VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

1. Contributors

1.1 Work contribution by Member 1:Siddharth Sharma(19BCE2689)

Worked on the backend using NodeJs and MongoDB along with Member 3

1.2 Work contribution by Member 2-Tuhin Chakrabarty(19BCE2691)

Worked on the design and frontend modules along with Member 4

1.3 Work contribution by Member 3-Nevin Mathews Kuruvilla(19BCE2507)

Worked on the backend stack along with Member 1

1.4 Work contribution by Member 4-Zihan Azad(19BCE2442)

Worked on the frontend stack along with Member 2

2. Title Justification

2.1 Problem Statement

Although the basic type of services offered by a bank depends upon the type of bank and the country, services provided usually include: Taking deposits from their customers and issuing current or checking accounts and savings accounts to individuals and business. Extending loans to individuals and business, Cashing cheque. Facilitating money transactions such as wire transfer and cashiers cheque, Consumer & commercial financial advisory services, financial transaction can be performed through many different channels.

At a time where going out to do these essential tasks hinders our probability of survival, we need to develop a system that helps people to accomplish such essential tasks in a hassle free way that requires the least knowledge of online systems.

2.2 Project Motivation

To develop a system that will overlook the activities going transaction the particular bank without manual processing. All transaction will be updated automatically by using the information stored in record. The main motive behind this project is to develop a system which will able to handle the overall tasks going inside the institutions without much effort.
The simple user interface available in our website will make it easy for customers to handle their day-to-day banking needs without being perplexed or overwhelmed.

2.3 Glossary of terms

2.3.1 Technical Terms

Routes, Database, APIs, concurrency, AES, SHA-256, modules, verification, local host, HTTPS, PassportJS, Input field, nth-child, UI, UX, prototype

2.3.2 Non-Technical Terms

Beneficiaries, Loan, Account Transfer, Dashboard, Stakeholders, Investors, Admin, principal amount, interest

2.4 Functional Requirement Specification

2.4.1 Stakeholders

The stakeholders for this online banking system would be:

- Investors: People who are willing to invest capital for the system.
- Bank employees: Personnel's who are involved in the functioning of a bank
- Customer: People who wish to do their banking services with us and entrusting us with the safety of their account information.

2.4.2 Actors and Goals

- User: Customers who wish to accomplish their banking activities. Their goal is to interact with the system and fulfill their needs at their own convenience.
- Admin: Bank employees who have access to customer information. Their goal is to help the user fulfill their wants and needs.

- Developer: Personnel who is in charge of maintaining and developing the website. Their goal is to make sure no technical errors occur and customers have a smooth experience with our online bank.

2.4.3 Intimating Actors

- User: Customers of the bank who wish to use the system to conduct their day to day banking activities
- Employees: Staff of the bank who is there to help the customers conduct themselves

2.4.4 Participating Actors

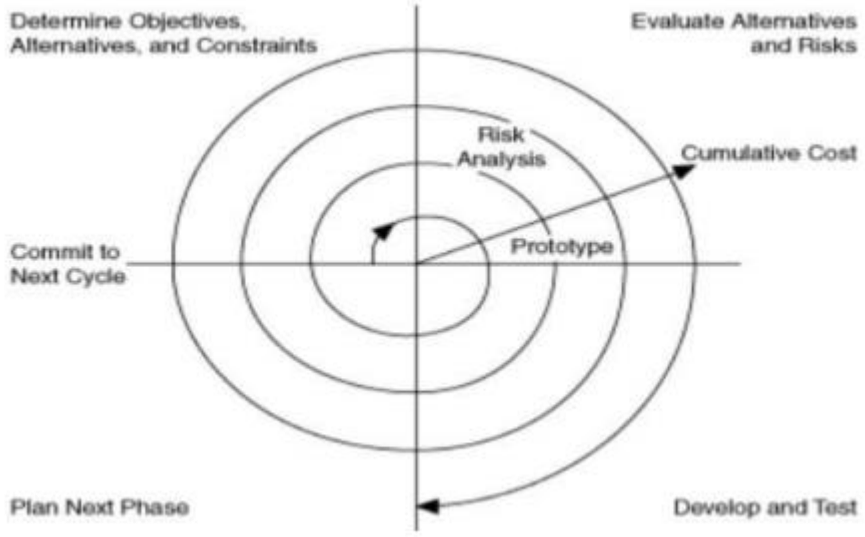
- User
- Admin
- Developer

3. Title Justification

3.1 Software Model

3.1.1 Model used in project

The Software Development Life Cycle (SDLC) model that was used for this project is the Spiral model. In this model each phase has well defined starting and ending points, with clearly identifiable deliverables to the next phase



3.1.2 Model Justification

The spiral model enables gradual releases and refinement of a product through each phase of the spiral as well as the ability to build prototypes at each phase. The most important feature of the model is its ability to manage unknown risks after the project has commenced; creating a prototype makes this feasible.
We chose this model because enabled continuous development. We were able to develop gradual prototypes and keep adding features without worrying about the risks.

3.2 Requirements

3.2.1 Functional Requirements

3.2.1.1. Login/Register new user

The customer or the employee can access their accounts by entering the correct username and password. If the user has not created an account yet, there is an option to register as a new user and a form will be generated which will require all the necessary details of the user.

LFR-001: System should display login page.
LFR-002: System should verify credentials of user by matching it with the data present in the database.
LFR-003: System should redirect user to homepage
LFR-004: System should display user registration form if user does not have an account
LFR-005: System should save the user information to the database.
LFR-006: User should be redirected to login page and be verified.

3.2.1.2. Account Details

The customer can check his/her account details and it automatically gets updated every time the user makes a change. The user can also check his/her balance which is a very vital part in any online banking experience.

AR001 – System should display account details icon.
AR002 – System should display account details page when user prompts.
AR003 – Any change that the user makes must be reflected on the account details page

3.2.1.3. Fund Transfer

The user can transfer funds from one account to another account. Both the accounts must be registered in the same bank and should be valid users.

FTR001 – System should display fund transfer icon.
FTR002 – System should display list of accounts that the current user is registered in and the user chooses one for the transfer.
FTR003 – User should be able to enter the amount for transfer and the beneficiary account number and name.
FTR004 – System should send a verification mail of the action that has taken place to the user's mail ID.
FTR005 – System should make the effective changes in the account balance of the user.

3.2.1.4. History of transactions

The user should be able to view the history of transactions and action statements made by him/her.

HR001 – System should display the history icon to the user.
HR002 – Once clicked, the system should display the list of transactions that took place along with the time it took place.
HR003 – The system must enable the user to filter out a desired time frame at which a particular transaction took place

3.2.1.2. Loan

The user should be able to apply for a loan after giving the required information.

L001 – System should display loan icon to the user
L002 – The system should then display a form that requires information for the user to enter such as loan type and loan purpose
L003 – The system should not let the user enter a value too small or a value too large as the principle amount
L004 – After application of loan is submitted, the system should display a success message
L005 – The user should then be able to see the status of his loan in the accounts page

3.2.2. Non- Functional Requirements

Performance Requirements

The system must support multiple users at the same time i.e it should have support for concurrency

Safety Requirements

The system hardware and software should not be compromised under heavy load or the instance of a cyber-attack.
It should be generating backups periodically and maintaining redundant copies.

Security Requirements

The system should not be compromised by any external entities
The system has to be designed with modularity as a high priority
Encryption and Hashing must be used as countermeasures to prevent attacks from hackers

Software Quality Attributes

The site has to be easy to use and navigate. Graphical representation of the feature must be provided along with its description (Use icons and glyphs). The quality of the system should be such that human attributes are taken into consideration while designing it. The design should be minimalistic, easy-to-use, intuitive and simple to use.

Business Rules

A decision-making hierarchy for invoice processing, where the values of certain invoices are tiered to determine which managers (on the other side) can approve.

3.2.3 User Stories

As a customer, I want a hassle free and intuitive mode of banking system that helps me fulfill my banking requirements without having to actually go to the bank.

As an employee, I want to manage user requests and customer requirements with the help of a system that makes it easier for me to manage.

As an investor, I want to continue the functionality of our bank even though people may not be able to visit personally.

3.2.4. Customer Statement of Requirements

The portal should provide the user not only the basic banking features like transferring money but also show transaction history, minimum balance to be maintained, status of PPF, etc. and if you feel like you see unusual account activity you can block the account within clicks.

- The site should allow the user to register
- This site should allow the user to login, if already registered
- The site must grant privileges base on the user’s authorization
- The user must be able to transfer money
- The user must be able to view his transactions on his account
- The user must be able to block his account
- The user must be able to add beneficiaries
- The user must get SMS notification for each transaction
- The user must see the status of his check

3.2.5. Network Protocols/ Hardware Requirements

Network Protocols

HTTPS (HyperText Transfer Protocol Secure) - A protocol used by the application layer for transmitting hyper-media documents such as HTML.

SMTP (Simple Mail Transfer Protocol) - SMTP is used by the node mailer to send an OTP (One Time Password) for a customer who wants to use the **transaction services**. It is used to verify the authenticity of the user.

Hardware Requirements

CPU: Pentium – 2GHz
Hard Disk: 15GB
RAM: 4GB
OS: Windows, MacOS,Linux
Browser: Google Chrome 87 and above, Mozilla Firefox 9.0.1 and above, Microsoft Edge 80 and above.

3.2.6 External Interface Requirements

User Interfaces

Bootstrap
HTML/CSS

Hardware Interfaces

RAM – 12 GB
Processor – Intel i7 4.9 GHz
Hard Disk – 512 GB

Software Interfaces

OS – Windows 10
Editor – VSCode
Version Control – Github
Browser- Google Chrome v89

Communications Interfaces

Transfer Protocol - HTTPS
Authentication and authorization – PassportJS

3.2.7 Database

Database Management System

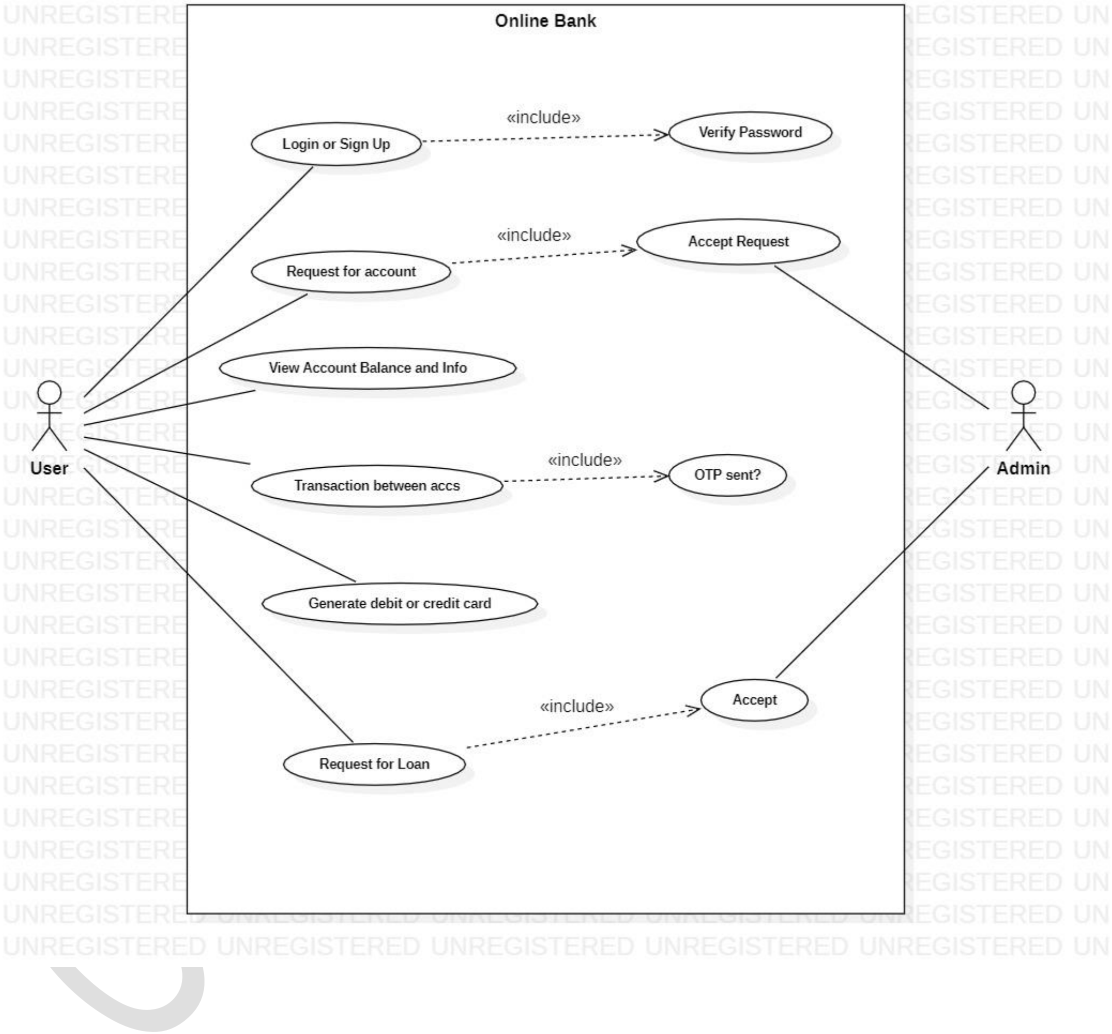
Stores all details of all users. Root access granted only to admins. “Backup” instances of primary database in case of storage failure, fraudulent activities, or malicious attacks

