

**NATIONAL INSTITUTE OF TECHNOLOGY  
ROURKELA, ODISHA**

**TABLE TECH  
RESTAURANT AUTOMATION SYSTEM**

**FINAL PROJECT REPORT**

**Submitted By:**

|   |            |
|---|------------|
| W | XXXXXXXXXX |
| X | XXXXXXXXXX |
| Y | XXXXXXXXXX |
| Z | XXXXXXXXXX |

**Submitted To:**

Department of Computer Science and Engineering  
National Institute of Technology  
Rourkela, Odisha

APRIL, 2023

## DECLARATION

We, the members of GROUP X declare that this report titled TABLE TECH: Restaurant Automation System is our original work and has not been submitted in whole or in parts for any other academic or professional award. All sources of information used in this report have been acknowledged and referenced respectively.

This project was undertaken as a requirement for the completion of our Software Engineering Lab at National Institute of Technology, Rourkela. The project was carried out between 07-March-2023 and 31-March-2023. During this time, we as a team were responsible for the process model selection, development of the SRS document and designing of the product. Table Tech is a restaurant automation system that aimed to streamline various restaurant operations and improve overall efficiency.

The project involved extensive research on the current restaurant management systems, identifying the key features required for the automation system, and developing a prototype. The prototype was then tested and refined to ensure that it met the specified requirements.

We would like to express our gratitude to our project supervisor Prof. XYZ for their guidance and support throughout this project. We would also like to thank our mentor Ms. XYZ for their help and efforts. We also thank the staff at XYZ RESTAURANT for providing valuable insights into a restaurant's operations, which greatly assisted in the development of the automation system.

We declare that this report represents OUR own work, and any assistance received from others has been acknowledged and appropriately referenced.

X (120CS0230) \_\_\_\_\_

Y (120CS0231) \_\_\_\_\_

Z (120CS0232) \_\_\_\_\_

A (120CS0233) \_\_\_\_\_

**DATE:** APRIL, 2023

## **ACKNOWLEDGEMENT**

This project is prepared in fulfilment of the requirement for the XYZ Lab. We owe our deepest gratitude to the Department of XYZ National Institute of Technology, Rourkela for providing us with an opportunity to work on this project.

We also extend our gratitude towards our project supervisor and mentor, Prof. XYZ and Ms. XYZ whose guidance and expertise have been invaluable in steering us towards success. We also thank other faculty members of the Department for their valuable feedback and suggestions.

Finally, we would like to thank our family and friends for their continuous support and encouragement throughout the project. We acknowledge the contributions of everyone who supported us in the creation of this project report.

Thank you all for your assistance and support.

The experience of working on this project will surely enrich our technical knowledge and also give us hands on experience of working on a project and help develop our team's skill set to a great extent.

# TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>DECLARATION</b>                                | <b>i</b>  |
| <b>ACKNOWLEDGEMENT</b>                            | <b>ii</b> |
| <b>1 INTRODUCTION</b>                             | <b>1</b>  |
| 1.1 Background . . . . .                          | 1         |
| 1.2 Motivation . . . . .                          | 1         |
| 1.3 Objectives . . . . .                          | 1         |
| 1.4 Problem statement . . . . .                   | 2         |
| 1.5 Scope of Project . . . . .                    | 2         |
| <b>2 REQUIREMENT ANALYSIS &amp; SPECIFICATION</b> | <b>1</b>  |
| 2.1 Feasibility Study . . . . .                   | 1         |
| 2.1.1 Market Analysis . . . . .                   | 1         |
| 2.1.2 Technology Assessment . . . . .             | 1         |
| 2.1.3 Operational Requirements . . . . .          | 2         |
| 2.1.4 Financial Analysis . . . . .                | 2         |
| 2.1.5 Risk Assessment . . . . .                   | 2         |
| 2.2 Selection of Process Model . . . . .          | 2         |
| 2.2.1 Process Models . . . . .                    | 2         |
| 2.2.2 Why Waterfall . . . . .                     | 3         |
| 2.2.3 Why Not . . . . .                           | 3         |
| 2.3 Software Requirements Specification . . . . . | 4         |
| 2.3.1 Introduction . . . . .                      | 4         |
| 2.3.2 Overall Description . . . . .               | 5         |
| 2.3.3 Specific Requirements . . . . .             | 7         |
| 2.3.4 Appendices . . . . .                        | 11        |
| <b>3 DESIGN SPECIFICATION</b>                     | <b>12</b> |
| 3.1 Use Case Diagram . . . . .                    | 13        |
| 3.2 Data Flow Diagram . . . . .                   | 13        |
| 3.2.1 Data Dictionary . . . . .                   | 15        |
| 3.3 Class Diagram . . . . .                       | 15        |
| 3.4 Sequence Diagram . . . . .                    | 16        |
| 3.5 Activity Diagram . . . . .                    | 20        |
| 3.6 State Chart Diagram . . . . .                 | 22        |
| <b>4 IMPLEMENTATION DETAILS</b>                   | <b>23</b> |

|          |                                  |           |
|----------|----------------------------------|-----------|
| 4.1      | Technology Stack . . . . .       | 23        |
| 4.2      | System Architecture . . . . .    | 23        |
| 4.3      | User Interface . . . . .         | 23        |
| 4.4      | Integration . . . . .            | 24        |
| 4.5      | Security . . . . .               | 24        |
| 4.6      | Testing and Deployment . . . . . | 24        |
| <b>5</b> | <b>CONCLUSION</b>                | <b>25</b> |
| <b>A</b> | <b>APPENDIX</b>                  | <b>26</b> |
| A.1      | References . . . . .             | 26        |
| A.2      | Project Timeline . . . . .       | 26        |

# **1. INTRODUCTION**

In this report, we will explore the life cycle of our Restaurant Automation System - TABLE TECH. The objective of a restaurant automation system is to reduce costs associated with labor and inventory management. Automation systems can help restaurants to optimize their inventory and reduce food waste, leading to significant cost savings.

## **1.1. Background**

In recent years, technology has advanced significantly, and as a result, the restaurant industry has undergone a transformation. One significant change that has taken place is the introduction of restaurant automation systems. These systems have become increasingly popular in the restaurant industry as they offer various benefits such as improving operational efficiency, reducing costs, enhancing customer experience, and increasing revenue.

## **1.2. Motivation**

Our motivation to work on this software product was because of the new world adoption of restaurant automation systems where it has become a necessity in today's fast-paced environment, where customers demand quick service and accurate orders. It has also become a way for restaurants to stand out in a highly competitive market by decreasing their downtime because of raw product shortage. However, despite the benefits, some restaurants are still hesitant to adopt these systems due to the perceived cost and complexity of implementation.

## **1.3. Objectives**

1. Improve operational efficiency and enhance the overall customer experience.
2. Automation of various processes in a restaurant, such as ordering, billing, inventory management, and customer management.

#### **1.4. Problem statement**

A restaurant owner wants to computerize his order processing, billing and accounting activities. He also expects the computer to generate statistical reports about sales of different items. A major goal of this computerization is to make supply ordering more accurate so that the problem of excess inventory is avoided as well as the problem of non-availability of ingredients required to satisfy orders for some popular items is minimized. The computer should maintain the prices of all the items and support changing the prices by the manager. Whenever any item is sold, the sales clerk would enter the item code and the quantity sold. The computer should generate bills whenever food items are sold. Whenever ingredients are issues for preparation of food items, the data is to be entered into the computer. Purchase orders are generated on a daily basis, whenever the stock of any ingredient falls below a threshold value. The computer should calculate the threshold value for each item based on the average consumption of the ingredients for the past three days and assuming that a minimum of two days stock should be maintained for all ingredients. Whenever the ordered ingredients arrived, the invoice data regarding the quantity and price is entered. If sufficient cash balance is available, the computer should print cheques immediately against invoice. Monthly sales receipt and expenses data should be generated whenever the manager would request to see them.

