School of Engineering and Applied Science,
Ahmedabad University

Software Engineering (CSE300)

Winter Semester - 2020/21

Project:

Food Ordering System - Foodigo

Requirements Understanding Document

Yashil Depani	AU1841005
Ridham Shah	AU1841007
Rushil Patel	AU1841008
Harshil Mehta	AU1841010
Raj Mehta	AU1841018
Vidish Joshi	AU1841019
Manav Patel	AU1841037
Harsh Agola	AU1841106
Sanket Shah	AU1841111

1. Introduction:

1.1. Purpose of the System

Food delivery systems are designed to make customer's food eating experience at home better alongside making Restaurant's business grow and profitable.

The proposed system aims to make the customer's eating experience better by providing varied up-to-the mark restaurant options to choose from and integrated delivery system with tracking. The system should help the food serving businesses by working as a bridge to bring more and varied customers and provide good experience to both customers and the business to retain customers. The system should also have a good model to track the delivery drivers and maintain the delivery drivers' information.

In essence the system should make the entire food delivery process seamless and enjoyable for the customer, the business as well as the delivery drivers.

1.2. Scope of the system

- The system should be able to maintain 3 types of users: the *customer*, the *business(restaurant)*, and the *delivery drivers*.
- The system should provide the customers choices of restaurants to choose from and menu options per each restaurant to choose what to order.

- The system should free the customer to get rid of the hassle of going to the restaurant in person by automatically assigning delivery drivers for their order.
- All the users should be able to sign in and out of the product. The users should be able to use their accounts for placing/accepting/delivering orders.
- The system should be secure and ensure the user's information safety.
- The system's UI should be friendly. Restaurants can customize their menu items, customers and add or delete orders and delivery agents can check the necessary information required to deliver the orders.
- All the users should be updated in real time about all the ongoing orders and their status.
- The system should ideally be deployed online so as to reduce the technical requirements on parts of the users.

2. Proposed System:

2.1. Overview of the system

- We have attempted to create a food ordering system and management system which can help all concerned parties; from customers to restaurant owners to delivery drivers.
- The app will provide different levels of access to the app based on the type of user using it.

- Customers can browse through the catalogs of registered restaurants and can order their favourite food via various online payment services as well as track the order.
- Restaurant owners can add food products to their food catalog, update certain information regarding them or remove food products if the restaurant has discontinued making it.
- Delivery drivers can view the order list and get all the required information like order and restaurant information and customer or user information.

2.2. Stakeholders

2.2.1. Stakeholder Requirements (Elicit requirements for stakeholders)

Stakeholder	Requirements
Customer	Register to the app. (User account)
	Login to the app.
	Edit personal details.
	Navigate through all the restaurants.
	Browse their food catalog.
	Add an item to the current order.
	Review the order before placing it.
	Update item information in the current order.

	Remove items from the current order.
	Choose payment options.
	Place the order.
	Track the order.
	Provide feedback to the app, the restaurant and the delivery driver.
Restaurant Manager	Register to the app. (Restaurant account)
	Login to the app.
	Edit restaurant information.
	Add food products to the catalog.
	Update information in the catalog.
	Remove food items from the catalog.
Delivery Drivers	Register to the app. (Restaurant account)
	Login to the app.
	Edit personal details.
	View order information.
	View information of the user who placed the order.
	View information of the restaurant where the order is placed.

3. Technical Requirements

3.1. System and Software requirements:

Dependency	*Version*
Python	*>= 3.6*
Django	*3.2*
Sqlite	*3.0*

4. Business Activities Requirements

As mentioned in **2.2**, the proposed business model has 3 stakeholders: the *customer*,the *restaurant*, and the *delivery driver.* Our business requirements should act for making the stakeholders' experience good and seamless.

4.1. Easy on-boarding process (BR1)*:*

All the stakeholders should have friendly and easy on boarding experience. New users should be able to sign up and login to the system. Necessary and required information should be taken from them at the time of signup.

4.2. Customizability of information (BR2)*:*

The stakeholders should be able to change the information they have given to the system.

- The customer should be able to change personal information as well as order information.
- The restaurant should be able to change its business information including the menu items and the pricing details, etc.
- The delivery drivers should be able to customize their personal information.

4.3. Information security (BR3)*:*

- The information provided by the users should remain safe while it is transferred via servers and to the end systems.
- For this, authenticating and authorizing of the users should be done before they make use of any features.

4.4. Real time system (BR4)*:*

The data shown to the users should be real time in nature.

- The customer should be able to place the order and receive order related updates in real time.
- The restaurant should be able to receive orders and change the status of the order in real time.
- The drivers should be able to their assigned order and preparation status in real time.

4.5. Food service (BR5)*:*

The entire process from ordering to delivery should be seamless.

- Customers should be able to choose and update from restaurants and from menu options and order desired items.
- Restaurants should be able to see placed orders and change the order's status till handed over for delivery.
- The delivery person should be able to check the assigned orders, the orders' status and change the orders' status.

