Industrial Internship Report on

”Banking Information System”

Prepared by

[Kapil Khatri]

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. The system provides a working preview of the key functionalities of a real banking system, demonstrating the core features, flows, and usability. The prototype was implemented using Java Swing for a graphical user interface (GUI) to enhance user interactivity and usability. 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

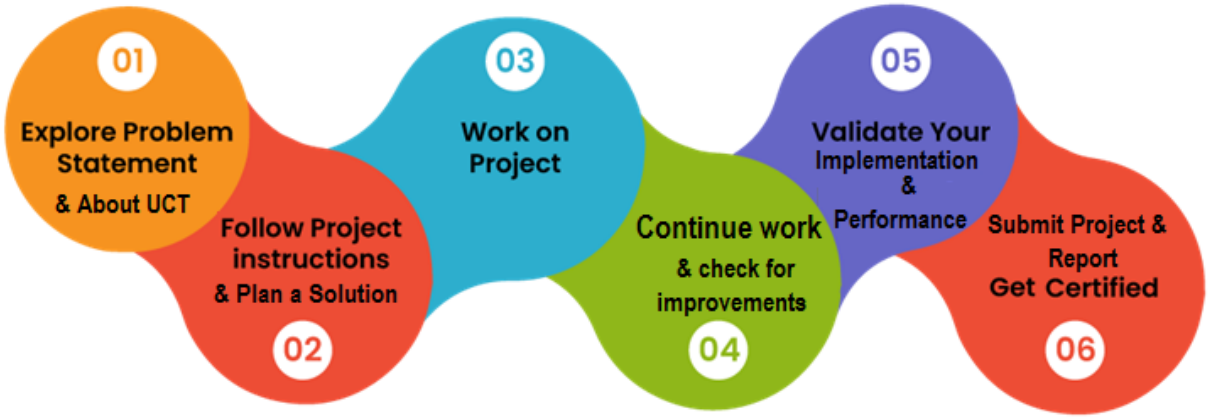
Summary of the whole 6 week's work.

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.

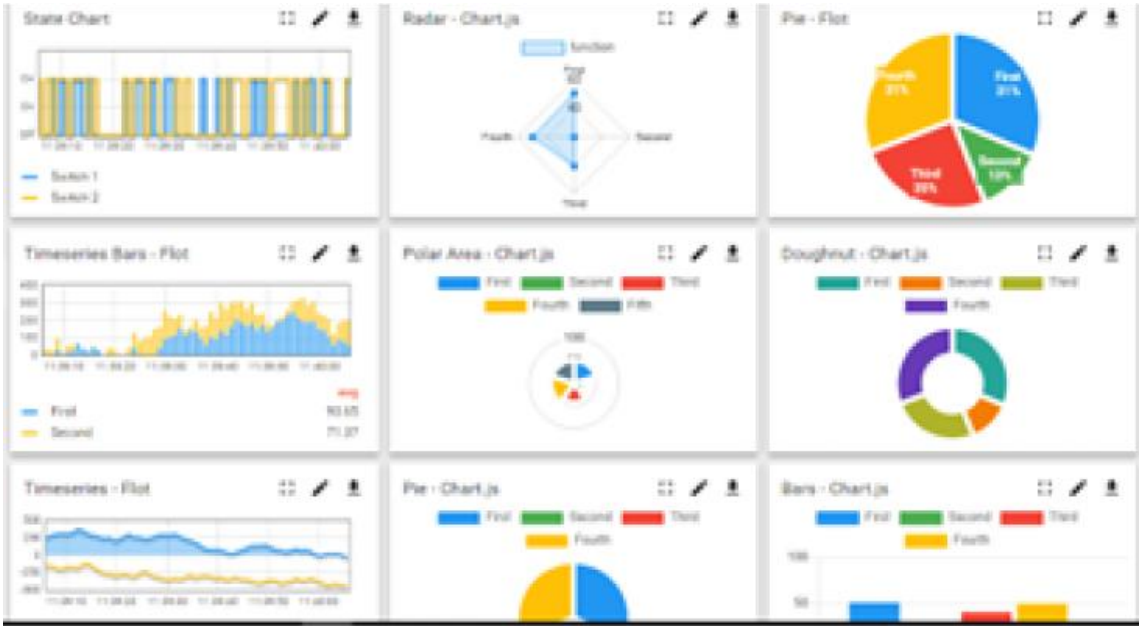
i. UCT IoT Platform (UCT Insight)

