Industrial Internship Report on

”Bank Management System”

Prepared by

Abhishek Raju Wadile.

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 In the dynamic landscape of modern banking, efficient management systems play a pivotal role in ensuring smooth operations, security, and customer satisfaction. The advent of technology has revolutionized traditional banking practices, leading to the emergence of sophisticated Bank Management Systems (BMS). These systems integrate advanced functionalities to streamline various banking operations, ranging from customer transactions to administrative tasks.

Java, renowned for its versatility, robustness, and platform independence, stands as a prime choice for developing Bank Management Systems. Leveraging Java's object-oriented programming paradigm and extensive ecosystem of libraries and frameworks, developers can construct scalable, secure, and efficient BMS solutions tailored to the unique requirements of financial institutions.

This project aims to develop a comprehensive Bank Management System using Java, encompassing essential features such as customer management, account handling, transaction processing, and administrative functionalities. By employing Java's rich toolset and adhering to best practices in software engineering, this system endeavors to enhance operational efficiency, minimize errors, and ensure regulatory compliance within the banking domain.

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

It really very difficult to understand in week 1 how to create this project but I decided to do it with in 5 week . I really got lot's of errors and I facing lot's of problems but still I have consistency to do it in batter way. I searching every problem and errors how to fix it. I continuously doing work on this project finally now I completed my project .

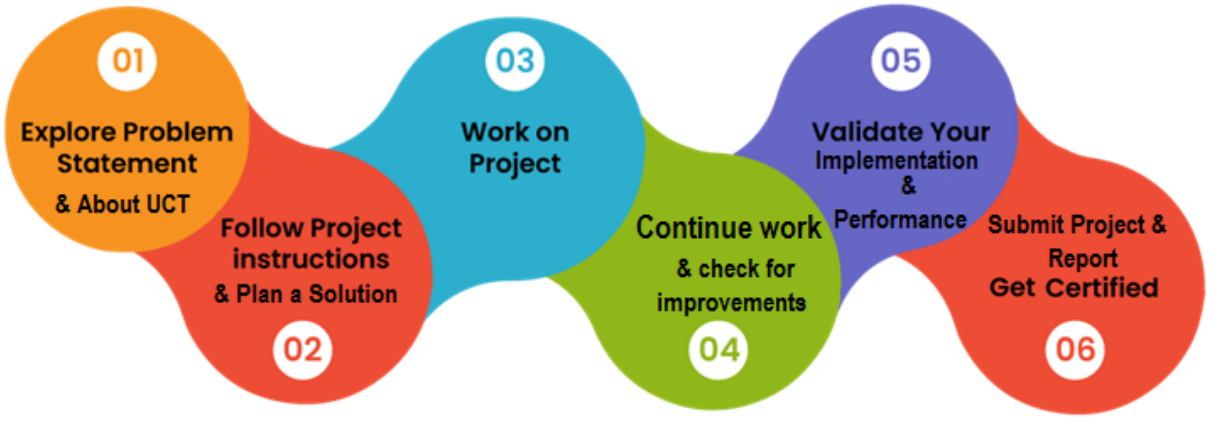
My project statement is **"BANK MANAGEMENT SYSTEM"** which includes creation of bank account of customer and also transfers money and deposit and withdrew the money .

USC and UCT gives me this opportunity to prove my skill and to build my first project in java language. Without them I couldn't to complete my this internship . I really want to say thanks wholeheartedly to USC and UCT who gives me this opportunity.

Here I learn how Java language works and how we can use our logic to build this type of projects and how we can faces lots of problem and I overcome on it where I gives my best to this project .

I want to say first thanks to USC and UCT who gave me this opportunity to show my skill and also I want to say thanks to W3School and YouTube which they solves code errors and guide me .

Eventually I want to say to juniors and peers that Don't Give Up if it doesn't work just move forward and try to understand what happen try to fix it up to till where we can't get our destination ,If fails don't worry put consistency .The A. P. J. Abdul Kalam says "A diamond is just a piece of charcoal that handle lots of stress and heat to become diamond from charcoal".



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.

