

## **Industrial Internship Report on "Banking Information System"**

**Prepared by  
Obaid Ullah Mansoori**

### *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 (Tell about ur Project)

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 .....	4
2.1	About UniConverge Technologies Pvt Ltd .....	4
i.	UCT IoT Platform .....	4
2.2	About upskill Campus (USC).....	8
2.3	The IoT Academy.....	10
2.4	Objectives of this Internship program .....	10
2.5	Reference.....	10
2.6	Glossary.....	10
3	Problem Statement.....	12
4	Existing and Proposed solution.....	14
5	Proposed Design/ Model .....	18
6	Performance Test.....	21
7	My learnings.....	24
8	Future work scope .....	26

## 1 Preface

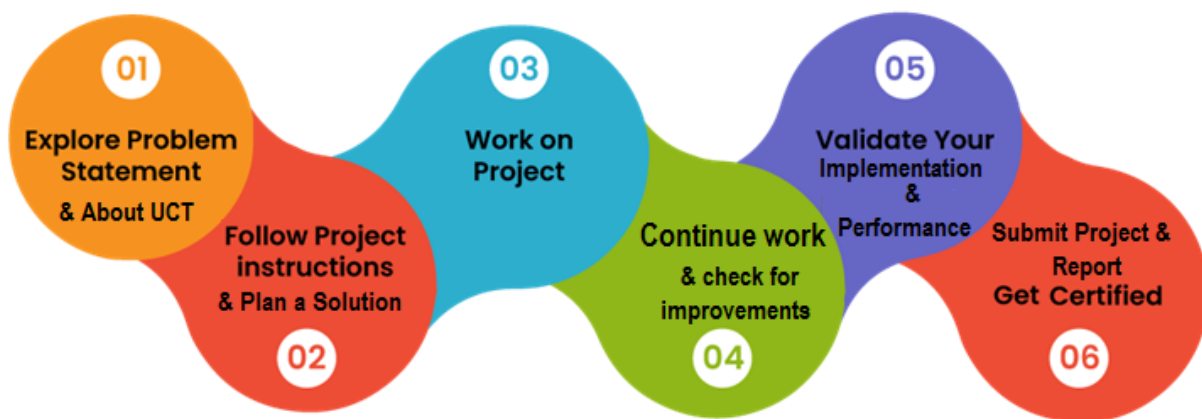
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