#### **1.5. Scope of Project**

To gain an insight into the scope of our project, let us first understand what the term scope of a project means in software development, the scope refers to the boundaries and limitations of a project. It also defines the features, functions, and requirements of the software being developed. The scope of a software project describes what the software will and will not do, and what is included and excluded from the project.

Our project report for TABLE TECH aims to design and develop an efficient and user-friendly system that automates various processes involved in the functioning of a restaurant.

The project aims to deliver the following key features:

- **Order Processing::** The system will automate the order processing process by providing a user-friendly interface for taking orders, managing orders, and communicating with the kitchen staff.

- **Inventory Management:** The system will track the inventory levels of various food items, manage stock levels, and generate alerts when the stock level falls below the minimum threshold.
- **Staff Management:** The system will manage the schedules of the staff, track their attendance, and manage their payroll.

The boundaries and limitations of TABLE TECH can also be defined by its scope, which also outlines the features, functions, and requirements of the system. Some of these boundaries and limitations are:

- **Menu Limitations:** The restaurant automation system may have limitations in terms of the types of menus it can support, such as complex menu structures or multiple menu languages.
- **Hardware Requirements:** The restaurant automation system may require specific hardware or software configurations, which may limit its compatibility with certain systems or platforms.
- **Staff Training:** The restaurant automation system may require staff training to use it effectively, which can be a limitation for restaurants with high staff turnover or limited training resources.
- **System Integration:** The restaurant automation system may have limitations in terms of its ability to integrate with other systems, such as accounting or payroll software.

It is important to identify and understand the boundaries and limitations of a restaurant automation system to ensure that the system meets the needs of the restaurant and its customers, and to avoid unexpected issues and costs. By defining the boundaries and limitations of the system, the restaurant can develop a clear understanding of what the system can and cannot do, and make informed decisions about its implementation and use.

Overall, the project report will provide a comprehensive understanding of the restaurant automation system. The report will serve as a valuable resource for restaurant owners, managers, and researchers interested in restaurant automation.



## **2. REQUIREMENT ANALYSIS & SPECIFICATION**

This section of the project report is the most critical element as, it provides a foundation for the entire project. It ensures that all stakeholders are aligned on the project's objectives, scope, and deliverables, and it provides a clear road map for the project team to follow.

This section is further divided into 3 sister sections:

- Feasibility Study
- Model Selection
- SRS

### **2.1. Feasibility Study**

A feasibility study is the first stepping stone into the development of any project, including our restaurant automation system; Table Tech. It involves assessing the potential for the project to be successful, which in turn includes evaluating the market, technology, financial aspects, and operational requirements.

#### **2.1.1. Market Analysis**

We perform a small scale analysis of the restaurants of Rourkela. Rourkela being a city which is in the prime of its development phase with the hosting of the hockey world cup and the opening of the airport. Rourkela boasts to harbour a large student population too. These factors act as a perfect indicator towards the success of Table Tech in the restaurant industry of Rourkela.

#### **2.1.2. Technology Assessment**

As, per our analysis of the market; the technology assessment can be understood by breaking the whole restaurant market into two sub-parts. One being the large scale ones, the ones which already own some kind of automation system. The wave of automation hasn't hit the small scale restaurants yet. So, making this sect of the market familiar with the technology is going to be a task we are going to face in the implementation of the system into the market.

### **2.1.3. Operational Requirements**

The technology assessment helps us realise the fact that implementation of Table Tech into the sect of large scale restaurants in Rourkela will be comparatively easier than in smaller scale restaurant because of the technological disconnect. The large scale restaurants would prefer our system as, it'll give them the provision of reducing their restaurant's downtime with the help of inventory management. A considerable amount of time and effort will have to be spent on the familiarising the smaller scale restaurants to the automation system.

### **2.1.4. Financial Analysis**

The product will be able to garner the return of investment within the first 30 users. We will charging them a one time premium for purchasing the software.

### **2.1.5. Risk Assessment**

The risk assessment of the market tells us that the success of our product is majorly dependent on the fact that how do the smaller scale restaurants receive the product.

## **2.2. Selection of Process Model**

The software life cycle process model is a framework that outlines the various stages involved in the development of a software application. So, choosing a life cycle process model is the stepping stone into the development of a software product.

### **2.2.1. Process Models**

The choice of software development process model for a restaurant automation system depends on various factors, such as project size, requirements, team size and available resources. The waterfall model is a traditional sequential development approach, where each phase must be completed before moving on to the next phase.

### 2.2.2. Why Waterfall

Here are some reasons why the waterfall model is the best choice for developing our restaurant automation system, Table Tech:

- **Clear requirements:** The waterfall model is best suited for Table Tech; as the problem statement provides us with a clear set of requirements. In our case of a restaurant automation system, the requirements are relatively straightforward, such as automating the order-taking process, managing inventory and generating reports.
- **Sequential phases:** The waterfall model follows a sequential approach, where each phase must be completed before moving on to the next one. This approach can help ensure that each phase is completed thoroughly before moving on to the next phase, which can reduce the chances of errors and rework. This model perfectly suits our team and project deadline too. We couldn't have followed an iterative approach for this project, as we have to make one final submission.
- **Predictable outcomes:** The waterfall model can provide more predictable outcomes in terms of time, cost, and scope. Since each phase is completed before moving on to the next phase, it can be easier to estimate the time and resources required for each phase. All this makes the waterfall model a perfect match for our team and project .
- **Document-driven:** Table Tech is the project submission for our final grade in Software Engineering Lab thus, it being a fully documented project is understood. The waterfall model is also heavily reliant on documentation, which matches with our submission requirement too. Documentation also helps ensure that everyone involved in the project is on the same page and can fully grasp the concept of the project.

### 2.2.3. Why Not

Every coin has two sides thus, we can't forget to consider that the waterfall model has some limitations too such as:

- Inflexibility
- Lack of adaptability

If the project requirements change or there is a need to iterate on certain features, it can be challenging to make changes once a phase is completed. Therefore, it's important to carefully evaluate the project requirements and constraints before deciding on a software development methodology but this very limitation doesn't act as one in our case.

## **2.3. Software Requirements Specification**

### **2.3.1. Introduction**

#### **1. Purpose**

The purpose of this Restaurant Automation System (RAS) is to provide a comprehensive and automated solution for restaurant management. Table Tech focuses on the idea of reducing the downtime of restaurants because of raw materials shortage. This document aims to capture the system requirements and features particularly of the functionalities mentioned in the problem statement. This section of the document presents a detailed explanation of the objectives, features, user interface and applications of RAS in real life. It will also describe how the system will perform and under which surroundings it must operate. Both the stakeholders and the developers of the system can benefit from this section of the document. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

#### **2. Scope**

Table Tech will be a local software that will be accessible from any device with the correct hardware requirements. The system will have multiple user roles with various levels of access, including restaurant managers and waiter staff. The system might also further integrate with third-party services such as payment gateways. The RAS can take orders, manage menu items, and can-do inventory and staff management.

#### **3. Definitions, Acronyms and Abbreviations**

The section will consist of three parts:

- **Definitions**

Waiter: A person who is responsible for taking orders. Limited privileges.

