**Industrial Internship Report on**

**Cloud based Financial Service**

**Prepared by:**

**Prit Kashyap Soni**

|  |
| --- |
| *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 :  **A cloud-based solution for financial services can help banks and other financial institutions to securely manage and process large amounts of financial data.**  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](#_Toc139702806)

[2 Introduction 4](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 4](#_Toc139702808)

[2.2 About upskill Campus 8](#_Toc139702809)

[2.3 The IOT acedemy 10](#_Toc139702810)

[2.4 Objective 10](#_Toc139702811)

2.5 References…………………………………………………………………………………………………………………………………..10

[3 Problem Statement 11](#_Toc139702813)

[4 Existing and Proposed solution 13](#_Toc139702814)

[5 Proposed Design/ Model 17](#_Toc139702815)

[5.1 High Level Diagram (if applicable) 18](#_Toc139702816)

[5.2 Low Level Diagram (if applicable) 19](#_Toc139702817)

[6 Performance Test 20](#_Toc139702819)

[6.1 Test Plan/ Test Cases 20](#_Toc139702820)

[6.2 Test Procedure 20](#_Toc139702821)

[6.3 Performance Outcome 20](#_Toc139702822)

[7 My learnings 21](#_Toc139702823)

# 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.

# Introduction

## 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.



1. 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

1. **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 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.

1.  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.

1. 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.



## 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

<https://www.upskillcampus.com/>

upSkill Campus aiming to upskill 1 million learners in next 5 year



## 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.

## 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.

## Reference

[1] Youtube

[2] Udemy

[3] Coursera

[4] Github

# Problem Statement:

**Title:** **Development of a Secure, Cloud-Based Financial Services Application**

**Objective:** To design and implement a cloud-based financial services application that provides secure, scalable, and efficient financial data management and processing for banks and financial institutions. The application should include features such as fraud detection, customer relationship management (CRM), and accounting. Additionally, it should have robust user authentication and authorization mechanisms.

**Background:** Financial institutions manage large volumes of sensitive data, necessitating a secure and efficient way to handle data processing and management. Traditional on-premises systems often struggle with scalability, security, and maintenance challenges. Cloud computing offers a solution to these issues, providing scalability, security, and cost-efficiency. However, migrating financial services to the cloud requires careful planning and implementation to ensure data security, compliance, and performance.

**Challenges:**

1. **Data Security:** Ensuring that sensitive financial data is securely stored and transmitted.
2. **Scalability:** Designing the application to handle increasing amounts of data and user traffic without performance degradation.
3. **Fraud Detection:** Implementing effective algorithms to detect and prevent fraudulent activities.
4. **Customer Relationship Management:** Managing customer data and interactions efficiently.
5. **Authentication and Authorization:** Implementing secure and user-friendly authentication mechanisms.
6. **Continuous Integration and Deployment (CI/CD):** Automating the build, test, and deployment processes to ensure quick and reliable updates.

**Requirements:**

1. **Cloud Infrastructure:** Utilize AWS for deploying the application, leveraging services such as EC2, S3, and RDS.
2. **Backend Development:** Use Python for core application logic, focusing on data processing, fraud detection, and CRM functionalities.
3. **Frontend Development:** Develop a responsive and user-friendly interface using React (or a similar framework) for user interactions and data visualization.
4. **Containerization and Orchestration:** Use Docker to containerize the application and Kubernetes (EKS) for orchestrating container deployment.
5. **CI/CD Pipeline:** Implement a CI/CD pipeline using Jenkins to automate the build, test, and deployment processes.
6. **Authentication:** Integrate AWS Cognito for user authentication and authorization.
7. **Data Security:** Implement encryption for data at rest and in transit, and use IAM roles and policies to restrict access.

**Expected Outcome:** A cloud-based financial services application that is secure, scalable, and efficient, providing features such as fraud detection, CRM, and accounting. The application will have a robust CI/CD pipeline and user authentication system, ensuring seamless and secure user interactions and data management.

