Bon Appetit

Software Requirements Specification

2.0

09/06/2019

**Priyanka Galla**

**Naga Sai Manoj Goppisetty**

**Nithya Vudayamarri**

**Nikitha Mandala**

**Lahari Thamatam**

**Vijaya Raja Mayuri Akula**

**Harshavardhan Reddy Bollam**

Submitted in partial fulfilment

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Date** | **Description** | **Author** | **Comments** |
| 09/06/2019 | Version 2.0 | Priyanka Galla  Naga Sai Manoj Goppisetty  Nithya Vudayamarri  Nikitha Mandala  Lahari Thamatam  Vijaya Raja Mayuri Akula  Harshavardhan Reddy Bollam | First Revision |
|  |  |  |  |
|  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**1.Introduction**

1.1.Purpose

1.2.Scope

1.3.Overview

**2.General Description**

2.1.Product Perspective

2.2.Product Functions

2.3.User Characteristics

2.4.General Constraints

2.5.Assumptions and Dependencies

**3.Specific Requirements**

3.1.External Interface Requirements

3.1.1.User Interfaces

3.1.2.Hardware Interfaces

3.1.3.Software Interfaces

3.1.4.Communications Interface

3.2. Functional Requirements

3.3. Use Cases

3.4. Class/Objects

3.5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

3.5.3. Availability

3.5.4. Security

3.5.5. Portability

3.6. Inverse Requirements

3.7. Design Constraints

3.8. Logical Database Requirements

3.9. Other Requirements

3.10. Prototypes (for complete project)

3.11. Use Case Diagrams

**4. Design**

4.1. ER diagram

4.2. GUI

**5. Technical Manual**

**6. User Manual**

1. **Introduction**

**1.1 Purpose:** The purpose is it is a modern day application for delivering food to the doorstep from favorite restaurants nearby with fastest delivery services and easy accessibility .

**1.2 Scope:** This is a food delivery web page along with an authentication and a new user can login through their credentials. User can search for different restaurants & cuisines in different locations. After placing the order the user is directed to the payment page, once the payment is done user will get a popup message like the order is placed successfully and users can track their order.

**1.3 Definitions, Acronyms, and Abbreviations**

**1.4. References**

**1.5. Overview:**

In this application, we will be developing a Home Page, Login Page, a Menu for all the restaurants we are tied-up with, allows the users to place their food Orders, also allows them to give their Ratings and Reviews for orders they placed and for the help from customer care if they got any, a Payment Page, a facility to apply Promotional Codes and use various restaurant provided and app provided Deals.

**2. General Description**

**2.1. Product Perspective**

The Bon Appetit Food Application project uses the Web development framework and is completely independent. Users can access this food application in windows operating system.

**2.2. Product Functions**

With this product, the users will be able go and check the different types of food available in different restaurants nearby. The user is given with an account which is linked to their mail, so that user will be able to access the orders and deals available. User can also send the selected items to the cart, they can view the reviews of the restaurant, nearby restaurants details, timings, menu and many more features. Additionally, the app provides the regular users exciting offers, coupons for new users to increase the users for the app.

**2.3. User Characteristics**

The customer is the only user. A set of instructions is provided in the application which would be enough for the user to order food items from the applciation.

**2.4. General Constraints**

Place definitely matters a lot similarly, the products are limited to particular places only. The product can accessed on the web.

**2.5. Assumptions and Dependencies**

The product is available only for Web. Users can add previous orders to the current orders. User can track the delivery using the tracking details .And also payment methods can be saved.

**3. Specific Requirements**

**3.1. External Interface Requirements**

**3.1.1. User Interfaces**

* A good user interface provides a "user-friendly" experience.
* GUI of our software program includes many user-friendly controls like
* A Menu Bar
* Input Selector
* Images
* Buttons, etc.

**3.1.2. Hardware Interfaces**

* This application is based on Web Development.
* Pentium Processor
* 60 MB of free hard-drive space
* 128 MB of RAM

**3.1.3. Software Interfaces**

* Operating System: Windows (Vista/7 or above)
* Web Browser: IE 10 or above, Mozilla FF 31 and above or Google Chrome
* Database used: MongoDB
* Drivers: Java Runtime Environment
* Integrated Development Environment: Eclipse J2EE or Apache Tomcat, Visual Studio, HTML, CSS and JavaScript.