Manager: A person who is responsible for managing the whole restaurant. Privileged.

- Acronyms
  - RAS: Restaurant Automation System
  - POS: Point of Sale
  - ERP: Enterprise Resource Planning
  - API: Application Programming Interface
  - SDK: Software Development Kit
- Abbreviations
  - Table Tech: Name of our RAS.

#### **4. References**

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

#### **5. Overview**

The document will mostly consist of two parts:

- Overall Description
- Specific Requirements

Overall description describes the major components of the system, assumptions and dependencies of the system, while specific requirements describes the functions of the system and their roles in the system and the constraints faced by the system.

##### **2.3.2. Overall Description**

###### **1. Product Perspective**

Table Tech will attempt to replace the traditional manual ordering process and will be a new self-contained software system that consists of two parts: one local application and the other is a database. The application will be used for placing orders, interacting with the inventory, while the database will be used for storing the inventory and ordering related information about the food items like pending and complete order queues, invoice data and statistics of items.

The main perspective of the product is to reduce the downtime of restaurants because of scarcity of raw materials. It covers the basic functionality of keeping a database of the

restaurant staff and the menu items but in addition it also keeps a check on the inventory to automate the ordering process when the inventory falls below a certain threshold value.

## **2. Product Functions**

Table Tech will have the following functions:

- **Order processing**

The system will allow waitstaff to take orders and process payments. The orders will be sent to the kitchen for preparation.

- **Inventory management**

The system will check and verify if the threshold value set for the inventory is not ambiguous.

- **Staff management**

The system will manage employee schedules, tips and payroll.

## **3. User Characteristics**

Table Tech will have two user classes:

- **Waiter**

The waiter will use Table Tech to take orders and update the inventory.

- **Manager**

The manager will update orders to the menu, give discounts, check inventory and generate reports.

## **4. Constraints**

The application should be compatible and must run smoothly on various platforms, operating system configurations like windows 10, macOS, Linux etc.

There are some constraints which cost more for the system. If those constraints can be overcome, then this whole system will perform best:

- Faster server system such as LINUX Server.
- Other Languages for different sects of people

## 5. Assumptions & Dependencies

The RAS assumes that the restaurant has an existing POS system and ERP system that can be integrated with the RAS. The system also assumes that customers have internet access and have the minimum technical acumen.

### 2.3.3. Specific Requirements

#### 1. External Interfaces

This part of the SRS discuss about the interfaces of Table Tech.

- **User Interface**
  1. Place Order: In this screen, the system shows a list of cards of dishes. Each dish will have an Image and its price per serving.
  2. Login page: Login with credentials or signup.
  3. Menu page: Shows the available dishes in the restaurant.
  4. Order Page: Shows the items added to the order
- **Manager Interface** The manager will have a screen where he will get a notification whenever an order is completed. The System will notify the manager about the order number. The manager also has a screen where all orders are listed, an option to give discounts and a status button to mark the order as paid..
  1. Home page: Home page of the application.
  2. Manager login page: Login with credentials for managers.
  3. Menu items edit page: Manager can edit menu items here.
  4. Order processing page : Manager will process the order.
  5. Inventory Management page : Manager will process and manage the ingredients and inventory.
- **Hardware Interface**

There are three external hardware devices used by Table Tech. These devices are surface computers, wireless tablets, and touch displays. The devices must also have good industrial design aesthetics. They should be fully capable computers that can use textual data from the server along with local UI.

All order and transaction records will be stored on the big query server. The hardware device takes information from the RAS and processes the information to display. It also provides user input information to the RAS.

- **Software Interface**

For Database Services System shall use any suitable DBMS of latest version that is compatible with our hardware interfaces, we used Big Query. The DBMS must be able to provide requests with low latency, data concerning the restaurant's menu, and available inventory requirements. Additionally, it should take and archive data provided to it by the RAS. This data will include records of all orders and transactions executed by the RAS. The DBMS must store all data such that it can be used for accounting, as well as accountability. The technology stack for Table Tech is chosen based on the requirements of the system. For Front end we used python the backend part we used Big Query.

## **2. Functions**

This section will consist of the functions of Table Tech:

- **Order Management**

The system shall allow customers to place orders. The system shall send orders to the kitchen for preparation. The system shall allow restaurant staff to view and manage.

- **Inventory Management**

The system will check and verify if the threshold value set for the inventory is not ambiguous.

- **Staff Management**

The system will manage employee schedules, tips and payroll.

## **3. Performance Requirements**

Table Tech must be interactive, and the delay involved must be minimal. So, in every action response of the software, there is no immediate delay. The system should be fast and responsive, enabling quick order taking, processing, and delivery to the kitchen.

It should also provide quick updates on order status and availability of menu items. Also, successful connection delay should be low for effective real-time communication when connecting to a database (MySQL) server.



#### **4. Logical Database Requirements**

The logical database requirements for a restaurant automation system includes:

- **Menu**  
The database should contain all menu items, including their names, descriptions, prices, ingredients, and allergens.
- **Orders**  
The database should store all orders placed by customers, including the date and time of the order, the items ordered, any modifications or special requests, and the table or delivery information.
- **Employee**  
The database should store information about the restaurant staff, including their name, contact information, job role, and schedule.
- **Inventory**  
The database should store information about raw materials.

#### **5. Design Constraints**

The design constraints of Table Tech include:

- **Budget**  
The cost of the automation system must be considered in the design phase, as the restaurant may have a limited budget for technological investments.
- **User interface**  
The system should be easy to use and understand by the staff, as well as the customers. The interface should be intuitive and require minimal training.
- **Integration with existing systems**  
The automation system should be designed to integrate with existing systems, such as POS systems and inventory management systems. This will help ensure that data is shared between systems and reduce the likelihood of errors.

## 6. Software System Quality Attributes

The quality attributes of Table Tech include:

- **Availability**

Ensure that the system is up and running most of the time and the database server is not down for more than a few minutes to avoid inconvenience for the customers. Fault tolerance must be considered in the system design so that the system is up and running without too much downtime.

- **Adaptability**

The design should work on multiple smartphone devices such as iOS, Android, Windows.

- **Usability**

The design should be simple and fast as the key measure of success is to reduce the downtime of the restaurant. A complicated system would defeat the purpose of the software.

- **Correctness**

The bill generated by the application must be accurate and the orders placed should exactly be the same as the user has selected.

- **Portability**

Software can be easily installed on devices and will run smoothly according to the requirement.

## 7. Object Oriented Models

- **Menu**

A menu can be modeled as an object with attributes such as name, description, and a list of items. Each menu item can be modeled as a separate object with attributes such as name, description, price, and ingredients. The menu object can have methods to add or remove menu items and to display the menu.

- **Order**

An order can be modeled as an object with attributes such as the table number, the list of items ordered, and the total price. Each item in the order can be modeled as a separate object, which is linked to the menu item object. The order object can have methods to add or remove items, calculate the total price, and print a receipt.

- **Employee**

An employee can be modeled as an object with attributes such as name, role, and employee ID. The employee object can have methods to perform various tasks such as taking orders, managing inventories.

#### **2.3.4. Appendices**

##### **1. Glossary**

Order: A request for food or drinks. Inventory: The stock of food and drinks available in the restaurant. Report: A summary of sales, inventory, and other key metrics.

##### **2. References**

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

### **3. DESIGN SPECIFICATION**

The design specification of a restaurant automation system involves identifying the requirements and functionalities of the system to be developed. In this section of the report we complete this very task by developing different diagrams.

