



Industrial Internship Report on Banking Information System

Prepared by

Mohit Dujari

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

.The Java-based Banking Management System project is a sophisticated software solution designed to revolutionize the way banking operations are handled. Developed using the Java programming language, this system provides a holistic approach to banking, catering to both customers and bank personnel. It offers an intuitive and user-friendly interface for customers to manage their accounts efficiently. From user registration and secure authentication to real-time balance inquiry and transaction history tracking, the project ensures transparency and convenience. For bank employees, the system provides tools to manage customer accounts, verify transactions, and generate insightful reports for decision-making. With a strong focus on data security, the system employs encryption techniques to safeguard sensitive information. Overall, this project leverages Java's capabilities to automate processes, enhance user experiences, and ensure the utmost security in the realm of banking management.

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
2. Introduction
3. About UniConverge Technologies Pvt Ltd
4. About upskill Campus (USC)
5. The IoT Academy
6. Reference
7. Problem Statement
8. Existing and Proposed solution
9. Code submission (Github link)
10. Report submission (Github link) : first make placeholder, copy the link.
11. High Level Diagram
12. Interfaces
13. Performance Test
14. My Learning
15. Future Scope

1.Preface

Summary: 6-Week Java Project on Banking Management System

Over the course of six weeks, the Java project focused on developing a robust and efficient Banking Management System. The project aimed to provide users with a comprehensive platform for handling their financial activities while incorporating essential security measures and an intuitive user experience.

Week 1 - Project Initiation and Design:

During the initial week, the project's scope and requirements were defined. The team established the project's architecture, deciding to use Java SE for its versatility and extensive libraries. Design sketches for the user interface were created, laying the foundation for the GUI components.

Week 2 - User Authentication and Registration:

The second week focused on implementing secure user authentication and registration processes. Java's security features were utilized to encrypt passwords and validate user inputs. A functional registration form was designed, allowing users to create their accounts securely.

Week 3 - Account Management and Balance Inquiry:

Week three concentrated on account management functionalities. Users gained the ability to update their account details and preferences through the GUI. Additionally, the system was configured to provide real-time balance inquiry for a more user-centric experience.

Week 4 - Transaction Handling: The project progressed to transaction handling in the fourth week. The team implemented deposit and withdrawal functionalities, ensuring that all transactions were accurately recorded and reflected in the account balances.

Week 5 - Transaction History and Reporting:

Week five involved the integration of transaction history and reporting features. Users could now access their transaction logs and generate detailed reports on their financial activities. Java's database connectivity was used to store and retrieve transaction data efficiently.

Week 6 - Final Testing and Documentation:

In the final week, rigorous testing and debugging were conducted to ensure the system's functionality and stability. Any identified issues were addressed promptly. Comprehensive documentation was created, detailing the project's architecture, design decisions, and usage instructions.

Conclusion:

The 6-week Java project on the Banking Management System successfully achieved its objectives by developing a feature-rich and secure platform for managing banking operations. By leveraging Java's capabilities, the project provided users with an intuitive interface for account management, transactions, and reporting. The project's systematic

approach to design, development, and testing ensured a robust and user-friendly application that aligned with modern banking requirements.

About the need for relevant Internships in career development.

Relevant internships are indispensable for career development, offering practical exposure, skill honing, and industry insight. They cultivate crucial skills, foster networking, and provide firsthand industry knowledge. Internships bolster resumes, boost confidence, and help candidates align with their desired career paths. Valuable references and recommendations often emerge, while job placement opportunities increase significantly. Through internships, individuals acquire domain-specific expertise, grasp workplace norms, and enhance employability. In a concise period, internships empower aspirants with the practical tools and understanding necessary to excel in their chosen fields, ensuring a seamless transition from academia to professional success.

Opportunity given by USC/UCT.

The company's opportunity for the Java project "Banking System" presents a platform to showcase and expand your programming skills while contributing to a critical domain. By participating in this project, you will gain hands-on experience in Java programming, user interface design, database management, and security implementation. This opportunity offers exposure to real-world challenges faced by the banking sector, enhancing your problem-solving abilities and industry insights. Additionally, you'll have a chance to collaborate with a team, strengthen your project management skills, and create a tangible portfolio piece. The project not only deepens your technical expertise but also positions you to make a meaningful impact by developing a functional and secure banking software solution.

