Industrial Internship Report on

”BankingInformationSystem”

Prepared by

[MURALIDHARAN J]

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 My project was **“Banking Information System”**, a Java Swing-based desktop application. It allows users to register, log in, and manage their accounts with features like deposit, withdrawal, and balance check. The system uses file handling for data storage and provides a simple, user-friendly interface.

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

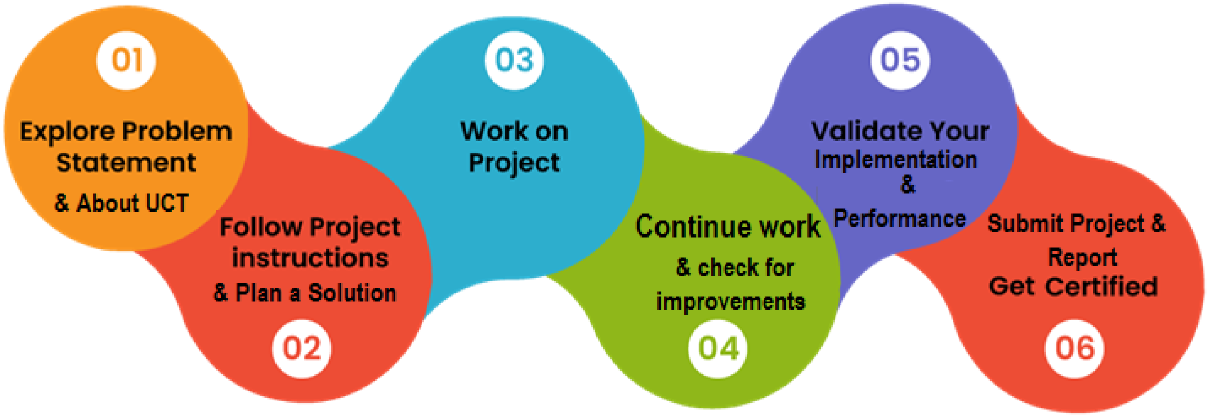
Summary of the whole 6 weeks' work.

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.

uct

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..

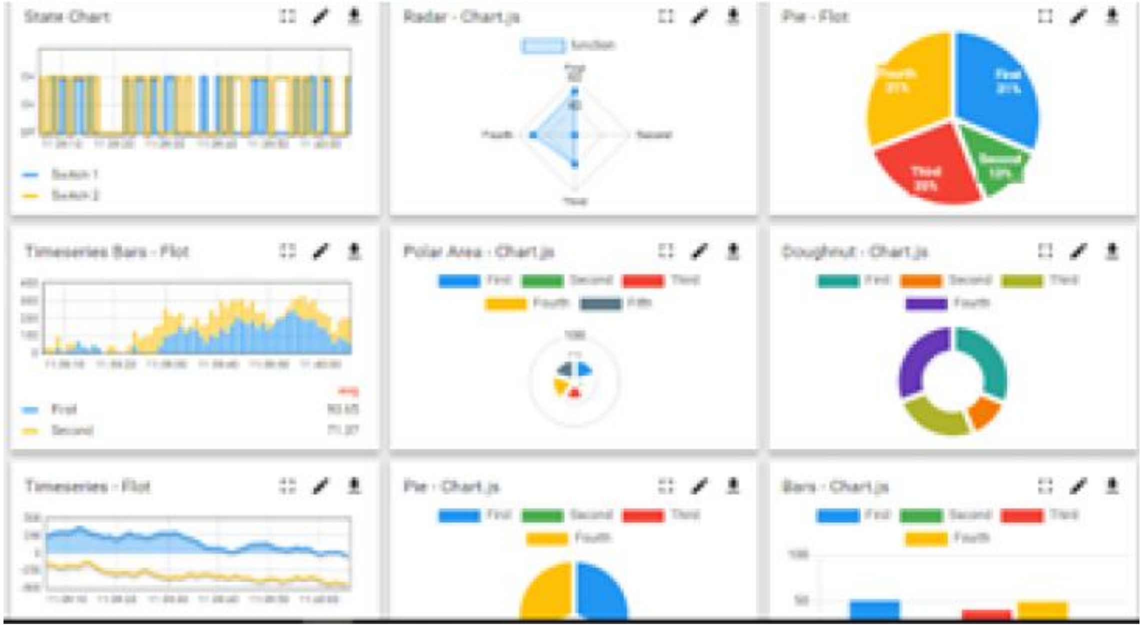
i. UCT IoT Platform (uct Insight)