## 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.



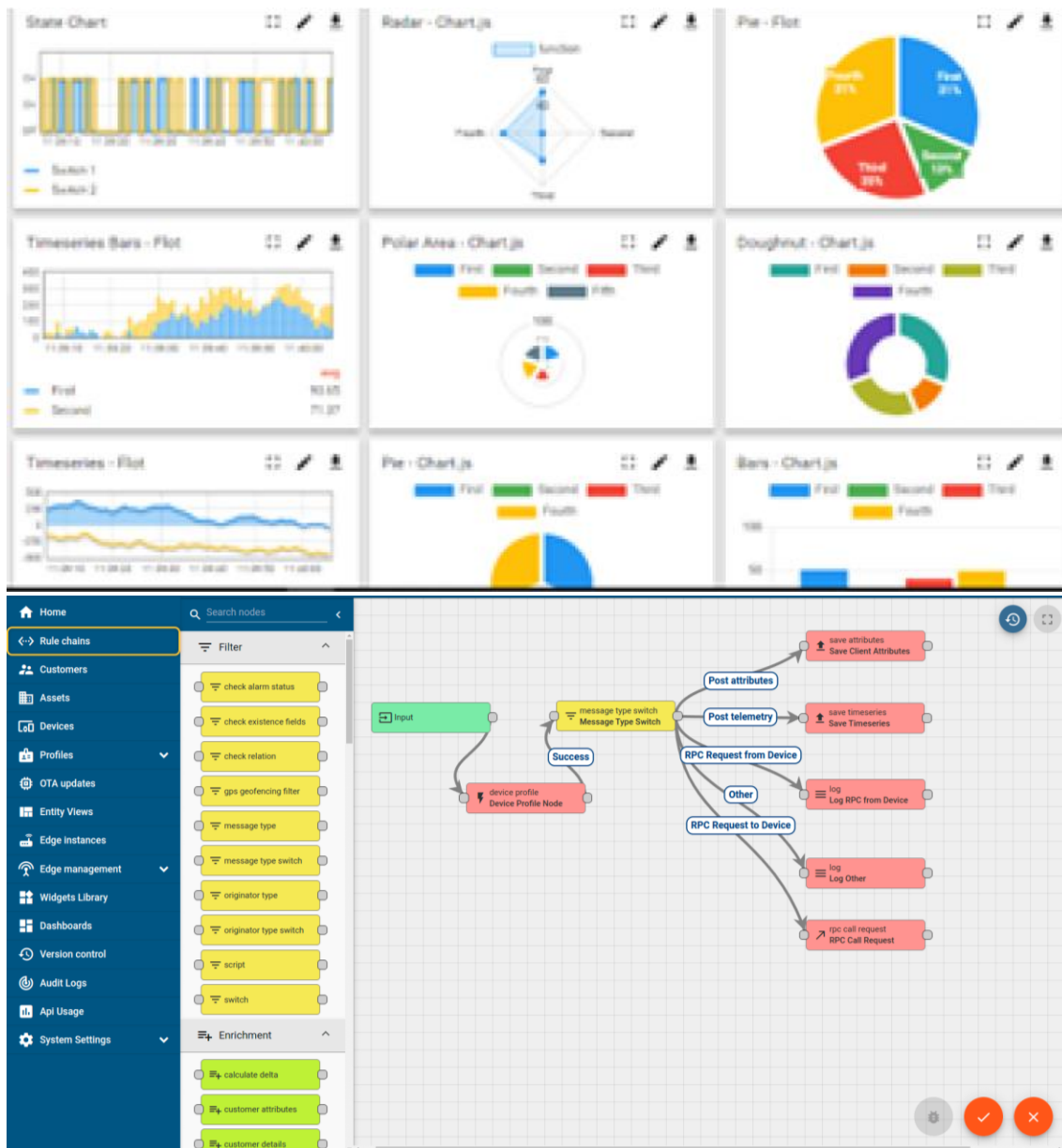
#### i. UCT IoT Platform ()

**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



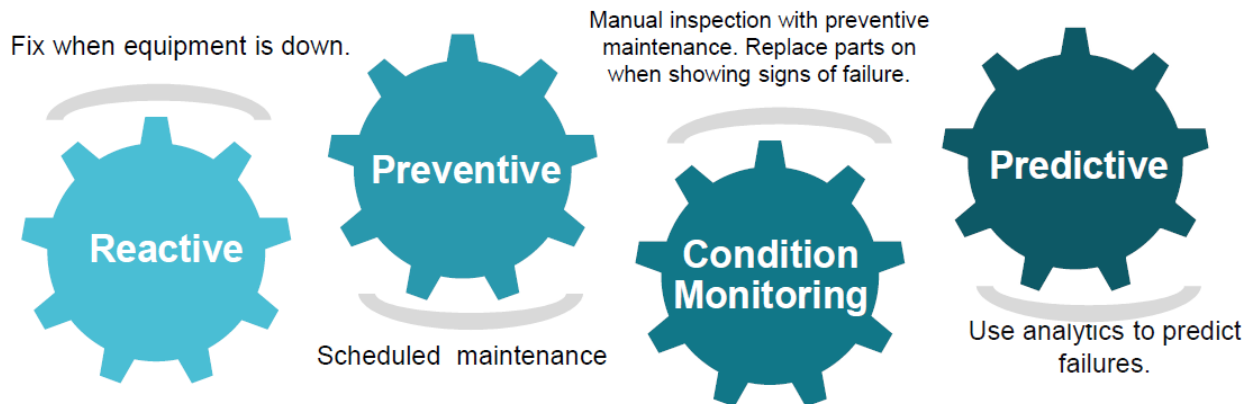


### iii. LoRaWAN based Solution

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.

