Industrial Internship Report on

"Banking Information System"

Prepared by

[kamasani likhitha]

| 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 software application developed using the Java programming language and JDBC (Java Database Connectivity) for database operations. It provides a menu-driven interface that allows users to perform various banking-related tasks.

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
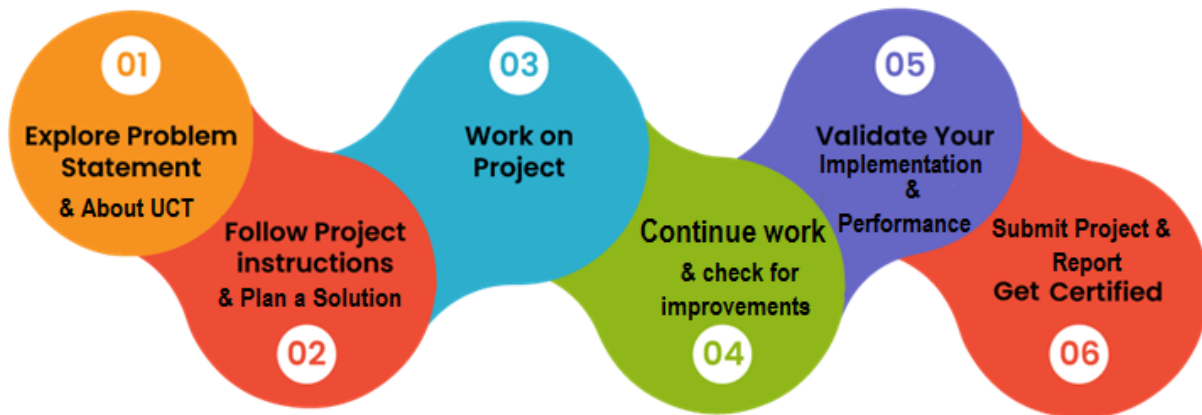
Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