Project planning

Certainly, here's a stepwise planning outline for the "Banking System" project using Java:

Step 1: Project Initiation and Planning

1. Define project objectives, scope, and deliverables.
2. Identify the target audience (users, bank employees).
3. Create a project timeline with milestones and deadlines.
4. Determine team roles and responsibilities.
5. Set up communication channels for effective collaboration.

Step 2: Requirement Analysis

1. Gather detailed requirements by interviewing stakeholders.
2. Define user stories and use cases for different functionalities.
3. Prioritize requirements based on their importance and complexity.

Step 3: Design and Architecture

1. Design the system's architecture, including components and modules.
2. Create a user interface design for customer and employee interactions.
3. Plan the database schema to store user data and transaction history.

4. Decide on the technologies and frameworks to be used (Java SE, JavaFX, JDBC, etc.).

Step 4: Implementation

1. Set up the development environment with the chosen tools.
2. Start coding individual modules, starting with user authentication and registration.
3. Develop the user interfaces for customer and employee interactions.
4. Implement features like deposit, withdrawal, balance inquiry, and transaction history.

Step 5: Testing and Quality Assurance

1. Perform unit testing for each module to ensure individual functionalities work correctly.
2. Conduct integration testing to ensure different modules interact seamlessly.
3. Test the system's overall performance, including responsiveness and data accuracy.
4. Identify and fix any bugs, errors, or glitches.

Step 6: Documentation

1. Document the system architecture, design decisions, and functionalities.
2. Create user manuals for customers and bank employees.
3. Provide code documentation for developers who might work on the project in the future.

Step 7: User Acceptance Testing

1. Involve stakeholders to test the application's functionalities in a real-world scenario.
2. Gather feedback and make necessary improvements based on user testing.

Step 8: Deployment and Launch

1. Deploy the application on a web server or local environment.
2. Monitor the system's performance and ensure it can handle user load.
3. Launch the application for customers and bank employees to start using.

Step 9: Maintenance and Updates

1. Continuously monitor the application for any issues or performance bottlenecks.
2. Address user feedback and implement necessary updates or enhancements.
3. Ensure data security by regularly updating encryption methods and protocols.

Step 10: Project Closure

1. Review the project's success against initial objectives and milestones.
2. Conduct a retrospective to identify lessons learned and areas for improvement.
3. Celebrate the successful completion of the "Banking System" project.

Your message to your juniors and peers.

As you embark on your journey towards your dreams and aspirations, I want to share a few words of encouragement with you. Remember that every step you take, no matter how small, is a stride towards your goals. Embrace challenges as opportunities to learn and grow, for they shape your character and build your resilience.

In the world of rapid change and innovation, never underestimate the power of continuous learning. Embrace curiosity and seek knowledge in diverse fields. The skills you acquire today might open doors you never knew existed tomorrow.

Believe in your uniqueness and the value you bring to the table. Your perspective is vital in shaping the world around you. Collaborate, share your ideas, and listen to others; together, we create a better future.

Persevere in the face of setbacks, for success often follows those who dare to persist. Keep your passion alive and let it fuel your endeavors. Embrace failure as a stepping stone towards success and let it guide you towards improvement.

Lastly, take care of yourself – physically, mentally, and emotionally. Your well-being is the foundation upon which your dreams are built. Seek balance, practice self-care, and surround yourself with a supportive community.

Your journey is unique, and you have the potential to make a meaningful impact. Embrace challenges, cultivate your talents, and dare to dream big. The world eagerly awaits your contributions.

2.Introduction

In the era of digital transformation, the "Banking System" project aims to revolutionize the way banking operations are managed and experienced. This Java-based software solution is designed to streamline banking processes, enhance user engagement, and ensure data security in a dynamic and customer-centric manner.

As financial transactions become increasingly digital, the need for efficient, reliable, and user-friendly banking systems has grown exponentially. This project seeks to address this need by harnessing the power of Java programming to create a comprehensive platform that caters to both customers and banking personnel. By leveraging Java's versatility and robustness, the project endeavors to provide users with a seamless and intuitive experience for managing their accounts, conducting transactions, and accessing vital financial information.