3.2.8 Objective

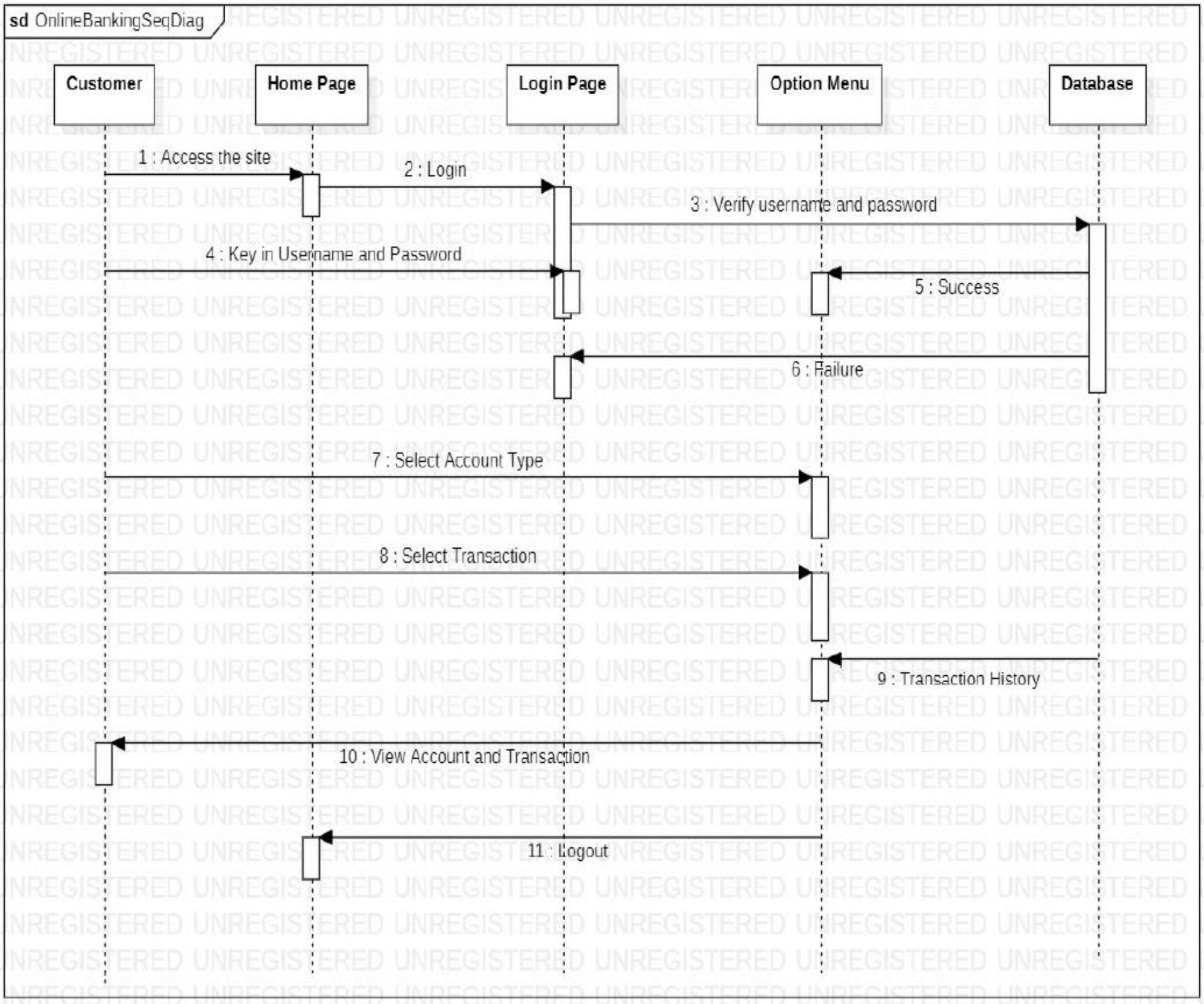
The Online Banking System is a website that allows users to have a simple and hassle free experience with their online banking transactions. The site provides the user a simple interface compared to other online banking websites. It provides basic banking features like transferring money, showing transaction history and many more.