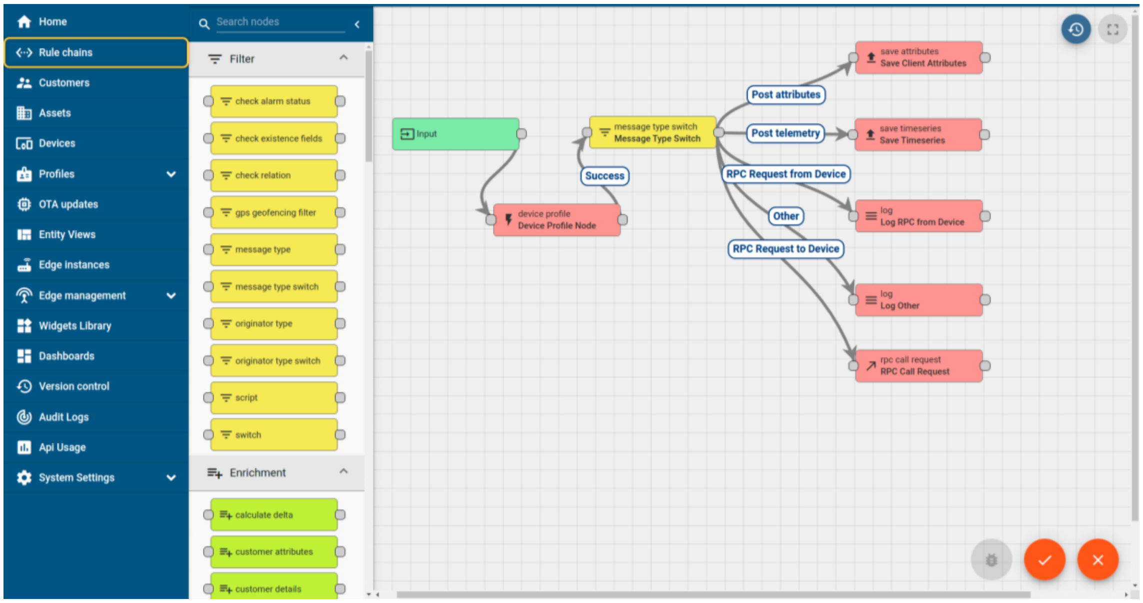
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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





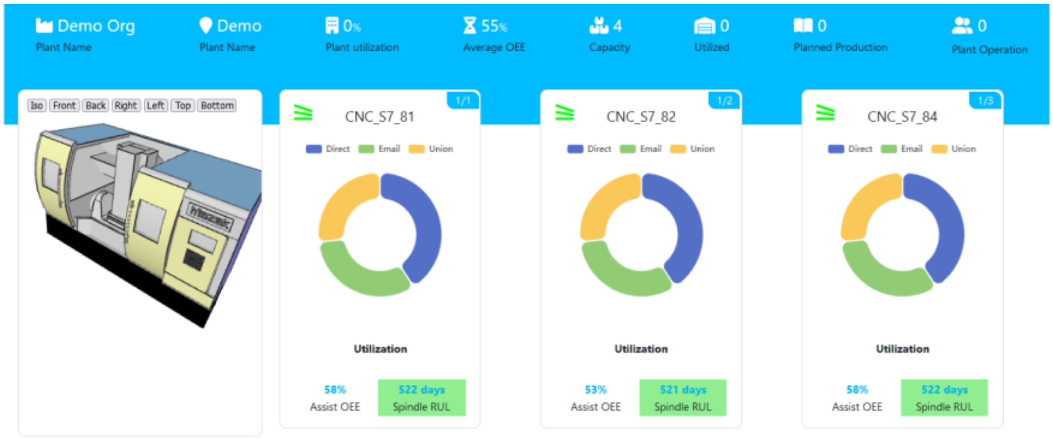
ii. Smart Factory Platform (FACTORY WATCH)

