5 View User List .....	57
5.2.6 Create Notification .....	58
5.2.7 Create mail .....	59
5.2.8 Encrypt File .....	59
5.2.9 Decrypt File .....	60
5.2.10 View Mail List .....	61
5.2.11 Check Notification .....	62
5.2.12 Live Chat .....	63
5.2.13 View Notification List .....	63
Chapter 6 .....	65
6.1: User Manual .....	65
6.1.1: Admin profile Login .....	69
6.1.2: Check user list .....	67
6.1.3: Create Notification .....	68
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
7.1 Github Link.....	78
7.2 Obstacles & Achievements.....	78
7.3 Future Scope.....	78
References.....	80
Plagiarism Report.....	81



## List of Figures

Figure 2. 1: SDLC model (Agile).....	13
Figure 3. 1: Use Case Diagram for User .....	22
Figure 3. 1: Use Case Diagram for Admin .....	22
Figure 3. 2: Activity Diagram for User .....	37
Figure 3. 3: Activity Diagram for Admin.....	38
Figure 3. 4: Sequence diagram for user Registration .....	39
Figure 3. 5: Sequence diagram for user Login .....	40
Figure 3. 6: Sequence diagram for create mail.....	41
Figure 3. 7: Sequence diagram for Live Chat.....	42
Figure 3. 8: Sequence diagram for encrypt file .....	43
Figure 3. 9: Sequence diagram for decrypt file .....	44
Figure 3. 10: Sequence Diagram for Admin Login .....	45
Figure 3. 11: Sequence Diagram for Create Notification.....	46
Figure 3. 12: Admin view user list .....	47
Figure 4. 1: Class diagram.....	53
Figure 4. 2: ER diagram .....	54
Figure 6. 1: Admin profile Login.....	69
Figure 6. 2: Check user list.....	67
Figure 6. 3: Create Notification.....	68
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**

### **1.1 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.

### **1.2 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.

### **1.3 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.

## **1.4 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.

## **1.5 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.

### **1.5.1 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.

### **1.5.2 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)

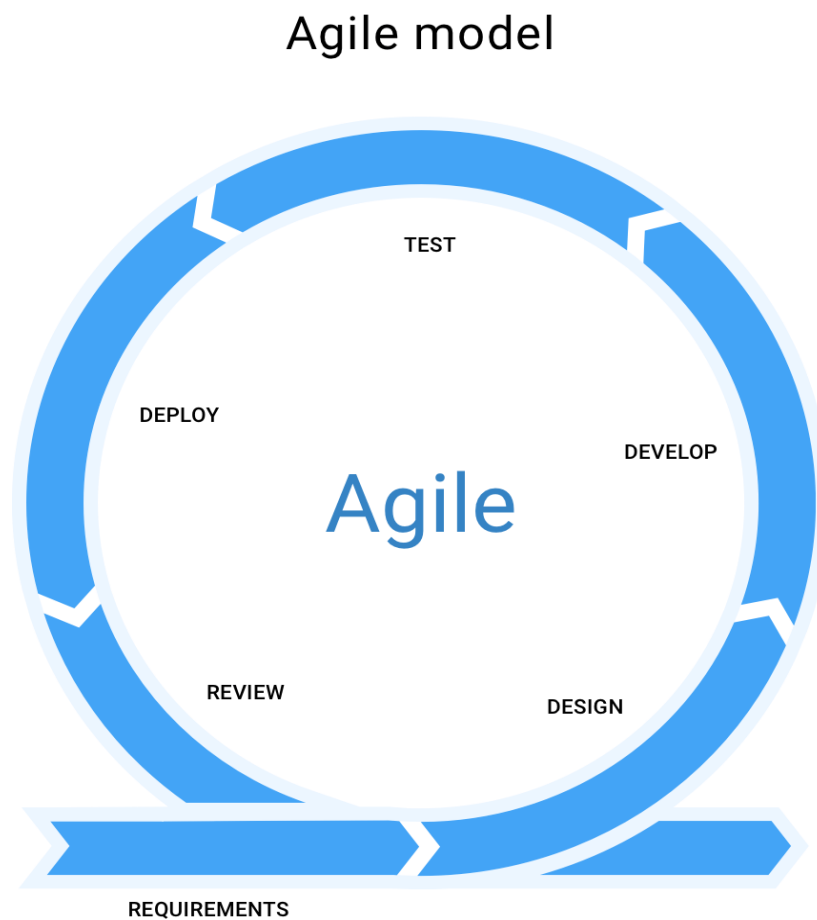
In this chapter we specified about the business requirements of this platform, which SDLC we use, functional and non-functional requirements.

#### 2.1 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.	

## 2.2 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)*

### 2.3 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.

<b>FR ID</b>	<b>Description</b>
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

### 2.3.1 User account

<b>FR 01</b>	Create an user account
<b>Description</b>	Unregistered users must complete their registration before accessing system functionality.
<b>Stakeholder</b>	Admin, User

### 2.3.2 User Id Verification

<b>FR 02</b>	User Id verification
<b>Description</b>	System verify user account
<b>Stakeholder</b>	User

### 2.3.3 Show User List

<b>FR 03</b>	Show User List
<b>Description</b>	Admin can check user list.
<b>Stakeholder</b>	Admin, User

### 2.3.4 Create Notification

<b>FR 04</b>	Create Notification
<b>Description</b>	Admin can create notification to send system update or user support.
<b>Stakeholder</b>	Admin



### 2.3.5 Create Mail

<b>FR 05</b>	Create Mail
<b>Description</b>	By using an online create mail method user can create a mail to send another user.
<b>Stakeholder</b>	User

### 2.3.6 Encrypt and Decrypt File

<b>FR 06</b>	Encrypt and Decrypt file
<b>Description</b>	User can encrypt/decrypt any extensional file through the encryption/decryption feature.
<b>Stakeholder</b>	User

### 2.3.7 Live chat between user

<b>FR 07</b>	Live Chat between user
<b>Description</b>	All user can use live chat feature to communicate with real-time medium.
<b>Stakeholder</b>	User

### 2.3.8 View Mail List(Send/Receive)

<b>FR 08</b>	View mail List
<b>Description</b>	User can check mail list, Which mail they are receive and send by them.
<b>Stakeholder</b>	User

### 2.3.9 View Notification List

<b>FR 09</b>	View Notification List
<b>Description</b>	In this feature user can receive notification from admin and they can check the notification list anytime.
<b>Stakeholder</b>	User

### 2.4 Non-Functional requirements

<b>NFR ID</b>	<b>Description</b>
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

### 2.4.1 Performance

<b>NFR 01</b>	Request and response will build within a second
<b>Description</b>	This is a Web Based Application and the responses are fetch from API.
<b>Stakeholder</b>	Admin, User

### 2.4.2 User sensitive data will disappear from others

<b>NFR 02</b>	User sensitive data will dissolve from others
<b>Description</b>	System will be designed as per requirement. So that, It's not harm for user.
<b>Stakeholder</b>	Admin, User

### 2.4.3 Reliability and Availability

<b>NFR 03</b>	The system must be available 24x7
<b>Description</b>	The system will available 24 hours always.
	Expect the system update/upgrade.
<b>Stakeholder</b>	N/A

#### 2.4.4 Hardware

<b>NFR 04</b>	Hardware
<b>Description</b>	User can reach this application from Mobile and Pc.
<b>Stakeholder</b>	N/A

#### 2.4.5 Response time

<b>NFR 05</b>	Response time
<b>Description</b>	As much as fast the internet speed.
<b>Stakeholder</b>	N/A

#### 2.4.6 Fault Tolerance

<b>NFR 06</b>	Always updated the system
<b>Description</b>	If any problem occurs, then admin will create notification to send all user about system update
<b>Stakeholder</b>	Admin

### 2.4.7 Maintenance

<b>NFR 07</b>	The system helps to update any Information in any time
<b>Description</b>	The admin can changes or updates any information in any situation from database.
<b>Stakeholder</b>	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.

#### 3.1 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.

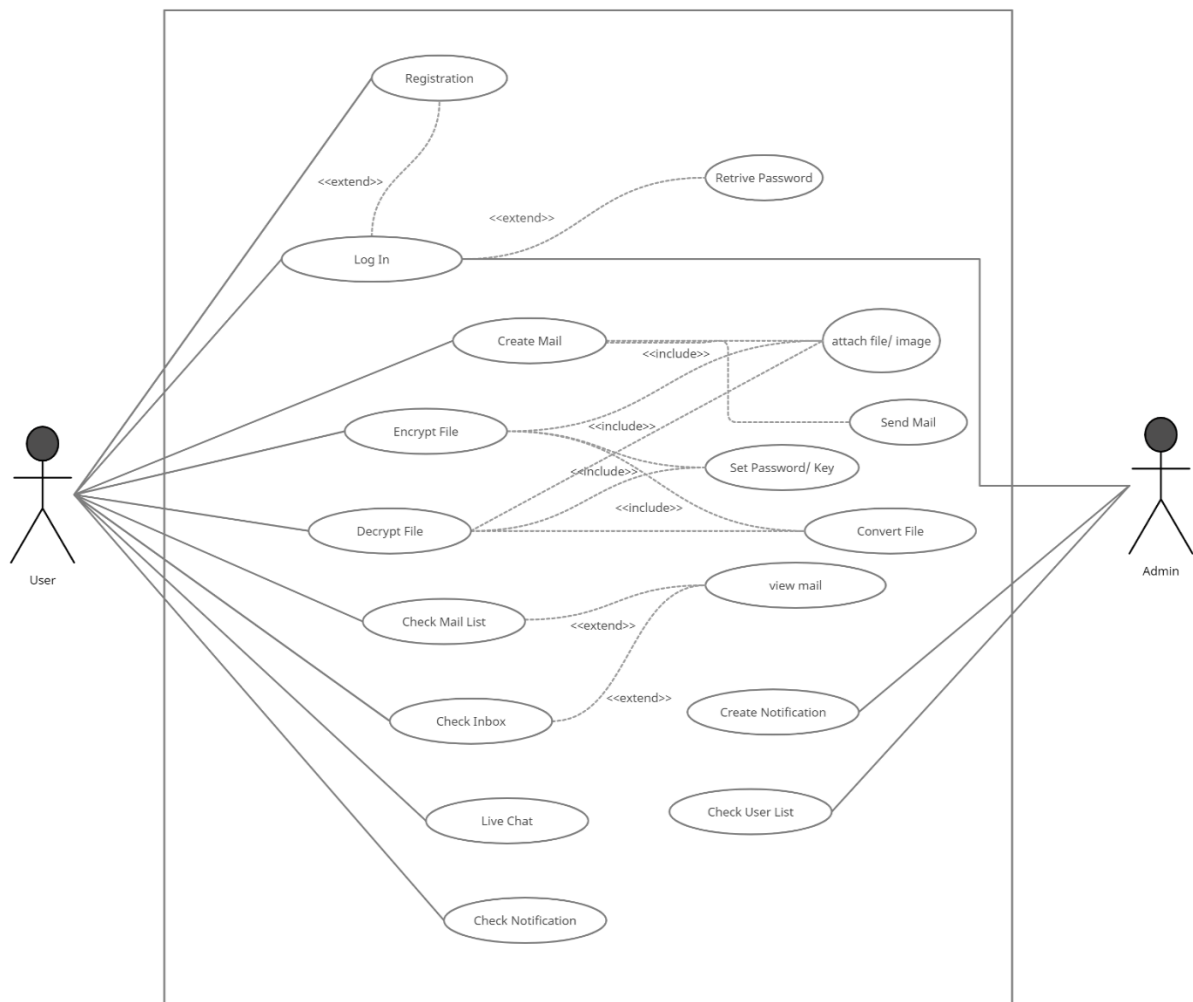


Figure 3. 1: Use Case Diagram

### 3.2 Use Case Description

#### 3.2.1 Registration with token based authentication for User

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

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

### 3.2.2 Login for Registered User

<b>Use Case</b>	Login for Registered User	
<b>Goal</b>	User will get access to the system.	
<b>Predictions</b>	The user can login with email and password.	
<b>Success end condition</b>	The user gets access to the system.	
<b>Failed end condition</b>	Has to get the authorization again to have accessing credentials.	
<b>Primary actors:</b>	User and Admin	
<b>Secondary actors:</b>	Admin	
<b>Trigger</b>	System is accessed by the user.	
<b>Description</b>	<b>Step</b>	<b>Action</b>
	1	Input email and password.
	2	System gives access.
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	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.

### 3.2.3 Check User List

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

<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	User may not receive the notification.
	2	
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirement</b>
	1	Admin can support to user serially.

### 3.2.4 Create Notification

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

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

### 3.2.5 Create Mail

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

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

### 3.2.6 Encrypt/Decrypt Mail

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

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

### 3.2.7 Log Out for Registered Admin

<b>Use Case</b>	Log Out for Registered Admin
<b>Goal</b>	Admin can log out of their account
<b>Predictions</b>	Admin is logged into the system

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

### 3.2.8 Registration with token based authentication for User

<b>Use Case</b>	Registration with token based authentication
<b>Goal</b>	User will be able to login into the system.
<b>Predictions</b>	The user has to fill up the registration form.

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

### 3.2.9 Login for Registered User

<b>Use Case</b>	Login for Registered User
<b>Goal</b>	User will get access to the system.

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



### 3.2.10 Live Chat between user

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

### 3.2.11 View mail list

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

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

### 3.2.12 View notification list

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

	1	System may crash because of the internet.
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirement</b>
	1	The user may get notify from admin.

