

Industrial Internship Report on Banking Information System

Prepared by
Manas Singh

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 Banking Information System

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.

TABLE OF CONTENTS

1	Preface	3
2	Introduction	5
2.1	About UniConverge Technologies Pvt Ltd	5
2.2	About upskill Campus	9
2.3	Objective	11
2.4	Reference	11
2.5	Glossary	11
3	Problem Statement	12
3.1	User Registration	12
3.2	Account management	12
3.3	Deposit and Withdrawal	12
3.4	Fund Transfer	12
3.5	Account Statements	13
3.6	Password Protection	13
3.7	Error Handling	13
3.8	User Interface	13
4	Existing and Proposed solution	14
5	Proposed Design/ Model	16
5.1	User-Centric Design	16
5.2	Core Functional Modules	16
5.3	Backend Logic	16
5.3	Backend Logic	16
6	Performance Test	18
6.1	Test Plan/ Test Cases	18
6.2	Test Procedure	18
6.3	Performance Outcome	18
7	My learnings	19
8	Future work scope	20

1 Preface

Summary of the Six Weeks' Internship Experience

Over the last six weeks, I have had the incredible opportunity to immerse myself in a practical work environment that not only tested my theoretical knowledge but also allowed me to expand my skill set significantly. Understanding the importance of relevant internships in career development has been a cornerstone of this journey. These experiences bridge the gap between academic knowledge and real-world application, providing a platform to apply classroom theories to solve practical problems, enhance professional skills, and understand workplace dynamics.

Project Overview

My project involved developing a Banking Information System using Object-Oriented Programming (OOP) principles and core Java. The goal was to create a robust system that could manage customer and bank data efficiently, offering functionalities like account creation, transaction management, and customer support services. This project was not only about coding but also about applying software engineering principles to design a system that is scalable, maintainable, and user-friendly.

Opportunity Provided by USC/UCT

The opportunity to work on this project was given by USC/UCT, highlighting their commitment to providing students with practical experiences that complement their academic pursuits. The program was meticulously planned, balancing project work with mentorship sessions, workshops on new technologies, and soft skill development, ensuring a holistic growth environment.

Learnings and Overall Experience

This internship has been a significant learning curve. It honed my technical skills in Java and deepened my understanding of OOP concepts. Moreover, I learned the importance of code readability, proper documentation, and the agile methodology in project management. Collaborating with peers, handling deadlines, and receiving constructive feedback were valuable aspects of my professional development. The experience of

transforming a concept into a functional software product was immensely satisfying and has ignited a passion for technology-driven solutions.

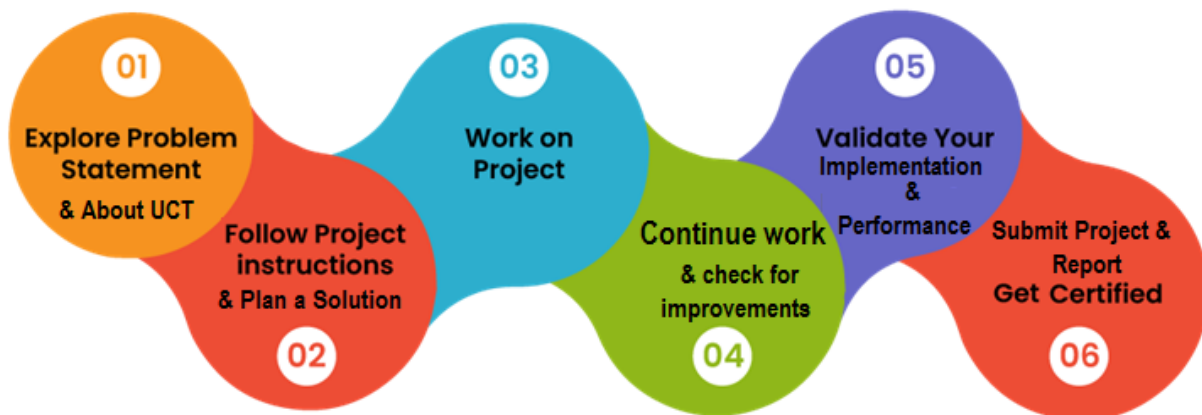
Acknowledgments

I extend my heartfelt thanks to Nikhil, my friend, whose support and insights were invaluable throughout this journey. His willingness to share knowledge and offer encouragement was a key factor in my project's success. I am also immensely grateful to Monu Kumar, my mentor, whose expertise and guidance helped me navigate challenges and learn from them. His mentorship has been a beacon of light, guiding me towards achieving project objectives and personal growth.

