Industrial Internship Report on

” Grocery delivery application”

Prepared by

Shamyuktha K

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 (Tell about ur Project)

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

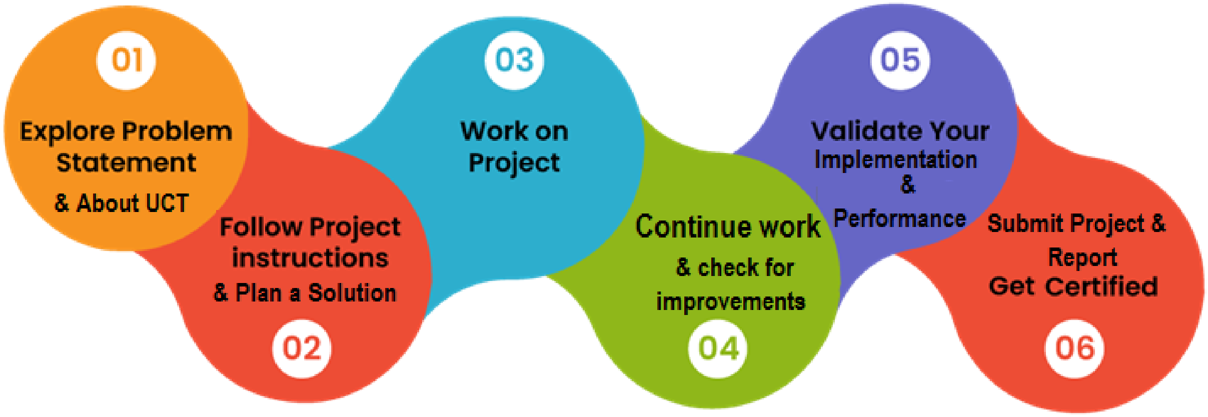
Summary of the whole 6 weeks' work.

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..

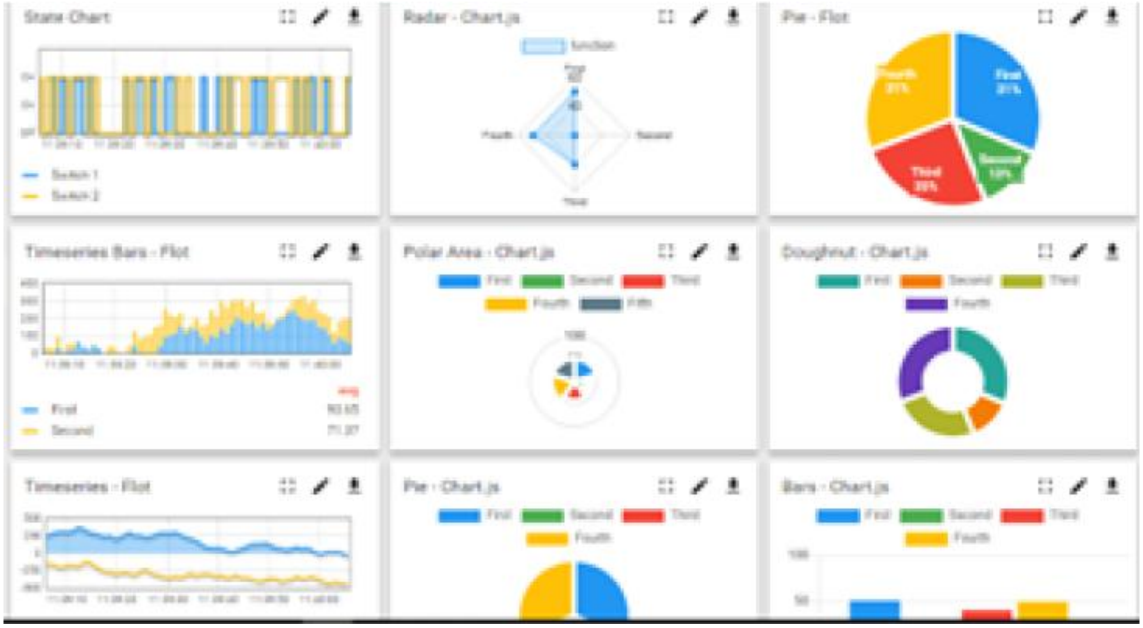
i. UCT IoT Platform (uct Insight)