4. UML Diagram

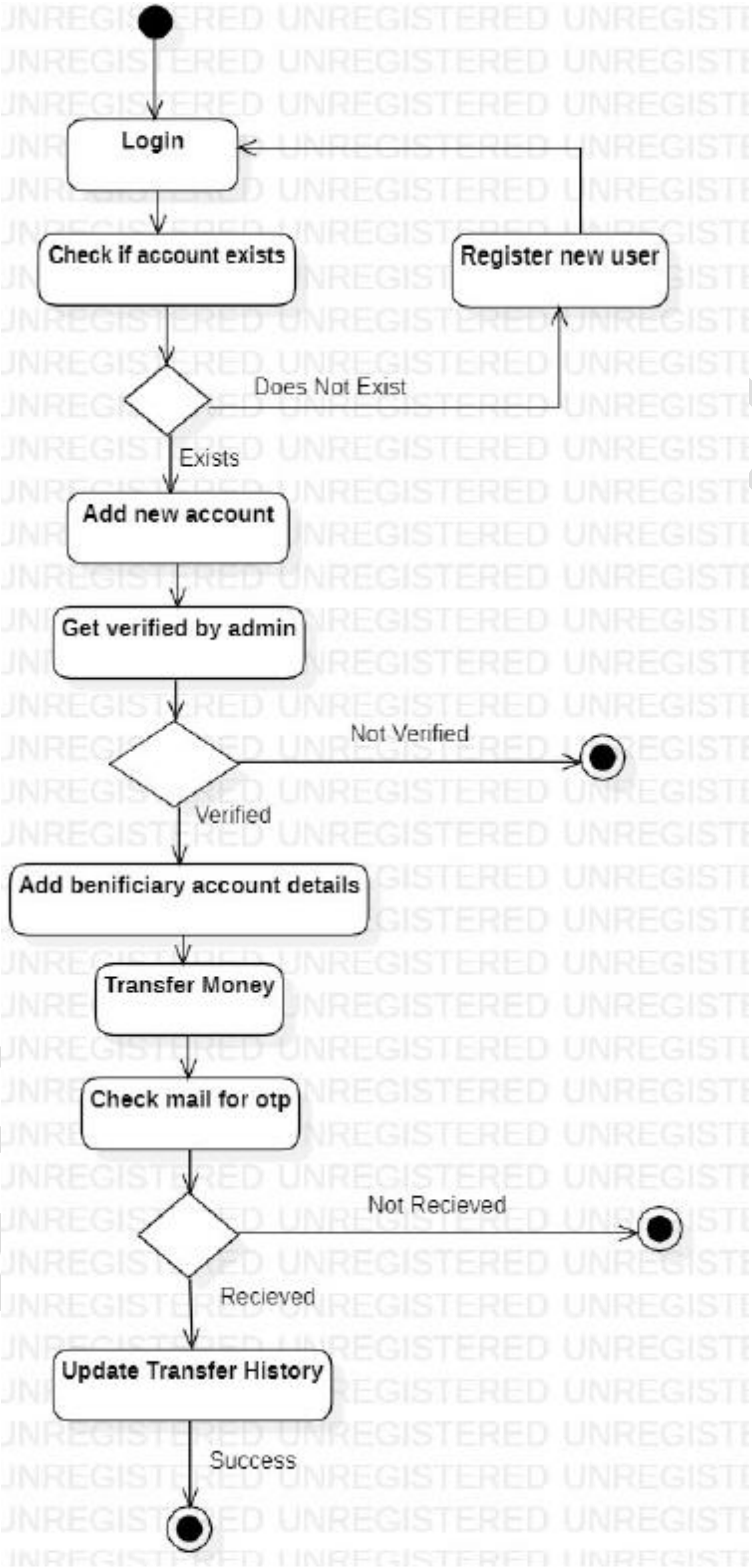
4.1 User case diagram



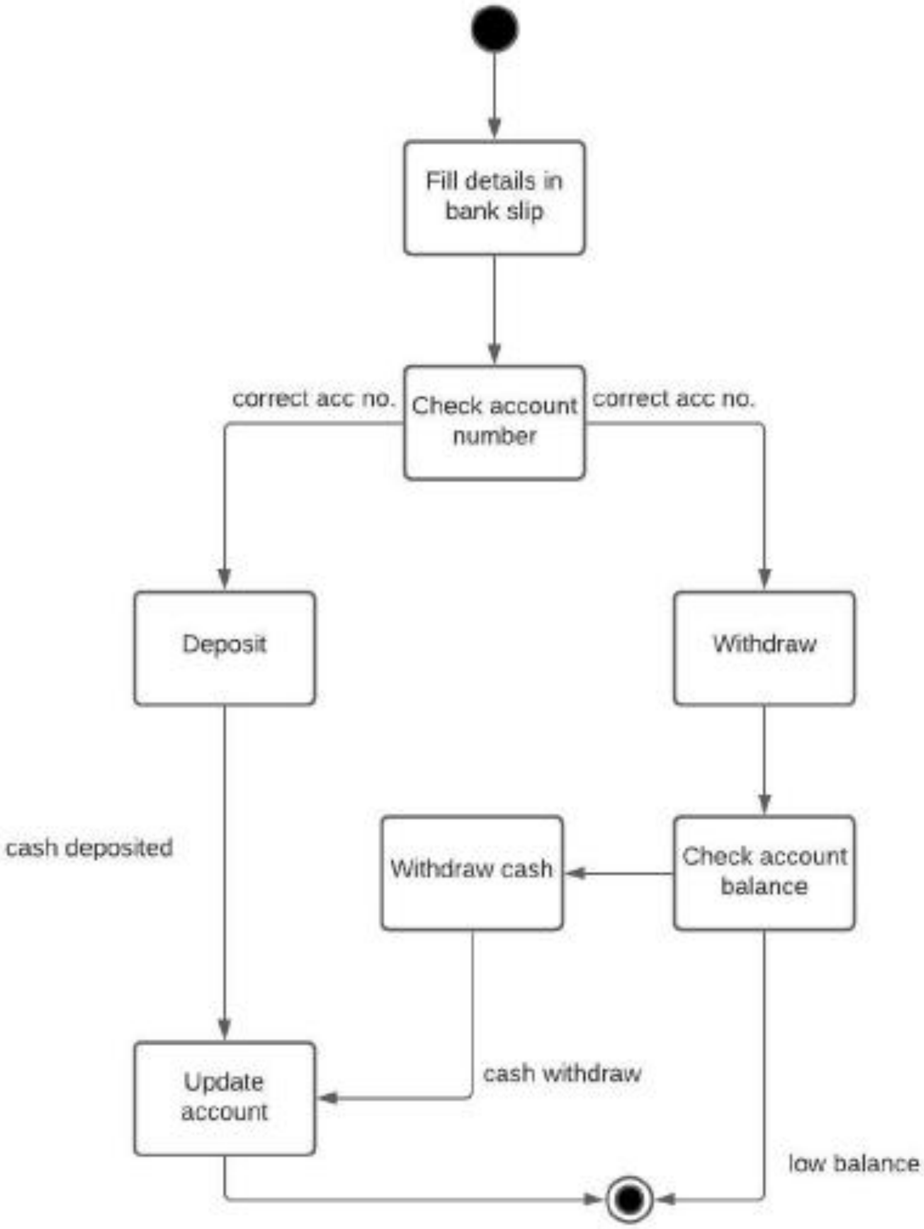
4.2 Sequence diagram



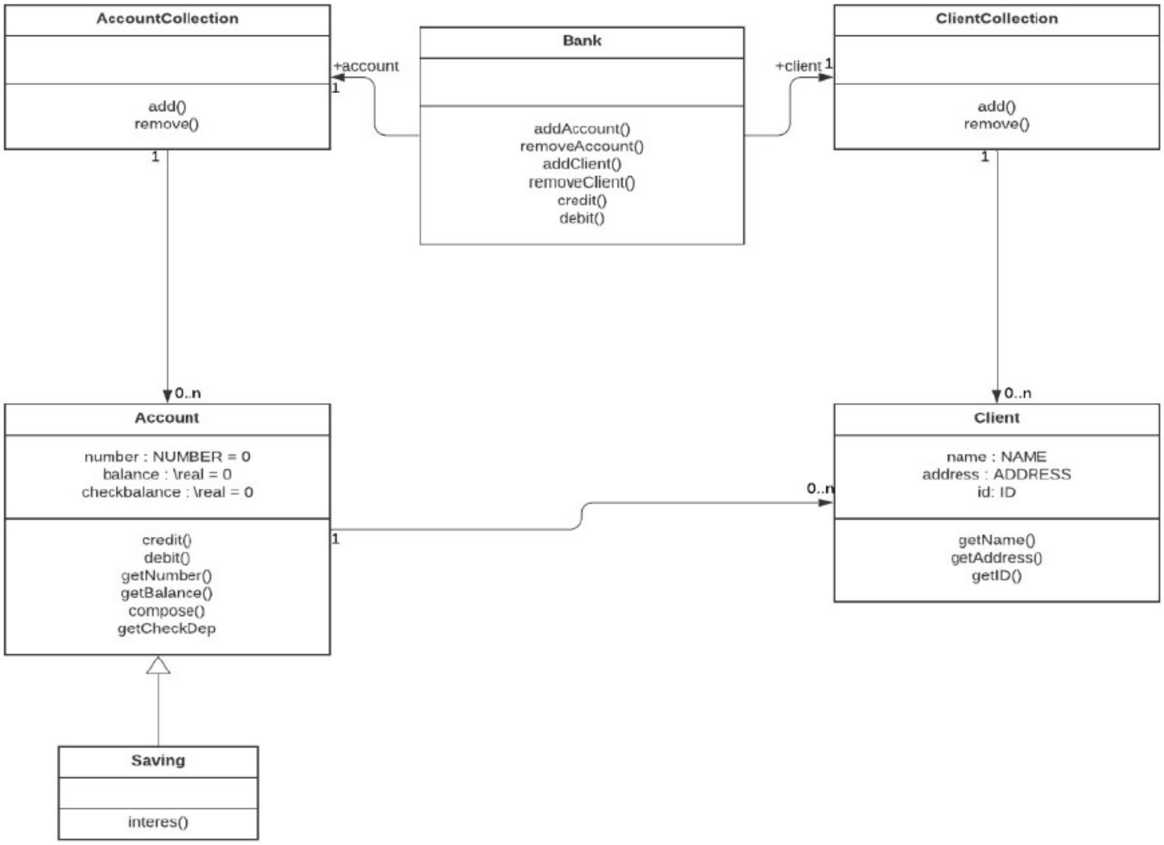
4.3 Activity Diagram



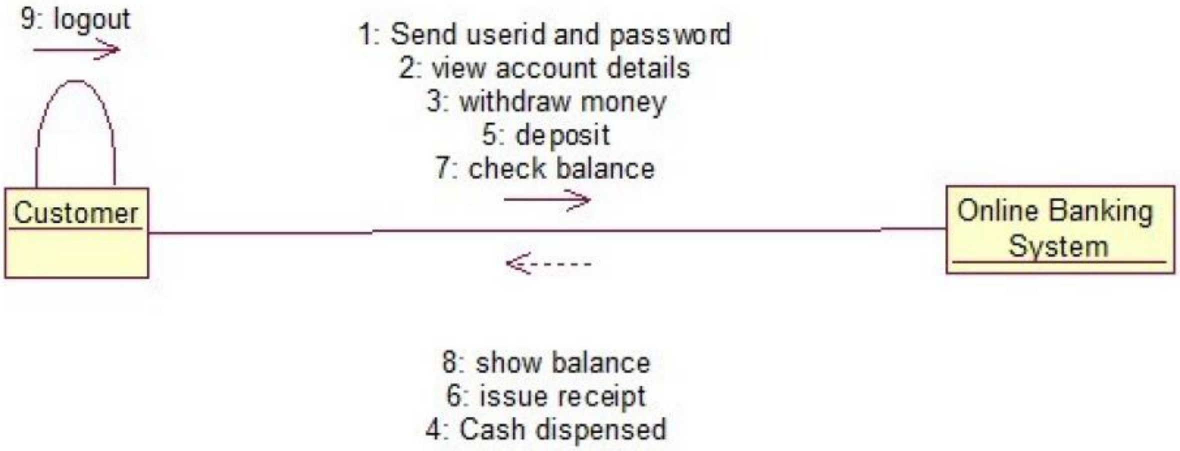
4.4 State Chart Diagram



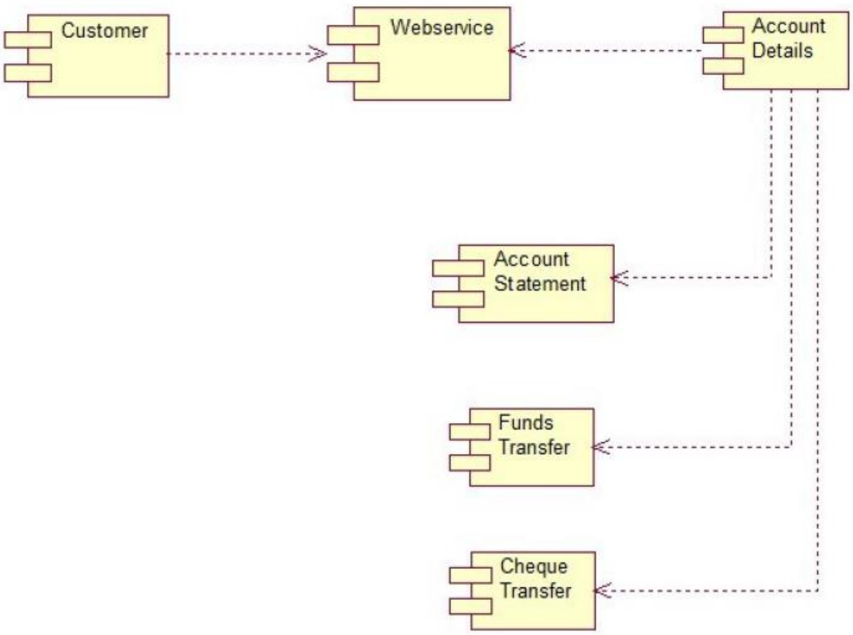
4.5 Class Diagram



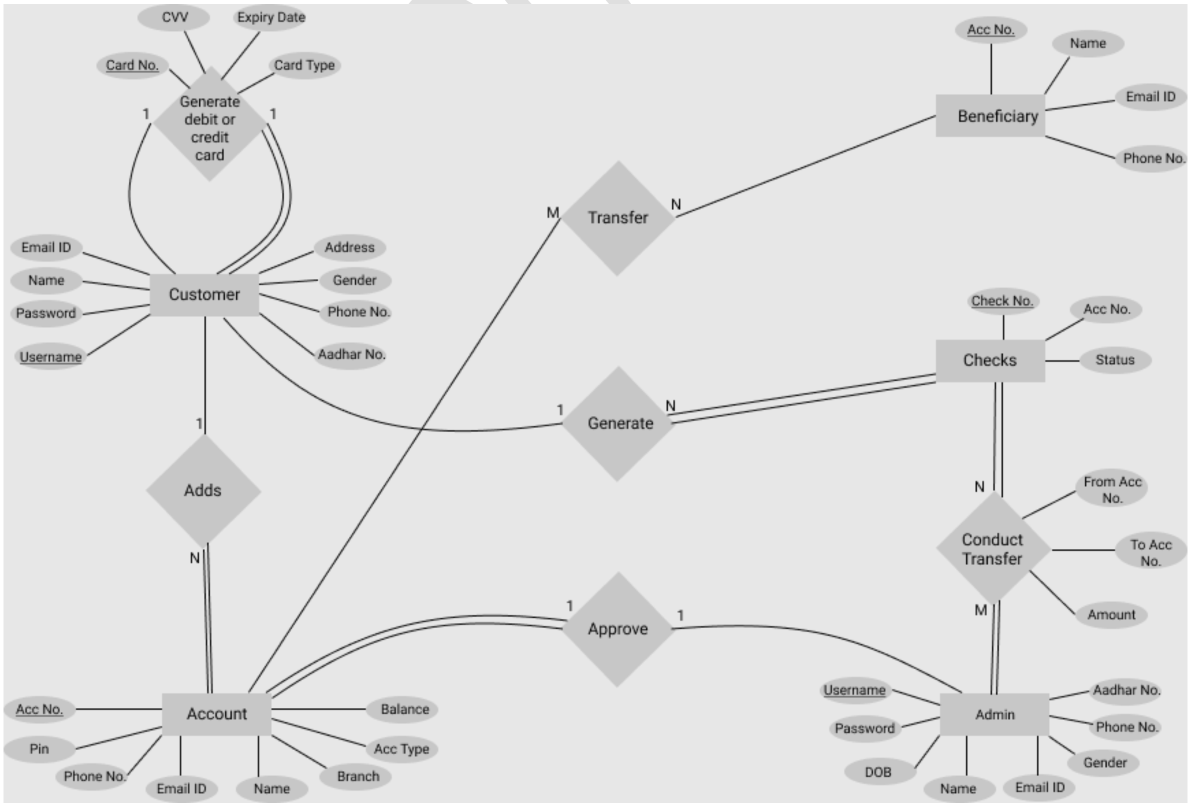
4.6 Collaboration Diagram



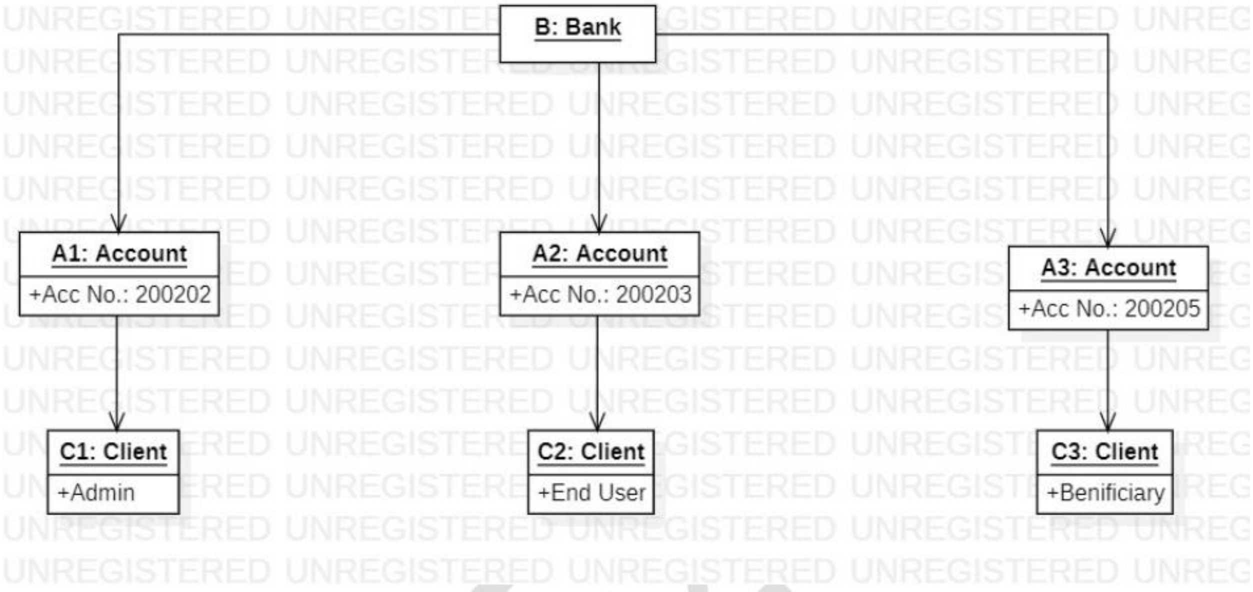
4.7 Component Diagram



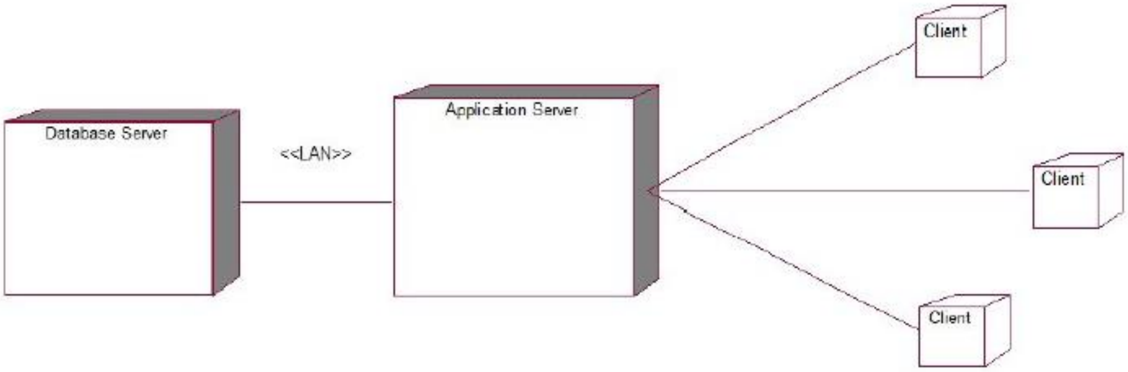
4.8 ER- Diagram



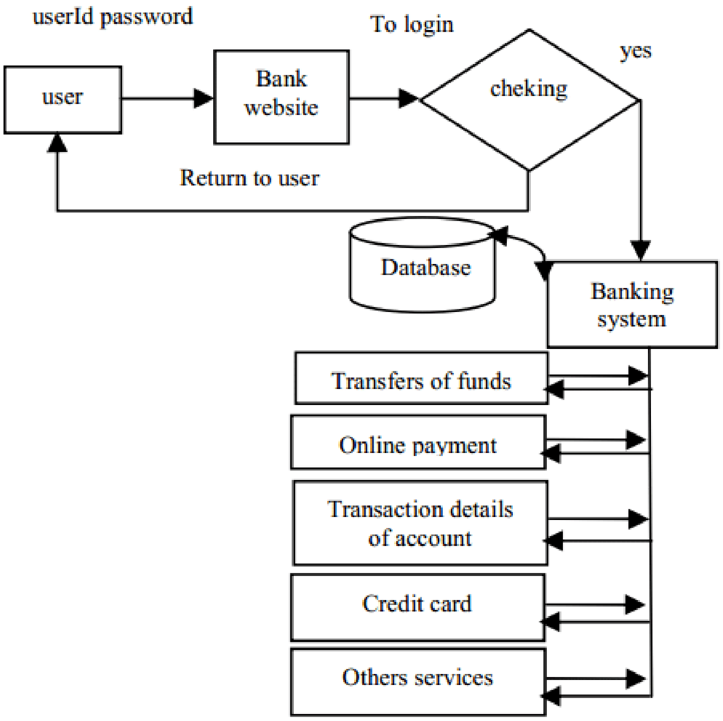
4.9 Object Diagram



4.10 Deployment Diagram

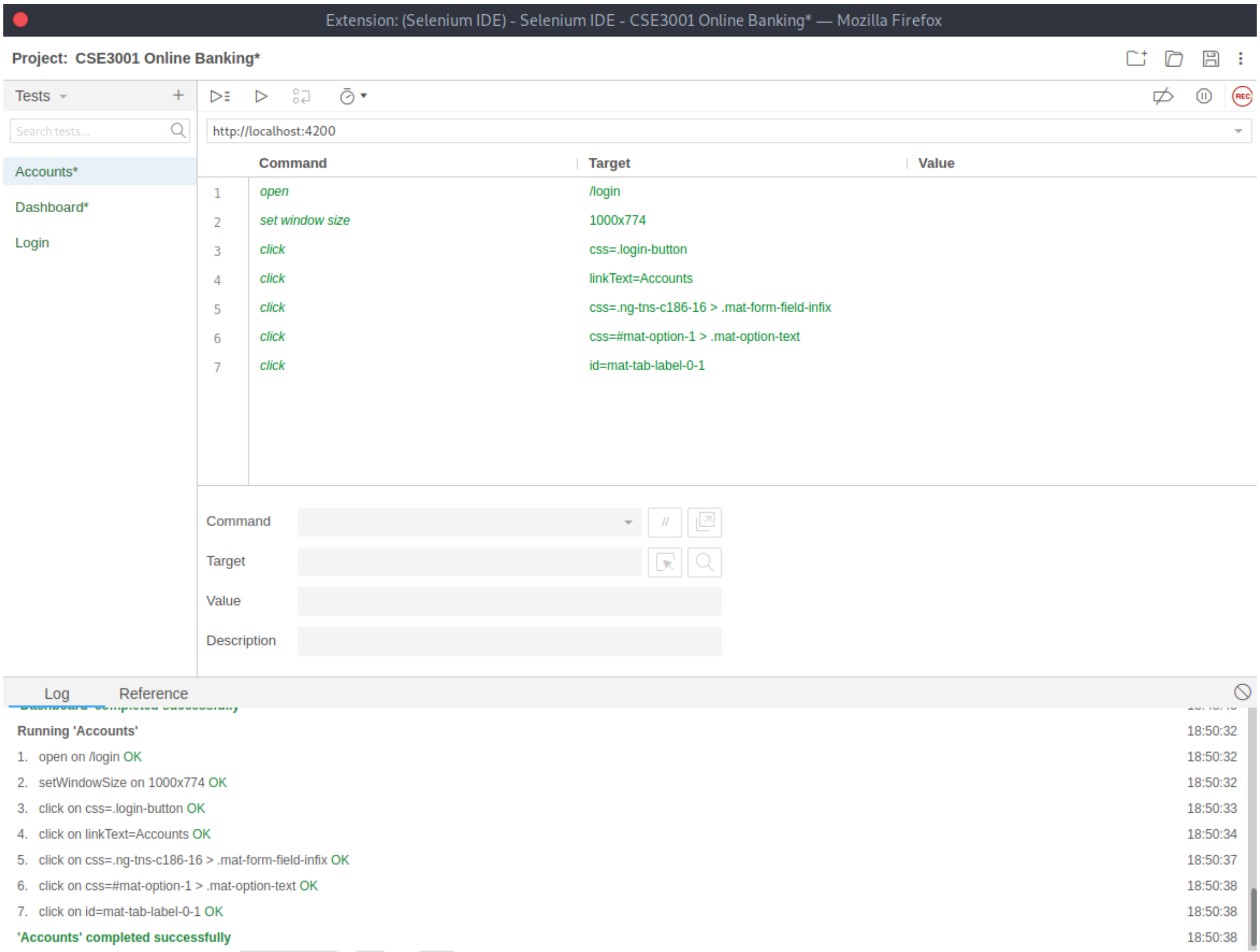


4.11 System Design



5. Test Cases

1. Accounts



Extension: (Selenium IDE) - Selenium IDE - CSE3001 Online Banking* — Mozilla Firefox
Tests
Search tests...
Accounts*
Dashboard*
Login
REC
Log
Reference

Project: CSE3001 Online Banking*

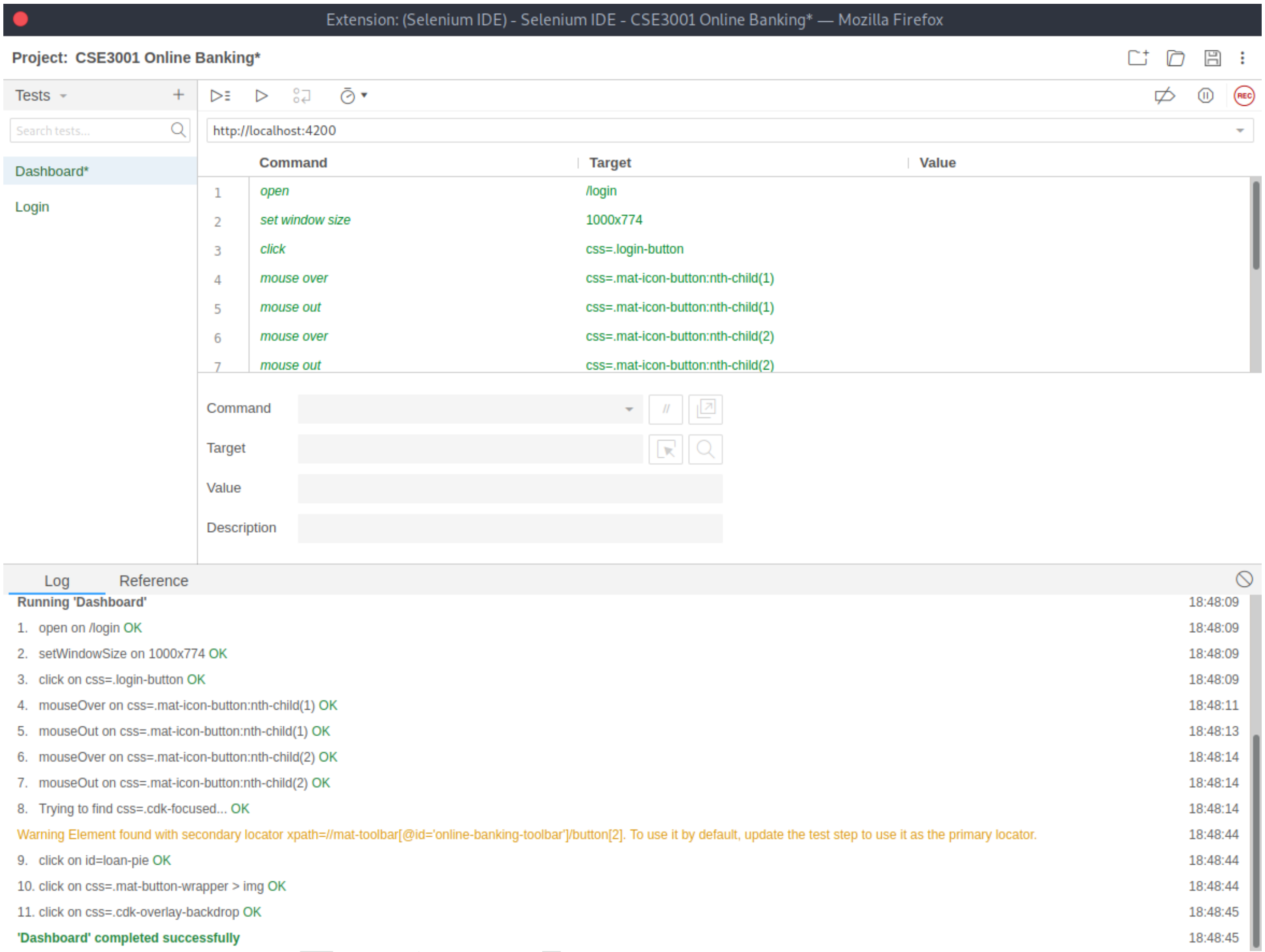
http://localhost:4200

	Command	Target	Value
1	*open*	/login	
2	*set window size*	1000x774	
3	*click*	css=.login-button	
4	*click*	linkText=Accounts	
5	*click*	css=.ng-tns-c186-16 > .mat-form-field-infix	
6	*click*	css=#mat-option-1 > .mat-option-text	
7	*click*	id=mat-tab-label-0-1	

Command	
Target	
Value	
Description	

```
Running 'Accounts'                                                    18:50:32
1.  open on /login OK                                                 18:50:32
2.  setWindowSize on 1000x774 OK                                      18:50:32
3.  click on css=.login-button OK                                     18:50:33
4.  click on linkText=Accounts OK                                     18:50:34
5.  click on css=.ng-tns-c186-16 > .mat-form-field-infix OK           18:50:37
6.  click on css=#mat-option-1 > .mat-option-text OK                  18:50:38
7.  click on id=mat-tab-label-0-1 OK                                  18:50:38
'Accounts' completed successfully                                     18:50:38