2 Introduction

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
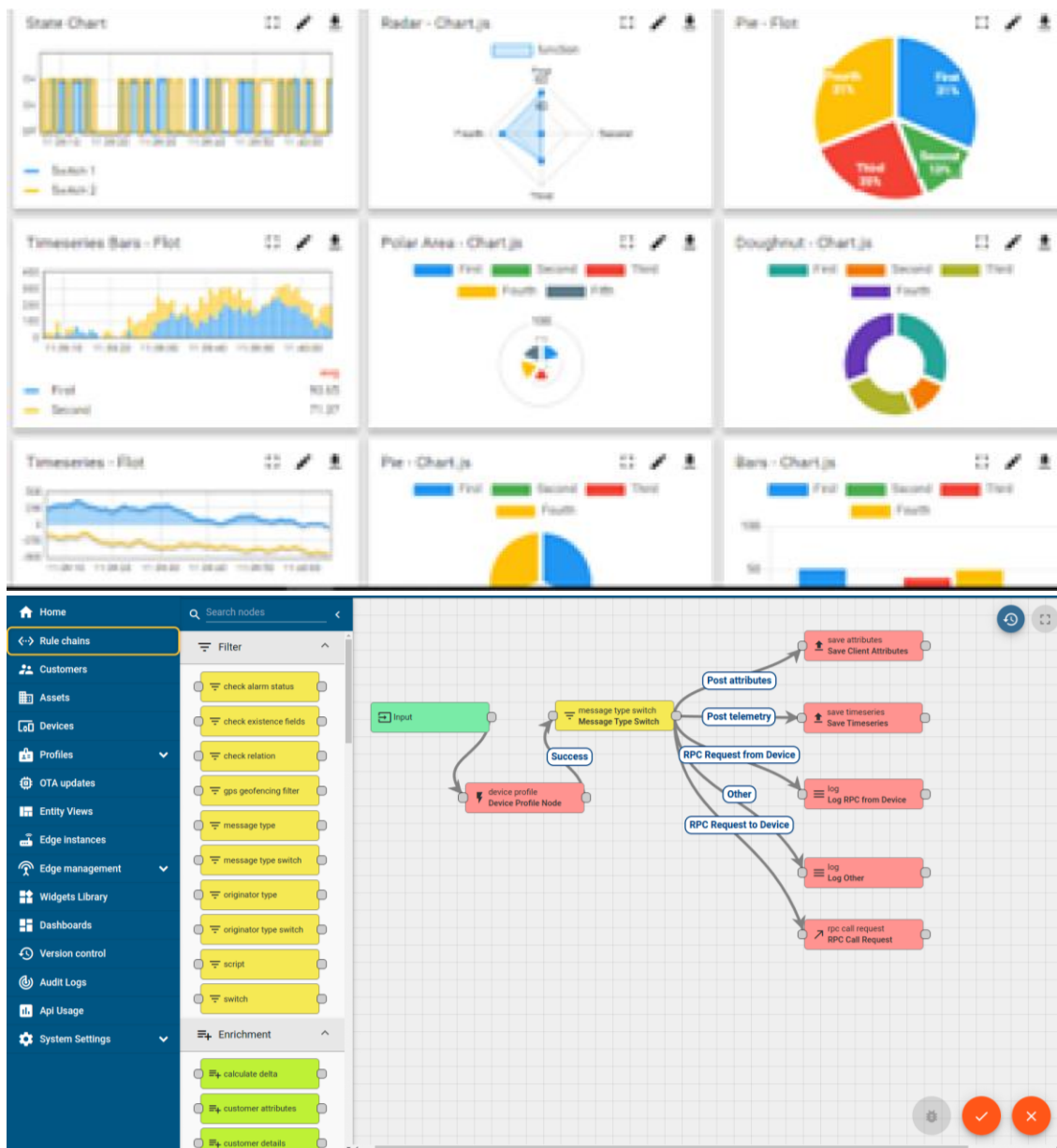
ii.   **Smart Factory Platform (** FACTORY WATCH **)**

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.
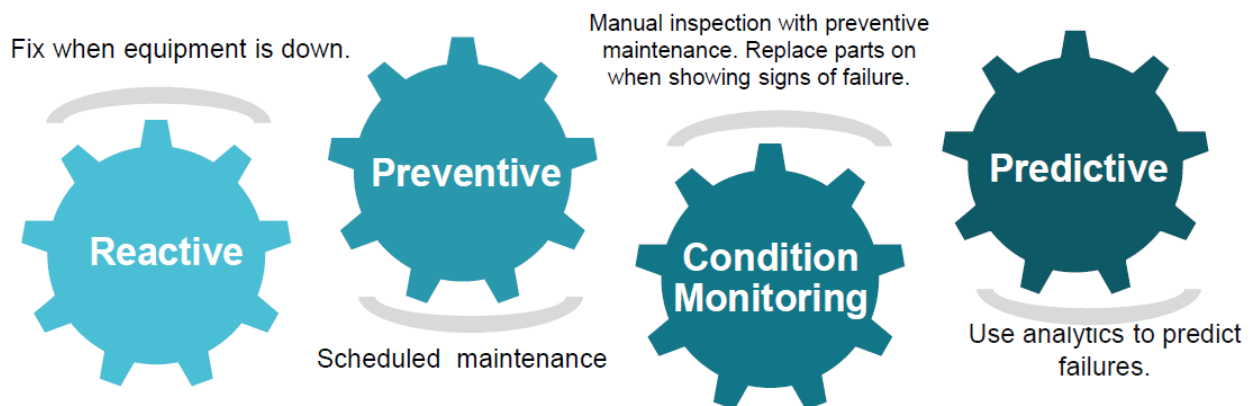
About upskill Campus (USC)