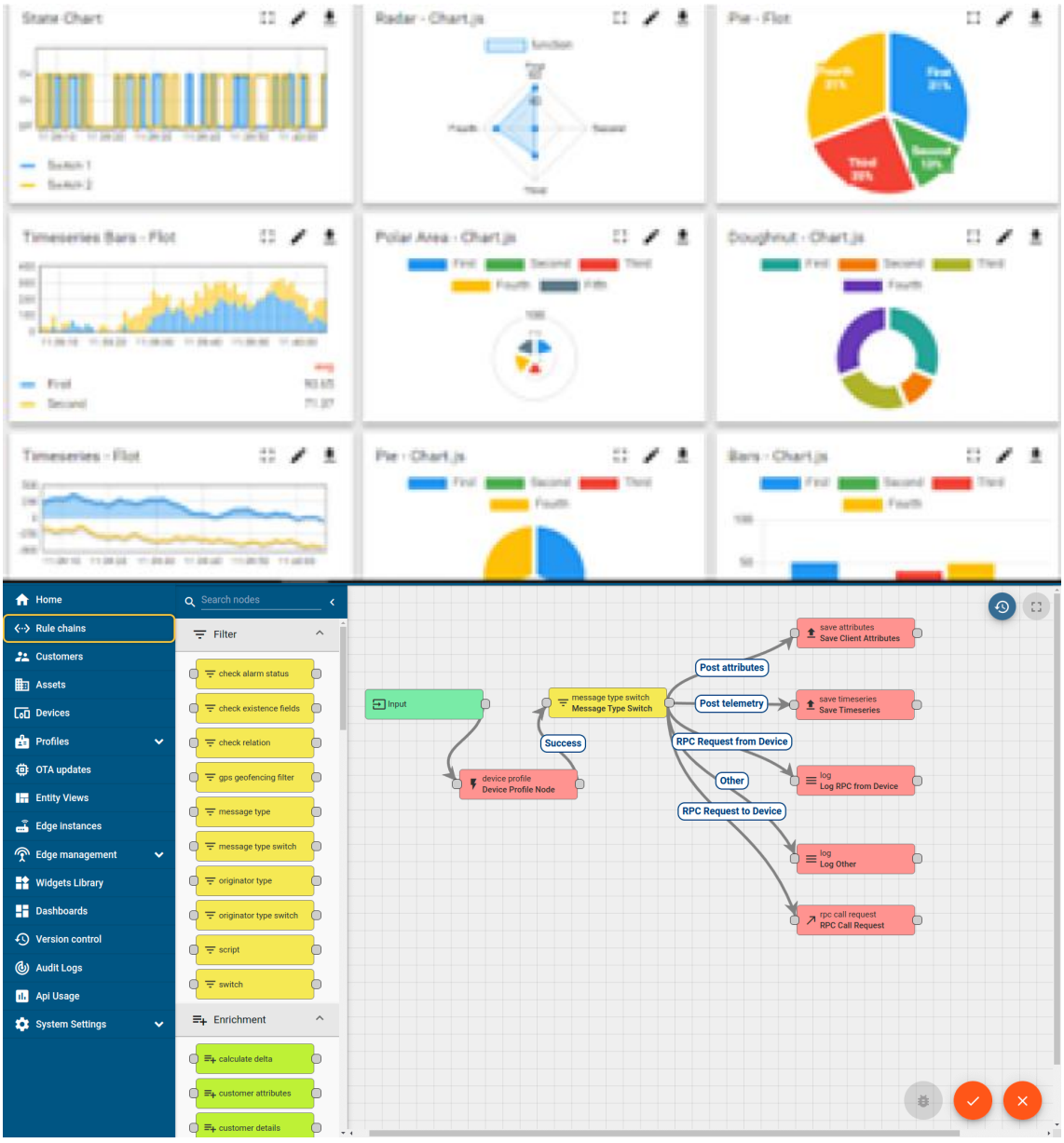
i. UCT IoT Platform(UCT Insight)

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)

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.