### 3.2.13 Log Out for Registered User

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

<b>Quality Requirements</b>	<b>Step</b>	<b>Requirement</b>
	1	N/A

### 3.3 Activity Diagram

#### 3.3.1 User

##### 3.3.1.1 Activity Diagram for User

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

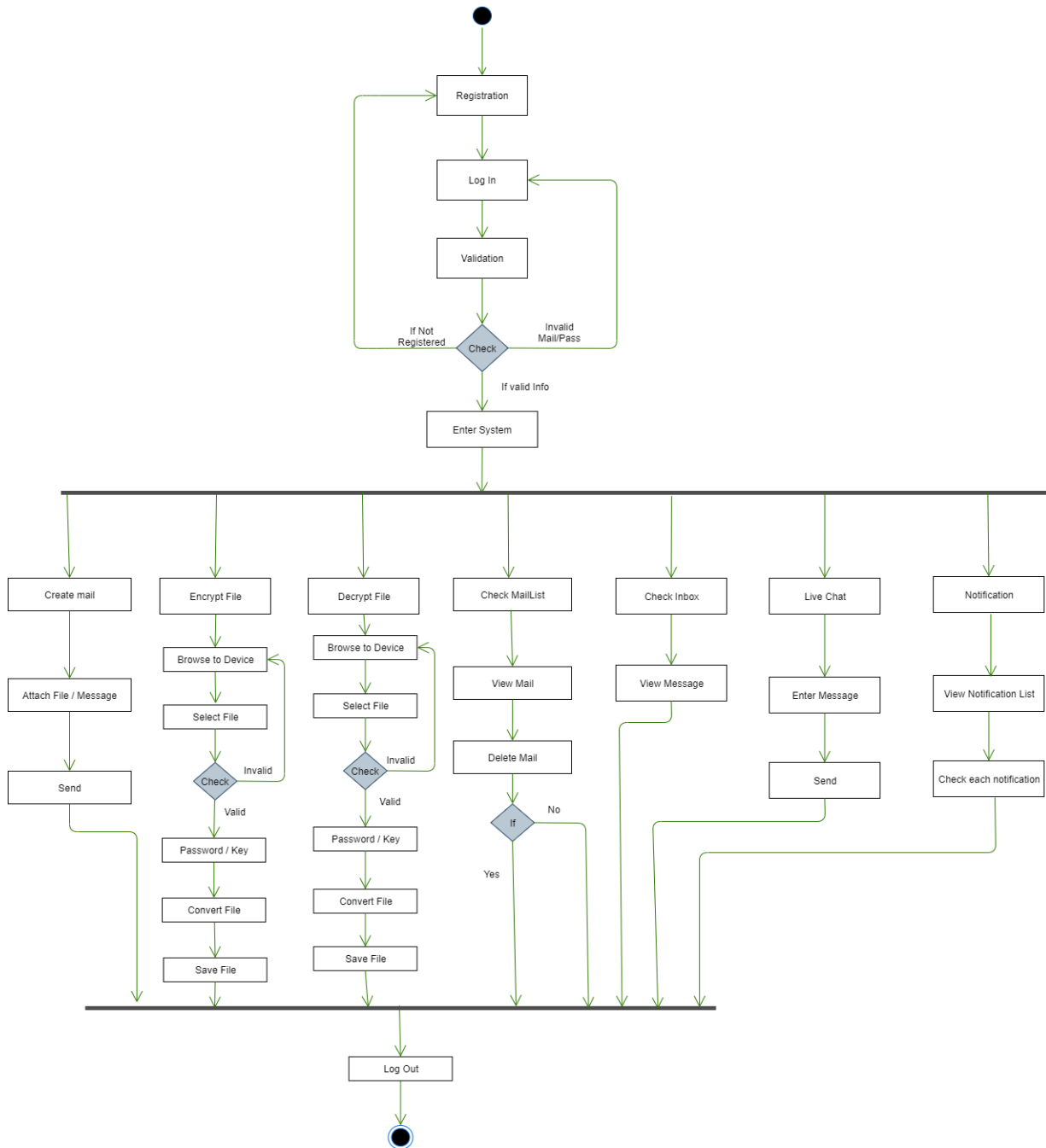


Figure 3. 2: Activity Diagram for User

### 3.3.2 Admin

#### 3.3.2.1 Activity Diagram for Admin

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

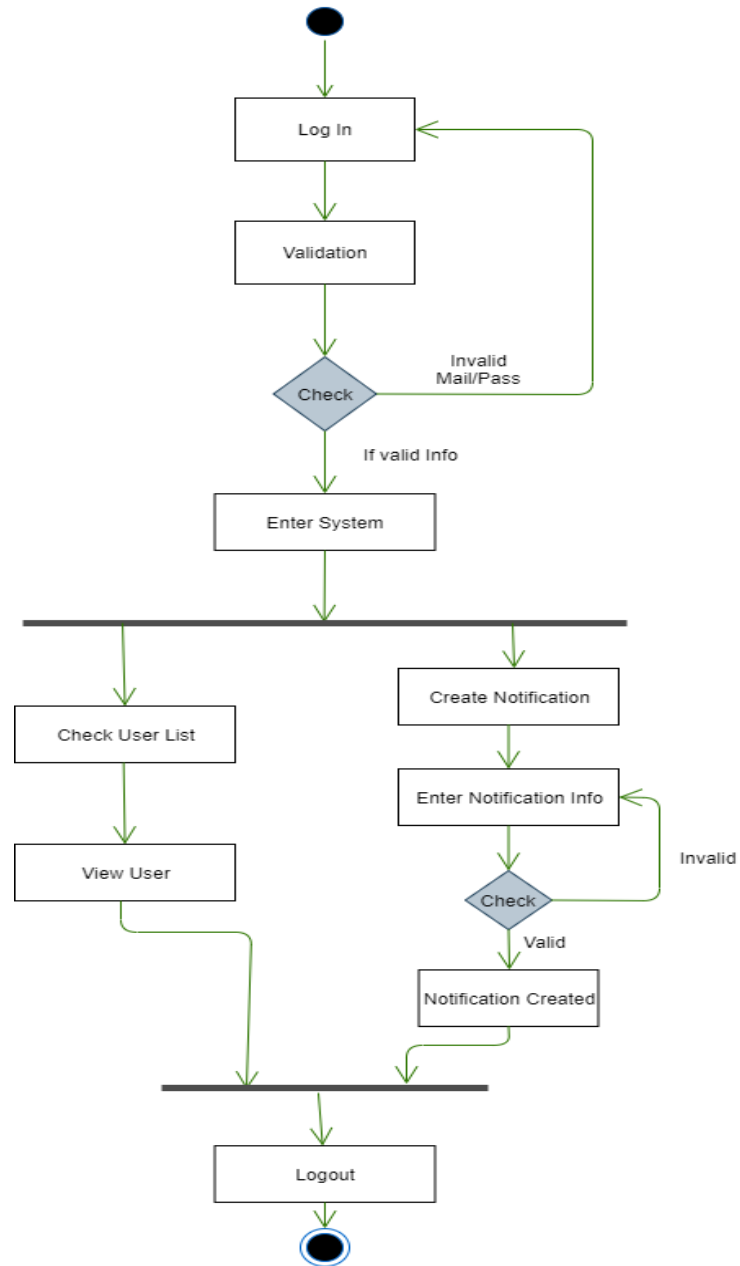


Figure 3. 3: Activity Diagram for Admin



### 3.4 Sequence Diagram

#### 3.4.1 User

##### 3.4.1.1 Sequence Diagram for User Registration

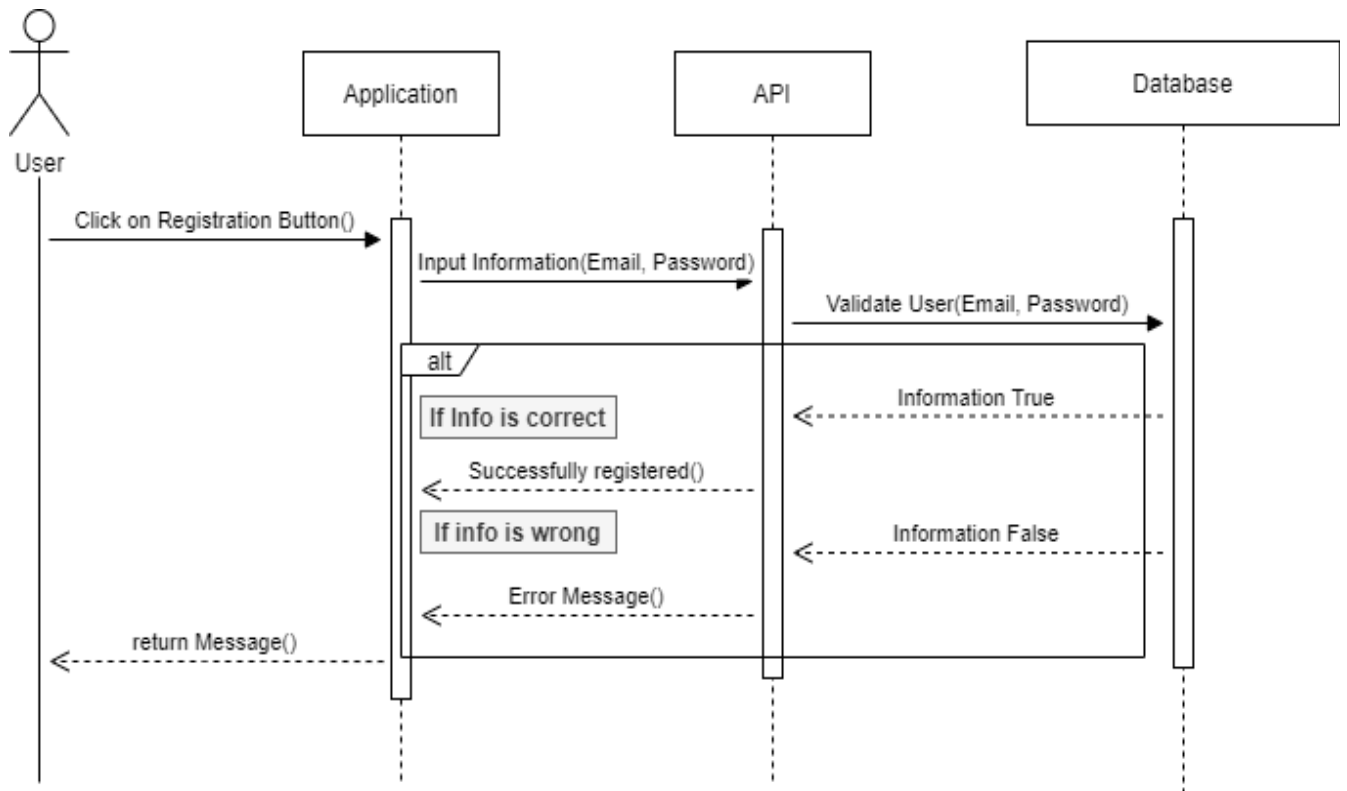


Figure 3.4: Sequence Diagram for Registration

### 3.4.1.2 Sequence Diagram for User Login

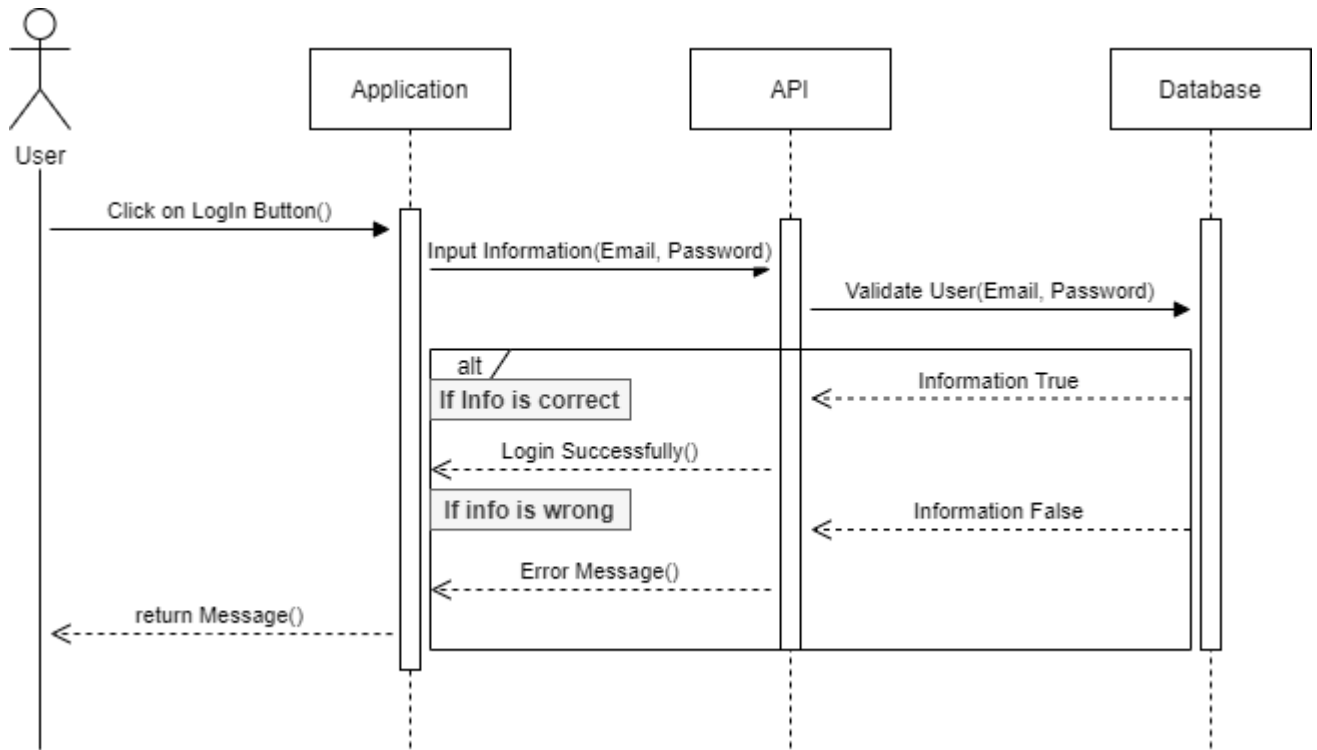


Figure 3.5: Sequence Diagram for Login

### 3.4.1.3 Create Mail

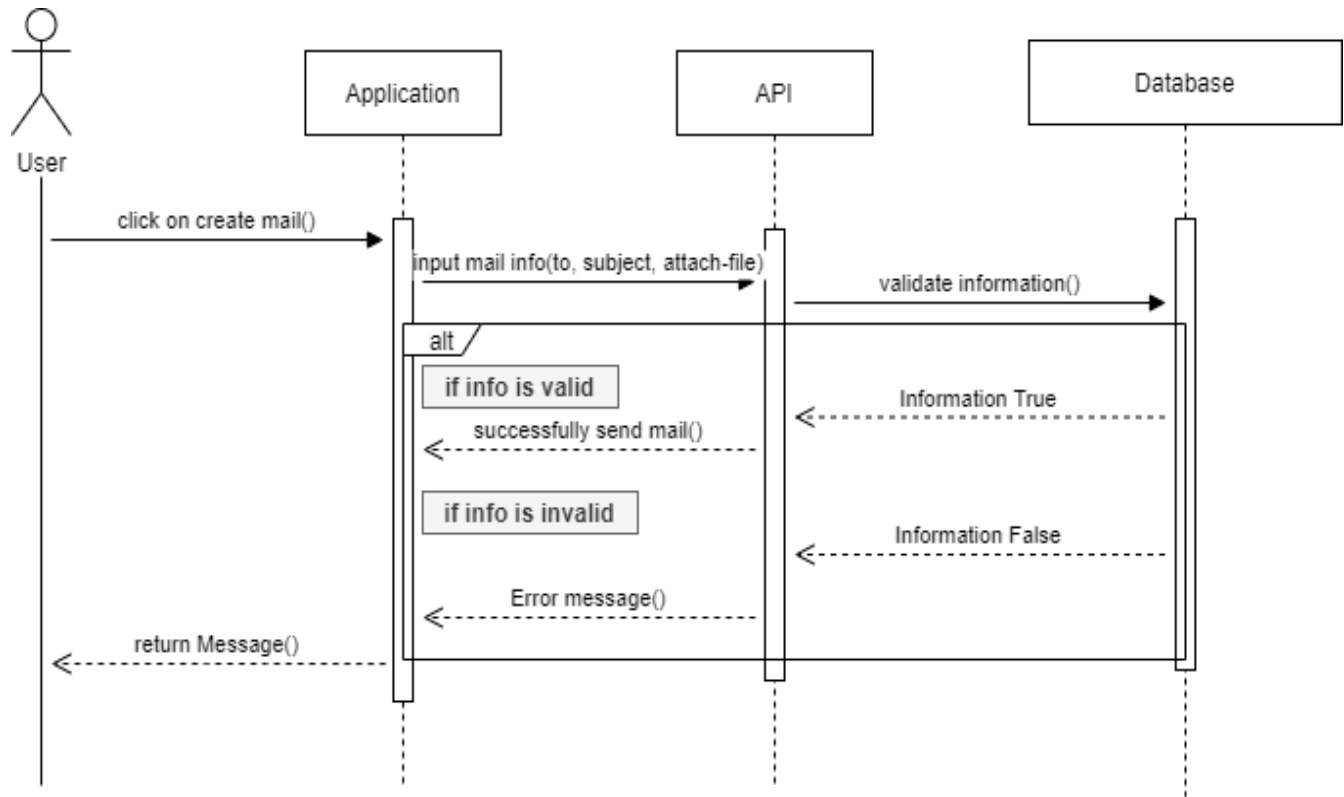


Figure 3.6: Create mail

### 3.4.1.4 Live Chat between user

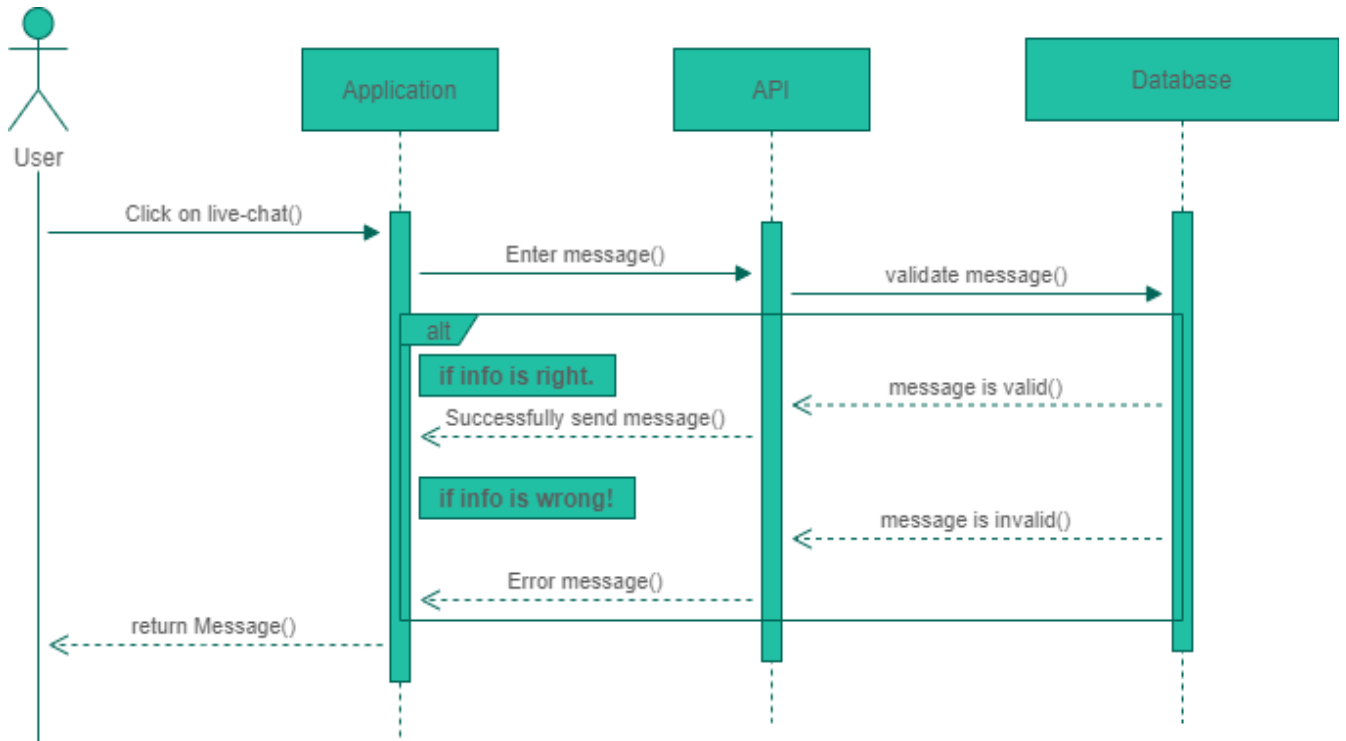


Figure 3.7: Live Chat

### 3.4.1.5 Sequence Diagram for user file encryption

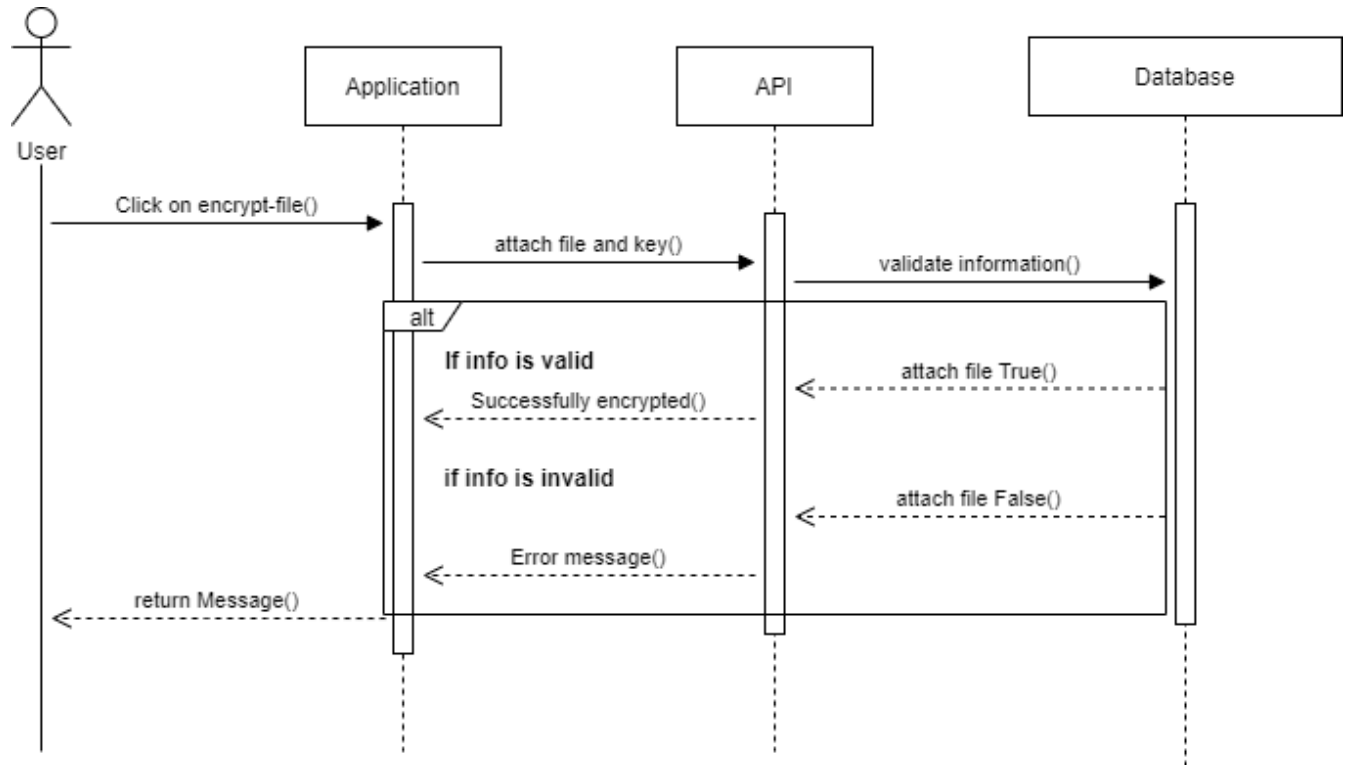


Figure 3.8: Sequence Diagram for user file encryption

### 3.4.1.6 Sequence Diagram for user file decryption

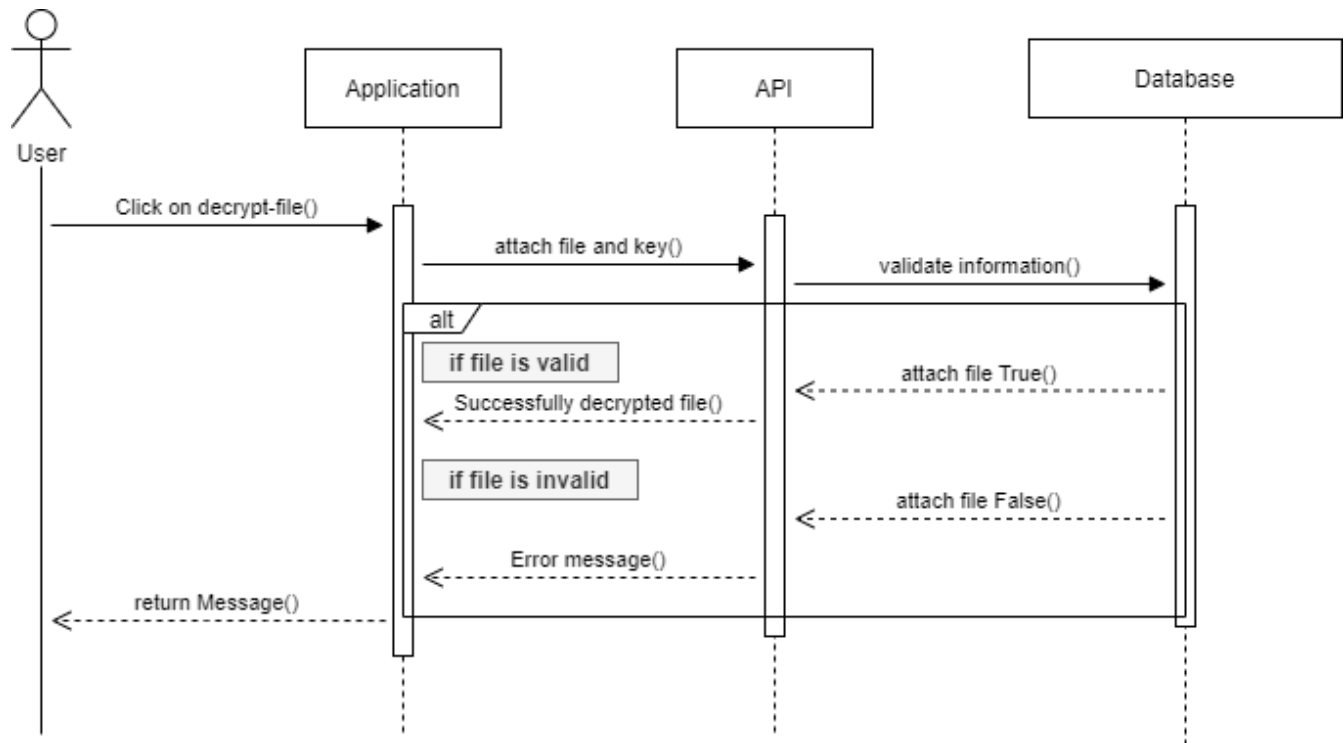


Figure 3.9: Decrypt file

### 3.4.2 Admin

#### 3.4.2.1 Admin Login Sequence Diagram

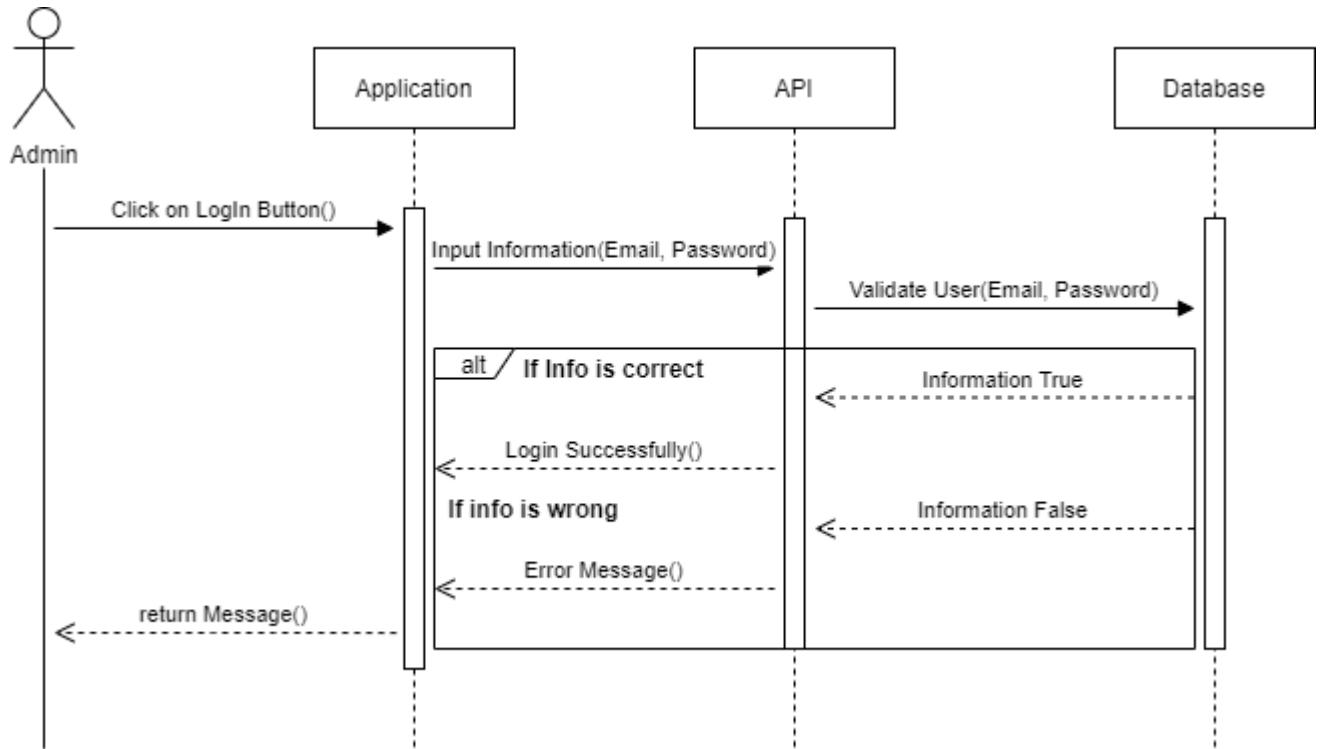


Figure 3.10: Admin Login

### 3.4.2.2 Create Notification

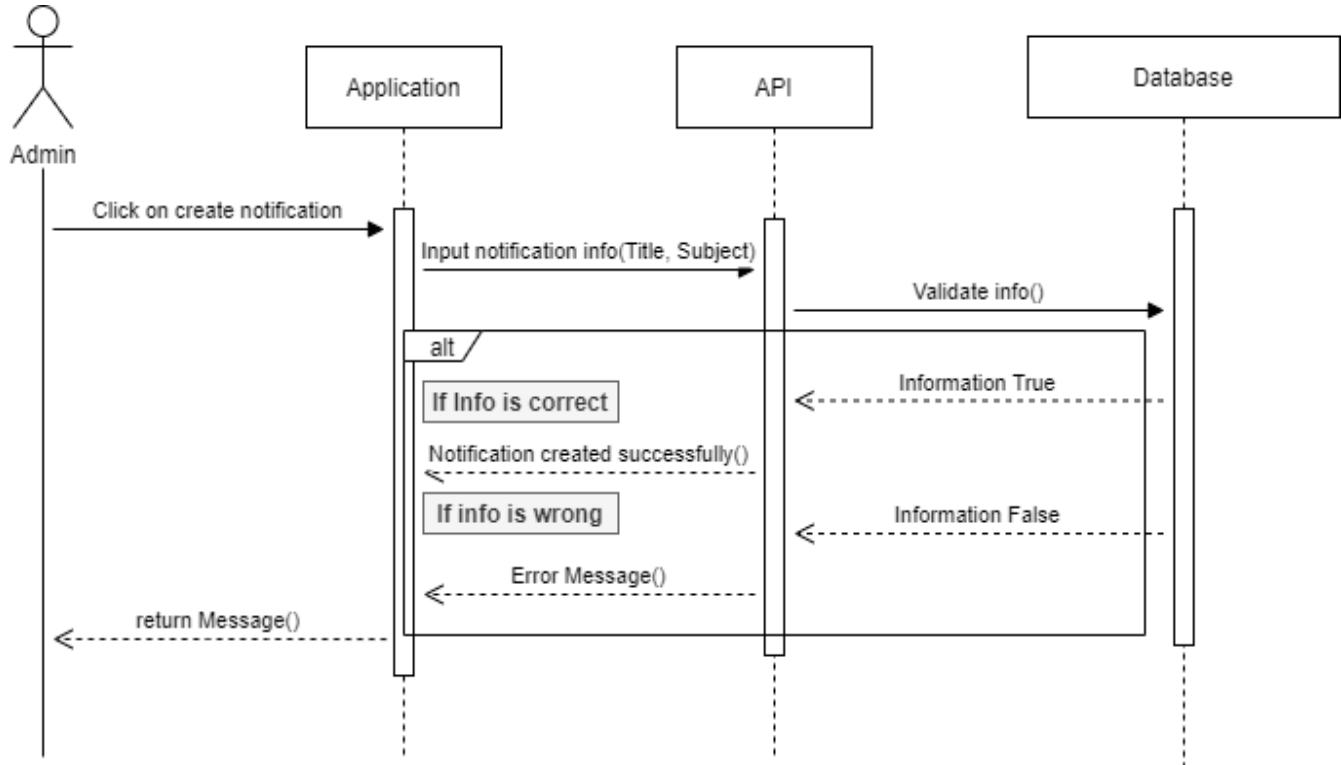


Figure 3. 11: Create and send notification



### 3.4.2.3 View user list

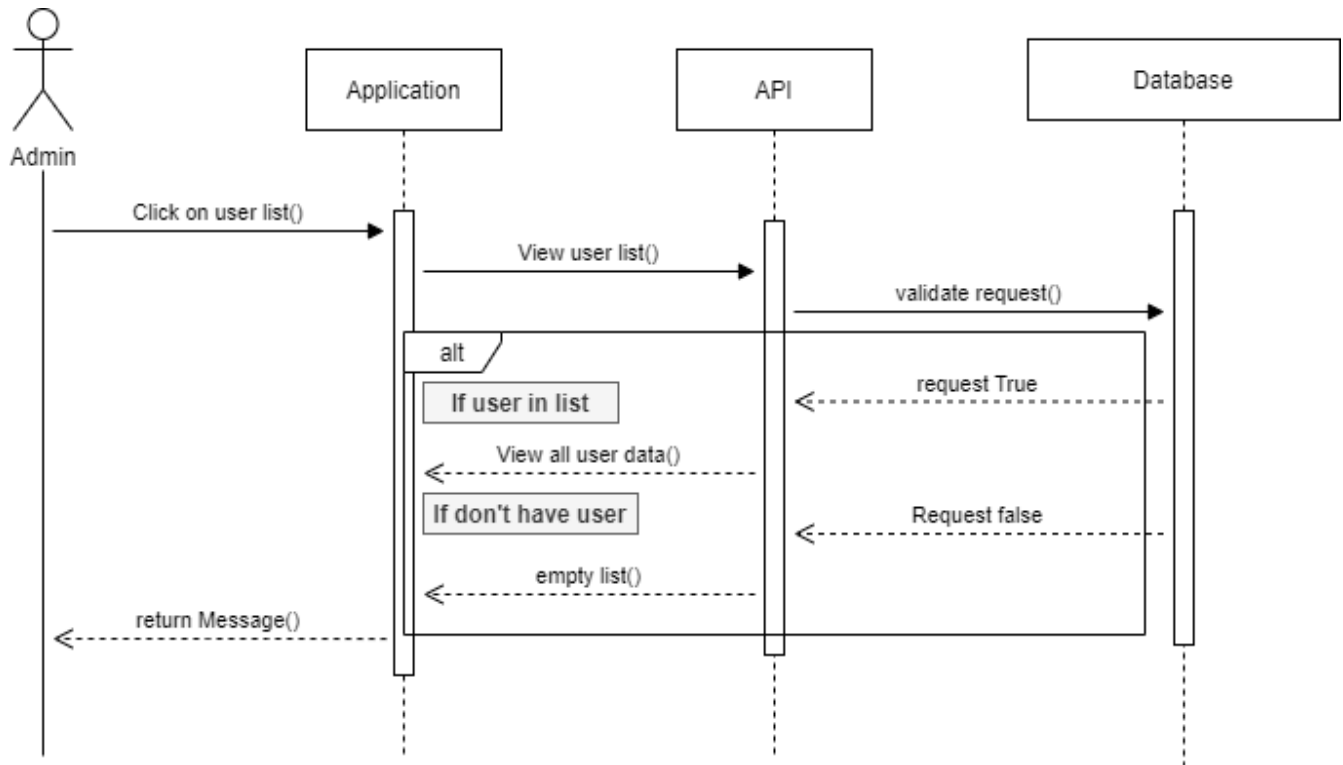


Figure 3. 12: View User List

## 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.

#### 4.1 Development Tools and Technologies

##### 4.1.1 API Development Technology

###### 4.1.1.1 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.

###### 4.1.1.2 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's V8 JavaScript engine.

###### 4.1.1.3 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**.

###### 4.1.1.4 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.

#### **4.1.1.5 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.

#### **4.1.1.6 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

#### **4.1.1.7 Jest**

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

#### **4.1.1.8 No-demon**

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