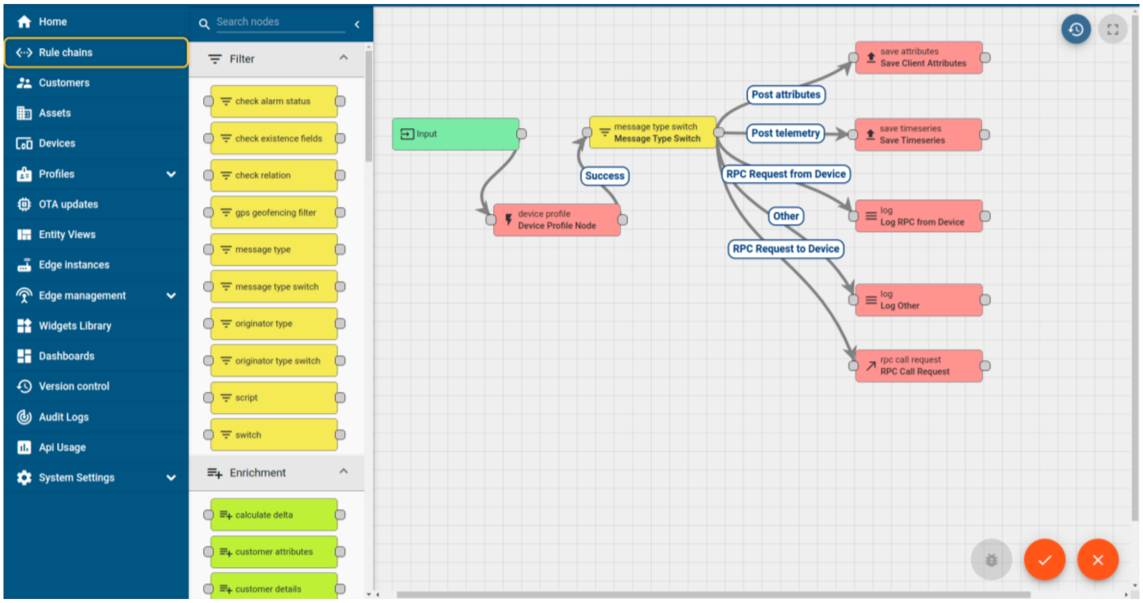
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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





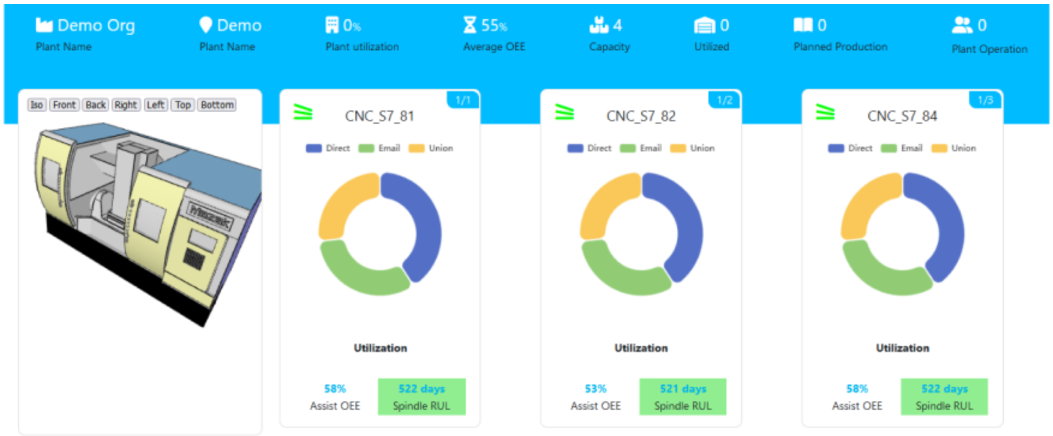
ii. Smart Factory Platform (FACTORY WATCH)

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)

