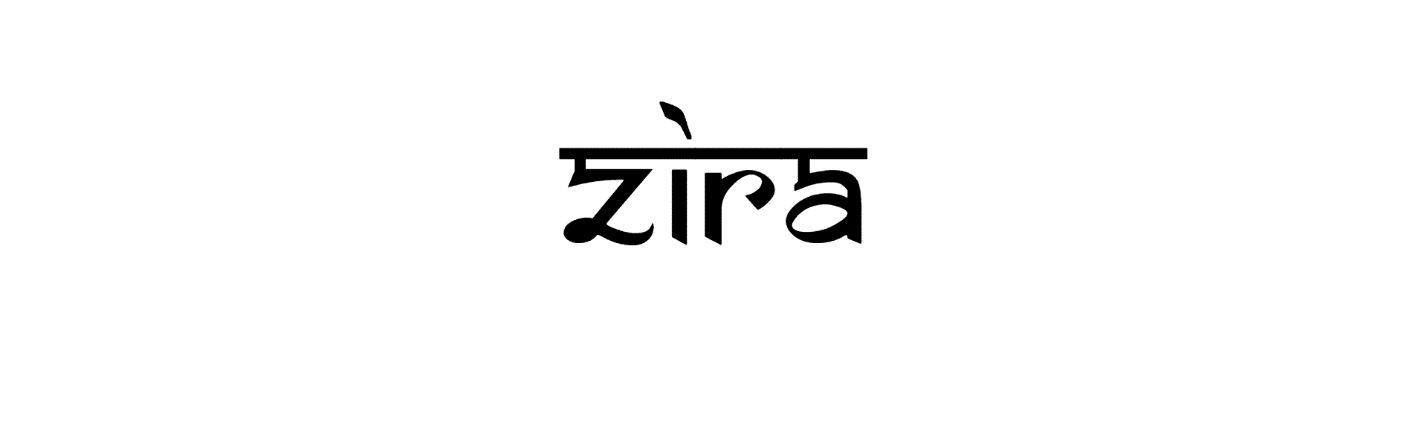
**SOFTWARE REQUIREMENTS SPECIFICATIONS**

**for**



Group 1

**Prepared by:**

Harshbardhan Singh

Abhishek Kumar

Vikalp Yadav

Manish Pramanik

Alyan Jamal Qureshi

Vishal Sharma

Anish Kumar Mishra

**Content:**

1.Objective.

2.Introduction.

3.Scope.

4.Significations.

5.Modules.

* Login Module.
* Menu Module.
* Specific Module.

6.User Interface Design.

7.Technologies Used.

* Front End.
* HTML.
* CSS.
* JavaScript.
* Back End.
* Java.
* Spring Boot.
* Spring Data JPA.
* Spring Security.
* Database.

8.Hard Ware Requirements.

9.Project Life Cycle.

10.Snapshots.

11.Dataflow Diagram.

12. Use Case Diagram.

**Abstract:**

Restaurant management system is a web application for Managing the restaurant and orders. It is built to automate the activity of restaurants. This application's hardware and software are easily available for any institution required.

It is used by Admin and staff both as well for managing Stores, product and for creating order and for paying bills. There is no need to maintain record manually.

The Restaurant Management is based on managing the Food and sell the Food online. This application is simple and impressive. The main objective behind this web application is to provide all the details about Food, Food type. This application is very helpful for maintaining the sales activity in Food because it is very complex task to manage the daily sales manually. In this project Admin can see the details of Food, Food type etc. Only Admin can edit or delete the details of Food.

**Objective:**

* To develop an Online Restaurant Management system. To help the Owners of the restaurant.
* To efficiently manage everything like Staff, Inventory etc.
* To track the Orders and manage it.

**Functionalities provided by Restaurant Management System are as follows:**

* Admin can ADD/VIEW/DELETE/UPDATE Staff Information.
* Admin can ADD/VIEW/DELETE/UPDATE Categories Information.
* Admin can ADD/VIEW/DELETE/UPDATE Product Information.
* Admin can ADD/VIEW/DELETE/UPDATE Table Information.
* Admin can ADD/UPDATE User Information on the request of User.
* Admin can VIEW User Information.
* Admin can VIEW Total products, Total User, Total Stores.
* Admin can ADD/VIEW/DELETE/UPDATE Suppliers Information.
* Admin can ADD/VIEW/DELETE/UPDATE Inventory.
* Admin can ADD/VIEW/DELETE/UPDATE Items in Menu.
* Admin can VIEW the Bills.
* Admin can VIEW/UPDATE its own information.
* Our application is Safe and Secure as the user details are encrypted in the Database.

**Introduction:**

The project titled Restaurant Management System is Restaurant Management Software for Monitoring and Controlling the Transactions in a Restaurant. The project “Restaurant Management System” is developed by using Java, Spring Boot which mainly focuses on basic operations in a Restaurant like adding new Users, new Items in Menu and Updating deleting the information, \* these transactions data stored in database.

It is designed to help users to maintain and organize everything in Restaurant. Our Software is easy to use for everyone. It has an attractive user interface and it has strong insertion and reporting Capabilities.

**Scope:**

Our project aims to safe and easy handling of restaurant. i.e. we have made a computerized process to store data and distribution.

1. It satisfies the admin(Owner).
2. It is easy and safe to store data.
3. It is easy to operator.
4. Have a good user interface.
5. It saves time and function faster.
6. It helps the owner to manage the restaurant.

We have tried to develop safe and secure software with above mentioned specifications.

**Significations:**

* Easy to update information.
* Work becomes speedy.
* Access of any information individually.
* Decrease the load of the person involve in existing manual system.
* Well-designed reports.
* Easy & fast retrieval of information.
* Accuracy in work.
* It contains better storage capacity.
* Creating and changing data at ease.

**Modules:**

Our system has one user module and one admin module.

Along this we have some more modules mentioned below.

**Login Module:**

Only admin can log in to the system as the end-user of the system on the behalf of the user. The user will get only those privileges that are given to the user for which one has registered.

