Secure Banking System

CSE545: Software Security- Project Report

Submitted to:

Professor Stephen.S.Yau

Group 13

Atin Singhal(L), Shivank Tiwari(DL)

Sriprashanth Ramamoorthy, Ayush Ray

Raj Buddhadev, Kangjian Ma

Uttam Bhat, Amitabh Das

May 2020

Abstract

This is the final documentation of the Secure Banking System(SBS) developed by group 13 for CSE 545:Software Security. The documentation comprises of:

1. Implementation design
2. Tier wise functional and non-functional features
3. Security measures implemented
4. Roles and the responsibilities of each team members
5. Security and the functional vulnerabilities reported by the end-users

Contents

1 Introduction **1**

2 Functional Requirements **2**

2.1 Users of the system 2

2.1.1 Customers 2

2.1.2 Employees 3

2.2 Functional Features 3

2.2.1 Debit and Credit Funds 3

2.2.2 Transfer Funds 3

2.2.3 Email/phone transfer function 4

2.2.4 Technical Account Access 4

2.2.5 Banking Statements 4

2.2.6 Opening additional accounts 4

2.2.7 Help & Support Center 4

3 Security Features Implementation **5**

3.1 One Time Password (OTP) 5

3.2 Secure Sockets Layer (SSL) 5

3.3 Public Key Infrastructure 6

3.4 Malicious Login Controls 6

3.5 CSRF Tokens 6

3.6 Form Validations 6

3.7 Hyperlegder Integration 6

3.8 URL protection 7

3.9 Hashed password storage 7

4 System Design **8**

4.1 System Architecture 8

4.2 System Decomposition 9

4.3 Data Design 10

4.3.1 Data Description 10

4.3.2 Data Dictionary 12

4.4 Component Design 15

5 Contributions **17**

6 Vulnerability Report **21**

6.1 Valid Security Vulnerabilities 21

6.2 Valid Functional Vulnerabilities 22

6.3 Invalid Vulnerabilities 24

7 User Guide **33**

7.1 Overview of the User Guide 33

7.1.1 Purpose of the User Guide 33

7.1.2 How to Use the Guide 33

7.2 Requirements 34

7.3 Overview of the Secure Banking System 34

7.4 Online Banking Interfaces 35

7.4.1 Login 35

7.4.2 Internal Users 36

7.4.3 External Users 37

7.5 Sample Users 39

7.6 Possible Errors Troubleshooting 40

8 Conclusion **54**

9 References **55**

List of Figures

4.1 System Architecture 9
4.2 System Decomposition 10
4.3 Detailed Flow Diagram 11
4.4 Class Diagram 15
4.5 Use Case Diagram 16
4.6 Misuse Case Diagram 16

7.1 Login Page 35
7.2 Registration Page 36
7.3 Forgot Password Page 36
7.4 Approve/Decline Transaction Page 37
7.5 Create Transaction Page 37
7.6 Issue Cashier Cheque Page 38
7.7 Deposit Cashier Cheque Page 38
7.8 Deposit Money Page 38
7.9 Withdraw Money Page 38
7.10 View Customer Account Page 39
7.11 Change Password Page 39
7.12 View Appointment of the Day Page 40
7.13 Tier-2 Employee Dashboard 40
7.14 Approve/Decline Transactions Page 40
7.15 Change Password Page 40
7.16 Manage Locked Account Page 41
7.17 Account Approval Page 41

7.18 Search Customer Account Page 42
7.19 Delete Customer Account Page 42
7.20 Update Customer Info Page 42
7.21 View Appointment of the Day Page 43
7.22 Tier-3 Admin Dashboard 43
7.23 Tier-3 Admin View Employee 43
7.24 Tier-3 Admin Create Employees 43
7.25 Tier-3 Admin Update Employee 44
7.26 Tier 3 Admin Delete Employee 44
7.27 Tier 3 Admin Sys Logs 44
7.28 Tier 3 Admin Update Password 45
7.29 Customer Dashboard Page 45
7.30 Deposit/ Withdraw Money Page 45
7.31 View Transactions Page 46
7.32 Transfer Funds Page 46
7.33 Service Request Page 46
7.34 OTP Page for Scheduling an Appointment 46
7.35 Page for Scheduling an Appointment 47
7.36 Order/Deposit Cashier's Cheque Pag 47
7.37 Opt Page for Opening a New Account 47
7.38 Open New Account Page 47
7.39 Set Primary Account Page 48
7.40 Change Password Page 48
7.41 Approve/Decline Page 48
7.42 49
7.43 50
7.44 51
7.45 52
7.46 52
7.47 53

Chapter 1

Introduction

Billions of monetary transactions happen on a daily basis and they are the ones which are prone to security threats. A small security vulnerability in these transactions could lead to a loss of millions of dollars.The Secure Banking System (SBS) developed, tackles these security threats by taking proactive measures and giving the customers a secure gateway to transact money. The need for 24 hour service is also a vital functionality of a globally recognize bank and, SBS caters these too by giving the users an around the clock service.SBS is available to 3 types of users : Customer,Internal Employees and the Admin.SBS can accommodate multiple users to use the system under sufficient security at any time and from any place as long as the internet and a web browser is accessible.

Several critical security measurements are implemented in SBS to prevent malicious attacks, including Public Key Certificates, OTP(One-Time Passwords), monitoring malicious login controls, data encryption and capturing the transaction on Blockchain. This paper would mainly concentrate on how these security measures are implemented in SBS.

Chapter 2

Functional Requirements

2.1 Users of the system

As mentioned in the Abstract, the application developed in this project is a skeleton of a banking system. As any basic banking system, our application has two categories of users - Customers and Employees.

2.1.1 Customers

The customers have the ability to register in our application. Once registered, the customers have the ability to transfer their money between different accounts. Each have atleast one of the following types of accounts: Savings Account, Checking Account and Credit Card Account. All of these type of accounts have the ability to transfer, credit or debit funds with a limit of 1000$ per transaction - which are known as critical transactions. This limit is enforced by the system design. There are two kinds of customers:

1. **Individuals:** Users with all the basic functionality.
2. **Merchants:** They are corporate users with the additional ability of processing payments from clients.

2.1.2 Employees

Admins

They are the super users of the system. They maintain the application, manage the accounts of all other employees, authorize or decline employee requests, have access to the system logs and can also create other class of employees.

Tier 2 Employees

Tier 2 manages the customer accounts - opening, closing, modification, etc. They also have the ability to authorize or decline the critical transactions initiated by the customers.

Tier 1 Employees

These employees have the least privileges among all other class of employees. They can perform transactions like fund transfer, credit, debit - with the customers consent. However, they cannot take part in any critical transactions.

2.2 Functional Features

2.2.1 Debit and Credit Funds

The customers have the ability to debit and credit money from and to their own accounts. This is still limited by the critical transaction feature enforced by the system where any transaction beyond 1000$ is to be approved by a Tier 2 Employee. Tier 1 Employees also have the ability to debit and credit money from and to any account, but they have to be approved by the customer who owns the account.

2.2.2 Transfer Funds

The customers have the ability to transfer money from any account they own to *any* other account. The critical transaction feature applies here as well. Tier 1 Employees have this ability too, but requires consent of the owner of the account from where the money is transferred from, in the transaction.

2.2.3 Email/phone transfer function

All customers can select one of their accounts to act as a Primary Account. A Primary Account is the account which is chosen when a transfer is made to an email ID belonging to the customer of that account. Without a Primary Account, this feature does not work.

2.2.4 Technical Account Access

An admin has access to other employees account to perform troubleshooting and maintenance operations.

2.2.5 Banking Statements

A customer has access to the banking statement of an account - all fund transfers, credit, debit, cheque activity through their accounts home page.

2.2.6 Opening additional accounts

A customer can open an addition account (of any type), if required, in the same profile assigned to the customer. This account must be authorized by a Tier 2 employee before it can be used.

2.2.7 Help & Support Center

Every Customer Profile has the capability to schedule an appointment with the Bank. This appointment is scheduled with a random employee in the Bank.

Chapter 3

Security Features Implementation

This chapter discusses about the security features of our application.

3.1 One Time Password (OTP)

A one-time password (OTP), likewise known as one-time pin or dynamic password that is good for only one login session or action, on a computer system or other digital device. Our implementation incorporates two-factor authentication by ensuring that the one-time password requires access to an email a user has registered in our system with. To validate highly sensitive requests/transactions, the system must make sure that the request is originating from the real user. The assumption here is that only the real user would have access to the email to which the banking system sends the OTP. This OTP is tied to a user session and an action. Therefore, there is no chance an OTP can be brute forced by any malicious attacker.

3.2 Secure Sockets Layer (SSL)

SSL, or Secure Sockets Layer, is an encryption-based Internet security protocol. This was implemented in our application by enforcing HTTPS only traffic to the AWS EC2 Instance. SSL encrypts data that is transmitted across the web. Simply put, anyone intercepting this data only sees a meaningless sequence of characters that is extremely difficult to decrypt.

3.3 Public Key Infrastructure

A self-signed certificate is used by the server to identify itself to the clients. This stops other malicious attackers from impersonating as our banking application.

3.4 Malicious Login Controls

We've enforced locking of user accounts if there are more than 5 failed attempts due to wrong password. If a user account gets locked, a Tier 2 employee must be involved to unlock the users account. We've also enabled IP address blocking for 24 hours if there is any 10 consecutive failed login attempts.

3.5 CSRF Tokens

A CSRF token is a unique, secret, unpredictable value that is generated by the server-side application and transmitted to the client in such a way that it is included in every form in our website. When a form is submitted, Spring-Security verifies if this CSRF token is valid, and only then forwards the request to the controller. CSRF tokens can prevent CSRF attacks by making it impossible for an attacker to construct a fully valid HTTP request suitable for feeding to a victim user.

3.6 Form Validations

There are limited type of forms in our banking application that input data into the database. The goal of validating the inputs of a form is to avoid XSS attacks on victim users. White-listing is easier than black- listing patterns. There are strict regex patterns for every kind of input our banking system accepts which avoids any kind of XSS inputs to the database. These checks are done both at the front end and the backend.

3.7 Hyperlegder Integration

Our secure banking system utilizes the blockchain technology to capture all valid transactions and record them on an immutable ledger. This immutability protects the users by preventing any form of tampering with the transaction records. The ledger also serves as a source of truth

due to its immutable characteristics.

3.8 URL protection

Various end points in our application are to be viewed by certain authorized users, for example - system logs, a users account details, etc. Therefore, these endpoints need to be protected. Roles are assigned to each user based on the type of the user. Spring-Security is configured to declare the roles a URL can be accessed by.

3.9 Hashed password storage

Even with strong security measures in place, it is possible that the database could fall into the wrong hands. Hashing sensitive data in the database can ensure that even if the database is acquired by a malicious attacker, the database is useless since the hashed data is meaningless to anyone without the knowledge of what the data actually is. We hash the passwords using BCrypt hashing algorithm with 10 rounds. Even if this hashed password falls into the wrong hands, decryption is near impossible and gives the users and the bank enough time to take corrective measures.

Chapter 4

System Design

This chapter discusses about the system design of our application.

4.1 System Architecture

According to the project requirements, the system is designed with a three tier architecture: Presentation, application and data tier as shown in the Fig.4.1. There are many benefits to using a 3-layer architecture including speed of development, scalability, performance, and availability. As mentioned, modularizing different tiers of an application gives development teams the ability to develop and enhance a product with greater speed than developing a singular code base because a specific layer can be upgraded with minimal impact on the other layers. It can also help improve development efficiency by allowing teams to focus on their core competencies. Many development teams have separate developers who specialize in front- end, server back-end, and data back-end development, by modularizing these parts of an application you no longer have to rely on full stack developers and can better utilize the specialties of each team. Scalability is another great advantage of a 3- layer architecture. By separating out the different layers you can scale each independently depending on the need at any given time.