#### **4.1.1.9 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.

## **4.1.2 API Development Tools**

### **4.1.2.1 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.

### **4.1.2.2 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.

### **4.1.2.3 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.

### **4.1.2.4 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**.

### **4.1.2.5 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.

### **4.1.2.6 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.

## 4.2 Class Diagram

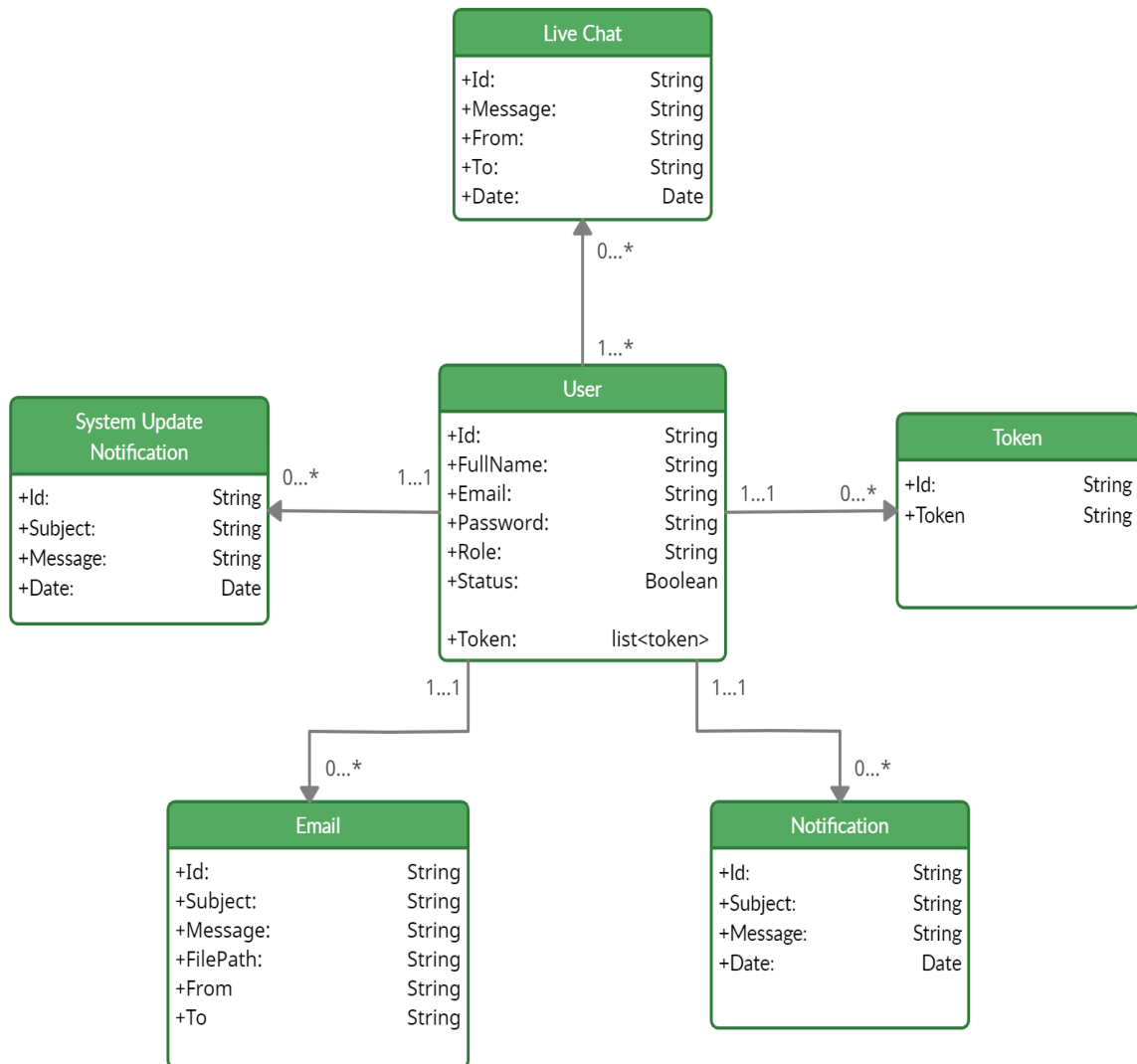


Figure 4. 1: Class diagram

### 4.3 ER Diagram

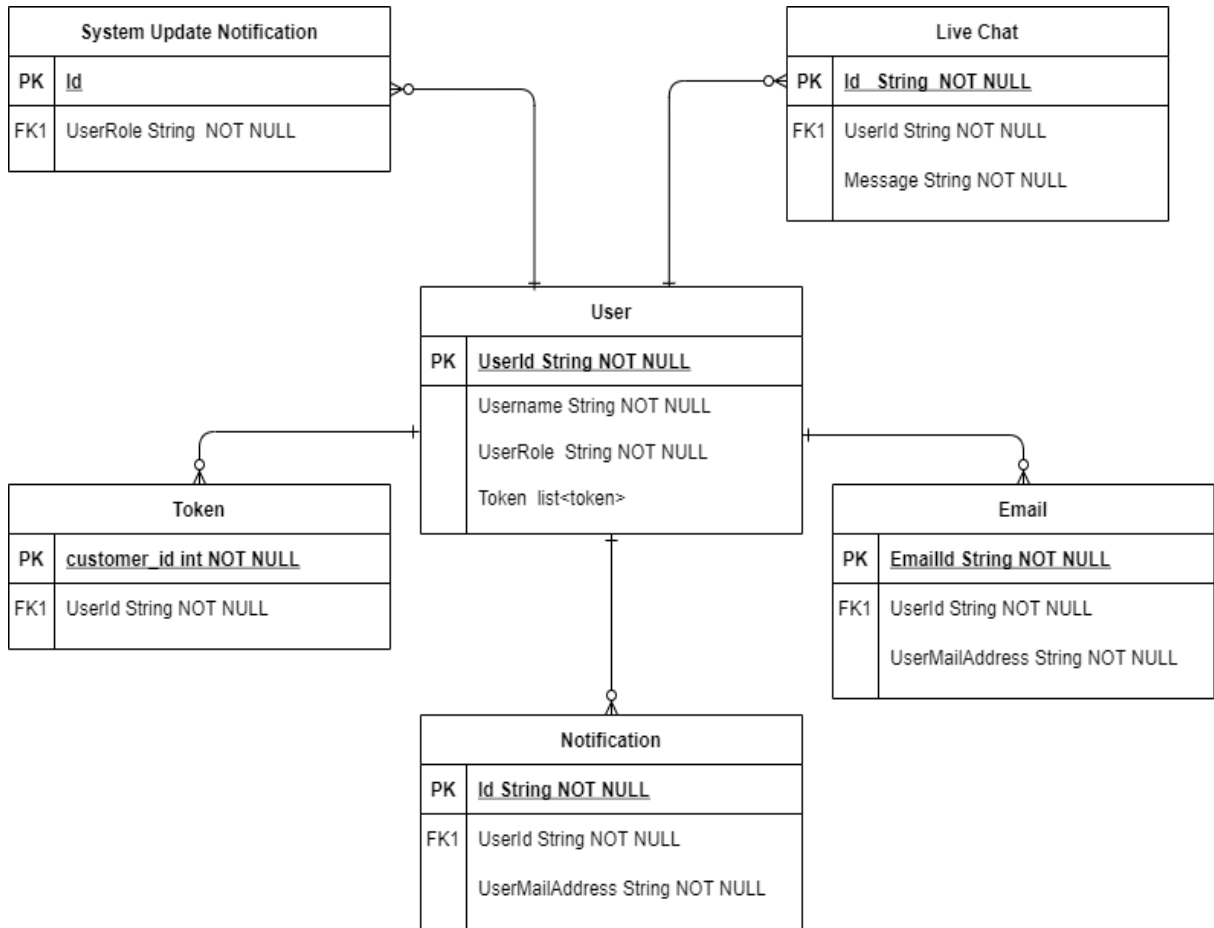


Figure 4. 2: ER diagram

## Chapter 5

### System Test

#### 5.1 Testing Features

Priority Level

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

##### 5.1.1 Admin Features

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

<b>Feature</b>	<b>Priority</b>	<b>Description</b>
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.



## 5.2 Test Cases

### 5.2.1 Account Registration (Admin)

<b>Test Case:</b> 5.2.1		<b>Test Case Name:</b> Account Registration (Admin)			
<b>System:</b> Secure mailbox		<b>Subsystem:</b>			
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>			
<b>Executed by:</b>		<b>Execution Date:</b>			
<b>Description:</b> Admin must register to gain access to the create notification. To register an account, must be provided an email and password.					
<b>Pre-condition:</b> <ul style="list-style-type: none"> <li>Email address must be unique.</li> <li>Passwords must be between 3 to 30 characters.</li> </ul>					
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	
<b>Post-condition:</b> A new user account has been registered successfully					



## 5.2.2 Account Registration (User)

<b>Test Case:</b> 5.2.2		<b>Test Case Name:</b> Account Registration (User)			
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>			
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>			
<b>Executed by:</b>		<b>Execution Date:</b>			
<b>Description:</b> User must register to gain access to the create schedule. To register an account, must be provided a valid email and password.					
<b>Pre-condition:</b> <ul style="list-style-type: none"> <li>Email address must be unique.</li> <li>Passwords must be between 3 to 30 characters.</li> </ul>					
Step	Email address	Password	Response	Pass/Fail	Comment
1	mazharul@securemail.com	mazharul123	Registration successfully	Pass	<a href="#">Figure 6.4</a>
2	mehedi@securemail.com	mehedi123	Registration successfully	Pass	<a href="#">Figure 6.4</a>
3	user2@securemail.com	Tanjina122	Enter correct mail / password	Fail	
<b>Post-condition:</b> A new user account has been registered successfully					

### 5.2.3 Login (Admin)

<b>Test Case:</b> 5.2.3		<b>Test Case Name:</b> Login (Admin)			
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>			
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>			
<b>Executed by:</b>		<b>Execution Date:</b>			
