

## Industrial Internship Report on "Hospital Appointment System"

Prepared by  
**[B.Ganga Bhavani]**

### *Executive Summary*

This report presents the work carried out during the Industrial Internship provided by **upskill Campus** and **The IoT Academy**, in collaboration with **UniConverge Technologies Pvt Ltd (UCT)**. The internship duration was six weeks and focused on solving a real-world problem through software development.

The project titled **Hospital Appointment Management System** aims to simplify and automate the hospital appointment booking process. The system allows patients to book appointments digitally and helps hospitals manage appointment records efficiently.

This internship provided valuable exposure to industry-oriented problem solving, basic system design, implementation using Java, and technical documentation. Overall, it was a productive learning experience that improved my technical skills and professional understanding.

## **TABLE OF CONTENTS**

1	Preface.....	4
2	Introduction.....	5
2.1	About UniConverge Technologies Pvt Ltd.....	5
2.2	About upskill Campus.....	9
2.3	Objective .....	11
2.4	Reference .....	11
2.5	Glossary .....	11
3	Problem Statement .....	12
4	Existing and Proposed solution.....	13
5	Proposed Design/ Model.....	14
5.1	High Level Diagram (if applicable).....	14
5.2	Low Level Diagram(if applicable).....	14
5.3	Interfaces(if applicable).....	15
6	Performance Test.....	17
6.1	Test Plan/Test Cases.....	17
6.2	Test Procedure.....	17
6.3	Performance Outcomes.....	17
7	My learnings.....	18

8	Future work scope.....	.....
		19



## 1 Preface

Internships play an important role in shaping a student's career by providing hands-on industry exposure. This six-week internship helped me understand how real-world problems are identified and solved using software solutions.

The project was planned and executed in a structured manner, starting from understanding the problem statement, designing the solution, implementing the system, testing it, and finally preparing the documentation.

The opportunity provided by **upskill Campus** and **UniConverge Technologies Pvt Ltd** helped me gain confidence in Java programming, logical thinking, and report writing. I am thankful to all mentors and instructors who guided me throughout this internship.

I strongly recommend my juniors to participate in such internships to gain practical knowledge and improve their career prospects.

## 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.**



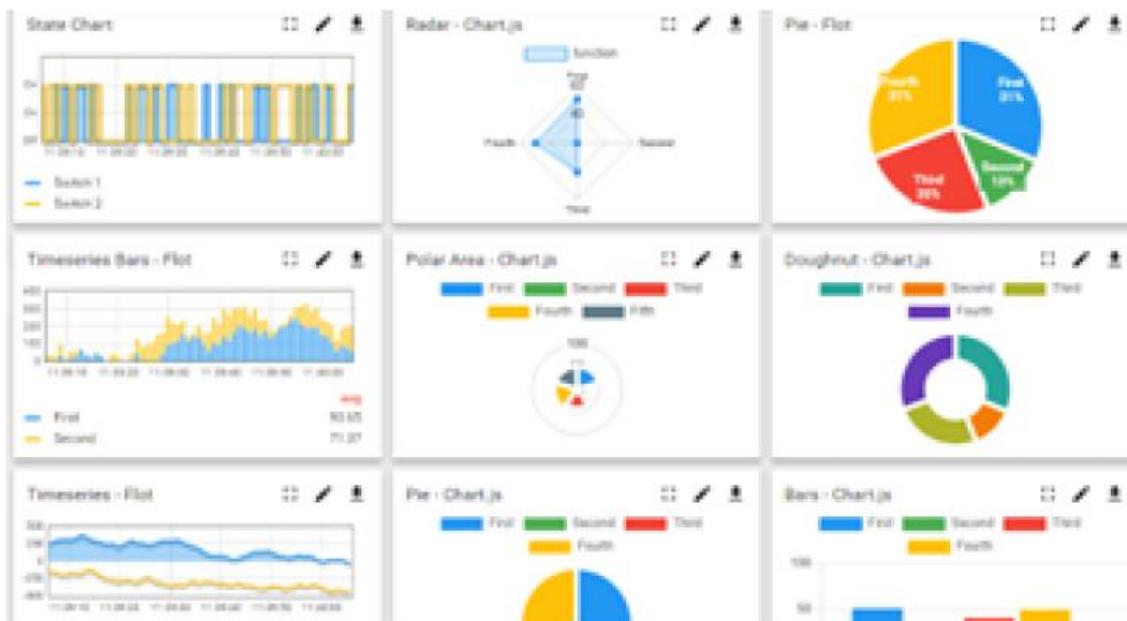
#### i. UCT IoT Platform ( **Insight**)

**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



The dashboard displays nine different chart types:

- State Chart: A bar chart comparing two series, Switch 1 (blue) and Switch 2 (yellow), across multiple categories.
- Radar - Chart.js: A radar chart with four axes: Function, Quality, Price, and Design.
- Pie - Plot: A pie chart divided into four segments: First (blue), Second (red), Third (green), and Fourth (yellow).
- Timeseries Bars - Plot: A line chart showing data over time for First (blue) and Second (yellow) categories.
- Polar Area - Chart.js: A polar area chart with four segments: First, Second, Third, and Fourth.
- Doughnut - Chart.js: A donut chart with four segments: First (teal), Second (orange), Third (light green), and Fourth (purple).
- Timeseries - Plot: A line chart showing data over time for First (blue) and Second (yellow) categories.
- Pie - Chart.js: A pie chart with four segments: First (blue), Second (red), Third (green), and Fourth (yellow).
- Bars - Chart.js: A horizontal bar chart comparing four categories: First, Second, Third, and Fourth.

**Rule chains** section (Left sidebar):

- Customers
- Assets
- Devices
- Profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management
- Widgets Library
- Dashboards
- Version control
- Audit Logs
- Api Usage
- System Settings

**Enrichment** section (Left sidebar):

- calculate delta
- customer attributes
- customer details

The main workspace shows a rule chain diagram:

```

graph LR
    Input[Input] --> DeviceProfile{Device Profile Node}
    DeviceProfile -- Success --> MessageSwitch{Message Type Switch}
    DeviceProfile -- Failure --> LogOther[Log Other]
    
    MessageSwitch -- Success --> PostAttributes[Post attributes]
    MessageSwitch -- Failure --> PostTelemetry[Post telemetry]
    
    PostAttributes --> SaveAttributes[save attributes  
Save Client Attributes]
    PostTelemetry --> SaveTimeseries[save timeseries  
Save Timeseries]
    
    PostAttributes --> RPCRequestFromDevice[RPC Request from Device]
    PostTelemetry --> RPCRequestFromDevice
    
    RPCRequestFromDevice --> LogRPC[log  
Log RPC from Device]
    RPCRequestFromDevice --> LogCallRequest[log  
RPC Call Request]
    
```

## FACTORY

### ii. Smart Factory Platform ( 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 unleashed 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.





### 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.

Fix when equipment is down.



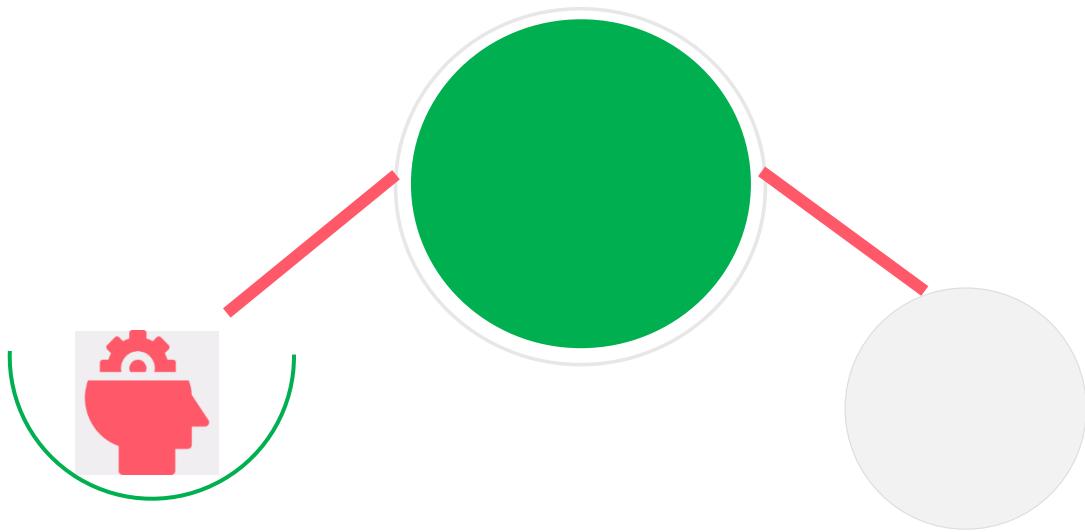
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.