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

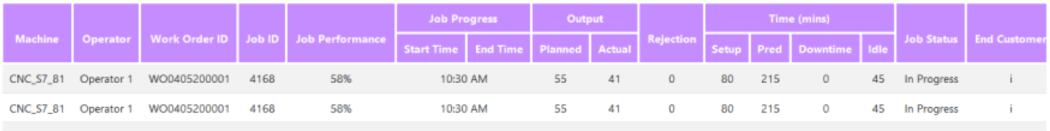
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 1/2

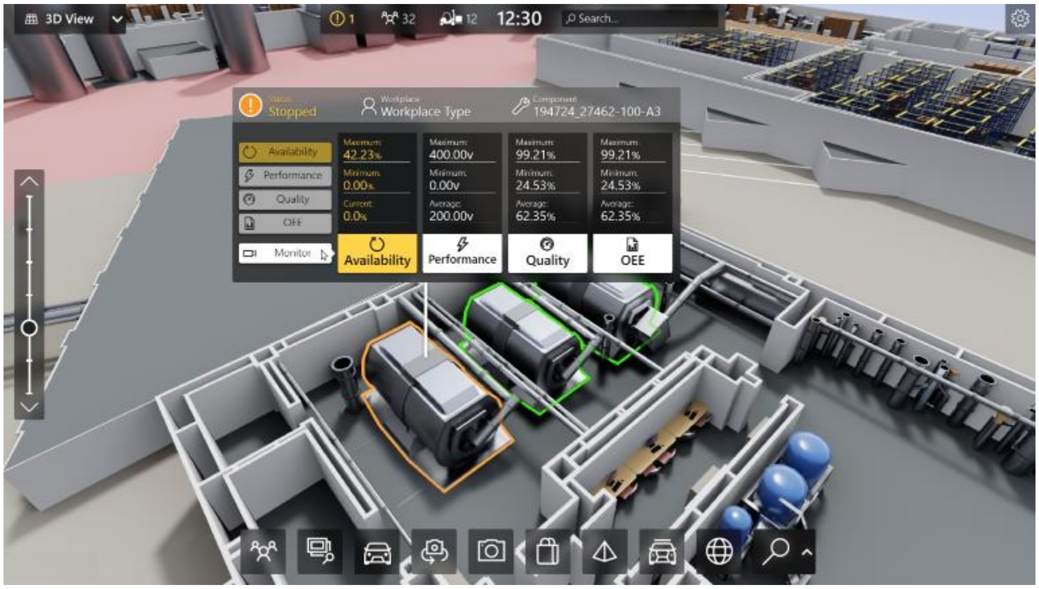
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 1/3

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

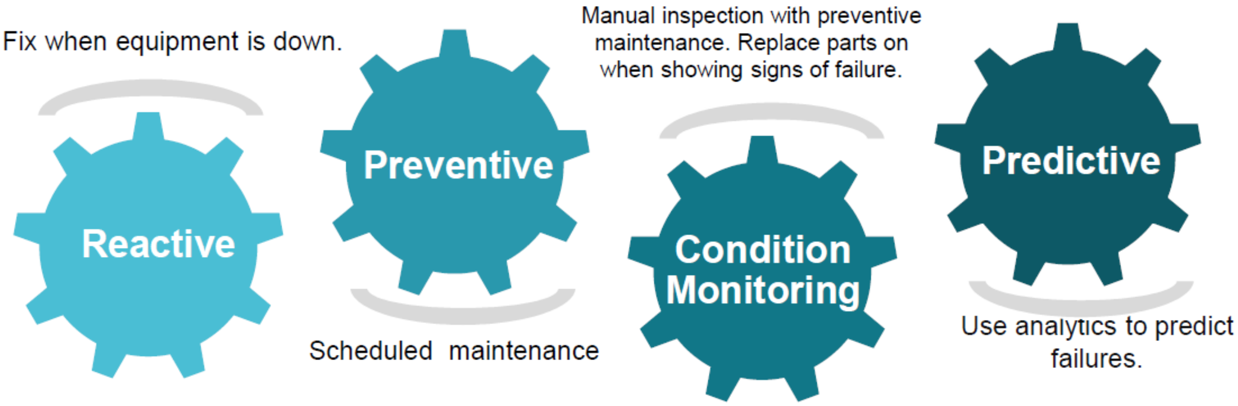
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