<b>Description:</b> Admin must login to gain access to the created notification. The system will check for authentication and authorization					
<b>Pre-condition:</b>					
<ul style="list-style-type: none"> <li>Admin must be use valid email and password</li> </ul>					
Step	Email address	Password	Response	Pass/Fail	Comment
1	admin1@secu remail.com	123	Login successfully	Pass	<a href="#">Figure 6.1</a>
2	admin1@secu remail.com	1456d	Incorrect credential	Fail	
<b>Post-condition:</b> Admin will successfully log into the system					

### 5.2.4 Login (User)

<b>Test Case:</b> 5.2.4		<b>Test Case Name:</b> Login (User)			
<b>System:</b> Secure mailbox		<b>Subsystem:</b>			
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>			

<b>Executed by:</b>		<b>Execution Date:</b>			
<b>Description:</b> User must login to gain access. The system will check for authentication and authorization					
<b>Pre-condition:</b>					
<ul style="list-style-type: none"> <li>• user must input valid email and password</li> </ul>					
Step	Email address	Password	Response	Pass/Fail	Comment
1	user1@securemail.com	123	Login successfully	Pass	<a href="#">Figure 6.5</a>
2	user1@securemail.com	675656cfdcfgg	Incorrect credential	Fail	
<b>Post-condition:</b> User will successfully log into the system					

### 5.2.5 View User List

<b>Test Case:</b> 5.2.5		<b>Test Case Name:</b> View User List			
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>			
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>			
<b>Executed by:</b>		<b>Execution Date:</b>			
<b>Description:</b> Registered admin can view the user list					
<b>Pre-condition:</b>					
<ul style="list-style-type: none"> <li>• Database has at least one user information to display</li> </ul>					
Step	Action	Response	Pass/Fail	Comment	

1	Admin Click on User List	System will display User list	Pass	
<b>Post-condition:</b> Display all old or new user				

### 5.2.6 Create Notification

<b>Test Case:</b> 5.2.6		<b>Test Case Name:</b> Create Notification		
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>		
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>		
<b>Executed by:</b>		<b>Execution Date:</b>		
<b>Description:</b> Admin can create notification				
<b>Pre-condition:</b>				
<ul style="list-style-type: none"> <li>Must be logged in</li> </ul>				
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	
<b>Post-condition:</b> Notification Created Successfully.				

### 5.2.7 Create Mail

<b>Test Case:</b> 5.2.7		<b>Test Case Name:</b> Create Mail		
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>		
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>		
<b>Executed by:</b>		<b>Execution Date:</b>		
<b>Description:</b> Registered user can create his/her mail				
<b>Pre-condition:</b>				