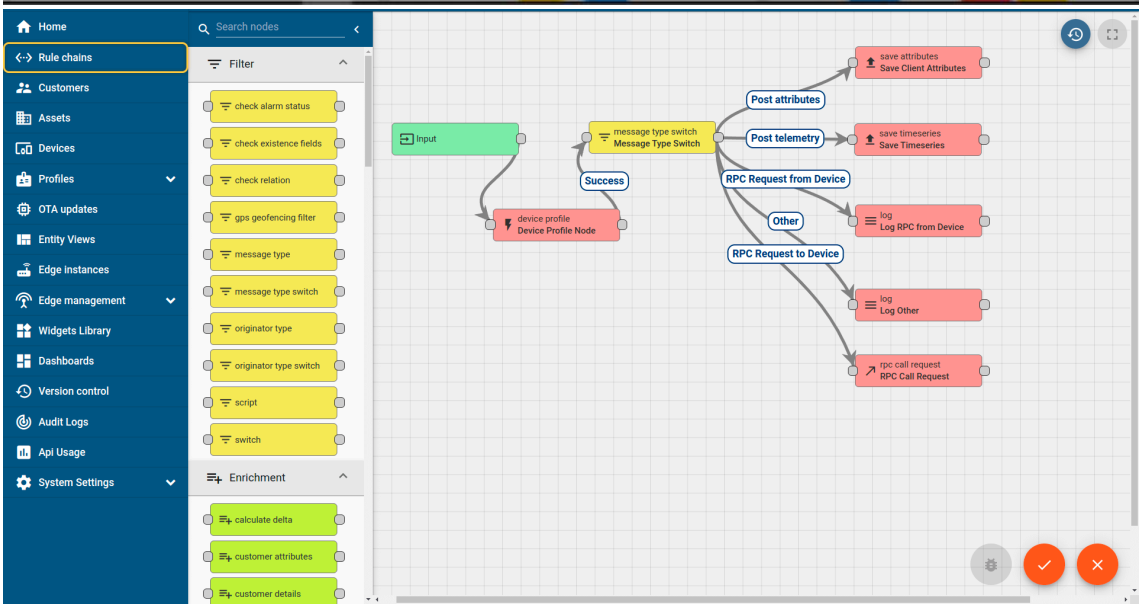
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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.

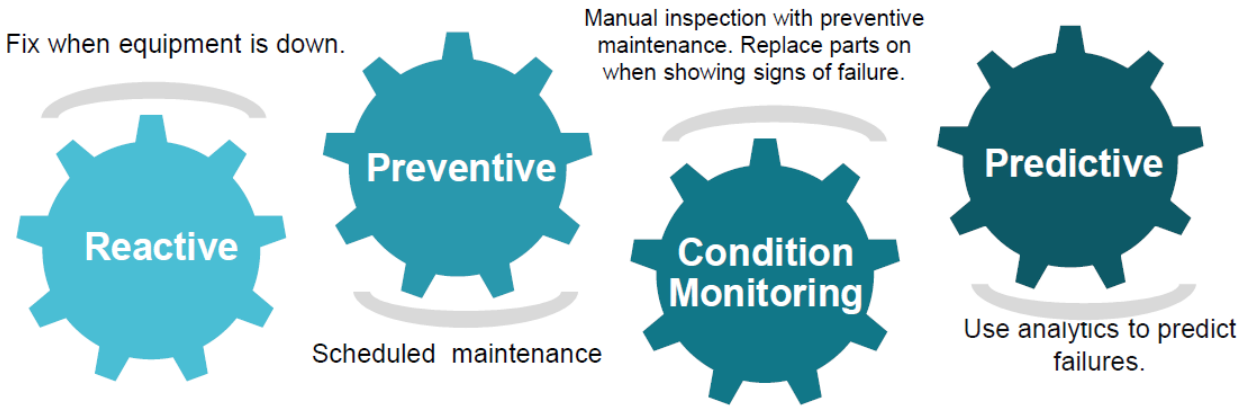
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