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.
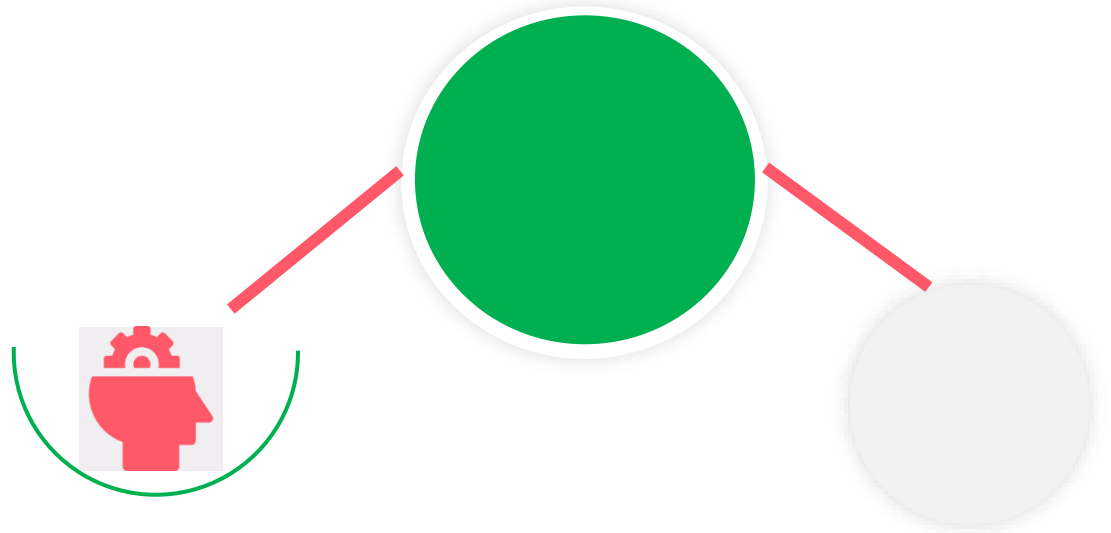
Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

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

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] https://www.javatpoint.com/banking-application-in-java

[2]https://projectsgeek.com/2016/02/complete-banking-system-java-project.html

[3]https://www.geeksforgeeks.org/banking-transaction-system-using-java/

Glossary

Terms	Acronym
UniConverge Technologies	UCT
Virtual Student Foreign Service	VSFT
Associate Consultant Intern	ACI
African Community Internship Placement Programme	ACIPP
Accounting Intern Scholarship Program	AISP

3 Problem Statement

DEFINATIONS:
Online bank transactions refers to bank transactions made on the specific bank site by
a particular individual in a secure network i.e. without using hard cash, we can use the facility of
debiting and crediting with the help of login ids and passwords provided by the bank.

WORKING:
1. Online registration:
Firstly, the user has to register for an online banking account on the banks website byfilling in
the necessary details like his name, account no., mailing address, his contact no. and other
necessary details.
2. Log in:
After registration, the user can now login in by entering his ID and password. After log in, the
user gets various options and can perform various banking tasks.
3. Options:
The various options available after successful log in are:
(a) Account statement:
It gives the details of the various transactions made over a selected period of time.The user can
get a detailed description of the transactions made by him over a selected period of time.
(b) Withdrawal:
This gives the user an option of withdrawing an amount of money from his account. The user
can withdraw a specified amount and thus, his balance gets deducted accordingly.
(c) Deposit:
This gives the user an option to deposit money to his account and on deposition, the
user's balance is increased accordingly.
(d) Current balance:
It gives the total amount of money left in the account after all the transactions have beenmade.
(e) Transfer:
This gives the option of transferring money to other accounts as well.
4. Log Out:
The user can safely log out of his account and go back to the home page of the banking website.

4 Existing and Proposed solution

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

1   Net banking may be challenging for those who are new to using it
2   An internet connection is necessary to use the online banking services
3   It is important to regularly change passwords to ensure the security of the bank account and protect it from hackers (e.g. password changes may be required every two months)

What is your proposed solution?

