IT 314 - Software Engineering

Lab 2 - Group 14

Team Members	Student ID
Pulkit Khandelwal	202001120
Shreyansh Khemesara	202001121
Prajapati Vrutik Navneetbhai (Team Leader)	202001124
Baraiya Dhruv Pravinbhai	202001140
Mandaviya Soham Hiteshbhai	202001142
Shah Nishadhkumar Nirajbhai	202001151
Buddhdev Harsh Nitesh	202001157
Chhagani Krunal Ajaybhai	202001158
Thakor Harshal Dipaksinh	202001169

Need for the Project

- **Improving efficiency:** If we automate the tasks such as taking orders , tracking orders and inventory through a software system ,which would help in saving the time and resources of customers as well as restaurant owners.
- **Increased Accuracy:** Automating the tasks will help in reducing the human errors such as incorrect orders, incorrect billing, and mismanagement of inventory so accuracy will be increased and customer experience will be improved.
- **Better Data Management:** Collecting and storing data can be done more efficiently with the help of software and it will provide real time insights into sales and inventory to the restaurant owners and managers.
- **Improved Customer Experience:** With automated systems customers will have a faster and more convenient and personalized experience.
- **Cost Reduction:** Automating tasks can reduce labor costs and it will increase the profit ,reduce waste and increase efficiency.

Features for the project:

Roles :

1. Admin
2. User

Features for Admin:

- **Order management:** A web system for taking orders from customers, processing payments and managing the workflow of the kitchen.
- **Table management:** A web system for managing table assignments , tracking customer wait times and reporting seat capacity.
- **Inventory management:** A web system, which helps the owners and managers to keep track of inventories and/or required stocks.
- **Active order tracking:** A web system which helps users as well as owners to keep track of current orders' status.
- **Serving management:** A web system which keeps track of current serving status.
- **Tracking of past orders**: A web system which stores data about previous orders of a particular user.
- **Food Categories:** A category function in a system which shows food by type of cuisine they fit in.
- **Billing management** : A feature which helps users to get online bills and hence reducing the manual cashier work and eradicating the loop holes in the billing process.

Features for Users:

- **User access management:** A user can create an account on the web system which can further be used to place and track orders.
- **Online Ordering:** A web system which helps customers to order from a place of convenience, and cancel their orders if they want to.
- **Feedback Portal** : Customers can easily give their feedback on the website after having their food.

Functional and non-Functional requirements

Functional Requirements :