**3.2. Functional Requirements**

**Home Page:** In home page we have search text box, search button where the users can search for cuisines and restaurants.

**Login Page:** Registered users can login with their username and password. New users can sign up. If users forget their password they can reset their password using email.

**Menu:** The users can view menu for any restaurant by clicking the menu.  
**Orders:** Items which are ordered can be viewed in orders.

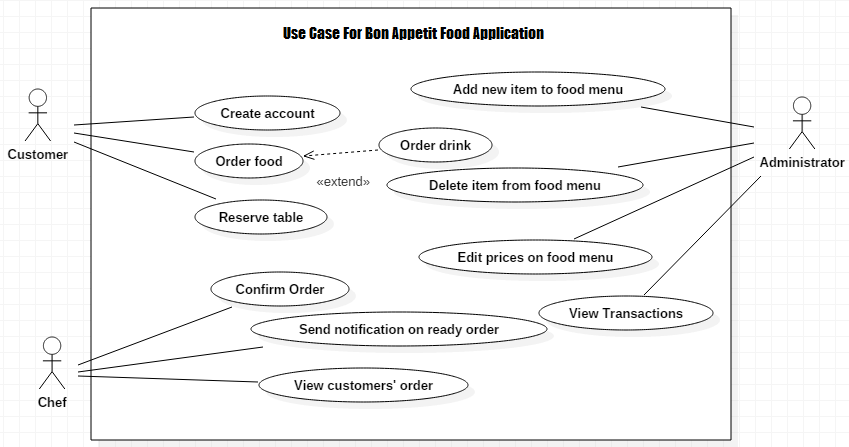
**Ratings and Reviews:** Users can give rating and review for a restaurant and view the other ratings and reviews also.

**Payment :** Users can pay using their credit/debit card and cash on delivery.

**Promotional Codes:** Using this code users can get a discount on the order.

**Deals:** Users can view deals applicable on any particular restaurant.

**3.3. Use Cases**

****

**3.4. Class/Objects**

**3.5. Non-Functional Requirements**

**3.5.1. Performance:**

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

**3.5.2. Reliability:**

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

**3.5.3. Availability:**

Availability is gauged by the period of time that the system’s functionality and services are available for use with all operations.

**3.5.4. Security:**

One or more requirements about the protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time ...) to break into the system.