The system should have an intuitive and user-friendly interface that is easy to use for both customers and staff. The system should be designed to handle high volumes of transactions and should be scalable to accommodate growth in the number of customers and locations.

The system should also be designed to integrate with other systems such as payment gateways, kitchen display systems, and loyalty programs.

We understand all these requirements better by developing the following diagrams of our system:

- Use Case Diagram
- Data Flow Diagram
- Class Diagram
- Sequence Diagram
- Activity Diagram
- State Chart Diagram

### 3.1. Use Case Diagram

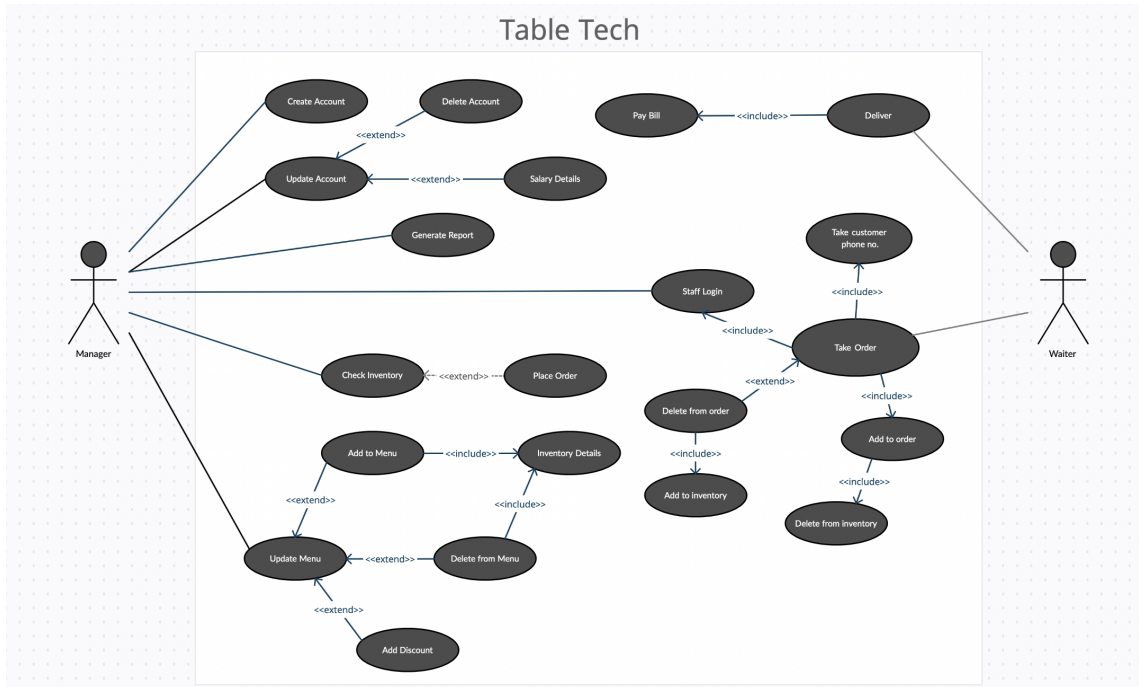


Fig 1: Use Case Diagram of Table Tech

### 3.2. Data Flow Diagram

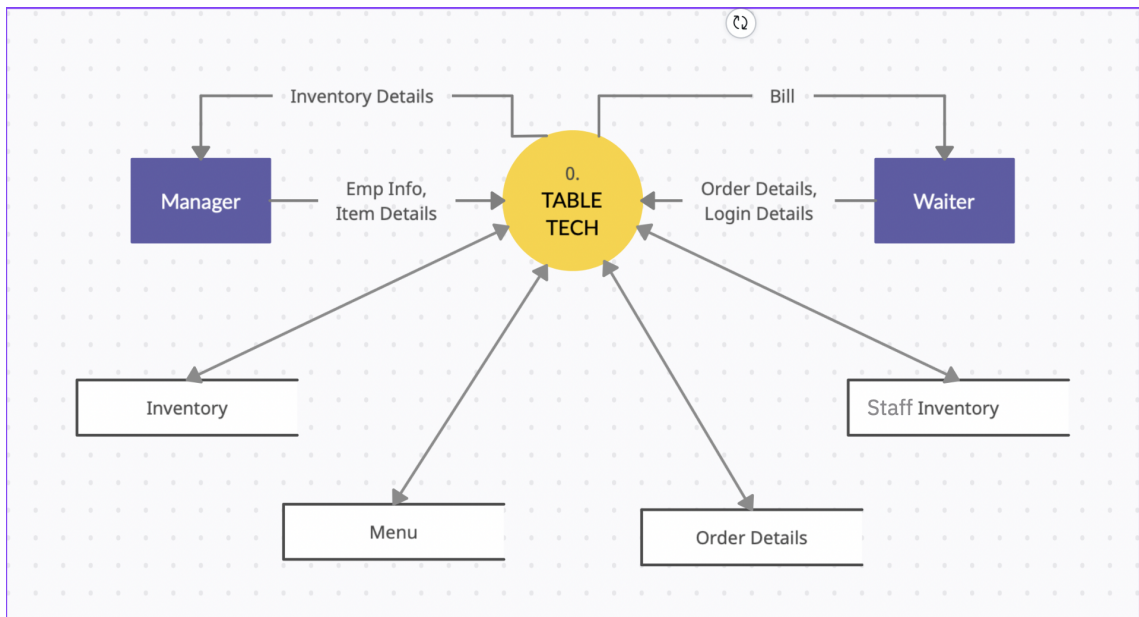
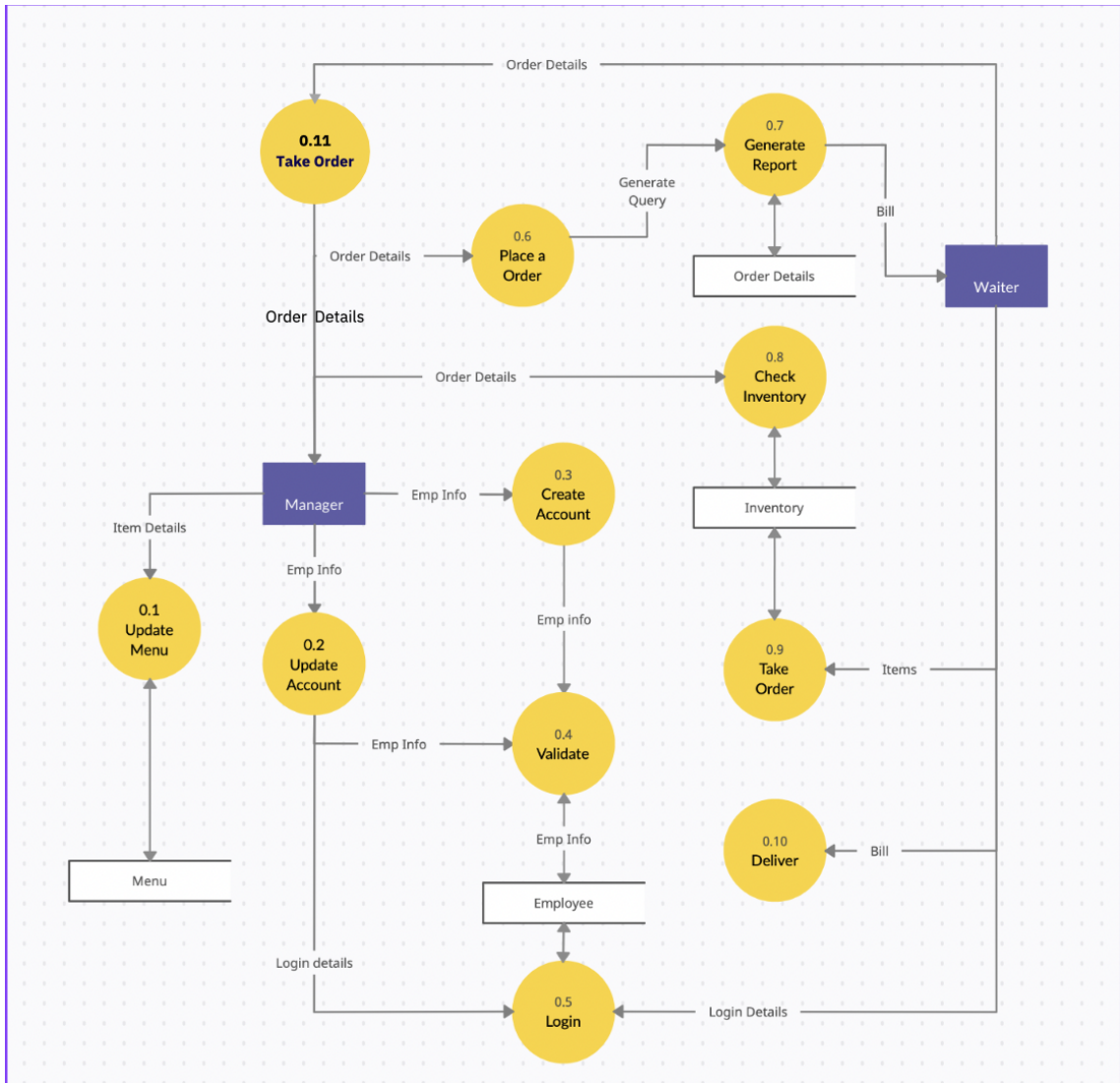
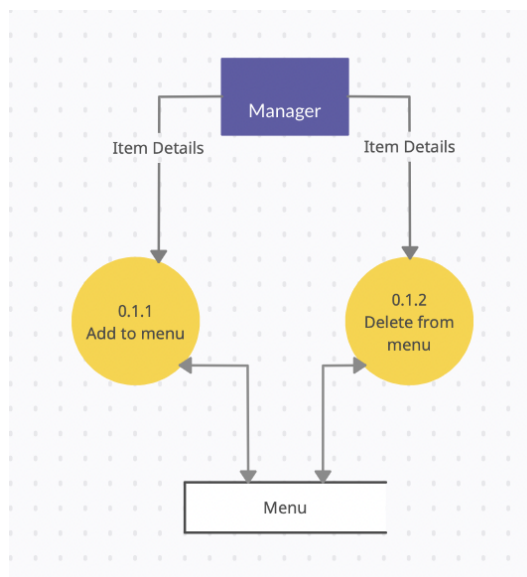