Presentation Tier - The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This tier is built on web technologies such as HTML5, JavaScript, CSS and communicates with others layers through JSPs.

Application Tier - The application tier contains the functional business logic which drives an application's core capabilities. It is built using Java and the Spring framework. The application

layer would communicate to the presentation layer through end points and corresponding JSPs.

Data Tier - The data tier comprises of the database/data storage system and data access layer. Data is stored in MySQL database for the banking system and would be accessed by Application Layer using JPA.

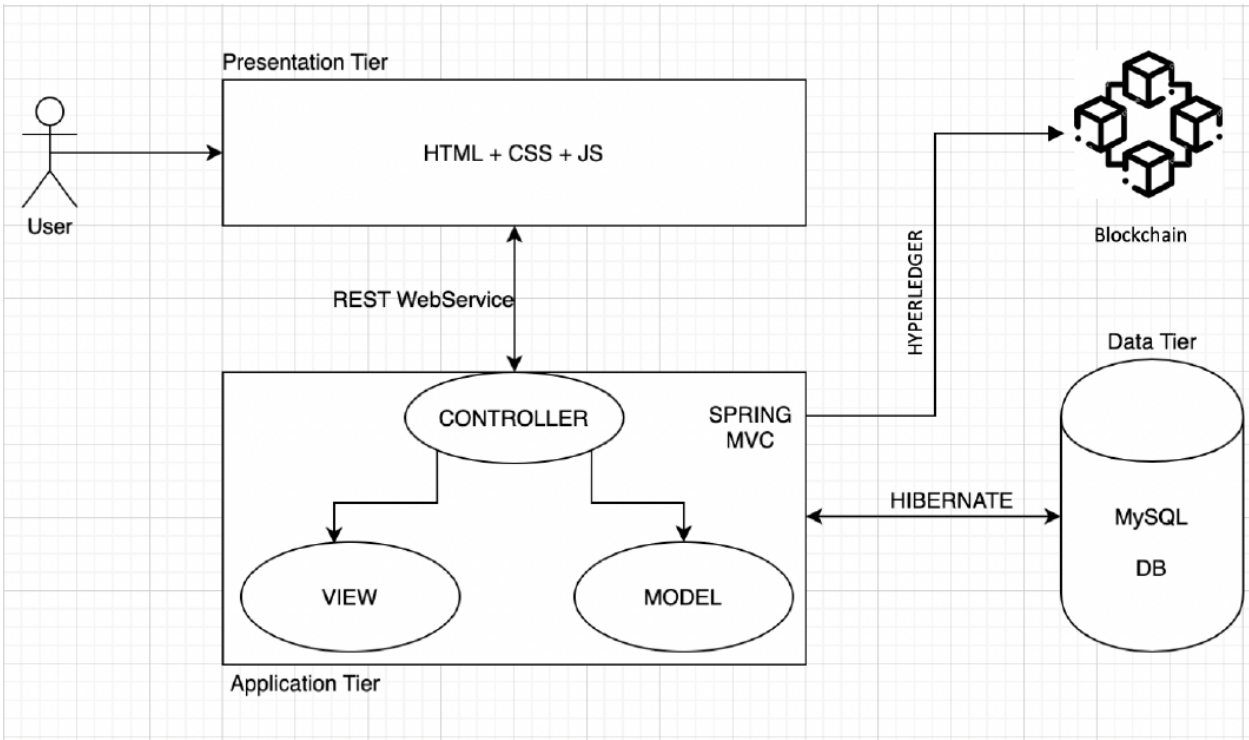

Figure 4.1: System Architecture

4.2 System Decomposition

The following data flow diagram, Fig.4.2 explains how information flows within the system. It indicates how data is utilized, processed and stored after each user action.

Fig.4.3 shows the modular level of decomposition of the entire system. It shows the relations between different users and data flow during a transaction. Security is the most important component in mind while deciding the architecture. We are using the spring framework and server-side back-end programming. The reason being Spring is easy to use/build and comes with strong security features.

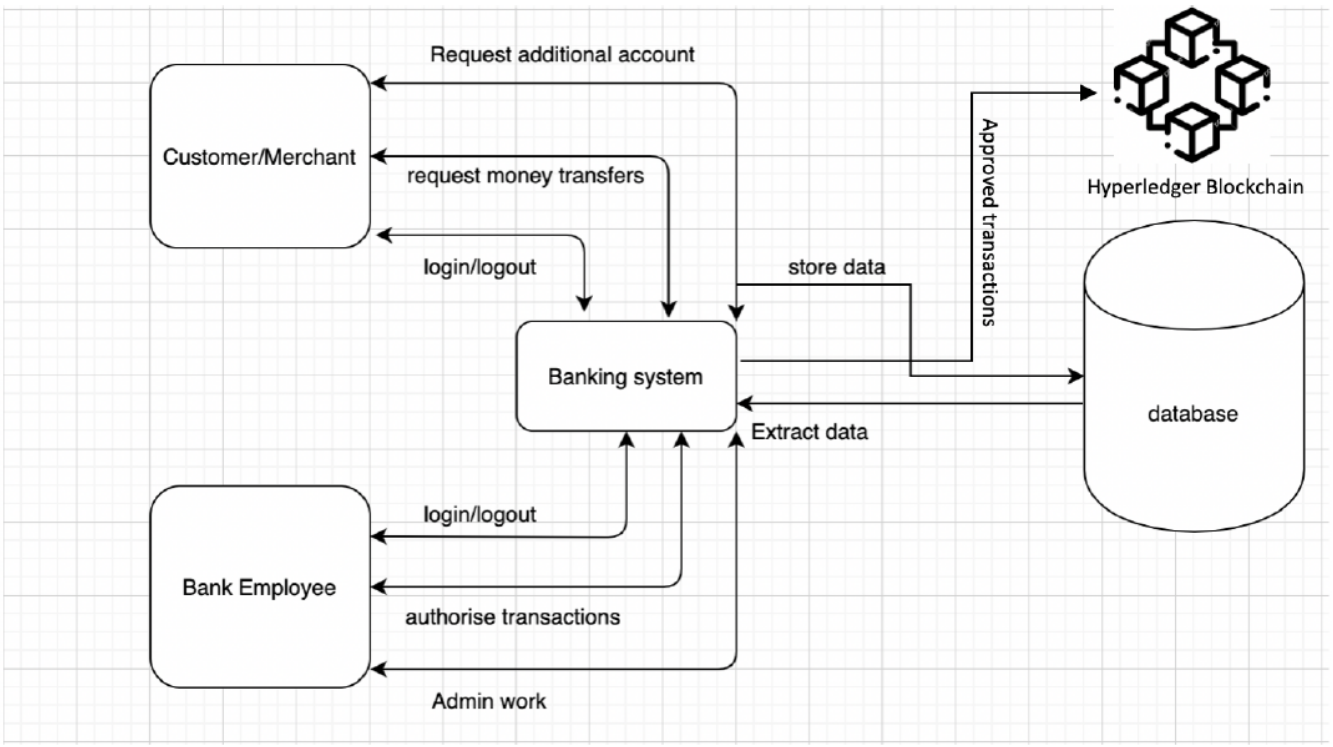

Figure 4.2: System Decomposition

4.3 Data Design

4.3.1 Data Description

Database is one of the important components of the secure banking system as there is a need of storing a lot of information pertaining to the banking system. Information about users, login details, account details, transactions, and appointments needs to be stored in the database. Hence, tables storing these different details are created to store this data.

The important fields that would be required to be stored in the database are explained below.

1. User Login Details - The user login details such as username, password, user created date, etc. are stored in the database.

2. User Type - The type of user i.e. Tier 1, Tier 2, Tier 3, Individual or Merchant is stored in the database. Different functionalities are made available to the user based on their type.

3. User Profile - All the details of the user such as name, email, phone, etc. is stored in order to retrieve user data when necessary.

4. Account Details of all users - Account details such as current balance, account open date,

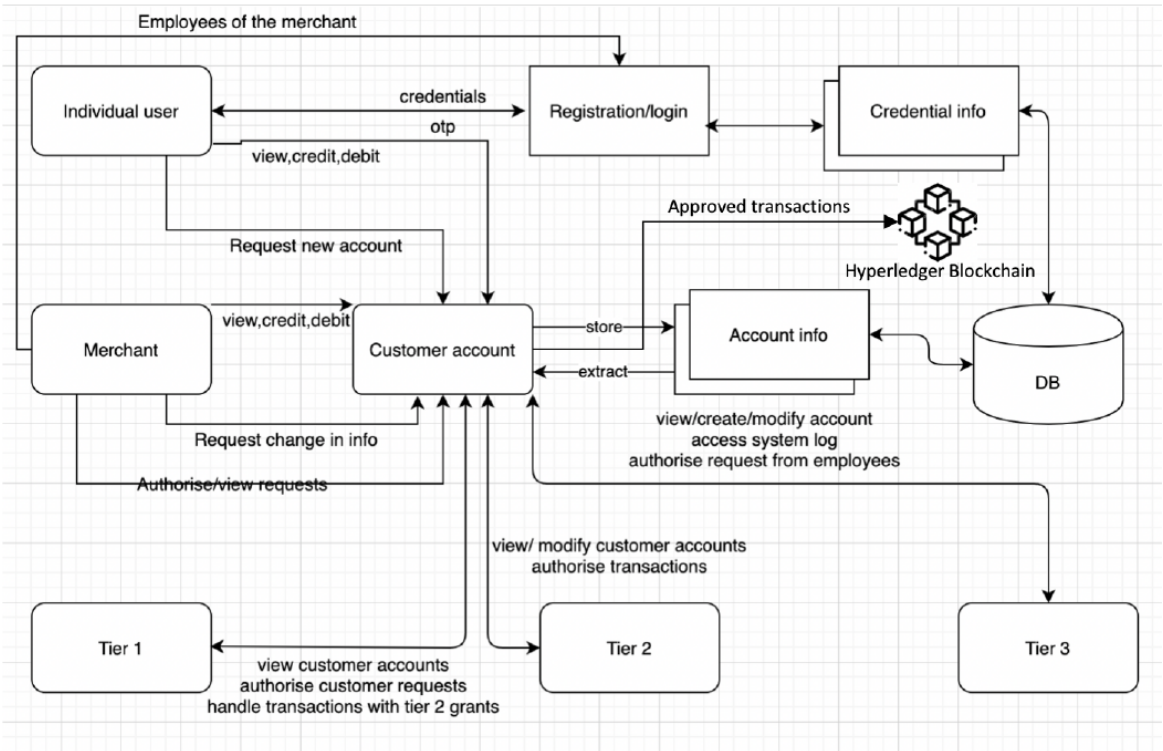

Figure 4.3: Detailed Flow Diagram

user associated with an account, etc. is stored in the database.

5. Transaction Details of every transaction initiated - Record of all the transactions needs to be maintained. Hence, all the transaction details would is stored.

6. Appointment Details of all users - All the details regarding the appointment requested and the details of the appointment are stored in the database.

7. Login History of all users - Login history of all the users along with the details like last login time, ip address, etc. are stored in order to keep track of user activity and avoid security issues from unknown sources.

8. OTP - OTP generated for various users and their specific activity is stored in the database and is matched when the user enters the OTP received through email.

Communication between database and backend is done using Hibernate framework supported by Java. By implementing hibernate, we have created a Java class for each table that will contain fields corresponding to columns in a table. Through this framework, we Are able to interact with the data in each of the tables.

4.3.2 Data Dictionary

The data of our system will be stored in different tables that are created by implementing database design principles. Below are the tables that will be created in MySQL database.