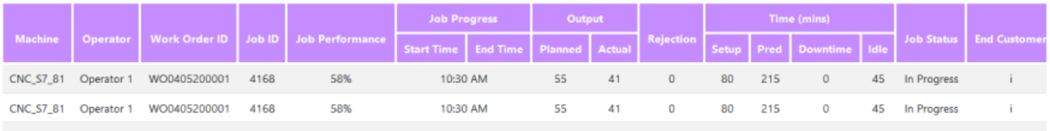
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 (1/2)

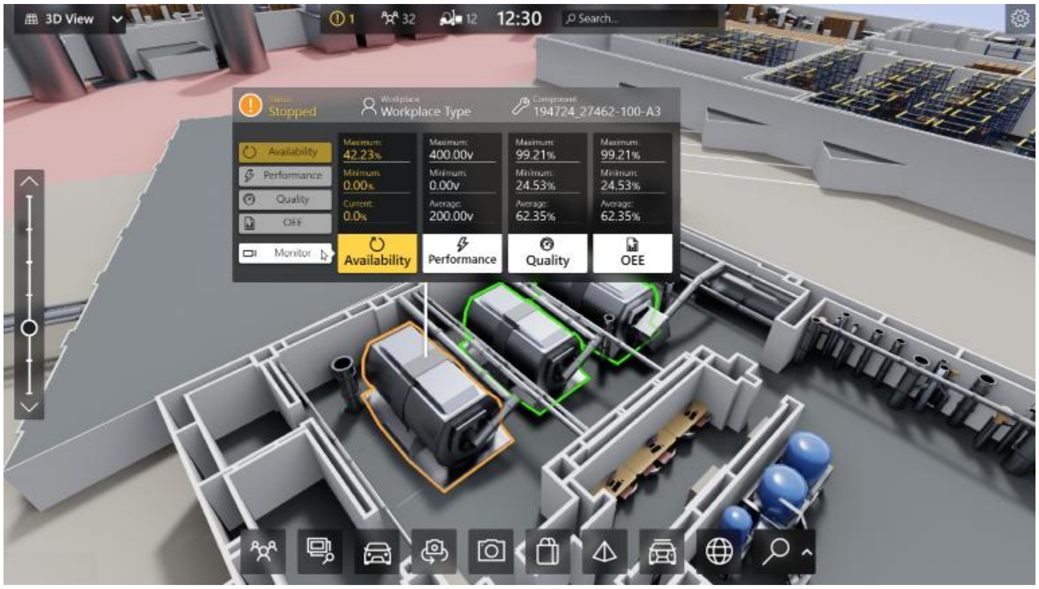
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 (1/3)

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

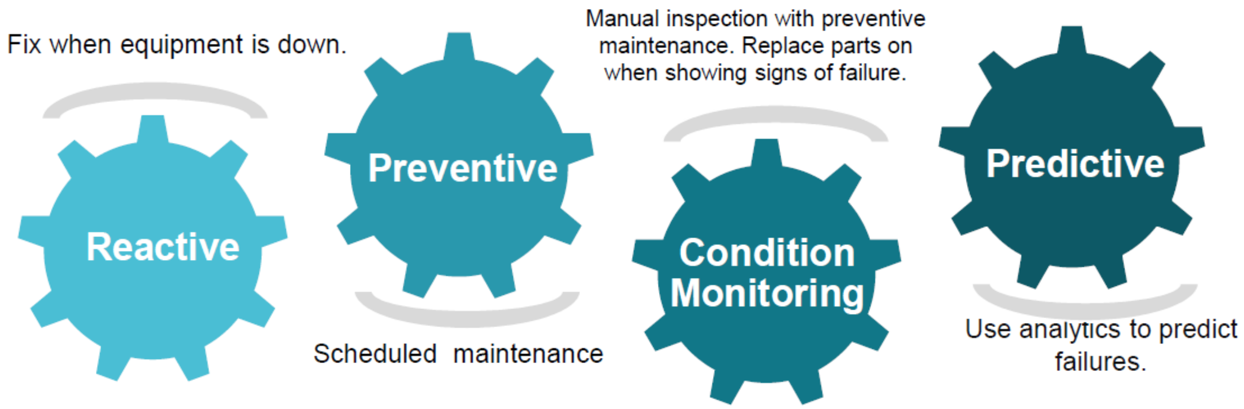
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