```

2. Dashboard


Extension: (Selenium IDE) - Selenium IDE - CSE3001 Online Banking* — Mozilla Firefox
Tests
Search tests...
Dashboard*
Login
REC
Log
Reference

Project: CSE3001 Online Banking*

http://localhost:4200

	Command	Target	Value
1	*open*	/login	
2	*set window size*	1000x774	
3	*click*	css=.login-button	
4	*mouse over*	css=.mat-icon-button:nth-child(1)	
5	*mouse out*	css=.mat-icon-button:nth-child(1)	
6	*mouse over*	css=.mat-icon-button:nth-child(2)	
7	*mouse out*	css=.mat-icon-button:nth-child(2)	

Command

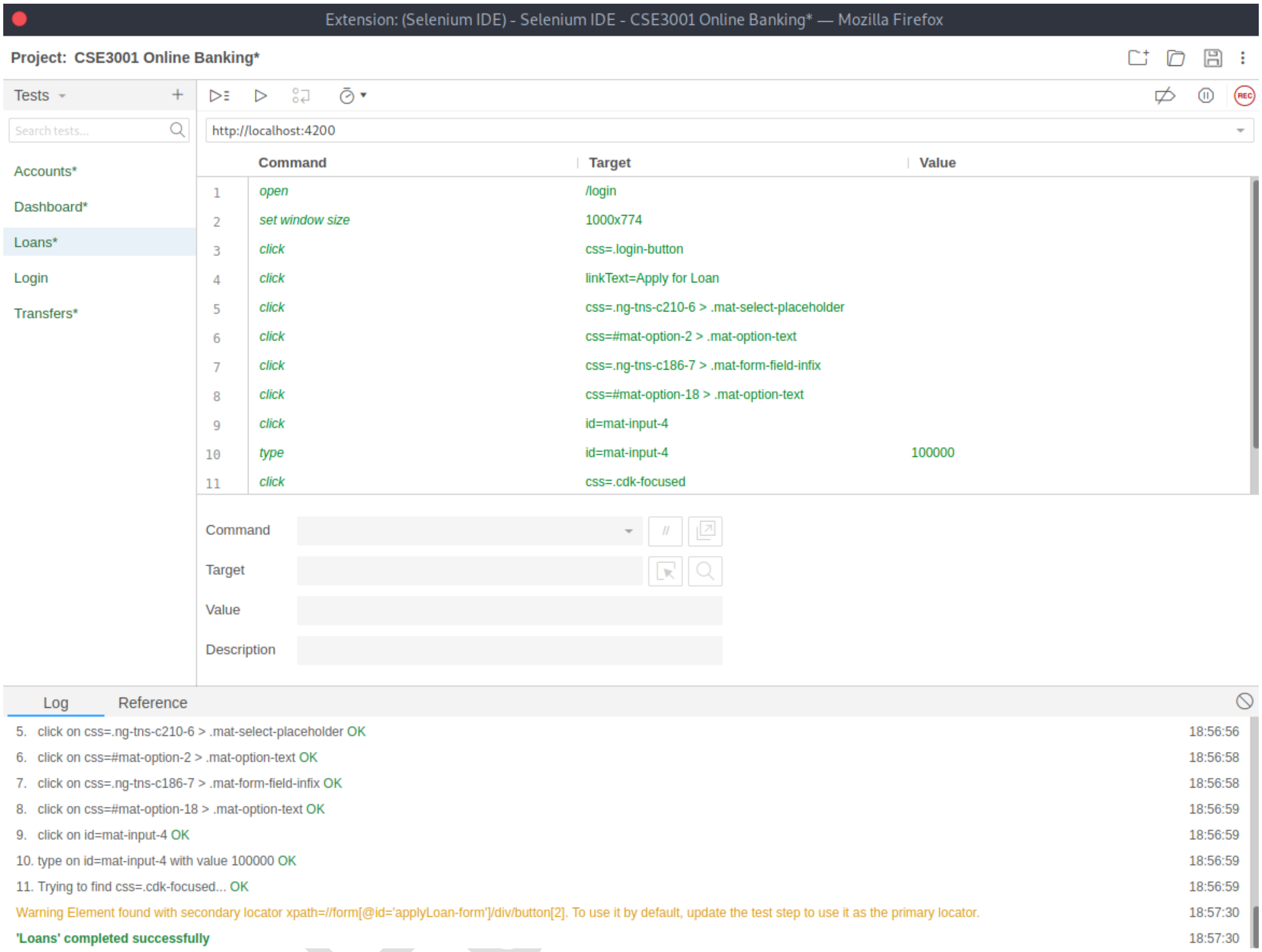
Target

Value

Description

```
Running 'Dashboard'                                                          18:48:09
1.  open on /login OK                                                        18:48:09
2.  setWindowSize on 1000x774 OK                                             18:48:09
3.  click on css=.login-button OK                                            18:48:09
4.  mouseOver on css=.mat-icon-button:nth-child(1) OK                        18:48:11
5.  mouseOut on css=.mat-icon-button:nth-child(1) OK                         18:48:13
6.  mouseOver on css=.mat-icon-button:nth-child(2) OK                        18:48:14
7.  mouseOut on css=.mat-icon-button:nth-child(2) OK                         18:48:14
8.  Trying to find css=.cdk-focused... OK                                    18:48:14
Warning Element found with secondary locator xpath=//mat-toolbar[@id='online-banking-toolbar']/button[2]. To use it by default, update the test step to use it as the primary locator.   18:48:44
9.  click on id=loan-pie OK                                                  18:48:44
10. click on css=.mat-button-wrapper > img OK                                18:48:44
11. click on css=.cdk-overlay-backdrop OK                                    18:48:45
'Dashboard' completed successfully                                           18:48:45
```