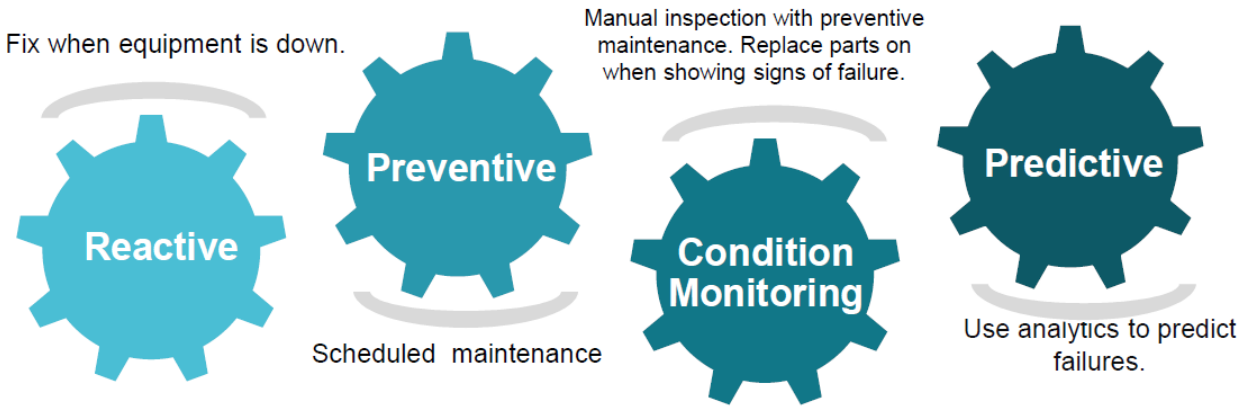
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

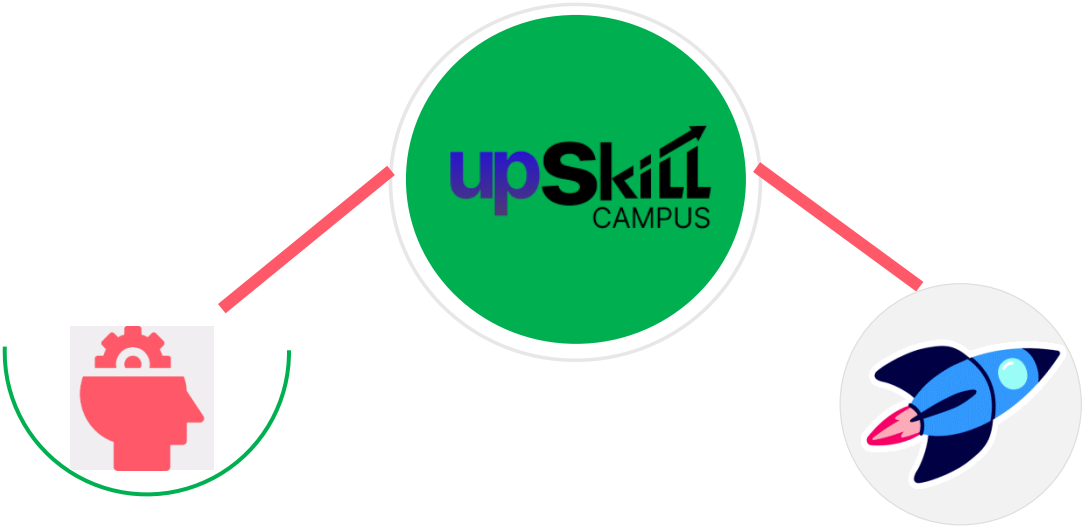
UCT isproviding Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

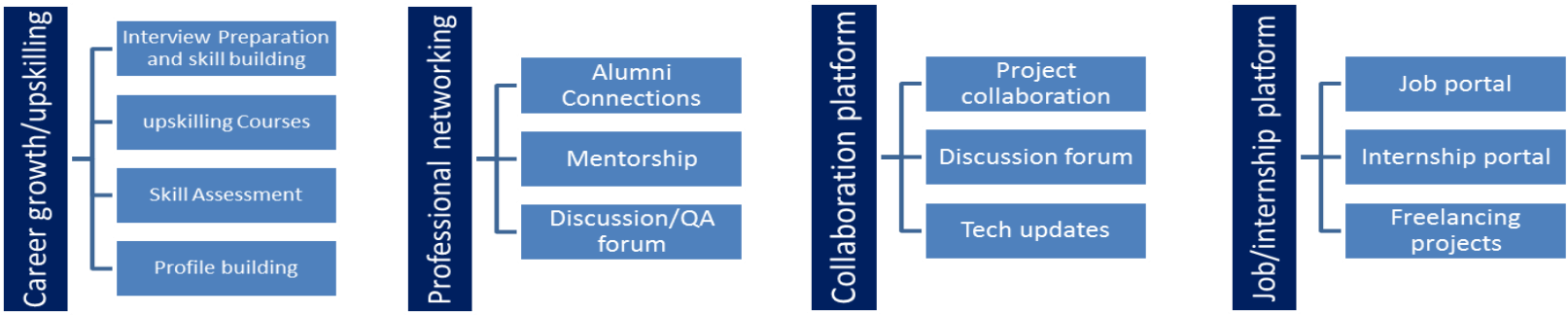
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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] https://www.uniconvergetech.in/

[2] https://learn.upskillcampus.com/

[3]

2.6 Glossary

Terms	Acronym

3 Problem Statement

A Bank Management System built using Java encompasses various modules such as customer management, account management, transaction processing, loan management, and administrative functions. These modules interact seamlessly to provide a comprehensive solution for managing banking operations efficiently.