USER:

Column Name	Data Type
id	int
username	varchar(255)
password	varchar(60)
status	int
incorrect_attempts	int
created_date	datetime
modified_date	datetime
role	varchar(100)

USER DETAILS:

Column Name	Data Type
id	int
user_id	int
first_name	varchar(255)
middle_name	varchar(255)
last_name	varchar(255)
email	varchar(255)
phone	varchar(15)
address1	varchar(255)
address2	varchar(255)
city	varchar(255)
province	varchar(255)
zip	int
ssn	varchar(15)
question1	varchar(255)
question2	varchar(255)

TRANSACTION:

Column Name	Data Type
id	int
approval_status	boolean
amount	decimal(10,5)
is_critical_transaction	boolean
requested_date	datetime
decision_date	datetime
from_account	varchar(255)
to_account	varchar(255)
transaction_type	varchar(100)
request_assigned_to	int
approval_level_required	varchar(100)
level_1_approval	boolean
level_2_approval	boolean
approved	boolean
customer_approval	int

ACCOUNT:

Column Name	Data Type
id	int
user_id	int
account_number	varchar(255)
account_type	varchar(100)
current_balance	decimal(30,5)
created_date	datetime
status	boolean
interest	decimal(10,5)
approval_date	datetime
default_flag	int

CASHIER'S CHECK:

Column Name	Data Type
id	int
from_account_number	varchar(255)
first_name	varchar(255)
middle_name	varchar(255)
last_name	varchar(255)
transaction_status	varchar(100)
deposit_amount	decimal(30,5)

OTP:

Column Name	Data Type
id	int
otp_key	varchar(255)
initiator	int
creation_date	datetime
expiry_date	datetime
mode	int
completed	boolean
ip_address	varchar(255)

APPOINTMENT:

Column Name	Data Type
id	int
appointment_user_id	int
assigned_to_user_id	int
created_date	datetime
appointment_status	varchar(255)

LOGIN HISTORY:

Column Name	Data Type
id	int
username	varchar(255)
ip_address	varchar(25)
logged_in	datetime

4.4 Component Design

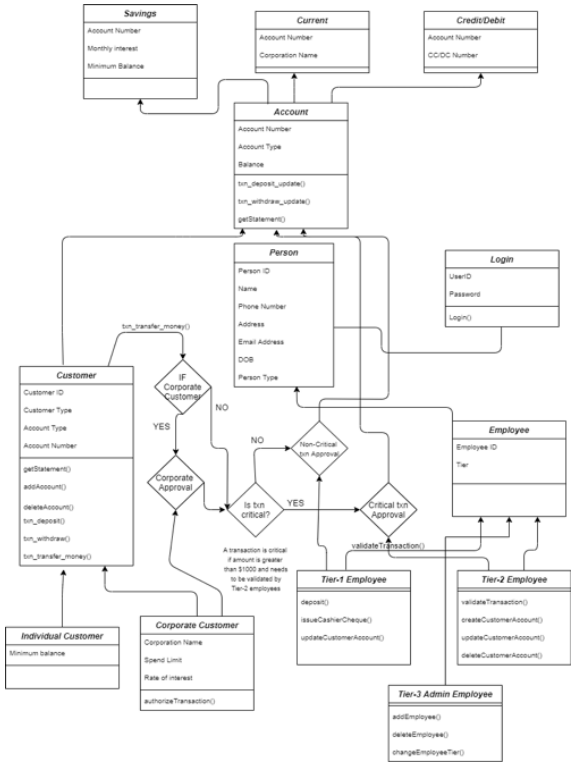

Figure 4.4: Class Diagram

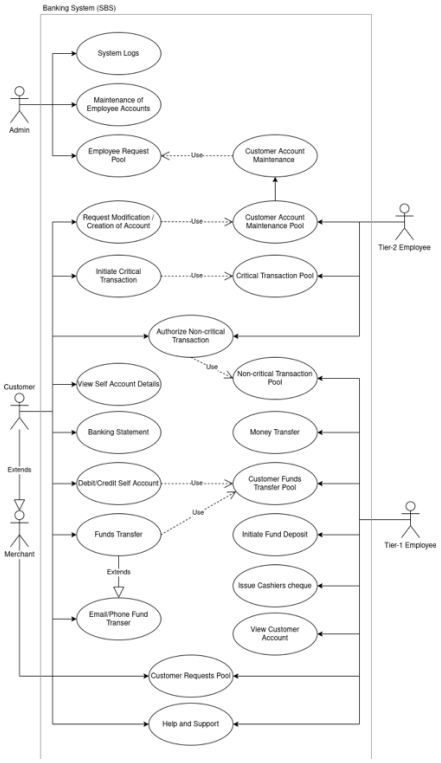

Figure 4.5: Use Case Diagram

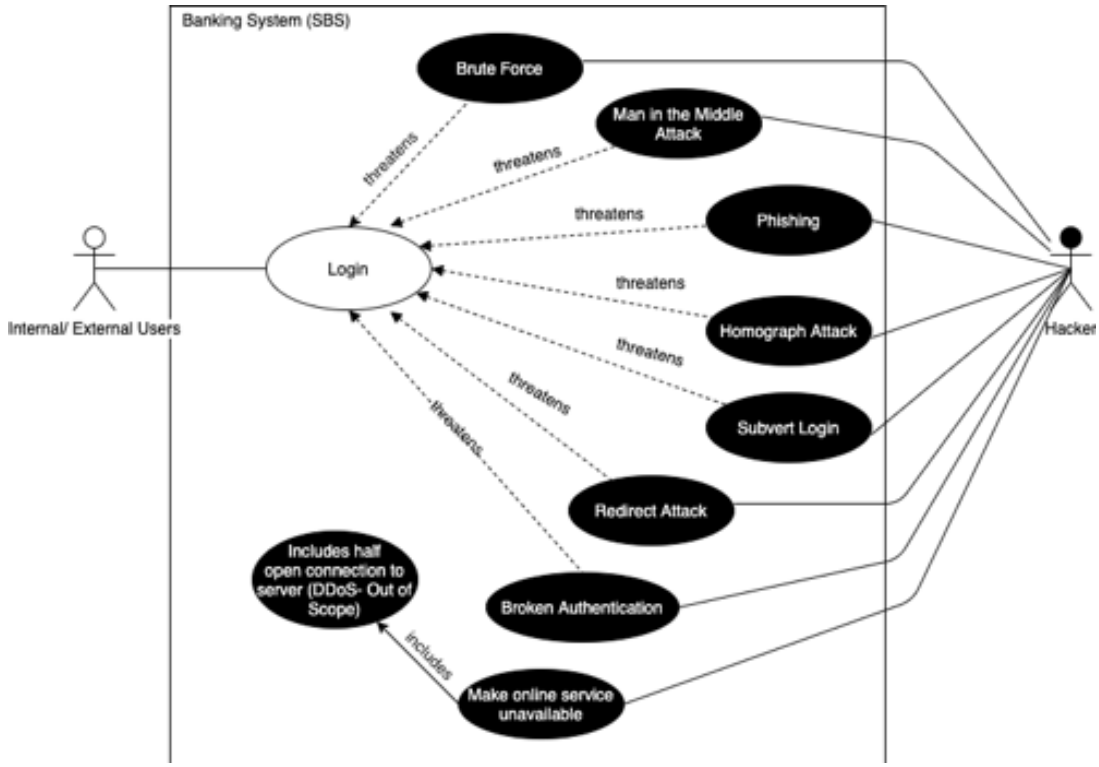

Figure 4.6: Misuse Case Diagram

Chapter 5

Contributions

- **Atin Singhal**- Being the group leader, was responsible for organizing meetings, submitting meeting minutes, drafting the work breakdown structure and having regular discussions with group members regarding the project status and implementations. Designed the misuse cases for the design document.

  Mainly responsible for the security implementations, hyperledger blockchain along with a few functional implementations. Added style sheets in the login page. Designed the error page for the application. Responsible for parts of Web Security Configuration file which include configurations about user sessions, access management etc. Implemented functionalities including system logs, user-wise IP logging, maintaining user login history, virtual keyboard for the OTP page. For security side, added PKI applications, responsible for creating Public Key Certificate, enforcing HTTPS redirection. Also created scripts for added security. Added blocking of back buttons, right click, copy/ cut/ paste functionality, text selection, etc. Used Hyperledger JavaSDK for integration with our application, which was further optimized by a few other group members. Responsible for creating smart contract for hyperledger platform and docker setup for the same.

- **Shivank Tiwari**- As the group's deputy leader, undertook responsibility for organizing meetings, submitting meeting minutes, drafting the work breakdown structure and having regular discussions with group members regarding the project status and implementations. These documentations were majorly reviewed and created by me along with team lead. Designed the responsibility traceability matrix, discussing scenarios of misuse

cases, use cases, class diagrams for the design document. Worked on making skeleton web pages to perform easy navigation between pages. Undertook work related to Security and Blockchain apart from working aside group members with their discussions and tasks. For blockchain part, I studied over its architecture and worked its applicability with team lead. For security related tasks, worked over different tools which can be of use for testing purposes like Arachni, Postman, Burp Suite etc. Tested the application for functional and security bugs mainly the employee oriented parts. Moreover added a support file to make compulsory abiding of JS on the user side. Worked for Public-Key Certificates, Self-Signed Certificates, Sign-In history, Security related configuration in session management etc.

- **Amitabh Das**- Responsible for the entire Spring framework for the project, with authorization controls and custom access handlers. He set up Malicious access controls by means of account locking and IP blocking. He also developed all the validation code for all the form inputs at back-end, also ensuring prevention of double form submission by implementing Post-Redirect-Get pattern. He implemented all of the core transaction logic - approval of critical transactions, customer authorizations and declines of transfer requests, cashiers cheques, money transfer, credit, debit, email funds transfer, etc which was used by Tier1, Tier2 and Customer code. He also implemented a mailer with thread-pooling capability to send out OTP. The database schema, the hibernate JPA classes and database session management code, along with encryption of sensitive data in the database was designed and developed by him.

  He also setup the AWS and RDS instance to deploy the application and database. The self-signed certificate was generated by him for use with nginx. He also wrote nginx scripts to redirect HTTP traffic to HTTPS, make use of security headers like STS. The application was then tested by him using JMeter to confirm capacity of 500 requests/sec. He also successfully developed and deployed a scaled-down version of hyperledger blockchain to capture all valid transactions.

  He played an important role in the continuous integration of the project, by setting up GitHub actions for sanity checks of all pull requests. He designed a code development road-map and assigned tasks to each of the members. He also single-handedly reviewed and tested all of the pull-requests and fixed all of the merge-conflicts.

- **Uttam Bhat**- Having worked in web development for a couple years and having an

understanding of a web application he was responsible laying out the system architecture in the design document. He was soley responsible for Customer functions part of the code. this includes frontend and backend.Including setting an account as primary, creating new account -savings,credit,checking; Implemented otp for setting new account and scheduling an appointment; Contributed towards organising the front end code and explaining to teammates what needs to be done on that front. Implemented deposit and withdrawal; view transactions and everything under the customer part. Attended meeting in the beginning to understand the requirements and put together an idea for the project..He also played an important role in discussions related to the design of the project,expected behavior and Integrating the components to make the product running.

- **Sriprashanth Ramamoorthy**- Having prior experience of full-stack development and knowing to view an application as a whole, he created the Sequence Diagrams for user authentication and Banking functionality in the design document. He was solely responsible of handling the stories related to Employee Management and Bank Account Management across all the fronts(frontend, backend and database). He worked on the handling the all the functionalities related to User Bank Account :View/Search Customer Bank Accounts,Bank Account Approval and Delete Bank account of external customer. He was responsible for all the employee Management stories - Update external and internal user Information, View/Search Employee Information ,Create new internal employees ,Delete existing internal employees. URL Protection for the Employee and Account Management to protect the endpoints from unauthorized access was also handled by him. The functionality to schedule and view appointments was also developed by him on all the fronts.He also played an important role in discussions related to the design of the project,expected behavior and Integrating the components to make the product running.

