**Industrial Internship Report on**

**Banking information system**

**Prepared by**

**[G.P.Hymavathi]**

|  |
| --- |
| *Executive Summary* |
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).  This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.  My project was on Banking information system  This internship gave me a very good opportunity to get exposure to Service sector in Banking system and design/implement solution for that. It was an overall great experience to have this internship. |

**TABLE OF CONTENTS**

[1 Preface 3](#_Toc139702806)

[2 Introduction 4](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 4](#_Toc139702808)

[2.2 About upskill Campus 8](#_Toc139702809)

[2.3 Objective 9](#_Toc139702810)

[2.4 Reference 9](#_Toc139702811)

[2.5 Glossary 10](#_Toc139702812)

[3 Problem Statement 11](#_Toc139702813)

[4 Existing and Proposed solution 12](#_Toc139702814)

4.1 Code Submission……………………………………………………………………………………………………………………………..13

4.2 Report submission (Github link) ………………………………………………………………………………………………………13

[5 Proposed Design/ Model 1](#_Toc139702815)4

[5.1 High Level Diagram (if applicable) 1](#_Toc139702816)5

[5.2 Low Level Diagram (if applicable) 1](#_Toc139702817)5

[5.3 Interfaces (if applicable) 1](#_Toc139702818)6,17

[6 Performance Test 1](#_Toc139702819)8

[6.1 Test Plan/ Test Cases 1](#_Toc139702820)9

[6.2 Test Procedure](#_Toc139702821) 19,20

[6.3 Performance Outcome](#_Toc139702822) 20

[7 My learnings](#_Toc139702823) 21

[8 Future work scope](#_Toc139702824) 22

# Preface

The CORE JAVA internship duration was of 6 weeks. 1st week of the internship was to explore the problem statement which were provided by the management and understand their background in order to start with the project. Also learned about UCT. 2nd week of the internship was to understand and follow the project instructions provided by UCT. And, also to plan for the solution of the existing problem. 3rd week of internship is to start for the actual Internships are an opportunity to network with great people and sharpen your skills before entering the workforce. They also help tremendously with figuring out your true passion. Companies often look at them as a way to gain experience and exposure to make a smooth transition into your role when hired.

Develop a prototype of a Banking Information System in Core Java that provides a working preview of the key functionalities of a real banking system. The prototype should demonstrate the core features and flow of the system, showcasing its functionality and usability

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.working of the project. 4th week of the internship was so to continue with the work on the project and check whether there are improvements required for the project. 5th week of the internship was to validate your implementation and evaluate your performance. And the final week of the project is to submit your project report and get certification.

# Introduction

## About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various**Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end**etc.



1. UCT IoT Platform **(****)**

**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

* It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
* It supports both cloud and on-premises deployments.

It has features to  
• Build Your own dashboard  
• Analytics and Reporting  
• Alert and Notification  
• Integration with third party application(Power BI, SAP, ERP)  
• Rule Engine

1. **Smart Factory Platform (****)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

* with a scalable solution for their Production and asset monitoring
* OEE and predictive maintenance solution scaling up to digital twin for your assets.
* to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
* A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

1.  based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

1. Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



## About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.





## The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

## Objectives of this Internship program

The objective for this internship program was to

 ☛ get practical experience of working in the industry.

 ☛ to solve real world problems.

 ☛ to have improved job prospects.

 ☛ to have Improved understanding of our field and its applications.

 ☛ to have Personal growth like better communication and problem solving.

## Reference

[1] <https://learn.upskillcampus.com/s/courses/6441224de4b0f11fbe0f621e/take>

## Glossary

|  |  |
| --- | --- |
| Terms | Acronym |
| Accuracy | Banking Information System |
| Confusion Matrix | An NAN table that summarize the user registrartion,Deposit and withdrawl,fund transfer. |
| Regression Data | Regression feature are the continous data |
| Linear Regression | A supervised model which predict the Banking information system |
| Matplotlib | An open Source Python 2D plotting library helps you to visualize |

# Problem Statement

1. Problem Statement for the Project: Banking Information System

Develop a prototype of a Banking Information System in Core Java that provides a working preview of the key functionalities of a real banking system. The prototype should demonstrate the core features and flow of the system, showcasing its functionality and usability.

