Software Engineering (CSE300)

Project: **FoodFrenzy- Online Food Ordering System**

Group-08

Requirement Understanding Document

Yug Patel	1741017
Yesha Shastri	1741035
Ratnam Parikh	1741036
Devshree Patel	1741075
Param Raval	1741083
Sajil Vohra	1741094

1. Introduction

1.1. Purpose of the System

Online food delivery systems enable customers to save time and energy by having them decide which place to dine in on their phones, anywhere in the city. This also helps the customers to fit their budget while ordering, by reducing the traveling expenses and being sure of what they wish to eat. These systems connect the customer with numerous restaurants and cafes, bringing their menus on a pallet, and displaying several features of the location to help them make an informed decision.

1.2. Scope of the System

- The goal of the project is to provide a customer a system to find a personalised selection of food outlets to order easily from. The end product can be deployed as a web application, initially, and then quickly move on to a mobile application since most users prefer or have access to the latter.
- The user can visit the application and login to it using the allotted credentials (as a restaurant, customer, delivery agent or restaurant owner).
- The restaurant owners can manage the content of their restaurant/s such as add/update menus or offer discounts/promotions. A particular restaurant can accept or reject incoming orders and send updates (preparing or dispatched) of accepted orders.
- Moreover, the customer can browse the list of outlets and order from the menus while the delivery agent can collect/deliver order, accept payments, and view locations of the collection and delivery point

2. Proposed System

2.1. Stakeholder Requirements

Customer

- Register for an account
- Login to the account
- Edit account details
- Navigate through the restaurants
- Browse for food items
- Add an item for the current order
- Review current order
- Remove items from current order
- Provide payment details
- Track the current order
- Place an order
- Give ratings to the restaurant, delivery agent and FoodFrenzy

Restaurant Owner

- Register for an account
- Login to the account
- Add restaurants
- Update restaurants
- Delete restaurants
- Update food items in menu
- Update price of food items

2.2. Overview

- In this project, we are attempting to create an online food ordering system accessible using a web application. The customer will be able to browse restaurants in their city and area, choose and order items from the menu, and make payment online or offline
- The app will be up and running 24/7 and the customer can order as long as the outlet is open and available for delivery
- The other possible logins, in addition to the admin, are via the restaurant, the restaurant owners, and the delivery agent

2.3. Business Requirements

2.3.1 Ease of Access (BR1)

- The system should provide functionality for the customer and the admin to easily access and view their account details.

2.3.2 Provide functionality for updation (BR2)

- Different stakeholders should be allowed to make updations as per their roles.

2.3.3 Data Authentication (BR3)

- The accounts of stakeholders should be protected by password and appropriate functionality should be given to reset it.

2.3.4 Integrated System (BR4)

- Stakeholders and admin should be able to access and utilise the functionalities in the system as a whole.

2.3.5 Food Ordering Process (BR5)

- Appropriate functionalities associated with placing an order should be provided.

2.4. Functional Specification Requirements

- Data Accessibility (FS1)
  - Every user is only able to access his/her own data. (FS1.1)
  - Admin can access data for all restaurants, delivery agents and customers. (FS1.2)
- Entity Profile Administration (FS2)
  - Restaurant owners will request to add restaurants of their own, then the administration will check for credentials and add appropriate restaurants to the database. (FS2.1)
  - Administration can block users as per the situation against the user. (FS2.2)
  - User can edit personal details from the website (FS2.3)
  - Restaurant owners can edit the menu according to the will, on the website. (FS2.4)
- User Accounts (FS3)
  - User accounts are password protected. (FS3.1)
  - Password reminders and resets are handled by the website. (FS3.2)

    - With the exception of system administrators, all user accounts are tied to a system. (FS3.3)

- Data Integrity . (FS4)
    - If feasible, system will provide alert/notification upon sign-in by entity user when any of the dates or data elements requires updating (FS4.1)
    - Consideration will be given to requiring a Profile update (FS4.2)
    - New profiles must be reviewed and approved by website before appearing online (FS4.3)

