Industrial Internship Report on

“Banking Information System”

Prepared by

KARTHIK P

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 to develop a prototype of a Banking Information System in Core Java that provides a working preview of the key functionalities of a real banking system.

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

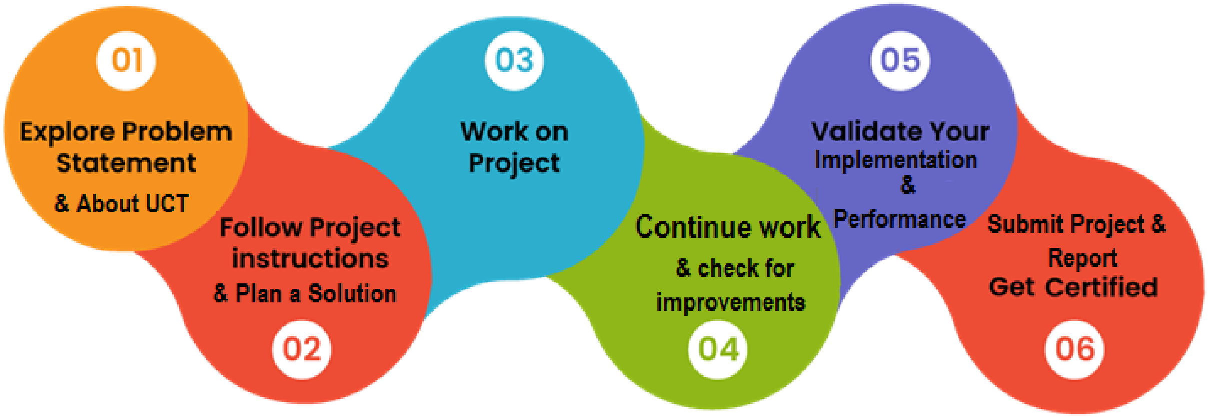
Over six weeks, I developed a Java console application for a Banking Information System. The process included initiation, core functionality implementation, testing, documentation, presentation, and reflection.

Internships are vital for practical learning, skill development, industry exposure, networking, and enhancing resumes.

The project aimed to create a user-friendly banking system with features like account management and transactions.

USC/UCT provided resources, mentorship, networking, exposure to industry practices, and potential career pathways.

The program planning involved needs assessment, objective setting, resource allocation, task breakdown, and monitoring mechanisms. Through structured planning and execution, the Banking Information System project was successfully completed within the given timeline.



Throughout this project, I improved my Java programming skills and learned to develop a console-based application. The experience taught me the importance of planning, testing, documenting, and being adaptable when facing challenges. I would like to extend my heartfelt thanks to upskill campus for their invaluable guidance and feedback.

To my juniors and peers: Stay curious, embrace challenges, collaborate, plan, document thoroughly, and seek feedback. These practices will help you grow and succeed.

Introduction

1.1 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.