Fig 2: Level 0 Data Flow Diagram



**Fig 3: Level 1 Data Flow Diagram**

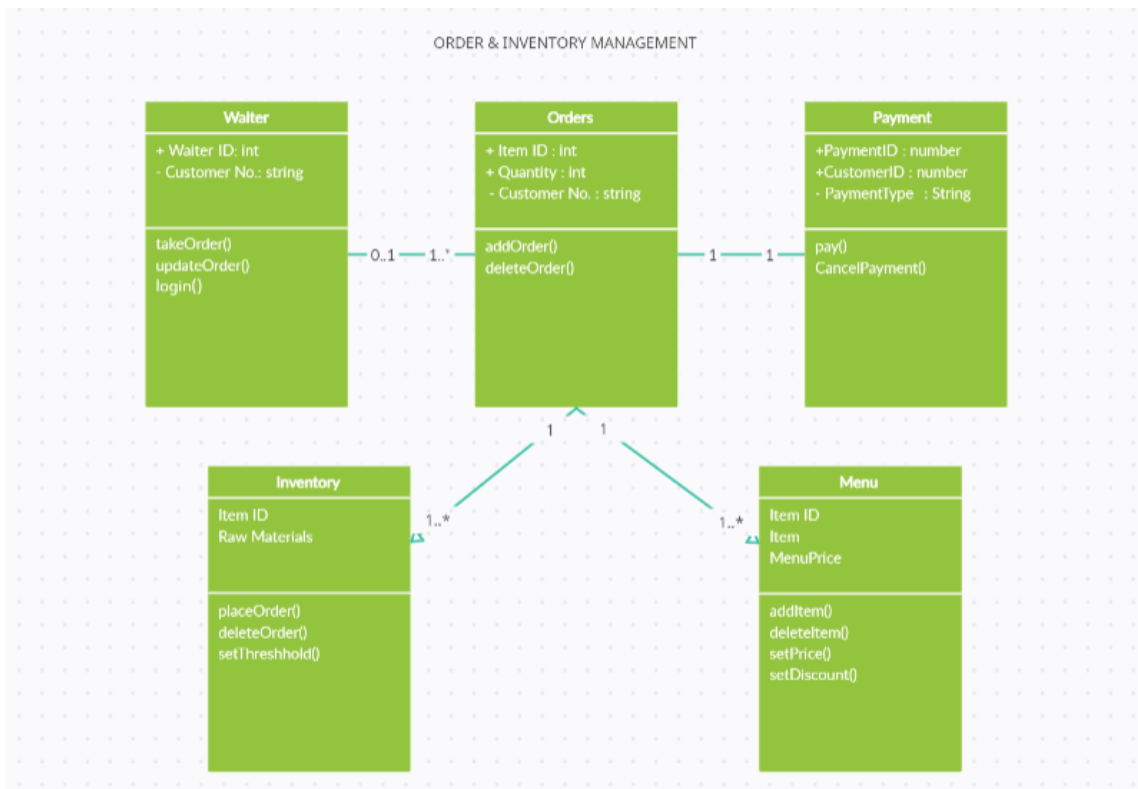


**Fig 4: Level 2 Data Flow Diagram**

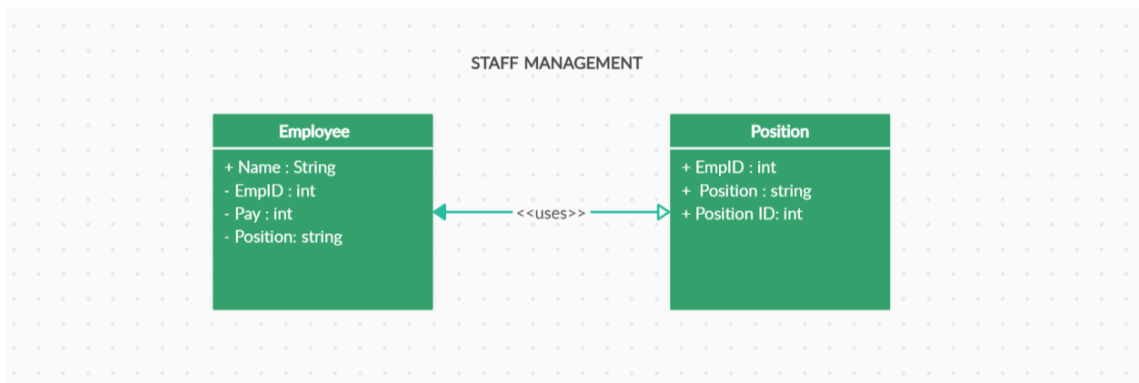
### 3.2.1. Data Dictionary

- Item Details: string
- Employee Info: Employee ID + Employee Password
- Login Details: Login ID + Login Password
- Order Details: string
- Generate Query: string
- Item: string
- Bill: string