Key Functionality to Include in the Prototype:

User Registration: Implement a simplified user registration process where users can provide basic details to create an account.

Account Management: Develop the ability to create and manage user accounts, including assigning unique account numbers and tracking account balances.

Deposit and Withdrawal: Enable users to make deposits and withdrawals from their accounts, updating the account balance accordingly.

Fund Transfer: Implement a simplified version of fund transfer functionality, allowing users to transfer funds between their accounts or to other registered users.

Account Statements: Provide users with a preview of their account statements, displaying transaction history, dates, amounts, and remaining balances.

Password Protection: Develop a basic login system with password authentication to ensure secure access to user accounts.

Error Handling: Implement basic error handling mechanisms to handle common exceptions, such as insufficient funds and invalid transactions, and display relevant error messages to users.

User Interface: Design a user-friendly interface for the prototype that allows users to navigate through the system, perform banking operations, and view relevant information.

Persistence: Implement basic data persistence by storing user account information and transaction history temporarily during the prototype session.

By developing this prototype, stakeholders will have a tangible working preview of the key features and functionality of the Banking Information System. This will allow them to evaluate the system's usability, identify any necessary improvements or enhancements, and make informed decisions for further development and deployment of the complete system.

Minimum Requirements and System Output

1. User Registration:

Form Creation: Create a user registration form that prompts users to input their personal details, such as name, address, contact information, and initial deposit amount.

Output: Upon successful registration, the system will generate a unique account number for the user, and the user's details will be stored in the system's memory or File System.

The output will be a confirmation message indicating successful registration.

2. Account Management:

Form Creation: Develop an account management form that allows users to view and update their account information, such as name, address, contact details, and account settings.

Output: After making any updates or changes, the system will display a confirmation message indicating that the account information has been successfully updated.

3. Deposit and Withdrawal:

Form Creation: Design a form where users can enter the amount they wish to deposit or withdraw from their account.

Output: Upon successful deposit or withdrawal transaction completion, the system will update the account balance accordingly and display a confirmation message indicating the transaction details, such as the transaction amount and the resulting balance.

4. Fund Transfer:

Form Creation: Create a form that allows users to specify the recipient's account number and the amount they wish to transfer.

Output: After a successful fund transfer, the system will deduct the transferred amount from the sender's account, add it to the recipient's account, and display a confirmation message indicating the transaction details, including the transferred amount and the updated balances of both accounts.

5. Account Statements:

Output: Implement a feature that allows users to view their account statements, which will be displayed as a comprehensive list showing transaction history, including dates, transaction amounts, and resulting balances. Users can access their account statements through a designated section of the system's user interface.

## Code submission (Github link)

## Report submission (Github link) : first make placeholder, copy the link.

Project Definition:

Understand the project requirements and objectives. In this case, it's to create a prototype of a Banking Information System.

Identify the core features that need to be demonstrated, such as account creation, balance inquiry, fund transfer, and transaction history.

Setup Development Environment:

Install a Java development environment, like Eclipse or IntelliJ IDEA.

Ensure you have the necessary Java Development Kit (JDK) installed.

Design Database Schema:

Design the database structure to store customer information, accounts, transactions, etc.

Choose a database system, like MySQL or SQLite, and create the necessary tables.

Create Java Classes:

Create Java classes to represent entities like Customers, Accounts, and Transactions.

Implement methods for each class to perform actions like creating an account, transferring funds, and fetching transaction history.

User Interface (UI) Design:

Decide on the user interface for your prototype. It could be a simple command-line interface or a basic graphical user interface (GUI).

Create UI elements for account creation, balance inquiry, fund transfer, and transaction history.