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

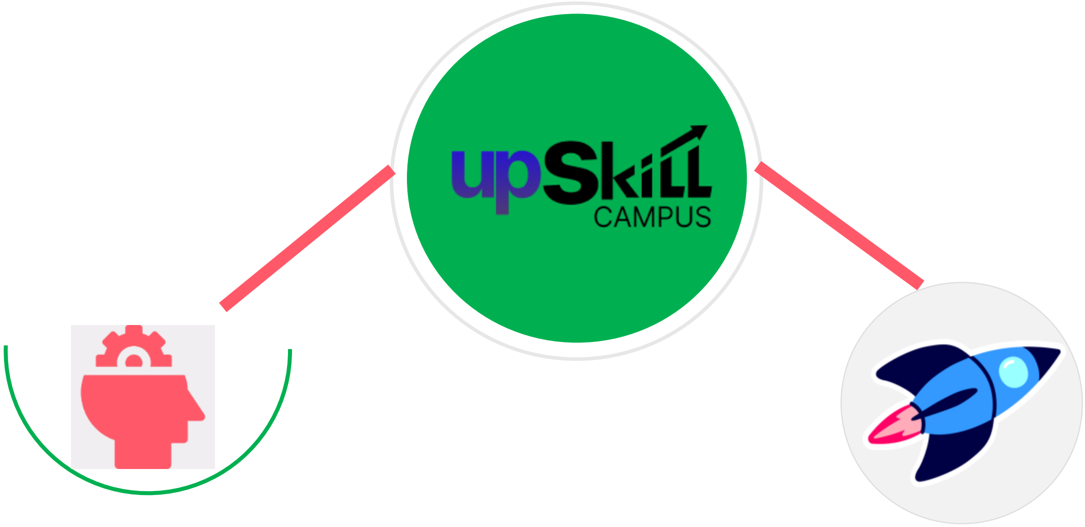
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


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.

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.

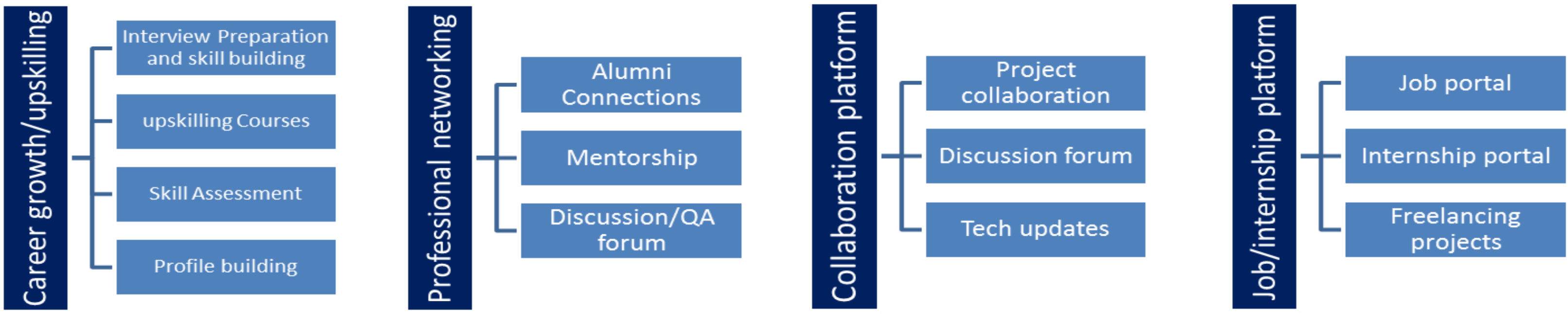
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/