### 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.

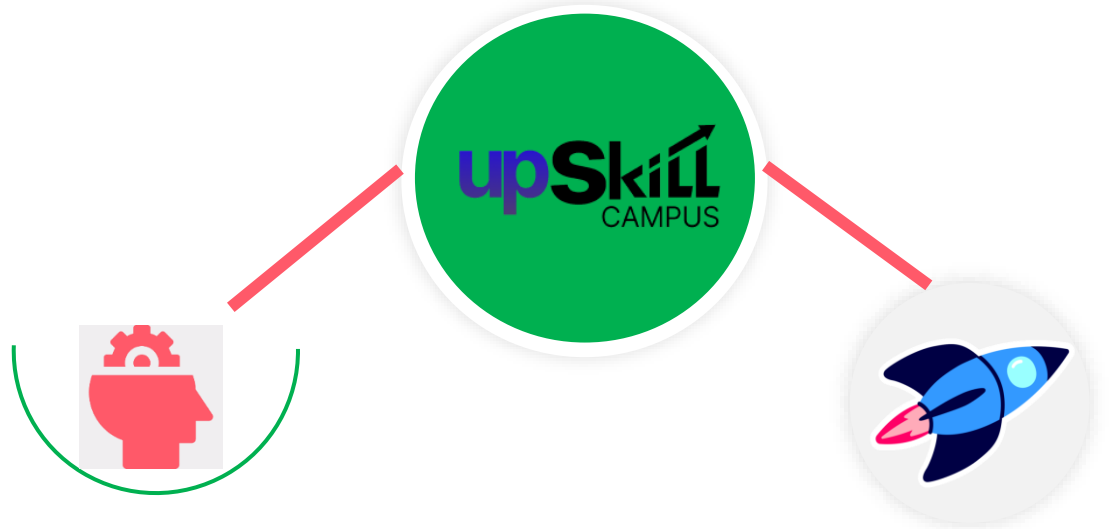


## 2.2 About upskill Campus (USC)

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.

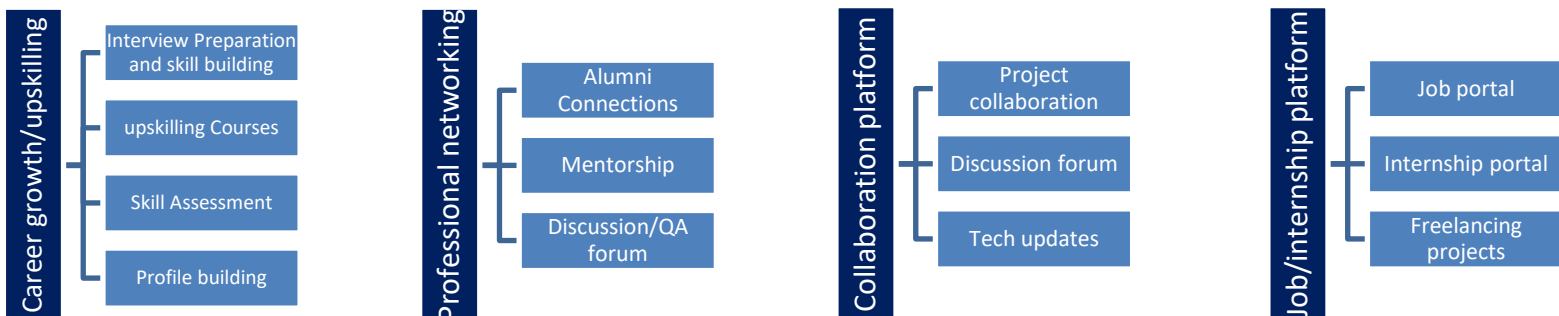




Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

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] Java SE Documentation: <https://docs.oracle.com/en/java/javase/index.html>
- [2] Oracle's Java Tutorials: <https://docs.oracle.com/javase/tutorial/>

## 2.6 Glossary

Terms	Acronym

<b>User Interface (UI)</b>	The graphical or textual representation of the system that allows users to interact with and control the software.
<b>Account Manager</b>	A component of the system responsible for managing user accounts, including user registration, authentication, and account-related operations.
<b>Transaction Manager</b>	A component responsible for handling financial transactions, such as deposits, withdrawals, and fund transfers.
<b>Authentication</b>	The process of verifying the identity of a user, typically through a username and password, to grant access to the system.
<b>Account Statement</b>	A report that provides a summary of a user's account activity, including transaction history, dates, and balances.
<b>Deposit</b>	The action of adding funds to a user's account.
<b>Withdrawal</b>	The action of deducting funds from a user's account.
<b>Fund Transfer</b>	The process of moving funds from one user's account to another user's account.
<b>Error Handling</b>	The mechanism for identifying and managing errors or exceptions that may occur during system operation.
<b>Persistence</b>	The ability of the system to store and retrieve data even after it is shut down or restarted.