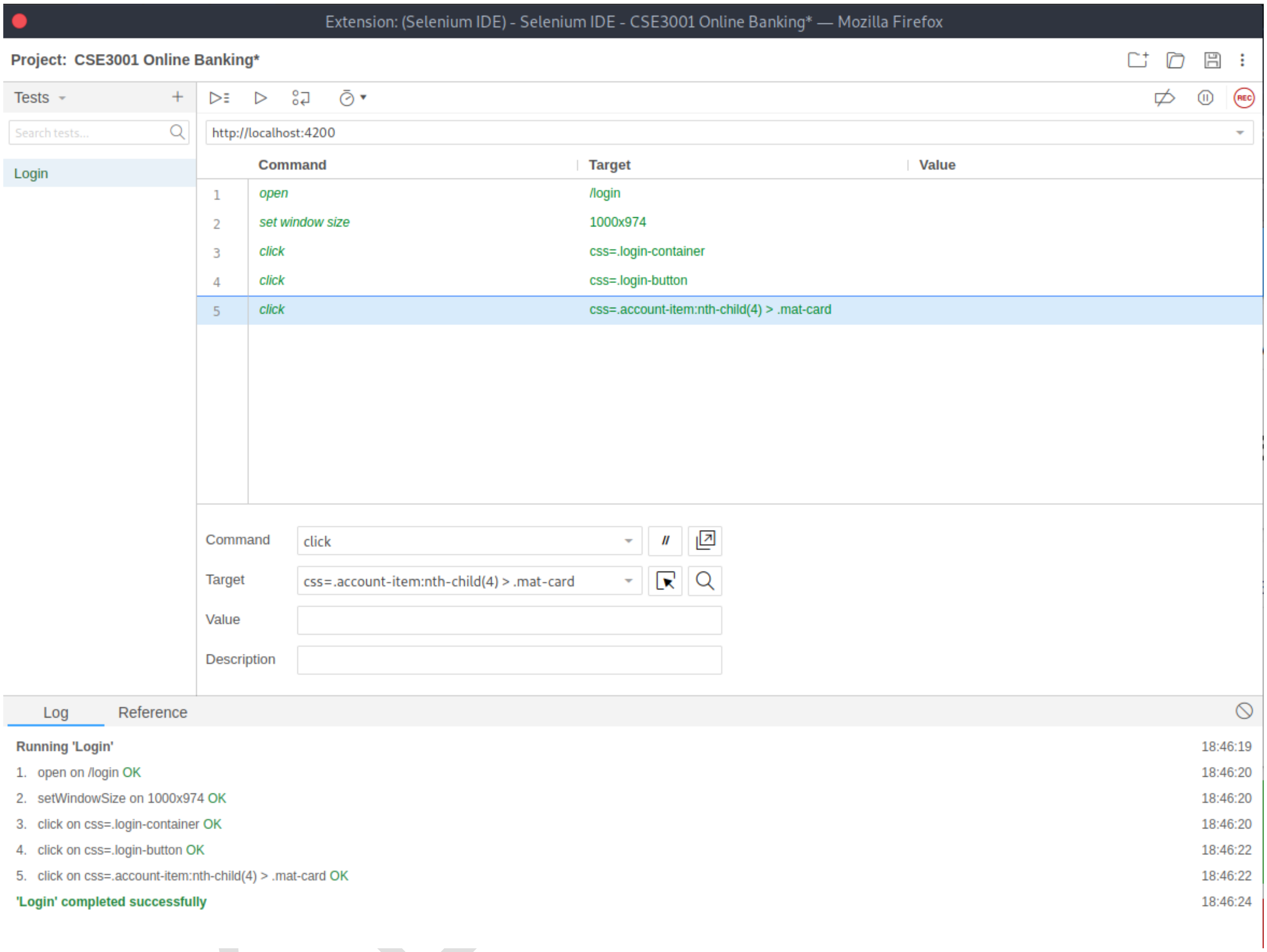
3. Loans



	Command	Target	Value
1	open	/login	
2	set window size	1000x774	
3	click	css=.login-button	
4	click	linkText=Apply for Loan	
5	click	css=.ng-tns-c210-6 > .mat-select-placeholder	
6	click	css=#mat-option-2 > .mat-option-text	
7	click	css=.ng-tns-c186-7 > .mat-form-field-infix	
8	click	css=#mat-option-18 > .mat-option-text	
9	click	id=mat-input-4	
10	type	id=mat-input-4	100000
11	click	css=.cdk-focused	

```
5.  click on css=.ng-tns-c210-6 > .mat-select-placeholder OK                18:56:56
6.  click on css=#mat-option-2 > .mat-option-text OK                        18:56:58
7.  click on css=.ng-tns-c186-7 > .mat-form-field-infix OK                  18:56:58
8.  click on css=#mat-option-18 > .mat-option-text OK                       18:56:59
9.  click on id=mat-input-4 OK                                              18:56:59
10. type on id=mat-input-4 with value 100000 OK                             18:56:59
11. Trying to find css=.cdk-focused... OK                                   18:56:59
Warning Element found with secondary locator xpath=//form[@id='applyLoan-form']/div/button[2]. To use it by default, update the test step to use it as the primary locator.   18:57:30
'Loans' completed successfully                                              18:57:30
```

4. Login



Extension: (Selenium IDE) - Selenium IDE - CSE3001 Online Banking* — Mozilla Firefox

Project: CSE3001 Online Banking*

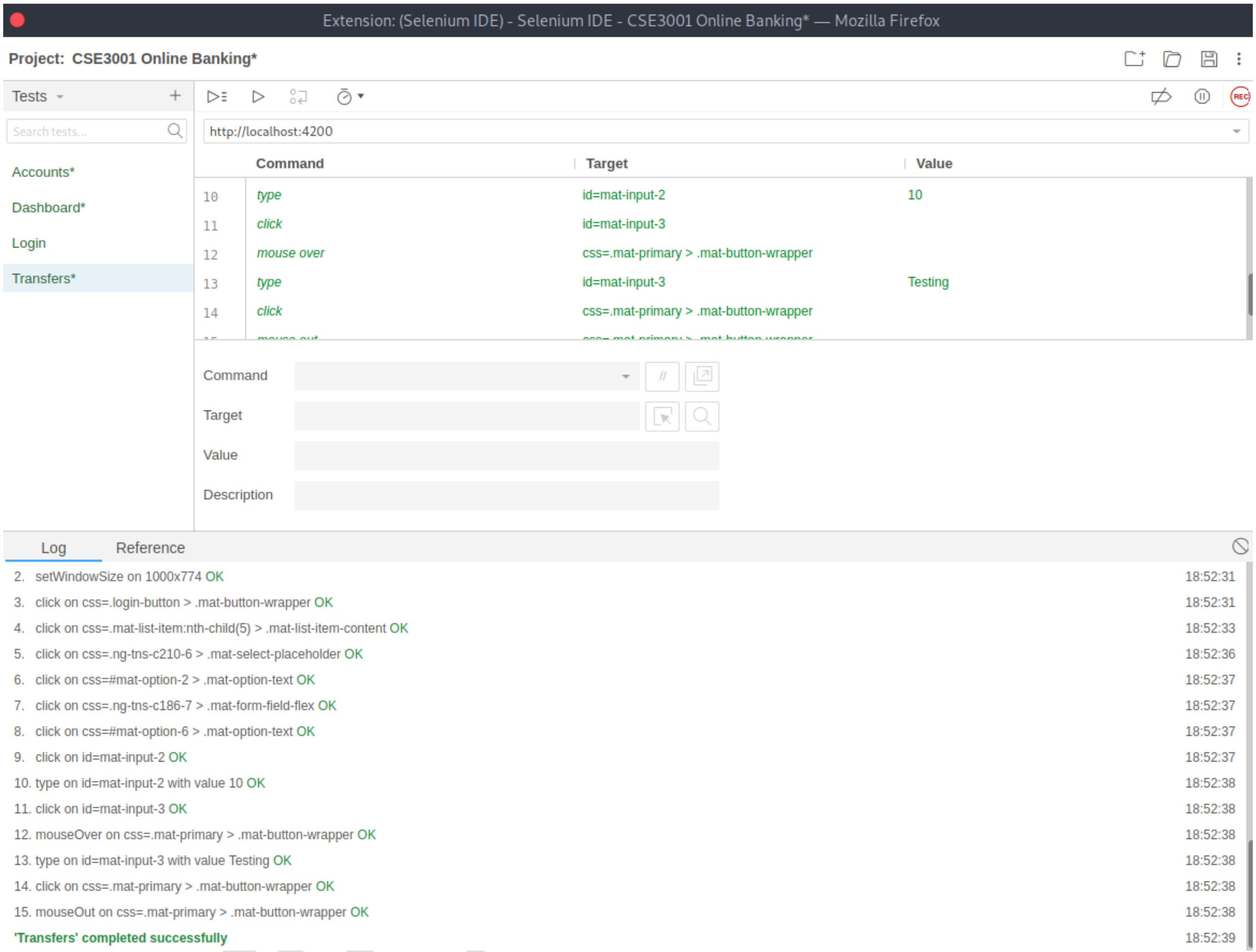
http://localhost:4200

	Command	Target	Value
1	*open*	/login	
2	*set window size*	1000x974	
3	*click*	css=.login-container	
4	*click*	css=.login-button	
5	*click*	css=.account-item:nth-child(4) > .mat-card	

Command	click
Target	css=.account-item:nth-child(4) > .mat-card
Value	
Description	

```
Running 'Login'                                                  18:46:19
1.  open on /login OK                                            18:46:20
2.  setWindowSize on 1000x974 OK                                 18:46:20
3.  click on css=.login-container OK                             18:46:20
4.  click on css=.login-button OK                                18:46:22
5.  click on css=.account-item:nth-child(4) > .mat-card OK       18:46:22
'Login' completed successfully                                   18:46:24
```