**3.5.5. Portability:**

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

**3.6. Inverse Requirements**

**3.7. Design Constraints**

**3.8.Logical Database Requirements**

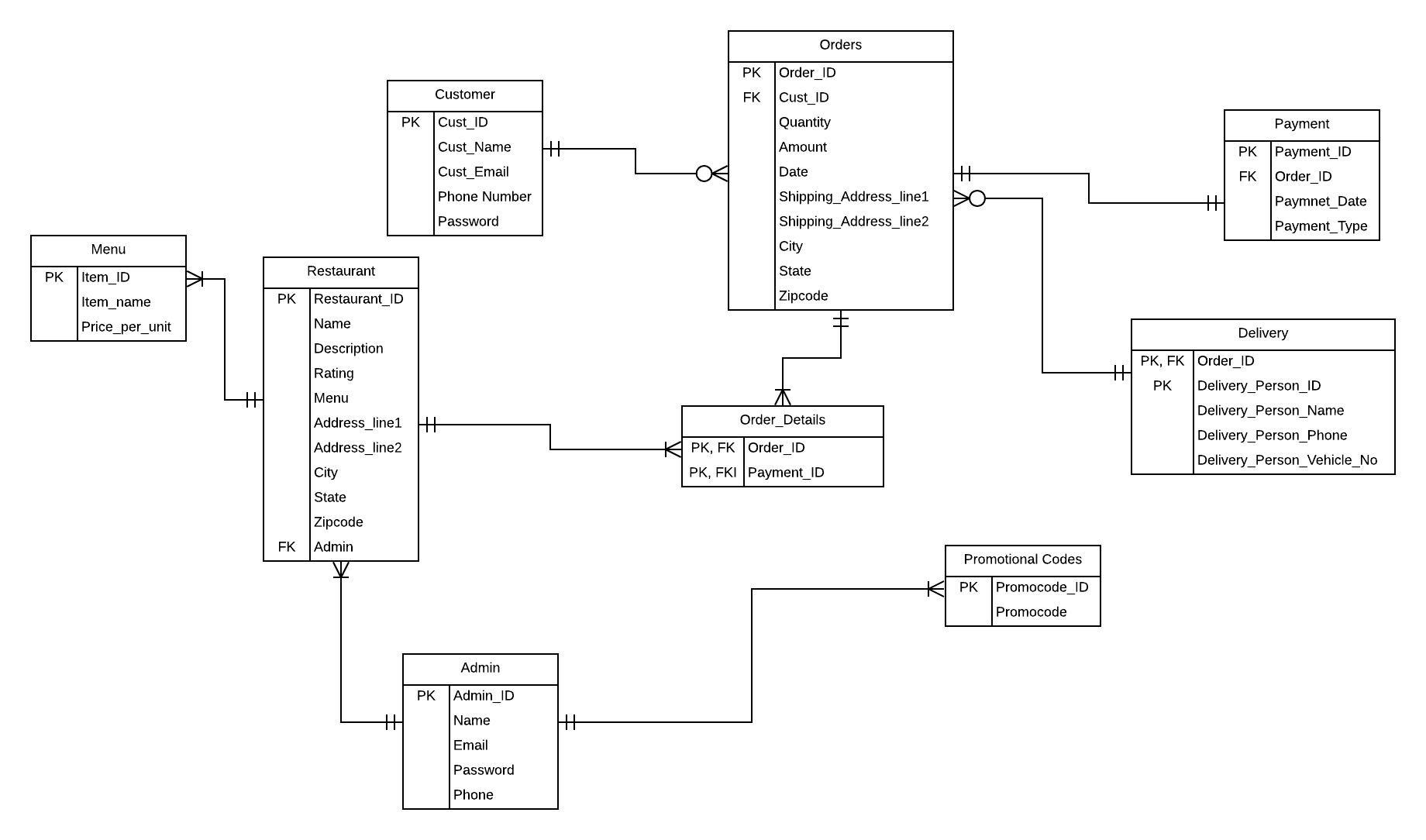
**3.9.Other Requirements**

**3.10. Prototypes (for complete project)**

**3.11. Use Case Diagrams**

**4. Design**

**4.1. ER diagram**

****

**In the above Entity Relationship diagram, it has 9 entities namely Customer, Restaurants, Admin, Orders, Order\_details, Promotional codes, Menu, Delivery and payment. The customer can be considered as user and admin can be considered as seller. So in the above diagram, we can also see that we have only one associative entity that is Order\_details which is a weak entity. Customer\_id, Restaurant\_id, Admin\_id, Order\_id, Promocode\_id, Payment\_id, Menu\_id, Delivery\_id, are primary and foreign keys.**