Implement Core Functionality:

Implement the core functionality of the banking system. For example:

Allow users to create accounts with unique account numbers.

Implement methods to deposit and withdraw funds from accounts.

Enable fund transfers between accounts.

Record transactions and maintain transaction history.

Testing and Debugging:

Test your prototype thoroughly to ensure all functionalities work as expected.

Debug and fix any issues or errors that you encounter during testing.

Documentation:

Create documentation that explains how to use your prototype, including user instructions and developer documentation for future enhancements.

Demo and Presentation:

Prepare a demonstration of your prototype, showcasing its key functionalities and usability.

Create a presentation to explain the project, its objectives, and the technologies used.

Feedback and Iteration:

Collect feedback from potential users or stakeholders and make necessary improvements to your prototype based on their input.

Finalization:

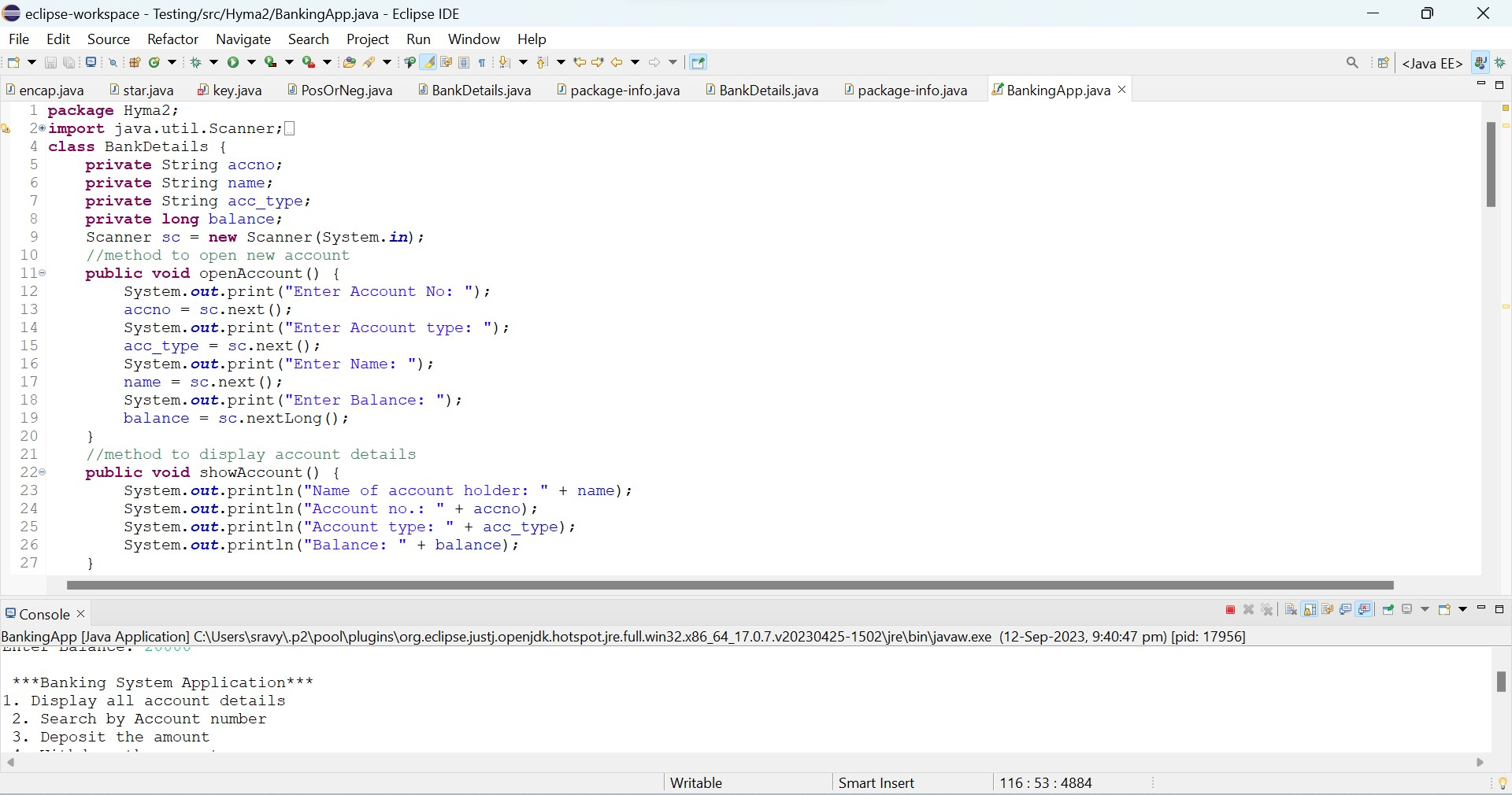
Once the prototype meets the project requirements and receives positive feedback, you can consider it complete.