## 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.

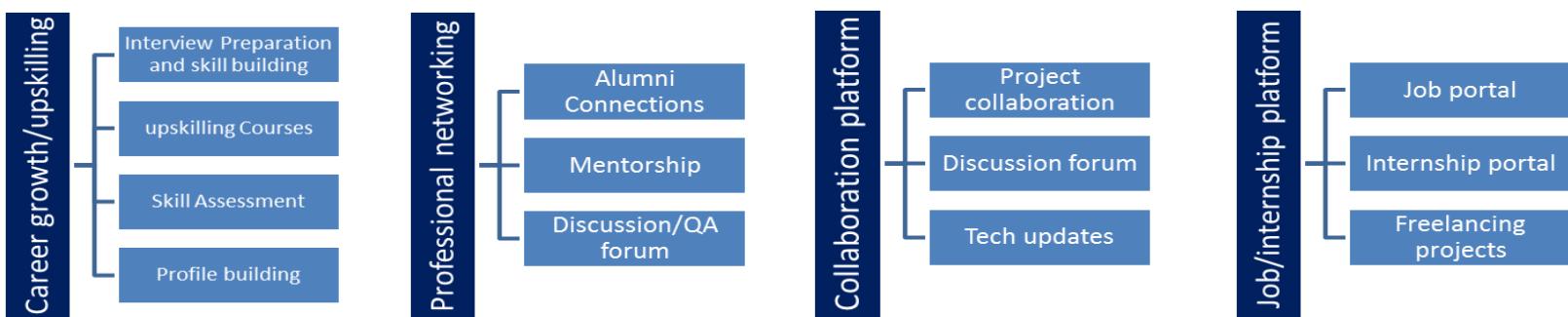
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way



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

- Java Official Documentation
- Software Engineering Lecture Notes
- Hospital Management System Articles

## 2.6 Glossary

Terms	Acronym
HMS	Hospital Management System
UI	User Interface
DB	Database

### 3 Problem Statement

Many hospitals still use manual methods such as registers or phone calls to manage patient appointments. These methods often result in long waiting times, appointment conflicts, data loss, and inefficient management.

There is a need for a simple and reliable computerized system that can manage hospital appointments efficiently and reduce manual work.

## 4 Existing and Proposed solution

### 4.1.1 Existing Solution

- Manual appointment registers
- Phone-based appointment booking
- Paper records
- High chances of human error

### 4.1.2 Proposed Solution

The **Hospital Appointment Management System** is a Java-based application that allows:

- Easy appointment booking
- Digital storage of appointment details
- Reduced paperwork
- Better organization of hospital data

### Value Addition:

The system improves efficiency, accuracy, and saves time for both patients and hospital staff.

### 4.2 Code submission (Github link)

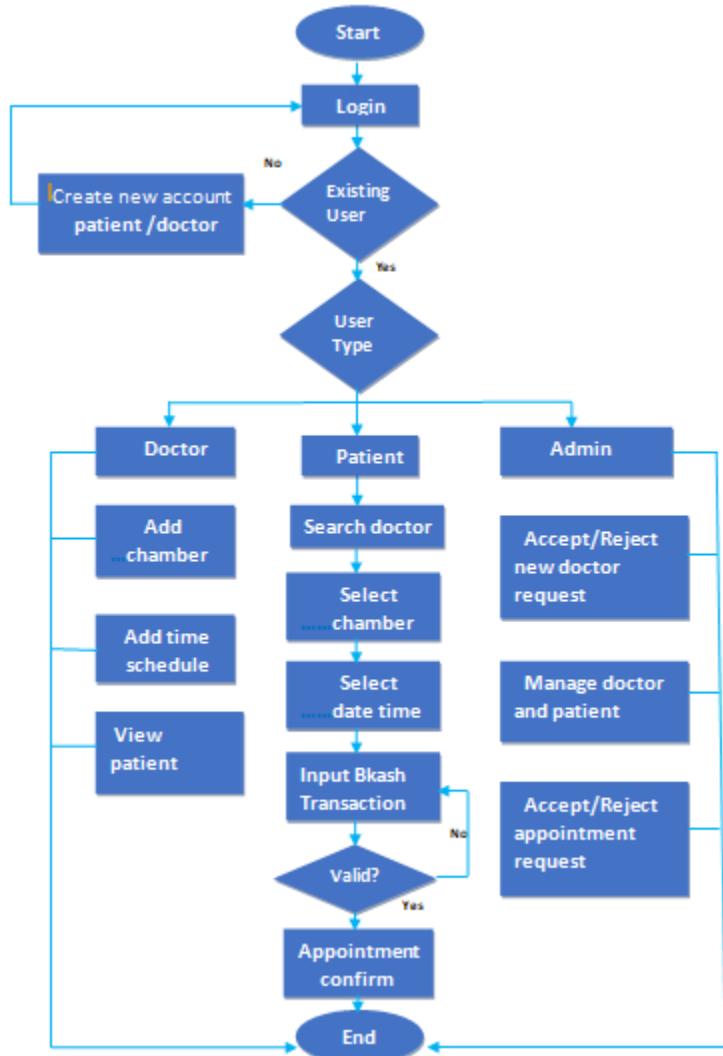
### 4.3 Report submission (Github link) : first make placeholder, copy the link.