- **Raj Buddhadev**- Responsible for implementation of the transaction features like deposit/withdraw money, cashier's check and authorization of the transactions. These features were implemented for tier1 employees, tier2 employees and customers. Along with the implementation of the transaction feature, he created JSPs for tier1 employee and integrated functionalities on back-end with the front-end code. He also contributed in integrating hyperledger to application by setting up a customized version of hyperledger implementation which was in sync with our application. He also played an important role

in testing the system to find functional and security vulnerabilities and bugs. He was able to find several issues in the system and the code, out of which some were fixed by him and others were fixed by other team members.

- **Ayush Ray**- Having prior industry experience as an Information Technology Consultant and knowledge in UI designing from his curriculum courses, he was responsible for looking over the front-end part of the project. Worked on dashboards and code for the login page. Laid the design for the pages and coordinated with other members explaining them the requirements and the process and helped them start. Also responsible for implementing the sign-in history logging function, another requirement for the project. In the initial system design phase, he coordinated with the members responsible for developing back-end to design the class diagram. Attended the group meetings to gather the requirements from time to time.

- **Kangjian Ma**- As a member of a large group, Kangjian is actively involved into the group discussion and assignment. In the system design phase, Kangjian draws system component relationship graphs. In the system implementation phase, Kangjian is assigned with the research and demonstration of Hyperledger platform implementation and front-end dashboards implementation in JSP. Kangjian Ma is mainly responsible for User Guide documentation after the implementation phase and participate in the system testing and debugging process. In general, Kangjian Ma follows the group assignment and was actively involved in the whole process of the system development, based on group need and my skillset background.

Chapter 6

Vulnerability Report

6.1 Valid Security Vulnerabilities

	Vulnerability	**Reported By**	**Fix**
1.	Users are able to deposit the same cheque multiple times.	Manjeshwar Ashish Padakannaya, Ankit Sharma, Yash Vijay, Deepan Karthik Nedumaran	Record the use of a cheque to prevent future use.
2.	Users can deposit any cheque number.	Manjeshwar Ashish Padakannaya, Ankit Sharma, Yash Vijay, Deepan Karthik Nedumaran	Associate a cheque with a target user so that only the intended user can deposit a cheque.
3.	Employees can use the portal even after the admin deletes an employee.	Manjeshwar Ashish Padakannaya	Update deleted employee session in Spring Security when an Admin deletes an employee.
4.	XSS vulnerability in system logs	Karan Ajith, Prasoon Deva, Manjeshwar Ashish Padakannaya, Mojca Stampar	Sanitize and escape text being rendered in JSP.

5.	Session Management	Karan Ajith, Yasaswini, Saumil Dixit	Configure Spring Security properly to prevent multiple sessions of a user at a time.
6.	Cookie Hijacking.	Karan Ajith, Yash Vijay	Change the value of the cookie with each and every request.
7.	Prevent Malicious logins	Karan Ajith, Alyssa Goldstein, Yucheng Lu, Deepan Karthik Nedumaran, Athul Pramod, Akshay Shah	Use a captcha in the frontend and IP/account blocking at the backend

6.2 Valid Functional Vulnerabilities

	Vulnerability	**Reported By**	**Fix**
1.	Employees can't view appointments of the day.	Saumil Dixit, Manjeshwar Ashish Padakannaya, Yash Vijay, Vasudev Sridhar, Yucheng Lu, Deepan Karthik Nedumaran	Display all future appointments, not just the current day appointments.
2.	Employee creation issue.	Manjeshwar Ashish Padakannaya, Ankit Sharma, Mojca Stampar, Yash Vijay	Provide
3.	Employee unlock issue.	Manjeshwar Ashish Padakannaya, Yash Vijay, Ankit Sharma	Provide an interface to the admin to unlock locked Employee accounts.

4.	Regex not accepting valid high school and last name.	Manjeshwar Ashish Padakannaya, Ankit Sharma	Revise regex to take all correct inputs.
5.	Account dropdown shows locked accounts of the viewing user.	Yash Vijay, Ankit Sharma, Sagar Parekh, Deepan Karthik Nedumaran	Don't display locked accounts in dropdown.
6.	Unlocking user accounts not associated with the user.	Yash Vijay, Ankit Sharma	Display the user name associated with the locked account in the UI to identify the acocunt being unlocked.
7.	Site shows improper error messages/ Doesn't explain password criteria clearly/ Poor feedback/ Username gives error even though only alphabet are used.	Yashaswini Addada, Manav Bagai, Prasson Deva, Saumil Dixit, Yucheng Lu, Vasudev Sridhar, Manjeshwar Ashish Padakannaya, Sagar Parekh, Athul Pramod, Akshay Shah, Ankit Sharma, Yash Vijay, Mojca Stampar, Karthik Radhakrishnan, Deepan Karthik Nedumaran, Sagar Parekh	Fix error message to clearly explain the requirements. There is length requirement for most fields.
8.	Unlock account requests are not generated for merchant accounts.	Ankit Sharma, Yash Vijay	Display locked merchant accounts in the UI so that Tier 2 can unlock those accounts.

9.	Unable to use new accounts other than the existing ones.	Yashaswini Addada, Manav Bagai, Prasoon Deva, Saumil Dixit, Yucheng Lu, Vasudev Sridhar, Manjeshwar Ashish Padakannaya, Sagar Parekh, Athul Pramod, Akshay Shah, Ankit Sharma, Yash Vijay, Mojca Stampar, Karthik Radhakrishnan, Deepan Karthik Nedumaran, Sagar Parekh	Use account numbers as strings instead of integers to prevent interger overflow. Due to this, the correct account number was not being passed to the backend.
10.	Virtual Keyboard	Karthik Neeragunda Radha Krishnan, Sagar Parekh, Deepan Karthik Nedumaran	Use an appropriate virtual keyboard in the frontend.

6.3 Invalid Vulnerabilities

	Vulnerability	**Reported By**	**Comment**
1.	Some users may not have middle name which is made as compulsory for the registration/ When creating new employee, Address2 and middle name have to be populated, even though some people don't have those.	Yasaswini Addada, Yucheng Lu, Mojca Stampar	Not a major flaw or a broken functionality. You can fill it with any character N/A for testing purposes.

2.	The users and employees are not able to request modification of their personal details.	Yasaswini Addada, Sagar Parekh	Design choice. You can schedule an appointment and an employee will help you with updation of the details.
3.	Session Management Error	Karan Ajith	You can't login as two different users simultaneously in the same browser.
4.	No 2FA for password change	Karan Ajith	Not a mandatory requirement.
5.	Cookie Hijacking	Karan Ajith, Yash Vijay	We're enforcing HTTPS and using HTTPOnly for Cookies. Headers are encrypted when transmitting over the network. No evidence provided. Copying cookie/ session id from your own browser isn't classified as cookie hijacking.
6.	System logs show code errors in raw format	Karan Ajith	Logs can only be accessed by the Admin.
7.	Session continues even if the browser or tab is closed by the user.	Prasoon Deva	Session invalidation time has been set to 5 minutes. Once you visit the homepage of bank, other sessions on the system are invalidated.

8.	Can unlock an account different from which I clicked.	Prasoon Deva	No screenshot or video evidence provided.
9.	Server SSL certificate is not trusted.	Prasoon Deva	It's a self signed certificate. The project requirements deem this as valid.
10.	nginx before versions 1.15.6 and 1.14.1has a vulnerability in the implementation of HTTP/2 that can allow for excessive memory consumption or CPU usage.This issue affects nginx compiled with the ngx_http_v2_module (not compiled by default) if the 'http2' option of the 'listen' directive is used in a configuration file.	Prasoon Deva	No attacks shown. Can't assume attacks.
11.	Open ssh port 22 on the server which is vulnerable to attack from the outside. Attacker can use brute force method to gain access	Prasoon Deva	SSH enabled only using key file.
12.	Security Questions insufficient.	Alyssa Goldstein	We're using OTP for verification where ever necessary.

13.	I already register on account use SSN 000-00-0000, then I create another account also use this SSN number. I can register successful. However, one user only have one SSN number.	Yucheng Lu	Not possible. We have set SSN as UNIQUE in Database. No evidence provided.
14.	Addresses are accepted even when no alphabets are entered. Takes an input which is entirely numbers./ Accepts 1 digit zip codes as input.	Yucheng Lu, Alyssa Goldstein	Not a requirement according to SRS to validate the address for correctness.
15.	I enter zip code as a 21 digit number (852811739134718371499), it's allowed. However, there is no zip code with so many digits. It may cause Fake user.	Yucheng Lu	The zip code field in DB is set as INT and the max input it can receive is 10 characters, 2147483647 to be precise. No evidence shown.
16.	User can login without OTP.	Yucheng Lu	Not a mandatory requirement.
17.	Login history time is incorrect. I login Apr.09, but login history show that 04-10	Yucheng Lu	Server time is logged, not user's local system time.
18.	Search account function is not working about employee account.	Yucheng Lu	The field clearly states to enter account number but you are trying to input username.

19.	When the registration is completed, my mobilephone and email will not receive any activationcode. This will cause the attacker to write ascript for automatic registration. In this way,there will be an equivalent account waiting forthe employee to activate, and the speed of theemployee identifying and activating the accountis much slower than the speed of the script.	Yucheng Lu	Many fields are set as unique for the form to be successfully submitted. No attack shown. Can't assume attacks.
20.	One user can open multiple accounts of the same type.	Manjeshwar Ashish Padakannaya, Ankit Sharma, Yash Vijay	SRS didn't say that users can't have multiple accounts of same type.
21.	Always says user present. Does not let me register. But when I go to forgot password it says if user present you will get an email. Never received any email.	Sagar Parekh	'User already exists in the DB' means username/ phone no./ email/ ssn were matching with any other user. Screenshots show same value of SSN for all your FAILED attempts.
22.	No option for new employee sign up.	Sagar Parekh	Employees can only be created by admin.

23.	For customers, Deposit/ Withdraw money does not wait for approval from the employees. Anyone can process any amount from their bank account like this.	Sagar Parekh	This page was just added so you and others could easily add funds to check functionalities.
24.	No option to download logs.	Sagar Parekh	Not mandatory according to requirements. Design choice.
25.	The form data (username and password) is visible in the console. / Sending password as plaintext.	Athul Pramod, Vasudev Sridhar	Console is visible only to the user. No evidence of exploitation shown. HTTPS is enforced and Headers will be encrypted before transmission. Server-side encryption is done.
26.	System crashed when typing random characters in the Amount field in the create, transaction for Tier 1 employees.	Athul Pramod	It's taking you to an error page. Error pages are common to make sure no unwanted information is displayed. Amount expects digits, not alphabets.
27.	The system is not imposing any limits on the number of wrong OTP attempts.	Athul Pramod, Akshay Shah, Deepan Karthik Nedumaran	OTP is tied to a session and an action. So only a legitimate customer can be able to use an OTP.

28.	I am able to make an appointment for any day and the system does not show the availability of the employe while making the appointment. Also the system allows appointments during sundays thus not considering the availability of the employees.	Athul Pramod	Not a mandatory requirement. Each bank policy is different. Maybe employees can attend to customers over 'Zoom' in our case.
29.	Change Password: Can change the password for the user to the password previously used.	Akshay Shah	Not a mandatory requirement. No exploitation shown.
30.	No way to change the role of an employee.	Ankit Sharma, Yash Vijay	Couldn't find any such requirement. Admin can change the roles.
31.	No user name or id is provided to tier 2 employees for unlocking locked accounts. / New Account Approval does not show employee details, hence just by account number you cannot know whose account is being unlocked.	Ankit Sharma, Yash Vijay	It's just the design followed to respect privacy of the user. No such requirement according to the SRS.

