ICIN Bank Portal

Software Requirements Specification

Payam Dowlatyari

Post Graduate Program in Full Stack Web Development - Capstone Project

Caltech CTME - Simplilearn

Sep 2021 - May 2022

Table of Contents

Introduction

Purpose

The main purpose of this document is to gather information and perform requirements engineering and analysis to define the details of a full stack development of a web portal and application named **ICIN Bank** for a company with the same name.

Intended Audience

- **Developers** include UX/UI designers, graphic designers, product designers, front-end engineers, backend engineers, database designers and developers, DevOps developers, test engineers, QA specialists, etc.
- **Scrum Team** includes product owner, scrum master, and development team.
- **The Marketing and Sales Team** includes sales specialists, advertising experts, business analysts, etc.
- **Employees** including technical and non-technical personnel of the ICIN Bank in all levels and branches
- **The Management Team** includes the CEO, lawyers, advisers, other board members, and other shareholders.
- **End-Users** include portal administrators and customers

Product Scope

ICIN is one of the top banking firms that accept deposits from the public for lending loans to the public. It also invests an amount in securities. Recently, the business analysts noticed a drop in the number of customers of the bank. They found out that online banking systems of banks like AXIS and American Express are gaining more profits by eliminating middlemen from the equation.

As a result, the team decided to hire a Full Stack developer who can develop an online banking web application with a rich and user-friendly interface. Using this application, customers can go on the portal, signup, or log in if they have already registered to manage their funds. They can make and update their profile, initiate money transfers to other accounts, display the transactions and their status, and request new checkbooks.

Likewise, bank employees can log in to Admin Portal and provide different services to customers. Admins can add, update, and remove customers and their account information from the database using the user interface and their admin credentials.

Iteration Process and Guidelines

- The prototype of the application shall be developed and be available to the management team and other stakeholders in four sprints with every sprint delivering a minimal viable product.
- It is mandatory to perform proper sprint planning with user stories to develop all the components of the project
- All tools and technologies shall be introduced including programming languages, libraries, frameworks, database management systems, DevOps tools, automation technologies, and cloud servers for application hosting and deployment, etc.
- The stakeholders shall approve the design of the database, security protocols, and user interface before developing a high-fidelity prototype of the application.
- The main repository should be created and become accessible to stakeholders.

Sprint Planning

1. **Kick-off meeting week zero:** High-level design of the database using Entity-Relationship diagram and object-oriented structure of the server-side application and their associations. Sketch the user interface according to available services and backend features and options.
2. **Sprint week one:** Implement the database in MySql and the structure of backend data models, entities, repositories, services, and controllers using the Spring framework. Create REST API functions and conduct testing with Postman to ensure the quality and security of HTTP requests.
3. **Sprint week two**: Develop the frontend application according to the prior design and the backend models. Build services to send HTTP requests to the server and receive the proper responses. Develop the structure of user interface components including, navbars, forms, tables, dashboards, etc.
4. **Sprint week three:** After successful testing and ensuring that the basic services work smoothly, the team can go forward and expand the features and services and conduct unit testing and debug each service along with the completion and enrichment of backend and frontend code.
5. **Sprint week four**: Specify the virtual machine configurations and start the process of deployment of the application on AWS EC2. Conduct system testing using automated services like Jenkins to ensure the consistency of services and usability testing while the application is up and running

Requirements

User Personas

Brianna

66 years old, a retired teacher

Brianna is a teacher who retired recently after 30 years of teaching history. A few years ago, her husband passed away and she had to rent a small apartment. She receives her income as a direct deposit and has to send a monthly check for rent to her landlord. Brianna used to drive to the bank every time she wanted to get money or request a new checkbook. However, she can manage everything online using ICIN User Portal.

Goals and Motivations:

- Monitor account balance, deposits, and charges
- Request checkbooks in advance
- Transfer money safely and easily

Daniel

35 years old, ICIN Bank employee

Daniel is an employee with 8 years of experience. He has married 2 years ago, has a one-year-old son, and waiting for the second baby to be born. He and his wife decided to move to another city to buy a house and raised their children. He wanted to quit his job but his manager told him about their new Admin Portal and offered him a remote position. Daniel is very happy because he can do his job online and spend more time with his family.

Goals and Motivations:

- Manage clients' requests with Admin Portal
- Approve transactions and issue checkbooks
- Work remotely and spend more time with his family

User Stories

1. As a bank customer, I want to be able to create an account, so that I can login to the user portal and check my account balance regularly.
2. As a bak customer, I want to be able to login into my user portal, so that I can transfer money to a different account safely.
3. As a back customer, I want to be able to request a new checkbook through the user portal, so that I can avoid going to the bank and waiting in line.
4. As a bank portal user, I want to be able to change my password anytime, so that I can ensure the safety of my funds.
5. As a bank portal user, I want to be able to navigate easily through the menu so that I can view all the possible options.
6. As a bank employee, I want to be able to login into the admin portal, so I can provide services to our clients.
7. As a bank portal admin, I want to be able to accept or reject customer requests, so I can easily follow the rules.
8. As a bank portal admin, I want to be able to create, update, and delete user profiles, so I can protect the clients' funds.
9. As an admin portal user, I want to be able to lock accounts or pause transactions, so that I can eliminate criminal activities.
10. As a portal user, I want to use the system easily, so that I can complete my transaction as fast as possible.