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

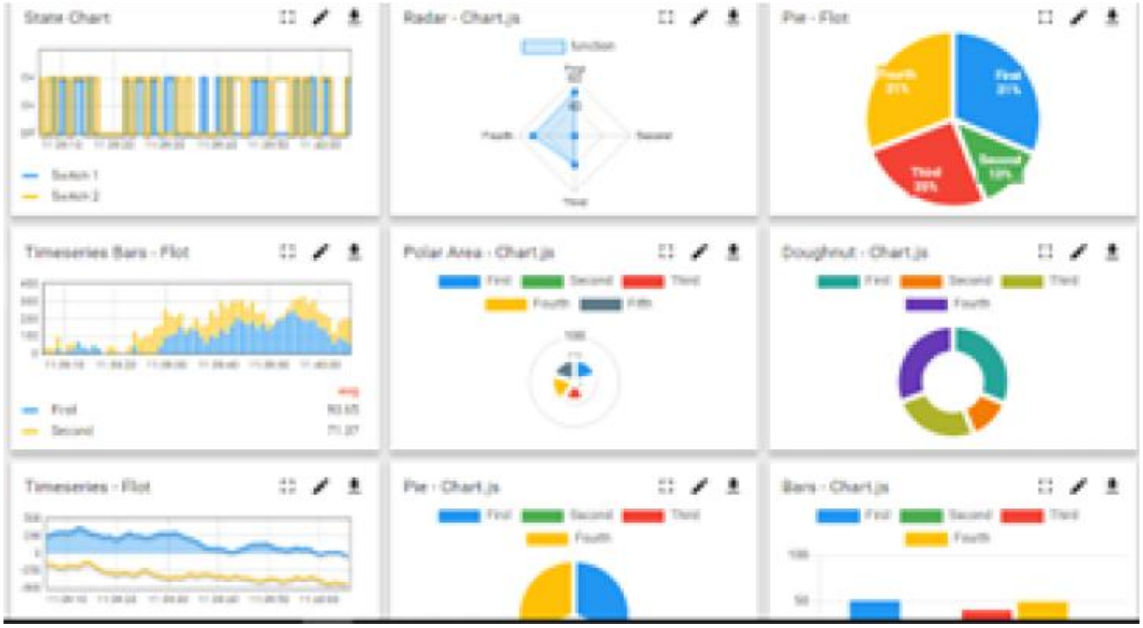
i. UCT IoT Platform (uct Insight)

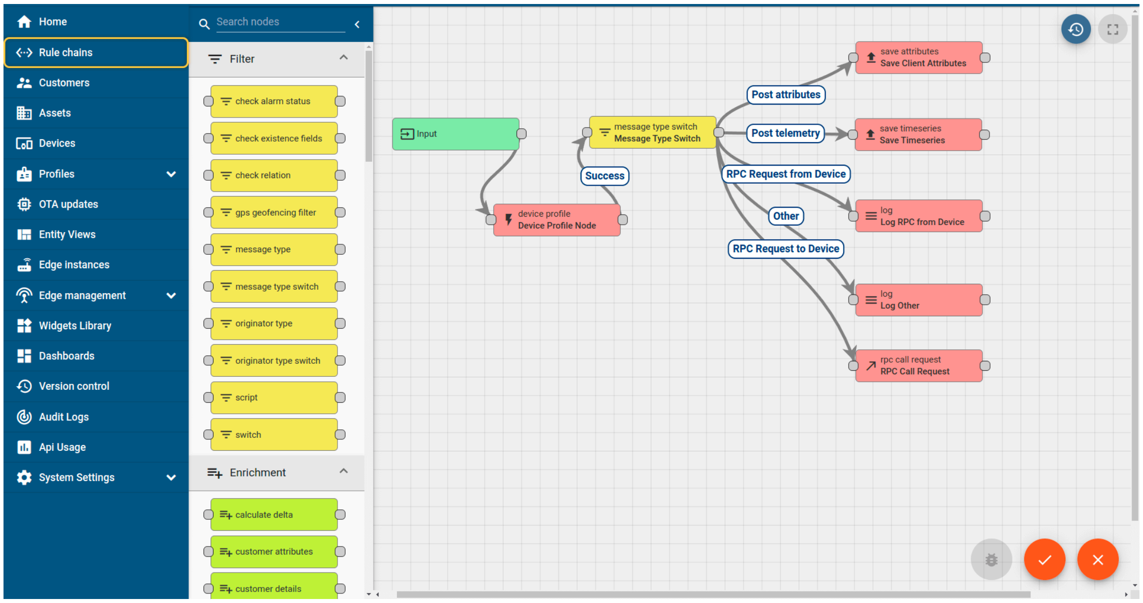
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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