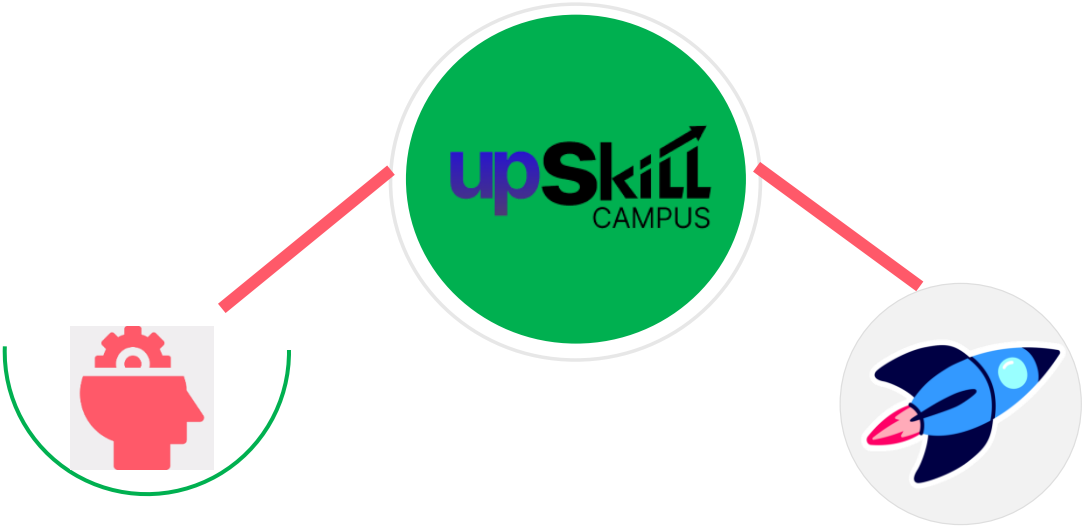
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

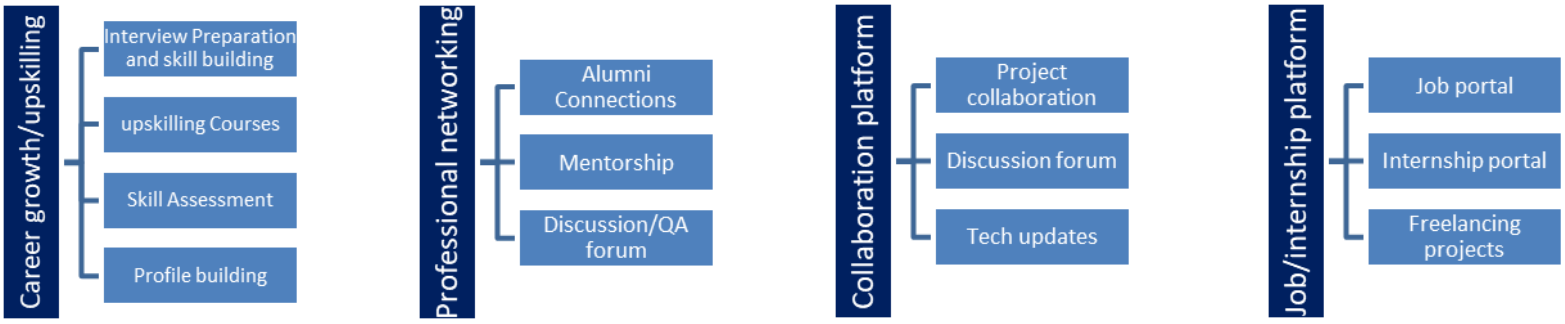
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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.javatpoint.com/java-tutorial

[2] YouTube

[3] Chatgpt

3 Problem Statement

In the assigned problem statement

1. **Limited Data Persistence:**
    - The prototype uses in-memory storage, meaning all user data and transactions are lost when the application is closed. This limitation prevents long-term data retention and is not suitable for a real-world banking system where persistent storage (e.g., a database) is necessary.
2. **Security Concerns:**
    - The prototype does not implement advanced security features such as password encryption or secure user authentication. In a real-world application, this would be a significant issue, as it could lead to unauthorized access to user accounts and sensitive information.
3. **Scalability Issues:**
    - The current implementation is not designed to handle a large number of users or transactions. The in-memory storage and basic data structures used may lead to performance degradation as the number of users and transactions increases.
4. **Basic Error Handling:**
    - The error handling implemented in the prototype is quite basic and may not cover all possible edge cases. For example, the system does not check for issues like invalid account numbers during transactions or handle concurrent transactions, which could lead to inconsistent data states.
5. **User Interface Limitations:**
    - Although the Swing-based GUI is functional, it is somewhat outdated compared to modern user interface frameworks. The design could be improved to provide a more modern and responsive user experience.
6. **Lack of Multi-user Support:**
    - The prototype is not designed for multi-user environments. It assumes a single user interacting with the system at a time, which is not realistic for a banking system where multiple users would be accessing the system concurrently.