<ul style="list-style-type: none"> <li>• Must be logged in</li> </ul>				
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	
<b>Post-condition:</b> Your mail send Successfully!				

### 5.2.8 Encrypt File

<b>Test Case:</b> 5.2.8	<b>Test Case Name:</b> Encrypt File
<b>System:</b> Secure Mailbox	<b>Subsystem:</b>

<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>		
<b>Executed by:</b>		<b>Execution Date:</b>		
<b>Description:</b> Registered user can encrypt their file after login the system.				
<b>Pre-condition:</b>				
<ul style="list-style-type: none"> <li>• Must be logged in</li> </ul>				
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	
<b>Post-condition:</b> Successfully your file is encrypted!				

### 5.2.9 Decrypt File

<b>Test Case:</b> 5.2.9	<b>Test Case Name:</b> Decrypt File
<b>System:</b> Secure Mailbox	<b>Subsystem:</b>
<b>Designed by:</b> Mazharul Islam	<b>Design Date:</b>
<b>Executed by:</b>	<b>Execution Date:</b>
<b>Description:</b> Registered user can encrypt their file.	



<b>Pre-condition:</b> 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	
<b>Post-condition:</b> Your file is decrypted successfully!				

### 5.2.10 View system notification List

<b>Test Case:</b> 5.2.10		<b>Test Case Name:</b> View Notification List		
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>		
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>		
<b>Executed by:</b>		<b>Execution Date:</b>		
<b>Description:</b> Registered user can check their notification list.				
<b>Pre-condition:</b> 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

### 5.2.11 Live Chat

<b>Test Case:</b> 5.2.11		<b>Test Case Name:</b> Live Chat		
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>		
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>		
<b>Executed by:</b>		<b>Execution Date:</b>		
<b>Description:</b> Registered user should send message each other.				
<b>Pre-condition:</b> <ul style="list-style-type: none"><li>• User must be logged in</li><li>• User select another to send text</li></ul>				
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	
<b>Post-condition:</b> Message sent successfully.				

### 5.2.12 Check message notification

<b>Test Case:</b> 5.2.12		<b>5.2.1</b> Test Case Name: Check message notification		
<b>System:</b> Secure Mailbox		<b>Subsystem:</b>		
<b>Designed by:</b> Mazharul Islam		<b>Design Date:</b>		
<b>Executed by:</b>		<b>Execution Date:</b>		
<b>Description:</b> Registered user can check notification list after their login.				
<b>Pre-condition:</b> <ul style="list-style-type: none"><li>• Must be logged in</li><li>• Must have at least a data</li></ul>				
Step	Action	Response	Pass/Fail	Comment
1	Click on notification icon	Show notification	Pass	
<b>Post-condition:</b> Show notification list.				