### 3.3. Class Diagram

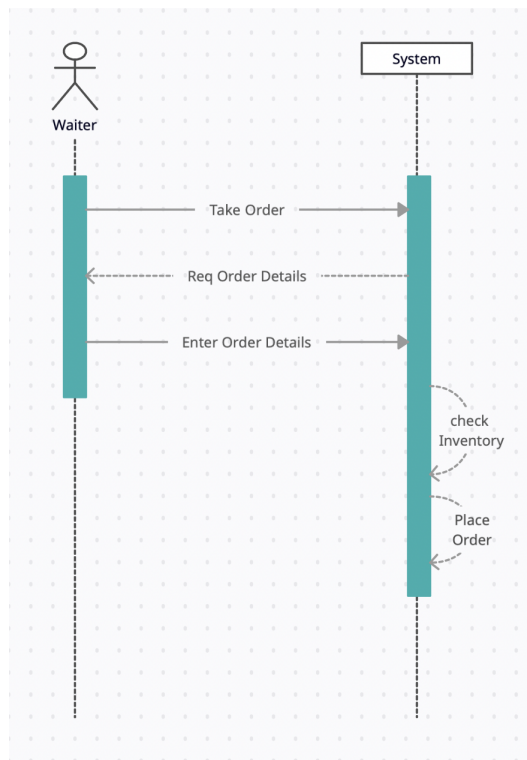


**Fig 5: Class Diagram for Order & Inventory Management**



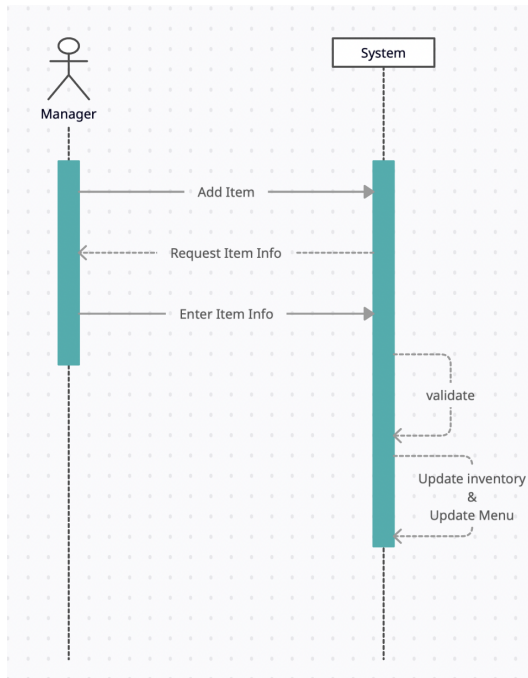
**Fig 6: Class Diagram for Staff Management**

### 3.4. Sequence Diagram

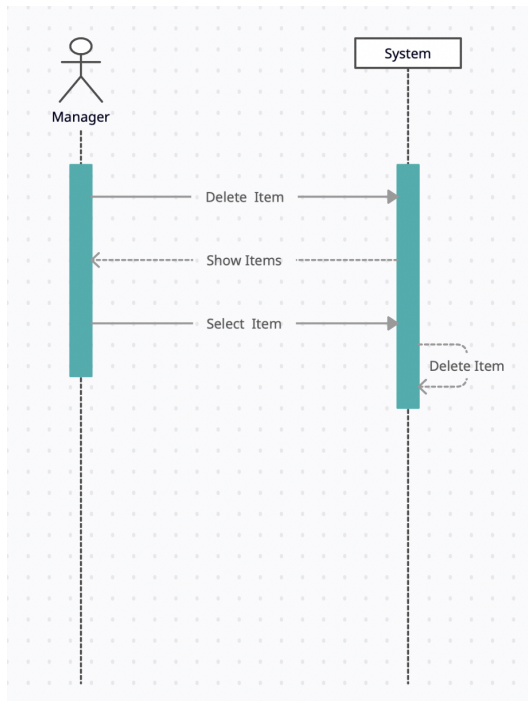


**Fig 7: Sequence Diagram for waiter to TAKE ORDER**

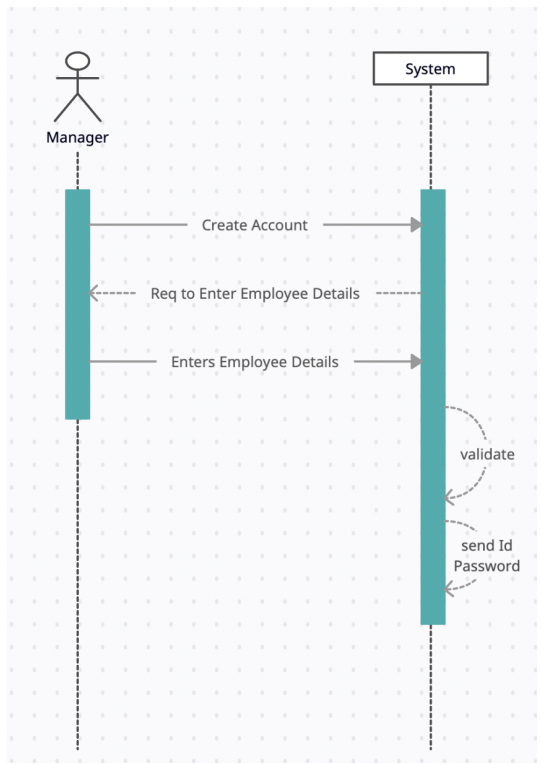




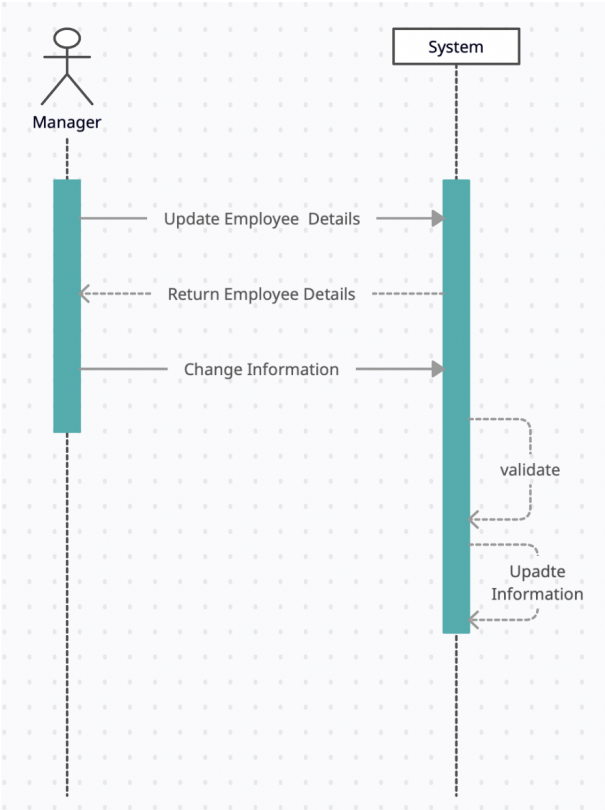
**Fig 8: Sequence Diagram for manager to ADD ITEM TO MENU**