Message to Juniors and Peers

To my juniors and peers, I urge you to embrace every opportunity for an internship with open arms. It is a unique learning experience that goes beyond textbooks, allowing you to apply your knowledge, explore your interests, and understand what it takes to succeed in your chosen field. Take initiative, seek feedback, and cherish every moment of this journey. Remember, it's not just about completing tasks but about growing as a professional and as an individual.

In closing, this internship has been a transformative phase in my academic and professional journey, setting a solid foundation for my future endeavors. I look forward to carrying forward the lessons learned and contributing to the tech world with enthusiasm and dedication.



2 Introduction

2.1 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.



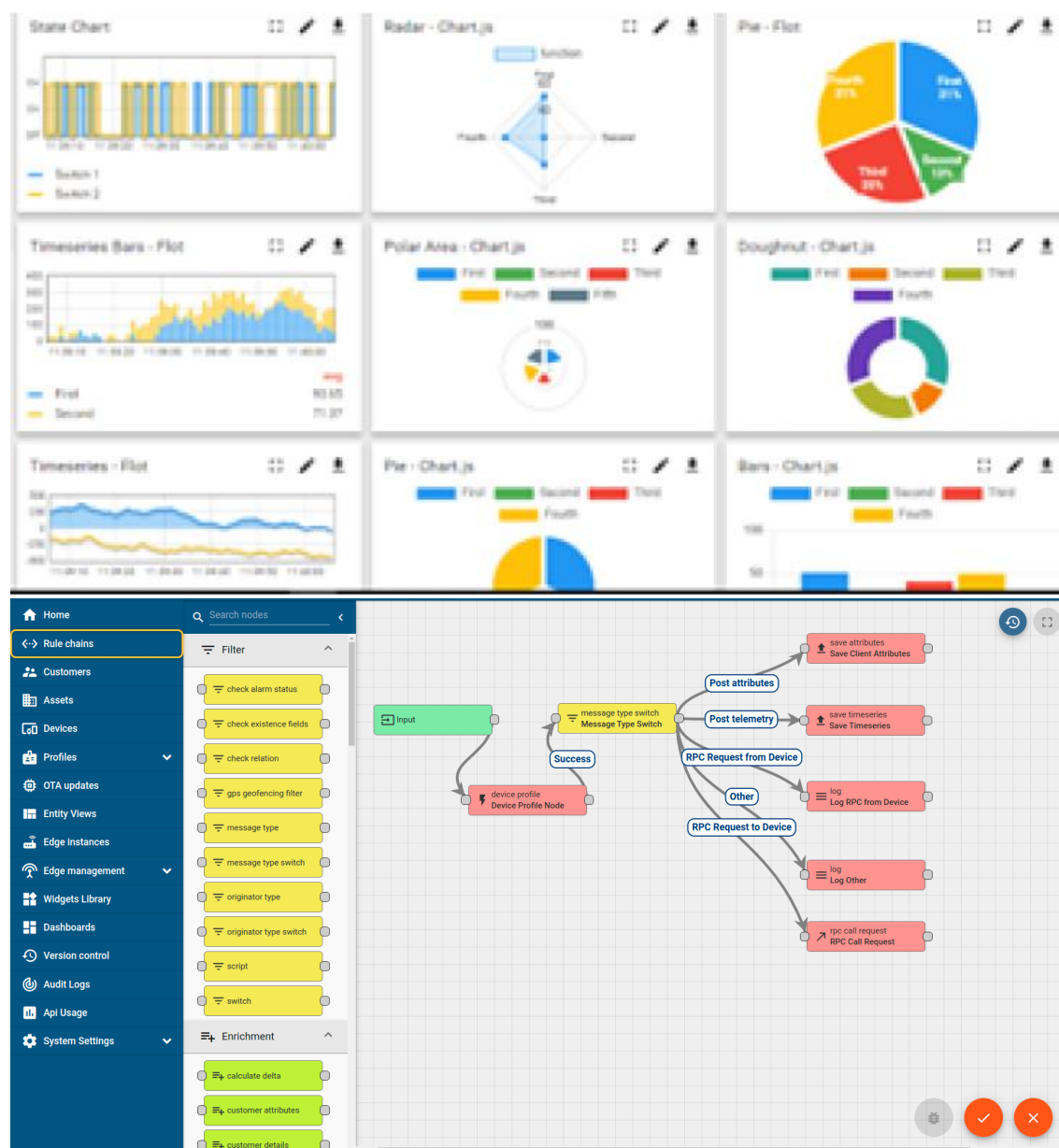
i. UCT IoT Platform ()

