./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1.1 | 11-11-2020 | Milind Mohapatra(99002670) |  |  |  |
| 1.1 | 11-11-2020 | S P Bala Sirisha(99002653) |  |  |  |
| 1.1 | 11-11-2020 | Anubhav Reddy S(99002481) |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc56068948)

[List of Figures 3](#_Toc56068949)

[Mini Project [Team]- Mobile Bank Application 4](#_Toc56068950)

[**Module** 4](#_Toc56068951)

[Topic and Subtopics 4](#_Toc56068952)

[**Objectives & Requirements** 4](#_Toc56068953)

[Objectives 4](#_Toc56068954)

[Requirements 4](#_Toc56068955)

[High Level Requirements 4](#_Toc56068956)

[Low Level Requirements 5](#_Toc56068957)

[Requirement Mapping 5](#_Toc56068958)

[**Design** 5](#_Toc56068959)

[**Test Plan** 7](#_Toc56068960)

[**Implementation Summary** 8](#_Toc56068961)

[Video Summary 8](#_Toc56068962)

[Git Link 8](#_Toc56068963)

[Git Dashboard 8](#_Toc56068964)

[**Individual Contribution & Highlights** 11](#_Toc56068965)

[Summary 11](#_Toc56068966)

[Challenges faced and how were they overcome 11](#_Toc56068967)

[Future Scope (If applicable) 11](#_Toc56068968)

# List of Figures

[Figure 1 System Level UML Diagram for Bank Application 6](#_Toc56068939)

[Figure 2 Sub system Design for Bank Application 7](#_Toc56068940)

[Figure 3 Java Git Dashboard 8](#_Toc56068941)

[Figure 4 Java Git Inspector 9](#_Toc56068942)

[Figure 5 Java Build 10](#_Toc56068943)

[Figure 6 Java Codacy 10](#_Toc56068944)

# Mini Project [Team]- Mobile Bank Application

## **Module**

Core Java programming and Testing

### Topic and Subtopics

Project Name - Mobile Banking Application.

Description- Mobile Banking is the most preferred way to avail Banking services in the present digital world.

The Mobile banking is different from internet banking as it uses a software, the app we develop consists of all the necessary banking services including main services like account balance, user details, transactions etc.

This application is secured has it will have the user registered details and doesn’t allow unknown person to have access.

## **Objectives & Requirements**

### Objectives

* To develop a Mobile Banking Application using Core Java Programming.
* The Mobile Banking Application will contain all the necessary Banking services.
* Application should be User friendly and highly secured.

### Requirements

### *High Level Requirements*

* The Software should be highly secured.
* Application should be a multifunctional service provider containing all the necessary Banking services.
* Once the application is opened, it must have login, registration and exit as the fields to access.
* Application must store all the user details after registration.
* The software has to have the transactions, view balance, Account details etc. as the basic banking operations.
* The logout and exit fields should work as per the standards.

### *Low Level Requirements*

* User class should take all the user registration details such as name, gender etc. and store it in a file.
* Login should be an outer class which verifies the user details and allows the access for banking services.
* Account details should be of an abstract class containing a details of savings, RD etc.

### *Requirement Mapping*

|  |  |
| --- | --- |
| **Product ID** | **Description** |
| H\_01 | High Level: - Should be highly secured |
| H\_02 | High Level: - Software Application should be multifunctional |
| H\_03 | Application should contain login, registration and exit as basic fields |
| H\_03\_L\_01 | High Level: - After opening an app registration field is a must  Low Level: - After clicking on Registration the user class must be able to take  user details as input and store it in a file |
| H\_03\_L\_02 | High Level: - Login Field  Low Level: - After user entering the details it should verify with the existing  Users and allow the access. |
| H\_04 | High Level: After logging in. it should contain all the basic necessary banking  services |
| H\_03\_L\_03 | High Level: Login Field  Low Level: Account details of the user should be hidden and displayed only  when user asks |
| H\_05 | After Completion of the services User should be able to logout and exit the  application |