### **3 Problem Statement**

#### **3.1 Introduction to the Problem**

In today's financial landscape, banking systems play a pivotal role in managing and safeguarding the financial assets of individuals and businesses. These systems serve as the backbone of the financial sector, facilitating transactions, account management, and financial security.

#### **3.2 Problem Description**

The existing banking systems often face challenges related to user account management, transaction processing, and security. Users encounter difficulties in accessing and managing their accounts efficiently, and security breaches can result in financial losses and compromised user data.

#### **3.3 Significance of the Problem**

Inefficient account management can lead to user frustration and inconvenience, while security vulnerabilities pose significant risks to both users and financial institutions. A well-designed banking information system is essential to enhance user experience, ensure financial security, and streamline banking operations.

#### **3.4 Objectives of the Project**

The primary objectives of the Banking Information System project are to develop a prototype that addresses these challenges. The project aims to create a functional system that simplifies user registration, improves account management, facilitates deposits and withdrawals, enables secure fund transfers, provides users with detailed account statements, implements password protection, and ensures robust error handling.

### **3.5 Scope of the Project**

This project focuses on developing a prototype of a Banking Information System. The prototype will showcase core features and functionalities, including user registration, account management, deposit and withdrawal capabilities, fund transfer functionality, account statement generation, password authentication, and error handling mechanisms.

### **3.6 Target Audience**

The target audience for this system comprises both account holders and bank staff. Account holders seek a user-friendly and secure banking platform, while bank staff require efficient tools for managing user accounts and financial transactions.

### **3.7 Constraints and Limitations**

The project operates within a limited timeframe and budget. The development is based on Core Java, and while the prototype demonstrates critical features, it may not encompass all advanced functionalities found in a complete banking system.

### **3.8 Motivation for the Project**

This project is motivated by the need to address challenges in existing banking systems, improve usability, enhance security, and streamline financial operations. It aims to provide stakeholders with a tangible preview of the key features and functionality of a Banking Information System, allowing for evaluation, feedback, and informed decisions regarding further development and deployment.

## 4 Existing and Proposed solution

### 4.1 Existing System

The existing banking systems often rely on legacy software with limited usability and outdated user interfaces.

User account management can be cumbersome, requiring manual intervention by bank staff.

Security vulnerabilities in authentication and transaction processing pose risks to users and financial institutions.

Account statements may lack detail and user-friendly presentation.

### 4.2 Proposed Solution

#### **User Registration:**

Existing: Manual registration processes.

Proposed: Streamlined user registration form for creating accounts with minimal user input.

#### **Account Management:**

Existing: Manual account management, often requiring bank staff intervention.

Proposed: User-friendly account management interface for users to update their information and settings independently.

#### **Deposit and Withdrawal:**

Existing: Traditional banking transactions, often requiring physical presence.

Proposed: Online deposit and withdrawal functionality for convenient and real-time transactions.

#### **Fund Transfer:**

Existing: Manual transfer processes with delays.

Proposed: Simplified and secure fund transfer features for user-to-user and user-to-account transactions.

**Account Statements:**

Existing: Static and less informative account statements.

Proposed: Dynamic and detailed account statements, displaying transaction history, dates, amounts, and balances.

**Password Protection:**

Existing: Basic password authentication.

Proposed: Robust password protection mechanisms to ensure secure access to user accounts.

**Error Handling:**

Existing: Limited error messages and handling.

Proposed: Comprehensive error handling to provide informative messages for common exceptions, ensuring a smooth user experience.

**User Interface:**

Existing: Outdated and non-intuitive interfaces.

Proposed: Modern and user-friendly interface design for easy navigation and improved user experience.

**Persistence:**

Existing: Limited data persistence.

Proposed: Implementation of basic data persistence to store user account information and transaction history temporarily.

## 4.3 Expected Benefits

**Improved User Experience:** The proposed system aims to enhance user experience by simplifying account management and providing convenient transaction features.

**Enhanced Security:** Robust password protection and secure fund transfer mechanisms will bolster security, reducing the risk of unauthorized access and fraud.



**Real-time Transactions:** Online deposit, withdrawal, and fund transfer capabilities will enable users to perform transactions in real-time, improving efficiency.

**Detailed Account Statements:** Dynamic and detailed account statements will provide users with a comprehensive view of their financial history.