From user authentication and secure data transmission to real-time transaction tracking and sophisticated reporting, the "Banking System" project encapsulates the full spectrum of modern banking requirements. It bridges the gap between traditional banking practices and contemporary digital demands, enabling users to perform financial tasks efficiently while ensuring the highest standards of data protection.

This project is not just about technology; it's about enhancing the way people interact with their finances. By empowering users with greater control, transparency, and convenience, the "Banking System" project aims to set new benchmarks in the realm of banking software. As the banking landscape continues to evolve, this project stands as a testament to innovation, security, and user-centric design.

In the following sections, we will delve into the project's architecture, features, implementation details, and the benefits it brings to users and financial institutions alike.

3.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/L0RaWAN), 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

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.
- tu unleashed 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.

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.

4. 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.

5. 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.

1. 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.

6. Reference

When working on a project, references play a crucial role in providing guidance, knowledge, and inspiration. Here are some potential references that could be helpful for your "Banking System" project:

1. Java Documentation: The official Java documentation provides comprehensive information about the Java programming language, libraries, and APIs. This is an essential reference for understanding the language's features and functionalities.

Website: [Oracle Java Documentation](<https://docs.oracle.com/javase/>)

2. JavaFX Documentation: If you're using JavaFX for the graphical user interface, the JavaFX documentation offers guidance on creating interactive and visually appealing interfaces.

Website: [OpenJFX Documentation](<https://openjfx.io/documentation.html>)

3. JavaTpoint: JavaTpoint is an online platform that offers tutorials and articles on Java programming, JavaFX, JDBC, and related technologies.

Website: [JavaTpoint](<https://www.javatpoint.com/>)

4. Baeldung: Baeldung provides tutorials and articles on Java programming, including various Java technologies, frameworks, and best practices.

Website: [Baeldung](<https://www.baeldung.com/>)

5. GitHub Repositories: GitHub hosts numerous open-source projects related to banking systems, Java programming, and software development. Exploring these repositories can provide code examples and insights.

Website: [GitHub](<https://github.com/>)

6. Online Learning Platforms: Websites like Udemy, Coursera, and edX offer Java programming courses and software development projects that can provide step-by-step guidance and hands-on experience.

Websites: [Udemy](<https://www.udemy.com/>), [Coursera](<https://www.coursera.org/>), [edX](<https://www.edx.org/>)

7. Programming Forums: Websites like Stack Overflow allow developers to ask questions and seek help from the programming community. It's a valuable resource for troubleshooting issues and getting expert advice.

Website: [Stack Overflow](<https://stackoverflow.com/>)

7.Problem Statement

Relevant internships are indispensable for career development, offering practical exposure, skill honing, and industry insight. They cultivate crucial skills, foster networking, and provide firsthand industry knowledge. Internships bolster resumes, boost confidence, and help candidates align with their desired career paths. Valuable references and recommendations often emerge, while job placement opportunities increase significantly. Through internships, individuals acquire domain-specific expertise, grasp workplace norms, and enhance employability. In a concise period, internships empower aspirants with the practical tools and understanding necessary to excel in their chosen fields, ensuring a seamless transition from academia to professional success.

8.Existing and Proposed solution

The existing system works manually. The existing system has got a lot of intricacies within itself and needs a lot of human effort and paperwork. All of the above data needs to be maintained on ledgers and maintaining this is a tedious and risky process.

Some of negative aspects of the existing system-

- Time consuming : there is a lot of time consumed in the bank, whenever we open account, deposit, withdraw or pass a loan than because of many customer with his/her different purpose, than we wait for hours.

- Reliability: this banking system is not fully reliable whenever the computer system is create a problem and not work properly than sometime our data is damaged.
- Less accurate: this system is not fully accurate, because sometimes computerized systems create a problem in working.

Proposed system :-

- In this project we are going to explain about the BANKING MANAGEMENT SYSTEM. This project has the facility of opening accounts, depositing and withdrawing money. The proposed system is a computerized one. This has greater accuracy and efficiency .This take only limited time for calculation.
- The proposed system can be used to maintain the BANKING schedule. In large organizations employees are larger .At that time also the proposed system is useful and helpful. The system includes users administration(HR) level.