5. Functional Requirements

5.1. User-based access to data (FS1)

5.1.1. Users should be able to sign-up/sign-in as per their type and access the features available to their type of user.(FS1.1)

5.1.2. Users should be able to access restaurant catalog data and *all* of their personal information. (FS1.2)

5.1.3. Restaurant owners should only be able to access the order information. (FS1.3)

5.1.4. Delivery drivers should only be able to access the restaurant information and some rudimentary information about the order and user. (FS1.4)

5.1.5. Administration should be able to access all the information. (FS1.5)

5.2. Modification privileges(FS2)

5.2.1. Users should be able to change their personal information in the app. (FS2.1)

5.2.2. Restaurant owners should be able to change information regarding the restaurant. (FS2.2)
5.2.3. Restaurants owners should also be able to change / update their food prices and other catalog information. (FS2.3)
5.2.4. Administration should be able to block any user. (FS2.4)

5.3. User Accounts (FS3)

5.3.1. User accounts should always be password protected. (FS3.1)
5.3.2. Passwords should be handled by the app itself without bothering the administration. (FS3.2)
5.3.3. All users should be confined in the system according to their roles except for the administration account. (FS3.3)

5.4. Data Integrity (FS4)

5.4.1. If possible, alerts and/or notifications will be sent to the users whenever a change in any data value occurs. The change can be payment made, order placed, or status of the order changed. FS(4.1)
5.4.2. Users will be suggested to update their profiles from time to time. (FS4.2)
5.4.3. New registrations must be reviewed by the administration before admitting those users into the system. (FS4.3)

5.5. Food Ordering (FS5)

5.5.1. Users should be able to navigate through different restaurants and view a restaurant's food catalog. (FS5.1)
5.5.2. Users should be able to add items to the current order as well as update quantities or specifications and delete items from it. (FS5.2)
5.5.3. Users should be able to choose a payment method of their choice. (FS5.3)
5.5.4. Orders should be verified and placed without any errors. (FS5.4)
5.5.5. Users should be able to provide feedback / ratings to restaurants and delivery drivers. (FS5.5)

5.6. Administration Panel (FS6)

5.6.1. The system should provide full administrative control to the administrators. (FS6.1)
5.6.2. Administrators should be able to create new entities and approve new profiles. (FS6.2)
5.6.3. Administrators should be able to suspend or even downright ban user accounts. (FS6.3)

5.7. System Security (FS7)

5.7.1. Users can only change their personal information after they have logged into the system. (FS7.1)
5.7.2. Updating a user's profile should require either that particular user's credentials or one of the administrator's credentials. (FS7.2)
5.7.3. Users should not have access to anything beyond their roles. (FS7.3)

5.7.4. Any user information (such as their email) must be visible only to the administration and the user itself; any other party should not be able to view any user details. (FS7.4)

6. Non-functional requirements

6.1. Scalability:

The current implementation of the system should be such that the further improvements in the system can be done in coming iterations with minimum change in the basic system structure. Care should be taken while choosing the technologies that are used while creating the first version of the system.

6.2. Reliability:

The system should be reliable in the sense that the current technologies and place of hosting if chosen should sustain the product for a prolonged period of time.

6.3. Regulatory:

Proper regulatory guidelines for the stakeholders should be put in place before launching the product and should be improved in future iterations. These regulations should include keeping a tab on the users based on their performance/reviews.

6.4. Security:

The system should be secure and should handle the information provided by the stakeholder with utmost care. The chosen technologies should have features that allow safe transfer of information between users and servers.

6.5. Compatibility:

The system should be compatible with modern browsers.

6.6. Availability:

When deployed, availability and performance of the system should be measured and maintained. Provisions should be made and system should be setup such that increase the availability and decrease the downtime.

6.7. Usability:

User experience should be of utmost importance while building the system. System should be easy to use and intuitive in flow to help bring in and retain users.

7. Requirements traceability matrix

Priority Levels:	High: These requirements are absolutely necessary to be implemented. Higher importance to these modules in sprints to be implemented first. Medium: These requirements are necessary for the product but their implementation is not of first priority. Low: These requirements can be considered as additional features for the product and are considered of lowest priority as they do not affect aim of the product in current version.

REQUIREMENTS TRACEABILITY MATRIX

Course Name:	Software Engineering (CSE300)	

Online Food Ordering System

Business Requirement ID:	Functional Requirement ID:	FS Task ID:	Description:	Priority	Status:
BR1	FS1	FS1.1	Sign-up / Log-in module for users of all type: Customer, Restaurant and Delivery driver	High	Pending
		FS1.2	Access to the restaurant menu table with view or edit access as per the user permissions.	High	Pending
		FS1.3	Access to the orders tabke with suitable permissions as per user type	High	Pending
		FS1.4	Access to the orders table with suitable permission as per the type of user	High	Pending
		FS1.5	Access to all the database tables with admin permissions	High	Pending
	FS6	FS6.1	Module should allow admin edit level access to all the databases	Medium	Pending
		FS6.2	Module to help the admins create new users	Medium	Pending
		FS6.3	Module to help the admins remove users	Medium	Pending
BR2	FS2	FS2.1	Module to allow a user to change self's information	High	Pending
		FS2.2	Module to allow restaurant to change general restaurant information	High	Pending
		FS2.3	Module to allow restaurant to change menu items	High	Pending
		FS2.4	Module to allow admin to edit or block any users and/or changes	Medium	Pending
BR3	FS3	FS3.1	Validate password protection required for login module	High	Pending
		FS3.2	Validate password does not leak from the current session	High	Pending
		FS3.3	Validate visibility permissions for each user type	High	Pending
	FS7	FS7.1	Validate edit permissions for user as per login status	High	Pending
		FS7.2	Validate module to allow edit access for users and admin	Medium	Pending
		FS7.4	Validate visibility permissions for each user type	High	Pending
BR4	FS4	FS4.1	Validate modules to show proper data changes to the user screen	High	Pending
		FS4.2	Module to keep track of user information and suggest updation/	Low	Pending
		FS4.3	Module to keep track of new users added to the system	Low	Pending
BR5	FS5	FS5.1	Module to allow customers to view restaurants and select restaurants	High	Pending
		FS5.2	Module to view menu, and items to cart while maintaing bill	High	Pending
		FS5.3	Payment module to make payment by the customer	High	Pending
		FS5.4	Module to verify the orders by the admin and the restaurants and accept or decline the orders	Medium	Pending
		FS5.5	Feedback module to give to the restaurants by the customers	Medium	Pending