User Acceptance Criteria

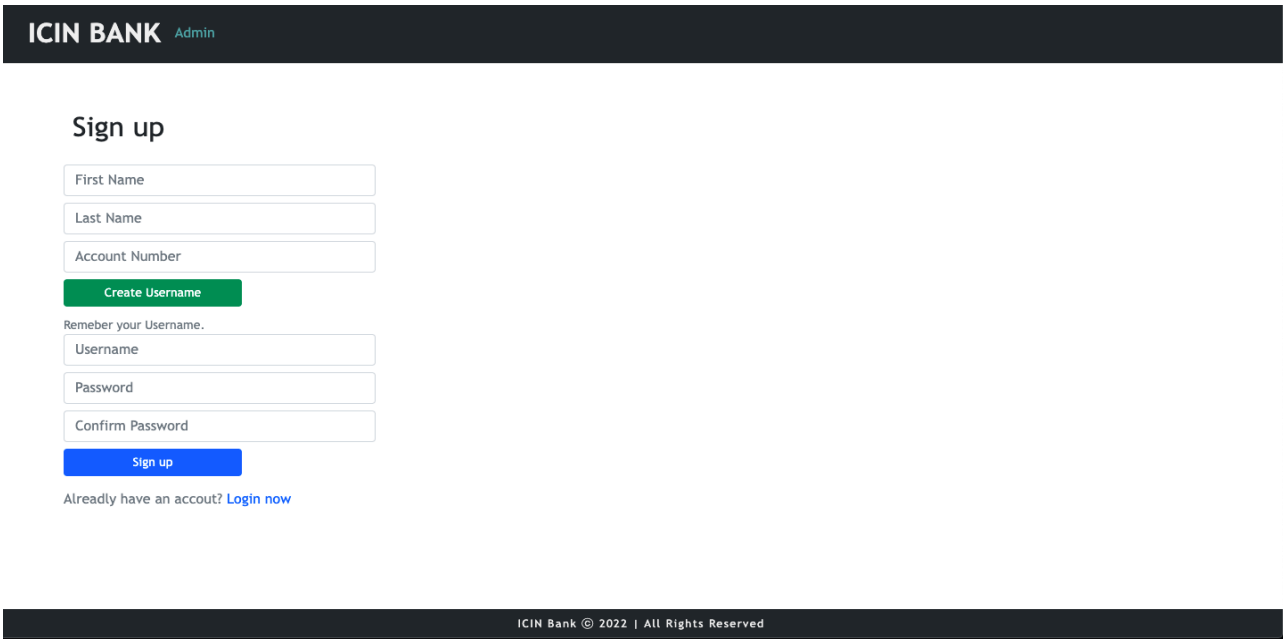
- Given the username and password, the system should login users into the user dashboard.
- Given the customer name, account number, and other required fields, the system shall allow the customers to create user profiles and immediately login and use the services.
- Given non-standard or incomplete inputs, the system shall throw an exception without breaking the flow.

- Given the required data from the admin, the system must create a new row in the database for each user, account, transaction, checkbook request, etc. in their respected database tables.
- Given an account number and transfer amount from users, the system must initiate a new transaction and create a new row in the database.
- Given the transfer amount from users, in case of insufficient funds, the system must respond properly and prompt the user with a clear message.
- Given a new checkbook request from a customer, the system must show the request status such as pending, accepted, or rejected.
- In case of any rejection or error, the system shall provide a brief and clear explanation, and inform the user about the status of the service.