5. Transfer

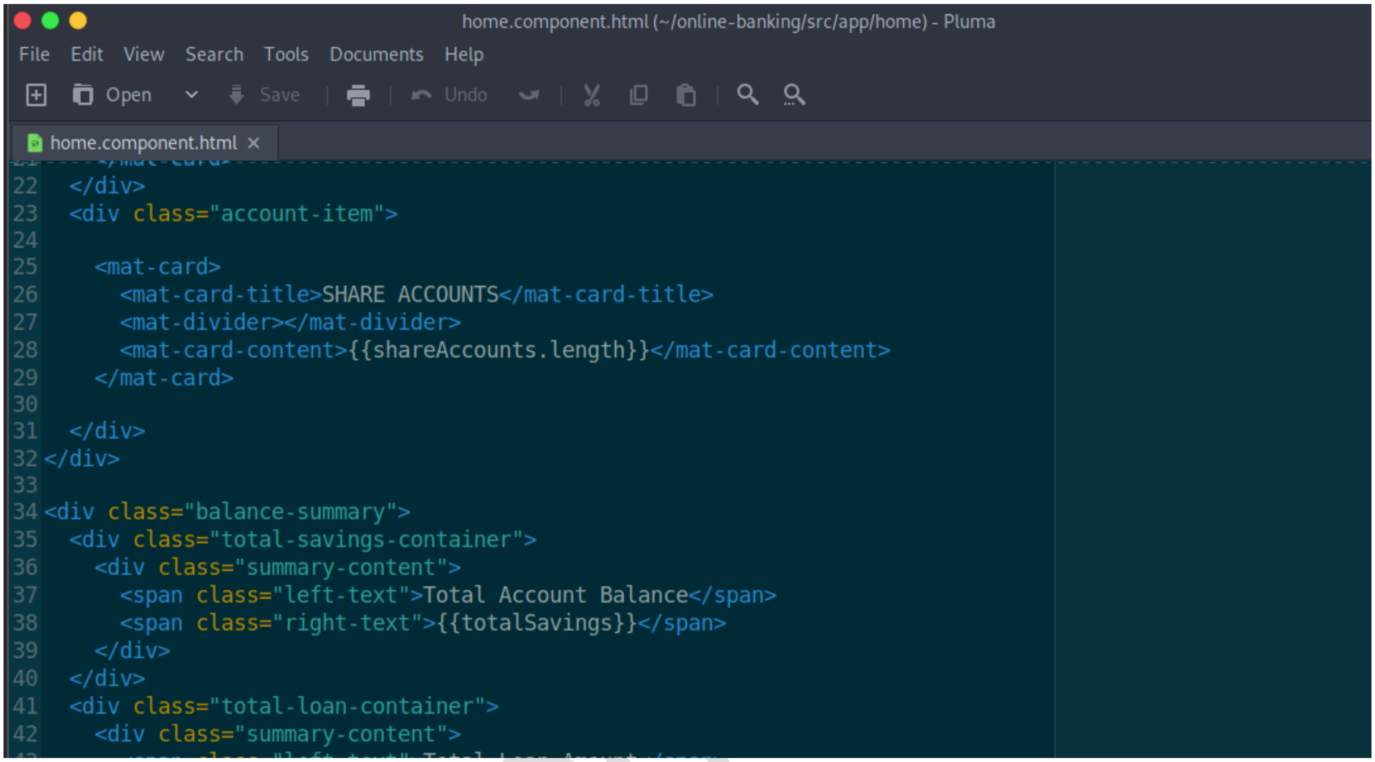


Extension: (Selenium IDE) - Selenium IDE - CSE3001 Online Banking* — Mozilla Firefox
Project: CSE3001 Online Banking*
Tests
Search tests...
Accounts*
Dashboard*
Login
Transfers*
http://localhost:4200
Command
Target
Value
Description
Log
Reference

	Command	Target	Value
10	*type*	id=mat-input-2	10
11	*click*	id=mat-input-3	
12	*mouse over*	css=.mat-primary > .mat-button-wrapper	
13	*type*	id=mat-input-3	Testing
14	*click*	css=.mat-primary > .mat-button-wrapper	

```
2.  setWindowSize on 1000x774 OK                                                18:52:31
3.  click on css=.login-button > .mat-button-wrapper OK                          18:52:31
4.  click on css=.mat-list-item:nth-child(5) > .mat-list-item-content OK         18:52:33
5.  click on css=.ng-tns-c210-6 > .mat-select-placeholder OK                     18:52:36
6.  click on css=#mat-option-2 > .mat-option-text OK                             18:52:37
7.  click on css=.ng-tns-c186-7 > .mat-form-field-flex OK                        18:52:37
8.  click on css=#mat-option-6 > .mat-option-text OK                             18:52:37
9.  click on id=mat-input-2 OK                                                   18:52:37
10. type on id=mat-input-2 with value 10 OK                                      18:52:38
11. click on id=mat-input-3 OK                                                   18:52:38
12. mouseOver on css=.mat-primary > .mat-button-wrapper OK                       18:52:38
13. type on id=mat-input-3 with value Testing OK                                 18:52:38
14. click on css=.mat-primary > .mat-button-wrapper OK                           18:52:38
15. mouseOut on css=.mat-primary > .mat-button-wrapper OK                        18:52:38
'Transfers' completed successfully                                               18:52:39
```

6. Code

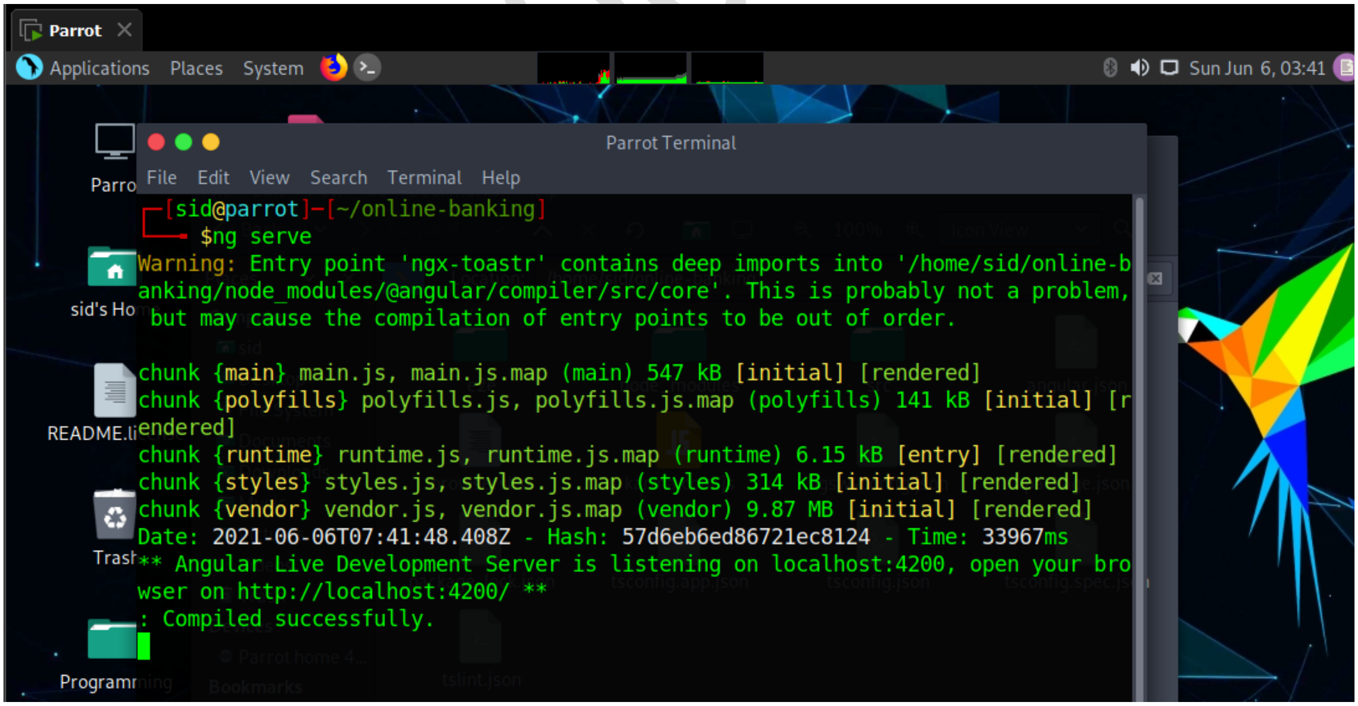

home.component.html (~/online-banking/src/app/home) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
home.component.html

```html
  </div>
  <div class="account-item">

    <mat-card>
      <mat-card-title>SHARE ACCOUNTS</mat-card-title>
      <mat-divider></mat-divider>
      <mat-card-content>{{shareAccounts.length}}</mat-card-content>
    </mat-card>

  </div>
</div>

<div class="balance-summary">
  <div class="total-savings-container">
    <div class="summary-content">
      <span class="left-text">Total Account Balance</span>
      <span class="right-text">{{totalSavings}}</span>
    </div>
  </div>
  <div class="total-loan-container">
    <div class="summary-content">
```



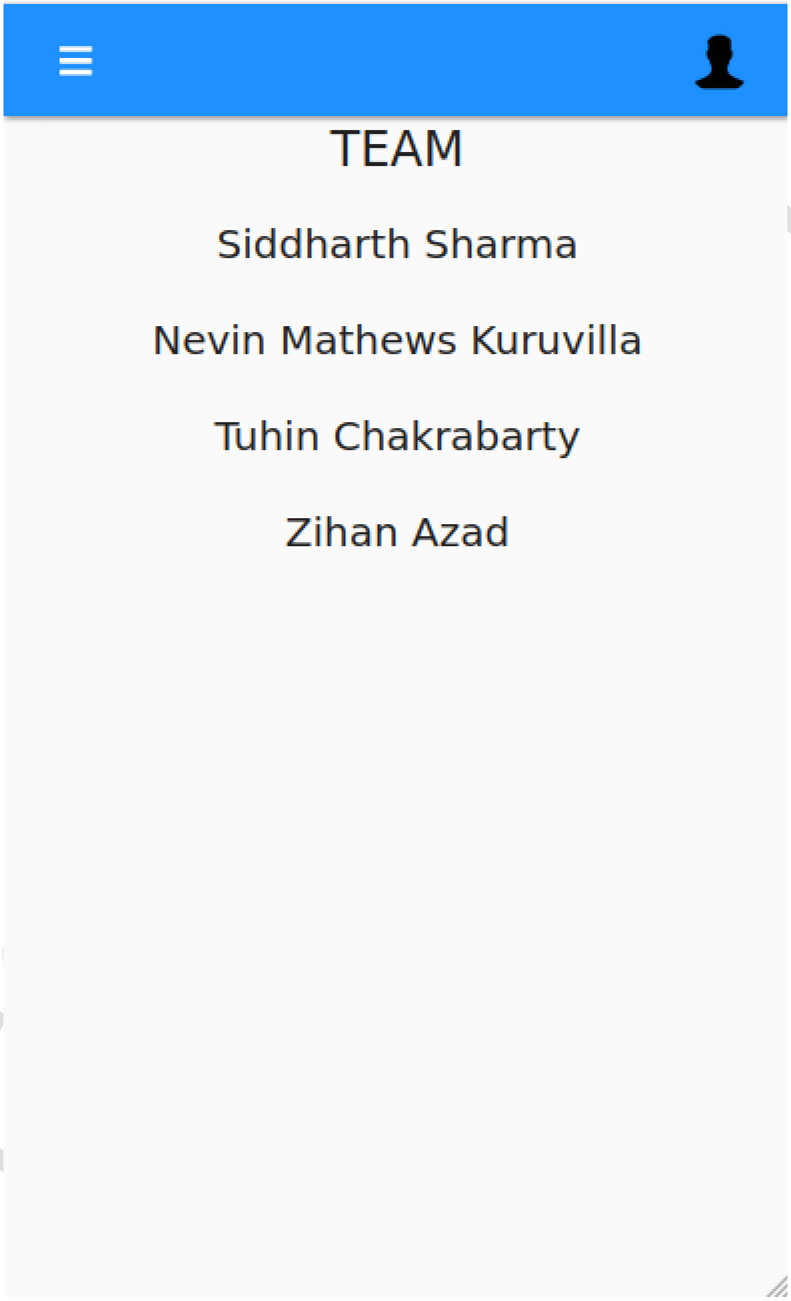
```
┌─[sid@parrot]─[~/online-banking]
└──╼ $ng serve
Warning: Entry point 'ngx-toastr' contains deep imports into '/home/sid/online-b
anking/node_modules/@angular/compiler/src/core'. This is probably not a problem,
 but may cause the compilation of entry points to be out of order.

chunk {main} main.js, main.js.map (main) 547 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [r
endered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 314 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 9.87 MB [initial] [rendered]
Date: 2021-06-06T07:41:48.408Z - Hash: 57d6eb6ed86721ec8124 - Time: 33967ms
** Angular Live Development Server is listening on localhost:4200, open your bro
wser on http://localhost:4200/ **
: Compiled successfully.
```

The full implementation of our code can be found on our GitHub repository: *https://github.com/sidx255/online-banking*

