

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

**Prepared by**

**[K.Chetan Varma]**

### *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 (Banking Information 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 .....	3
2	Introduction .....	4
2.1	About UniConverge Technologies Pvt Ltd .....	4
2.2	About upskill Campus .....	8
2.3	Objective .....	9
3	Problem Statement .....	11
4	Existing and Proposed solution .....	12

## 1 Preface

This project represents the culmination of six weeks of dedicated effort to build a fully functional banking system using Java. The system is designed to simulate real-world banking operations, including user registration, login, deposit, withdrawal, fund transfer, and account statement generation. My primary goal was to create a system that is both user-friendly and secure, providing a robust platform for managing basic banking transactions.

Throughout this project, I focused on integrating core principles of object-oriented programming, such as encapsulation, inheritance, and polymorphism, to ensure that the system is scalable and maintainable. Additionally, I incorporated a simple frontend interface using HTML, CSS, and JavaScript to demonstrate how the backend logic can be extended to a user-facing application.

**Week 1-2: System Design and Planning** During the initial weeks, I focused on designing the system architecture. I mapped out the key functionalities that the banking system would require, such as user management, account handling, and transaction tracking. I also decided on the technologies and tools I would use, settling on Java for the backend due to its reliability and scalability.

**Week 3: Implementing Core Features** In the third week, I began coding the core features of the system. This included creating the `User` class to manage user data and transactions and the `BankingSystem` class to handle operations like user registration, login, and account management. I also implemented the `AccountNumber` class to generate unique account numbers for users.

**Week 4: Refinement and Security Enhancements** With the basic functionality in place, I spent the fourth week refining the code and adding security features. This included ensuring that the system securely handles user passwords and prevents actions like overdrawing from an account. I also implemented error handling to manage exceptional cases, such as insufficient funds during withdrawals or transfers.

**Week 5: Frontend Development** In the fifth week, I turned my attention to the frontend. I developed a basic interface using HTML, CSS, and JavaScript that allows users to interact with the banking system. This interface is designed to be simple and intuitive, providing users with easy access to the system's functionalities.

**Week 6: Testing and Final Adjustments** The final week was dedicated to testing the system thoroughly. I performed unit tests to ensure that all functions work as expected and conducted user acceptance testing to verify that the system meets the requirements. I also made final adjustments to the code and interface, ensuring that the project was polished and ready for presentation.