1   More secure information will give a layer of security of authentication and authorization.
2   Required very little manpower.
3   Simplify the problem of editing.
4   Maintain the clearance level by the hierarchy.

What value addition are you planning?

▪   Maintain data integrity Validate the manual calculations avoid calculation error.
▪   Safeguard the data accuracy.
▪   More reliable and efficient.
▪   More user-friendly interface.

4.1 Code submission (Github link)

https://github.com/Likhitha1234567/upskill-campus.git

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

https://github.com/Likhitha1234567/upskill-campus.git

5 Proposed Design/Model

Given more details about design flow of your solution. This is applicable for all domains. DS/ML Students can cover it after they have their algorithm implementation. There is always a start, intermediate stages and then final outcome.
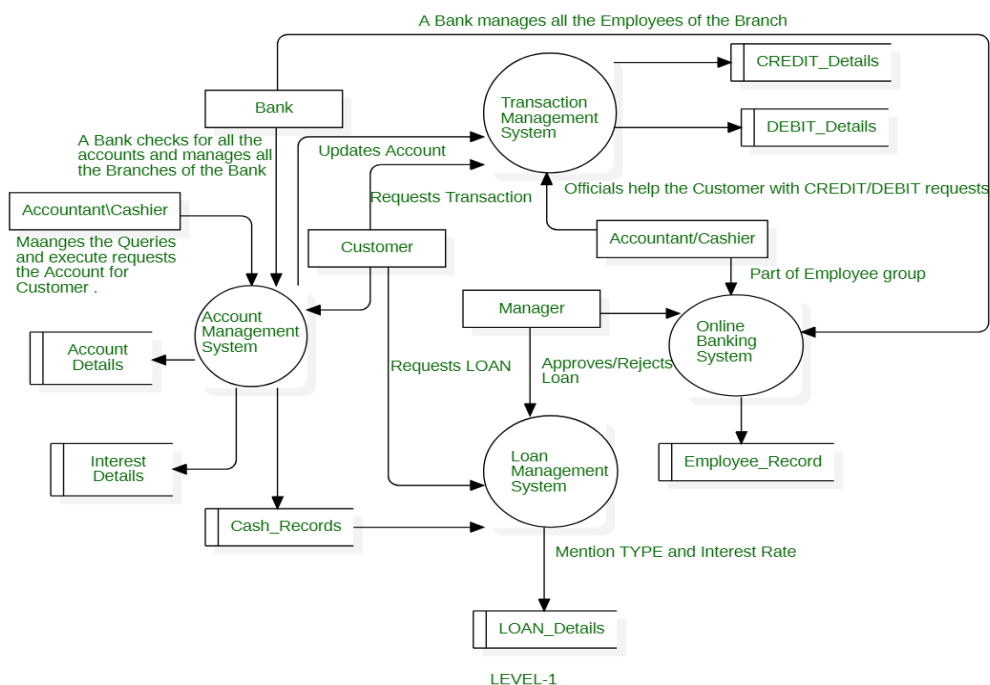
5.1 High Level Diagram (if applicable)

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram (if applicable)