# Existing and Proposed solution

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

**Existing Solutions:**

1. **On-Premises Financial Software:**
   * **Examples:** Oracle Financial Services, SAP Financial Services, IBM Financial Services.
   * **Description:** These are traditional software solutions installed and run on the financial institution's own servers and infrastructure.
   * **Limitations:**
     + **Scalability Issues:** Scaling on-premises solutions to handle increasing amounts of data and users can be expensive and complex.
     + **High Maintenance Costs:** Requires significant investment in hardware, software updates, and IT staff for maintenance and troubleshooting.
     + **Limited Accessibility:** Accessing the system remotely can be challenging and often requires VPNs or other secure access methods.
     + **Slow Deployment:** Implementing updates and new features can be time-consuming, leading to longer deployment cycles.
2. **Cloud-Based Financial Services Platforms:**
   * **Examples:** Finacle by Infosys, FIS Cloud, Temenos.
   * **Description:** These solutions offer financial services through cloud platforms, providing better scalability and remote access.
   * **Limitations:**
     + **Cost:** These platforms can be costly, especially for small to mid-sized financial institutions.
     + **Complexity:** Integration with existing systems and customization can be complex and require specialized skills.
     + **Vendor Lock-In:** Dependency on a single vendor can limit flexibility and increase costs over time.
     + **Security Concerns:** Ensuring data security and compliance with regulations remains a critical concern.

What is your proposed solution?

**Proposed Solution:**

**Development of a Secure, Cloud-Based Financial Services Application**

1. **Cloud Infrastructure:**
   * **Platform:** Utilize AWS for deploying the application.
   * **Services:** EC2 for compute power, S3 for storage, RDS for database management, and EKS for Kubernetes-based container orchestration.
2. **Backend Development:**
   * **Technology:** Use Python for core application logic.
   * **Features:** Implement fraud detection algorithms, customer relationship management (CRM), and accounting functionalities.
3. **Frontend Development:**
   * **Framework:** Develop a responsive user interface using React.
   * **Features:** User-friendly design for interacting with financial data and visualizations.
4. **Containerization and Orchestration:**
   * **Tools:** Docker for containerization, Kubernetes (EKS) for managing and deploying containers.
5. **CI/CD Pipeline:**
   * **Tool:** Use Jenkins for continuous integration and deployment.
   * **Pipeline:** Automate build, test, and deployment processes to ensure quick and reliable updates.
6. **Authentication:**
   * **Service:** Integrate AWS Cognito for user authentication and authorization.
   * **Features:** Secure sign-up, sign-in, and user management.
7. **Data Security:**
   * **Measures:** Implement encryption for data at rest and in transit, IAM roles and policies for access control.

What value addition are you planning?

**Value Addition:**

1. **Cost Efficiency:**
   * **Lower Infrastructure Costs:** Leveraging AWS services reduces the need for costly on-premises hardware and maintenance.
   * **Pay-as-You-Go:** AWS's pricing model allows for cost savings by paying only for the resources used.
2. **Scalability:**
   * **Auto-Scaling:** AWS auto-scaling capabilities ensure that the application can handle increased loads without performance degradation.
   * **Global Reach:** Deploying on AWS allows for a global presence, ensuring low latency and high availability.
3. **Enhanced Security:**
   * **Robust Security Measures:** AWS provides advanced security features, including encryption, IAM, and compliance with global standards.
   * **Authentication and Authorization:** AWS Cognito ensures secure and scalable user management.
4. **Rapid Development and Deployment:**
   * **CI/CD Pipeline:** Jenkins integration allows for continuous integration and deployment, reducing time to market and improving the reliability of updates.
   * **Containerization:** Docker and Kubernetes streamline the deployment process, making it easier to manage and scale the application.
5. **Advanced Features:**
   * **Fraud Detection:** Implementing machine learning algorithms to detect fraudulent activities in real-time.
   * **CRM:** Efficient management of customer data and interactions, improving customer service and engagement.
6. **Customization and Flexibility:**
   * **Modular Design:** The application is designed to be modular, allowing for easy customization and integration with other systems.
   * **Vendor Independence:** While leveraging AWS, the solution is designed to avoid vendor lock-in, allowing for flexibility in future technology choices.