## 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.



#### i. UCT IoT Platform( )

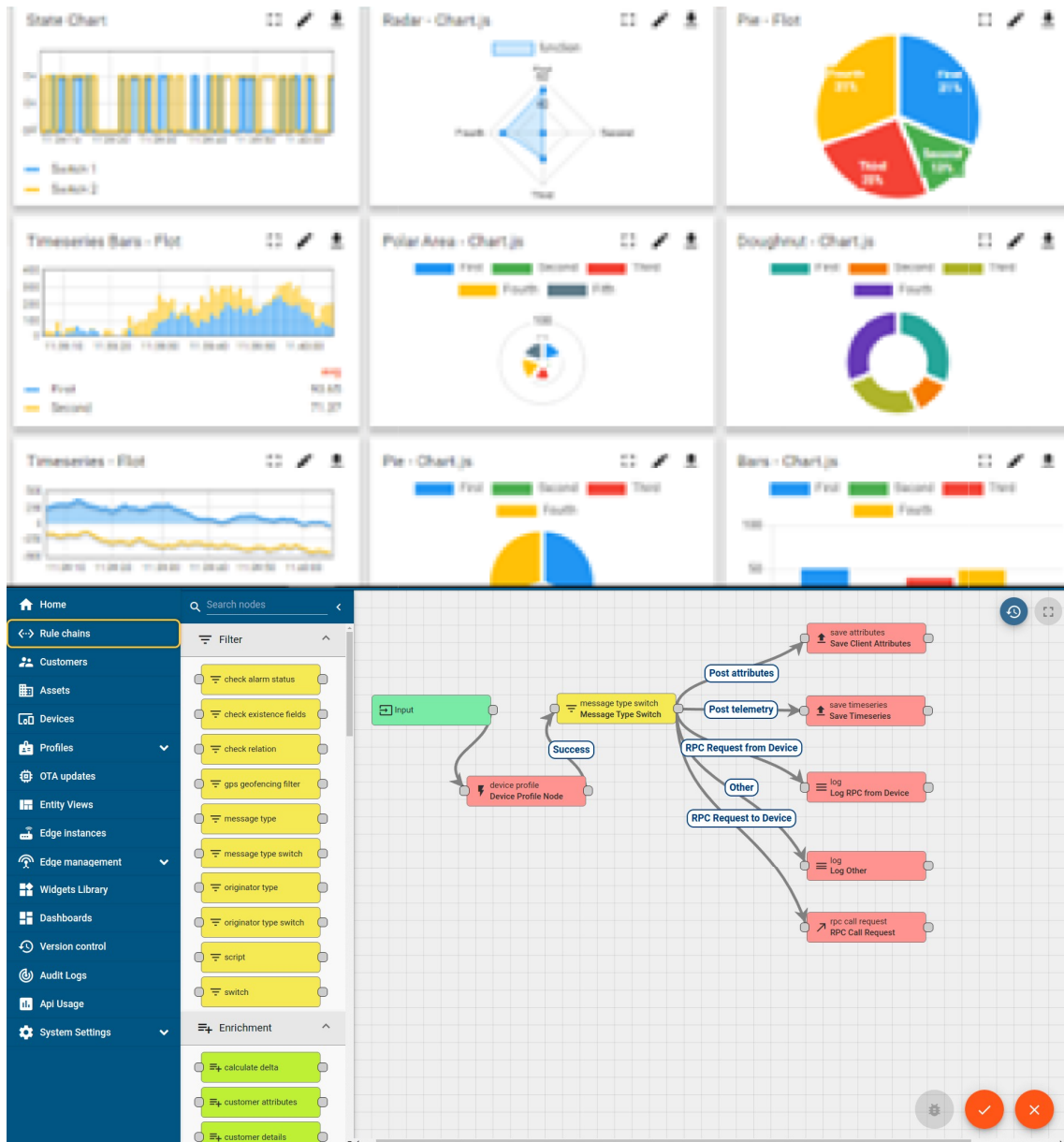
**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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.
- to unleash the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they want to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (min)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Prod	Downtime	Idle		
CNC_S7_81	Operator 1	WO040520C001	4168	58%	10:30 AM		55	41	0	30	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO040520C001	4168	58%	10:30 AM		55	41	0	30	215	0	45	In Progress	i

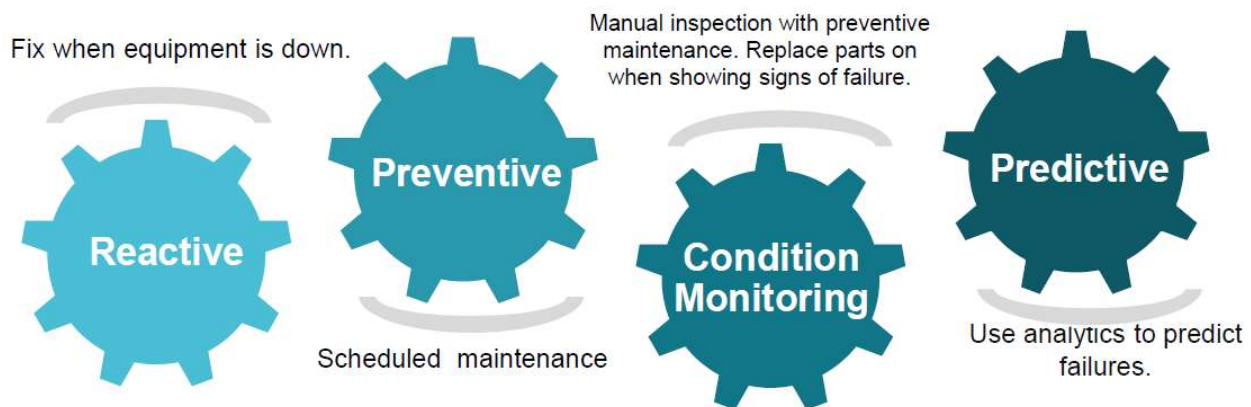


### iii. based Solution

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.