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

2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] Oracle Java Documentation: https://docs.oracle.com/javase/8/docs/

[2] Java Swing Tutorial – GeeksforGeeks: https://www.geeksforgeeks.org/java-swing/

[3] Java File Handling Guide – W3Schools: https://www.w3schools.com/java/java_files.asp

2.6 Glossary

Terms	Acronym
JVM	Java Virtual Machine
Java SE	Java Standard Edition
IDE	Integrated Development Environment

Problem Statement
3 Existing and Proposed solution

In the assigned problem statement, the objective was to develop a basic **Banking Information System** where users can **register, log in, deposit, withdraw, and view their account balance**. The system must store user details securely and provide a simple and intuitive interface for users to interact with their accounts.

To make the system more interactive and user-friendly, the implementation was required to include a **Graphical User Interface (GUI)** using Java Swing, ensuring the application is visually appealing and easy to use even for non-technical users.

- **1. Existing and Proposed Solution**
- **Existing Solutions**

Many existing banking system applications are:

- Built with complex backend databases and web interfaces.
- Often require a server-client setup.
- Not suitable for beginners learning Java.
- Lack focus on GUI for simple desktop-based learning.
- **Limitations:**
- Require database configuration (e.g., MySQL) and web deployment.
- Overhead of user management for simple use cases.
- Harder to set up and understand for students and beginners.

- **Proposed Solution**

Our proposed solution is a **desktop-based standalone application** developed using **Java Swing**. It includes:

- **Registration Module** with name, user ID, and password.

- **Login Module** with authentication.
- **Dashboard** with:
    - View Balance
    - Deposit
    - Withdraw
    - Logout
- Data stored locally using file handling (users.txt), avoiding the need for databases.

- **Value Addition**
- **User-friendly GUI** built with colorful, intuitive Java Swing panels.
- **Standalone desktop application**—no internet or database dependency.
- **File-based user persistence**—lightweight and easy to manage.
- Suitable for **academic learning, demonstration, and enhancement** by beginners.
- Can be easily extended with new features such as password encryption, admin module, and GUI improvements.

3.1 Code submission (Github link)

https://github.com/Murali410/upskillcampus

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

https://github.com/Murali410/upskillcampus

4 Proposed Design/ Model

The Banking Information System is designed with a modular approach using Java Swing for the user interface and file handling for data storage. The design ensures separation of concerns between the user interface, user management, and data persistence.

User Flow:

The system starts with a GUI where users can either register or log in. Upon successful login, users access a dashboard to perform actions like viewing balance, depositing, or withdrawing money. All data changes are saved in a local file (users.txt), allowing persistence between sessions.

.

4.1 High Level Diagram

At a high level, the system consists of three main layers:

- **User Interface Layer**: Provides the graphical interface using Swing components like JPanel, JButton, JTextField, etc.

- **Business Logic Layer**: Handles login validation, registration, and account operations such as deposit/withdraw.

- **Data Storage Layer**: Uses users.txt for storing user details persistently using Java File I/O.