Key features of a Bank Management System developed in Java may include:

1. User Authentication and Authorization: Secure login mechanisms to authenticate users and provide access based on their roles and privileges.

2. Customer Management: Facilities for managing customer information, including account creation, updating personal details, and maintaining KYC (Know Your Customer) compliance.

3. Account Management: Functions for creating and managing different types of accounts such as savings accounts, current accounts, and fixed deposit accounts. This includes features like balance inquiry, fund transfer, and account closure.

4. Transaction Processing: Capability to process various types of transactions such as deposits, withdrawals, transfers, and bill payments securely and efficiently.

5. Loan Management: Tools for managing loan applications, approvals, disbursements, and repayments. This may include features like loan eligibility checks, EMI (Equated Monthly Installment) calculations, and interest rate management.

6. Reporting and Analytics: Generation of reports and analytics to monitor the bank's performance, track transactions, and comply with regulatory requirements.

7. Security and Audit: Implementation of robust security measures to protect sensitive data and prevent unauthorized access. Regular auditing to ensure compliance with internal policies and regulatory standards.

8. Integration with External Systems: Integration with external systems such as payment gateways, credit bureaus, and core banking systems to facilitate seamless operations and improve efficiency.

4 Existing and Proposed solution

Existing Bank Management Systems developed by various vendors often provide comprehensive features for managing banking operations. However, some common limitations include:

1. Proprietary Software: Many existing solutions are proprietary, meaning they may have high licensing costs and limited flexibility for customization.

2. Platform Dependency: Some solutions may be platform-dependent, making it challenging to deploy and maintain across different operating systems.

3. Scalability Issues: Certain systems may face scalability challenges, especially during peak transaction periods, leading to performance issues and system downtimes.

4. Security Concerns: Security vulnerabilities such as weak authentication mechanisms, inadequate data encryption, and insufficient access controls may pose risks to sensitive financial data.

5. Lack of Integration: Integration with external systems and third-party services may be limited, hindering interoperability and workflow automation.

6. User Experience: The user interface of some existing systems may be outdated or not intuitive, leading to poor user experience for bank staff and customers.

7. Regulatory Compliance: Ensuring compliance with evolving regulatory standards and requirements can be complex and may not be fully addressed by existing solutions.

Proposed Solution:

Our proposed solution is to develop a Bank Management System using Java that addresses the limitations of existing solutions while providing enhanced features and capabilities. Key aspects of our proposed solution include:

1. Open-Source Framework: Utilizing open-source technologies and frameworks to promote flexibility, reduce costs, and encourage community contributions for continuous improvement.

2. Platform Independence: Developing the system to be platform-independent, allowing deployment across various operating systems and environments seamlessly.

3. Scalability and Performance: Implementing scalable architecture and optimizing performance to handle large transaction volumes efficiently without compromising system stability.

4. Enhanced Security Measures: Incorporating robust security measures such as multi-factor authentication, encryption of sensitive data, role-based access controls, and regular security audits to ensure data protection and compliance with regulatory standards.

5. Integration Capabilities: Designing the system with flexible APIs and support for industry-standard protocols to facilitate seamless integration with external systems, payment gateways, and other financial services.

6. User-Centric Design: Prioritizing user experience by designing an intuitive and user-friendly interface for bank staff and customers, enhancing productivity and satisfaction.

7. Regulatory Compliance: Staying up-to-date with regulatory requirements and standards, and incorporating features for compliance monitoring, reporting, and audit trails to ensure adherence to legal and regulatory obligations.

Value Addition:

Our proposed solution aims to add value by:

1. Cost-Effectiveness: Providing a cost-effective alternative to proprietary solutions by leveraging open-source technologies and community support.