# Chapter 6

## User Manual

### 6.1 Admin

#### 6.1.1 Login (Admin)

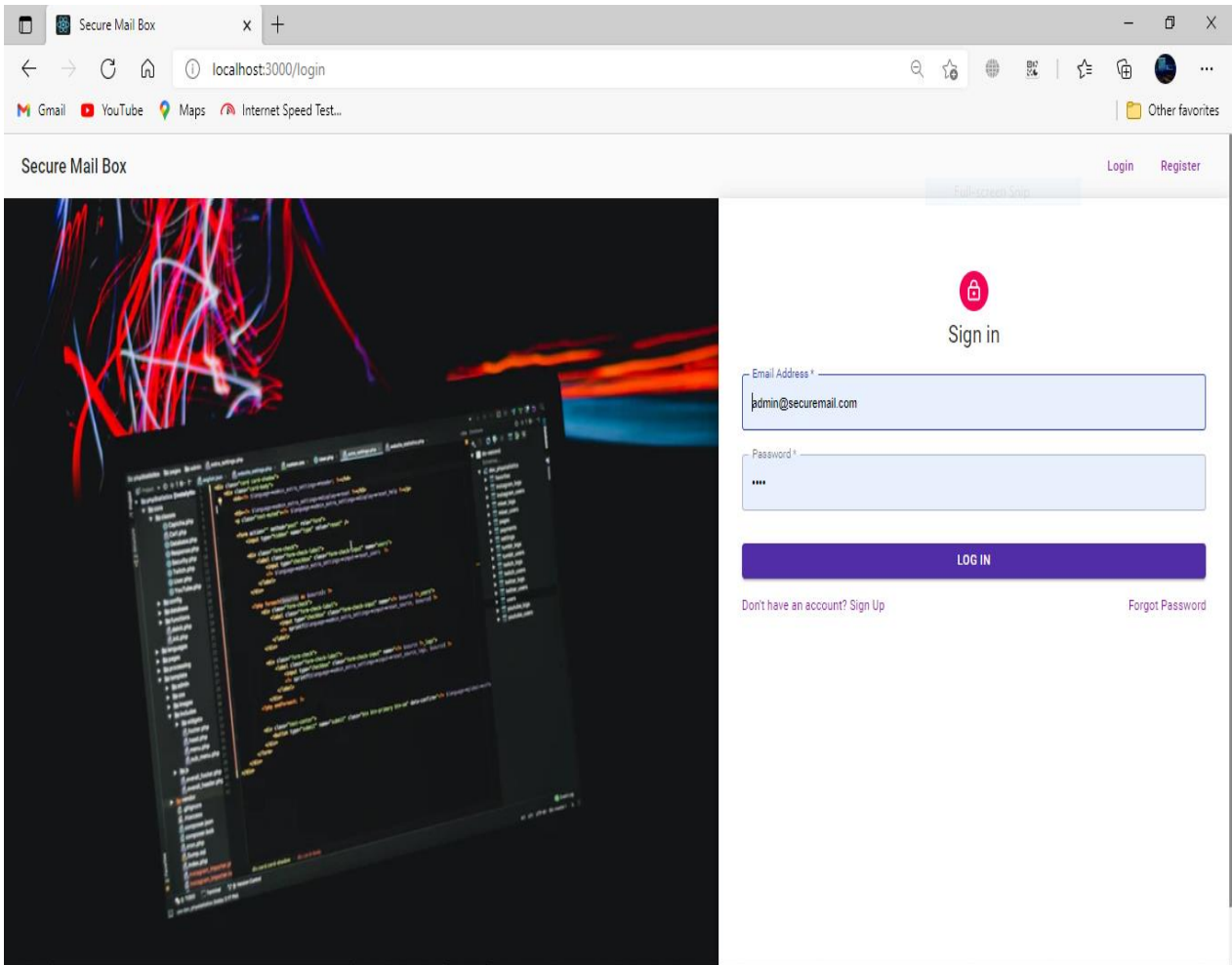
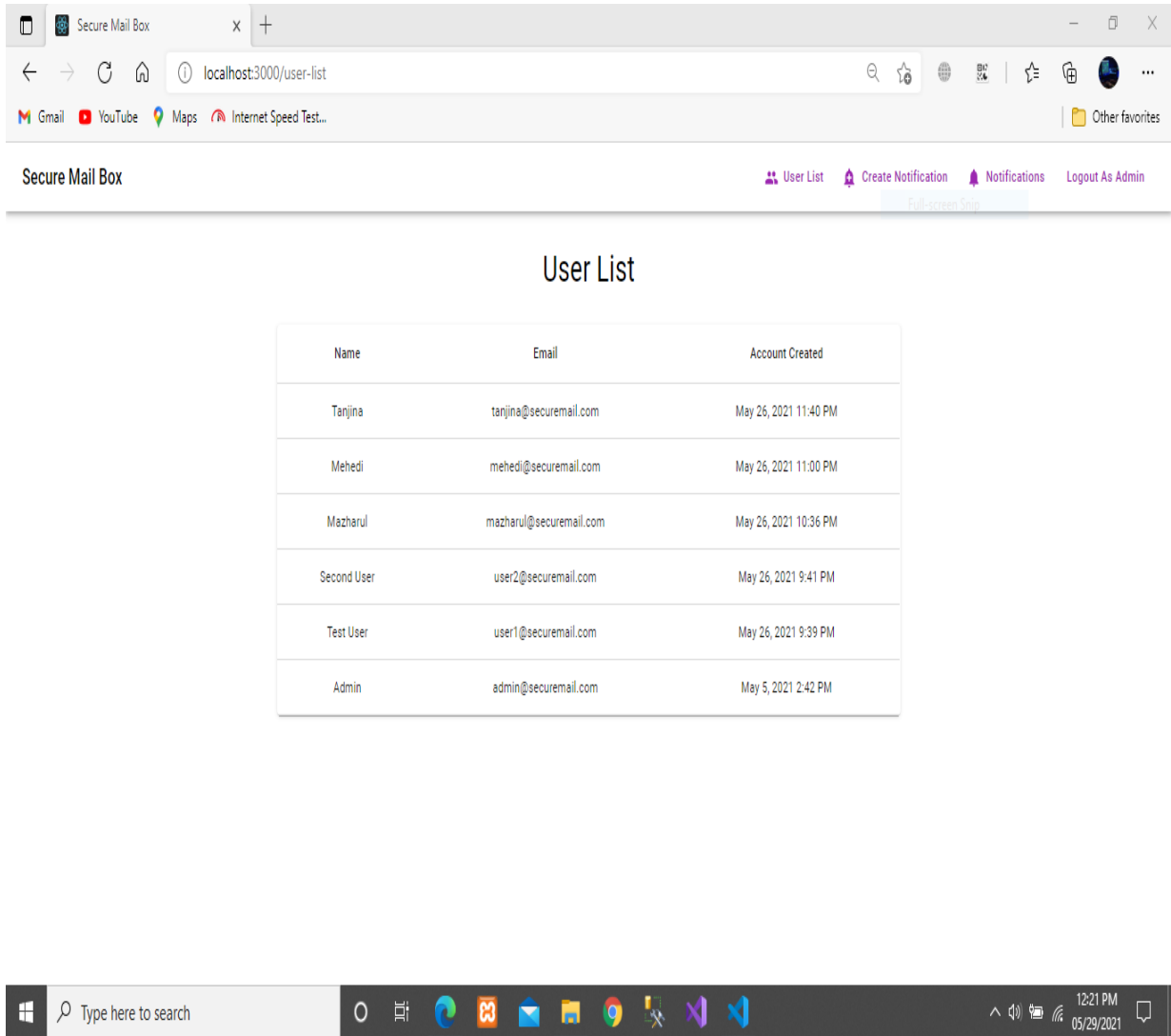


Figure 6.1: Login

## 6.1.2 Check User List



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/user-list'. The page title is 'Secure Mail Box'. The navigation bar includes links for 'User List', 'Create Notification', 'Notifications', and 'Logout As Admin'. The main content area is titled 'User List' and contains a table with the following data:

Name	Email	Account Created
Tanjina	tanjina@securemail.com	May 26, 2021 11:40 PM
Mehedi	mehedi@securemail.com	May 26, 2021 11:00 PM
Mazharul	mazharul@securemail.com	May 26, 2021 10:36 PM
Second User	user2@securemail.com	May 26, 2021 9:41 PM
Test User	user1@securemail.com	May 26, 2021 9:39 PM
Admin	admin@securemail.com	May 5, 2021 2:42 PM

**Figure 6.2: User List**

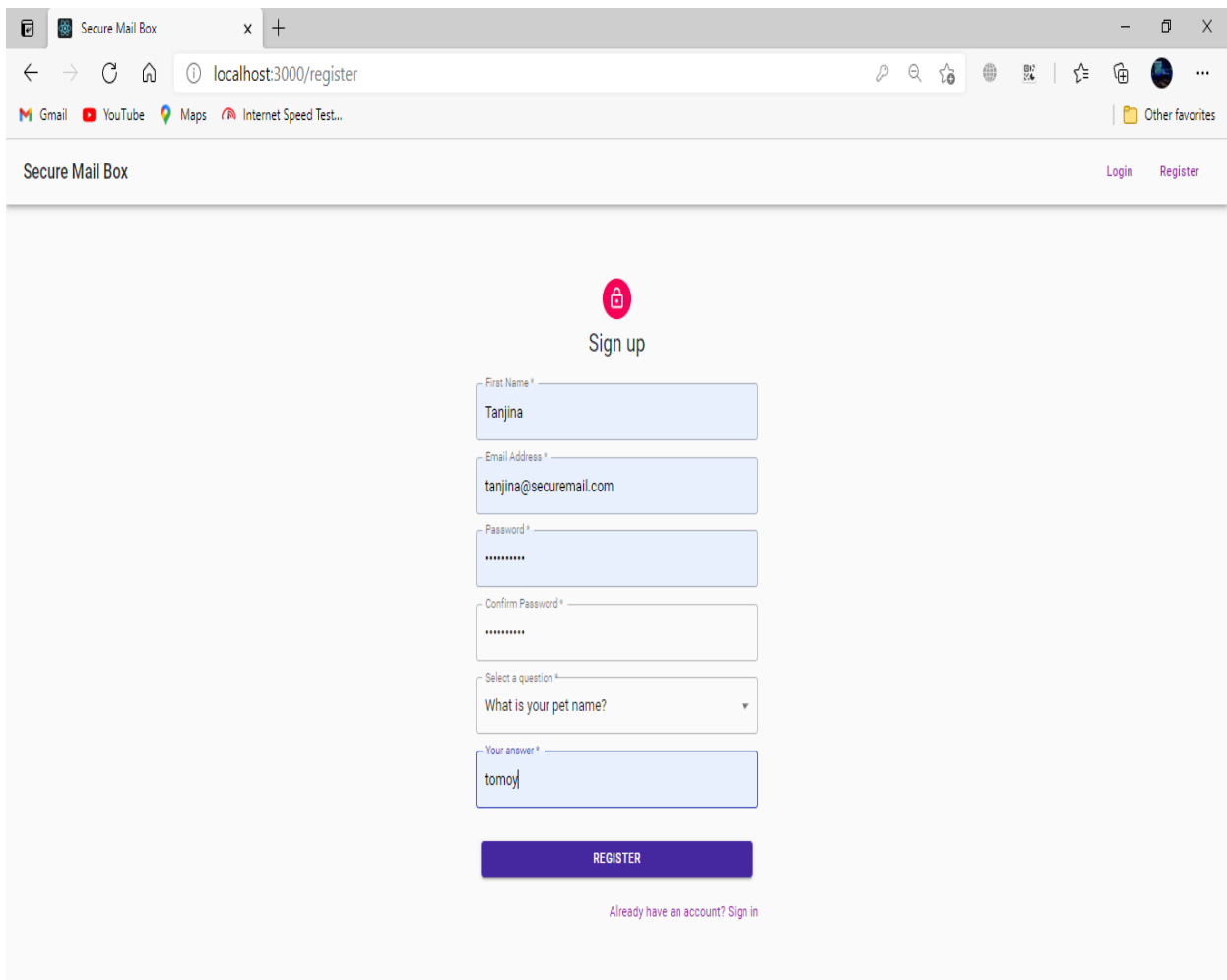
## 6.1.3 Create Notification

The screenshot shows a web browser window with the address bar displaying `localhost:3000/create-notification`. The browser's address bar includes navigation icons (back, forward, refresh, home) and search, zoom, and other utility icons. Below the address bar, there are several favorite links: Gmail, YouTube, Maps, and Internet Speed Test... The page title is "Secure Mail Box". In the top right corner of the page, there is a navigation menu with the following items: "User List", "Create Notification" (which is highlighted), "Notifications", and "Logout As Admin". Below the navigation menu, there is a "Full-screen Snip" button. The main content area is titled "Create Notification". It contains two text input fields. The first field is labeled "Notification Title" and contains the text "System Update". The second field is labeled "Details" and contains the text "05-05-2021 Server will be down for server maintenance!". Below these fields is a large blue button labeled "CREATE NOTIFICATION". At the bottom of the screenshot, the Windows taskbar is visible, showing the search bar with the text "Type here to search", several application icons (Edge, Word, Mail, File Explorer, Chrome, Task View, Visual Studio Code, and Teams), and the system tray with the date and time "12:27 PM 05/29/2021".

**Figure 6.3: Create Notification**

## 6.2 User

### 6.2.1 User Registration



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The page title is 'Secure Mail Box'. In the top right corner, there are links for 'Login' and 'Register'. The main content area features a 'Sign up' heading with a lock icon. Below the heading are several input fields: 'First Name \*' with the value 'Tanjina', 'Email Address \*' with 'tanjina@securemail.com', 'Password \*' and 'Confirm Password \*' both masked with dots, 'Select a question \*' with a dropdown menu showing 'What is your pet name?', and 'Your answer \*' with the value 'tomoy'. A blue 'REGISTER' button is positioned below the form. At the bottom, there is a link: 'Already have an account? Sign in'.