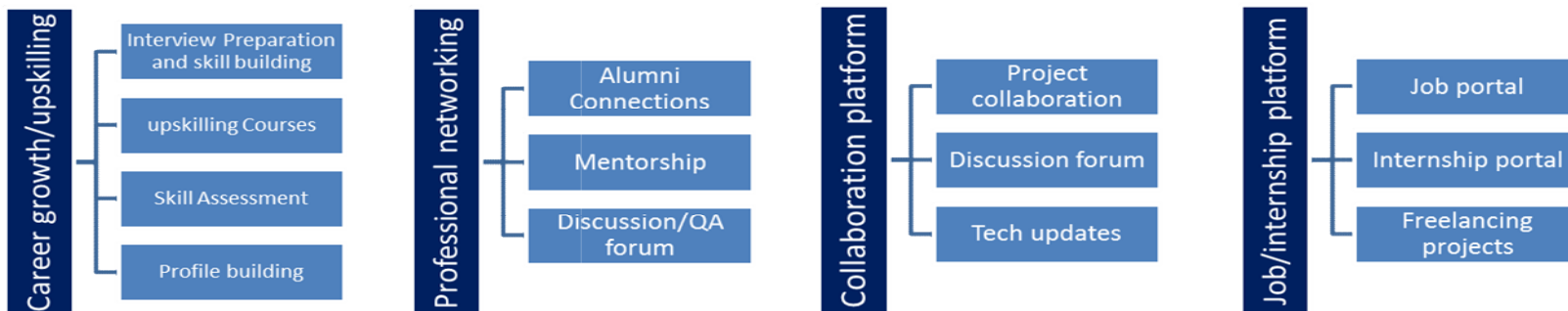


## 1.2 About upskill Campus (USC)

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.





### 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.

## 2 Problem Statement

In the assigned problem statement

In today's digital age, the need for secure and efficient banking systems is paramount. Banks must provide customers with the ability to manage their accounts, conduct transactions, and access services quickly and safely. However, many existing systems are either too complex for users to navigate or lack sufficient security features, leading to potential vulnerabilities.

The challenge is to design and implement a banking system that balances ease of use with robust security. The system should allow users to perform essential banking functions such as:

- Registering a new account with an initial deposit.
- Logging into their account securely.
- Managing account details such as personal information.
- Conducting financial transactions, including deposits, withdrawals, and transfers.
- Viewing account statements to track transaction history and account balance.

The system should also ensure that:

1. **Security:** User data, particularly sensitive information like passwords and account balances, must be securely managed to prevent unauthorized access.
2. **Usability:** The system should be intuitive enough for users of varying technical abilities to navigate with ease.
3. **Scalability:** The system should be designed in a way that allows for future expansion, whether it be more features or increased user capacity.
4. **Error Handling:** The system should gracefully handle errors, such as insufficient funds or invalid login attempts, providing clear feedback to users.

This project aims to address these challenges by developing a Java-based banking system with a simple frontend interface. The system will demonstrate how core banking operations can be managed efficiently while ensuring security and user-friendliness.

### 3 Existing and Proposed solution

**Provided Code:** Basic password protection is implemented, but there are no advanced security measures.

**Typical Banking Application:** Uses advanced security protocols such as encryption (e.g., AES, RSA), multi-factor authentication (MFA), and secure hashing algorithms (e.g., bcrypt) for password storage. Additionally, HTTPS is used to encrypt data in transit.

**Provided Code:** The code is single-threaded and doesn't handle multiple users concurrently. **Typical Banking Application:** Designed to handle thousands of concurrent users, with advanced concurrency control, load balancing, and distributed system architectures to ensure high availability and performance.

**Provided Code:** Likely tested manually with limited or no automated tests.

**Typical Banking Application:** Subjected to rigorous testing, including unit testing, integration testing, user acceptance testing, and possibly even stress and load testing. Automated test suites and continuous integration/continuous deployment (CI/CD) pipelines are commonly used.

#### 3.1 Code submission (Github link)

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