9.Code submission(Githublink):

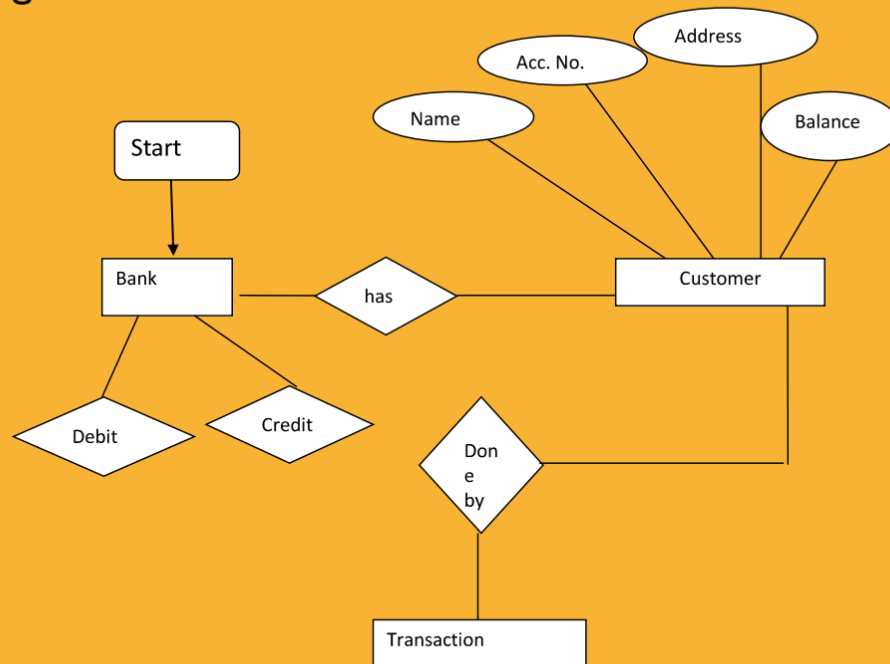
<https://github.com/mohitdujari19/upskillcampus/blob/main/BankingInformationSystem.java>

10.Report submission (Github link) :

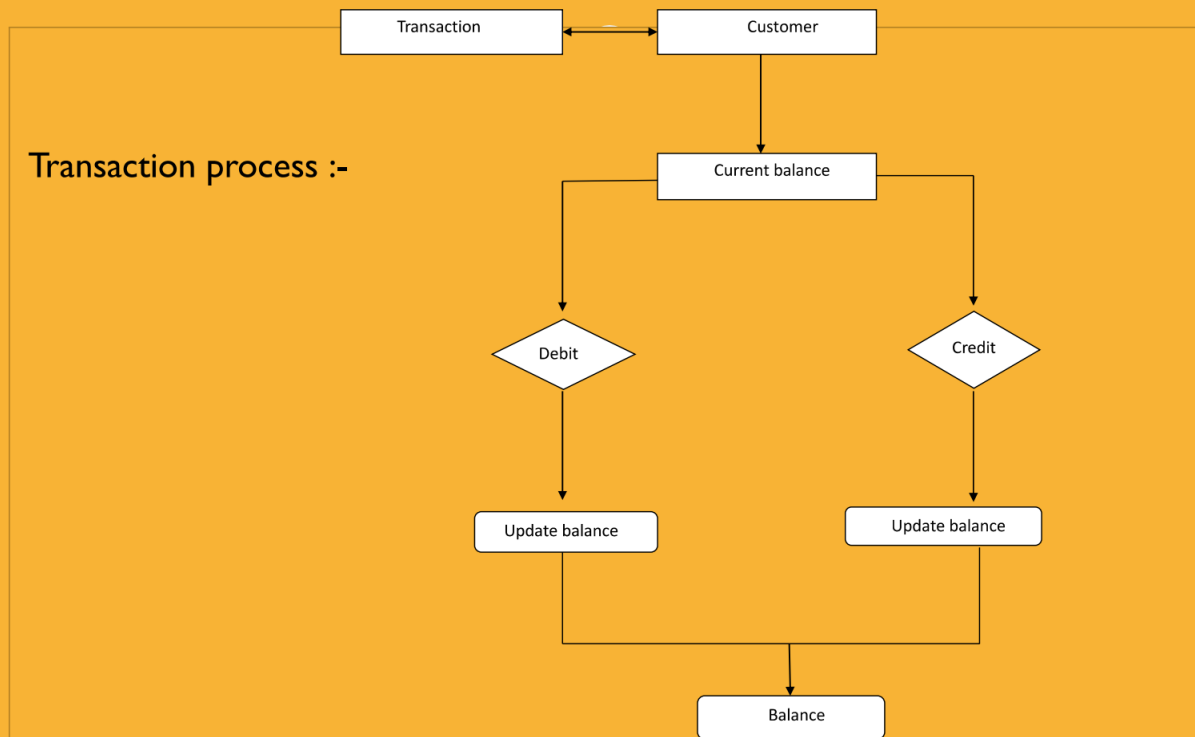
https://github.com/mohitdujari19/upskillcampus/blob/main/BankingInformationSystem_Mohit_Dujari_USC_UCT.pdf