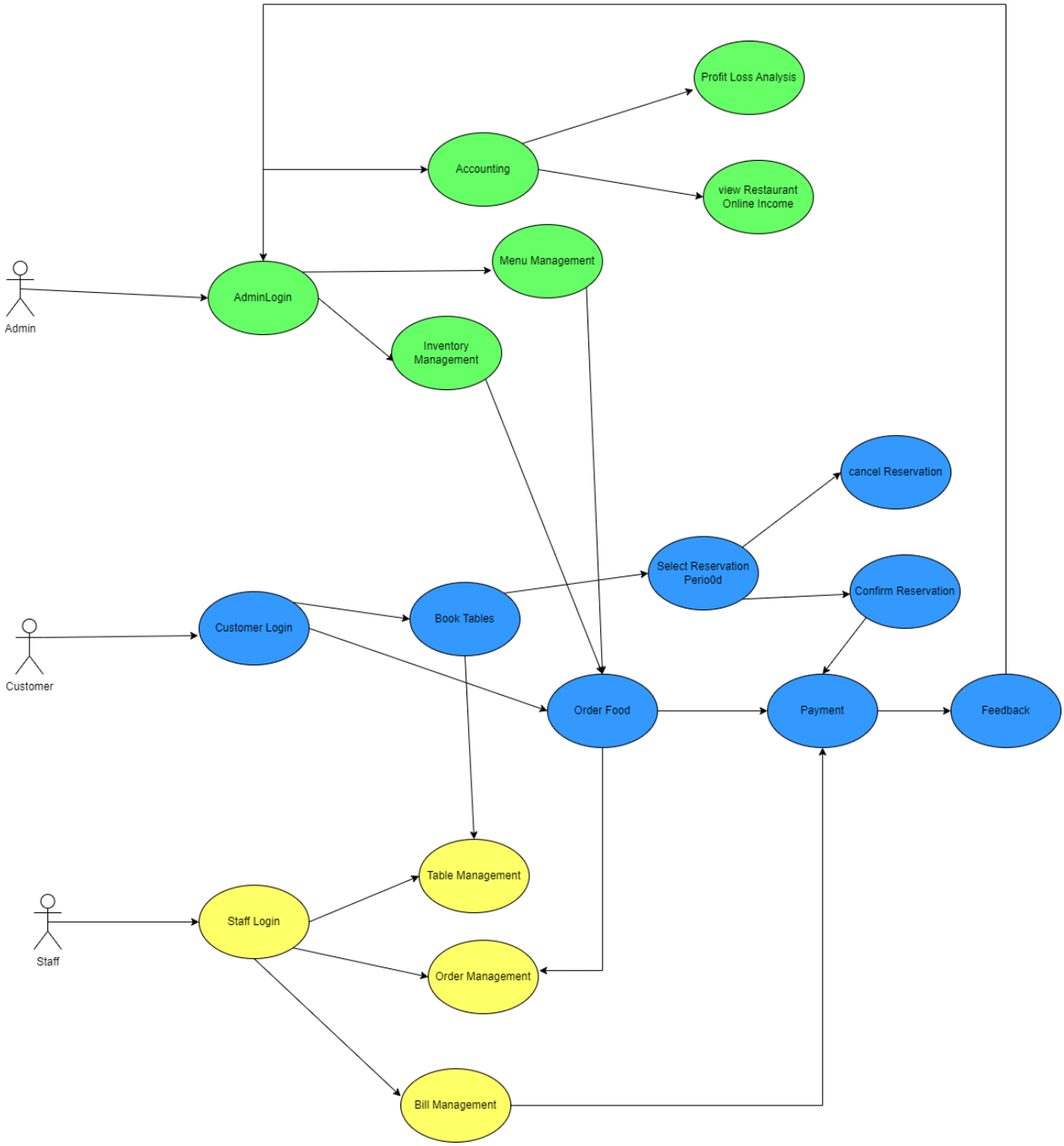
- **Order Taking:** The system should allow restaurant managers to take orders from the customers.
- **Menu Managemen**t: The system should allow restaurants to manage and update their menu, including the ability to add, edit, and remove menu items.
- **Inventory Management:** The system should allow restaurants to manage their inventory, including tracking the quantities of ingredients and supplies on hand and generating alerts when supplies are running low.
- **Kitchen Operations:** The system should allow restaurants to manage kitchen operations, including the ability to view and manage orders, track the status of dishes, and communicate with kitchen staff.
- **Payment Processing**: The system should allow customers to make payments through various methods, such as credit card, cash, or mobile payment, and provide the restaurant with real-time payment tracking and reporting.
- **Reporting and Analytics:** The system should provide restaurants with reporting and analytics capabilities, including sales reports, and performance metrics.
- **Online Accessibility:** The system should be accessible online allowing customers to place orders, make payments, and receive updates and notifications.

- **Customer Feedback :** The system should allow customers to give their feedback about their experience at the restaurant.