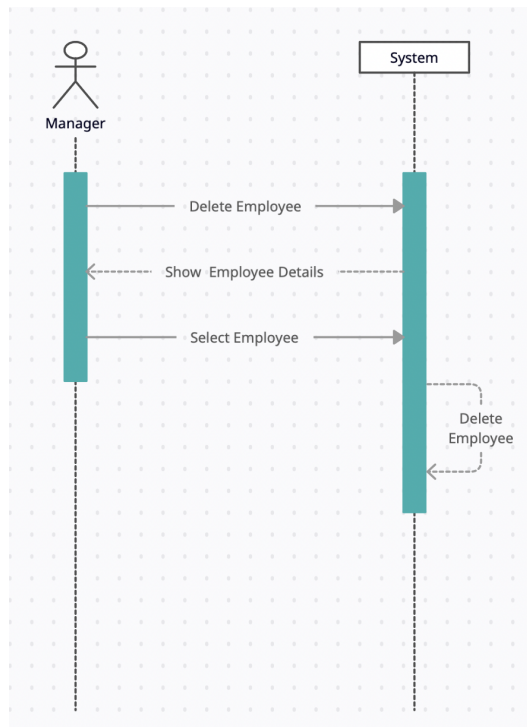
**Fig 9: Sequence Diagram for manager to DELETE ITEM TO MENU**