## **Design**

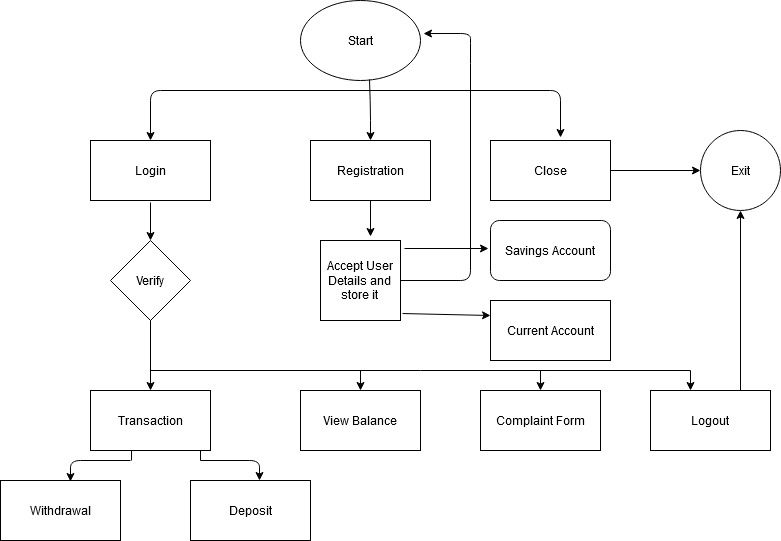


Figure 1 System Level UML Diagram for Bank Application

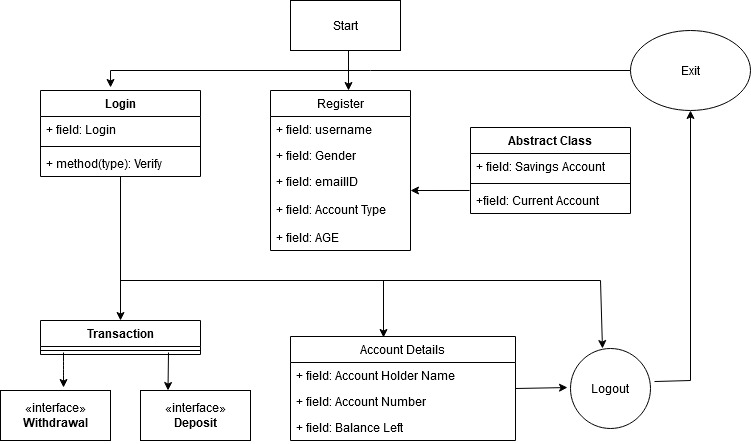


Figure 2 Sub system Design for Bank Application

## **Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Expected** | **Result** |
| T01\_H01 | No access should be allowed without login | Invalid login credentials must restrict login | Passed |
| T02\_H02 | Application must serve multiple purposes | Application should have more than one functionality | Passed |
| T03\_H03 | App must give options for login, register and exit | Menu should be displayed to user having options of login, register and exit | Passed |
| T03\_H03\_L01 | App must register the user | App must take user data and write to a file | Passed |
| T03\_H03\_L02 | App must login the user | App must take user data and verify against user file and allow login if credentials match | Passed |
| T03\_H04 | App must provide basic banking services after login | After login, user is provided with a menu of basic banking operations | Passed |
| T04\_H05\_L01 | User must be able to see account details | Users account details are shown when user wants to see | Passed |
| T04\_H05\_L02 | User must be able to perform transactions | User can deposit or withdraw money in their account | Passed |
| T04\_H05\_L03 | User must be able to see available balance | User’s account balance must be shown | Passed |
| T05\_H06 | User must be able to logout | User logs out | Passed |

## **Implementation Summary**

The bank class provides the user with register and login functionalities and once an user logs in, the control is transferred to the user class. Then the user is provided with different banking options. If the user chooses to make a transaction, then the transaction interface and the transaction implementation class handles that. If the user chooses to view balance then the user balance is displayed. The user details are members of an Account abstract class that are extended by different types of account classes.

### Video Summary

<https://github.com/99002670/2009MYSEMB03-JAVA-4/tree/master/5.PPT%20and%20Video>

### Git Link

[Github Repository](https://github.com/99002670/2009MYSEMB03-JAVA-4)