**Figure 6.4: User Registration**

## 6.2.2 Login (User)

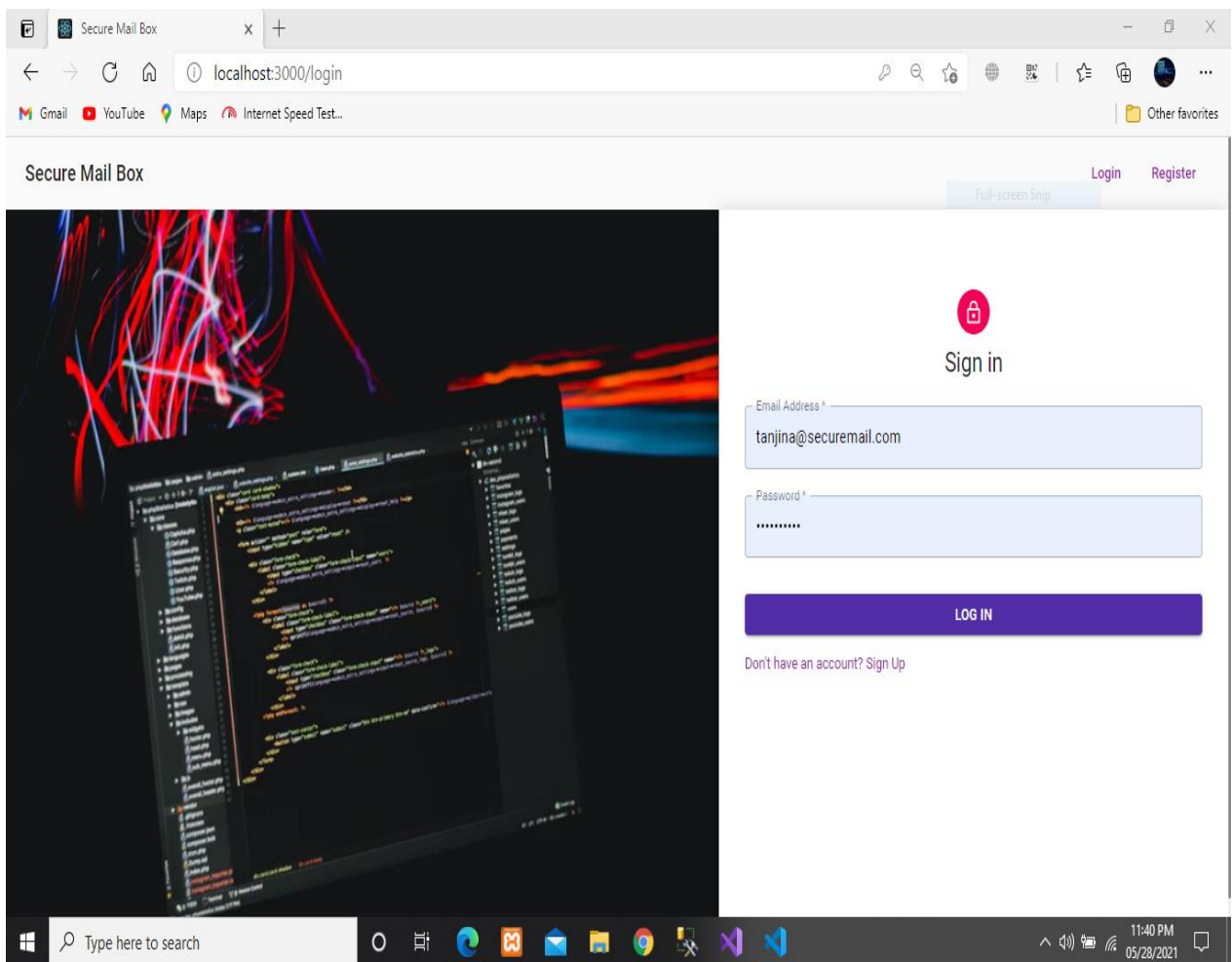


Figure 6.5: Login



## 6.2.3 User Check Mail List

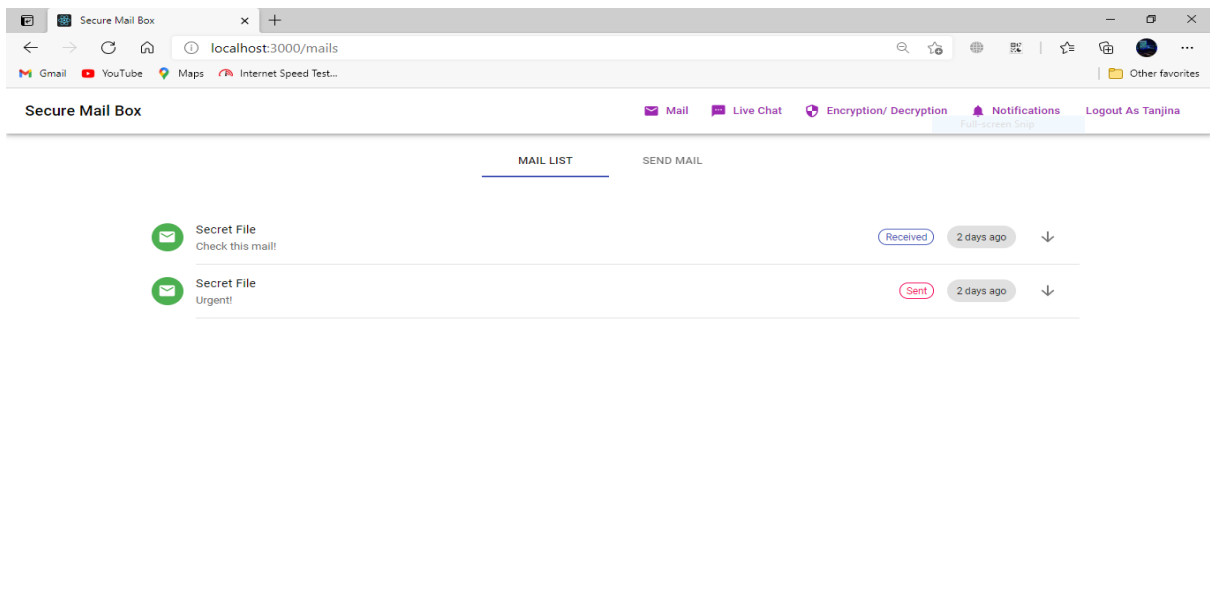
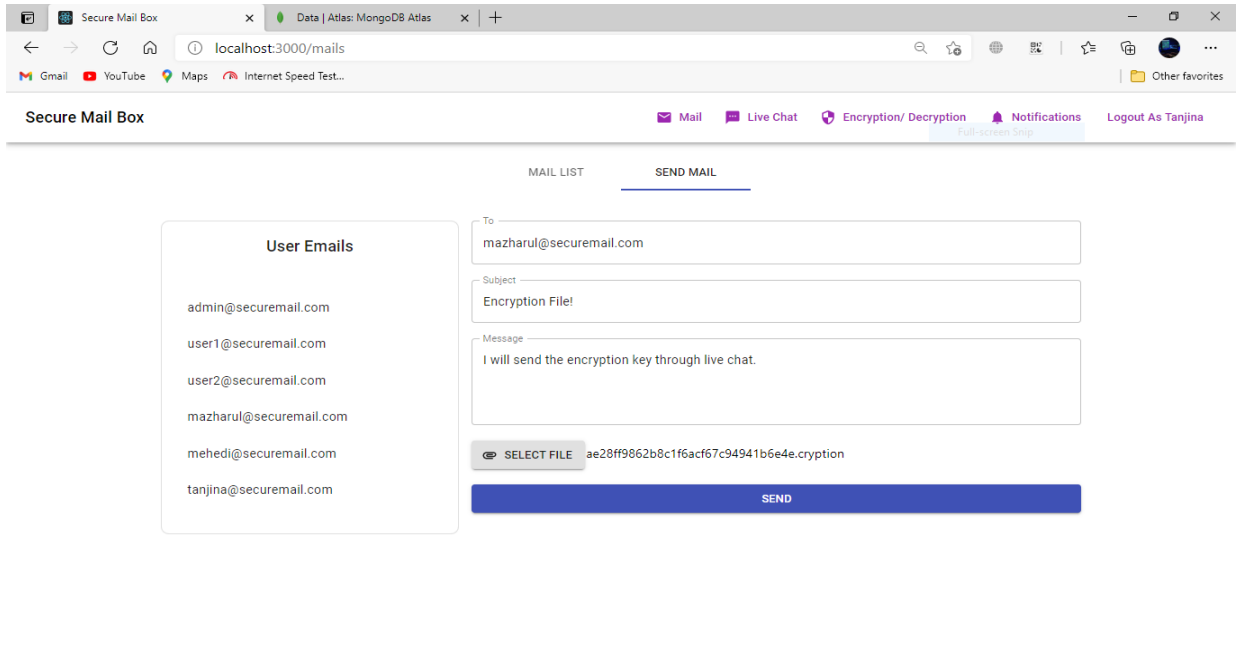