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

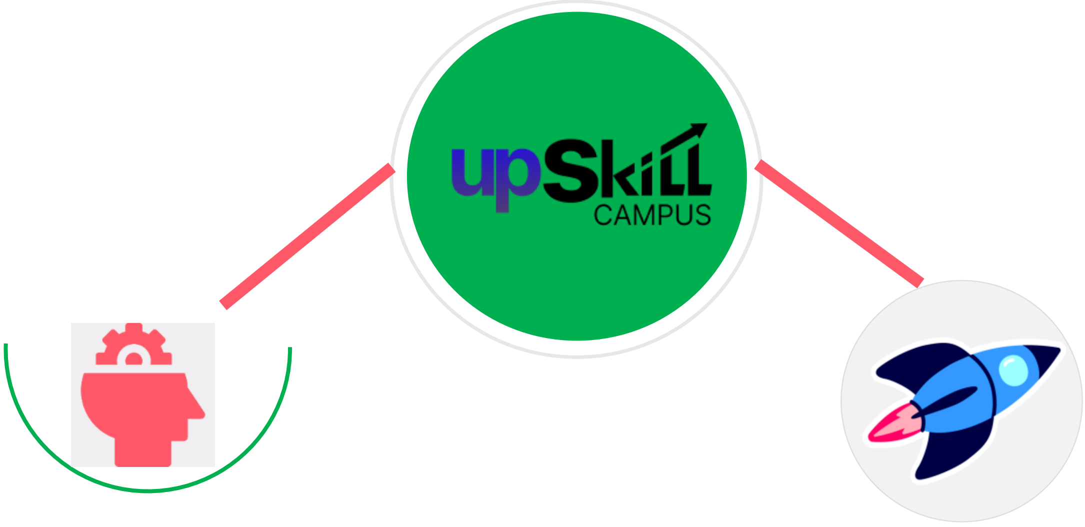
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


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.

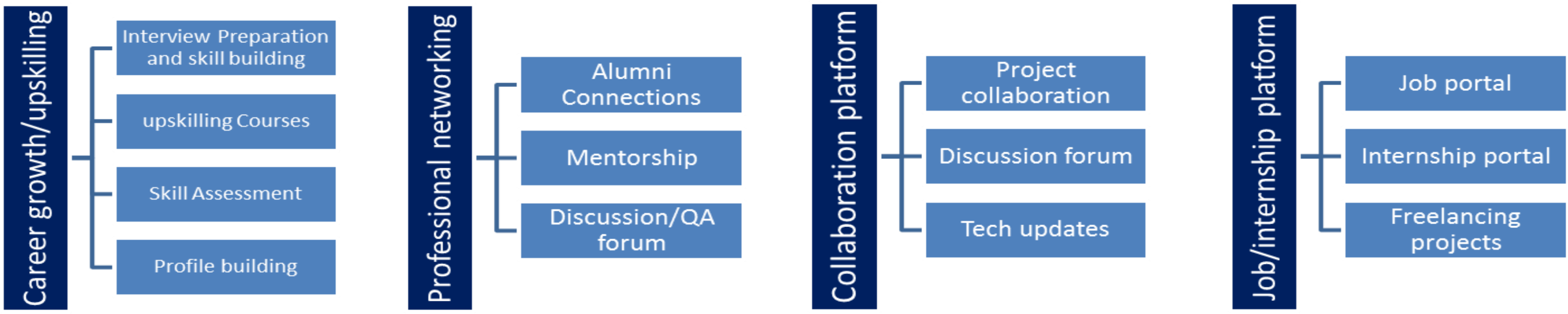
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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] Craig Walls, *Spring in Action*, 6th Edition, Manning Publications, 2022.

[2] Alex Banks and Eve Porcello, *Learning React: Modern Patterns for Developing React Apps*, 3rd Edition, O'Reilly Media, 2022.