User Interface

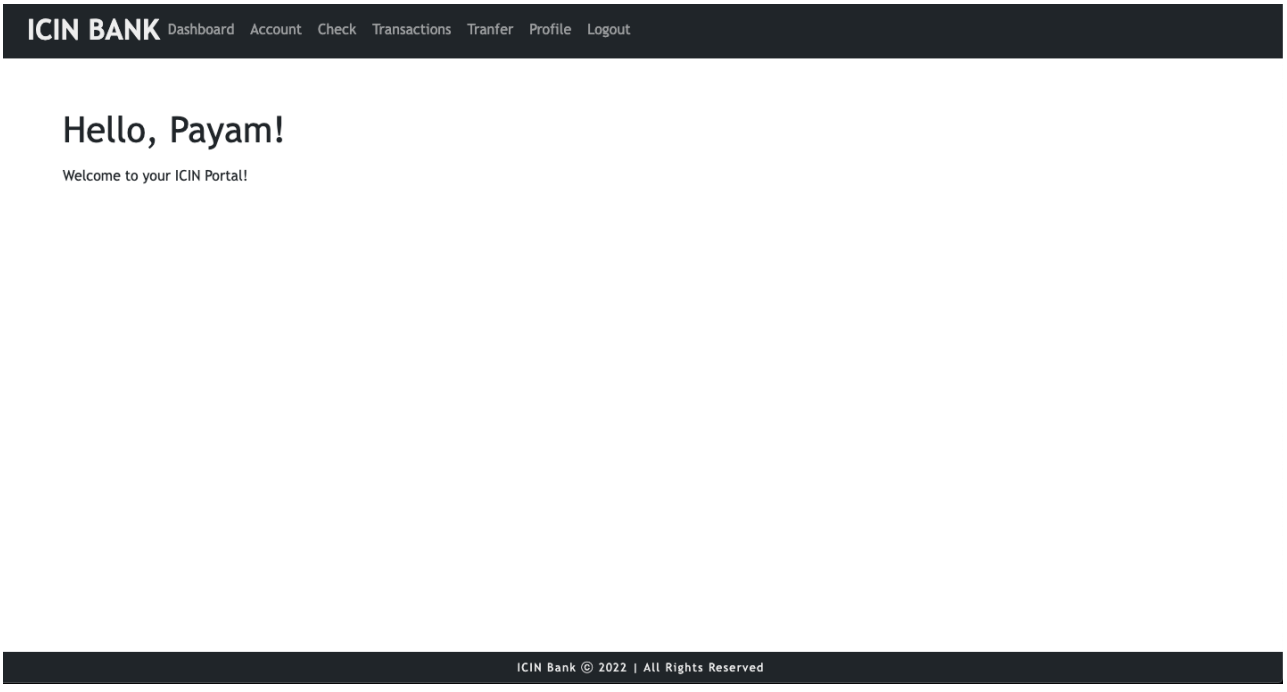
Home Page - User Login

Homepage - User Signup

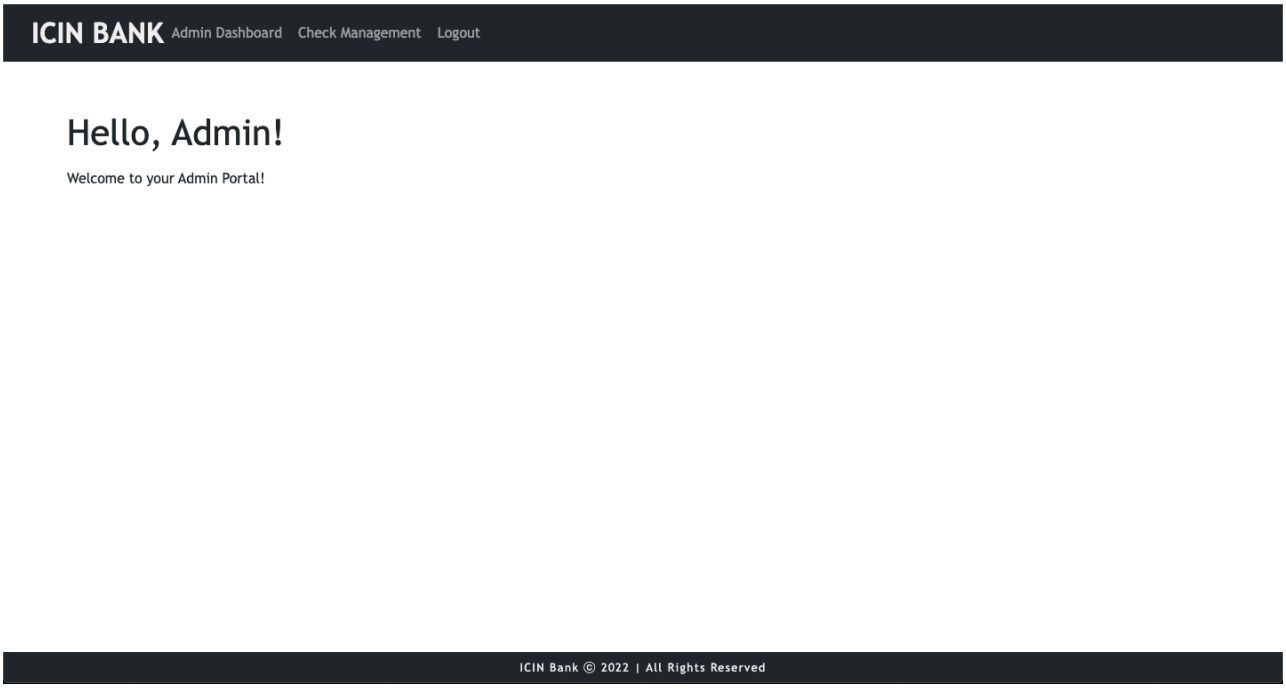


Admin Login

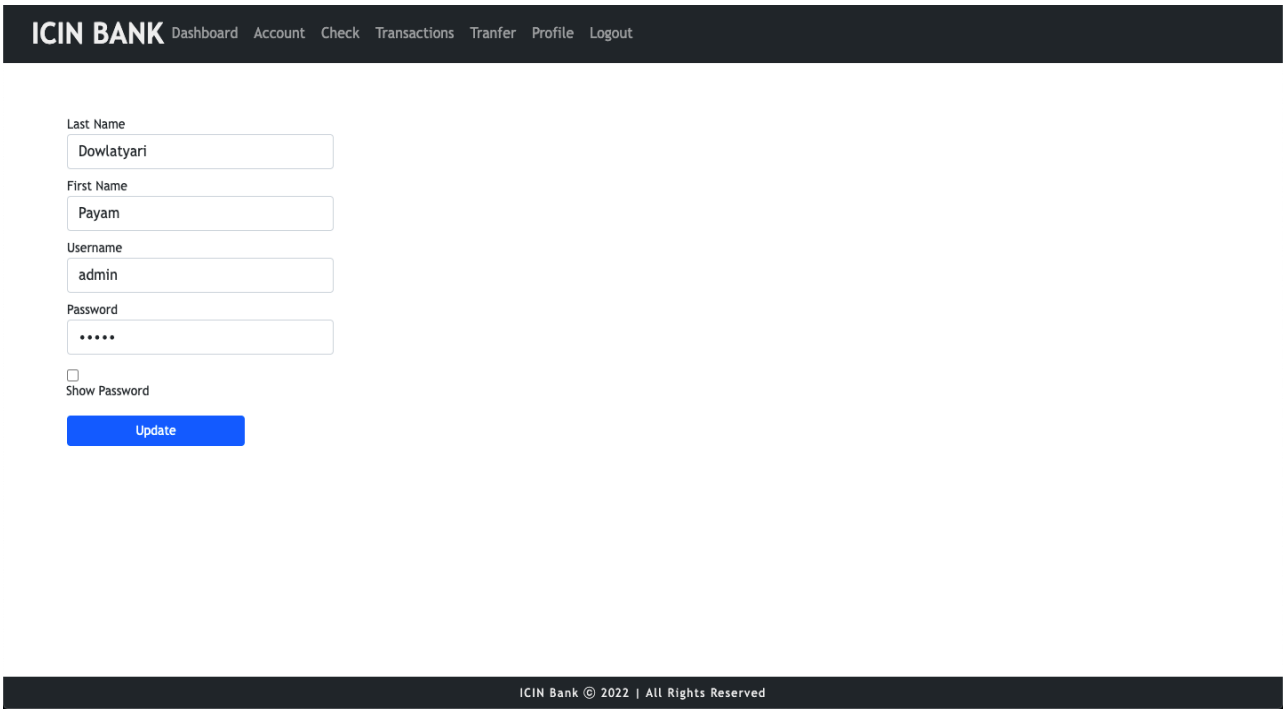
User Dashboard



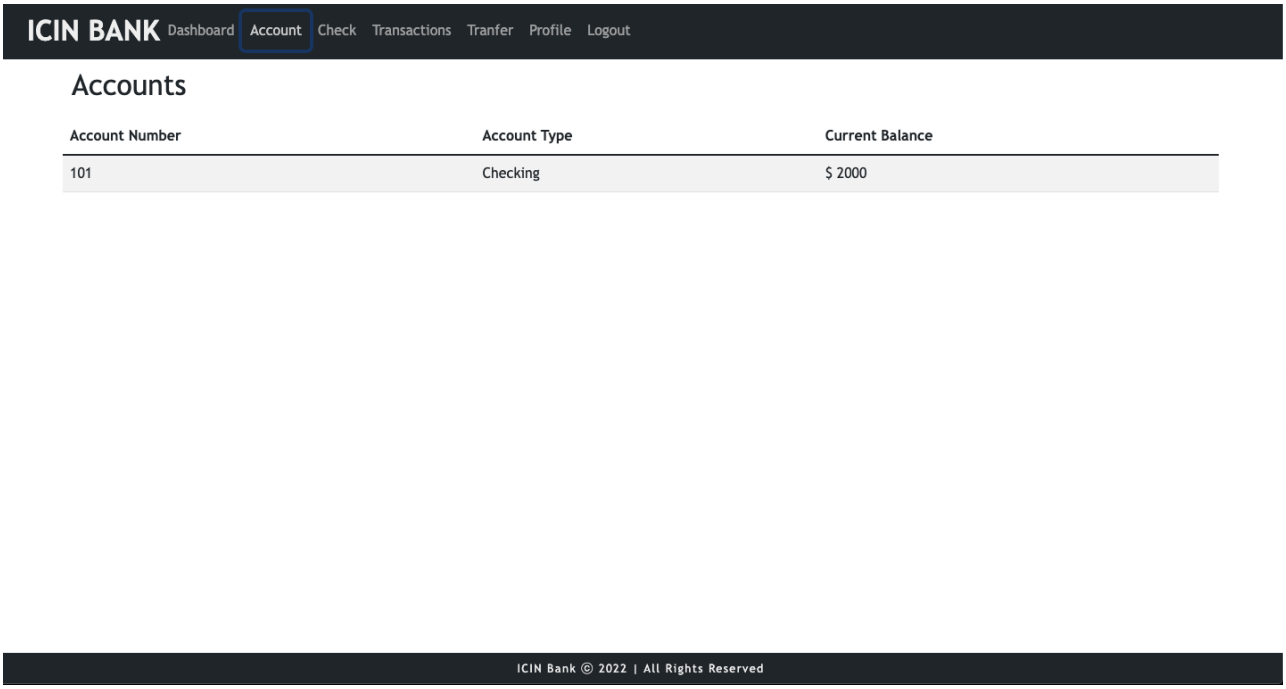
Admin Dashboard



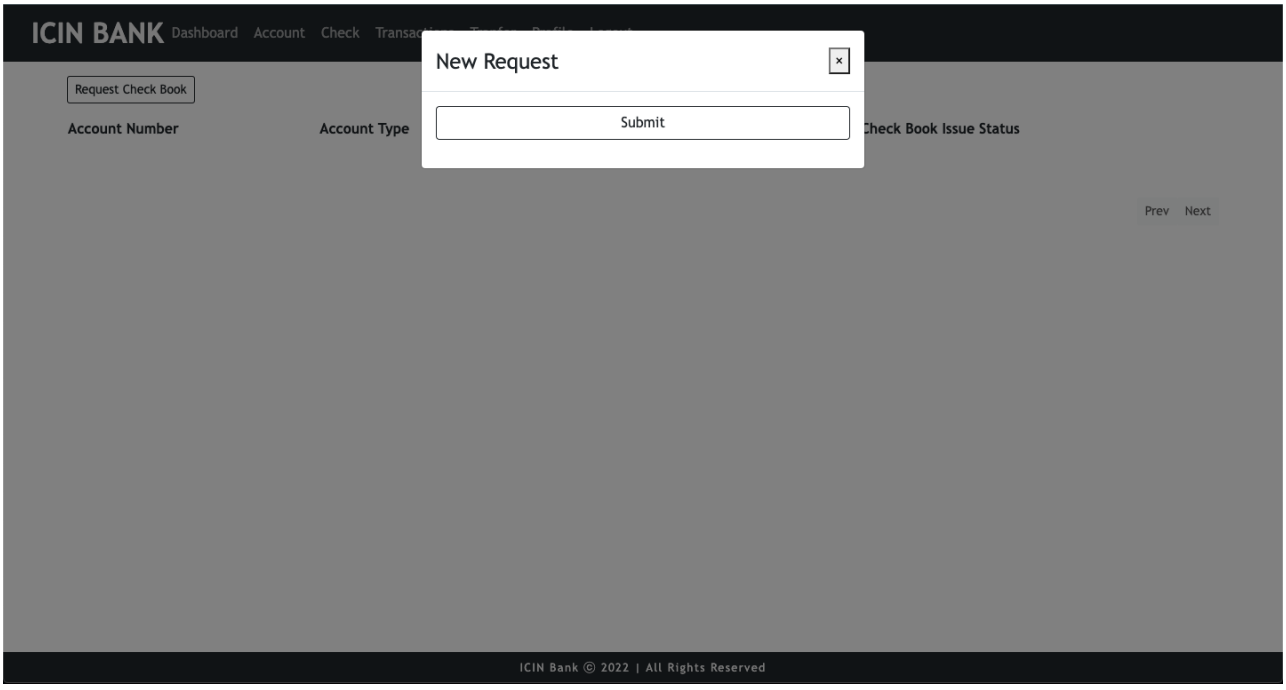
User Profile



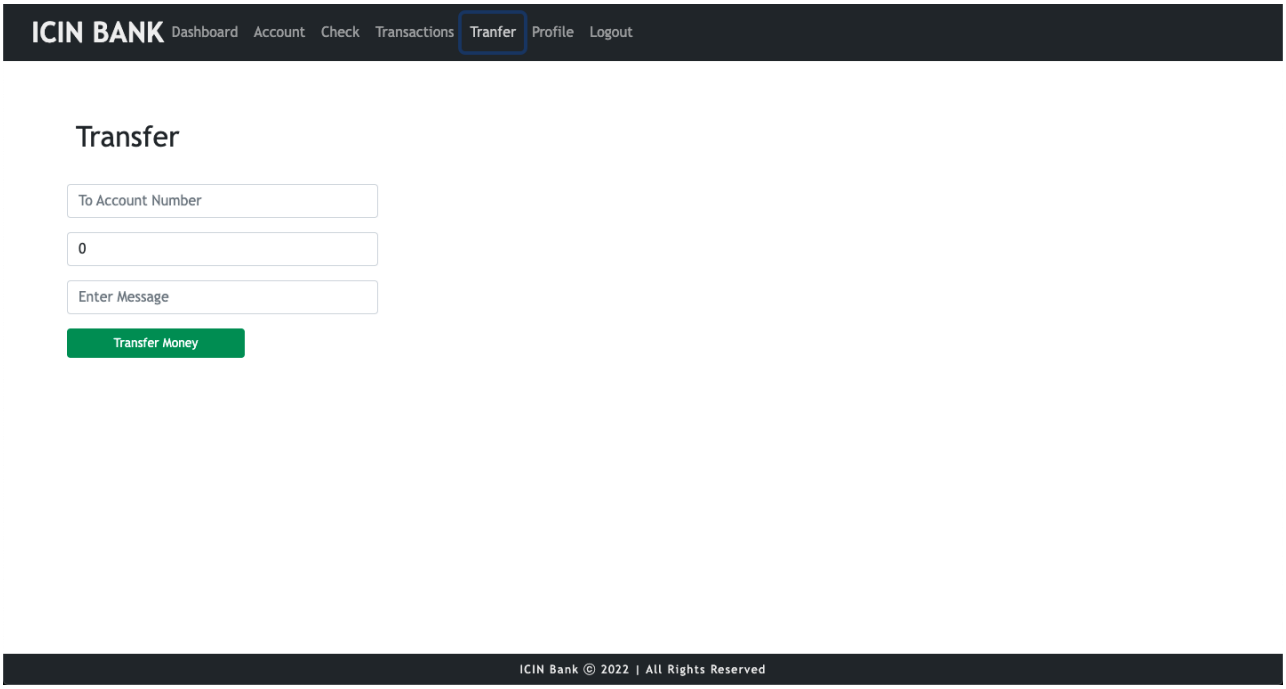
User Accounts Details



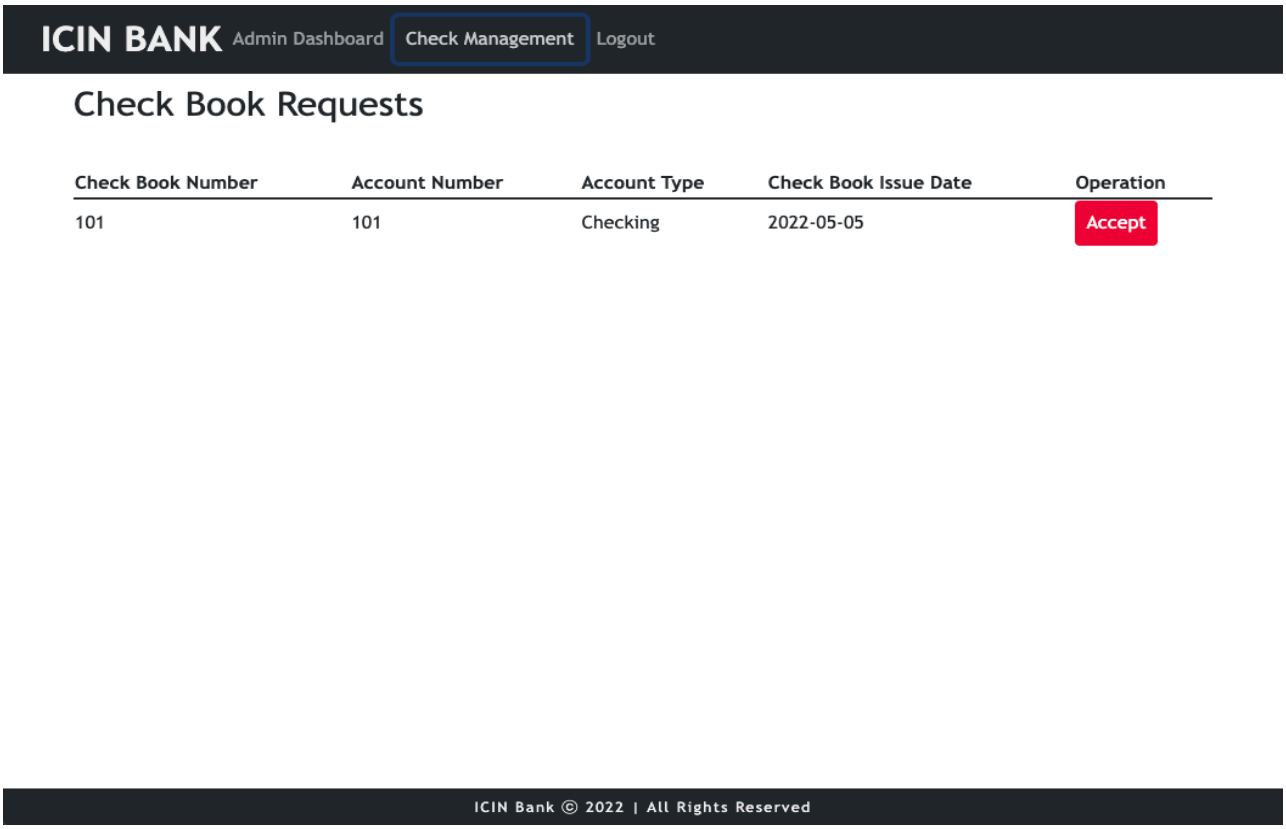
Check Book Request



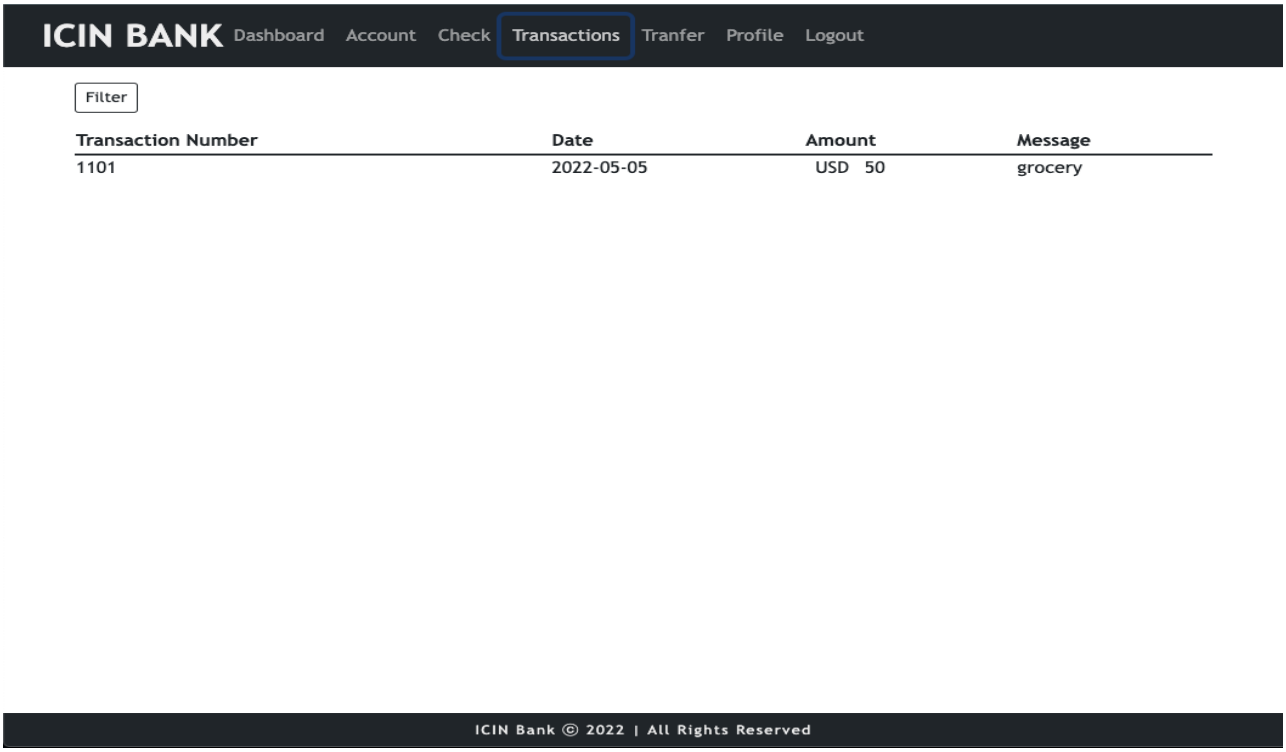
Transfer Page

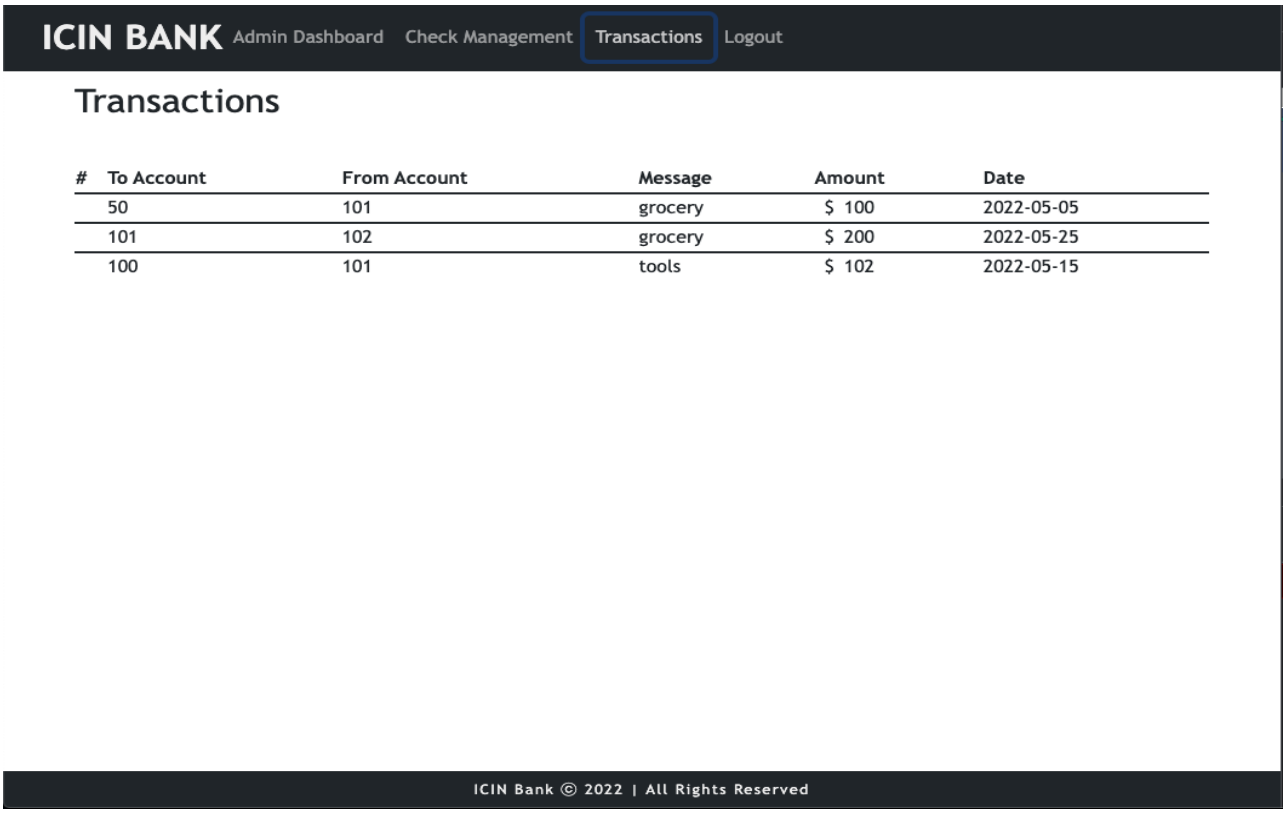


Check Books Admin Approval



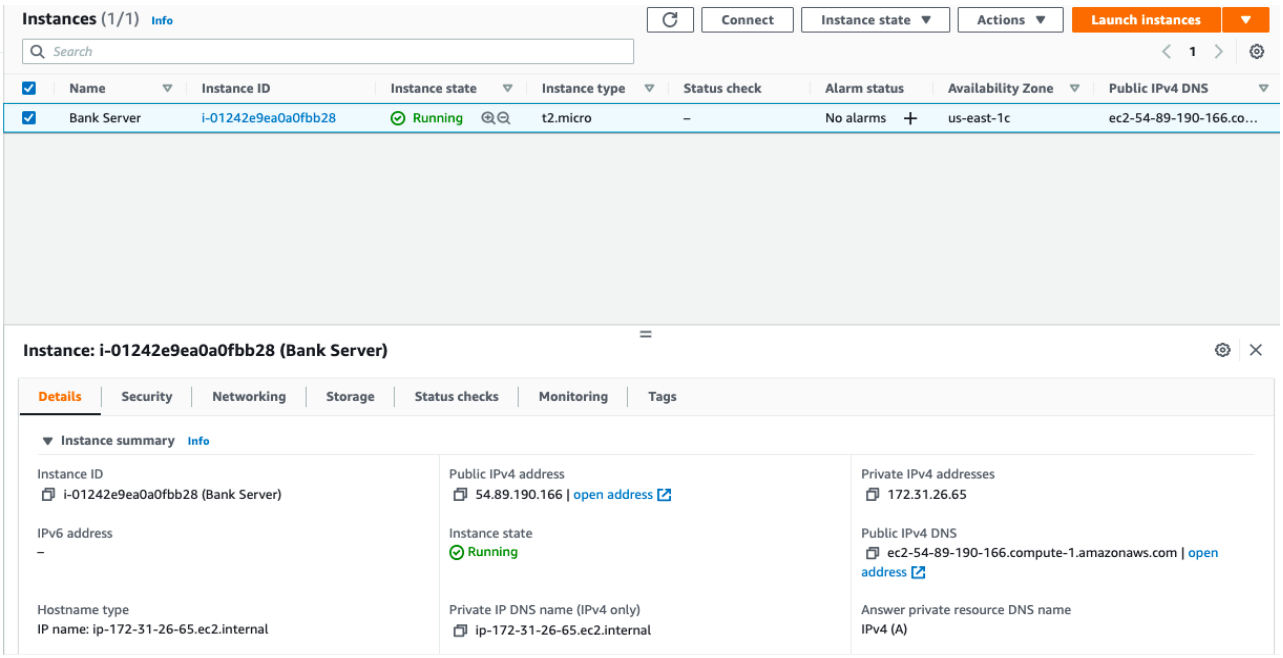
Account Transactions



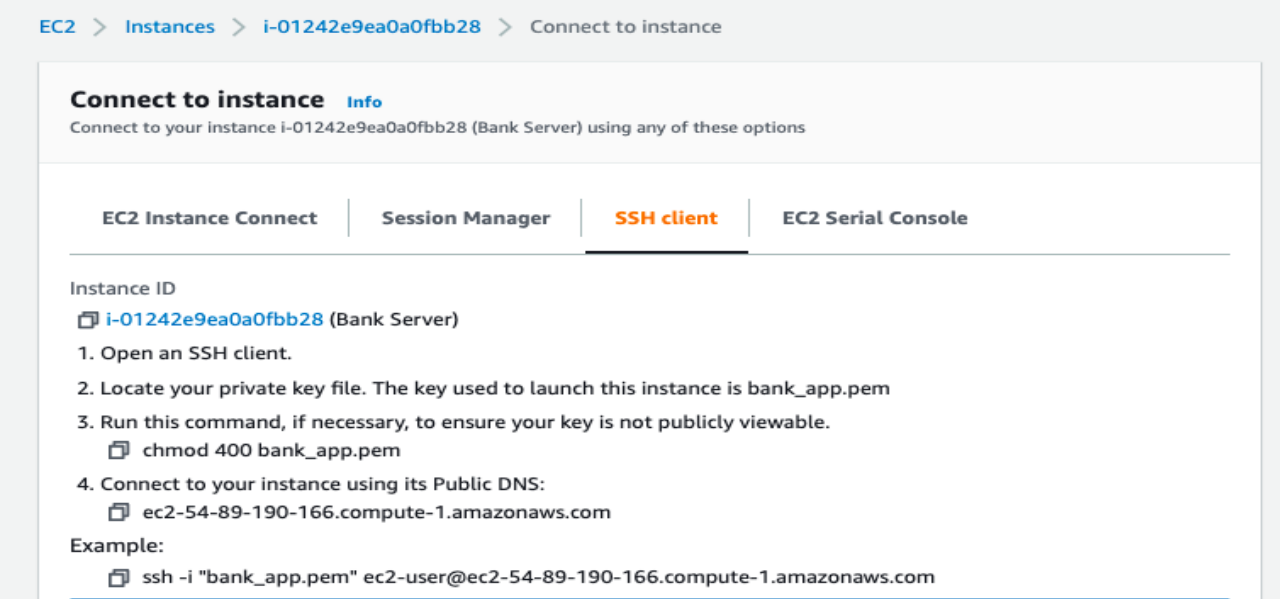


Deployment of Bank Server

Launch an AWS EC2 Instance



Connection Details



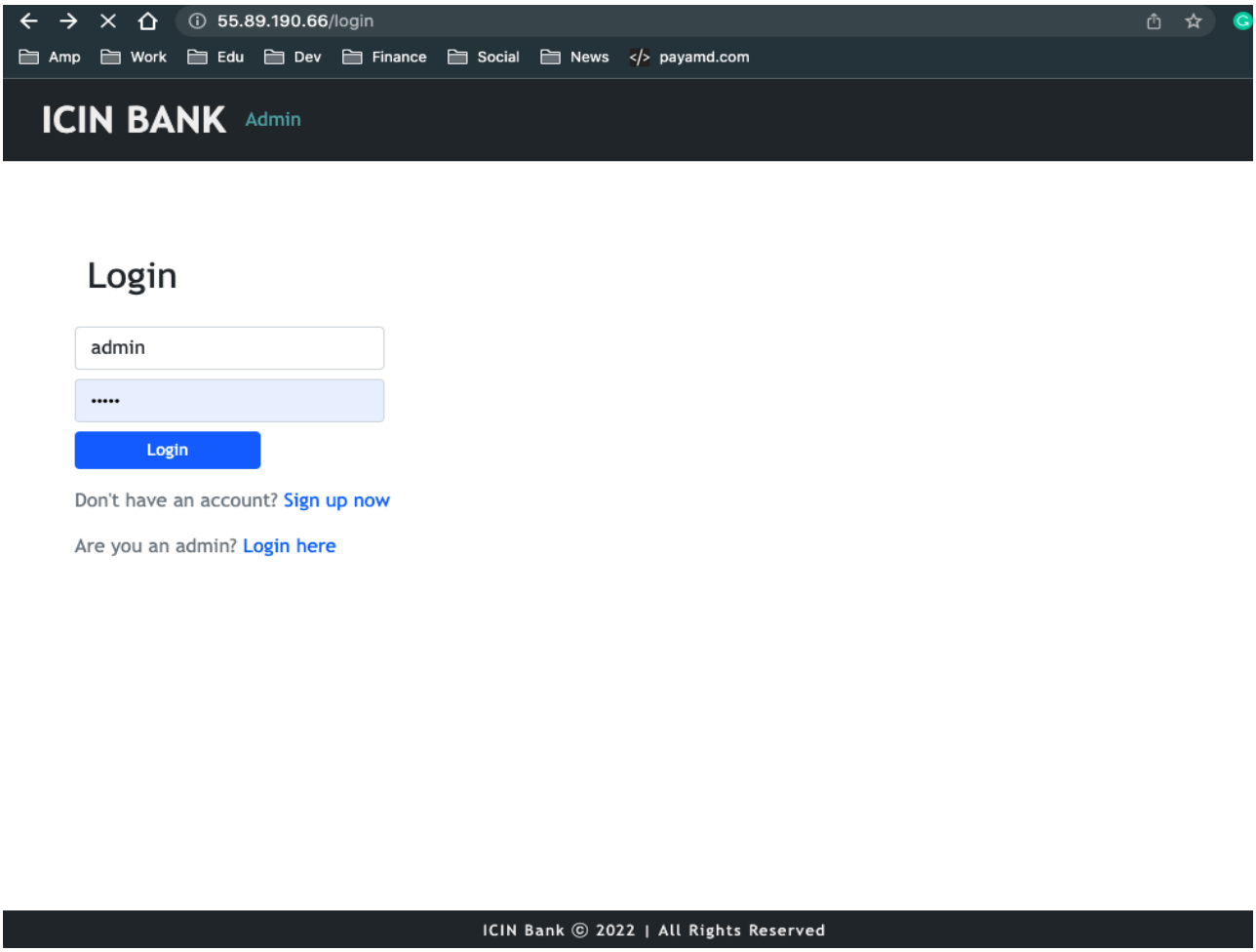