**Fig 10: Sequence Diagram for manager to CREATE EMPLOYEE ID**

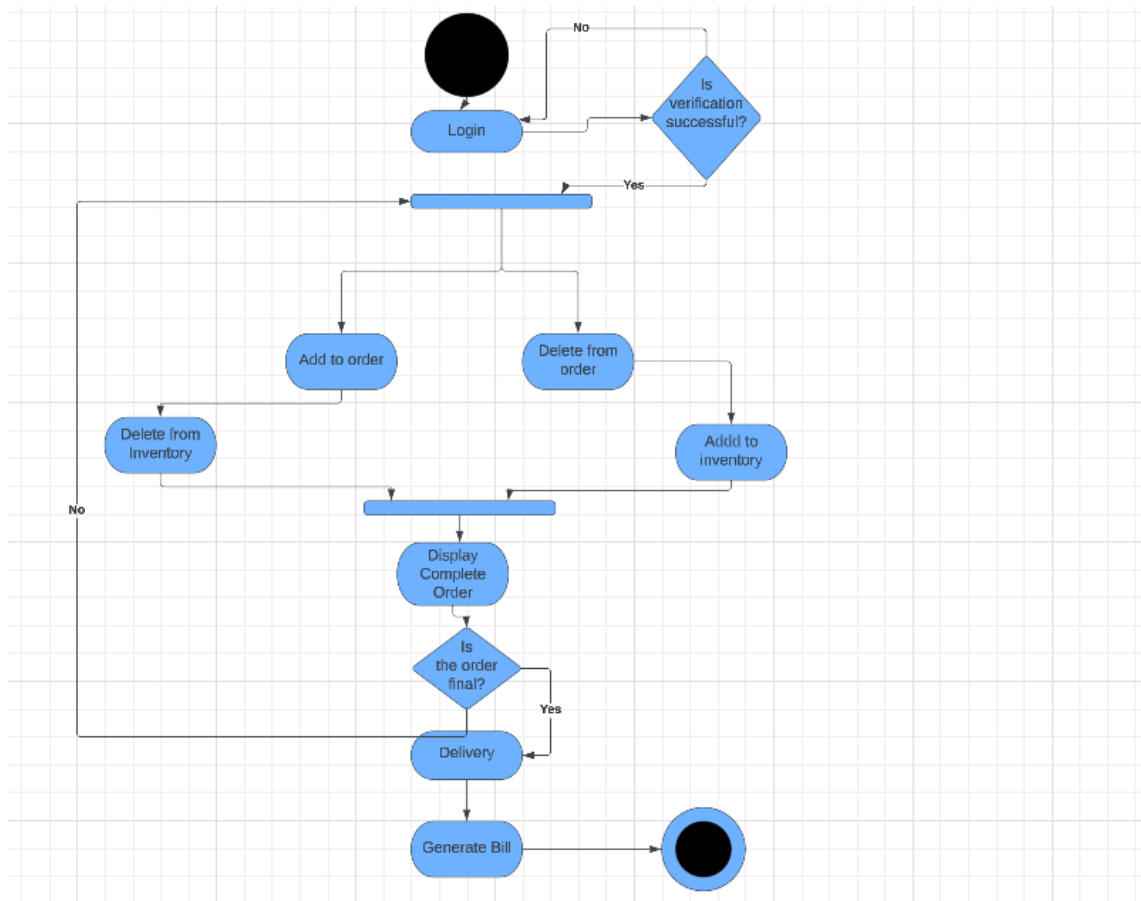


**Fig 11: Sequence Diagram for manager to UPDATE EMPLOYEE ID**

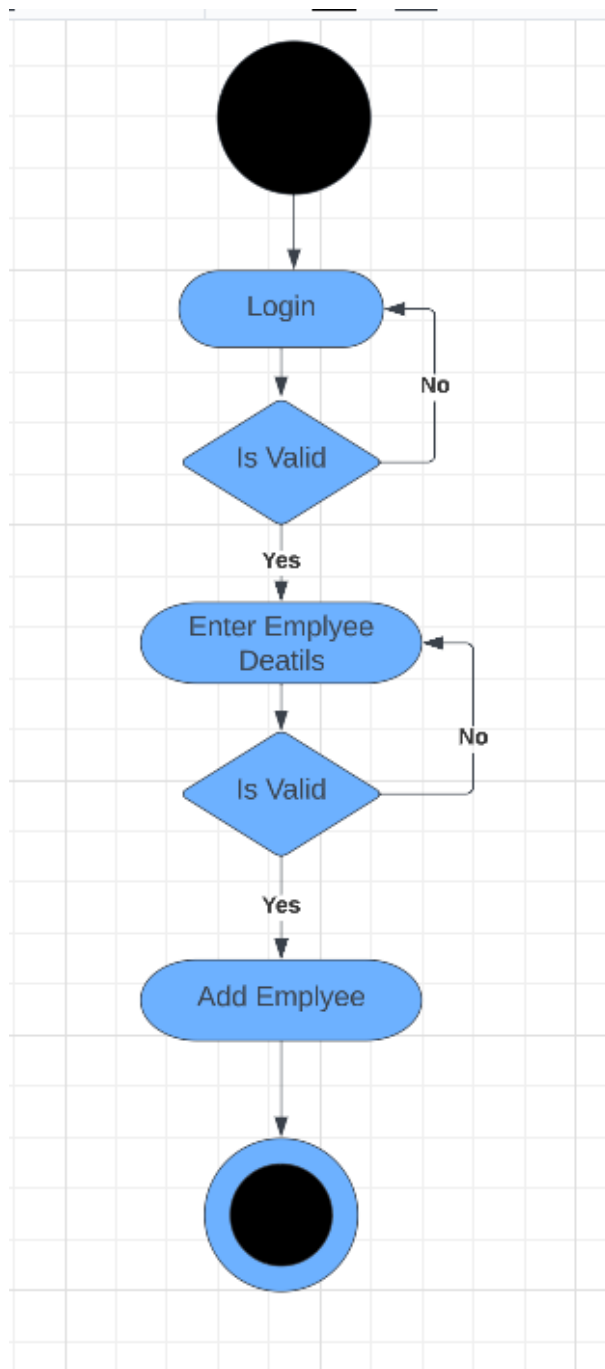


**Fig 12: Sequence Diagram for manager to DELETE EMPLOYEE ID**

### 3.5. Activity Diagram

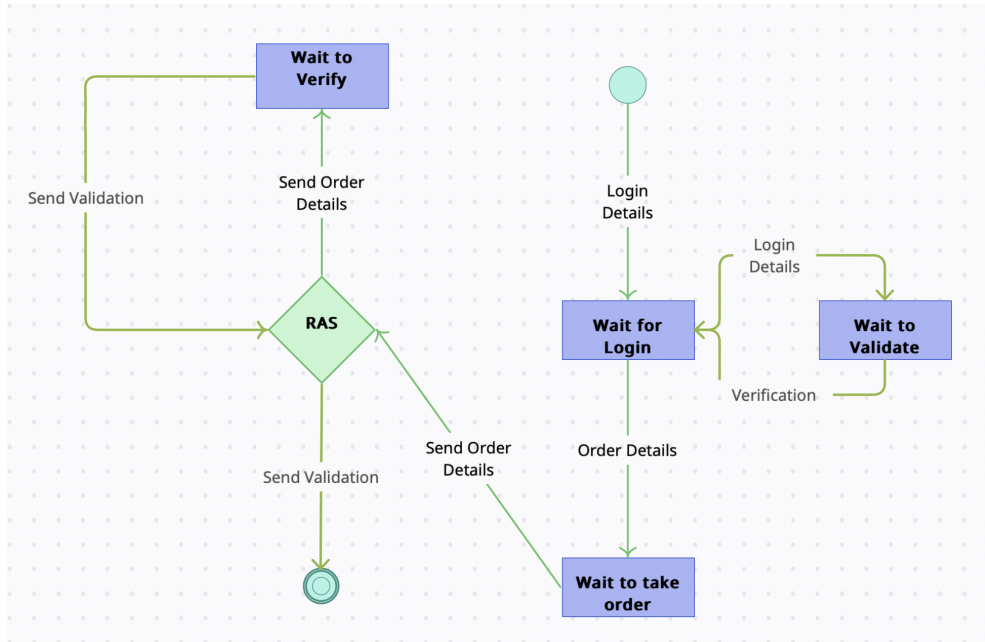


**Fig 13: Activity Diagram for Taking Order**

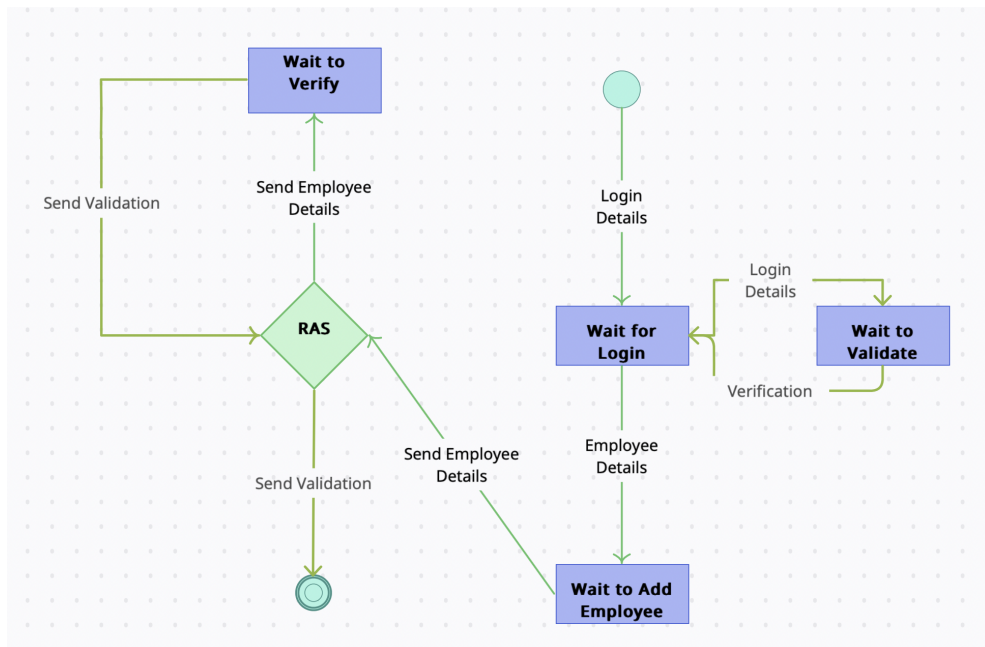


**Fig 14: Activity Diagram for Adding Employee**

### 3.6. State Chart Diagram



**Fig 15: State Chart Diagram to Take Order**



**Fig 16: State Chart Diagram to Add Employee**

## **4. IMPLEMENTATION DETAILS**

A restaurant automation system is a computerized solution that simplifies and streamlines restaurant operations. It's main function is to remove the downtime of a restaurant because of raw material shortage. It is used to manage tasks such as order management, inventory management and staff management. Further in this section we talk about the implementation details to consider when developing Table Tech.

### **4.1. Technology Stack**

The technology stack for Table Tech is chosen based on the requirements of the system. For Front end we used python, the back end part we used google's cloud service Big Query.

### **4.2. System Architecture**

While making a decision about the architecture of our system we need to make sure that the system efficiency doesn't decrease in any way. So, we decided to develop Table Tech with a scalable and maintainable architecture that should be able to handle large amounts of data and traffic. A common architecture for such systems is the three-tier architecture, which separates the presentation layer, application and the data layer.

### **4.3. User Interface**

The user interface of Table Tech will be user-friendly and intuitive. It will allow customers to place orders and make payments quickly and easily, while also allowing restaurant staff to manage orders and inventory efficiently.

The user interface will not require a lot of technical acumen so that we can target a larger set of audience which is comfortable with such a user interface with minimal complexities.

#### **4.4. Integration**

Table Tech will need to integrate with third-party systems such as payment gateways and business analysis software. We need to consider the APIs and SDKs that will be used to enable these integrations.

#### **4.5. Security**

Security is a critical part of Table Tech. The system should be designed to prevent unauthorized access to data and to protect sensitive information such as customer no.

#### **4.6. Testing and Deployment**

Table Tech will undergo rigorous testing in line with our Software Engineering curriculum; these tests will help us ensure that Table Tech meets the functional and performance requirements as mentioned in the problem statement. We can also use continuous integration and deployment practices to streamline the development and deployment process.

Taking all the subsections into consideration, we were successfully able to develop a robust and efficient restaurant automation system: Table Tech that can streamline restaurant operations, reduce the restaurant's downtime and enhance customer experience.



## 5. CONCLUSION

Table Tech provides numerous benefits to restaurant owners, managers and customers. By automating various processes :

- Order Management
- Inventory Management
- Staff Management

A restaurant can reduce costs, increase efficiency and improve the overall dining experience for customers.

In the development of this project report, we have successfully worked as a team and as a team, we actually grasped the concepts of software engineering and the tasks that come hand in hand with the development of a software product. The development stage helped us understand real world implications of the umbrella activities that come under a software process model. All in all, the opportunity has allowed us to understand the importance of making documentation while developing a software product.

Our system hasn't yet been tested and evaluated in a real-world restaurant setting but we are very sure that the results will show that it can significantly improve the efficiency of restaurant operations, reduce errors, and increase customer satisfaction Overall, our restaurant automation system is a valuable tool for restaurant owners who are looking to streamline their operations and hop onto the train of future, Table Tech provide a better experience for their customers

## **A. APPENDIX**

### **A.1. References**

In the development of this project report we have extensively referenced slides of our Software Engineering curriculum by Prof. XYZ.

### **A.2. Project Timeline**

Table Tech project timeline: 07 March 2023 to 03 April 2023