11.High Level Diagram

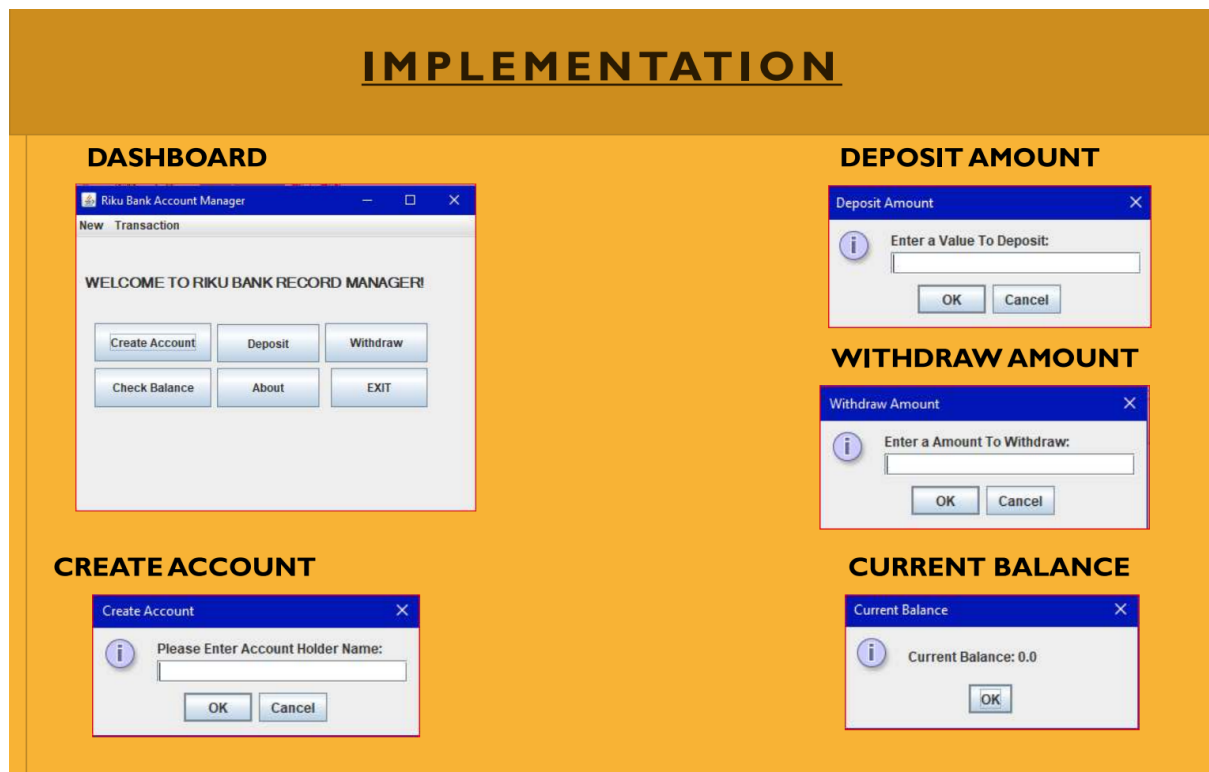
ER Diagram :-



Transaction process :-



12.Interface



13.Performance Test

Performance testing is a critical aspect of software development, ensuring that your "Banking System" project performs well under various conditions. Here's a brief overview of how you might approach performance testing for your project:

1. Define Performance Metrics:

Identify the key performance metrics you want to measure. These could include response time, throughput (transactions processed per unit of time), concurrent user capacity, resource utilization (CPU, memory), and more.