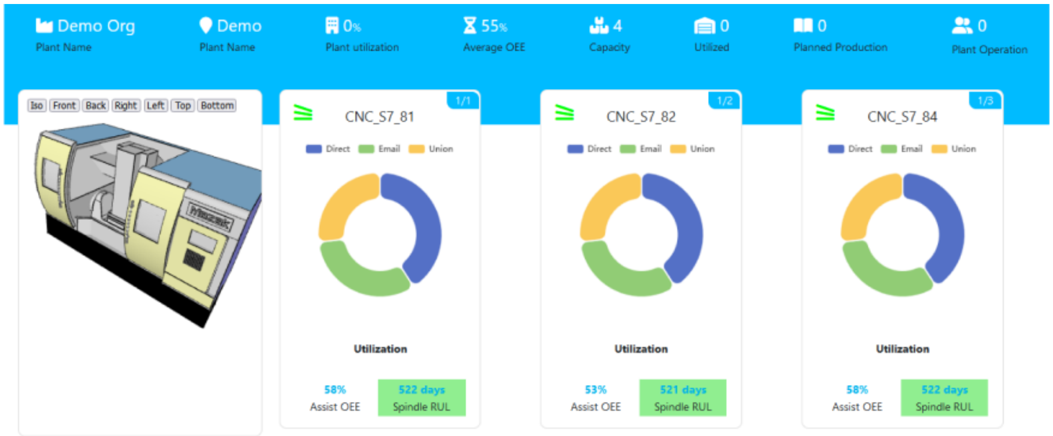
ii. Smart Factory Platform (FACTORY WATCH)

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.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

4
Capacity

0
Utilized

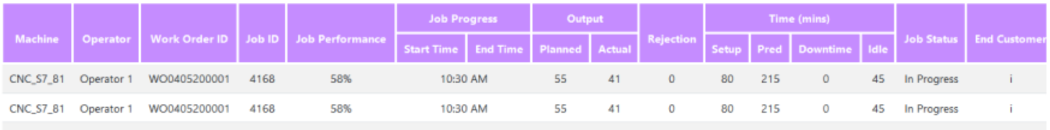
0
Planned Production

0
Plant Operation

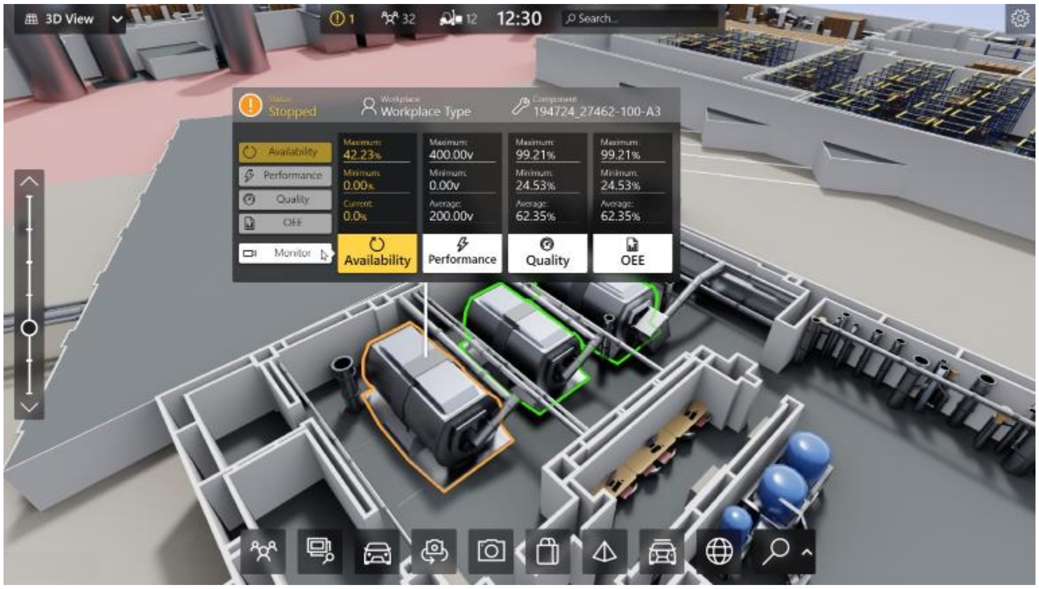
CNC_S7_81 (1/1)
Direct · Email · Union
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 (1/2)
Direct · Email · Union
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 (1/3)
Direct · Email · Union
Utilization
58% Assist OEE
522 days Spindle RUL