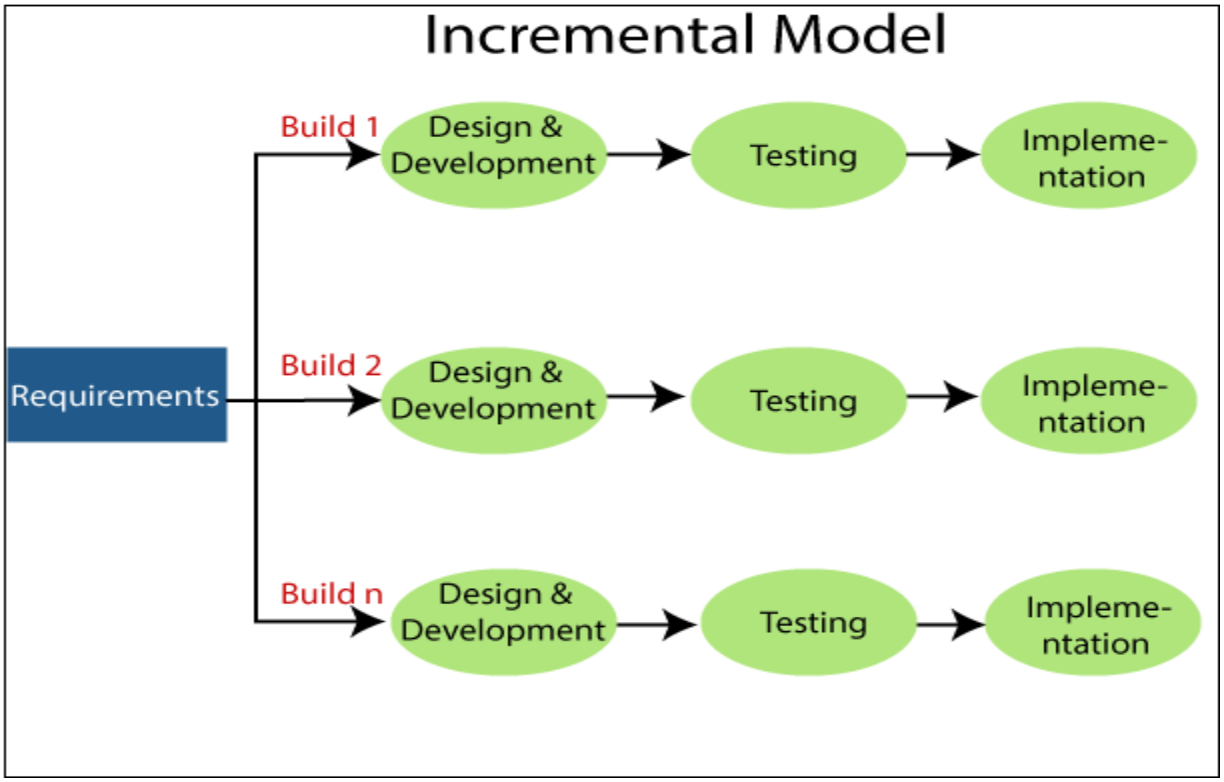
Non-functional requirements:

- **Security:** The system should be secure and protect sensitive information from unauthorized access and data breaches.
- **Reliability:** The system should be reliable and consistently available, with minimal downtime and technical issues.
- **User-Friendliness** : The system should be easy to use and navigate, with a user-friendly interface that requires minimal training.
- **Responsiveness** : The system should be responsive ,irrespective of the platform on which it is running.
- **Scalability :** The system should be scalable and flexible, able to accommodate the growing needs of the restaurant over time.
- **Performance:** The system should have fast and efficient performance, with quick load times and minimal lag.
- **Support:** The system should be backed by a responsive and knowledgeable support team, with access to technical assistance and updates as needed.

Use Case diagram



Process model: Incremental



There are following reasons for choosing incremental model :

- **Cost and Budget Control**: Developing a restaurant automation system as a whole can be a costly and time-consuming process. By using an incremental approach, the project can be broken down into smaller, more manageable pieces that can be developed and implemented in phases. This allows the restaurant to spread the cost over time and prioritize the most critical components of the system.
- **Flexibility**: An incremental process model allows for changes and adjustments to be made along the way, as the system is being

developed and used. This can be especially useful in a dynamic industry like the restaurant business, where menus, prices, and processes can change frequently.

- **Early Feedback**: By implementing the system in stages, restaurant staff and customers can provide feedback on each increment, allowing the developers to make adjustments and improve the system as it is being developed.
- **Reduced Risk**: Developing a complex system all at once can be risky, as there is a greater chance for something to go wrong. With an incremental approach, the risks are spread out over time, and any problems can be addressed as they arise.

To summarize, the incremental process model can be an effective way to develop a restaurant automation system because it allows for cost control, flexibility, early feedback, and reduced risk.