**User Module:**

**Specific Module:**

In this module admin can check and update all the information like all transactions, availability of menu, menu id’s, search of menu, delete menu.

**User Interface Design:**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

• The system user should always be aware of what to do next.

• The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.

• Use display attributes sparingly.

• Default values for fields and answers to be entered by the user should be specified.

• A user should not be allowed to proceed without correcting an error.

• The system user should never get an operating system message or fatal error.

**Technologies Used:**

**Front End:**

**HTML:**

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

**CSS:**

Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

**JavaScript:**

JavaScript (JS) is a scripting language, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JavaScript is an interpreted language. Thus, it doesn't need to be compiled. JavaScript renders web pages in an interactive and dynamic fashion. This allowing the pages to react to events, exhibit special effects, accept variable text, validate data, create cookies, detect a user’s browser, etc.

[**Angular**](https://angular.io/)**:**

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

**Back End:**

[**Spring**](https://spring.io/) **Boot:**

Spring Boot is an open-source micro framework maintained by a company called Pivotal. It provides Java developers with a platform to get started with an auto configurable production-grade Spring application. With it, developers can get started quickly without losing time on preparing and configuring their Spring application.

[**Spring**](https://spring.io/) **Data JPA:**

JPA is a Java specification that is used to access, manage, and persist data between Java object and relational database. It is a standard approach for ORM.

[**Spring**](https://spring.io/) **Security:**

Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications. Spring Security is a framework that focuses on providing both authentication and authorization to Java applications.

**Database:**

[**Oracle**](https://www.oracle.com/database/technologies/) **SQL:**

SQL (pronounced sequel) is the **set-based, high-level declarative computer language** with which all programs and users access data in an Oracle database. Although some Oracle tools and applications mask SQL use, all database tasks are performed using SQL.

**Hard Ware Requirements:**

|  |  |  |
| --- | --- | --- |
|  | **Minimum System Requirement.** | **Recommended System Requirement.** |
| Processor | 7th Gen Intel Core i7 | 9th Gen Intel Core i7 or better |
| RAM | 4GB | 8GB or more |
| Storage | 256GB SSD | 512GB or more |
| Display | 14-inch FHD (1920 X 1080) | 15.6-inch FHD IPS(1920 X 1080) |
| Graphics | 4GB NVIDIA GeForce GTX 1060 | 8GB NVIDIA GeForce GTX 2070 |
| Battery | Up to 2 hours | Up to 5 hours |

**Project Life Cycle:**

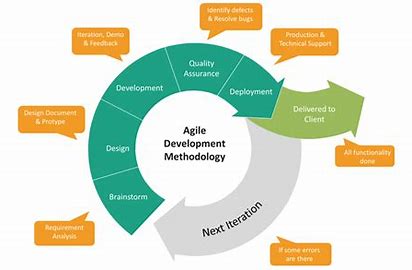
**Agile Software Development Methodology:**

The Agile software development methodology is one among the only and effective processes to show a vision for a business need into software solutions. Agile may be a term want to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to vary.

The agile software development emphasizes on four core values.

* Individual and team interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

**Phases of Agile Methodology:**



**Agile Methodology Phases**

**Phase-1:** Requirement Analysis: - In this phase, we gather data and analyses how restaurant management system works. Also collected requirements after reviewing earlier papers & websites.

**Phase-2:** Design: - On the basis of gathered information we designed and build a model.

**Phase-3:** Development: - Deliver the working software based on iteration, requirements or feedback.

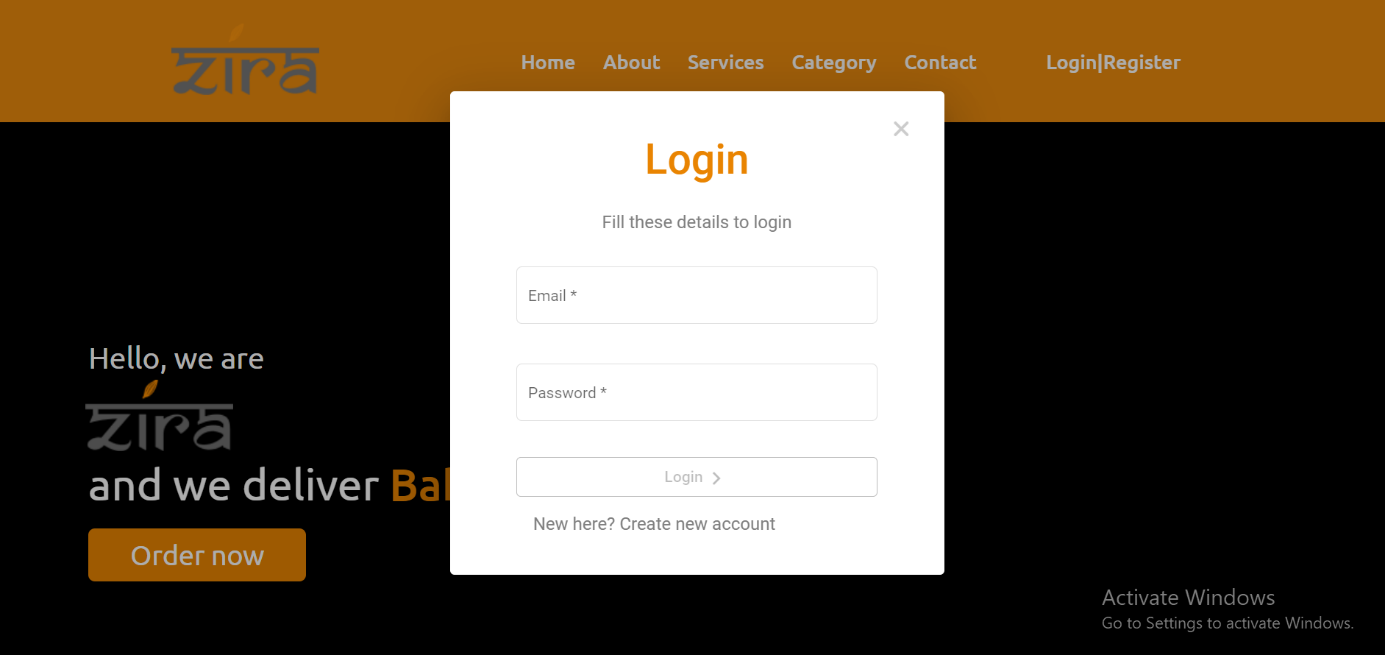
**Phase-4:** Quality Assurance: - This is a testing phase where we test our model.