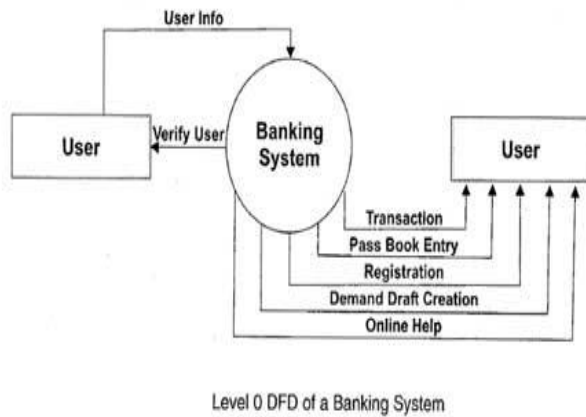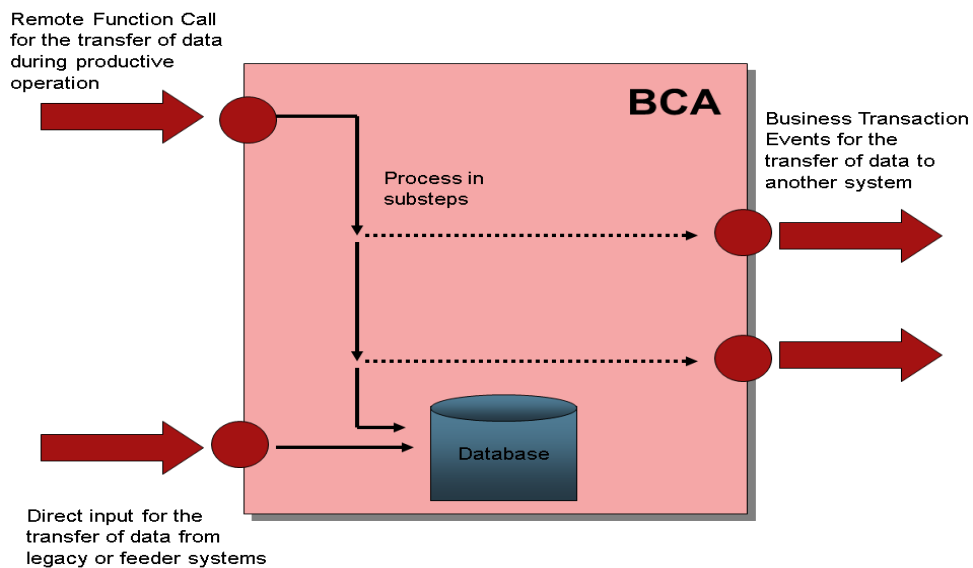Level 0 DFD of a Banking System

Figure 2: LOW LEVEL DIAGRAM OF THE SYSTEM

5.3 Interfaces (if applicable)

Update with Block Diagrams, Data flow, protocols, FLOW Charts, State Machines, Memory Buffer Management.

6 Performance Test

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

The constraints in a banking information system can arise from various factors, such as regulatory requirements, security concerns, scalability, performance, and data privacy. It's crucial to identify these constraints early in the design process to ensure they are appropriately addressed.

How those constraints were taken care in your design?

Compliance: The system must comply with relevant banking regulations and data protection laws to ensure customer data security and privacy.

Security: Robust security measures, including encryption, access controls, and authentication mechanisms, should be implemented to safeguard sensitive financial data from unauthorized access.

Scalability: The system should be designed to handle increasing data volumes and user demands as the bank grows.

Performance: Ensuring the system operates efficiently with minimal latency is critical for providing a seamless banking experience to customers.

Reliability: The system must be highly available and resilient to prevent any service disruptions or data loss.

User Experience: The system should be user-friendly and intuitive for both bank employees and customers to enhance productivity and satisfaction.

What were test results around those constraints?

Testing is essential to ensure that the banking information system meets the identified constraints. Various types of testing can be conducted:

Security Testing: Penetration testing, vulnerability assessments, and data privacy audits can help identify and fix potential security vulnerabilities.

Performance Testing: Load testing, stress testing, and benchmarking can verify whether the system performs efficiently under different workloads and stress conditions.

Compliance Testing: Audits and assessments can ensure the system adheres to relevant regulatory standards and data protection laws.

User Acceptance Testing (UAT): UAT involves testing the system with end-users to ensure it meets their needs and expectations.