32.	Able to view the account number even before approval in Cashiers check page/ Unapproved accounts can be used for ordering cheques/ Accounts being reflected even when not approved.	Ankit Sharma , Yash Vijay, Deepan Karthik Nedumaran	The evidence just shows a list of accounts. No exploitation is shown. As accounts are not approved, it'll throw an error.
33.	Not able to create account for customer. Account creation requires OTP. In order to receive OTP, email ID must be update.	Vasudev Sridhar	The email ID must be updated to a valid one before requesting an OTP.
34.	Transfer Funds Not Working - 2: When I transfer funds externally, the amount is not getting removed from account.	Yash Vijay	The screenshots show 'to' and 'from' account as same.
35.	Not all provided users actually work.	Mojca Stampar	All credentials were tested before sharing Someone might've have changed them while testing.
36.	Every time you go back to main page from user account it logs you out.	Mojca Stampar	That's a good thing, you navigate away, it logs you out.
37.	Debit and credit for more than $1000 not asking for approval.	Karthik Neeragunda Radha Krishnan	No evidence shown. Unable to recreate issue.

38.	Able to send money to individual when clicked on Merchant pay i.e account not validated as merchant or individual.	Deepan Karthik Nedumaran	No evidence provided.
39.	Number of attempts is 5 but on the approval page it shows 4.	Deepan Karthik Nedumaran	No evidence provided.
40.	Expects authorization for paying merchant(self authorization?)	Deepan Karthik Nedumaran	No evidence shown. Also, not clear.
41.	On existing customer accounts, debit/credit doesn't get updated in transaction records.	Deepan Karthik Nedumaran	No evidence provided.
42.	DOB is accepting dates from future.	Yasaswini Addada, Alyssa Goldstein	No evidence shown. Unable to replicate.
43.	Does not accept phone number in the specified format. [The phone number in screenshot is +1(129)-764-8376]	Alyssa Goldstein	The phone number you're trying is invalid. There's no US state area code that starts with 1xx.
44.	Not able to update details of a customer/ employee. Username field is disabled.	Vasudev Sridhar	Username field is disabled for safety so users can't change it.

Chapter 7

User Guide

7.1 Overview of the User Guide

7.1.1 Purpose of the User Guide

This User Guide is designed to assist the online bank users, including bank employees and customers, to be familiar with the developed secure online banking system. Step by step, we will show each page and functionalities for different users .Moreover,the security features will be explained and described in detail to assure user's safety to use the online banking system.

7.1.2 How to Use the Guide

In chapter 7.2, the guide will describe the requirements to use the online banking system. In Section 7.3, an overview of the secure banking system will be introduced including the security features. In section 7.4, the guide will describe the use for users with different roles, including tier-1 employee, tier-2 employee, tier-3 employee, customer and merchant. First of all, the login page, which is shared by all users, will be described. Then the user can look into different specifications according to his/her role to grasp the design, feature and functionality of the user's subsystem.If a user has any further question or concern,a technical help and support team can be reached at cse598group13@gmail.com.

7.2 Requirements

The secure banking system is deployed on AWS which has all necessary hardware to host the server. The user just needs a qualified operation system and browser to visit the server as a client.

- Operating System Requirements
  - Apple OS X or above
  - Ubuntu 14.05 or above
  - Microsoft Windows 7 or above
- Browser Requirements
  - Safari 6.2.9 or above
  - Internet Explorer 10 or about
  - Google Chrome 45.0 or above
  - Mozilla Firefox 40.0 or above

7.3 Overview of the Secure Banking System

The secure banking system is hosted on a virtual machine and be 24*7 available to customers and employees online through a web browser. Two kinds of users are hypothesized to use the system, categorized as internal users and external users. The internal users can be classified into 3 groups with gradually higher privileges, i.e. tier-1 employees, tier-2 employees, and tier-3 employees. The external users include individual customers. Three kinds of accounts, i.e. checking accounts, saving accounts and credit accounts, are designed as in the real world. Basic functionalities are developed to allow users to create, view, modify, close accounts, debit, credit, and transfer money, request, grant and decline authorization, view log-in and transaction history or bank statement in a secure environment.

The system can be accessed at https://3.19.26.35/

7.4 Online Banking Interfaces

7.4.1 Login

Login page is the entry to the online banking system. Users including internal employees and external customers need to log into the system by entering his/her registered user name and password shown in Fig. 7.1. Once the credentials are validated, the user will be directed to the dashboard based on their role(customer, admin, tier-1, etc). So, all users, internal or external, associated with the bank shall login through the same page. Prospective customers who are interested in opening a new account can also register into the bank from this page.

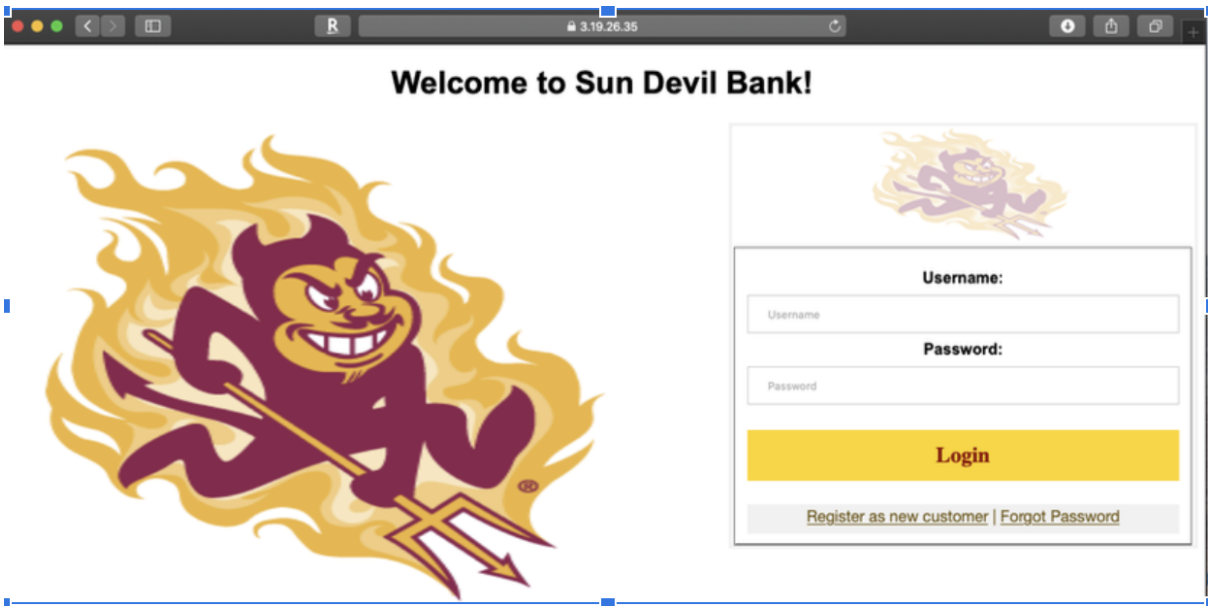

Figure 7.1: Login Page

If this is your first time using the system, you can create your external user account information on the Registration Page (Fig.7.2) by clicking the "Register as new customer" button on the Login Page. The required information includes Customer Type, Desired UserName, Email, Password, Name, Address, Phone Number, Date of Birth, SSN, City, Province, Zip Code. After inputting all the required blanks, you finish registering a new external user by clicking the "Register" button at bottom.

OTP is sent by email if you want to reset your account password. The Forgot Password Page is shown in Fig. 3.

Figure 7.2: Registration Page

Figure 7.3: Forgot Password Page

7.4.2 Internal Users

There are 3 kinds of internal users, including Tier-1 employee, Tier-2 employee, and Tier-3 employee with increasing privileges.

Tier-1 Employee

The Tier-1 Employ's Dashboard is shown in Fig. 7-3. From the dashboard, tier-1 employ can approve/decline non critical transaction (less than 1000 dollars) as in Fig 7-4, create non critical transaction as in Fig. 7-5, issue cashier cheque as in Fig.7-6, deposit cash cheque as in Fig. 7-7 deposit money as in Fig. 7-8, withdraw money in Fig. 7-9, view customer accounts in Fig. 7-10, change password in page 7-11, view appointments of the day in page 7-12. images

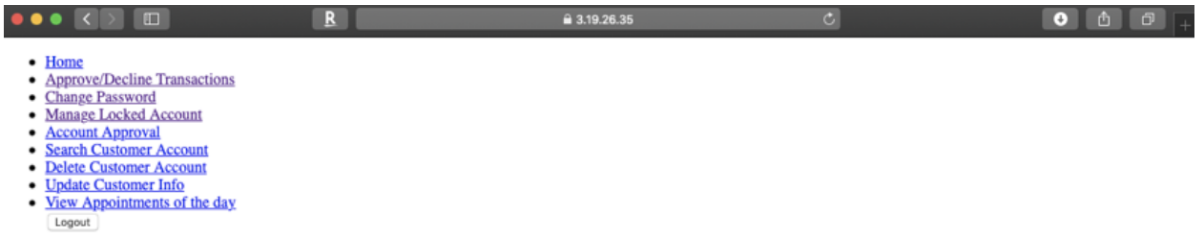

Figure 7.4: Approve/Decline Transaction Page

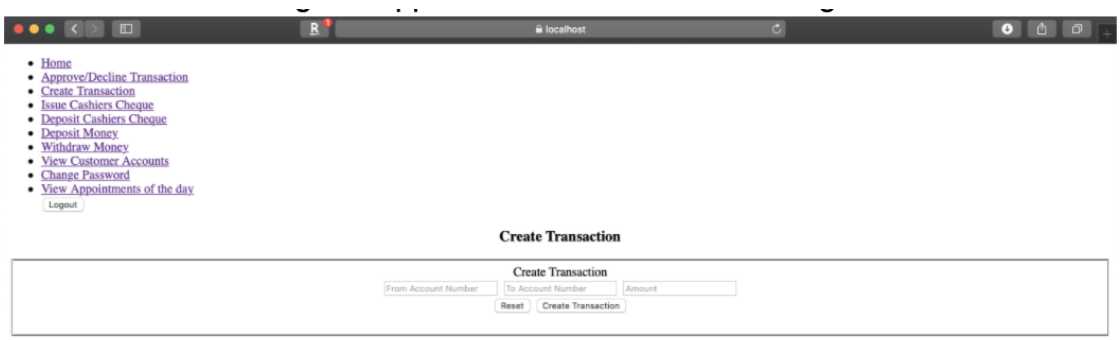

Figure 7.5: Create Transaction Page

Tier-2 Employee

A Tier-2 Employee(Fig 7.13) can approve/decline critical transactions(greater than 1000$)(Fig 7.14). view(Fig 7-15),modify(Fig 7.16) and delete accounts(Fig 7.17) of the customers.A Tier-2 Employee also has the privileges approve and decline account creation requests (Fig 7.18).A Tier-2 Employee can also change his own password (Fig 7.19).The appointment schedules to the Tier-2 Employees for that day can be viewed(Fig 7.20).A Tier-2 Employee can also manage the locked accounts (Fig 7.21).

Tier-3 Employee (Admin)

A Tier-3 Employee(Fig 7-22) can view(Fig 7-23), create(Fig 7.24),modify(Fig 7.25) and delete employees(Fig 7.26) from tier-1 and tier-2 .A Tier-3 Employee also has the privileges to view the System logs (Fig 7.27) in case of any internal issues.A Tier 3 Employee can also change his own password (Fig 7.28).