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



Status
Stopped

Workplace
Workplace Type

Component
194724_27462-100-A3

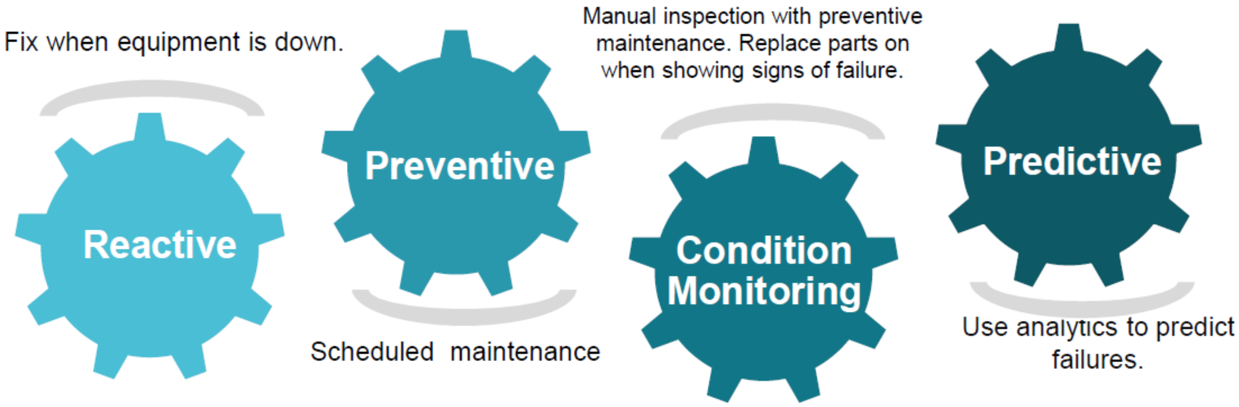
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

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

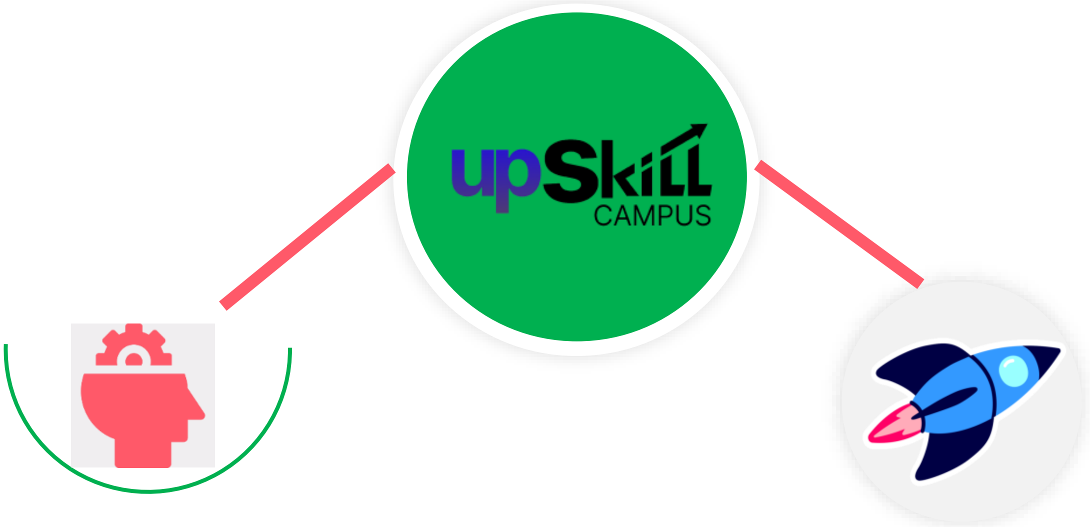
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

1.2 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.