**Relationship:**

**Customer: Orders - 1: M**

**Orders: Payment - 1: 1**

**Orders: Delivery - 1: 1**

**Orders: Order\_details - 1: M**

**Restaurant: Order\_details - 1: M**

**Orders: Promo codes - 1:1**

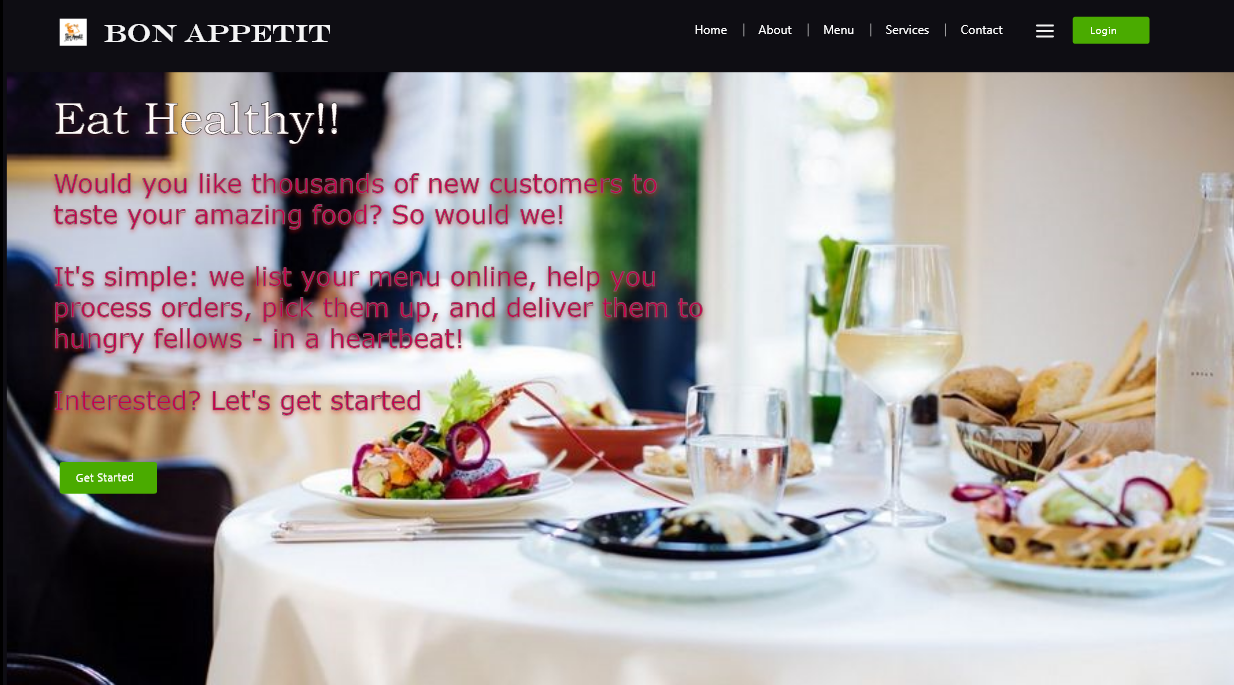
**Restaurant: Admin - 1: M**

**Admin: Promo codes - 1: M**

**4.2. GUI**

**Modified GUI:-**

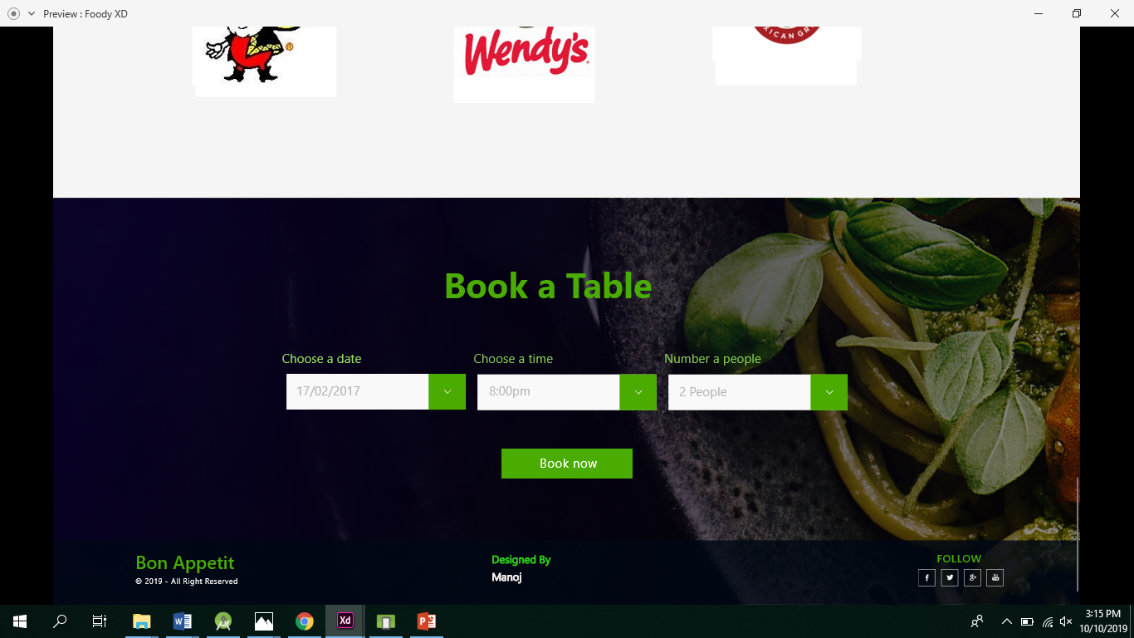
* + **Home Page :**

****

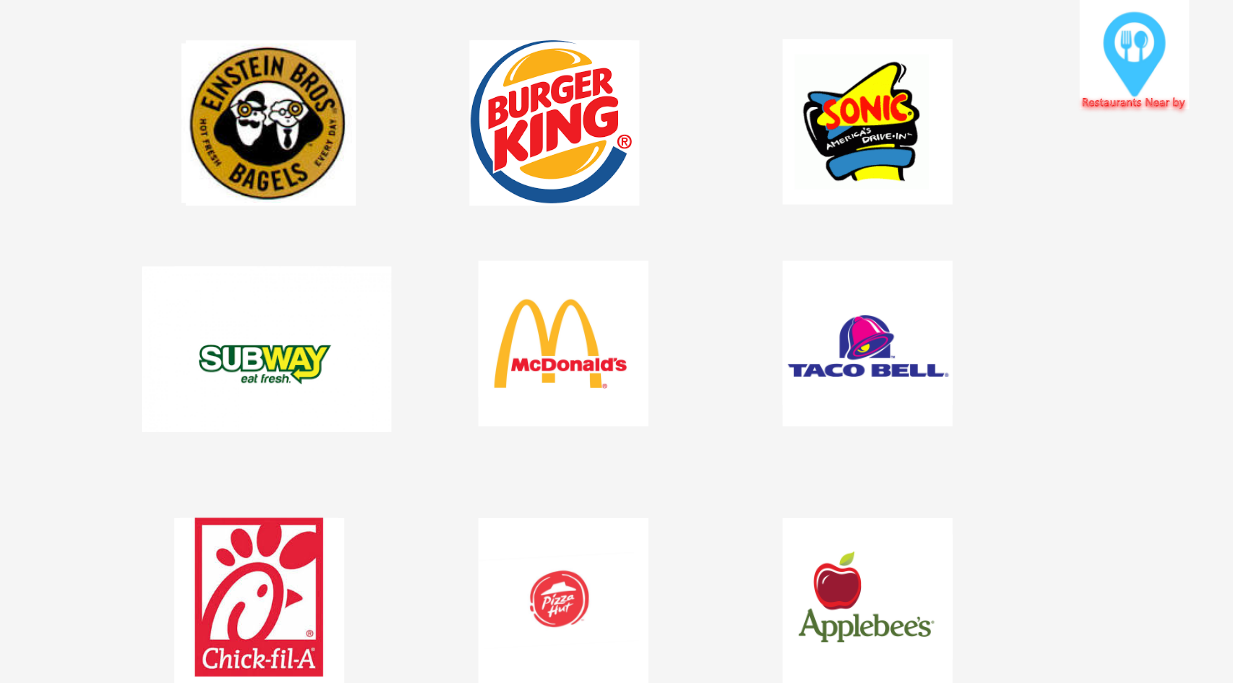
* + **Login Page:**

****

* + **Reservation Page:**

****

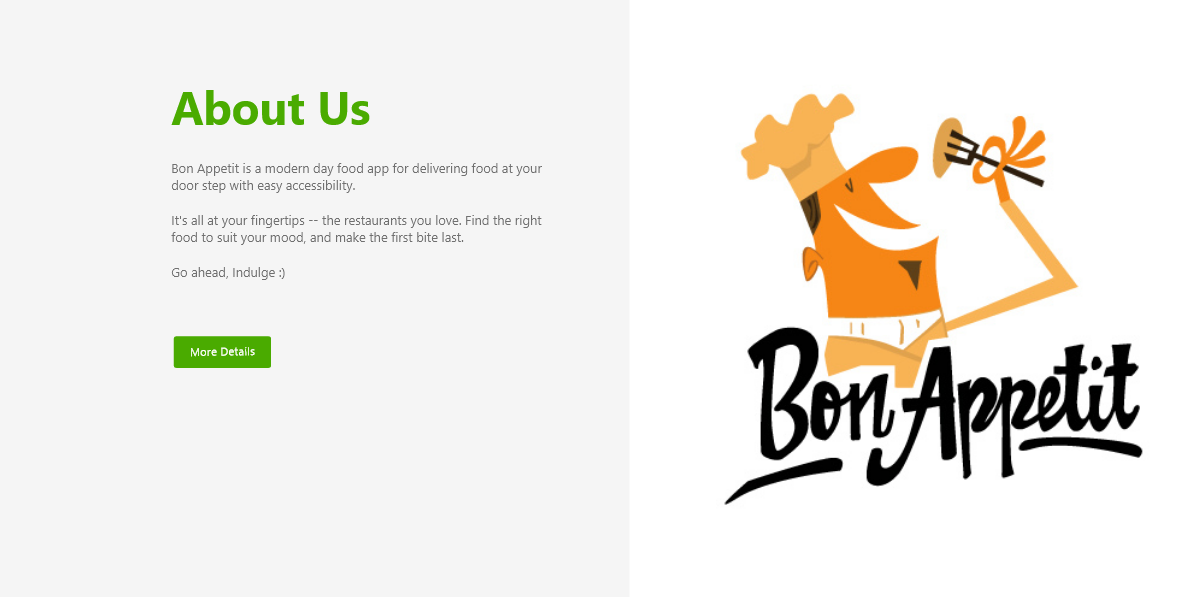
* + **Restaurant Page:**

****

* + **Sign Up Page:**

****

* + **About us page:**

****

* + **Payment Page:**

****

**6. Technical Manual**

**6.1 List of Activities**

1. **Home Page Activity**
2. **Sign Up Activity**
3. **Login Activity**

A brief explanation of the code related to each of these activities that we have implemented so far:

**1. Home Page Activity:**

The Home Page activity of our app has all the HTML, CSS and JavaScript code that gives the user a brief overview of our app that constitutes our app logo, a search option to search for food items or restaurants and all the other copyright details etc. It also has a navigation bar on the top of the screen. This