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



FACTORY WATCH

ii. 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 unleash 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 want 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.



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



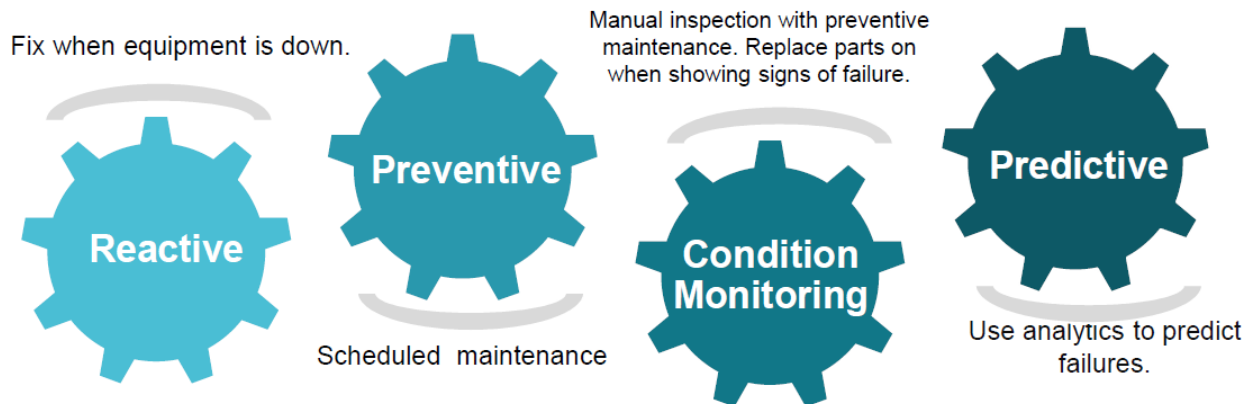


iii. based Solution

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

iv. Predictive Maintenance

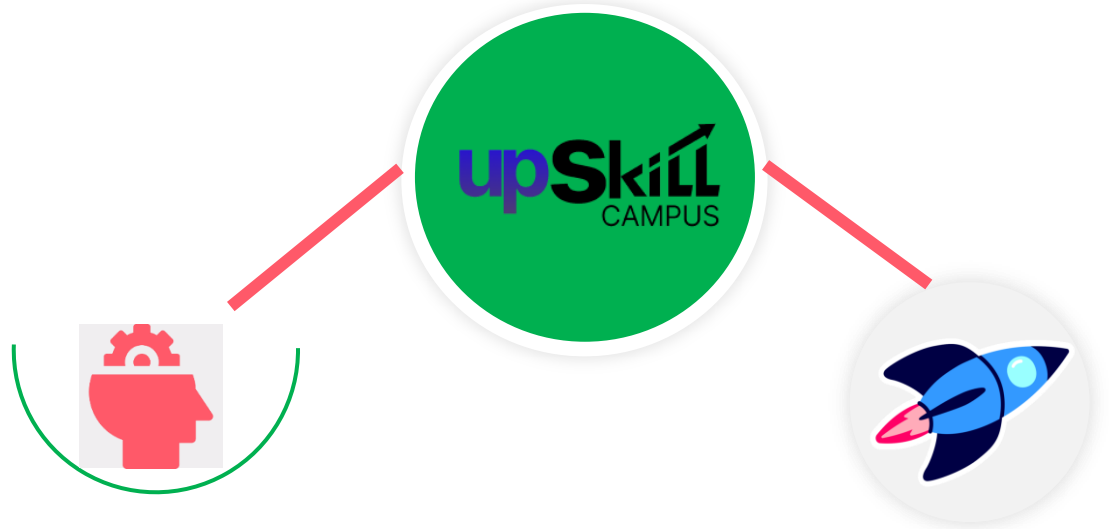
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 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.