7.4.3 External Users

Customer

Fig.7-29 shows the dashboard of the customer account. Customers can find the account information including savings account, checking account, and credit account. For each account,

Figure 7.6: Issue Cashier Cheque Page

Figure 7.7: Deposit Cashier Cheque Page

Figure 7.8: Deposit Money Page

Figure 7.9: Withdraw Money Page

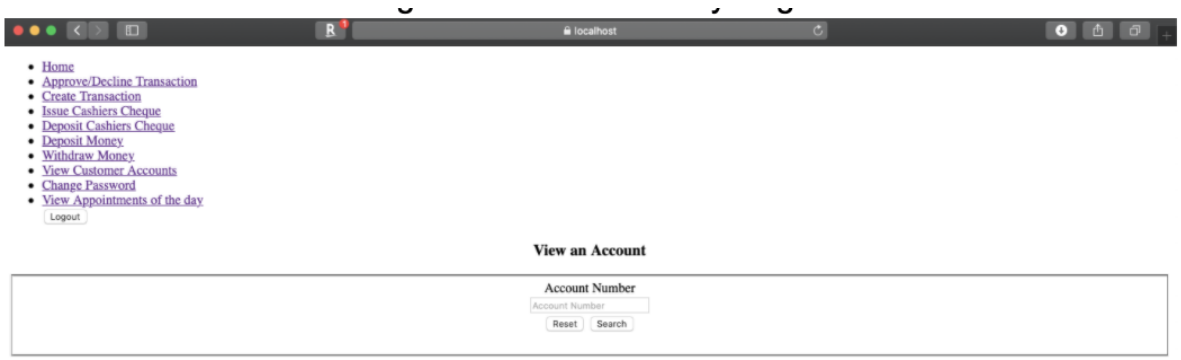

Figure 7.10: View Customer Account Page

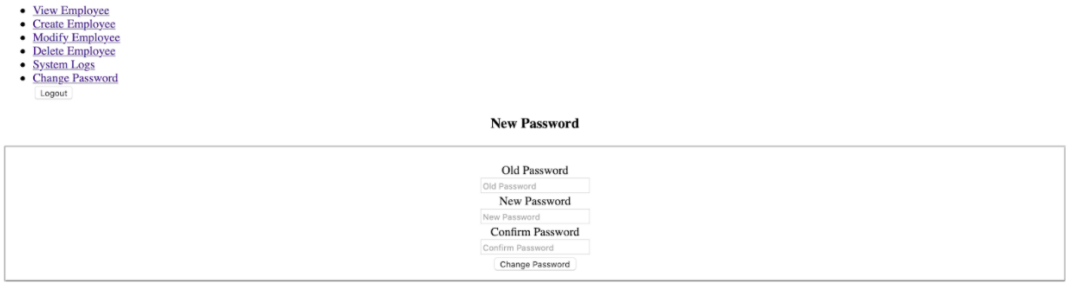

Figure 7.11: Change Password Page

customers can deposit/withdraw money (Fig 7-30), view recent transactions (Fig 7-31), and transfer funds(Fig 7-32).

7.5 Sample Users

Using the SystemTo get started, the database has been preloaded with sample Tier 1, Tier 2 and Admin employees. These profiles can be used for the purpose of testing our banking system. tables

Note All the above profiles have 1234 as their passwords, for the sake of testing. More customer and employee profiles can be created. The new customer profiles you create have to be approved by a Tier 2 employee before you can start using them. Same goes for the employee profiles, except that these will be created by the admin. A new customer doesn't have any bank account linked by default. You must request for a bank account number by placing a request for an additional bank account. The same must then be approved by a Tier 2 employee. (Hint: New accounts show up on the bottom- in the Account Approval tab)

Figure 7.12: View Appointment of the Day Page

Figure 7.13: Tier-2 Employee Dashboard

Figure 7.14: Approve/Decline Transactions Page

Figure 7.15: Change Password Page

7.6 Possible Errors Troubleshooting

If you are greeted with a page, as shown above. You need to install the certificate. To do that, follow these steps:

Profiles

User Name	Created Date	Modified Date	Incorrect Login Attempts	
bailey27	1982-11-22 04:24:43.0	1981-05-07 03:17:48.0	6	Unlock
leuschke.jodie	2007-11-15 05:46:27.0	1983-10-01 11:39:38.0	9	Unlock
dooley.darby	2014-02-28 10:41:13.0	1985-10-13 04:43:09.0	5	Unlock
kuhn.erika	2014-12-29 05:49:43.0	2012-10-16 21:32:58.0	1	Unlock
towne.sammie	2012-08-26 01:38:03.0	1992-11-11 14:43:11.0	8	Unlock
lemke.kyleigh	1996-02-02 17:24:30.0	1978-03-18 06:18:19.0	6	Unlock
barton.mae	1993-07-06 00:33:00.0	1977-09-01 17:37:09.0	0	Unlock

Figure 7.16: Manage Locked Account Page

- Home
- Approve/Decline Transactions
- Change Password
- Manage Locked Account
- Account Approval
- Search Customer Account
- Delete Customer Account
- Update Customer Info
- View Appointments of the day

Logout

Account ID	Balance	Status		
4555266457458	20993.91000	Pending	Authorize	Decline
4556483743521342	45445.90000	Pending	Authorize	Decline
5174903381802945	8259.21000	Pending	Authorize	Decline
4911534786759215	12177.14000	Pending	Authorize	Decline
6011365094409063	12039.35000	Pending	Authorize	Decline
5123163392888266	12519.05000	Pending	Authorize	Decline
4024007135233760	28830.65000	Pending	Authorize	Decline
6011200923559032	7114.21000	Pending	Authorize	Decline
5402053575367734	28283.50000	Pending	Authorize	Decline
370605648482856	3799.71000	Pending	Authorize	Decline
4539309266608	34428.61000	Pending	Authorize	Decline
5513164323818555	1303.45000	Pending	Authorize	Decline
5381527919175434	29094.44000	Pending	Authorize	Decline
5126090228173787	14155.70000	Pending	Authorize	Decline
6011524565957127	39332.87000	Pending	Authorize	Decline
4532167720665706	44197.22000	Pending	Authorize	Decline
4539668143480	47003.06000	Pending	Authorize	Decline

Figure 7.17: Account Approval Page

- Click on Non-Secure.
- Download the certificate file.
- If you're on Windows, open the certificate file and click install.
- If you're on Mac, import the certificate in the Keychain and make sure 'Always Trust' is selected.

You should now be able to access the site with no issues.