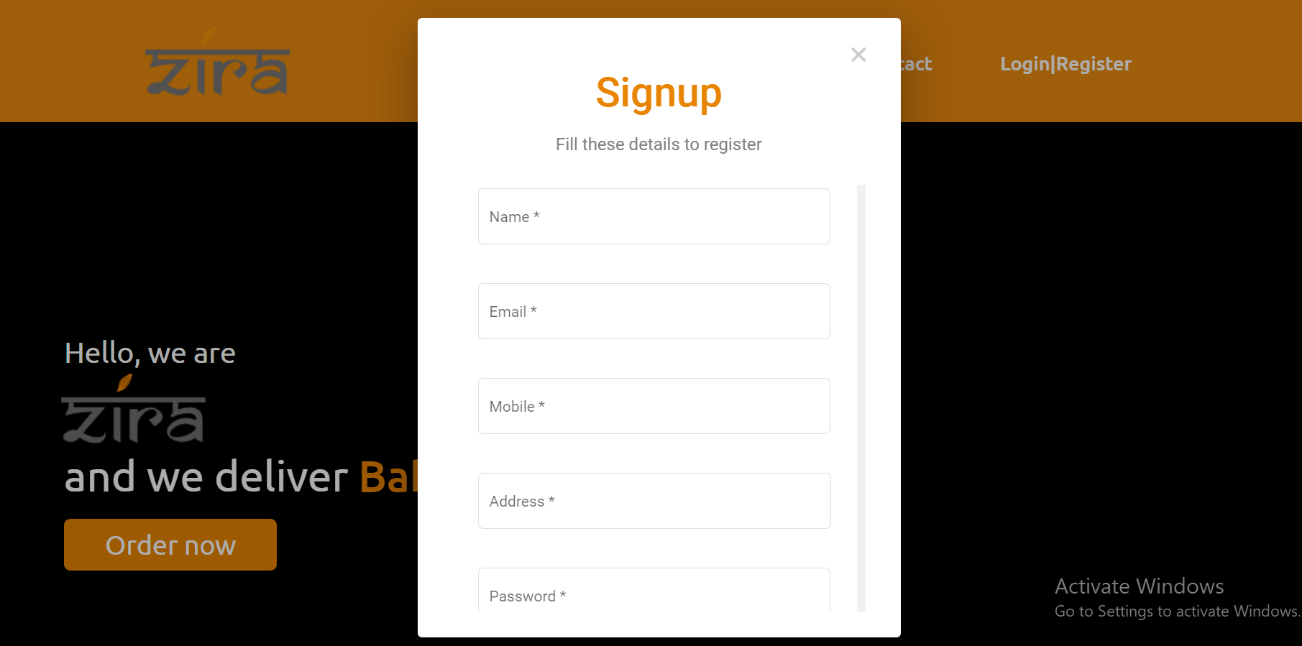
**Phase-5:** Deployment: - In this phase we deploy our final release of the iteration into production.