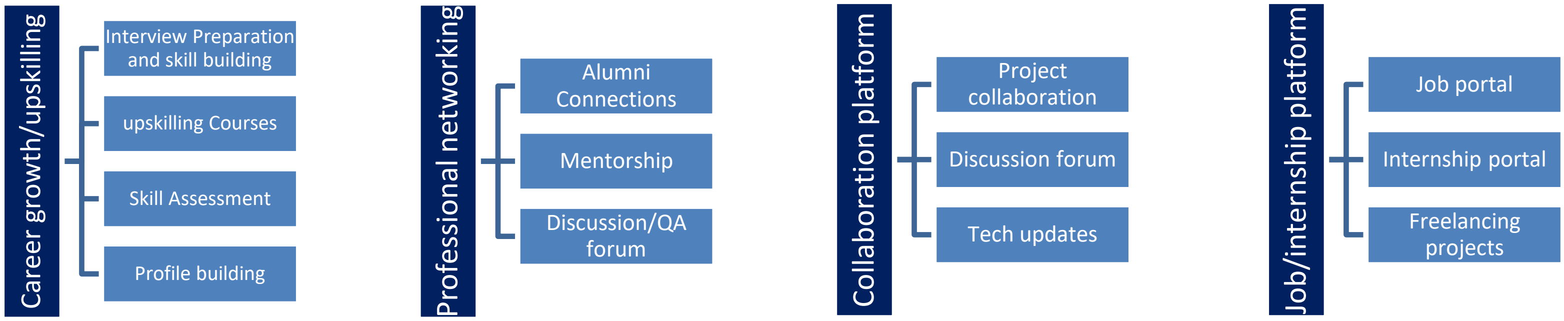
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/



Career growth/upskilling
- Interview Preparation and skill building
- upskilling Courses
- Skill Assessment
- Profile building

Professional networking
- Alumni Connections
- Mentorship
- Discussion/QA forum

Collaboration platform
- Project collaboration
- Discussion forum
- Tech updates

Job/internship platform
- Job portal
- Internship portal
- Freelancing projects

1.3 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.

1.4 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.

1.5 Reference

[1] Official Java Documentation

[2] Java Programming Tutorials

[3] Java Code Geek

2 Problem Statement

The project involves creating a Banking Information System using a Java console application. The main goals are to allow users to perform common banking tasks efficiently and securely. Key functionalities include:

User Registration:

- Users can sign up by providing their name, address, contact info, and an initial deposit.
- The system generates a unique account number for each new user.
- Registration includes checks to ensure information is accurate and complete.

Account Management:

- Users can view and update their account details, including personal information and balance.
- Security measures ensure user data is protected during updates.

Deposits and Withdrawals:

- Users can deposit money into their accounts, with the system validating the deposit amount and updating the balance.
- Users can withdraw money, with the system checking that the amount does not exceed the balance and updating the balance if the transaction is valid.
- The system provides feedback messages for successful or failed transactions.

Fund Transfers:

- Users can transfer money between accounts within the system.
- The system validates that both the sender and recipient accounts exist and that the sender has enough funds. Balances are updated securely, and users receive transaction confirmations.

Account Statements:

- Users can view their transaction history, including deposits, withdrawals, and transfers.
- Account statements are presented clearly, showing transaction details like dates, types, amounts, and balances.

3 Existing and Proposed solution

Provide summary of existing solutions provided by others, what are their limitations?

What is your proposed solution?

What value addition are you planning?

3.1 Code submission (Github link)

https://github.com/Karthikdecode/upskillcampus_BANKINFORMATION_PROJECT/blob/main/Banking%20Information%20System.java

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

4 Proposed Design/ Model

The Banking Information System is a console-based application in Java designed to handle fundamental banking operations. The design flow consists of a structured sequence of stages from user interaction to the final outcome.

Start: User Interaction Initialization

1. **Main Menu Display:**

- The application begins by displaying a main menu, which includes options such as user registration, account management, deposits, withdrawals, fund transfers, account statements, and exit.
- Users interact with the application by selecting an option corresponding to the operation they wish to perform.