### Git Dashboard

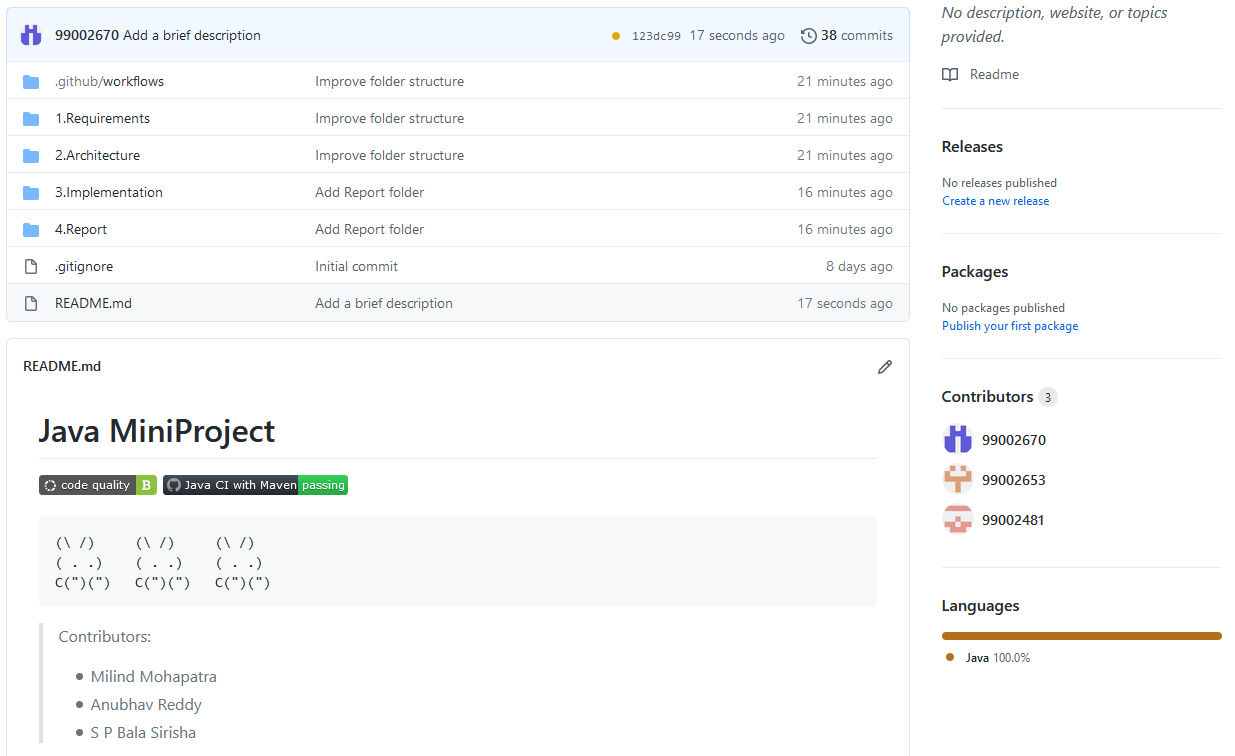


Figure 3 Java Git Dashboard

#### **Git inspector summary**

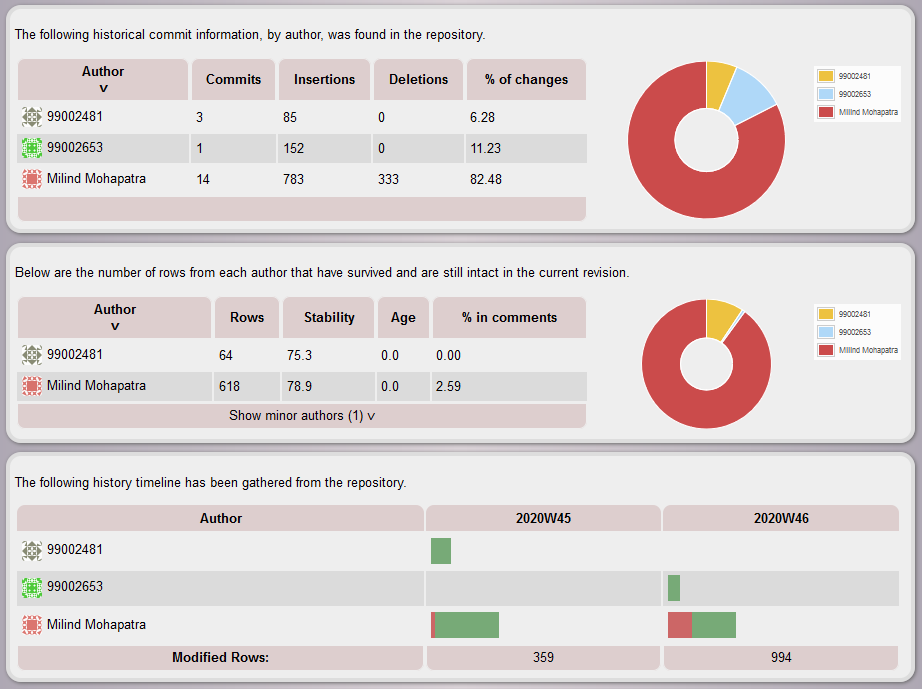


Figure 4 Java Git Inspector

#### **Build**

We built the project using a maven workflow with the following build command

<mvn package -f BankApplication/pom.xml>

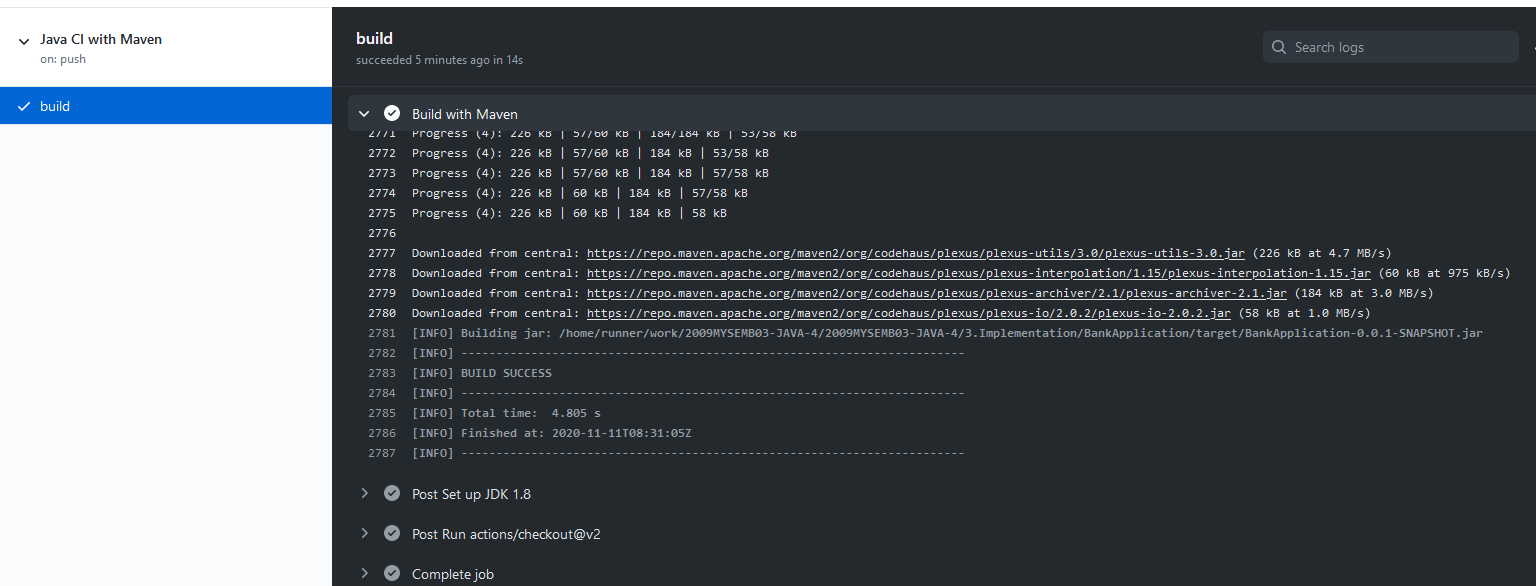


Figure 5 Java Build

#### **Code quality and Issues or Bug Tracking**



Figure 6 Java Codacy

## 

## **Individual Contribution & Highlights**

S P Bala Sirisha, 99002653:

1.Abstract class creation and interfacing with account details and type of account

2. Git Documentation

Anubhav Reddy S, 99002481:

1.Created transaction class and interface.

2. Git Documentation

Milind Mahopatra, 99002690:

1.Created account class, login and registration class

2.Integrated the classes and performed Git Build

### 

### Summary

The GitHub repository has the following files and each file has a README.md file which explains documentation of the project:

1.       Requirements: This contains the high level and low level requirements of the project

2.       Architecture: This contains the necessary UML diagrams present in the project

3.       Bank Application: This contains the source code of the following project

4.       Test Plan: This contains the test plan and the obtained outputs of the project

5.       Report: This contains the report of the project

6.       Images and videos: This contains the images and the working of project

### Challenges faced and how were they overcome

* Integrating the contributions of all three members was a little difficult
* Git inspector does not show all the commits that are made

### Future Scope (If applicable)

* Add GUI interface to the application
* Add encryption to file containing user data
* Add show passbook feature by keeping track of transactions

### 