## Code submission (Github link)

<https://github.com/Bax-47/Cloud-based-Financial-Services.git>

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

<https://github.com/Bax-47/Cloud-based-Financial-Services/blob/main/test>

# Proposed Design/ Model

|  |
| --- |
| Initial Setup and Planning  | - Define Scope and Requirements  | - Set Up AWS Infrastructure  | - Plan Architecture and Workflow  | - Set Up Version Control (Git) |
| Arrow: Straight with solid fill |
| Backend Development  | - Develop Core Logic (Python)  | - Implement Security Measures  | - Data Processing and ML Algorithms |
| Arrow: Straight with solid fill |
| Frontend Development  | - Design Responsive UI (React)  | - Integrate with Backend  | - Implement Data Visualizations |
| Arrow: Straight with solid fill |
| Authentication and Authorization  | - Set Up AWS Cognito User Pool  | - Implement Authentication in  | Backend (Boto3) and Frontend  | (AWS Amplify) |
| Arrow: Straight with solid fill |
| Containerization and Orchestration  | - Create Dockerfiles for Services  | - Containerize Application (Docker)  | - Deploy Containers with Kubernetes  | (EKS) |
| Arrow: Straight with solid fill |
| Continuous Integration and  | Deployment (CI/CD)  | - Set Up Jenkins  | - Configure Jenkinsfile  | - Automate Build, Test, Deploy |
| Arrow: Straight with solid fill |
| Monitoring and Maintenance  | - Monitor with AWS CloudWatch  | - Implement Centralized Logging  | - Conduct Regular Updates and  | Security Audits |

## High Level Diagram (if applicable)

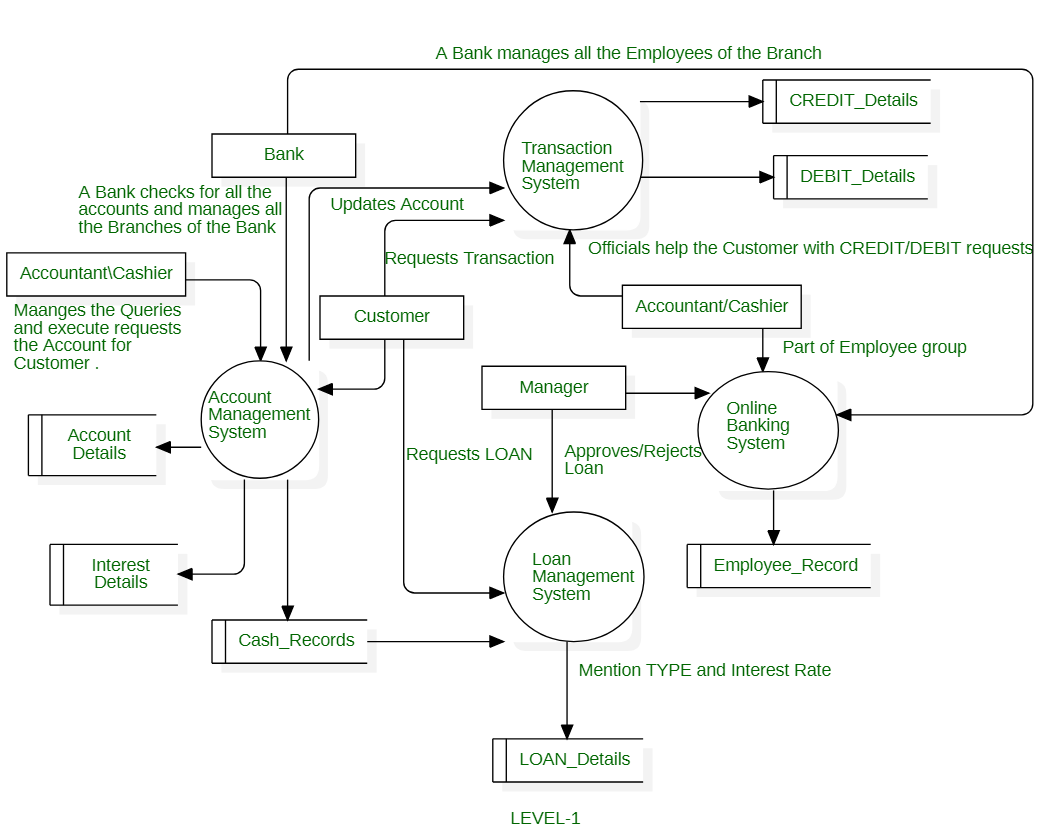
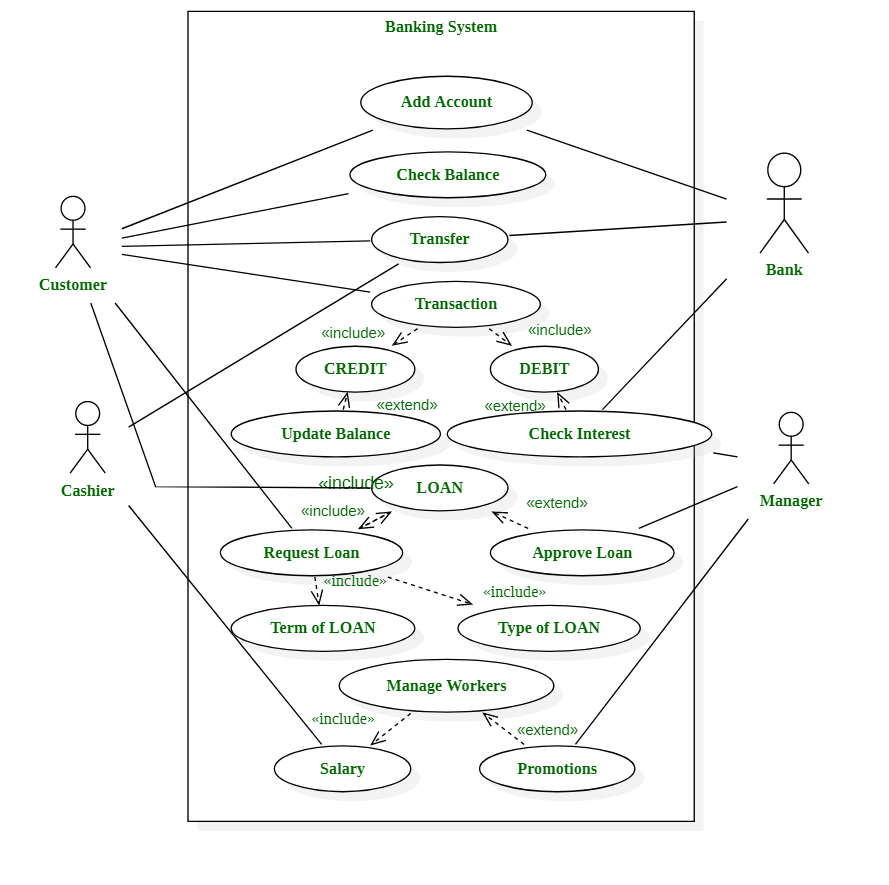


