

# Software Requirements Specification (SRS)

## Cravings Dash

### List of Stakeholders

Edwin Rodríguez, Noah Renella, Dennis Jimenez, Natalie Chang, Dawson Vong, Joshua Botkin

### Description of Problem

A lot of food service and delivery applications do not focus on learning the user's preferences based on their order history or search. It is also very inconvenient for users to search for food that meets their strict dietary needs (e.g., vegan, vegetarian).

### Document Purpose

This document presents a detailed description of the objectives, features, and requirements to build an online system to manage food booking and delivery from the users' favorite restaurants.

## 1. Product Scope

- a. Cravings Dash is a web application service that allows users to order food online and get it delivered. It will also allow users to sort and filter restaurants, catering towards specific categories based on their preferences (such as vegan-friendly).
- b. Background  
Amazon is one of the most successful companies in e-commerce. The company makes it possible for people to buy anything that we could list, except foods from restaurants. There are already a lot of food booking services, but they have not optimized the users experience yet. This is a chance for us to deliver a more complete and efficient version of all the food delivery apps in the market.
- c. References  
Edwin Rodríguez

## 2. Overall Description

- a. Product Perspective  
A distributed food booking database system stores the following information:
  - Customer Description:  
It includes customer ID, name, address and phone number. This information is used for delivering food to the customer.  
It also includes preferences of the user and food booking history of the users in order to display a more appropriate menu to the users.
  - Restaurant Description:  
It includes the menu of the restaurants and the rating of them.  
There are also names, ID, descriptions of the restaurant about what it serves, phone and address. These could help the users to have an

**Order Description:**  
It includes the name, itemID, date, totalPrice, restaurant, address, status of delivery, and the name of the delivery driver, so that the users are able to keep track of what they are ordering and what from where and when.

The major features of food booking database system as shown in below entity-relationship model (ER model)

The diagram illustrates the database structure for a food delivery system. It includes the following entities and their attributes:

- Driver**: DriverID (PK), Profile (FK), Rating (int), Available (boolean).
- Profile**: ProfileID (PK), FirstName (string), LastName (string), Username (string), Password (string), Email (string), Phone (int), Recommended (FK).
- Order**: OrderID (PK), Date (datetime), Profile (FK), TotalPrice (double), Restaurant (FK), ToAddress (FK), Shipped (boolean), Delivered (boolean), Driver (FK).
- Order\_MenuItem**: OrderID (FK), MenuItemID (FK, PK).
- Restaurant**: RestaurantID (PK), Name (string), Description (string), Phone (int), Address (FK), Rating (int).
- Restaurant\_Tag**: RestaurantID (FK, PK), TagID (FK, PK).
- Menu**: MenuID (PK), RestaurantID (FK), Name (string), Description (string).
- MenuItem**: MenuItemID (PK), MenuID (FK), Name (string), Description (string), Price (double).
- MenuItem\_Tag**: MenuItemID (FK, PK), TagID (FK, PK).
- Profile\_Address**: AddressID (FK, PK), ProfileID (FK, PK).
- Address**: AddressID (PK), AddressLine (string), City (string), State (string), Zipcode (int), Country (String).
- Tag**: TagID (PK), Tag (string).
- Review**: ReviewID (PK), Author (FK), Target (FK), Date (datetime), Title (string), Body (string), Rating (int).
- Preference\_Tag**: ProfileID (FK, PK), TagID (FK, PK).
- Restriction\_Tag**: ProfileID (FK, PK), TagID (FK, PK).
- Profile\_Recommend\_MenuItem**: ProfileID (FK, PK), MenuItemID (FK, PK).

Relationships are indicated by lines with crow's foot notation, showing one-to-many and many-to-many connections between the entities.

c. User Characteristics

The user should be able to do the following functions:

- Create an order:
  - Add items to the order list from different restaurants
  - Set up the time and date to deliver
  - Adding coupon (if necessary)
  - Set payment method and check out
  - Confirmation
- Cancel an order (if not shipped)
  - Choose specific item to cancel
  - Cancel confirmation
- View order history
- Review item and restaurant
  - Allow to review restaurants even order is not delivered
- Searching for key tag
  - Filter for some type of food

d. Operating Environment

Operating environment for the for the food booking services:

- Distributed database
- Client/server system
- Operating system: any
- Database: MongoDB database
- Platform: Javascript

### 3. Requirements

a. External Interface Requirements

i. User Interfaces

1. Front-end software: React
2. Back-end software: Node.js, Express, MongoDB

ii. Hardware Interfaces

1. Windows
2. A browser which supports HTML, Javascript

iii. Software Interfaces

1. Operating System: For user-friendliness, any operating system can be used.
2. Database: to save the users' records, order records, restaurants. We have chosen MongoDB.
3. Language: to implement the project we have chosen Javascript

iv. Communication Interfaces

1. This project should support all types of web browsers and android systems.

b. Functional Requirements

- i. There will be a prompt of preferences for the users to select what they like when they first create an account, so that the web displays the foods and restaurants based on the user preferences.
- ii. The web changes the users' preferences over time based on the order history of the users.
- iii. Users should be able to filter vegan and non-vegan menus.
- iv. Users should be able to review restaurants appropriately (be able to review if the order was not delivered as it was a mistake of the restaurant).
- v. The basic functions of a food booking service should have.
- vi. Users should have the estimated delivery time of the food before they order.

c. Non-Functional Requirements

- i. Performance requirements
  1. All of the application data is stored in a MongoDB database.
  2. The server hardware can be any computer that is capable of running both the web and database servers and handling the expected traffic.
- ii. Security requirements
  1. Users should be required to log into their account for all operations except viewing the menu.
  2. User's personal information like phone number or credit card information should be encrypted before storing in the database.
- iii. Software quality attributes
  1. Reliability: The service should deliver the food as it is described on the time it stated.
  2. Availability: The user should be able to access the system using a web browser at all times and on any android devices.
  3. Maintainability: The service should be able to keep track of the delivery time and the update menu of the restaurant as well.
  4. Usability: There should be any type of food on the service menu as long as the user could list it.