Intermediate Stages: Core Functional Modules

a. User Registration:

- Input Collection: Prompts the user to enter their name, address, contact information, and initial deposit amount.
- Account Creation: Creates a new BankUser object with the provided information and assigns a unique account number.
- Confirmation: Confirms successful registration and displays the new account number.

b. Account Management:

- Account Lookup: The user enters their account number to view account details.
- Display Details: Retrieves and displays the user's personal information and current balance.

c. Deposits:

- Account Lookup: The user provides their account number.
- Amount Entry: The user inputs the deposit amount.
- Validation and Update: Validates the deposit amount and updates the balance.
- Confirmation: Confirms successful deposit and shows the updated balance.

d. Withdrawals:

- Account Lookup: The user enters their account number.
- Amount Entry: The user inputs the withdrawal amount.
- Validation and Update: Checks if the amount is within the current balance and updates the balance if valid.
- Confirmation: Confirms successful withdrawal and shows the updated balance.

e. Fund Transfers:

- Sender and Recipient Lookup: The user provides the account numbers of both the sender and the recipient.
- Amount Entry: The user inputs the transfer amount.
- Validation and Transfer: Validates both accounts and processes the transfer by updating both balances.
- Confirmation: Confirms successful transfer.

f. Account Statements:

- **Account Lookup**: The user enters their account number.
- **Display Statement**: Displays the account's transaction history and details.

Final Outcome: User Exit

- **Exit Option**: Users can choose to exit the application from the main menu.
- **Save and Secure**: The system ensures all data is saved and secured before closing.

4.1 High Level Diagram (if applicable)

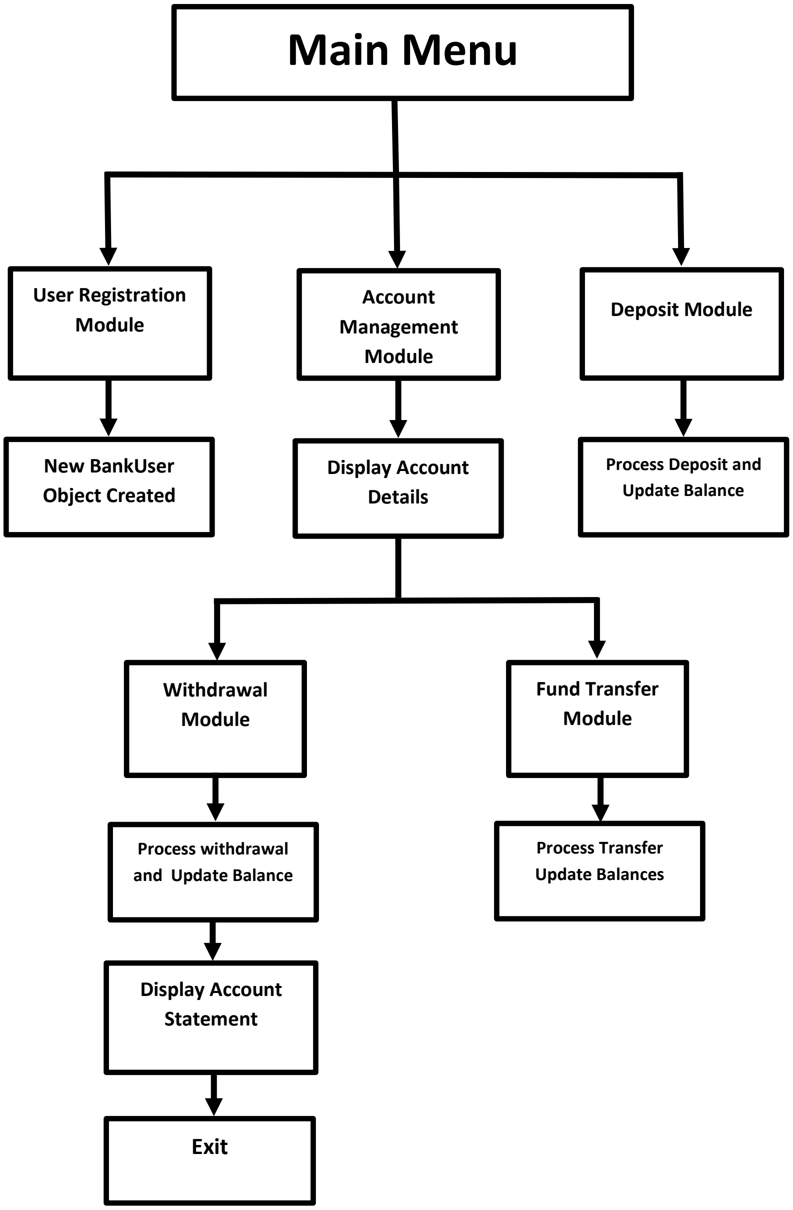
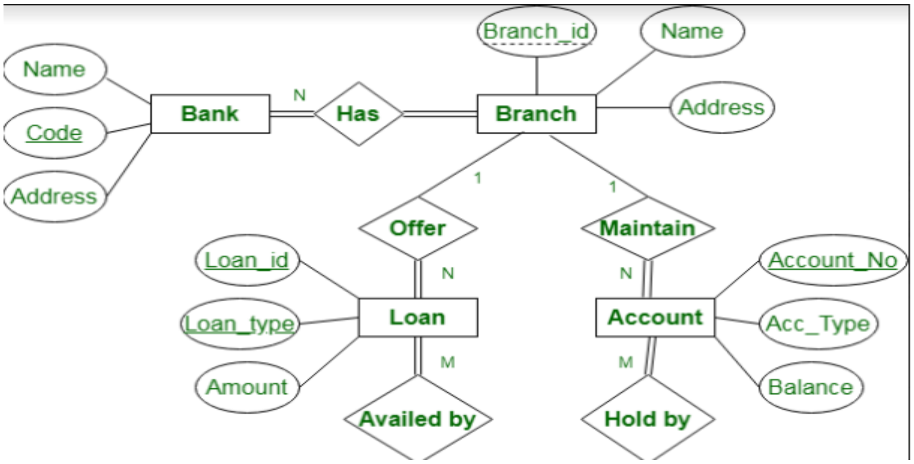