7. Screenshots

About Us



TEAM

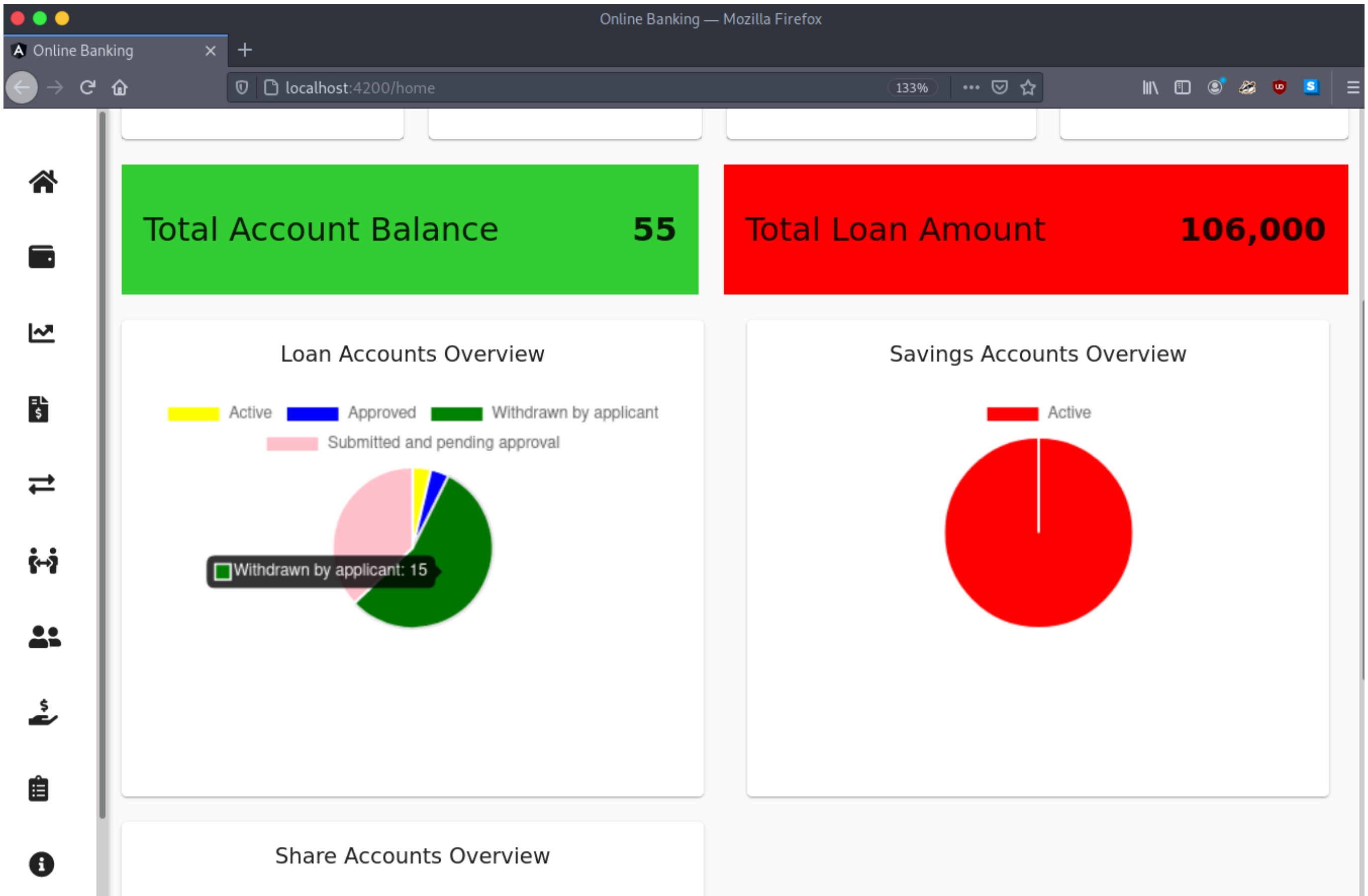
Siddharth Sharma

Nevin Mathews Kuruvilla

Tuhin Chakrabarty

Zihan Azad

Interactive plots


Online Banking — Mozilla Firefox
Online Banking
localhost:4200/home
133%

Total Account Balance 55

Total Loan Amount 106,000

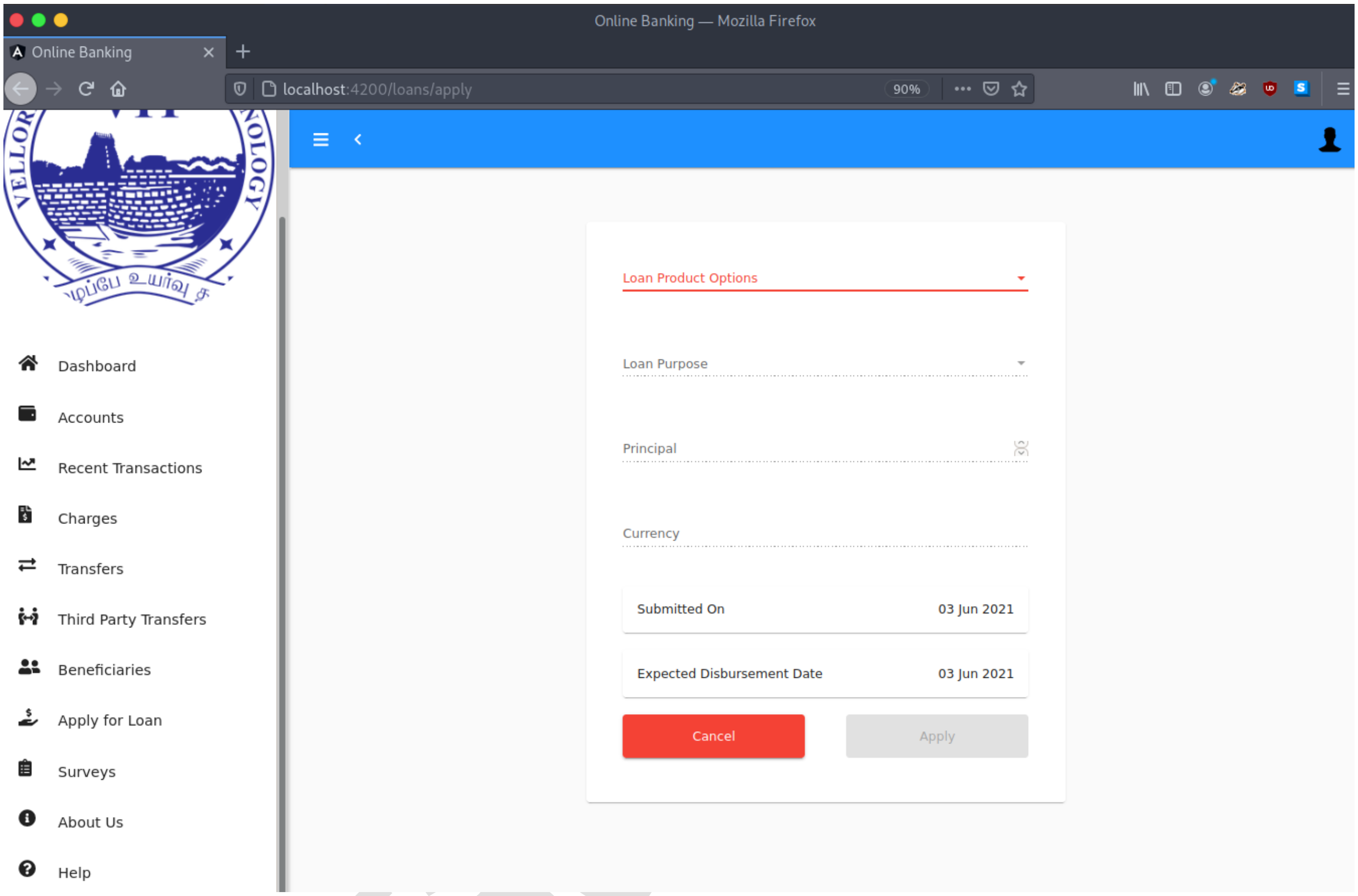
Loan Accounts Overview

🟨 Active 🟦 Approved 🟩 Withdrawn by applicant
Submitted and pending approval
Withdrawn by applicant: 15