6.1 Test Plan/ Test Cases

1.Test Plan

Introduction:

- Purpose
- Scope
- Objectives
- Assumptions
- Test Deliverables
- Test Environment

Test Strategy:

- Testing Approach
- Test Levels
- Test Types

Test Schedule:

- Test Milestones
- Resource Allocation

Test Environment:

- Hardware Requirements
- Software Requirements
- Test Data

2.Test Cases

Functional Test Cases:

- Verify user registration process
- Validate login functionality with valid and invalid credentials
- Test fund transfer between different accounts (within the same bank and interbank)
- Check balance update after a successful transaction
- Verify the ability to add and manage beneficiaries for fund transfers
- Test bill payment process
- Validate ATM withdrawal process
- Check the loan application and approval process

Integration Test Cases:

- Verify the integration between the front-end application and the back-end banking server
- Test data synchronization between different modules (e.g., account information, transaction history, etc.)
- Validate integration with third-party payment gateways

Performance Test Cases:

- Conduct load testing to verify the system's performance under typical user loads
- Test system response time for various transactions
- Verify the system's ability to handle peak loads

Security Test Cases:

- Test user authentication and authorization mechanisms
- Check for SQL injection and other security vulnerabilities
- Verify data encryption during transmission
- Test session management and logout functionality

3. Test Execution:

- Execute test cases as per the test plan
- Document test results and any defects found during testing
- Retest fixed defects to verify the resolution

4. Defect Reporting and Tracking:

- Use a defect tracking system to report and track issues
- Assign severity and priority levels to each defect

5. Test Completion:

Evaluate the test results and provide a summary report

Obtain sign-off from stakeholders

6.2 Test Procedure

1. Test Procedure Overview:

• Provide an overview of the test procedure, including its purpose, scope, and objectives.

• Mention the modules or functionalities that will be tested.

2. Test Environment Setup:

• Specify the hardware and software requirements for the test environment.

• Set up the necessary databases and configurations for testing.

3. Test Data Preparation:

• Prepare relevant test data, including user accounts, transaction details, and various scenarios to simulate real-world usage.

• Ensure that the test data covers both positive and negative test scenarios.

4. Test Execution Steps:

• List the step-by-step instructions to execute the test cases.

• Provide clear guidance on how to set up and configure the test environment if needed.

• Include any preconditions or prerequisites necessary for executing specific test cases.

5. Test Case Execution:

• For each test case, provide the following details:

• Test case identifier

• Test case name/description

• Test data to be used

• Expected results

• Execute the test cases in a structured manner.

6. Test Result Recording:

• Record the actual results for each executed test case.

• Document any deviations or issues encountered during testing.

7. Defect Reporting:

• If any defects are identified during testing, clearly document them in a defect tracking system.

• Include necessary details like steps to reproduce, severity, and priority.

8. Test Case Status and Traceability:

• Keep track of the status of each test case (Pass/Fail/Blocked).

• Ensure proper traceability between test cases and requirements to demonstrate test coverage.

9. Regression Testing:

• After fixing defects or implementing changes, conduct regression testing to ensure that the fixes do not introduce new issues.

• Document the results of regression testing.

10. Performance Testing:

• If applicable, conduct performance testing to assess the system's response time, throughput, and scalability under different loads.

• Record performance test results.

11. Security Testing:

• Conduct security testing to identify vulnerabilities and ensure data protection.

• Document any security-related findings and their resolutions.

12. User Acceptance Testing (UAT):

• If possible, involve end-users in UAT to validate that the system meets their requirements and expectations.

• Record UAT results and feedback.

13. Test Completion and Summary:

• Summarize the overall test results, including the number of test cases executed, passed, failed, and blocked.

• Provide a conclusion on whether the system meets the specified criteria for release.

14. Test Sign-Off:

• Obtain sign-off from relevant stakeholders to indicate that testing has been completed satisfactorily.

15. Test Artifacts and Documentation:

• Ensure that all test artifacts, including test cases, test results, defect reports, and any additional documentation, are properly organized and stored.