- Food Ordering (FS5)
    - Customers should be able to browse restaurants and view the restaurants. (FS5.1)
    - Customers should be able to add or delete items for an order. (FS5.2)
    - Option for choosing payment type should be provided (FS5.3)
    - Order should be able to get placed and verified. (FS5.4)
    - Option for providing ratings should be given (FS5.5)

- Administration Panel (FS6)
    - System provides designated administrators with additional rights. (FS6.1)
    - Administrators create new entities and approve new profiles (FS6.2)
    - Administrators designate the first primary user for the entity. (FS6.3)
    - Administrators can create, edit, and disable user accounts. (FS6.4)
    - All restaurants, delivery agents will be added only by administrators upon request. (FS6.5)

- System Security (FS7)
    - Public has access to only personal details via the website. (FS7.1)

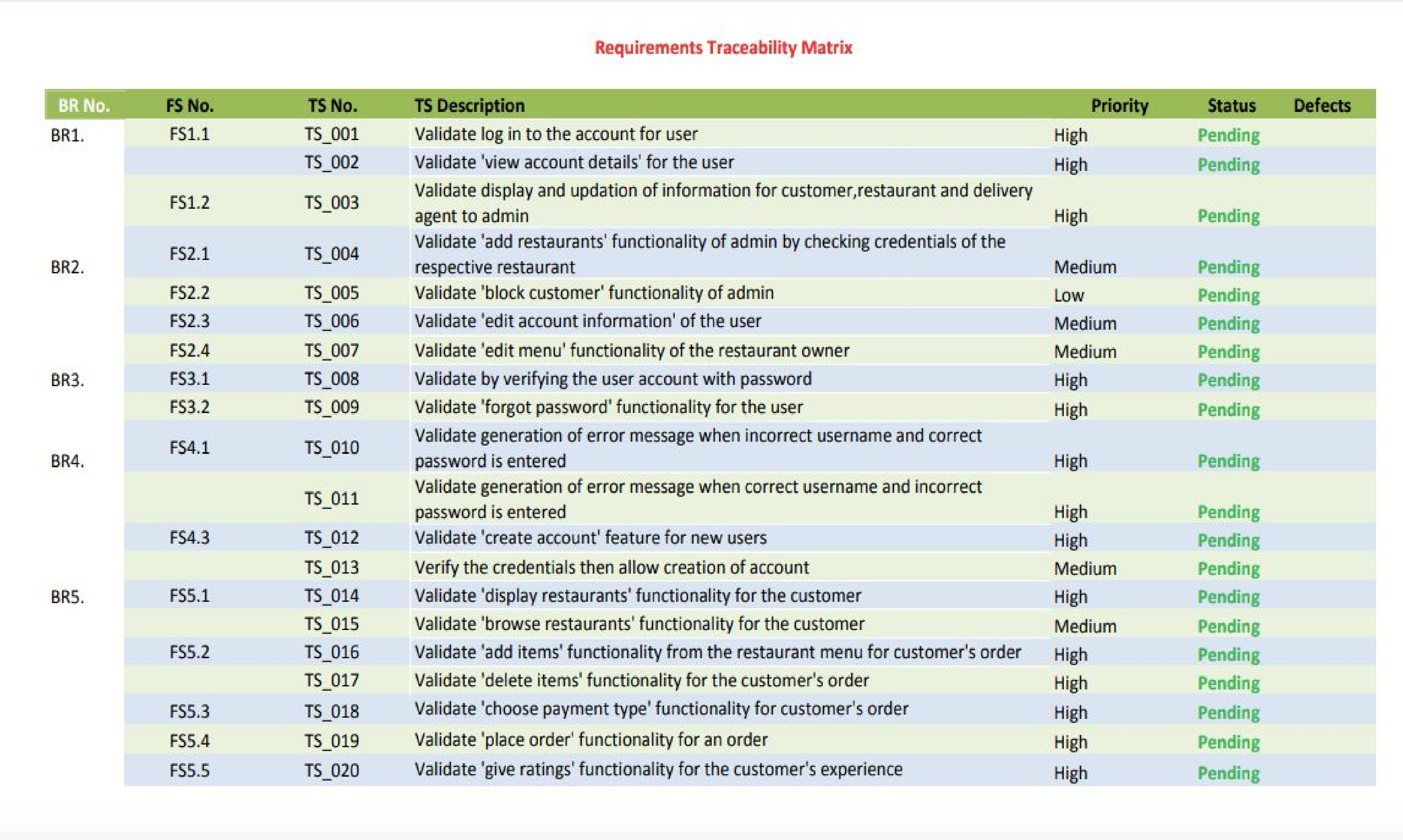
- Profile editing requires a password-protected user account with manually-assigned rights. (FS7.2)
- Change personal details only for the user logged-in. (FS7.3)
- Consideration will be given to potentially hiding the email address from public view to avoid collection by spam senders. (FS7.4)