Connect to Bank Server from Local Machine

```
payamdowlatyari@Payams-MacBook-Air ~ % ssh -i "bank_app.pem" ec2-user@ec2-54-89-
190-166.compute-1.amazonaws.com
The authenticity of host 'ec2-54-89-190-166.compute-1.amazonaws.com (54.89.190.1
66)' can't be established.
ECDSA key fingerprint is SHA256:S5D6Xl194wzWaSZchjxfF+wN3eSjKrYLCI7IGeRz0ao.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-89-190-166.compute-1.amazonaws.com,54.89.190.
166' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 6 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-65 ~]$
```

Clone Git Repository and Install Dependencies

```
[root@ip-172-31-26-65 ~]# git clone https://github.com/payamdowlatyari/PGFSD-Cap
stone.git
Cloning into 'PGFSD-Capstone'...
remote: Enumerating objects: 663, done.
remote: Counting objects: 100% (663/663), done.
remote: Compressing objects: 100% (342/342), done.
remote: Total 663 (delta 388), reused 548 (delta 275), pack-reused 0
Receiving objects: 100% (663/663), 402.64 KiB | 17.51 MiB/s, done.
Resolving deltas: 100% (388/388), done.
[root@ip-172-31-26-65 ~]# ls
PGFSD-Capstone
[root@ip-172-31-26-65 ~]#
```

Application Running on AWS



Other Requirements

Development Details and Guidelines

- The application must be implemented with Java and Spring framework mapping to database tables.
- MySQL shall be designed and implemented for the database connected to the server-side application.
- Angular framework shall be used to develop the frontend.
- The web application should be responsive and should fetch or send data dynamically without hardcoded values.