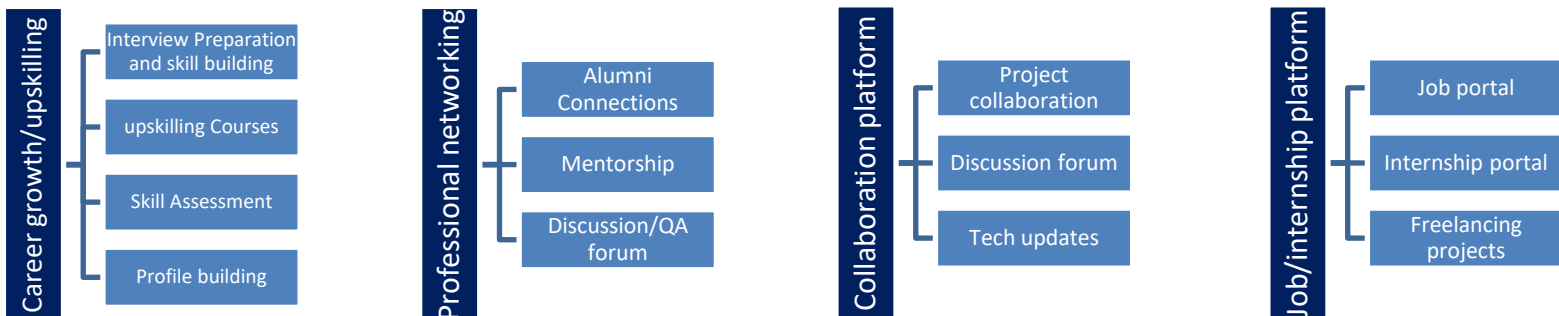
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

<https://www.upskillcampus.com/>



2.3 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.

2.4 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.

2.5 Reference

[1] Oracle. (n.d.). *The Java™ Tutorials*. Retrieved Month Day, Year, from <https://docs.oracle.com/javase/tutorial/>

[2] Stack Overflow. (n.d.). Retrieved Month Day, Year, from <https://stackoverflow.com> (use specific question links if applicable)

2.6 Glossary

Terms	Acronym
Object-Oriented Programming	OOP
Java Development Kit	JDK
Java Runtime Environment	JRE
Java Virtual Machine	JVM
Integrated Development Environment	IDE

3 Problem Statement-

In the assigned problem statement

This Banking Information System prototype is a foundational step towards creating a comprehensive digital banking solution, crafted using Core Java to simulate essential banking operations. The prototype is designed to showcase the system's primary capabilities, focusing on user registration, account management, financial transactions, and data security. Here's an expanded overview of its key components and functionalities:

3.1 User Registration:

The prototype begins with a straightforward user registration process. Prospective users are required to fill out a form detailing personal information such as name, address, contact details, and an initial deposit amount. Upon form submission, the system generates a unique account number for each user, symbolizing successful account creation. This process is crucial for establishing a secure and personalized banking experience from the outset.

3.2 Account Management:

Once registered, users have the flexibility to manage and update their account information. This feature supports modifications to personal details, ensuring that users can keep their information current and accurate. The system confirms any changes made, reinforcing the sense of control users have over their banking details.

3.3 Deposit and Withdrawal:

A core functionality of the prototype, this feature allows users to conduct deposit and withdrawal transactions. Users input the desired transaction amount, and the system dynamically updates the account balance, providing immediate feedback through a confirmation message. This feature is essential for maintaining an accurate and up-to-date account balance, reflecting the real-time nature of banking transactions.

3.4 Fund Transfer:

The fund transfer capability enables users to move funds between accounts within the system. By entering the recipient's account number and the amount to be transferred, users can execute transfers seamlessly. The system updates both the sender's and

recipient's account balances accordingly and confirms the transaction, showcasing the interconnectedness of user accounts within the banking system.

3.5 Account Statements:

This feature offers users a detailed view of their transaction history, presenting a comprehensive account statement. It lists transaction dates, amounts, and the resulting balances, allowing users to track their financial activities over time. This functionality is vital for transparency and financial planning, giving users insight into their spending and saving patterns.