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

## Low Level Diagram (if applicable)



# 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.

## Test Plan/ Test Cases

6.1 Test Environment: • Web browser for accessing the web application • tool for sending API requests • AWS Management Console for monitoring AWS services

## Test Procedure

1. Storing Customer Data.

2. Retrieving Customer Data.

3. Security and Access Control.

4. Data Integrity.

5. Performance.

6. Error Handling.

## Performance Outcome

A cloud-based financial services application that is secure, scalable, and efficient, providing features such as fraud detection, CRM, and accounting. The application will have a robust CI/CD pipeline and user authentication system, ensuring seamless and secure user interactions and data management.

# My learnings

**1. Object-Oriented Programming (OOP) Concepts:**

* **Core Principles:** Understanding of encapsulation, inheritance, polymorphism, and abstraction.
* **Application in Java:** Ability to implement these principles using Java, including class design, object manipulation, and method overriding.

**2. Cloud-Based Financial Services:**

* **Scalability:** Insight into how cloud services can handle variable workloads efficiently.
* **Cost Efficiency:** Knowledge of cost-saving mechanisms provided by cloud infrastructure.
* **Security and Compliance:** Awareness of security measures and regulatory compliance essential for financial services.
* **Performance Optimization:** Understanding of how cloud solutions can enhance performance and reduce latency.
* **Disaster Recovery and Flexibility:** Familiarity with disaster recovery strategies and the flexibility offered by cloud platforms.
* **Impact on Career Growth**

**1. Enhanced Technical Skills:**

* Mastery of OOP principles and Java will improve your coding proficiency, enabling you to design and implement robust, maintainable software solutions.

**2. Cloud Expertise:**

* Proficiency in cloud-based financial services will make you valuable in roles involving financial technology, system architecture, and cloud infrastructure management.

**3. Competitive Edge:**

* The combination of advanced programming skills and cloud knowledge sets you apart in the job market, making you a strong candidate for positions in technology-driven financial sectors.

**4. Problem-Solving Abilities:**

* Your ability to leverage cloud solutions for performance optimization and disaster recovery will enhance your problem-solving skills, especially in high-stakes environments.

**5. Career Advancement:**

* With these skills, you are well-positioned for roles such as Software Engineer, Cloud Solutions Architect, or Financial Systems Analyst, paving the way for career growth and leadership opportunities.