Main Menu
User Registration Module
Account Management Module
Deposit Module
New BankUser Object Created
Display Account Details
Process Deposit and Update Balance
Withdrawal Module
Fund Transfer Module
Process withdrawal and Update Balance
Process Transfer Update Balances
Display Account Statement
Exit

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

4.2 Low Level Diagram (if applicable)



5 Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Here we need to first find the constraints.

How those constraints were taken care in your design?

What were test results around those constraints?

Constraints can be e.g. memory, MIPS (speed, operations per second), accuracy, durability, power consumption etc.

In case you could not test them, but still you should mention how identified constraints can impact your design, and what are recommendations to handle them.

5.1 Test Plan/ Test Cases

It is System Test Plan for Interment Banking System, console application, provides access to Account holders and guest users from anywhere in the world. It has two interfaces one is Admin interface another is User interface. Admin can be accesses by Bank authorized users, user interface can be accessed by Bank account holders and guest users. The propose of the system (Application) is to provide bank information and services console , Bank account holders can get Banking services from any ware, without visiting the Bank branches.

5.2 Test Procedure

5.3 Performance Outcome

6 My learnings

Working on this banking system project has provided me with several valuable lessons and insights, both in terms of technical skills and overall project management. Here's a summary of my learnings and how they will contribute to my career growth:

1. **Object-Oriented Programming (OOP) Principles**
   - **Encapsulation**: I learned how to encapsulate data and methods within a class to create a modular and maintainable code structure. This is evident in how the `BankUser` class encapsulates user data and provides methods to interact with it.
   - **Static Variables and Methods**: Understanding the use of static variables and methods, such as `accountNumberCounter` to ensure unique account numbers across all instances.
   - **Separation of Concerns**: By creating separate classes like `BankUser` and `WithdrawalModule`, I practiced separating different functionalities to enhance code readability and maintainability.
2. **User Interaction and Input Handling**
   - **Scanner Class**: I gained experience using the `Scanner` class for handling user input in a console-based application. This helped me understand the importance of input validation and error handling to ensure a smooth user experience.
   - **Control Flow**: Implementing the main menu with a `do-while` loop and switch-case statements improved my understanding of control flow structures in Java.
3. **Basic Banking Operations Implementation**
   - **Deposit and Withdrawal Logic**: Implementing methods for deposit and withdrawal operations reinforced my understanding of basic financial transactions and the importance of checking conditions like balance sufficiency.
   - **Fund Transfer**: Developing the fund transfer feature taught me about handling multiple operations sequentially and ensuring data consistency.
4. **Code Optimization and Efficiency**
   - **Linear Search**: I realized the limitations of linear search in large datasets, which highlighted the need for more efficient data structures like `HashMap` for faster lookups.