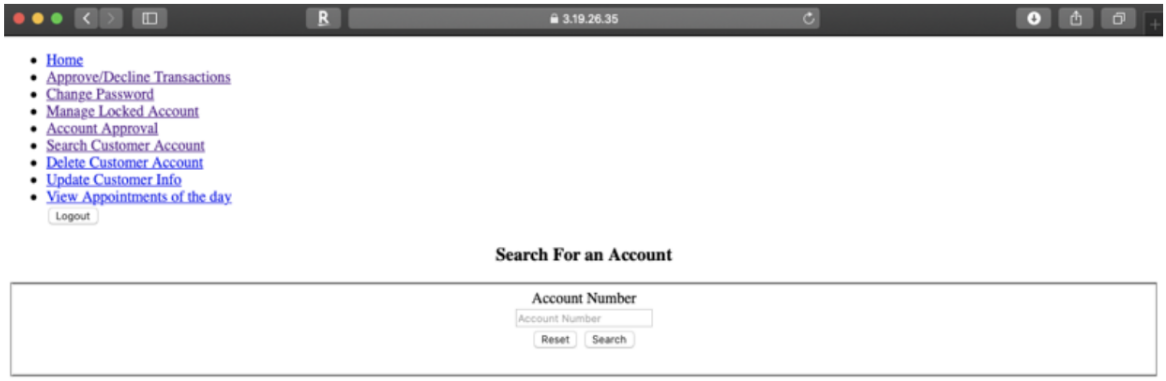

Figure 7.18: Search Customer Account Page

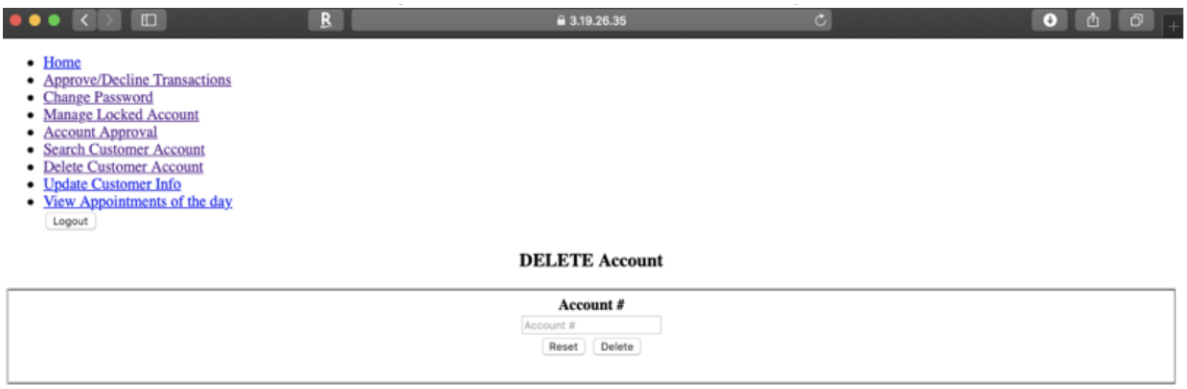

Figure 7.19: Delete Customer Account Page

Figure 7.20: Update Customer Info Page

Figure 7.21: View Appointment of the Day Page

Figure 7.22: Tier-3 Admin Dashboard

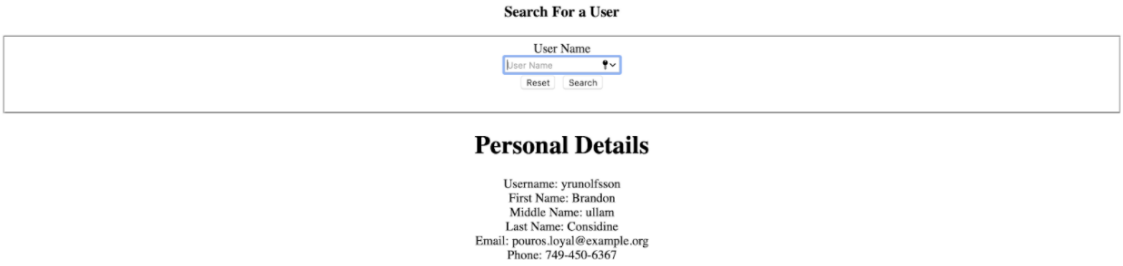

Figure 7.23: Tier-3 Admin View Employee

Figure 7.24: Tier-3 Admin Create Employees

Figure 7.25: Tier-3 Admin Update Employee

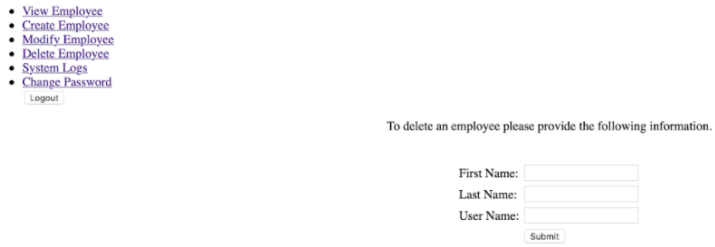

Figure 7.26: Tier 3 Admin Delete Employee

Figure 7.27: Tier 3 Admin Sys Logs

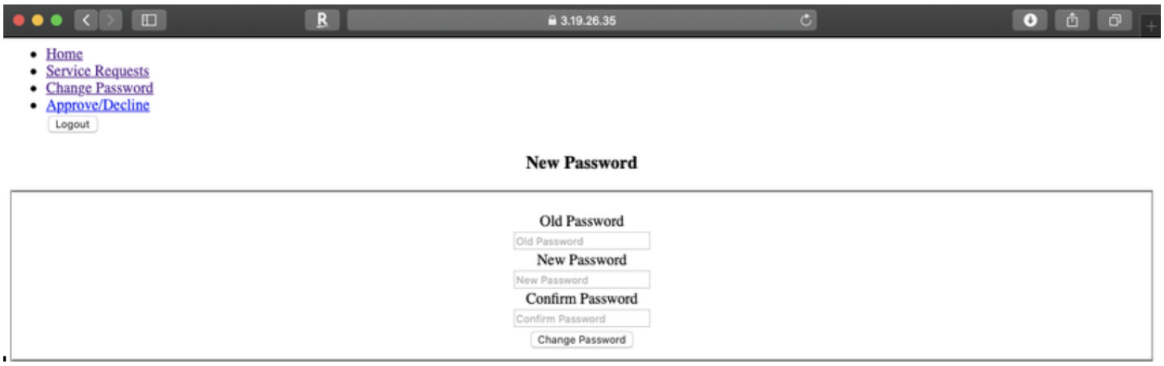

Figure 7.28: Tier 3 Admin Update Password

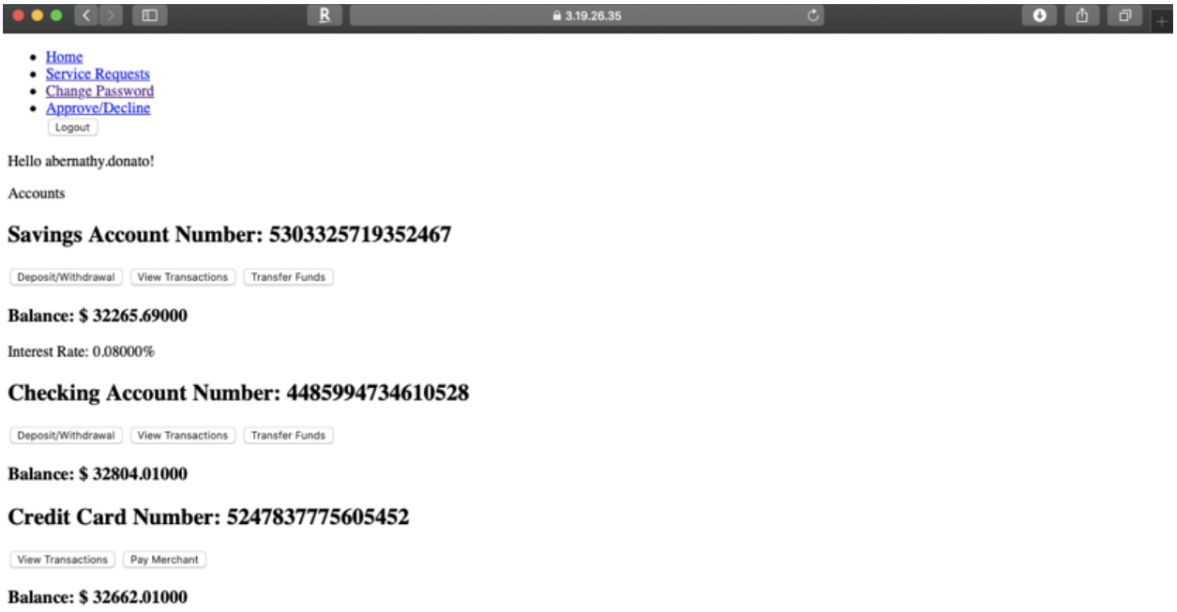

Figure 7.29: Customer Dashboard Page

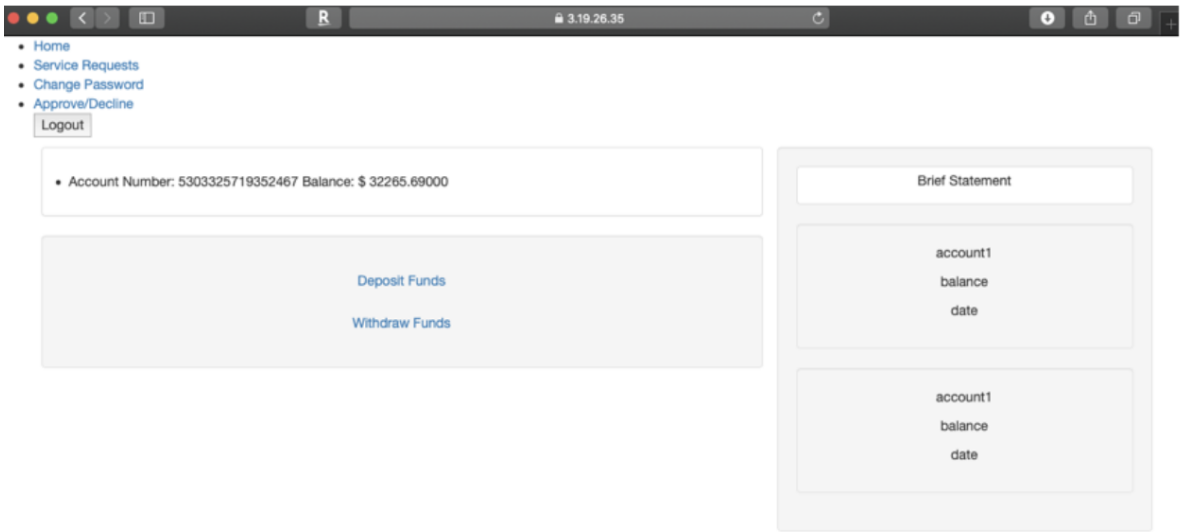

Figure 7.30: Deposit/ Withdraw Money Page

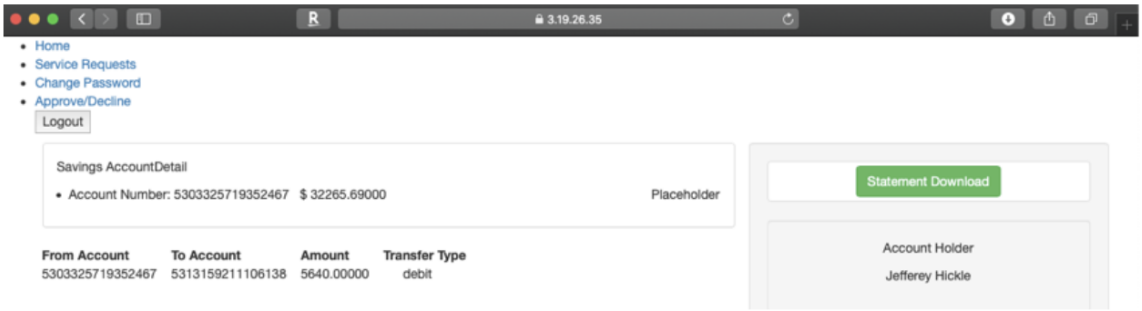

Fig 7-3 View Transactions Page

Figure 7.31: View Transactions Page

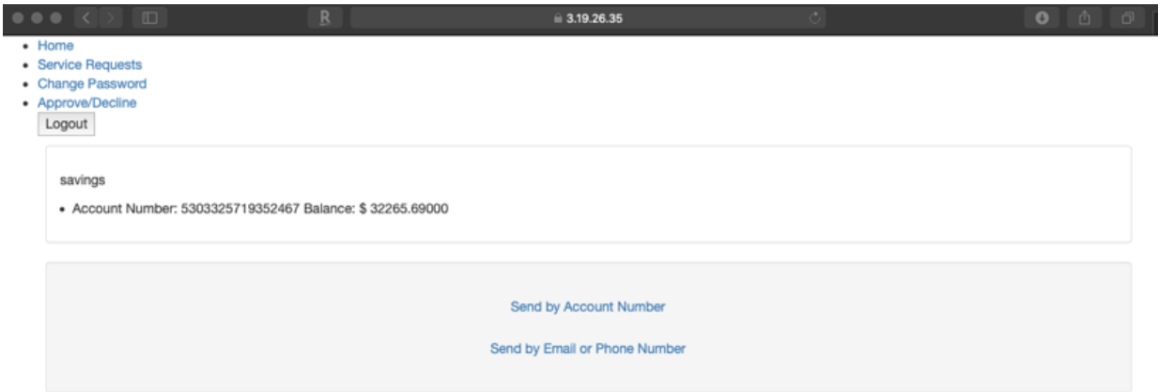

Figure 7.32: Transfer Funds Page

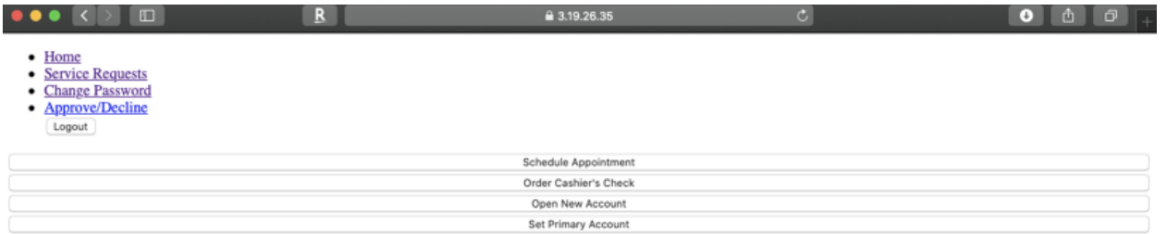

Figure 7.33: Service Request Page

Figure 7.34: OTP Page for Scheduling an Appointment

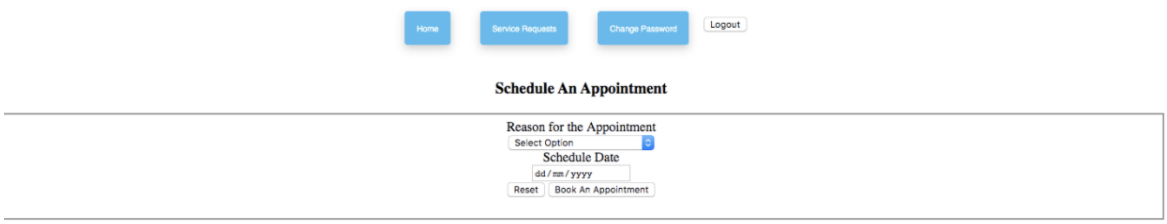

Figure 7.35: Page for Scheduling an Appointment

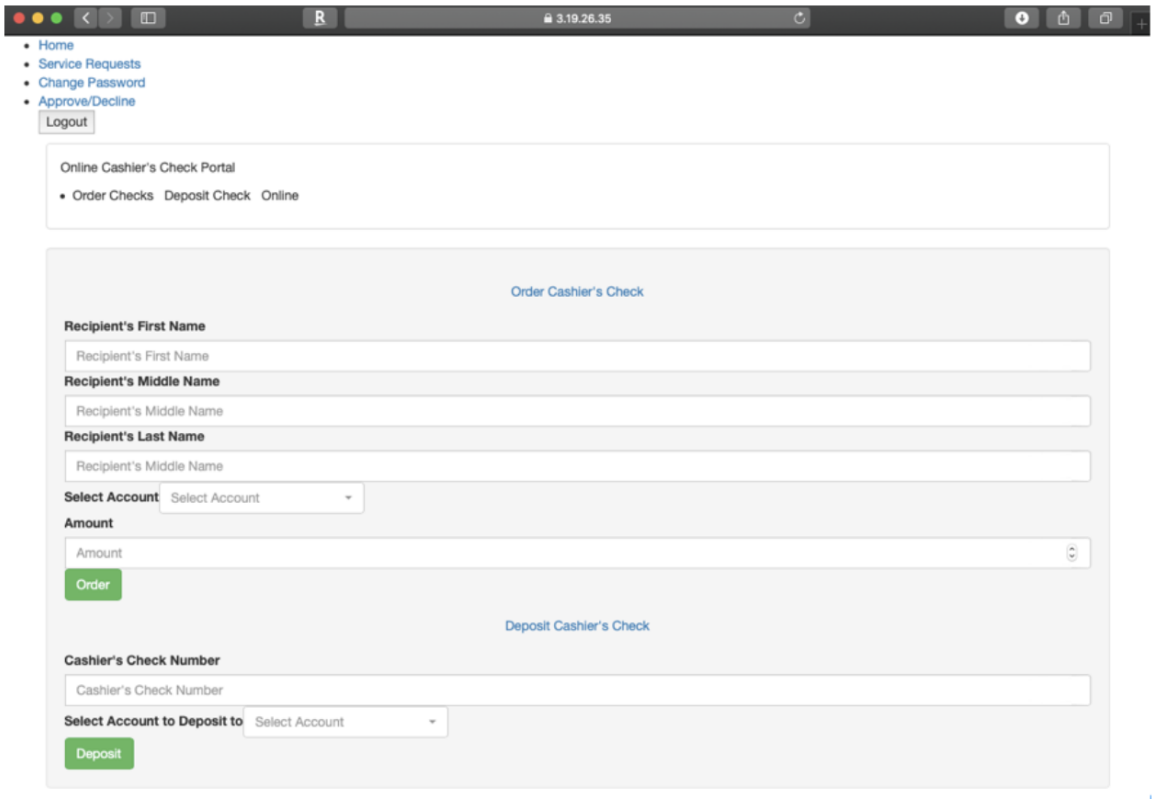

Figure 7.36: Order/Deposit Cashier's Cheque Pag

Figure 7.37: Opt Page for Opening a New Account

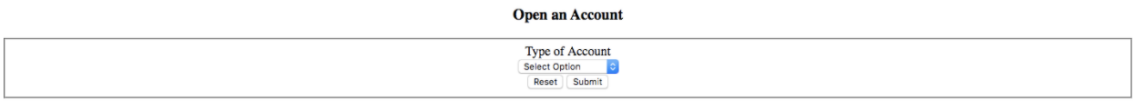

Figure 7.38: Open New Account Page

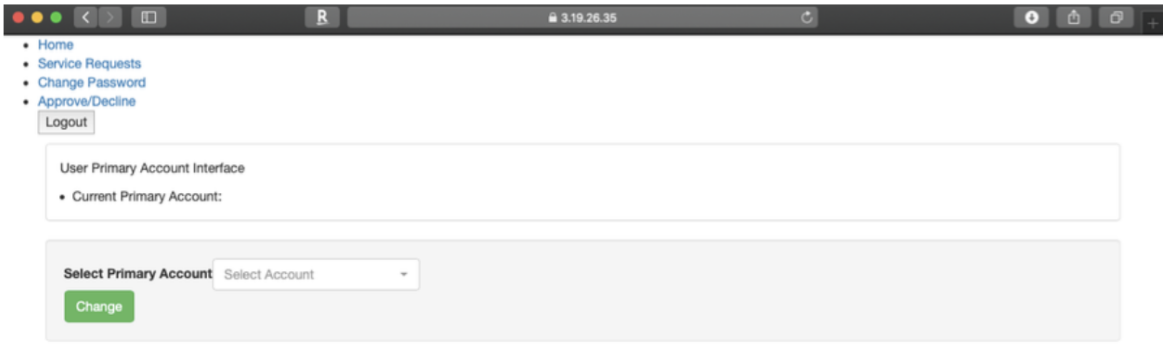

Figure 7.39: Set Primary Account Page

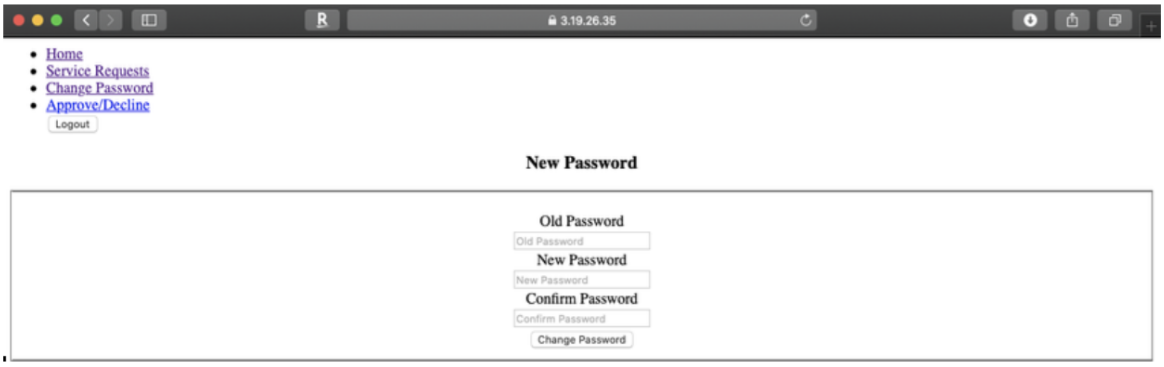

Figure 7.40: Change Password Page

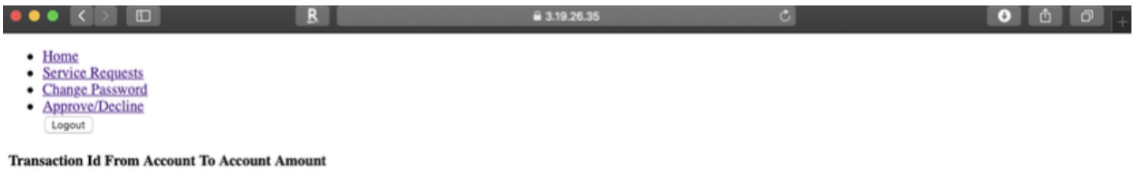

Figure 7.41: Approve/Decline Page

Sample Customers

Username	Password
becker.westley	1234
murazik.dominique	1234
francisca.robel	1234