2. Flexibility and Customization: Offering flexibility for customization and adaptation to specific bank requirements, ensuring that the system aligns closely with the bank's business processes and objectives.

3. Improved Security and Compliance: Enhancing security measures and compliance features to protect sensitive data and mitigate risks, thereby enhancing trust and confidence among stakeholders.

4. Enhanced User Experience: Prioritizing user experience through intuitive design and streamlined workflows, leading to increased efficiency and satisfaction for both bank staff and customers.

5. Scalability and Performance: Ensuring scalability and optimal performance to accommodate future growth and evolving business needs, without compromising system reliability or responsiveness.

4.1 Code submission (Github link)

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

5 Proposed Design/ Model

1. Display All Account Details:

- Upon selecting this option, the system will retrieve and display all account details stored in the database. This information may include account number, account holder name, account type, balance, and any other relevant details.

- The system will present the data in a tabular format or through a user-friendly interface, allowing bank staff to view and access account information efficiently.

2. Search by Account Number:

- When selecting this option, the system prompts the user to enter the account number they wish to search for.

- Upon receiving the input, the system will query the database to retrieve the account details associated with the provided account number.

- If the account number exists, the system will display the corresponding account information. If not, it will notify the user that the account does not exist.

3. Deposit the Amount:

- Upon selecting this option, the system will prompt the user to enter the account number and the amount they wish to deposit.

- The system will validate the input and verify the existence of the account.

- If the account exists, the specified amount will be added to the account balance, and the transaction will be recorded in the transaction history.

- The updated account balance will be displayed to confirm the successful deposit.

4. Withdraw the Amount:

- When choosing this option, the system will request the user to enter the account number and the amount they want to withdraw.

- The system will validate the input, verify the existence of the account, and check if the account balance is sufficient for the withdrawal.

- If the account exists and has enough balance, the specified amount will be deducted from the account balance, and the transaction will be recorded.

- The updated account balance will be displayed to confirm the successful withdrawal.

5. Exit:

- Selecting this option will terminate the execution of the Bank Management System, ending the user session and closing the application.

Proposed System Flow:

- Upon launching the Bank Management System, users will be presented with a menu containing the above options.

- Users can navigate through the menu by selecting the desired option using numeric input or corresponding keywords.

- The system will validate user inputs at each step to ensure accuracy and prevent errors.

- All transactions and interactions with the system will be logged for auditing and security purposes.

- The system will provide clear and informative prompts and messages to guide users through each operation and handle exceptions gracefully.

- Users can perform multiple operations sequentially or choose to exit the system when they have completed their tasks.

6 Performance Test

Identifying Constraints:

1. Memory Usage:

- Constraint: The system should efficiently utilize memory resources to avoid excessive memory consumption, which could lead to performance degradation or even system crashes.

- Approach: Implement efficient data structures and algorithms to minimize memory usage. Use memory profiling tools to identify and optimize memory-intensive components.

2. Processing Speed (MIPS - Million Instructions Per Second):

- Constraint: The system should process transactions and queries within acceptable timeframes to ensure responsiveness and meet user expectations.

- Approach: Optimize algorithms and database queries to minimize processing time. Utilize multithreading or parallel processing techniques for concurrent operations.

3. Accuracy:

- Constraint: The system must accurately perform financial calculations, transaction processing, and data retrieval to maintain data integrity and prevent financial discrepancies.

- Approach: Implement robust error handling mechanisms, conduct thorough testing, and ensure compliance with financial regulations and standards.

4. Durability:

- Constraint: The system should maintain data integrity and ensure uninterrupted operation even in the event of hardware failures, software crashes, or system disruptions.

- Approach: Implement transaction logging, database backups, and failover mechanisms to ensure data durability and system resilience.

5. Power Consumption:

- Constraint: The system should consume minimal power to operate efficiently, especially in scenarios where it runs on battery-powered devices or in energy-constrained environments.

- Approach: Optimize code for energy efficiency, minimize resource-intensive operations, and implement power-saving features where applicable.

Handling Constraints in the Design:

1. Memory Usage:

- Implementing efficient data structures such as balanced trees or hash maps for storing account information.

- Optimizing database queries and caching frequently accessed data to reduce memory overhead.

- Employing memory profiling tools during development and testing phases to identify and address memory leaks or inefficiencies.

2. Processing Speed:

- Utilizing indexing and query optimization techniques in the database to improve query performance.