Savings Accounts Overview

Share Accounts Overview

Loan form



Loan Product Options

Loan Purpose

Principal

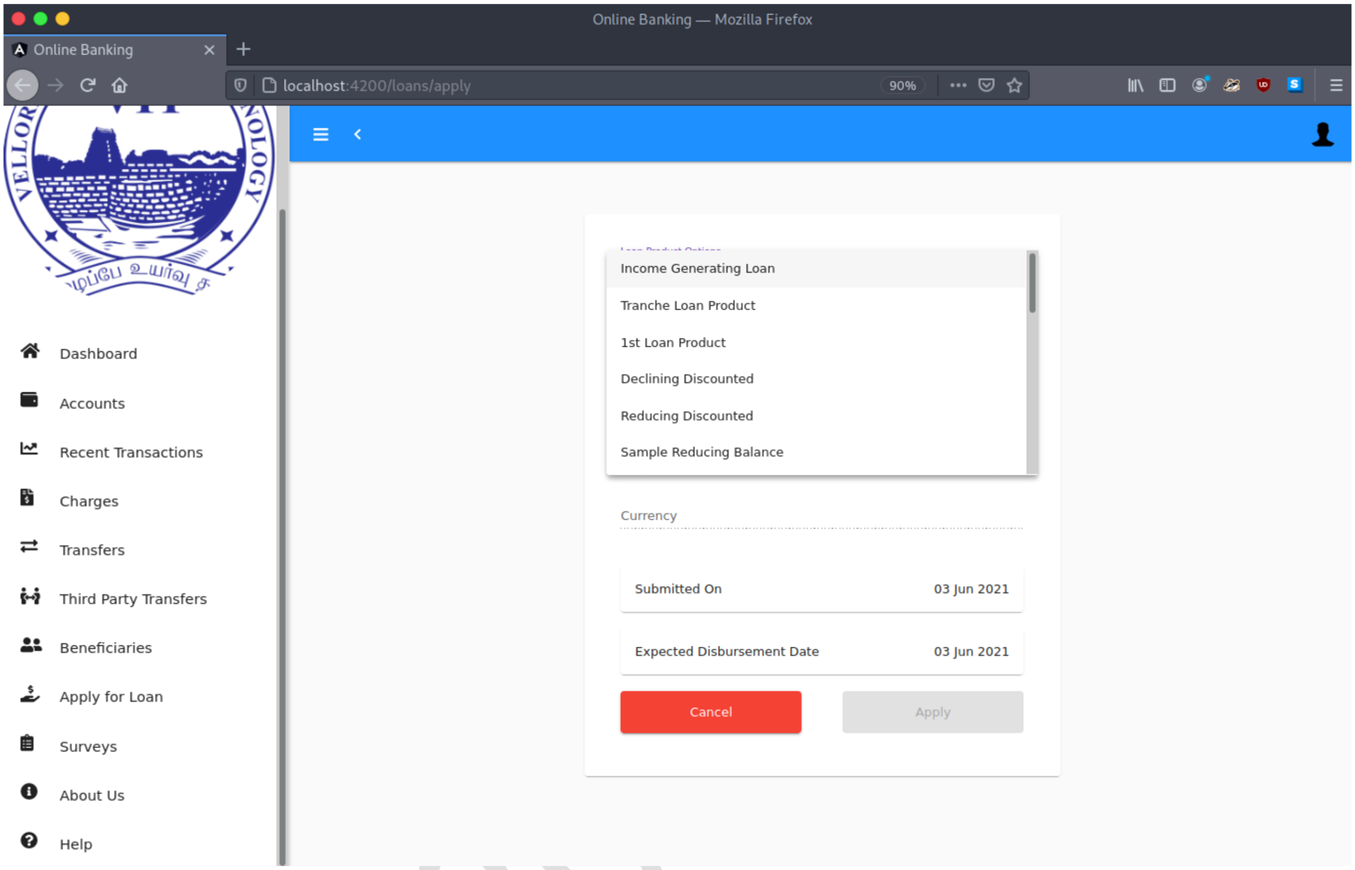
Currency

Submitted On 03 Jun 2021

Expected Disbursement Date 03 Jun 2021

Cancel Apply

Loans

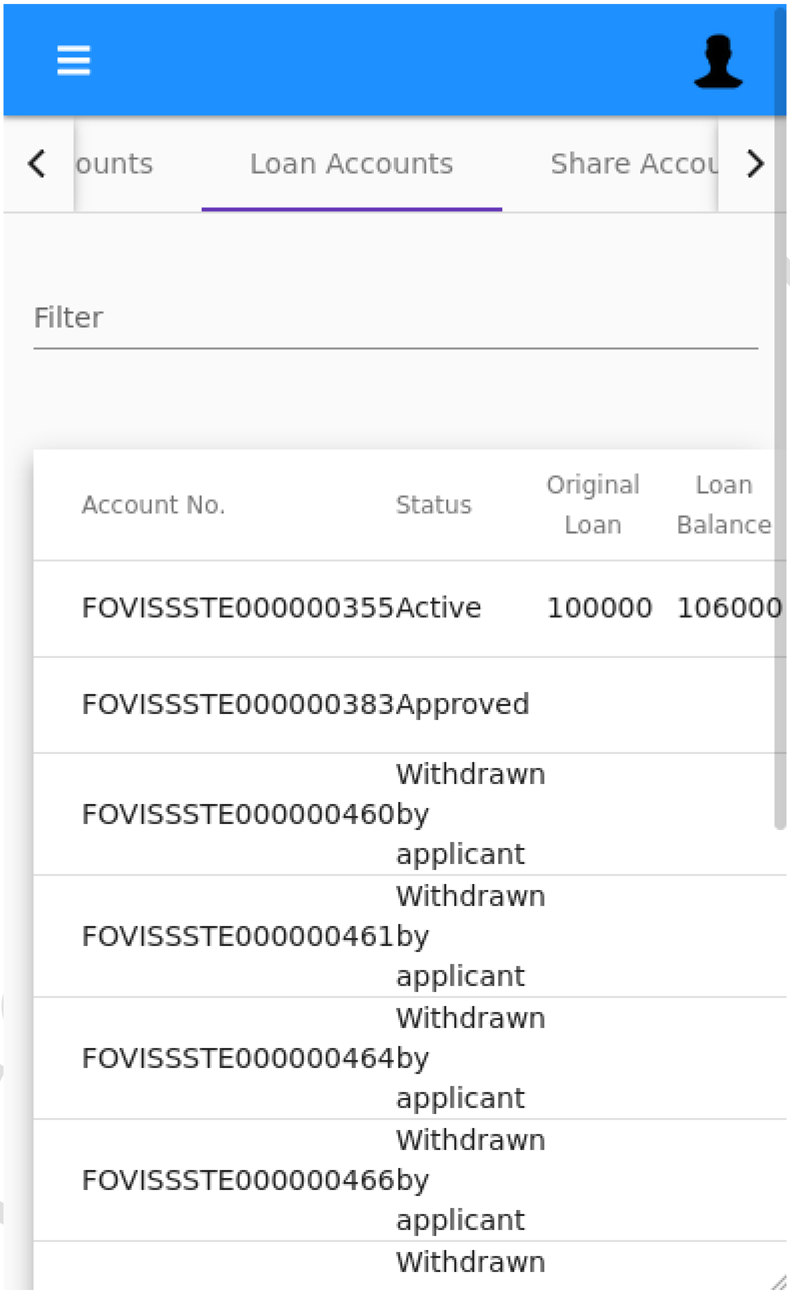


Currency

Submitted On 03 Jun 2021

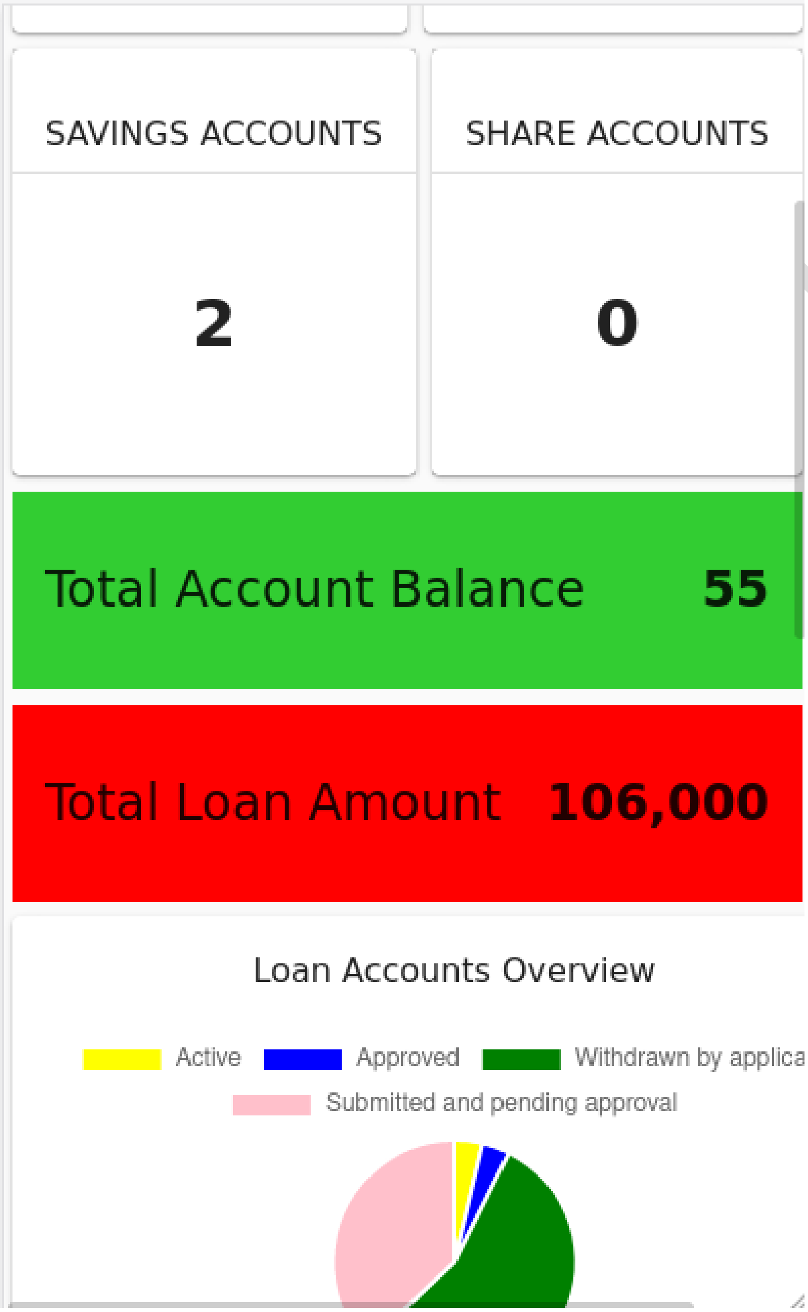
Expected Disbursement Date 03 Jun 2021

Mobile accounts



Account No.	Status	Original Loan	Loan Balance
FOVISSSTE000000355	Active	100000	106000
FOVISSSTE000000383	Approved		
FOVISSSTE000000460	Withdrawn by applicant		
FOVISSSTE000000461	Withdrawn by applicant		
FOVISSSTE000000464	Withdrawn by applicant		
FOVISSSTE000000466	Withdrawn by applicant		
	Withdrawn		

Mobile dashboard



SAVINGS ACCOUNTS

2

SHARE ACCOUNTS

0

Total Account Balance 55

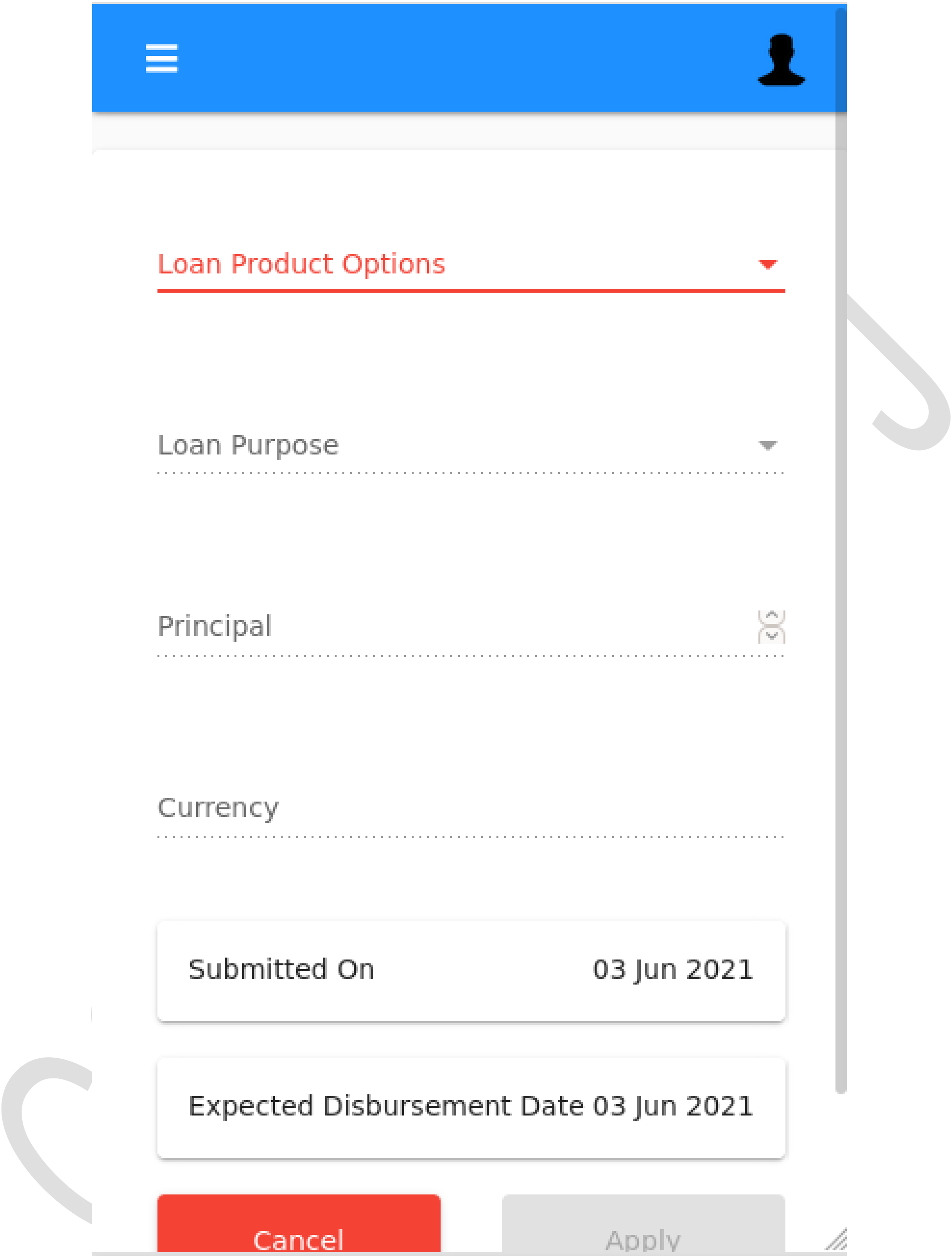
Total Loan Amount 106,000

Loan Accounts Overview

🟨 Active 🟦 Approved 🟩 Withdrawn by applica

Submitted and pending approval

Mobile loan



Loan Product Options

Loan Purpose

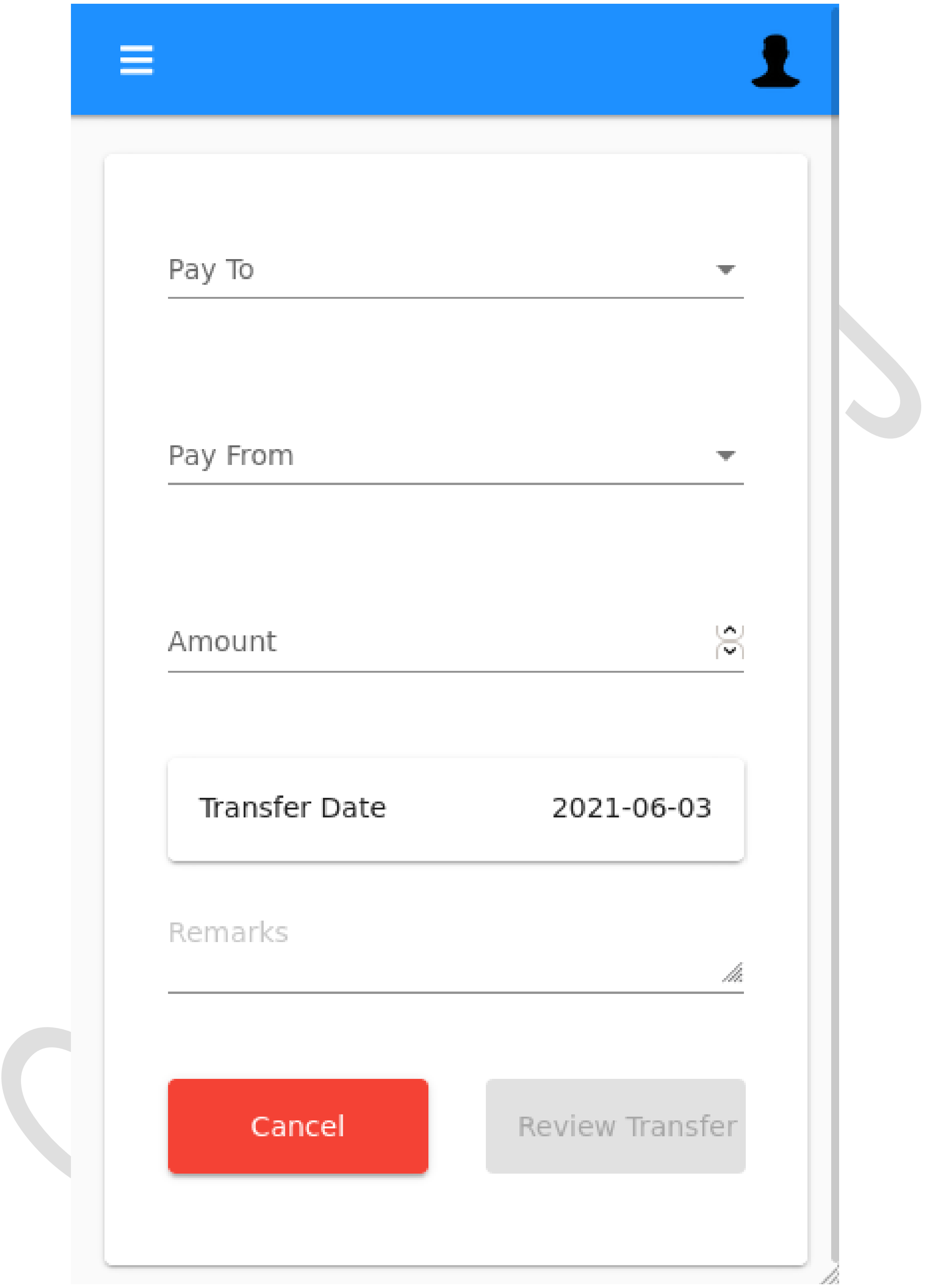
Principal

Currency

Submitted On 03 Jun 2021

Expected Disbursement Date 03 Jun 2021

Mobile transaction



Transfer Date 2021-06-03

8. Conclusion & Future work

In conclusion, our online banking system was a successful project. It has successfully run all test cases and functions as a simple yet intuitive banking portal. Our main objective was to provide the users a hassle free way for banking and we have successfully achieved it. Our team consisted of personals who are not experienced or skilled in web development, yet we took it as a challenge and worked together as a team to complete this project.

For our future work, we would like to increase the security for the system using cryptographic systems. Banking systems are prone to attacks and we must make sure that none of our customers' data gets breached or their identities get compromised. A standard AES encryption and decryption system would definitely increase the security. Using a hash such as SHA-256 to hash and store passwords, would make our customers feel more secure and will trust with their information.

9. References

https://krazytech.com/projects
https://relevant.software/blog/software-requirements-specificationsrs-document/