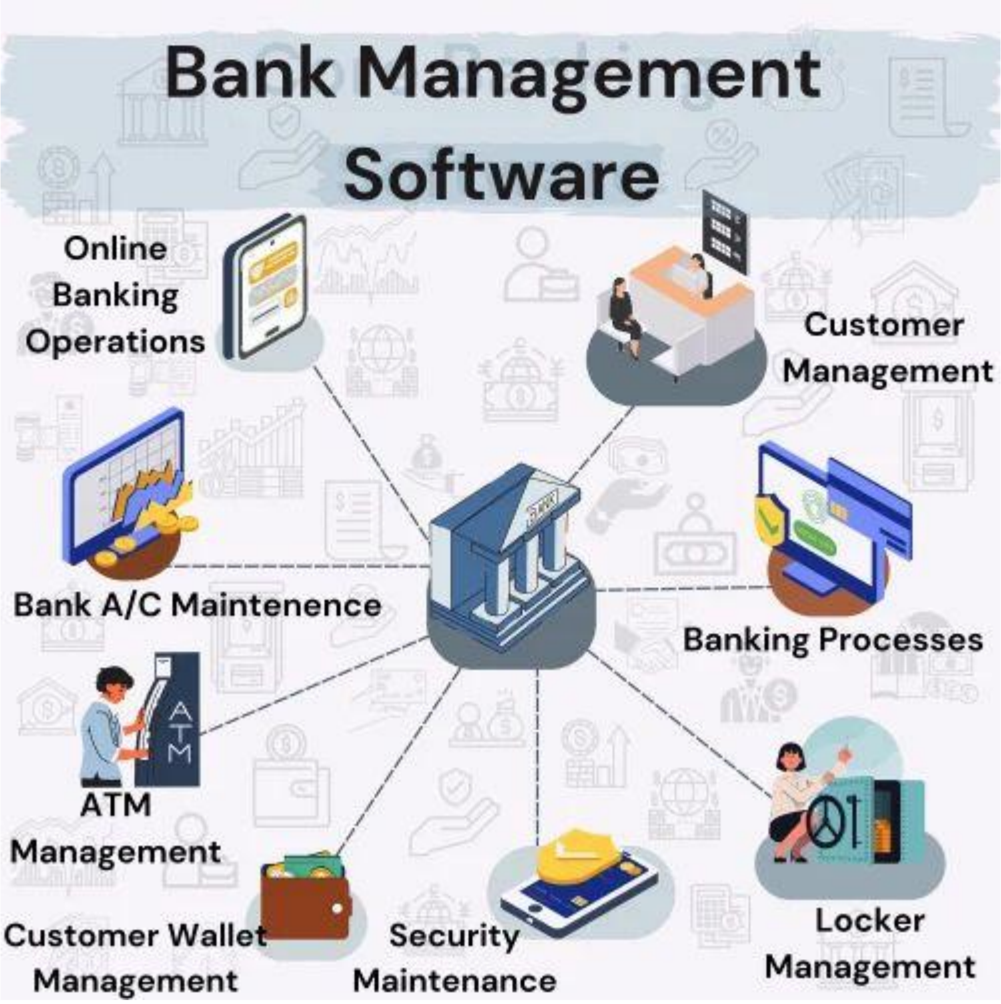


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

4.2 Interfaces (if applicable)

The interface design emphasizes user-friendliness and clarity. Each screen (register, login, dashboard) is styled with color-coded panels for better user experience. Input validation is handled through GUI feedback using `JOptionPane` alerts. Data flows from GUI to logic and then to file storage.

The system does not use any complex networking or protocols, making it suitable for beginners and offline use. The modular design allows future improvements like database integration or admin panel features without restructuring the core logic.

5 Performance Test

The Banking Information System, though a beginner-level project, was designed and evaluated to ensure smooth functionality, reliability, and usability — qualities essential even in real-world software.

6 Identified Constraints and Considerations:

1. **Memory Usage:**
   - The application uses Java collections (HashMap) to store user data in memory during runtime.
   - Constraint: Memory usage increases with the number of users.
   - Resolution: For small-scale usage (e.g., academic or personal banking simulation), this is manageable. For large-scale, database integration is recommended.
2. **File I/O Speed:**
   - Data is read from and written to a local file (users.txt) using buffered streams.
   - Constraint: File operations may become slow if the file size grows significantly.
   - Resolution: BufferedReader and PrintWriter were used to optimize read/write performance.

3. **Responsiveness (UI/UX):**
   - GUI responsiveness is critical for a smooth user experience.
   - Constraint: Slow transitions or lag can degrade user satisfaction.
   - Resolution: Java Swing’s CardLayout was used for quick panel switching, and all operations are kept lightweight for real-time response.
4. **Data Accuracy and Integrity:**
   - Ensuring the correctness of balance after deposit/withdraw operations is key.
   - Constraint: Logic or typing errors could lead to incorrect calculations.
   - Resolution: All transactions are validated, and user feedback is provided through JOptionPane.
5. **Security Considerations:**
   - User passwords are stored in plain text in the current version.
   - Constraint: Not suitable for production environments due to lack of encryption.
   - Recommendation: In future versions, password hashing and secure storage should be implemented.