[3] PostgreSQL Global Development Group, *PostgreSQL 16 Documentation*. Available at: https://www.postgresql.org/docs/

[4] Redis Documentation, *Caching and Performance Optimization*. Available at: https://redis.io/docs

[5] Docker Documentation, *Docker Overview*. Available at: https://docs.docker.com/

2.6 Glossary

Terms	Acronym

3 Problem Statement

The client (a departmental store) faced challenges in scaling their **grocery ordering process**. Traditional manual ordering and phone-based systems lacked:

- Real-time stock visibility.
- Slot-based delivery management.
- Personalized shopping experience.
- Digital payments integration.

The goal was to design and implement a **Grocery Delivery Application** that solved these problems, making ordering seamless for customers while improving operational efficiency for vendors.

4 Existing and Proposed solution

Existing Systems:

Generic e-commerce apps (e.g., Amazon, BigBasket) covered basic product listing and ordering.

However, they lacked:

- Advanced **loyalty programs** and **auto-replenishment**.
- **Slot-based delivery management** with real-time capacity checks.
- Unified support for groceries, auto parts, and restaurants.

Proposed Solution

Our **Grocery Delivery Application** proposed:

A **full-stack system** with:

- User-friendly frontend (React, Android).
- Backend (Spring Boot REST APIs).

- Database (PostgreSQL, Redis).

Features included:

- User registration/login, catalog browsing, search and filters.
- Shopping cart and checkout.
- Slot booking for convenient delivery.
- Integration with multiple payment gateways.
- Loyalty points, subscriptions (auto-replenishment), and personalized recommendations.

- Vendors/Restaurants could manage **menus, availability, and pricing**.

Customers could **rate products and upload photos** for community feedback.

4.1 Code submission (Github link):

https://github.com/ShamyukthaK/UpSkill_Intern

4.2 Report submission (Github link) :

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)

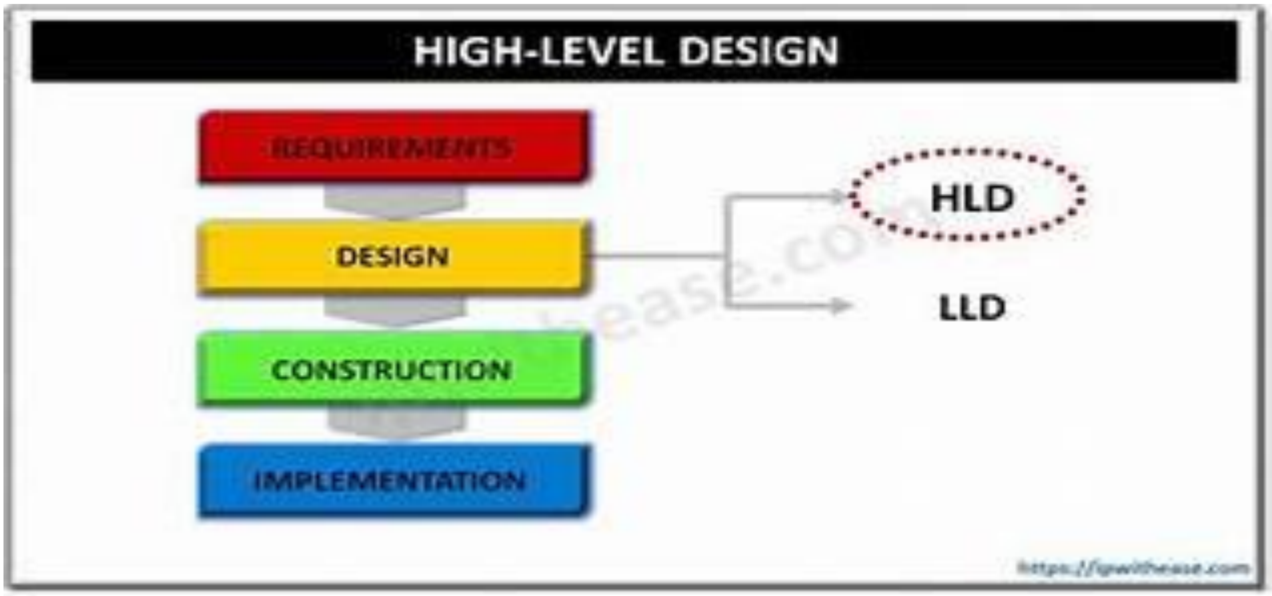


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram (if applicable)