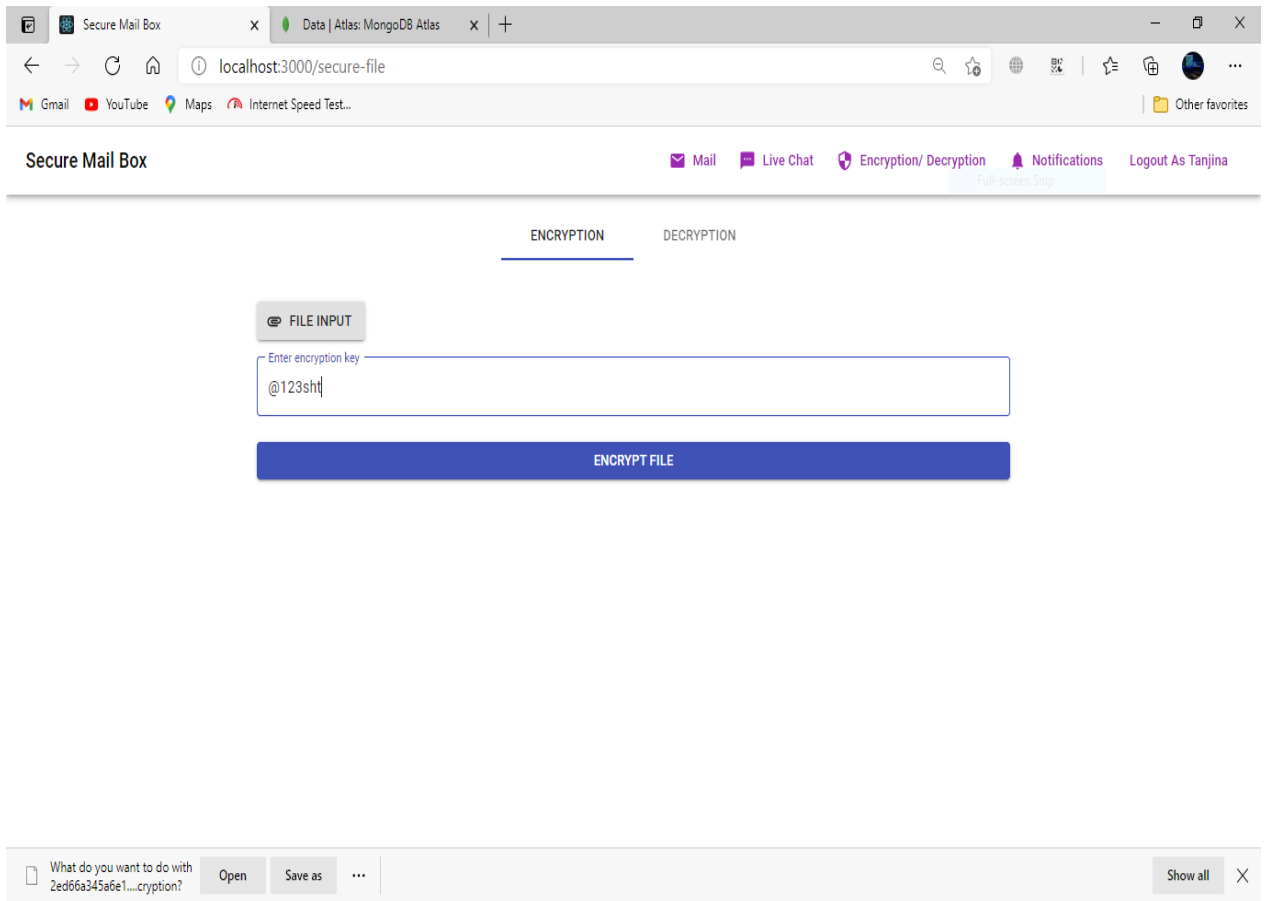
Figure 6.6 Mail List

## 6.2.4 User Create Mail



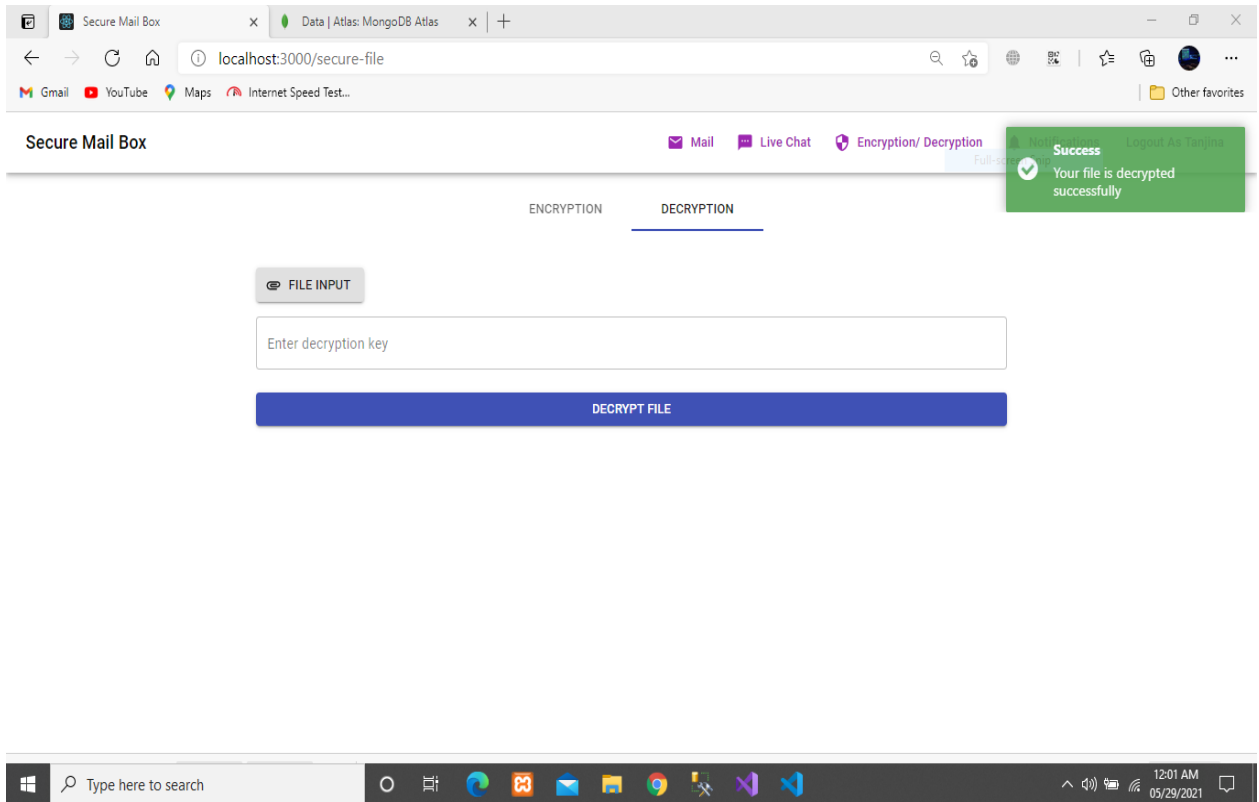
**Figure 6.7 Create Mail**

## 6.2.5 Encrypt File



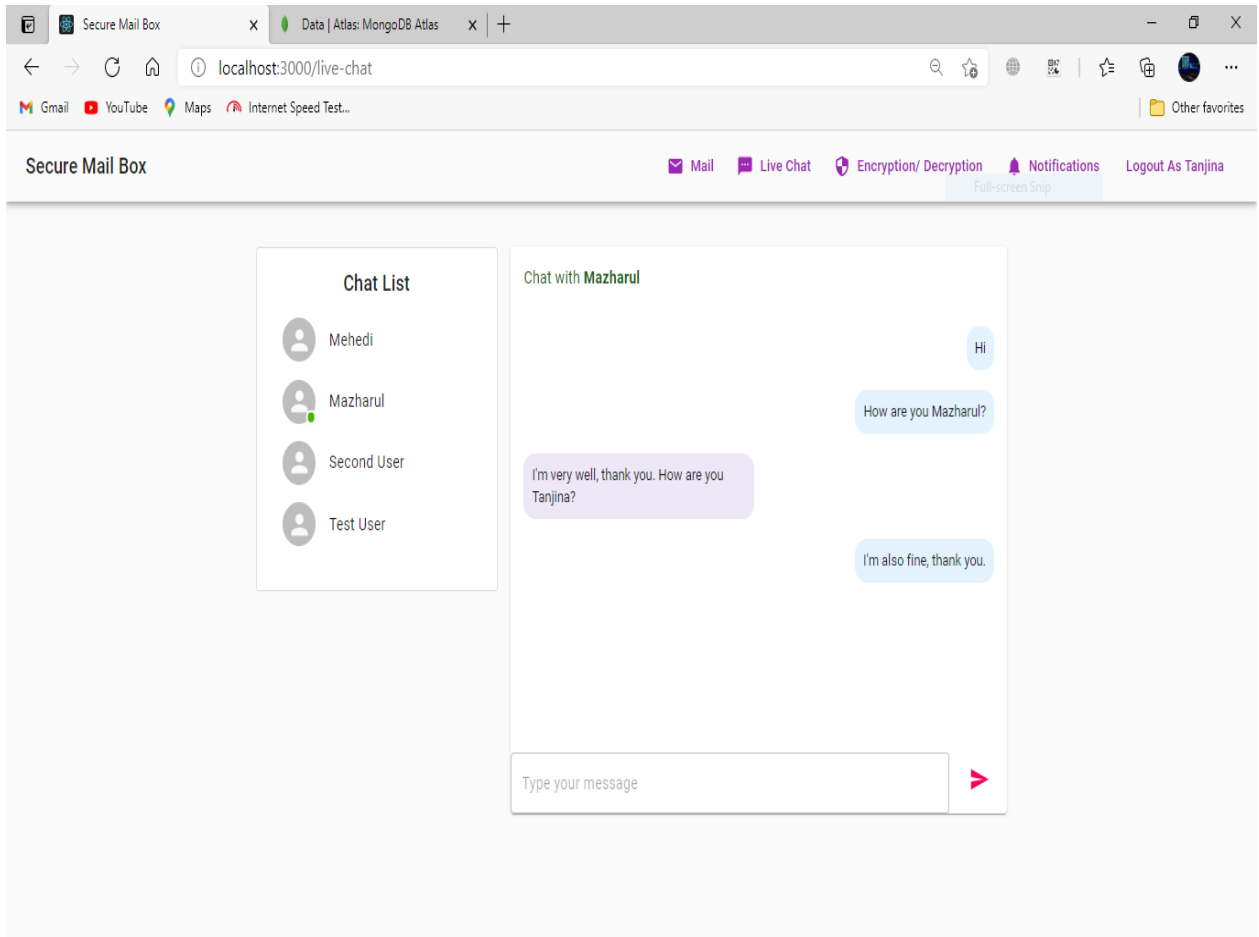
**Figure 6.8: Encrypt File**

## 6.2.6 Decrypt File



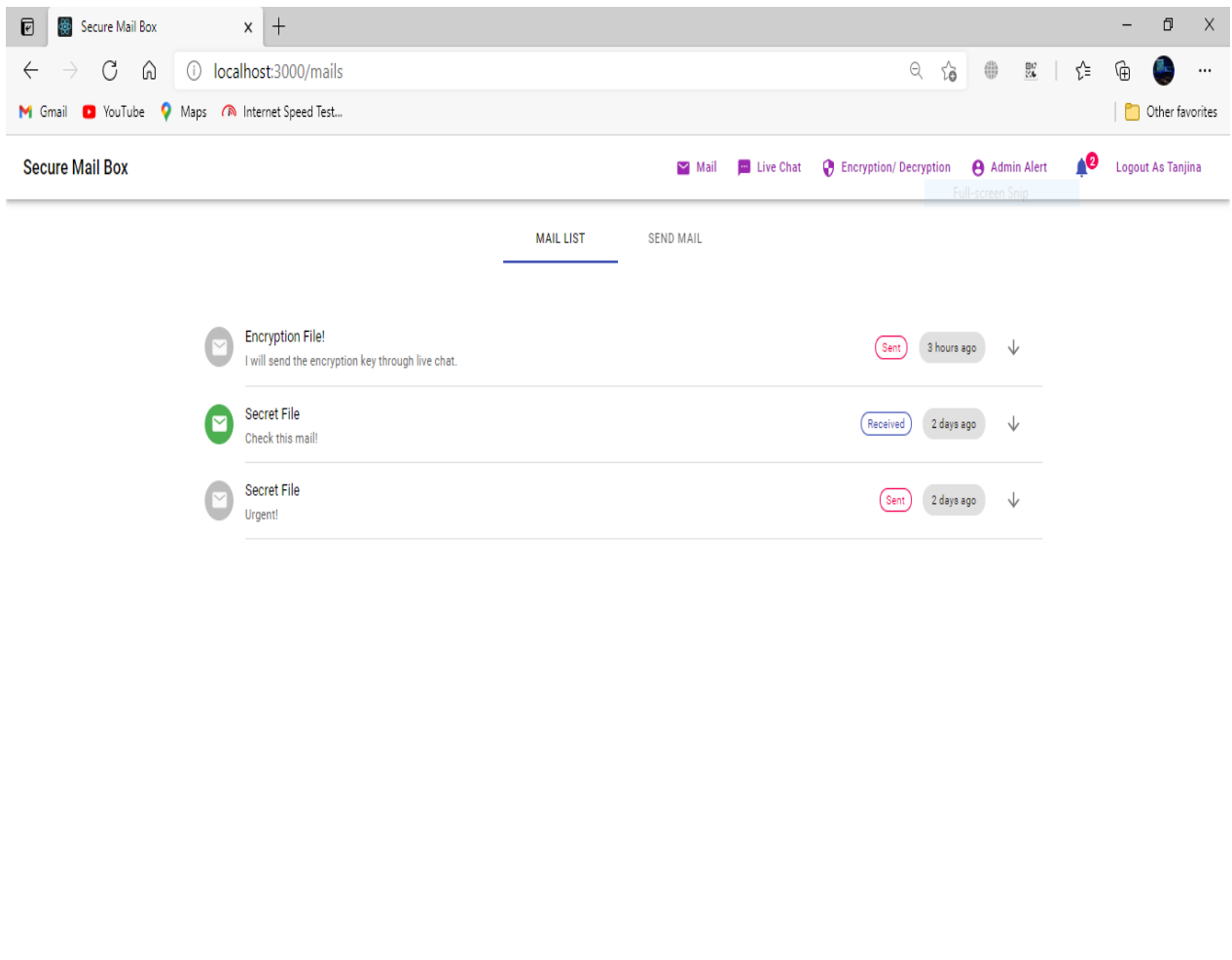
**Figure 6.9: Decrypt File**

## 6.2.7 Live Chat



**Figure 6.10: Live Chat**

## 6.2.8 Notification List



**Figure 6.11: Check Notification List**

## Chapter 7

### 7.1 Github Link



Github Link:

<https://github.com/MAZHARULISLAM11981/SecureMailbox>

### 7.2 Obstacles & Achievements:

When I 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.

**7.3 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/>

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/>

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](#)

Development Guideline: [Cryptographic Standards and Guidelines | CSRC \(nist.gov\)](#)

---



# Plagiarism Report

6/24/2021

Turnitin

<p><b>Turnitin Originality Report</b></p> <p>Processed on: 24-Jun-2021 15:02 +06                  ID: 1611495314                  Word Count: 6864                  Submitted: 1</p> <p>171-35-1798 By Md. Mazharul Islam</p>		<table border="1"> <tr> <td>Similarity Index</td> <td><b>24%</b></td> </tr> <tr> <td colspan="2"><b>Similarity by Source</b></td> </tr> <tr> <td>Internet Sources:</td> <td>20%</td> </tr> <tr> <td>Publications:</td> <td>2%</td> </tr> <tr> <td>Student Papers:</td> <td>17%</td> </tr> </table>	Similarity Index	<b>24%</b>	<b>Similarity by Source</b>		Internet Sources:	20%	Publications:	2%	Student Papers:	17%
Similarity Index	<b>24%</b>											
<b>Similarity by Source</b>												
Internet Sources:	20%											
Publications:	2%											
Student Papers:	17%											

4% match (student papers from 02-Apr-2018) <a href="#">Submitted to Daffodil International University on 2018-04-02</a>
3% match (Internet from 01-Apr-2020) <a href="https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report">https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report</a>
2% match (Internet from 15-Mar-2020) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3553/P13659%20%2829%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3553/P13659%20%2829%25%29.pdf?isAllowed=y&amp;sequence=1</a>
2% match (student papers from 28-Jan-2021) <a href="#">Submitted to Deptford Township High School on 2021-01-28</a>
1% match (Internet from 06-Jan-2020) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3547/P13646%20%2824%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3547/P13646%20%2824%25%29.pdf?isAllowed=y&amp;sequence=1</a>
1% match (Internet from 24-Feb-2020) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3555/P13663%20%2821%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3555/P13663%20%2821%25%29.pdf?isAllowed=y&amp;sequence=1</a>
1% match (Internet from 28-Oct-2018) <a href="https://www.researchgate.net/publication/320068250">https://www.researchgate.net/publication/320068250</a>
1% match (student papers from 24-Aug-2020) <a href="#">Submitted to Higher Education Commission Pakistan on 2020-08-24</a>
1% match (Internet from 19-Nov-2020) <a href="https://swagger.io/docs/specification/2-0/what-is-swagger/">https://swagger.io/docs/specification/2-0/what-is-swagger/</a>
1% match (student papers from 31-Oct-2006) <a href="#">Submitted to Sim University on 2006-10-31</a>
< 1% match (student papers from 31-Mar-2018) Class: Article 2018 Assignment: Journal Article Paper ID: <a href="#">939036910</a>
< 1% match (student papers from 11-Apr-2018) Class: April 2018 Project Report Assignment: Student Project Paper ID: <a href="#">944874623</a>
< 1% match (Internet from 10-Nov-2020) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3552/P13658%20%287%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3552/P13658%20%287%25%29.pdf?isAllowed=y&amp;sequence=1</a>
< 1% match (Internet from 05-Jun-2020) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3927/P148856%20%2828_%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3927/P148856%20%2828_%29.pdf?isAllowed=y&amp;sequence=1</a>
< 1% match (Internet from 26-Nov-2020) <a href="https://www.slideshare.net/PriyankaDeo@are/online-marketplace-report">https://www.slideshare.net/PriyankaDeo@are/online-marketplace-report</a>
< 1% match (Internet from 25-Jun-2015)