**User-Friendly Interface:** A modern and intuitive user interface will make navigation and banking operations more accessible to users.

**Streamlined Processes:** Automation and streamlined processes will reduce the need for manual intervention by bank staff.

## Conclusion:

The proposed Banking Information System seeks to address the limitations of existing banking systems by introducing user-friendly features, enhanced security, and improved functionality. This project aims to create a prototype that demonstrates these core features, providing stakeholders with a tangible preview of the potential benefits and paving the way for further development and deployment.

#### 4.4 Code submission (Github link):

<https://github.com/obaidullah01/upskillcampus/blob/master/src/bankinginformationsystem.java>

(Note : The Project was created in IntelliJ Idea using many classes the above link leads to the main class

For the whole project class link is : <https://github.com/obaidullah01/upskillcampus/tree/master/src> )

#### 4.5 Report submission (Github link):

[https://github.com/obaidullah01/upskillcampus/blob/master/BankingInformationSystem\\_Obaid\\_USC\\_UCT.pdf](https://github.com/obaidullah01/upskillcampus/blob/master/BankingInformationSystem_Obaid_USC_UCT.pdf)

## 5 Proposed Design/ Model

### 5.1 System Architecture

The Banking Information System will be developed as a client-server application.

It will have a three-tier architecture consisting of a presentation layer, application layer, and data layer.

### 5.2 Presentation Layer

#### User Interface (UI):

The UI will be designed using Java Swing for a modern and user-friendly look and feel.

It will provide easy navigation and access to key functionalities.

#### User Registration:

A simplified registration form will capture user details, including name, address, contact information, and initial deposit amount.

Upon successful registration, a unique 15-digit account number will be generated and displayed as a confirmation.

#### Main Menu:

The main menu will be the central hub for users, allowing them to perform various banking operations.

### 5.3 Application Layer

#### Account Manager:

Responsible for user registration, account management, and user authentication.

Manages user accounts and their associated information.

#### Transaction Manager:

Handles deposit, withdrawal, and fund transfer transactions.

Ensures real-time transaction processing and updates account balances accordingly.

**Account Statement:**

Generates dynamic account statements displaying transaction history, dates, amounts, and balances.

Allows users to access their statements through the UI.

**Authentication Manager:**

Ensures secure access to user accounts by verifying account numbers and passwords.

## 5.4 Data Layer

**Data Storage:**

User account information and transaction history will be temporarily stored during the prototype session.

For a complete system, a relational database system can be integrated for data persistence.

## 5.5 Security

**Password Protection:**

User passwords will be securely hashed and stored in the database.

Passwords will be encrypted during transmission.

**Error Handling:**

Robust error handling will be implemented to prevent unauthorized access and protect against invalid transactions.

## 5.6 User Feedback

Users will have the option to provide feedback or report issues through the UI.

Feedback will be considered for future system enhancements.

## 5.7 Testing

Comprehensive testing, including unit testing and user acceptance testing, will be conducted to ensure system reliability and accuracy.

## 5.8 Scalability

The proposed design will be scalable to accommodate a growing user base and additional features in future iterations.

## 5.9 Documentation

Detailed documentation will be provided to aid in system understanding, troubleshooting, and maintenance.

## 5.10 Conclusion

The proposed design for the Banking Information System outlines a client-server architecture with a modern user interface, robust security measures, and support for key banking functionalities. This design aims to address the challenges of existing banking systems and provide a foundation for the development of a user-friendly, secure, and efficient banking platform.

## 6 Performance Test

### 6.1 Objectives

#### **Scalability Assessment:**

Evaluate the system's ability to scale and handle an increasing number of concurrent users, transactions, and data volume.

#### **Response Time Analysis:**

Measure and optimize response times for various banking operations, such as user login, transaction processing, and account statement generation.

#### **Resource Utilization:**

Monitor and analyze CPU, memory, and network usage to ensure efficient resource utilization and identify potential bottlenecks.

#### **Stress Testing:**

Determine the system's breaking point by subjecting it to loads beyond its capacity to understand how it behaves under extreme conditions.

### 6.2 Types of Performance Testing

#### **Load Testing:**

Simulate expected loads to assess system behavior under typical usage scenarios. Test scenarios will include concurrent user access, transactions, and account inquiries.