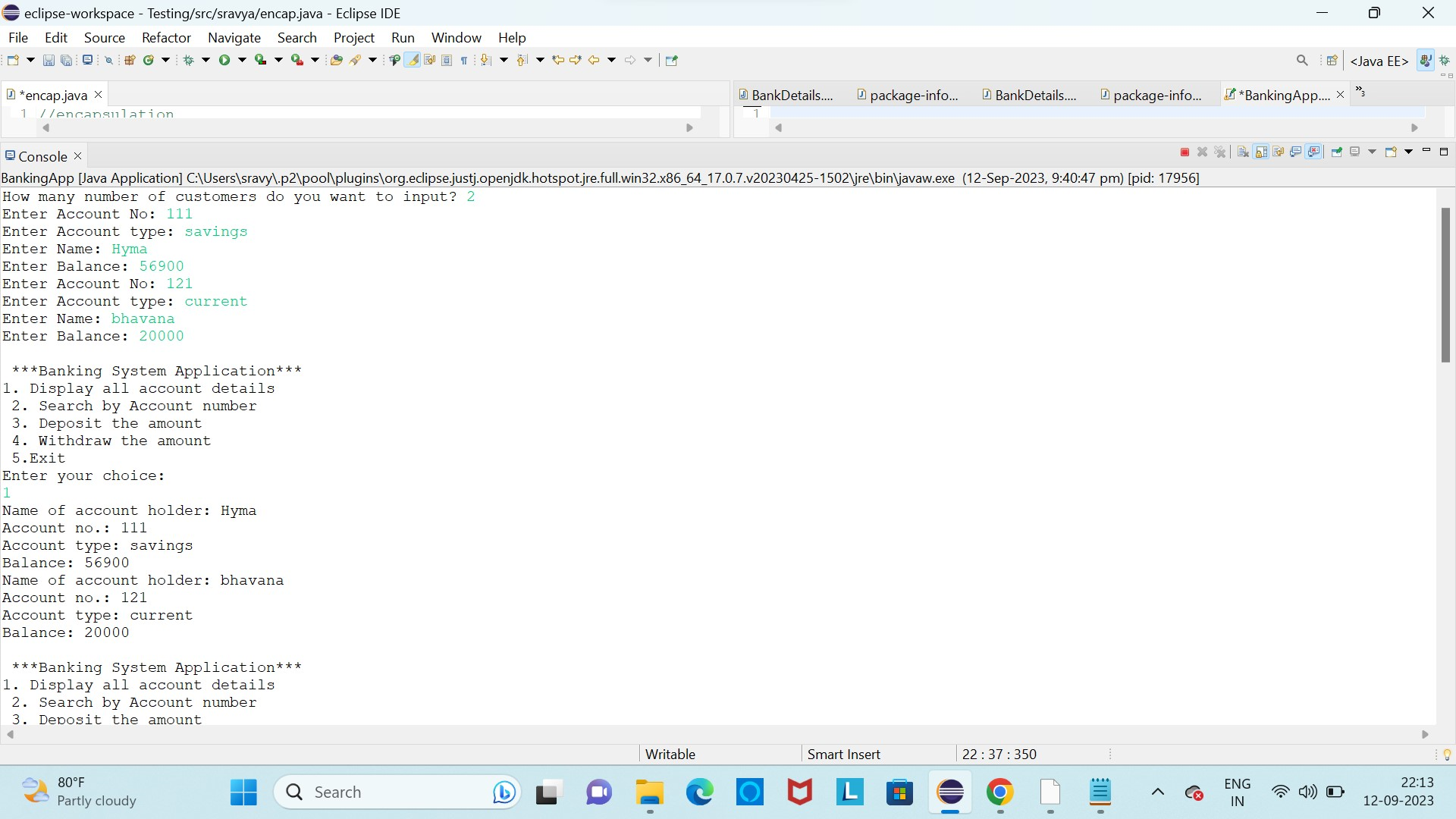
Future Development:

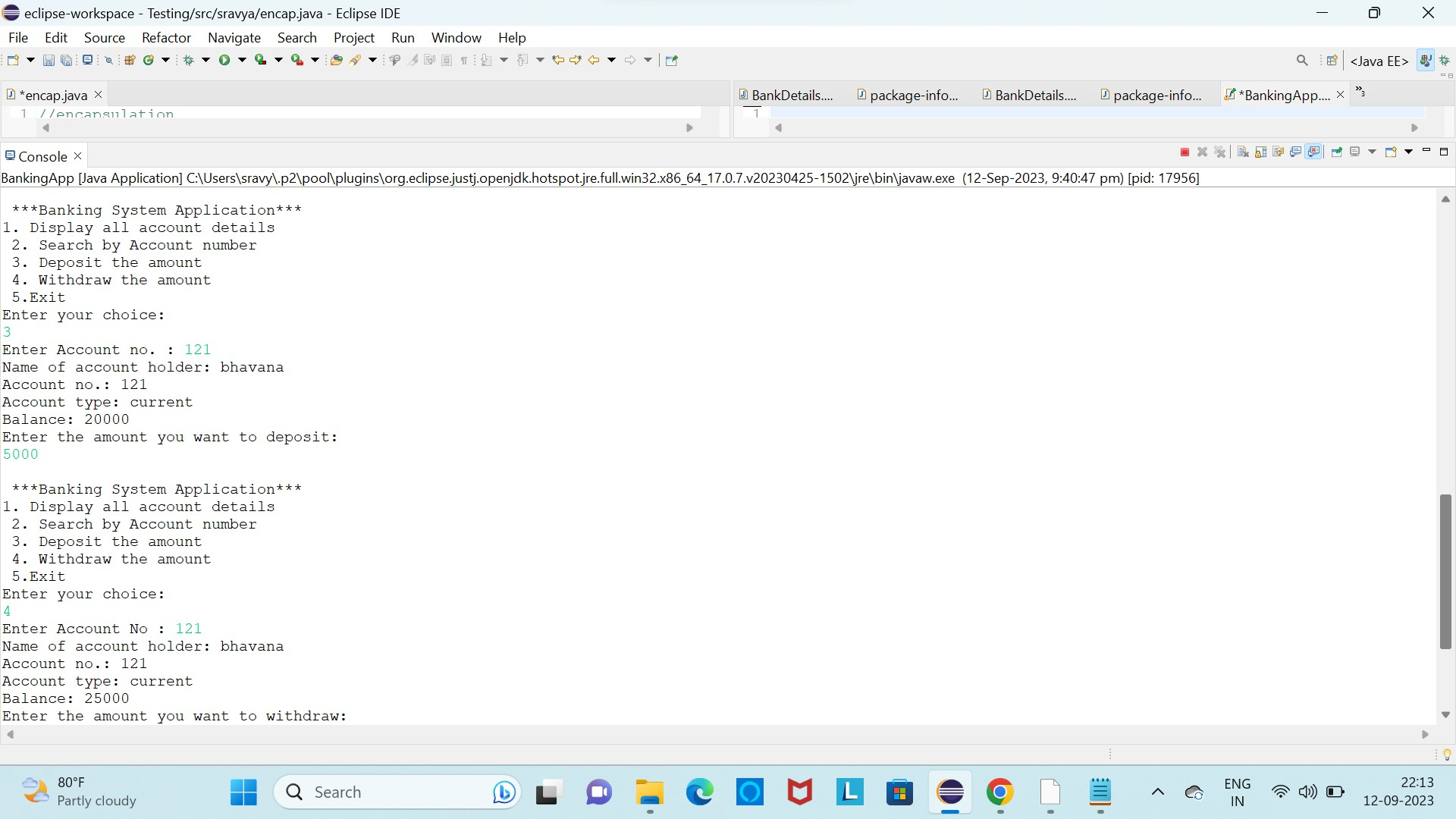
If this prototype is intended to evolve into a full-fledged banking system, plan for future development phases, including security enhancements, scalability, and additional features.