**Phase-6:** Feedback: - Accept the user feedback and work it into the requirements.

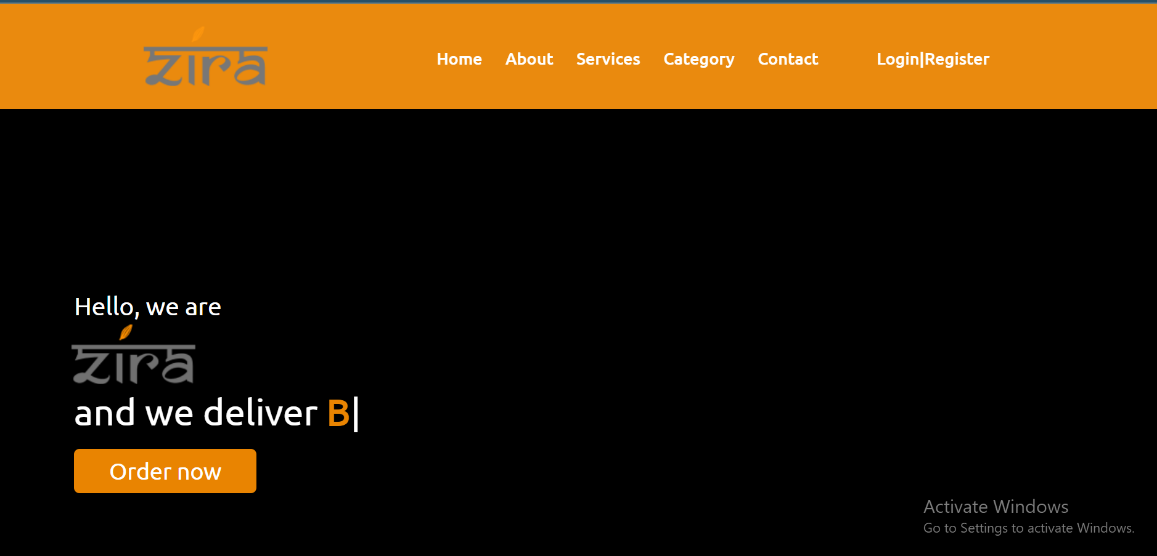
**SNAPSHOTS:**

**Login page:**

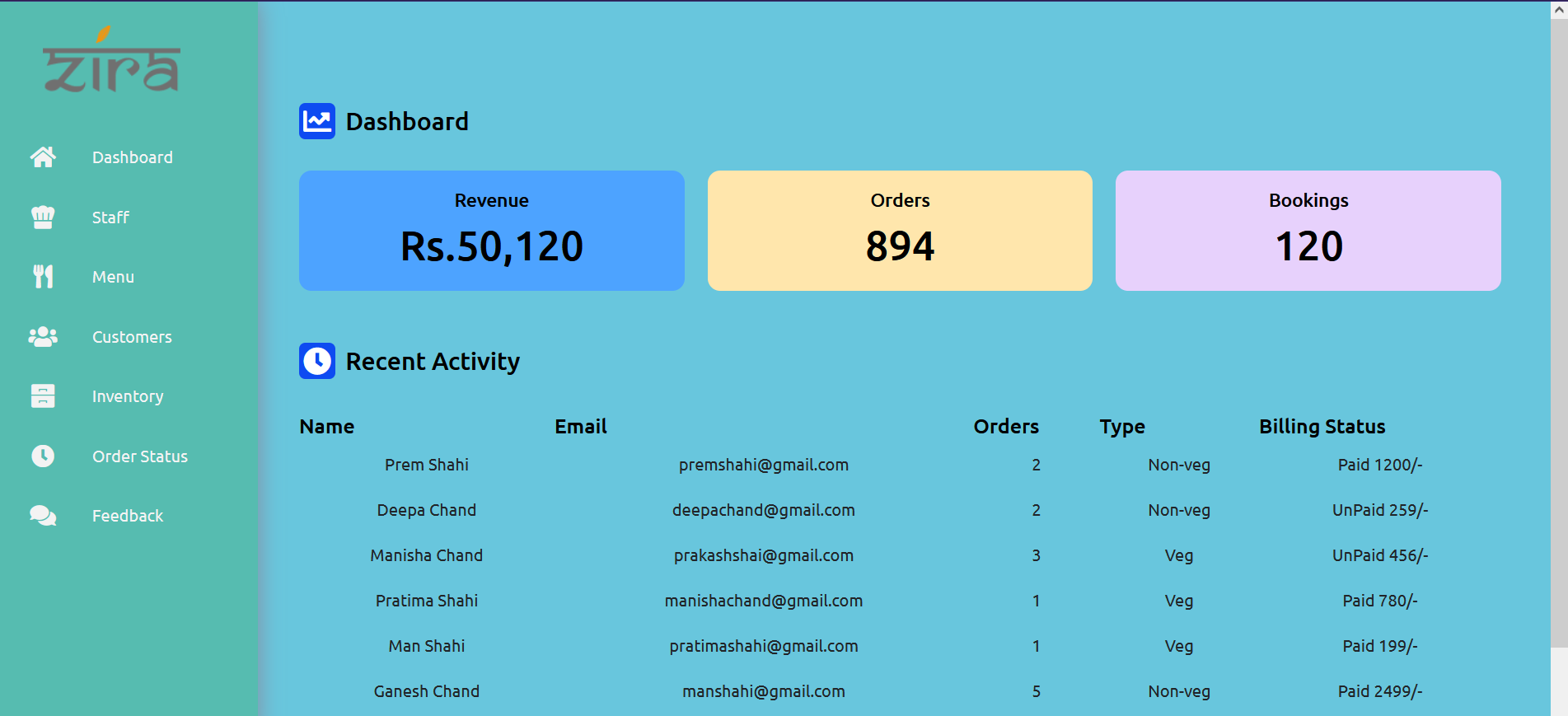
****

****

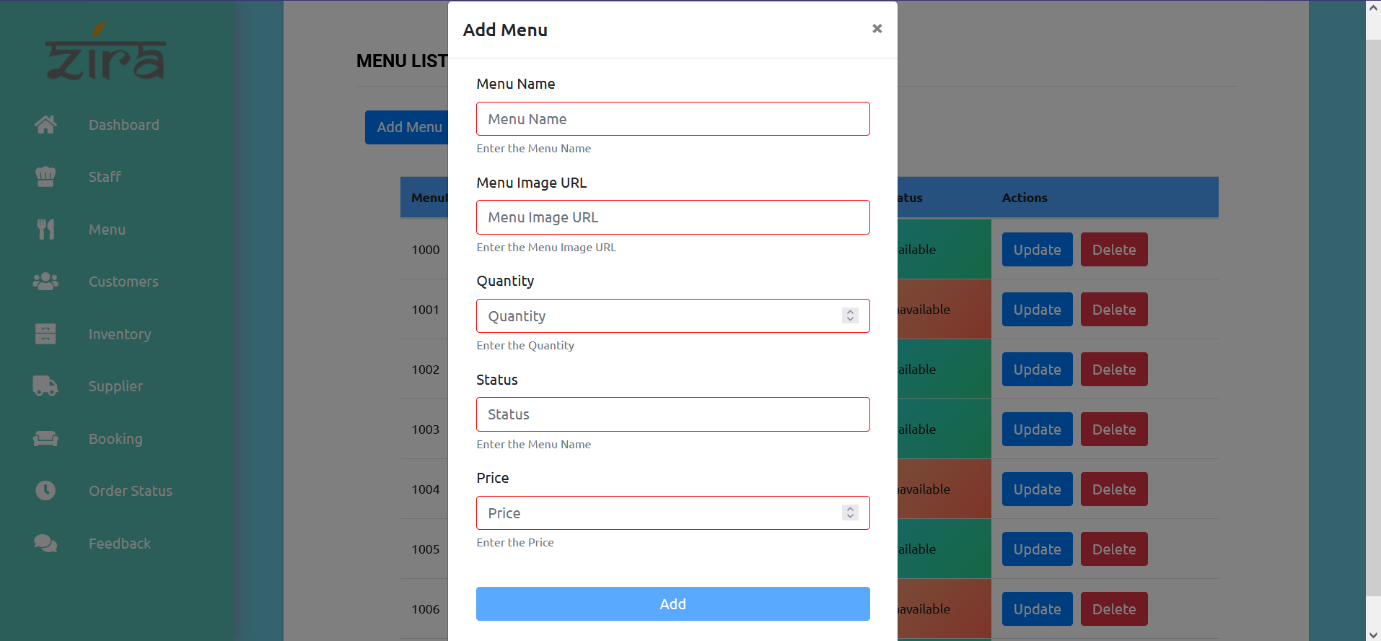
**Home Page:**

****

**Dashboard:**

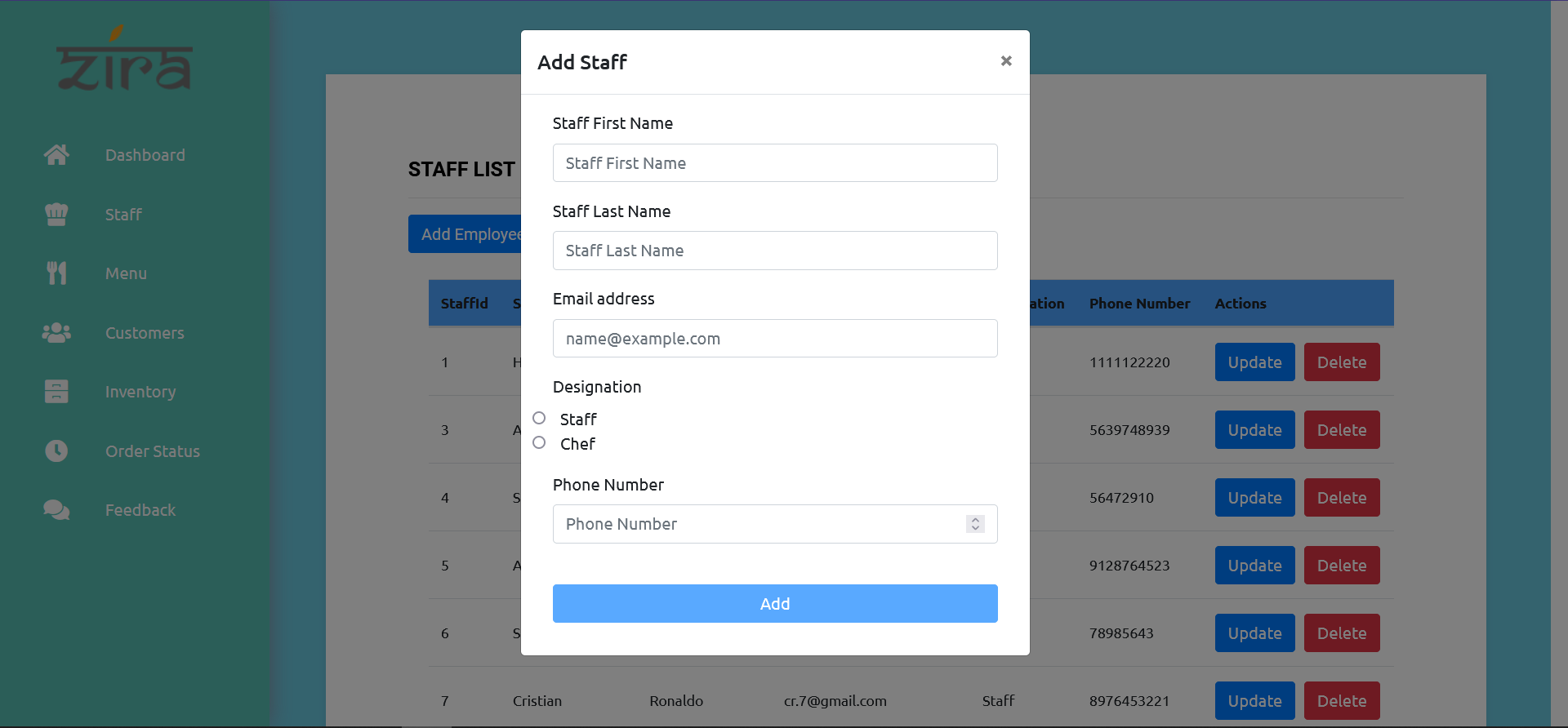
****

**Add Menu:**

****

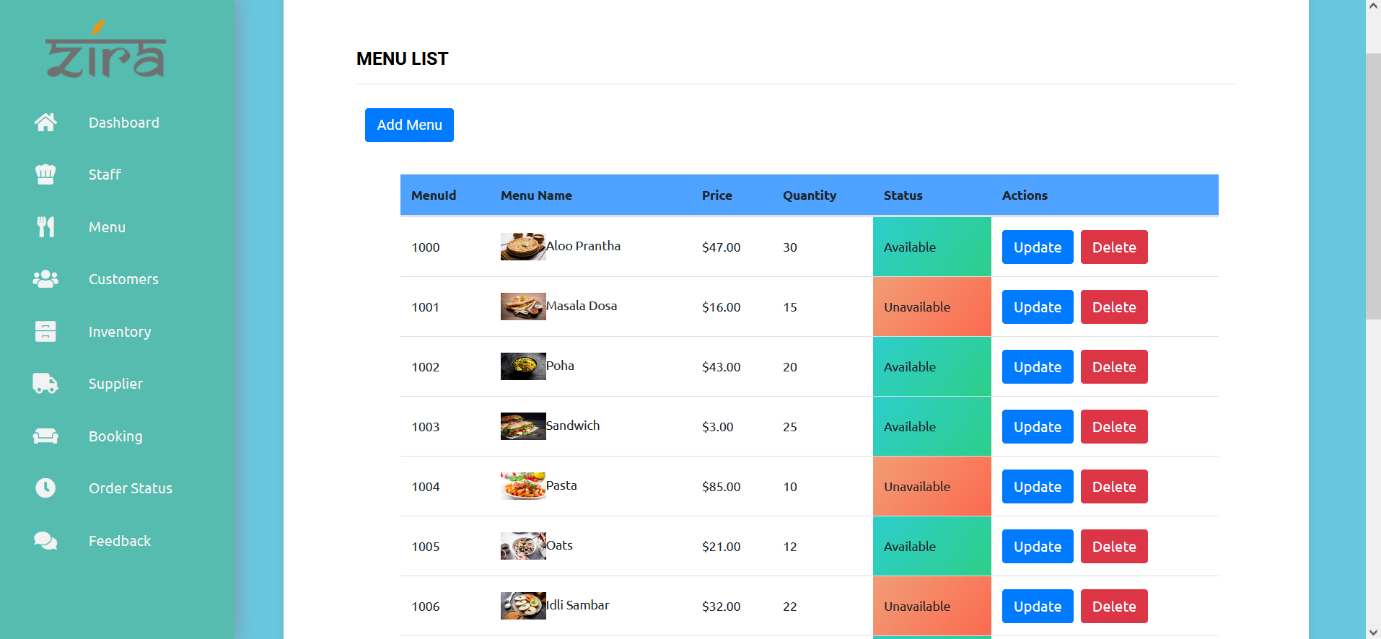
|  |  |
| --- | --- |
| Brief Description | **Add new menu.** |
| Basic Flow | This use case to describe how an admin can add new menu to the system.   1. The admin should add menu to the system. 2. After the successful adding, shows a popup. 3. The following information is required during adding the menu.  * Menu name. * Menu image url. * Quantity. * Status. * Price. |
| Alternate Flow | 1. The system will validate the information provided. If any invalid data is found, the input form will be redirected with error message. |
| Validation | 1. Menu name, image url, status and Minimum 3 characters and max 30 characters. |
| Pre-Conditions | Admin should have network access and Browser with latest updates. |
| Post-Conditions | Success popup should be shown. |