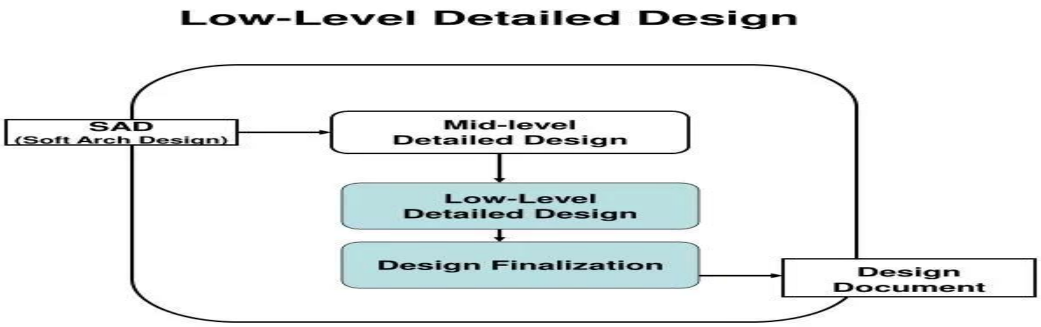


Figure 2: LOW LEVEL DIAGRAM OF THE SYSTEM

5.3 Interfaces (if applicable)

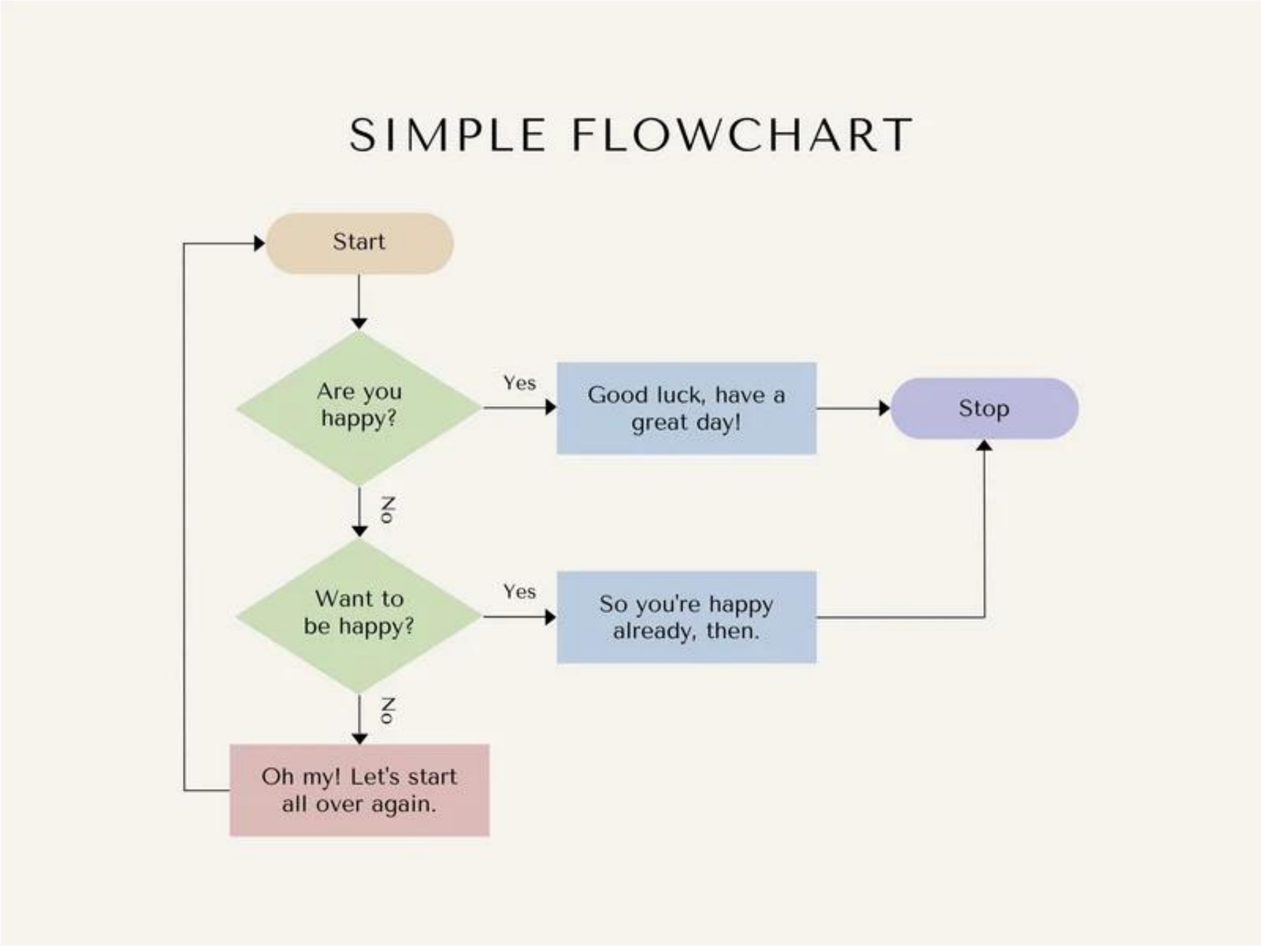

SIMPLE FLOWCHART
Start
Are you happy?
Yes
Good luck, have a great day!
Stop
No
Want to be happy?
Yes
So you're happy already, then.
No
Oh my! Let's start all over again.

Figure 3: FLOW CHART OF THE SYSTEM

6 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.

6.1 Test Plan/ Test Cases

The purpose of this test plan is to validate the functionality, performance, reliability, and usability of the Grocery Delivery Application. The goal is to ensure the application meets all requirements, works under real-world conditions, and provides a smooth user experience.

6.2 Test Procedure

The test procedure describes the **step-by-step method** followed to execute the test cases and validate the system functionalities. It ensures that the testing process was systematic, repeatable, and covered all critical modules of the Grocery Delivery Application.

1. Environment Setup

- Installed **Docker Compose** to run backend (Spring Boot), frontend (ReactJS), PostgreSQL, and Redis.
- Populated the database with **sample grocery items** (e.g., milk, rice, vegetables, fruits, oil).
- Configured payment gateway sandbox (Razorpay test mode) for safe testing.
- Used **Postman** for API testing, **JMeter** for performance testing, and **Selenium** for UI testing.

2.**Test Execution Flow**

- **User Module Testing**
    - Verified user registration and login with valid and invalid data.
    - Checked password encryption and JWT authentication for secured sessions.
- **Catalog Module Testing**
    - Tested product browsing, pagination, and category filters.