Remember to follow best practices in Java programming and design patterns throughout your project. This step-by-step approach should help you create a functional prototype of a Banking Information System in Core Java

## Interfaces (if applicable)

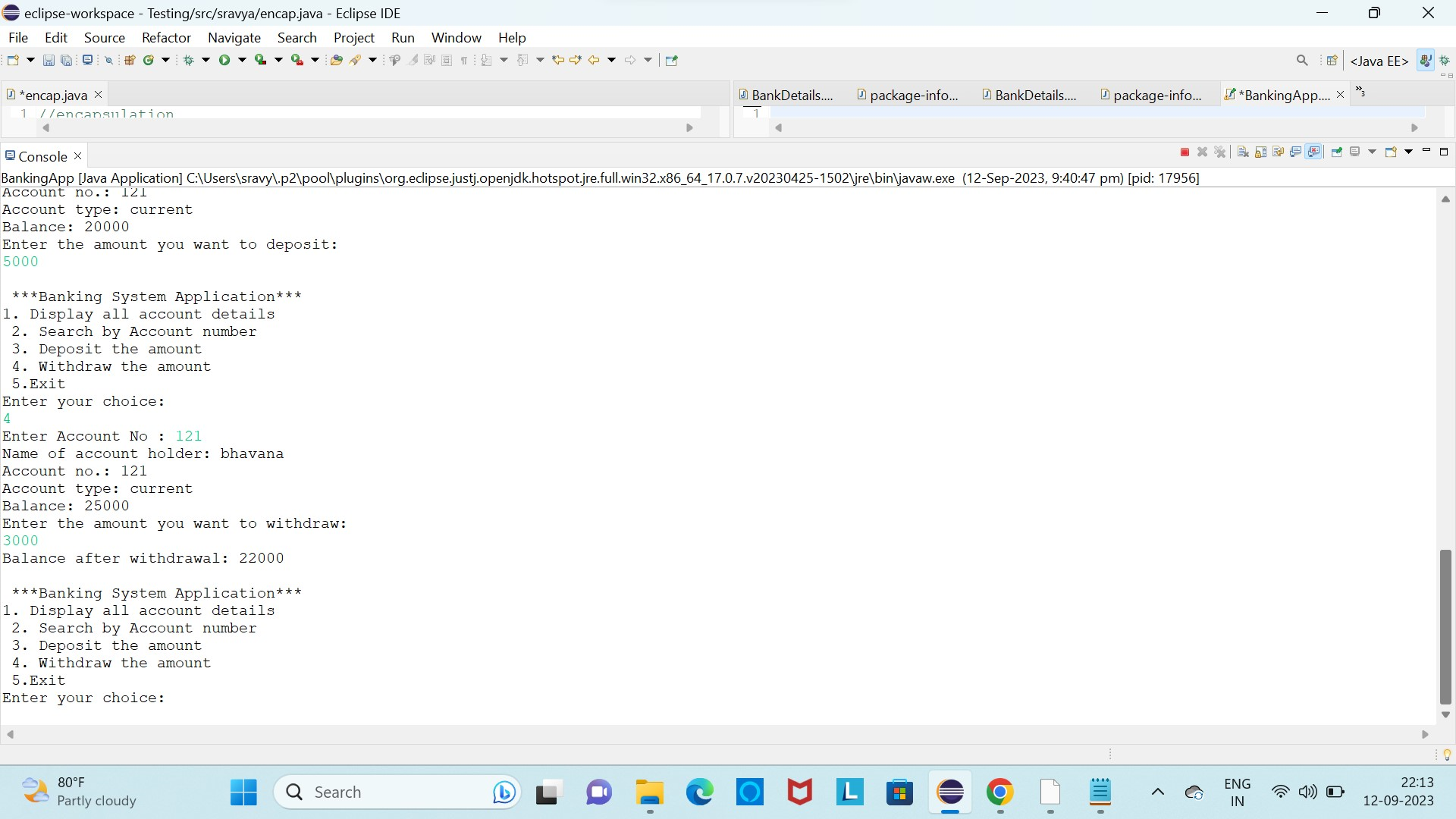






**To find Correlation**

# Performance Test



## Test Plan/ Test Cases

## Test Procedure

Functionality Testing:

Verify that the system allows users to create new accounts (savings, checking, etc.).

Test if users can deposit funds into their accounts and that the balance updates correctly.

Ensure that users can withdraw funds and that the balance updates accurately, considering overdraft limits (if applicable).

Test transferring funds between accounts, both within the same user's profile and to other users.

Check if the system can display account balances and transaction history.

User Authentication Testing:

Test the login process for customers and bank employees (if applicable).

Verify that only authorized users can access specific functionalities (e.g., only employees can close accounts).

Error Handling:

Test for error messages and ensure they are displayed when necessary (e.g., insufficient funds, incorrect login credentials).

Ensure the system gracefully handles unexpected errors or exceptions.

Security Testing:

Verify that sensitive user information, such as passwords and account details, are securely stored and transmitted.

Test for vulnerabilities like SQL injection or cross-site scripting (XSS).

Performance Testing:

Assess the system's response time when performing various transactions.

Test its scalability by simulating a high volume of concurrent users.

Usability Testing:

Have users who are not familiar with the system navigate through it to assess its ease of use.