3.6 Password Protection:

Security is a paramount concern in digital banking, and the prototype addresses this through a basic yet effective login system. Password authentication ensures that only authorized users can access their accounts, providing a layer of security against unauthorized access.

3.7 Error Handling:

The system is equipped with error handling mechanisms to address common banking errors, such as insufficient funds or incorrect transaction inputs. By providing clear error messages, the system helps users navigate these issues, enhancing the user experience.

3.8 User Interface:

The prototype features a user-friendly interface designed to simplify navigation and make banking operations accessible to all users. The intuitive layout and clear instructions guide users through various banking processes, from registration to executing transactions

4 Existing and Proposed solution

Existing banking systems, whether they are traditional or modern digital platforms, offer a wide range of functionalities including account management, financial transactions, and user authentication. These systems prioritize security, user-friendliness, and comprehensive financial services. However, several common limitations are observed across various solutions:

1. **Complexity:** Many banking systems come with complex interfaces that may overwhelm new users or those not tech-savvy.
2. **Scalability:** Some systems struggle to efficiently manage a large user base or a high volume of transactions without performance degradation.
3. **Customization:** Limited personalization options for users, making it hard to tailor the banking experience to individual preferences.
4. **Integration:** Challenges in seamlessly integrating with other financial systems or services for a unified banking experience.
5. **Accessibility:** While digital banking has increased accessibility, there are still gaps in serving users in remote or underserved regions.

What value addition are you planning?

4.1 Proposed Solution: Banking Information System Prototype

The prototype developed in Core Java addresses these limitations through a simplified, yet comprehensive approach to banking system design. By focusing on core functionalities such as user registration, account management, and transaction processing, the system offers a streamlined user experience. Here's how the proposed solution stands out:

1. **Simplicity:** With an intuitive interface and straightforward process flows, the system reduces complexity, making it accessible to all users.
2. **Modularity:** Designed with modularity in mind, it allows for easy scaling and incorporation of additional features as needed.
3. **Customization:** The prototype lays the groundwork for future enhancements, including user customization options to personalize the banking experience.
4. **Integration Ready:** The system's architecture is designed to facilitate easy integration with external services and systems in future developments.
5. **Focused on Core Needs:** By concentrating on essential banking operations, it ensures reliability and efficiency in user transactions and account management.

4.2 Code submission (Github link)

GitHub Repo - <https://github.com/ManasSingh77/upskillCampus>

GitHub Code Link- <https://github.com/ManasSingh77/upskillCampus/blob/main/BankingSystem.java>

4.3 Report submission (Github link) :

GitHub Repo - <https://github.com/ManasSingh77/upskillCampus>

5 Proposed Design/ Model

The proposed design model for the Banking Information System developed in Core Java focuses on simplicity, modularity, and scalability to cater to essential banking needs while ensuring user-friendly experiences. Here's a brief overview of the design model:

5.1 User-Centric Design

- **User Interface (UI):** A straightforward and intuitive UI that guides users through registration, account management, and transaction processes without overwhelming them with complexities.
- **User Experience (UX):** Emphasis on a seamless and efficient experience, ensuring that common banking operations are completed with minimal steps and clear feedback is provided for all actions.

5.2 Core Functional Modules

1. **User Registration Module:** Facilitates easy sign-up for new users, requiring basic information and initial deposit amount, and assigns a unique account number.
2. **Account Management Module:** Allows users to view and update their personal and account information, ensuring users have control over their banking details.
3. **Transaction Processing Module:** Handles deposits, withdrawals, and transfers with real-time updates to account balances, incorporating basic error handling for common issues like insufficient funds.
4. **Security Module:** Implements basic password protection for user login, laying the groundwork for more advanced security features such as encryption and multi-factor authentication in future iterations.

5.3 Backend Logic

- **Data Storage:** Utilizes a simple in-memory data structure (HashMap) for storing user information and transaction history, demonstrating the system's capability for data persistence and retrieval.
- **Error Handling:** Includes basic mechanisms to manage errors and provide informative feedback, ensuring a robust user experience even in edge cases.
- **Transaction Validation:** Ensures all transactions are validated for criteria such as sufficient balance before processing, minimizing errors and maintaining account integrity.

5.4 Scalability and Future Enhancements

- The design is modular, allowing for easy addition of new features such as loan management, interest calculation, and integration with external financial services.