## 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.

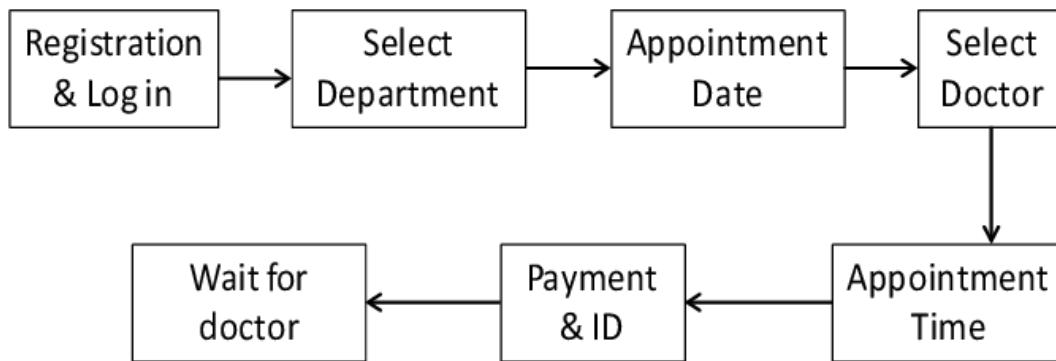
### 5.1 High Level Diagram (if applicable)

Patient → Hospital Appointment System → File Storage → Hospital Staff



## 5.2 Low Level Diagram (if applicable)

Input → Validation → Processing → Storage → Output



## 5.3 Interfaces (if applicable)

The Hospital Appointment Management System uses a **simple console-based interface** to interact with the user. The system follows a clear data flow between different components to ensure smooth operation.

### 5.3.1 Block Diagram Description

The block diagram represents the interaction between the user and the system modules. The patient provides input through the console interface. The appointment processing module validates the input and forwards it to the data storage module. The stored data can later be accessed by hospital staff.

#### Block Flow:

Patient → User Interface → Appointment Processing → File Storage → Output Display

### 5.3.2 Data Flow Description

The data flow begins when the patient enters personal and appointment details. These details are processed by the system and stored in a text file using file handling. After successful storage, a confirmation message is displayed on the console.

#### Data Flow Steps:

1. Patient enters details
2. System processes input

3. Data is written to file
4. Confirmation message shown

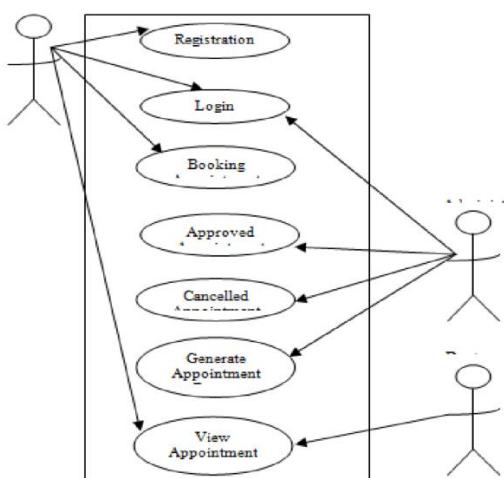
### 5.3.3 Flow Chart Description

The flow chart represents the step-by-step execution of the system. It starts with user input, followed by data validation, appointment booking, data storage, and ends with output display.

#### Flow Sequence:

Start → Enter Details → Validate Input → Book Appointment → Store Data → Display Confirmation → End

#### Flowchart Image:



### 5.3.4 Protocols

Since this is a console-based standalone application, no network communication protocols are used. All operations are performed locally within the system.

### 5.3.5 State Machine Description

The system operates in the following states:

- Idle State
- Input State
- Processing State
- Storage State
- Output State

The system transitions from one state to another based on user actions.

### 5.3.6 Memory Buffer Management

The application uses system memory temporarily to store user input during execution. Once the data is successfully written to the file, the memory is released. This ensures efficient memory usage and smooth execution.

## 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

Test Case	Input	Expected Output
TC01	Valid patient details	Appointment booked
TC02	Invalid input	Error handled
TC03	Empty fields	Prompt message

### 6.2 Test Procedure

1. Run the Java program
2. Enter patient details
3. Submit appointment
4. Verify output
5. Check stored file

### 6.3 Performance Outcome

- Fast execution
- Accurate data storage
- Reliable performance
- Suitable for small hospitals

## 7 My learnings

Through this internship, I learned:

- Java programming basics
- File handling concepts
- System design approach
- Testing methodologies
- Technical report writing

This experience helped improve my confidence and prepared me for future industry roles.

## 8 Future work scope

Future enhancements of this project include:

- Web-based hospital portal
- MySQL database integration
- Doctor and admin login
- SMS/email notifications
- Online payment system