Collect feedback on the user interface and any suggested improvements.

Data Integrity and Persistence Testing:

Ensure that data is consistently stored and retrieved accurately from a database or file system.

Test data backup and recovery procedures to prevent data loss.

Compatibility Testing:

Verify that the system works on different operating systems and browsers (if it has a web interface).

Regulatory Compliance Testing:

If applicable, ensure the system complies with banking regulations and standards in your jurisdiction.

Documentation Review:

Review the project documentation to ensure it accurately reflects the system's functionality and usage.

Load and Stress Testing:

Simulate heavy loads and stress on the system to determine its breaking point and measure its stability under extreme conditions.

Integration Testing:

If your system interacts with external services (e.g., payment gateways), test the integration to ensure smooth data exchange.

User Acceptance Testing (UAT):

Involve actual users or stakeholders to perform final testing and gather their feedback to make necessary adjustments.

Accessibility Testing:

Ensure the system is accessible to users with disabilities, following relevant accessibility standards.

Regression Testing:

After each change or update to the prototype, perform regression testing to ensure that existing functionalities still work as expected.

Scalability Testing:

Test the system's ability to handle a growing user base and increased data volume.

Compliance Testing:

Check if the system complies with any industry-specific standards or regulations.

Performance Optimization:

Identify and address any performance bottlenecks or issues found during testing.