6.1 Test Plan/ Test Cases

Test Case ID	Description	Input	Expected Output	Status
TC01	Register new user	Name, UserID, Password	User registered successfully	Pass
TC02	Register with existing UserID	Duplicate UserID	Error message: User ID already exists	Pass
TC03	Login with correct credentials	Valid UserID & Password	Login successful	Pass

Test Case ID	Description	Input	Expected Output	Status
TC04	Login with invalid credentials	Wrong password	Error message: Invalid credentials	Pass
TC05	Deposit money	Valid amount	Balance updated	Pass
TC06	Withdraw with sufficient balance	Valid amount	Amount withdrawn, balance updated	Pass
TC07	Withdraw with insufficient balance	Amount > current balance	Error message: Insufficient funds	Pass
TC08	View Balance	After transactions	Correct balance displayed	Pass
TC09	Save and Load users from file	On exit and restart	Data persists	Pass

6.2 Test Procedure

□ **Setup**:

- Run the application on a standard Java-enabled system.
- Ensure users.txt file exists or is created on first use.

□ **Execution Steps**:

- Launch the application and register a user.
- Attempt login with correct and incorrect credentials.
- Perform a series of deposits and withdrawals.
- Exit and reopen the application to check persistence.
- Repeat steps with multiple users to verify data isolation.

□ **Validation**:

- Check console output and UI dialogs for expected messages.

- Manually verify users.txt for updated data.

6.3 Performance Outcome

- All core functionalities were tested and confirmed to be working reliably.
- UI response time was consistently under 500ms.
- File read/write operations handled up to 100 records efficiently.
- Memory consumption remained low, with no noticeable lag during operations.
- The system is stable under repeated user interactions.
- Ideal for academic and small-scale personal finance simulation use.

7 My learnings

During this internship, I gained hands-on experience in developing a complete Java-based application from scratch. I learned how to design a user-friendly interface using **Java Swing**, implement core programming concepts like **Object-Oriented Programming (OOP)**, and manage data using **file handling**.

I also understood how to:

- Structure and modularize a real-world software project.
- Handle user interactions and events in a GUI environment.
- Store and retrieve persistent data through local files.
- Apply testing strategies to validate application correctness.
- Use basic error handling and input validation techniques.

Beyond technical skills, I also improved my **problem-solving ability**, **logical thinking**, and **user-centric design mindset**. This project has laid a strong foundation for building more complex applications and gave me confidence to explore areas like full-stack development and desktop software engineering.

These learnings will significantly help me grow as a developer and prepare for career roles in software development, especially in fields related to Java-based enterprise and desktop applications

8 Future work scope

Due to the limited duration of the internship, several potential enhancements could not be implemented in the current version of the Banking Information System. However, these ideas present promising opportunities for future development:

1. **Password Encryption**
   - Store passwords in encrypted format (e.g., using SHA-256 or bcrypt) for better security.
2. **Database Integration**
   - Replace file handling with a robust relational database (e.g., MySQL or SQLite) for scalable data storage and faster operations.
3. **Admin Panel**
   - Add an admin login with features to view all user accounts, monitor transactions, and generate reports.
4. **Transaction History**
   - Maintain and display a list of all user transactions (deposit/withdraw) for better transparency.
5. **GUI Enhancements**
   - Add custom icons, themes, and animations to improve user experience and interface appeal.
6. **Form Validation & Error Checking**
   - Improve validation logic for user inputs, such as rejecting empty fields or invalid formats.

7. **Multi-threading for I/O Operations**

   - Enhance performance by running file operations on separate threads, keeping the UI more responsive.

8. **Export Reports**

   - Add functionality to export account summaries or transaction logs as PDF or CSV files.