#### **Stress Testing:**

Apply extreme loads to assess system stability and determine its failure points. This test will help identify vulnerabilities and weaknesses under heavy loads.

**Volume Testing:**

Evaluate the system's ability to handle large volumes of data, such as transaction records and account statements. This test will ensure that the system can efficiently manage data growth.

**Endurance Testing:**

Assess system performance over an extended period to identify memory leaks, resource exhaustion, and performance degradation over time.

**Concurrency Testing:**

Evaluate how the system handles multiple simultaneous transactions and user interactions, ensuring that it maintains data consistency and integrity.

## 6.3 Methodology

**Test Environment:**

Create a test environment that closely resembles the production environment, including hardware, software, and network configurations.

**Test Scenarios:**

Define realistic test scenarios that replicate user behaviors, including registration, login, deposit, withdrawal, fund transfer, and account statement generation.

**Load Generation:**

Utilize load testing tools to generate user load, simulate transactions, and measure response times.

**Monitoring and Profiling:**

Employ performance monitoring and profiling tools to collect metrics on resource utilization, response times, and system health during tests.

**Analysis and Optimization:**

Analyze test results to identify performance bottlenecks, inefficiencies, and areas for improvement. Optimize code, configurations, and database queries as needed.



### **Reporting:**

Document test results, including response time metrics, resource utilization, and recommendations for performance improvements.

## **6.4 Conclusion**

Performance testing is an integral part of the Banking Information System project to ensure that the system meets performance expectations and delivers a seamless user experience. By conducting various types of performance tests and optimizing the system based on test results, we aim to create a robust and scalable banking platform that can serve users efficiently even under high loads.

## 7 My learnings

### **Java Programming Proficiency:**

The project required a strong grasp of Java programming concepts, including object-oriented programming, data structures, and exception handling.

### **System Architecture:**

Understanding and implementing a three-tier architecture enhanced knowledge of software design patterns and separation of concerns.

### **Security Considerations:**

Implementing password hashing and secure authentication methods highlighted the importance of security in software development.

### **Error Handling:**

Developing error handling mechanisms improved the ability to anticipate and address potential issues in software.

### **Project Management:**

Managing tasks, organizing code, and adhering to coding standards demonstrated the significance of structured project management.

### **Problem Solving:**

Resolving errors and challenges throughout the project honed problem-solving skills and the ability to troubleshoot issues effectively.

### **User-Centric Approach:**

Focusing on user needs and feedback reinforced the importance of designing software with the end user in mind.

### **Testing and Quality Assurance:**

Conducting unit testing and user acceptance testing underscored the value of quality assurance in ensuring software reliability.

### **Documentation Skills:**

Creating detailed documentation for the project emphasized the importance of clear and comprehensive documentation for future reference.

#### **Performance Testing:**

Planning and executing performance tests provided insights into system scalability and optimization.

## 8 Future work scope

### Advanced Security Features:

- Implement two-factor authentication (2FA) for enhanced security.
- Integrate biometric authentication methods, such as fingerprint or facial recognition.
- Enhance security auditing and monitoring to detect and respond to suspicious activities.

### Internationalization and Localization:

- Support multiple languages and locales to cater to a global user base.
- Implement currency conversion for real-time exchange rate updates.

### Enhanced User Profiles:

- Allow users to customize their profiles, including profile pictures and personal preferences.
- Enable users to reset forgotten passwords securely.

### Notification System:

- Implement a notification system to alert users about account activity, transaction confirmations, and important announcements.

### Mobile Application:

- Develop a mobile app version of the Banking Information System for on-the-go banking.

### Account Alerts:

- Allow users to set up alerts for low balances, large transactions, and other account activities.

### Bill Payment and Transfer Scheduling:

- Integrate bill payment features, enabling users to schedule and automate recurring payments.
- Implement fund transfer scheduling for future-dated transactions.

### Investment and Savings Accounts:

- Extend the system to support various types of accounts, including savings, fixed deposits, and investment accounts.

**Data Encryption:**

- Strengthen data encryption methods to protect sensitive information during transmission and storage.

**Advanced Reporting:**

- Provide users with advanced reporting options, including graphical representations of financial data.
- Implement predictive analytics for financial forecasting.

**Regulatory Compliance:**

- Ensure compliance with financial regulations and standards, such as Know Your Customer (KYC) and Anti-Money Laundering (AML) requirements.