Final Review and Sign-Off:

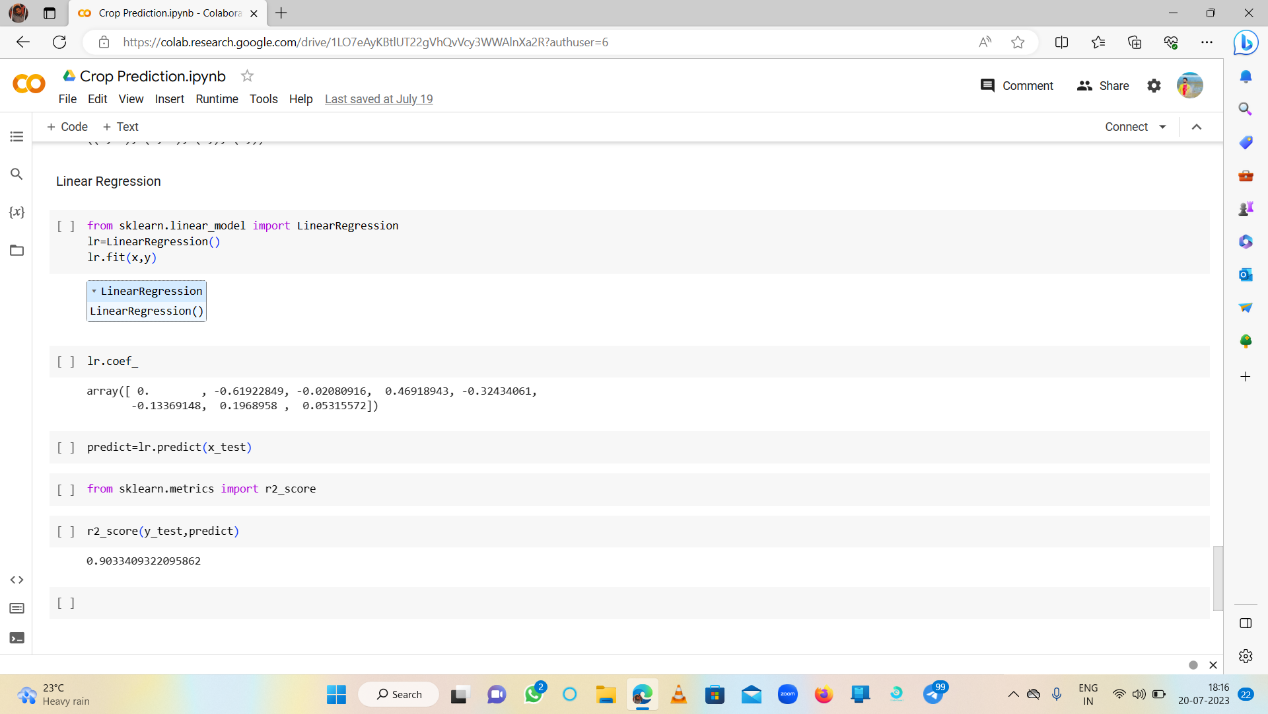
Conduct a final review to ensure all issues have been addressed, and obtain approval from stakeholders before deployment.

Documentation Update:

Update the documentation to reflect any changes or improvements made based on testing results.

Remember to document all test cases, results, and any issues encountered during testing. This comprehensive testing process will help ensure that your Banking Information System prototype functions reliably and meets the specified requirements.

## Performance Outcome



# My learnings

Studied the basics of core java and its applications. Studied about java applications in depth. Glanced through the instructions of the internship program and understood the process. Went through the profile of UCT. Selected the project for the core java internship.

1. Problem Statement for the Project: Banking Information System

Develop a prototype of a Banking Information System in Core Java that provides a working preview of the key functionalities of a real banking system. The prototype should demonstrate the core features and flow of the system, showcasing its functionality and usability.

Key Functionality to Include in the Prototype:

User Registration: Implement a simplified user registration process where users can provide basic details to create an account.

# Future work scope

Core Java is the fundamental and foundational part of the Java programming language. It encompasses a wide range of features and has a broad scope.