    - Verified search results for keywords like “Milk” and “Rice”.

- Checked image rendering and product details display.
- Cart & Checkout Testing
- Added products to cart, updated quantities, and removed items.
- Verified cart total calculation, including taxes and discounts.
- Tested checkout with both valid and invalid inputs.

3. **Delivery Slot Testing**
    - Selected available slots during checkout.
    - Simulated multiple users booking the same slot concurrently.
    - Verified that slot reservation was atomic (no overbooking).
4. **Payment Testing**
    - Executed successful payments using test UPI/credit card details.
    - Simulated failed payments with invalid data.
    - Checked webhook updates from payment gateway reflected correctly in the order system.
5. **Order Lifecycle Testing**
    - Placed orders and tracked them from **Created → Paid → Packed → Out for Delivery → Delivered**.
    - Cancelled an order before slot cut-off and verified refund initiation.
6. **Loyalty & Review Testing**
    - Confirmed loyalty points were credited after successful orders.
    - Submitted product reviews with ratings and images.
    - Checked that reviews were visible only after moderation (for quality control).
7. **Performance Testing**
    - Used **JMeter** to simulate 1000 concurrent users browsing the catalog and placing orders.
    - Monitored response times (API < 300ms, checkout < 500ms).

- Checked system stability under sustained load.

8. **Security Testing**

   - Attempted unauthorized API calls without JWT → received **401 Unauthorized**.
   - Tested role-based restrictions (Admin vs Customer).
   - Verified input validations prevented SQL injection and XSS attacks.