weissnat.evan	1234
lera.runolfsdottir	1234

Figure 7.42

Sample Tier 1 Employees

Username	Password
grimes.sister	1234
herzog.sophia	1234
katheryn.runte	1234
steuber.hassie	1234
donnell.dietrich	1234

Figure 7.43

Sample Tier 2 Employees

Username	Password
mreynolds	1234
esteban10	1234
nhalvorson	1234
karelle.fritsch	1234
cole.meaghan	1234

Figure 7.44

Sample Admin

Username	Password
schuster.jalyn	1234
tillman.matilde	1234
kailey.gibson	1234
davis.lorenz	1234
uheathcote	1234

Figure 7.45

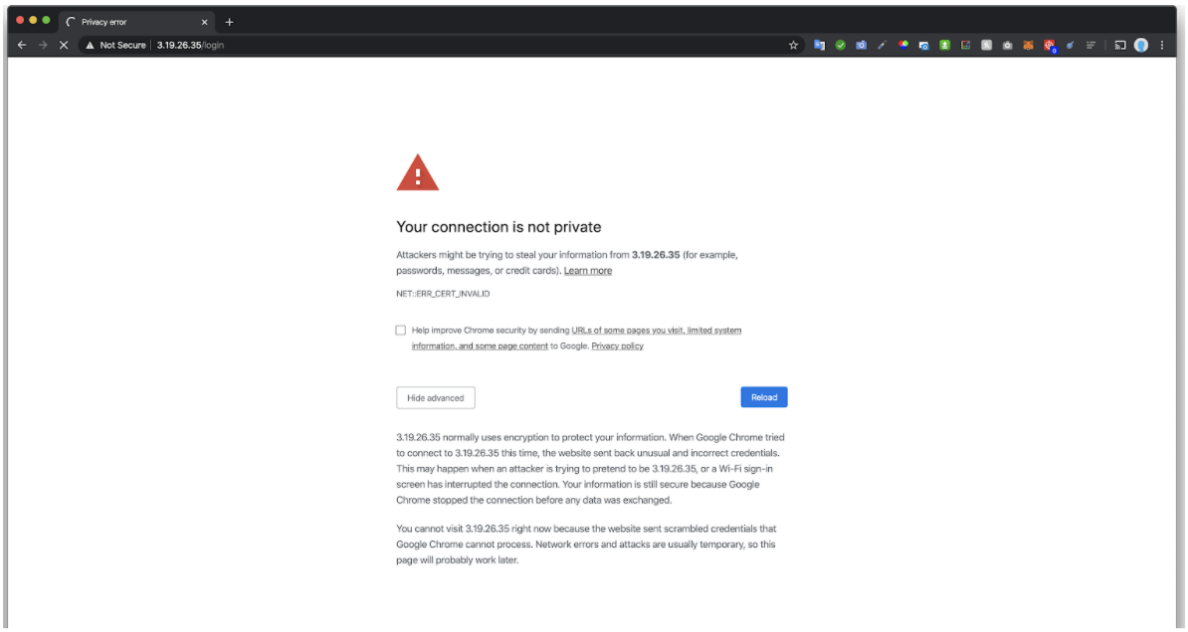

Figure 7.46

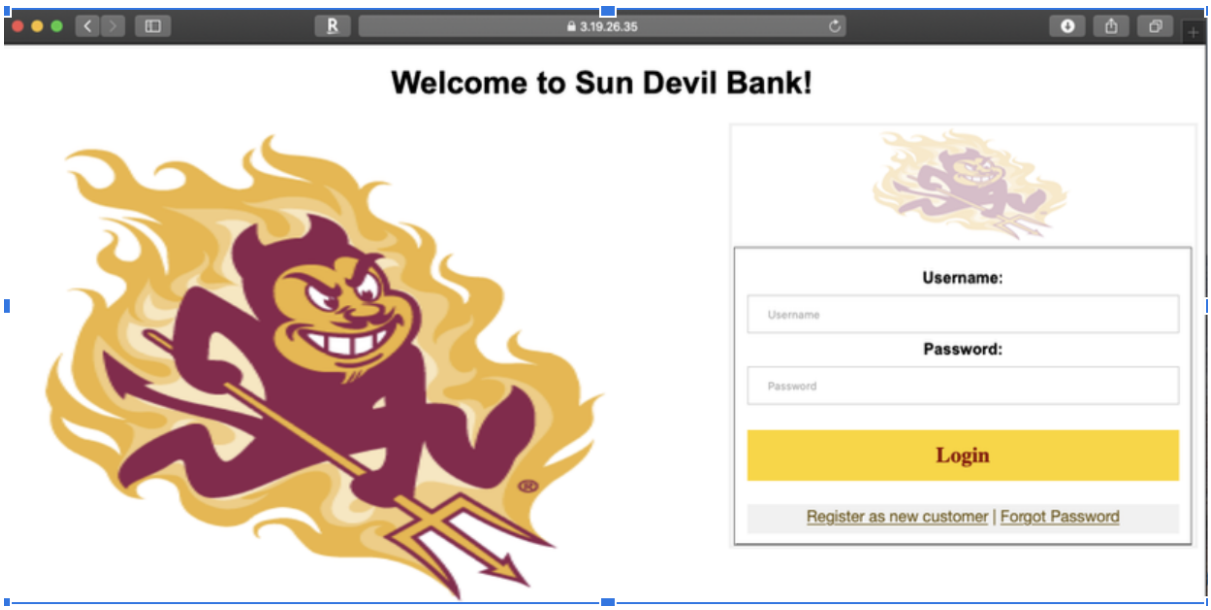
3.19.26.35
Welcome to Sun Devil Bank!
Username:
Username
Password:
Password
Login
Register as new customer | Forgot Password

Figure 7.47

Chapter 8

Conclusion

A 24 hour secure banking system was developed to give the users a safe and a secure way to transact money coupled with a smooth user experience.Strict security measures were followed including,

- One Time Password (OTP)
- Secure Sockets Layer (SSL)
- Public Key Infrastructure
- Malicious Login Controls
- CSRF Tokens
- Form Validations
- Hyperlegder Integration
- URL protection
- Hashed password storage

Although the above mentioned security measures were implemented in the system, users of the system were able to find some security and functional vulnerabilities in the application. Those vulnerabilities and bugs have been addressed and appropriate solutions have been discussed in this report.

Chapter 9

References

[1] https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

[2] https://spring.io/projects/spring-security

[3] https://www.nginx.com/resources/wiki/

[4] https://en.ryte.com/wiki/Post-Redirect-Get

[5] https://owasp.org/www-project-top-ten/

[6] https://hyperledger-fabric.readthedocs.io/en/release-2.0/