- Plans for incorporating advanced technologies like AI for personalized services and improved security measures to protect user data and transactions.

This design model serves as a foundation for a Banking Information System that is not only functional and user-friendly but also scalable and adaptable to future banking innovations and user needs.

6 Performance Test

6.1 Test Plan/ Test Cases

- **User Registration Performance:** Test the system's ability to handle multiple simultaneous user registrations without significant delays or errors.
- **Account Management Response Time:** Evaluate how quickly the system updates and reflects changes in user account details.
- **Transaction Efficiency:** Measure the time taken for deposit, withdrawal, and fund transfer operations to complete and the system's capability to handle concurrent transactions.
- **System Stability under Load:** Simulate a high number of users performing various operations simultaneously to assess the system's stability and response time.

6.2 Test Procedure

1. **Setup:** Prepare the environment with necessary data and tools for monitoring system performance.
2. **Execution:** Run the defined test cases, simulating various operations such as user registrations, deposits, withdrawals, and fund transfers.
3. **Monitoring:** Use tools to monitor system metrics like response time, error rates, and resource utilization.
4. **Analysis:** Analyze the results to identify any bottlenecks or performance issues.

6.3 Performance Outcome

The outcome should detail the system's performance, including how well it managed high-load scenarios, its stability, and any limitations observed. It will highlight areas for optimization, ensuring the system can handle real-world usage efficiently.

7 My learnings

Throughout this project, I gained valuable experience in applying Object-Oriented Programming (OOP) principles using Java to solve real-world problems. I learned about the importance of designing a system with both user experience and performance in mind. Implementing features like user registration, account management, and transactions deepened my understanding of data structures and algorithms, specifically in handling and securing sensitive user data.

This project also improved my problem-solving skills, as I navigated challenges related to system design, data persistence, and error handling. Working on performance testing taught me about the critical aspects of software quality assurance and the importance of building scalable systems.

These skills are crucial for my career growth, as they are applicable across a wide range of software development roles. The experience of designing and implementing a banking information system from scratch has prepared me for tackling complex projects in the future, making me a more competent and versatile software developer.

8 Future work scope

The prototype of the Banking Information System developed presents a solid foundation for a comprehensive banking solution. However, due to time constraints, several enhancements and features were not implemented but can be considered for future work to elevate the system's capabilities and user experience.

8.1 Advanced Security Measures

- **Two-Factor Authentication (2FA):** Integrate two-factor authentication to add an extra layer of security for user logins and critical operations like fund transfers.
- **Encryption of Sensitive Data:** Implement encryption for sensitive data storage and transmission, ensuring user information and transaction details are protected against unauthorized access.

8.2 Enhanced User Experience

- **Mobile App Version:** Develop a mobile application version of the system to provide users with the convenience of banking services on their smartphones.
- **User-Friendly Dashboard:** Create a more interactive and intuitive dashboard for users, offering a quick overview of their account status, recent transactions, and quick access to common operations.

8.3 Additional Banking Features

- **Loan Management System:** Integrate a loan management feature where users can apply for loans, view loan offers, and manage loan repayments.
- **Investment Services:** Offer investment services, allowing users to explore different investment options, monitor their investments, and receive personalized investment advice.

8.4 Scalability and Performance Enhancements

- **Cloud Deployment:** Migrate the system to a cloud platform to improve scalability and reliability, ensuring the system can handle a growing number of users and transactions efficiently.
- **Use of Microservices Architecture:** Refactor the system into a microservices architecture to enhance modularity, facilitate easier updates and maintenance, and improve system performance.

8.5 Integration with External Services

- **Payment Gateway Integration:** Integrate with payment gateways to enable users to make payments for various services directly from their accounts.

- **Real-time Notifications:** Implement real-time notifications for transactions, account alerts, and personalized offers, enhancing user engagement and satisfaction.

8.6 Artificial Intelligence and Machine Learning

- **Fraud Detection:** Utilize machine learning algorithms to analyze transaction patterns and detect potential fraudulent activities, enhancing the security of the banking system.
- **Personalized Banking Experience:** Implement AI-driven insights to offer personalized banking advice, financial tips, and product recommendations based on users' spending habits and financial goals.

These future enhancements not only aim to improve the functionality and security of the Banking Information System but also seek to provide a more personalized and convenient banking experience for users.