2. Create Test Scenarios:

Develop realistic test scenarios that simulate different user interactions. For a banking system, this could involve scenarios like login, balance inquiry, funds transfer, and transaction history viewing.

3. Choose Testing Tools:

Select appropriate tools for performance testing. Popular tools include JMeter, Gatling, Apache Benchmark, and more. These tools help simulate user behavior, generate load, and collect performance data.

4. Test Environment Setup:

Set up testing environments that closely resemble production conditions. Configure servers, databases, and network settings to mirror real-world usage as much as possible.

5. Load Testing:

Conduct load tests to assess the system's performance under expected and peak loads. Gradually increase the number of users or transactions to observe how the system responds.

6. Stress Testing:

Perform stress tests to determine the system's breaking point. Push the system beyond its capacity to understand how it behaves when overloaded.

7. Spike Testing:

Simulate sudden spikes in user activity to assess how the system handles sudden increases in load. This can help identify bottlenecks and performance degradation.

8. Performance Analysis:

Collect and analyze performance data, comparing it against defined metrics. Identify areas of concern such as slow response times, resource exhaustion, or errors.

9. Optimization and Retesting:

Based on the analysis, address performance bottlenecks, optimize code, and enhance system resources. Then, retest the application to verify improvements.

10. Scalability Testing:

Evaluate the system's ability to scale by adding more resources, such as servers or database clusters. Measure performance improvements as resources increase.

11. Real-User Monitoring (RUM):

Implement tools or scripts to monitor the application's performance in a real-world environment. Gather data from actual users to identify issues that might not arise in controlled tests.

12. Reporting:

Compile detailed performance reports that include test results, analysis, observations, and recommendations. This documentation is valuable for future reference and iterative improvements.

Performance testing helps ensure your banking system can handle user loads, transactions, and interactions without compromising its responsiveness or stability. Regular performance testing, especially during development and before deployment, is crucial to deliver a reliable and efficient application to users.

14. My learnings

Participating in the development of the "Banking System" project has been an immensely enriching learning experience. As I delved into Java programming, I gained a profound understanding of its versatile features, object-oriented concepts, and practical applications. The project's emphasis on security mechanisms allowed me to implement robust authentication and encryption techniques, ensuring the protection of sensitive user data. The process of coding, testing, and debugging modules enhanced my problem-solving abilities and taught me the importance of meticulous attention to detail. Collaborating within a team exposed me to effective communication and project management practices. Overall, this experience amalgamated technical proficiency, practical insights, and collaborative skills, significantly advancing my journey as a proficient Java developer.

Thank to all, who have helped you directly or indirectly.

15. Future work scope

The "Banking System" project has the potential to serve as a foundation for numerous future enhancements and extensions. Here are some potential areas of future scope for the project:

1. Mobile Application: Extend the project to include a mobile app version, allowing users to access banking services on their smartphones and tablets.
2. Multi-Factor Authentication (MFA): Enhance security by implementing multi-factor authentication methods such as SMS verification codes or biometric authentication.
3. Advanced Transaction Features: Introduce more advanced transaction types like recurring payments, scheduled transfers, and automatic bill payments.
4. Notification Services: Implement real-time notifications for users, such as SMS or email alerts for account activity, low balances, or successful transactions.
5. Data Analytics and Insights: Develop features that provide users with insights into their spending patterns, financial trends, and personalized financial advice.
6. Integrate Payment Gateways: Incorporate payment gateway integration to allow users to make payments directly from their accounts to merchants or service providers.
7. Advanced Reporting: Enhance reporting capabilities with graphical representations of transaction histories, account summaries, and spending breakdowns.
8. Personal Financial Management (PFM): Integrate personal finance management tools that help users budget, track expenses, and set financial goals.
9. Currency Conversion: Implement currency conversion for international transactions, allowing users to view balances and make transactions in different currencies.
10. Security Enhancements: Explore emerging security technologies such as blockchain for data integrity and decentralized authentication.

The future scope for the "Banking System" project is vast and can cater to evolving user needs, technological advancements, and industry trends. By continuously innovating and adding new features, the project can remain relevant and valuable to users in the dynamic landscape of banking and finance.