nav bar has links to the Home Page itself, the About Us page, the Restaurants page, the Reservation page, the Contact page and also two buttons that allow the user to sign in and sign up.

🡪 **Home Page Front-End Code:**

Main body code of the home page:

<main>

<div style="font-family: Times New Roman; margin-top: 50px;">

<h1>

Bon Apetit

</h1>

<div width="1000px" height="1000px" style="font-size: 20px;">Feeling hungry?<br/>

Feeling tired and couldn't cook your meal now?<br/>

You are having a bunch of company files left to complete but food has already taken over your mind?<br/><br/>

We are here to help you and suppress your hunger right away!!!<br/>

From swanky upscale restaurants to the cosiest hidden gems serving the most incredible food, Bon Apetit covers it all.<br />

Explore menus, and millions of restaurant photos and reviews from users just like you, to find your next great meal.

</div>

<br />

<button onclick="location.href = 'login'">GET STARTED</button>

</div>

</main>

A header that includes a nav bar with home, about, contact, login, register tabs:

<header>

<!-- Fixed navbar -->

<!-- https://fezvrasta.github.io/bootstrap-material-design/docs/4.0/examples/sticky-footer-navbar/ -->

<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">

<a class="navbar-brand" href="/">

<div>

<img src="../../images/app\_logo.jpg" class="img-fluid" style="width:40px; height: 40px;">

</div>

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse"

aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<ul class="navbar-nav mr-auto">

<li class="nav-item">

<a class="nav-link active" href="/">Home<span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="/about\_us">About Us</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/contact">Contact</a>

</li>

<% if (!user) { %>

<li class="nav-item">

<a class="nav-link" href="/login">Login</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/register">Register</a>

</li>

<%} else { %>

<li class="nav-item">

<a class="nav-link" href="/logout">Logout</a>

<% } %>

</ul>

</div>

</nav>

</header>

1. **Sign Up/Registration Activity:**

We have a button on the nav bar of every page and also on the home page to lead us to the sign up page. This page has the front end that enables user to give all the user’s details such as name, mobile number, email etc. The back end of this page stores the details in the database that we maintain to store all the user profile.