2.5. Non-functional Specification Requirements

1. Usability: The main use of our app will be done by admin and customers willing to order food. The system will be made by considering the ease of all stakeholders. System will save time for customers by ensuring delivery in short-span of time.
2. Maintainability- The admin manages to add or delete a restaurant on the basis of the feedback received from the customer. In a similar manner, admin can report or block a particular user if necessary. The application features can further be improved from the feedback received.
3. Recoverability- The system is capable of recovering the data of a user or a restaurant in case the password is lost by the admin or customer.
4. Security- The system is secure in case of transactions. Transactions made online are facilitated via payment gateway which ensures the security of user data.
5. Data Integrity- User data is stored on the basis of a unique id of the user. Data of past orders is preserved in case of need in future. Users can lodge a complaint upon dissatisfaction from previous orders using that information.
6. Availability- The application will be available for anytime of the day. Users can order food as long as the restaurant is open for delivery.
7. Scalability- The application will be able to support multiple restaurants open for delivery at a time.It can be further improved to support multiple users (in hundreds or thousands) after successful testing.
8. Compatibility- The application is compatible on any browsers or operating systems.

9. Performance- The application is expected to be highly responsive. It does not take much time to process the request of the user. Also, the locations will be updated in real time.
10. Data Integrity- The data of the restaurant or customer will be updated in the database of the application when any change is made. Any monetary amount will be stored accurately.

2.6. Requirements Traceability Matrix



Requirements Traceability Matrix

BR No.	FS No.	TS No.	TS Description	Priority	Status	Defects
BR1.	FS1.1	TS_001	Validate log in to the account for user	High	Pending	
		TS_002	Validate 'view account details' for the user	High	Pending	
	FS1.2	TS_003	Validate display and updation of information for customer,restaurant and delivery agent to admin	High	Pending	
BR2.	FS2.1	TS_004	Validate 'add restaurants' functionality of admin by checking credentials of the respective restaurant	Medium	Pending	
	FS2.2	TS_005	Validate 'block customer' functionality of admin	Low	Pending	
	FS2.3	TS_006	Validate 'edit account information' of the user	Medium	Pending	
	FS2.4	TS_007	Validate 'edit menu' functionality of the restaurant owner	Medium	Pending	
BR3.	FS3.1	TS_008	Validate by verifying the user account with password	High	Pending	
	FS3.2	TS_009	Validate 'forgot password' functionality for the user	High	Pending	
BR4.	FS4.1	TS_010	Validate generation of error message when incorrect username and correct password is entered	High	Pending	
		TS_011	Validate generation of error message when correct username and incorrect password is entered	High	Pending	
	FS4.3	TS_012	Validate 'create account' feature for new users	High	Pending	
		TS_013	Verify the credentials then allow creation of account	Medium	Pending	
BR5.	FS5.1	TS_014	Validate 'display restaurants' functionality for the customer	High	Pending	
		TS_015	Validate 'browse restaurants' functionality for the customer	Medium	Pending	
	FS5.2	TS_016	Validate 'add items' functionality from the restaurant menu for customer's order	High	Pending	
		TS_017	Validate 'delete items' functionality for the customer's order	High	Pending	
	FS5.3	TS_018	Validate 'choose payment type' functionality for customer's order	High	Pending	
	FS5.4	TS_019	Validate 'place order' functionality for an order	High	Pending	
	FS5.5	TS_020	Validate 'give ratings' functionality for the customer's experience	High	Pending	