- The developers must maintain the version of the application over GitHub and every new change should be sent to the repository.
- The developers must implement a CI/CD pipeline using Jenkins.
- The developers should also deploy and host the application on an AWS EC2 instance.
- The developers should also implement automation testing before the application enters the CI/CD pipeline.
- The developers should use Git branching to separately perform the basic automation testing of the application.
- The developers should make a rich front-end of the application, which is user-friendly and easy for the user to navigate through the application.
- There will be two portals in the application, namely the admin and user portal.

  Admin Portal:

  It deals with all the back-end data generation and product information. The admin user should be able to:

  - Authorize the roles and guidelines for the user
  - Grant access to the user regarding money transfers, deposits, and withdrawal
  - Block the user account in case of any threat
  - Authorize the checkbook requests

User Portal:

It deals with user activities. The user should be able to:

- Register or log in to the application to maintain a record of activities
- Deposit and withdraw money from the account
- View transactions and balance in the primary and savings account
- Transfer funds between different accounts and add recipients
- Request checkbooks for different accounts

Non-Functional Requirements

- The services must be designed and built as highly secure and reliable.
- The system must be platform-independent and works on different operating systems such as macOS, Windows, and Linux.
- The application should work fast and run smoothly.
- The user interface should be user-friendly, informative, and easy to navigate.

Constraints and Assumptions

- The application can only support the English language.
- Users must have access to a keyboard for providing their inputs.
- Assuming the data is saved and backed up repeatedly.
- The creation of admin credentials cannot be done through GUI due to security concerns.

Future Iterations

The next sprints would focus on improving the UX design and expansion of services and features based on users' feedback.

- Improving the user experience design and usability of the platform
- Adding new feasible services based on the customers' needs
- Periodically testing, bug fixing, and maintenance
- Conducting user surveys and other techniques to get feedback