**🡪 Signup/Register Page Front-End Code:**

<main>

<div style="font-family: Times New Roman; margin-top: 50px;">

<h1>

REGISTER

</h1>

<form action="register" method="post">

<div class="sm 6" style="position: absolute; left: 200px;">

<span class="labels">USER NAME</span>

<input type="text" placeholder="" name="username" required>

<br /><br />

<span class="labels">GENDER</span>

<input type="text" placeholder="" name="gender" required>

<br /><br />

<span class="labels">AGE</span>

<input type="text" placeholder="" name="age" required>

<br /><br />

<span class="labels">EMAIL</span>

<input type="text" placeholder="" name="email" required>

<br /><br />

<span class="labels">MOBILE NUMBER</span>

<input type="text" placeholder="" name="mobile\_number" required>

<br /><br />

<span class="labels">ADDRESS LINE-1</span>

<input type="text" placeholder="" name="address\_line\_1" required>

<br /><br />

</div>

<div class="sm 6" style="position: absolute; right: 200px;">

<span class="labels">ADDRESS LINE-2</span>

<input type="text" placeholder="" name="address\_line\_2" required>

<br /><br />

<span class="labels">STREET</span>

<input type="text" placeholder="" name="street" required>

<br /><br />

<span class="labels">CITY</span>

<input type="text" placeholder="" name="city" required>

<br /><br />

<span class="labels">STATE</span>

<input type="text" placeholder="" name="state" required>

<br /><br />

<span class="labels">ZIPCODE</span>

<input type="text" placeholder="" name="zipcode" required>

<br /><br />

<span class="labels">PASSWORD</span>

<input type="password" placeholder="" name="password" required>

<br /><br />

</div>

<br /><br />

<br /><br />

<br /><br />

<br /><br />

<br /><br />

<br /><br />

<br /><br />

<br /><br />

<button type="submit">REGISTER</button>

</form>

</div>

<br />

</main>

1. **Login Activity:**

We have a button on the nav bar of every page and also on the sign up page to lead us to the login page if the user is already a member on the app. This page has the front end that enables user to enter the username and the password. The back end of this page allows the user to login to the app in order to perform all the activities on the app such as search for food items, restaurants, order food, make payments, cancellations etc. if the given credentials are correct.

🡪 **Login Page Front-End Code:**

<main>

<div style="font-family: Times New Roman; margin-top: 50px;">

<h1>

LOGIN

</h1>

<br />

<div>

<form action="login" method="post">

<input type="text" placeholder="Enter Username" name="uname" required>

<br /> <br />

<input type="password" placeholder="Enter Password" name="pwd" required>

<br /> <br />

<button type="submit">LOGIN</button>

</form>

<label for="new user">New user?</label>

<label id="label\_register\_here" for="register here" onclick="location.href = 'register'"><b>Register here</b></label>

<br />

<label>

<input type="checkbox" name="remember"> Forgot Password?

</label>

</div>

</main>

**7. END USER MANUAL**

**7.1 How to Login**

You always need a login to authenticate yourself and access our personal information and the application pages. You can log in on the Bon Appetit homepage.

* **Username**: Type your username in this field
* **Password**: Provide your password. If you enter your password incorrectly 5 times in a row during one session, access is locked for 5 minutes before you can try again.
* Click the Login button to access your account

**7.2 How to Register**

* If you do not have an account, and our application allows self-registration for an account. Go to the Home page where you can find a register Button or else You can find the register option in the login screen itself.
* Once you click the register button, you will be directed to the Registration screen where you can create a new account.
* Provide the required information in the registration form and once you are done filling the form, click submit.
* Then a new account would be created for you.

**7.3 How to Retrieve Password**

* If you do not remember your username or password to your account, click on the forgot password field which would take you to the password retrieval page.
* There you can request an email with your details.
* It will contain your username and a link to change your password.
* You can click the link provided and create a new password.