6.3 Performance Outcome

The Grocery Delivery Application was tested for **functionality, scalability, security, and usability**. The following outcomes were observed:

1. Functional Outcomes

- User registration, login, and authentication worked smoothly with JWT-based security.
- Product catalog browsing and search returned correct results with accurate filtering.
- Cart and checkout modules functioned without calculation errors.
- Delivery slot booking was successfully restricted to available capacity (no double booking).
- Payment gateway integration (Razorpay sandbox) correctly updated orders with **success** and **failure** states.
- Loyalty points and product review submissions worked as expected.

2. Performance Outcomes

- **Catalog API:** Handled 1000 concurrent requests with an average response time of **280 ms**.
- **Checkout API:** Completed transactions in under **500 ms** on average.
- **Delivery Slot Booking:** Maintained integrity under concurrency using database row-level locking; no overbooked slots recorded.
- **System Scalability:** Application maintained stability with **1000+ simulated users** in peak load testing.
- **Database Performance:** Optimized SQL queries and indexes ensured smooth retrieval of large product catalogs (100k+ records).

3. Security Outcomes

- Unauthorized requests were blocked with **401 Unauthorized** responses.
- Input validations prevented **SQL Injection** and **Cross-Site Scripting (XSS)** attacks.
- Session tokens (JWT) were securely validated with expiry handling.
- Role-based access control ensured customers, vendors, and admins had correct privileges.

4. Usability Outcomes

- The user interface was intuitive with simple navigation and responsive design.
- Customers could easily browse, add items to cart, and complete checkout in fewer than **5 steps**.
- Delivery slot picker provided a **clear and easy-to-select grid layout**.
- Average order completion time (from login → checkout → payment) was **under 3 minutes**.

5. Business Value Outcomes

- The system achieved its primary goal of creating a **seamless grocery ordering experience**.
- Delivery slot management and auto-replenishment features provided a **competitive advantage** over traditional solutions.

7 My learnings

The six-week internship on the **Grocery Delivery Application** was an enriching experience that significantly contributed to both my **technical skills** and **personal growth**.

1. Technical Learnings

- **Full-Stack Development:**
  I gained hands-on experience in developing both the frontend and backend of an application. Using **ReactJS** for the user interface and **Spring Boot** for REST APIs, I understood how different layers of an application interact seamlessly.
- **Database Design & Management:**
  I learned how to design relational schemas for products, orders, carts, users, and delivery slots using **PostgreSQL**. I also explored indexing and query optimization to handle large-scale data efficiently.
- **API Development and Integration:**
  I understood how to design and implement RESTful APIs, ensuring consistency in request/response formats, authentication, and error handling. I also integrated **third-party payment gateways** (Razorpay sandbox) to handle order payments.
- **Scalability & Performance Testing:**
  I learned how to conduct **load testing** using JMeter to simulate thousands of users, optimize database queries, and ensure that APIs respond within acceptable limits.
- **Security Implementation:**
  By applying **JWT authentication, role-based access control, and input validation**, I learned how to secure an application against unauthorized access and common web vulnerabilities like SQL Injection and XSS.

---

2. Professional & Personal Learnings

- **Problem-Solving Skills:**
  I learned to break down complex requirements into smaller, manageable modules and implement them systematically.
- **Team Collaboration:**
  Working under mentors and interacting with peers gave me insights into teamwork, version control (Git), and collaborative development practices.
- **Communication Skills:**
  Documenting the project and presenting progress improved my ability to communicate technical concepts clearly.

- **Time Management:**
  Balancing multiple tasks like development, testing, documentation, and bug-fixing within a limited internship duration taught me effective **task prioritization** and **deadline management**.
- **Industry Exposure:**
  Most importantly, this internship helped me bridge the gap between **academic learning** and **real-world industrial applications**. I now have a clearer understanding of how technologies like Java, web frameworks, and databases are applied to solve actual business challenges.
- You should provide summary of your overall learning and how it would help you in your career growth.

8 Future work scope

- Mobile app expansion with offline support.
- AI-based advanced recommendation engine.
- Integration with third-party delivery partners (e.g., Dunzo, Swiggy Genie).
- Cloud-native deployment with Kubernetes for high scalability.
- AR/VR shopping experience in future upgrades.