**Adding Form:**

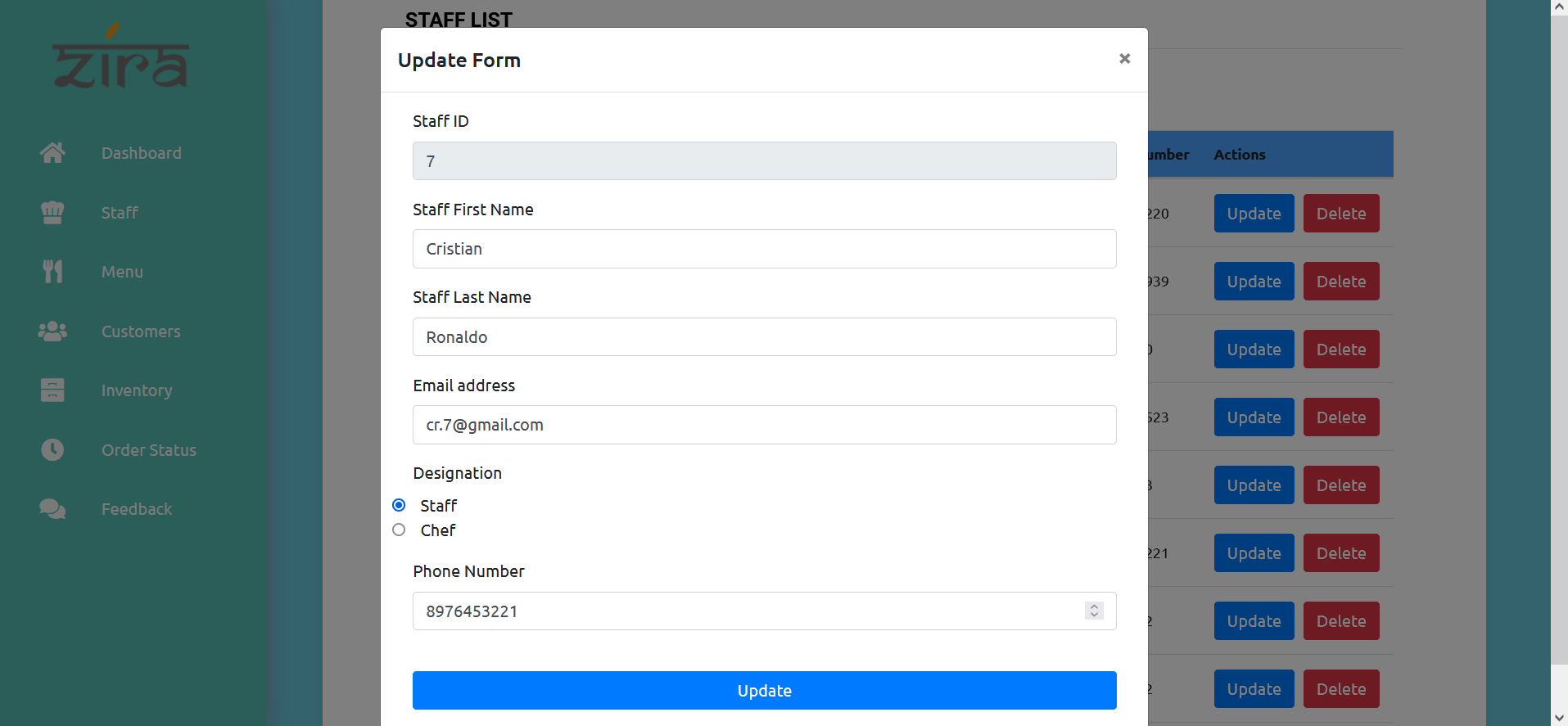
****

|  |  |
| --- | --- |
| Brief Description | **Add Staff.** |
| Basic Flow | This use case describes how an admin can add new member to system.   1. Admin have to login. 2. Select add staff button. 3. Enter details and add new student. 4. The following information is required during adding the member.  * Name. * Email. * Designation. * Phone Number. |
| Alternate Flow | 1. The system will validate the information provided. If any invalid data is found, the input form will be redirected with error message. |
| Validation | 1. Name, phone number, email, and designation are required and minimum 3 characters and max 30 characters. 2. Provided details should valid. |
| Pre-Conditions | Admin should have network access and Browser with latest updates. |
| Post-Conditions | Success popup should be shown. |