4 Existing and Proposed solution

What is your proposed solution?

Problem: The prototype currently uses in-memory storage, which means all user data and transactions are lost when the application is closed. This limitation is not suitable for a real-world banking system, where data persistence is critical.

Solution:

- **Database Integration:** Implement a database (such as MySQL, PostgreSQL, or an embedded database like SQLite) to store user account information, transaction history, and other relevant data. This ensures that data is retained across sessions.

What value addition are you planning?

1. Enhance the system's security to protect user data
2. Provide reliable data storage and retrieval mechanisms to ensure data persistence and integrity.
3. Improve the user interface to provide a modern, intuitive, and accessible user experience.

4.1 Code submission (Github link)

https://github.com/KhatriKapil/upskillcampus/blob/main/BankingInformationSystem.java

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

5 My learnings

1. Software Design and Architecture

Learning:

- **Modular Design:** You learned the importance of breaking down a complex system into modular components, each responsible for a specific functionality (e.g., user registration, account management, transactions).
- **Layered Architecture:** Understanding the separation of concerns between the user interface (UI), business logic, and data persistence layers helps in building maintainable and scalable software.

---

2. Core Java Programming

Learning:

- **Java Fundamentals:** Reinforced your understanding of core Java concepts such as object-oriented programming (OOP), exception handling, and collections.
- **Swing for GUI Development:** Gained hands-on experience with Java Swing for creating graphical user interfaces, including working with components like JFrame, JButton, and JOptionPane.
- **Concurrency:** Exposure to basic multi-threading concepts and the challenges of handling concurrent operations in a system, which is crucial for building responsive applications.

---

3. User Interface Design

Learning:

- **User Experience (UX) Design:** You learned the importance of designing user-friendly interfaces that are intuitive and easy to navigate, which enhances the overall user experience.
- **JavaFX vs. Swing:** Gained insights into the limitations of Swing and the potential benefits of using more modern UI frameworks like JavaFX for building richer, more responsive interfaces.

---

4. Security Practices

Learning:

- **Importance of Security:** Recognized the critical role of security in software systems, especially in applications dealing with sensitive information like banking systems.

- **Encryption and Authentication:** Learned about the need for encryption of sensitive data (e.g., passwords) and the implementation of secure user authentication mechanisms to protect user data.

---

5. Data Persistence and Management

Learning:

- **Temporary vs. Persistent Storage:** Understood the difference between in-memory storage and persistent storage, and the importance of data persistence in real-world applications.
- **Database Integration:** Gained awareness of how databases can be used to store and manage large volumes of data, ensuring data integrity and consistency across sessions.

---

6. Error Handling and Validation

Learning:

- **Robust Error Handling:** Learned the importance of comprehensive error handling to manage potential issues such as invalid inputs, insufficient funds, and system errors.
- **Input Validation:** Reinforced the necessity of validating user inputs to prevent common issues like data corruption and security vulnerabilities.

---

7. Scalability and Performance Optimization

Learning:

- **System Scalability:** Understood the challenges of scaling a system to handle increasing numbers of users and transactions, and the techniques that can be used to improve performance (e.g., load balancing, caching).
- **Efficient Data Structures:** Recognized the importance of choosing appropriate data structures for handling large datasets efficiently.

---

8. Project Management and Planning

Learning:

- **Feature Prioritization:** Learned how to prioritize features and plan for future enhancements to ensure that the system meets user needs and adapts to changing requirements.
- **Incremental Development:** Understood the value of developing software incrementally, starting with a prototype and gradually adding features and improvements based on feedback and testing.

- **9. Real-World Application Considerations**

Learning:

- **Compliance and Regulations:** Gained insights into the importance of considering legal and regulatory requirements (e.g., data privacy, KYC) when developing software for real-world applications like banking systems.
- **User Support:** Recognized the need for providing users with adequate support, including help centers, feedback mechanisms, and customer service options, to ensure a positive user experience.

---

10. Continuous Improvement

Learning:

- **Iterative Development:** Understood that software development is an iterative process, where continuous feedback and improvement are necessary to refine the system and meet evolving user needs.
- **Learning from Challenges:** Learned how to analyze and address challenges encountered during development, turning problems into opportunities for learning and improvement.

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