- Employing asynchronous processing for non-blocking operations to enhance system responsiveness.

- Load testing the system to identify performance bottlenecks and optimizing critical components accordingly.

3. Accuracy:

- Implementing comprehensive unit tests, integration tests, and end-to-end tests to verify the accuracy of financial calculations and transaction processing.

- Conducting peer reviews and code inspections to identify potential sources of inaccuracies and address them proactively.

4. Durability:

- Implementing transaction logging to ensure data consistency and recoverability in case of system failures.

- Setting up database replication and failover mechanisms to minimize downtime and maintain continuous operation.

5. Power Consumption:

- Minimizing unnecessary background processes and idle resource consumption.

- Implementing power-saving features such as sleep modes or dynamic frequency scaling for energy-efficient operation.

Test Results and Recommendations:

- Performance testing should be conducted under realistic scenarios simulating expected user loads and transaction volumes.

- Monitor system metrics such as CPU usage, memory utilization, response times, and transaction throughput during performance testing.

- Identify any performance bottlenecks and address them through optimization techniques or infrastructure upgrades.

- Implement monitoring and alerting mechanisms to detect and respond to performance degradation in real-time.

- Regularly review and fine-tune system performance to ensure optimal operation under varying conditions.

7 My learnings

Summary:

Over the past weeks, my focus has been on improving the user experience and reliability of a beginner-level Bank Management System. I have emphasized the importance of providing a smooth experience for customers, ensuring trust and reliability in the system. By leveraging Java packages and frameworks, I have aimed to enhance scalability and runtime speed, making the system more efficient.

Key areas of progress include:

1. User Experience Enhancement: Prioritizing user experience improvements to ensure a seamless and intuitive interface for customers interacting with the Bank Management System.

2. Reliability and Trustworthiness: Implementing measures to enhance the reliability and trustworthiness of the system, thereby instilling confidence in customers regarding the security and accuracy of their financial transactions.

3. Utilization of Java Packages: Leveraging existing Java packages and libraries to streamline development and enhance functionality, thereby accelerating the development process while maintaining reliability.

4. Scalability and Runtime Optimization: Focusing on enhancing the scalability and runtime speed of the system to accommodate growing demands and ensure optimal performance under varying loads.

5. Full-Stack Development: Exploring the capabilities of Java for both front-end and back-end development, including integration with databases, to create a comprehensive solution for managing banking operations.

8 Future work scope

Subject: Future Ideas for Enhancing the Bank Management System

As we continue to refine and improve our Bank Management System, there are several ideas that we may not have been able to implement due to time limitations but hold promise for future development. These ideas aim to further enhance the user experience, reliability, and functionality of our system. Here are some suggestions:

1. Enhanced Security Measures:

- Implementation of biometric authentication methods such as fingerprint or facial recognition for an added layer of security during login and transaction authorization.

- Integration of blockchain technology to ensure immutable transaction records and enhance data integrity and security.

2. Personalized Customer Experience:

- Development of personalized dashboards for customers, providing insights into their account activities, spending patterns, and personalized financial recommendations.

- Introduction of chatbots powered by natural language processing (NLP) to assist customers with inquiries, account management tasks, and financial advice.

3. Advanced Analytics and Reporting:

- Integration with machine learning algorithms to analyze customer behavior, predict future trends, and offer proactive financial advice and product recommendations.

- Generation of comprehensive reports and visualizations to provide insights into bank performance, customer demographics, and market trends.

4. API Integration and Open Banking:

- Opening up APIs to third-party developers to foster innovation and allow for the creation of new financial services and applications that integrate seamlessly with our Bank Management System.

- Integration with external financial services such as investment platforms, insurance providers, and budgeting apps to offer a holistic financial management experience for customers.

5. Mobile Banking Enhancements:

- Development of a native mobile banking application with offline capabilities, allowing customers to access their accounts and perform transactions even in areas with limited or no internet connectivity.

- Implementation of location-based services and push notifications to provide relevant offers, alerts, and reminders to customers based on their geographic location and transaction history.

6. Accessibility Features:

- Implementation of accessibility features such as screen readers, voice commands, and high-contrast interfaces to ensure that the Bank Management System is usable by individuals with disabilities.

7. Gamification Elements:

- Introduction of gamification elements such as reward points, badges, and challenges to incentivize desired financial behaviors, encourage engagement, and foster a sense of achievement among customers.