**Menu List:**

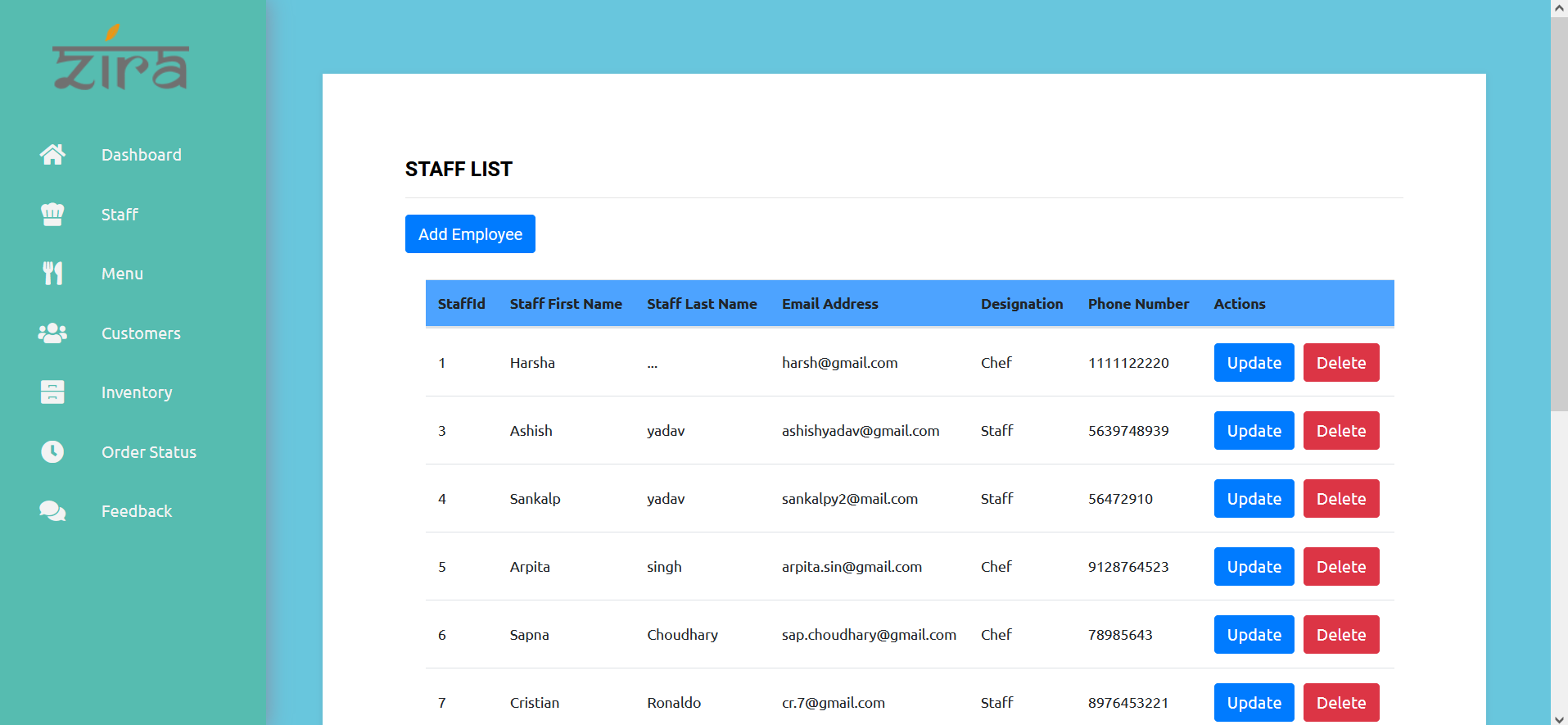
****

**Update Form:**

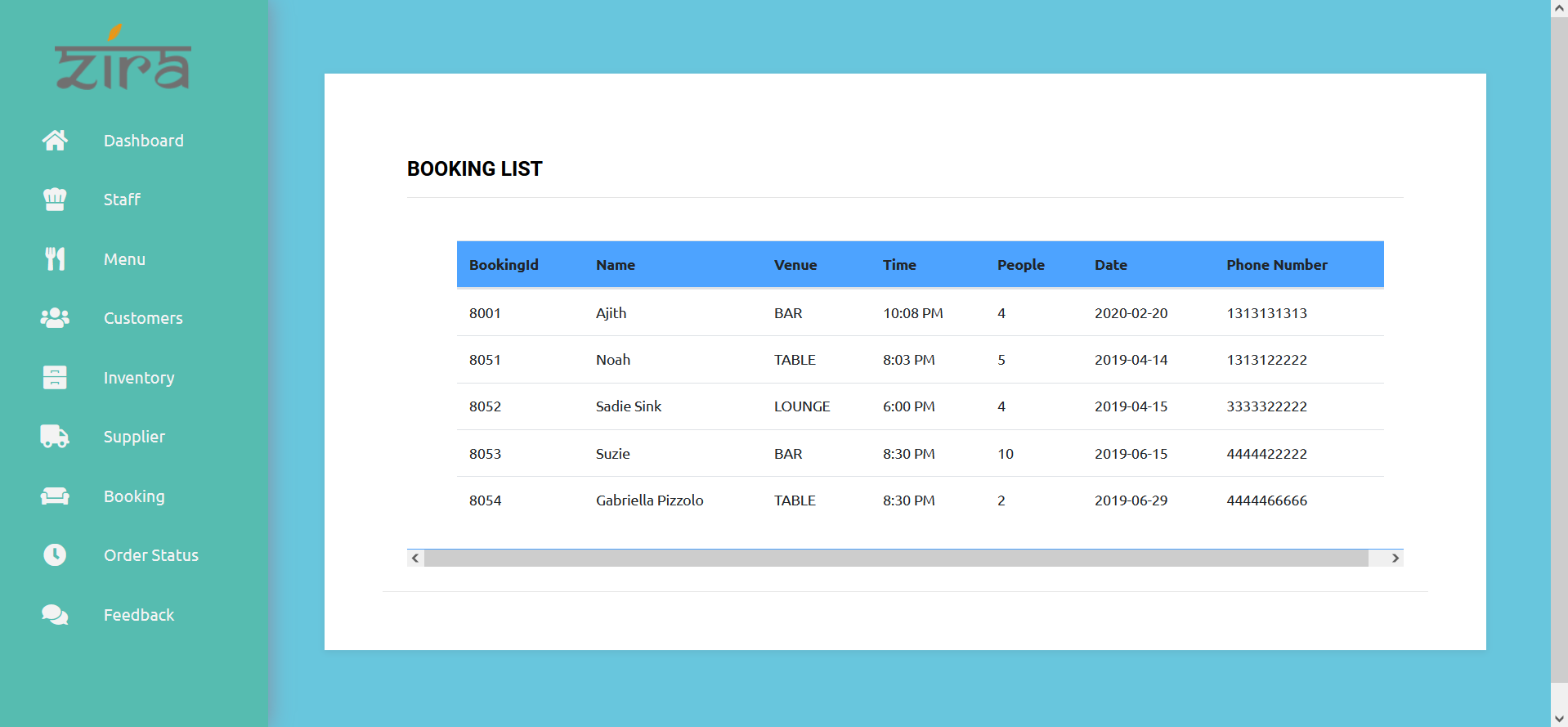
****

|  |  |
| --- | --- |
| Brief Description | **Update Staff.** |
| Basic Flow | This use case describes how a admin to update staff from system.   1. Admin have to login. 2. Select the update button. 3. Enter the details you want to update. 4. Update following details.  * Staff First Name. * Staff Last Name. * Email Address. * Designation * Phone Number |
| Alternate Flow | 1. The system will validate the information provided. If any invalid data is found, the input form will be redirected with error message. |
| Validation | 1. Staff Id is not required as it is registered before and you can’t change it. 2. Provided details should be valid. |
| Pre-Conditions | Admin should have network access and Browser with latest updates. |
| Post-Conditions | Success popups should be shown. |

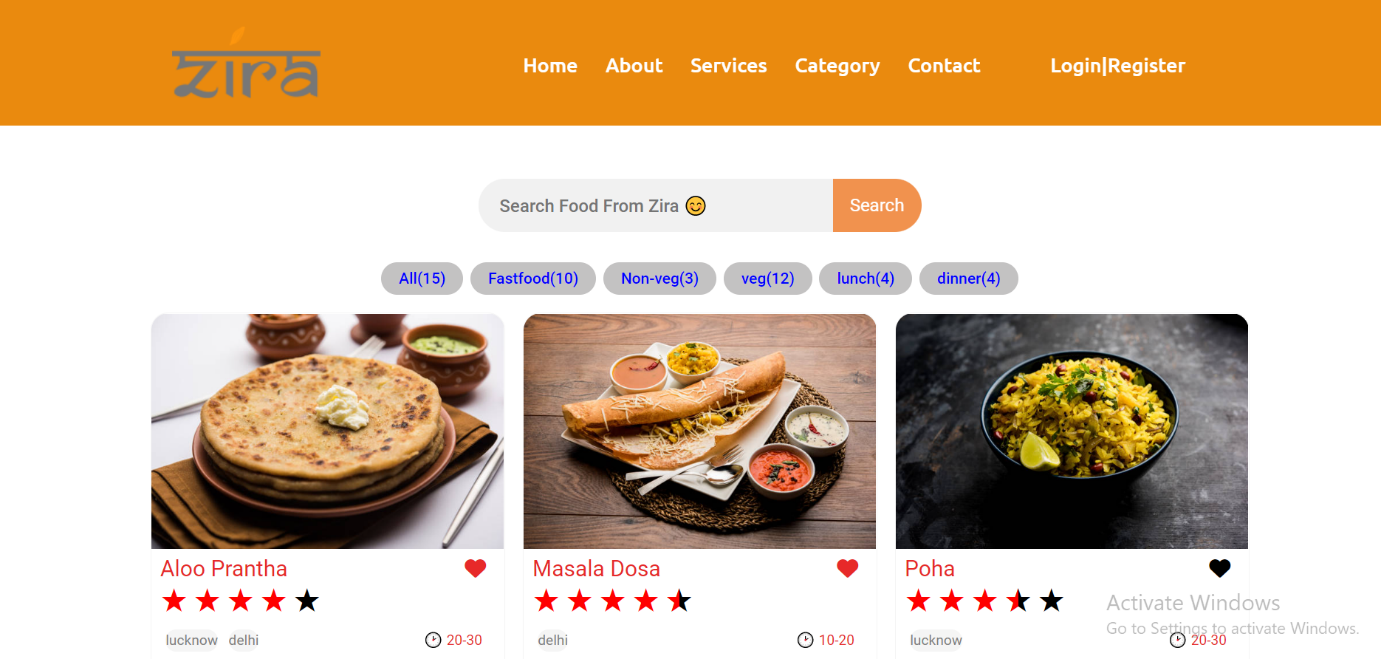
**Staff List:**

****

**Booking List:**

****

**Search:**

****

**Data Flow Diagram:**

A data flow diagram should be the first mechanism used by framework analyst to model system fundamental. These fundamentals are the system operations; the information used by this operations and external organization that interact with the framework and the data flows in the system.

Feedback Management

Inventory Management

Restaurant Management System

Employees Management

Orders Management

Menu Management

Admin

Check Credentials

Login to System

Add Employee Details

Manage Modules

Update Employee Details

Delete Employee Details

Fetch Employee Details

Delete Order Details

Fetch Order Details

Add Inventory Details

Update Inventory Details

Add Menu Details

Delete Inventory Details

Update Menu Details

Fetch Inventory Details

Delete menu Details

Fetch Menu Details

Delete Customers Details

Fetch Customer Details

Add Supplier Details

Update Supplier Details

Delete Supplier Details

Fetch Supplier Details

**ER Diagram:**