- **Fixed-Size Arrays**: Using a fixed-size array for storing users pointed out scalability issues, leading me to explore dynamic data structures like `ArrayList`.

5. **Scalability and Data Persistence**
   - **Scalability Challenges**: The project underscored the importance of scalable solutions in software development, motivating me to learn about more advanced data structures and database integration for real-world applications.
   - **Data Persistence**: The lack of data persistence in the project highlighted the need for implementing file I/O or database connectivity to save user data across sessions.
6. **User Experience and Interface Design**
   - **User-Friendly Interfaces**: Designing a simple console-based interface taught me the importance of clear prompts and feedback messages to guide users through various operations.
   - **Future GUI Development**: This project has motivated me to explore graphical user interfaces (GUIs) for more sophisticated and user-friendly applications.

6.1.1 Career Growth Impact

- **Enhanced Coding Skills**: This project has significantly improved my coding skills, particularly in Java. The principles and practices I applied here are transferable to any object-oriented programming language.
- **Problem-Solving Abilities**: Tackling various challenges in the project, such as handling user input and managing account operations, has honed my problem-solving abilities and logical thinking.
- **Project Management**: Working on a project from start to finish has given me a better understanding of project management, including planning, implementation, testing, and iteration.
- **Foundation for Advanced Topics**: The learnings from this project provide a solid foundation for exploring more advanced topics like data structures, algorithms, database management, and GUI development.
- **Real-World Application**: Understanding the practical aspects of building a banking system prepares me for real-world software development scenarios, making me a more competent and competitive candidate in the job market.
- **Confidence and Motivation**: Successfully completing this project has boosted my confidence and motivated me to take on more complex projects in the future.

Overall, the knowledge and experience gained from this project will be instrumental in my career growth, helping me become a more proficient software developer and opening up opportunities for working on more complex and impactful projects.

4o

You should provide summary of your overall learning and how it would help you in your career growth.

7 Future work scope

While the current banking system project successfully implements several core functionalities, there are numerous enhancements and additional features that can be explored in the future to make it more robust, user-friendly, and scalable. Here are some ideas for future work:

1. **Data Persistence**
   - **File I/O**: Implement file input/output operations to save user data and transaction history to a file, ensuring data is preserved between sessions.
   - **Database Integration**: Integrate a database management system (DBMS) like MySQL or SQLite to handle data storage, retrieval, and management more efficiently.
2. **Enhanced User Interface**
   - **Graphical User Interface (GUI)**: Develop a GUI using JavaFX or Swing to replace the console-based interface, providing a more intuitive and user-friendly experience.
   - **Web Interface**: Create a web-based interface using Java Servlet, JSP, or a framework like Spring Boot, enabling users to access the banking system via a web browser.
3. **Security Improvements**
   - **User Authentication**: Implement a login system with user authentication to secure access to accounts, including password hashing and session management.
   - **Encryption**: Use encryption techniques to protect sensitive data such as account details and transaction information.
4. **Advanced Banking Features**
   - **Transaction History**: Maintain and display a detailed transaction history for each user, including deposits, withdrawals, and fund transfers.
   - **Account Statements**: Provide downloadable account statements in formats like PDF or CSV for users to review their financial activities.
   - **Interest Calculation**: Implement interest calculation for savings accounts and provide periodic interest updates.
5. **Scalability and Performance**
   - **Optimized Data Structures**: Replace the fixed-size array with dynamic data structures like `ArrayList` or `HashMap` for better performance and scalability.
   - **Concurrency Handling**: Introduce multi-threading to handle multiple users and transactions simultaneously, ensuring data consistency and improving performance.

You can put some ideas that you could not work due to time limitation but can be taken in future