System Design Document

Project: <project name>

By: <Vijendra Singh>

# Architecture

## Web Framework

1. **User Interface (responsive for Mobile, Tablet and desktop View):**

This is the front end of the application where user can browse search for restaurants.

View menu, location, rating and perform action to save as favorite, order and pay

1. **Client-side Components:**
2. User Authentication:

These components handle user registration, login and authentication.

1. Search Interface:

Where user can view restaurant listing, search query and filter result.

1. User profile:

User can create and mange their profile and add delivery address, contact detail, payment method, favorite restaurants.

1. **Server-side Components:**
2. Application Server:

Core application to handle user requests and communicate with database, manages business logic and maintain flow of data between various components.

1. Database:

To store data related to restaurants, user, order, review etc. we will use default SQLite database.

1. Media:

To store all images related to restaurants, user, order, review etc.

1. Order Processing:

Manage order Lifecycle including order placement, payment processing, order Tracking and send notification to user and restaurant.

1. Payment Gateway:

To securely handle payment transaction and validation.

1. Geolocation Services:

Providing geolocation services for search nearby restaurants, location order and provide estimate delivery time.

1. **Notification:**
2. Push Notification:

Send notification to user for order status, order tracking.

1. Email Notification:

Send order confirmation, receipts and promotion/marketing email to user.

1. **Admin Dashboard:**

Provide Restaurant owner and admin to create, edit and delete restaurants listing, menu, review, and order.

Design Architecture

* Project\_FoodSearch:
* App\_UserProfile:
  + Model
    - FK:User:one-to-one relationship with user
    - Phone\_number:
    - Address
    - Payment\_method
    - Date\_created:
    - Date\_modified
* App\_Resturant:
  + Model
    - Name
    - Location
    - Owner
    - Description
    - Menu\_Item
      * name
      * description
      * price
    - Date\_created:
    - Date\_modified
* App\_Order:
  + Model
    - FK:User
    - FK:Resturant
    - Order\_item
    - Order\_date
    - Order\_status
    - Delivery \_address
    - Order\_amount
* App\_Review:
  + Model
    - FK: User
    - FK: restaurant
    - Rating
    - Comment
    - Date\_created:
    - Date\_modified
* App\_Favorties\_Resturants
  + Model
    - Fk:User
    - FK:one- to-many: restaurant
    - Date\_created:
    - Date\_modified
* Template
  + Admin dashboard
  + Home page
  + Listing page
  + Edit listing page
  + Add listing page
  + Login page
  + Registration page
  + User profile page
  + Edit user profile page
  + Favorite restaurant page
  + Order page
* Static
  + Images
    - Company\_Logo
    - Default\_images
      * Profile\_image
      * Restaurant\_image
      * Menu\_item\_image
* Media
  + Resturant
    - Logo
    - Menu
    - Menu\_item
  + User
    - Profile image
* Database
  + SQLite

# Data Access Design

## Model Design (Data Model)

*This is the “model” part of the MVC pattern.*

*We refine the high-level business domain model and create a detailed data model with all the entities that we will need to store in the database, with their attributes and relationships.*

## Functional Decomposition

*Select the key entities in the data model. From the user stories identify the types of functions needed on each entity, and which users will be performing them. This will provide you with a basis for identifying the Views that you will need.*

## Database Investigation

*List which database management systems you investigated, and which is more appropriate for your web app. Justify your decision.*

# Security Design

## Framework Security

Which security features are provided by the web framework, which ones will you use?

## Security Mechanisms

How will you implement each of the security mechanisms (if provided by framework, how will you use them, e.g. which settings, etc.)

## User Roles and Permissions

# User Interface Design

## View Design (UI List)

*This is the “view” part of the MVC pattern.*

*List the user interfaces needed to perform each user story.*

### User Story 1

### User Story 2

## UI Design

Record palette, fonts, look and feel, bootstrap used, etc. for your web app.

Include a wireframe or mockup for the landing page to illustrate the style you are aiming for.