6.3 Performance Outcome

The performance outcome for a banking information system refers to how well the system meets its performance requirements and objectives when subjected to various loads and user interactions. The performance outcome is crucial in ensuring that the system functions efficiently and provides a satisfactory user experience. Here are some key performance outcomes for a banking information system:

1. Response Time: The response time measures how quickly the system responds to user actions, such as logins, account inquiries, fund transfers, and bill payments. A low response time is desirable, as it indicates that the system processes user requests promptly.

2. Throughput: Throughput refers to the number of transactions or operations the system can handle within a given time frame. Higher throughput indicates better efficiency in processing user requests and handling concurrent transactions.

3. Scalability: Scalability measures how well the system performs as the number of users and transactions increases. A scalable banking information system can handle increased loads without significant degradation in performance.

4. Stability: The stability of the system is evaluated based on its ability to maintain consistent performance over time and under varying conditions. A stable system should not exhibit unexpected crashes or downtime.

5. Load Handling Capacity: This metric measures the system's ability to handle peak loads and stress conditions without impacting its performance negatively. A well-performing banking information system should be able to handle heavy loads during peak business hours or seasonal spikes.

6. Resource Utilization: Resource utilization assesses how efficiently the system uses its hardware resources (CPU, memory, disk, etc.). Optimized resource utilization ensures that the system operates efficiently and minimizes resource bottlenecks.

7. Concurrency: Concurrency refers to the system's ability to handle multiple user interactions simultaneously without compromising response times or data integrity.

8. Transaction Success Rate: The transaction success rate indicates the percentage of successful transactions compared to the total attempted transactions. A high success rate is essential for a reliable and trustworthy banking system.

9. Error Rate: The error rate reflects the percentage of failed or erroneous transactions or system responses. A low error rate is desirable to ensure data accuracy and system reliability.

10. User Experience: Although not solely a performance metric, user experience plays a vital role in the overall success of a banking information system. A positive user experience is often associated with fast response times, intuitive navigation, and minimal disruptions.

7 My learnings

Throughout my training, I've been exposed to a vast range of information on various topics, including banking information systems. I've learned about the fundamental principles of banking, the components of banking systems, security protocols, regulatory requirements, and best practices for designing and testing such systems. I've gained knowledge about different banking processes, transaction types, and user interactions, allowing me to understand the complexities and challenges involved in banking information systems.

8 Future work scope

There are some ideas that you could not work due to time limitation but can be taken in future.

AI-driven Personalization: Implementing advanced AI algorithms to personalize the banking experience for each customer. This could involve customized product offerings, personalized financial insights, and tailored recommendations based on individual financial goals and behavior.

Enhanced Fraud Detection: Developing more sophisticated fraud detection mechanisms using machine learning and AI to identify and prevent fraudulent activities in real-time. This could include anomaly detection, behavioral analysis, and biometric authentication.

Blockchain Integration: Exploring the integration of blockchain technology to enhance security, transparency, and efficiency in banking transactions, particularly for cross-border payments and settlement processes.

Biometric Authentication: Integrating biometric authentication methods (e.g., facial recognition, fingerprint scanning) to strengthen security and streamline the login and authorization processes for customers.

Voice Banking: Introducing voice-activated banking features, allowing customers to perform banking tasks and obtain information through voice commands using natural language processing (NLP) technology.

Advanced Data Analytics: Leveraging big data and advanced analytics techniques to gain deeper insights into customer behavior, market trends, and risk assessment, leading to data-driven decision-making.

Open Banking APIs: Developing and offering open banking APIs to enable third-party developers to build innovative financial applications and services, fostering a vibrant banking ecosystem.

Mobile Payment Innovations: Exploring new mobile payment solutions, such as contactless payments, peer-to-peer transfers, and in-app payment options, to enhance convenience and security for customers.

Social Banking Integration: Integrating social media platforms with banking services to offer personalized financial advice, social payment options, and improved customer engagement.

Real-time Transaction Monitoring: Implementing real-time transaction monitoring and alerting mechanisms to provide customers with instant notifications of account activities, enhancing security and fraud prevention.