

**VEGAN FOOD ORDERING PLATFORM**

**Software Requirement Specification**

Ho Chi Minh City, May 12nd, 2021

Contents

[**I. OVERVIEW - 4 -**](#_Toc72012587)

[**1. Introduction - 4 -**](#_Toc72012588)

[**2. System functions - 5 -**](#_Toc72012589)

[**a. Screen flows - 5 -**](#_Toc72012590)

[**b. Screen details - 5 -**](#_Toc72012591)

[**c. User authorization - 7 -**](#_Toc72012592)

[**d. Non-screen functions - 7 -**](#_Toc72012593)

[**3. Common requirements - 7 -**](#_Toc72012594)

[**II. FUNCTION REQUIREMENTS - 8 -**](#_Toc72012595)

[**1. Security - 8 -**](#_Toc72012596)

[**a. Authentication - 8 -**](#_Toc72012597)

[**b. Authorization - 8 -**](#_Toc72012598)

[**2. Register - 8 -**](#_Toc72012599)

[**3. Profile management - 9 -**](#_Toc72012600)

[**a. Get user’s profile - 9 -**](#_Toc72012601)

[**b. Update use’s profile - 9 -**](#_Toc72012602)

[**4. Customer’s order management - 9 -**](#_Toc72012603)

[**a. Create a new order - 9 -**](#_Toc72012604)

[**b. Cancel an order - 10 -**](#_Toc72012605)

[**c. Get a list of orders - 10 -**](#_Toc72012606)

[**d. Get an order’s detail - 10 -**](#_Toc72012607)

[**5. User’s dishes management - 11 -**](#_Toc72012608)

[**a. Get a list of available dishes - 11 -**](#_Toc72012609)

[**b. Get detailed information of a dish - 11 -**](#_Toc72012610)

[**6. Home-cook’s dishes management - 11 -**](#_Toc72012611)

[**a. Create a new dish - 11 -**](#_Toc72012612)

[**b. Update a dish - 12 -**](#_Toc72012613)

[**c. Get a list of dished - 12 -**](#_Toc72012614)

[**7. Menu management - 12 -**](#_Toc72012615)

[**a. Get a list of menu - 12 -**](#_Toc72012616)

[**b. Create a menu - 13 -**](#_Toc72012617)

[**c. Update a menu - 13 -**](#_Toc72012618)

[**8. Home-cook’s order management - 13 -**](#_Toc72012619)

[**a. Get a list of new order - 13 -**](#_Toc72012620)

[**b. Get a list of completed order - 13 -**](#_Toc72012621)

[**c. Cancel an order - 14 -**](#_Toc72012622)

[**d. Get an order’s detail - 14 -**](#_Toc72012623)

[**9. Home-cook management - 14 -**](#_Toc72012624)

[**a. Create a home-cook account - 14 -**](#_Toc72012625)

[**b. Get a list of home-cook account - 15 -**](#_Toc72012626)

[**c. Deactivate/activate a home-cook account - 15 -**](#_Toc72012627)

# Overview

## Introduction

The Vegetarian Ordering System is a platform which connects vegans who want to have quality meals at home, and home-cooks, who want to spread the joy of cooking and spread the vegan spirit to others people. The system is expected to allow registered customers to place an order on the platform; the home-cook can add some dishes and create menus; The admin will manage the available home-cook account.



## System Functions

### Screen flows



### Screen details

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Login | Login | This is the screen where user will enter information to authenticate themselves. The system will also authorized them and redirect them to appropriate page |
| 2 | Register | Register | This is the screen where users will enter information to verify themselves. The system will then check the input data and grant the user access to the system |
| 3 | Get list of dishes by menu | Customer Landing Page | This is the screen where user can get a list of available menus and dishes. The system will query the database for active menu and display them on this screen |
| 4 | Get user’s profile | User Profile | This is the screen where user’s detailed information is displayed. The system will query for all data related to the user before showing it on the screen |
| 5 | Update user’s profile | Edit Profile | This is the screen where user can edit their personal information by entering corresponding information. The system will then check the input data and update their information in the database |
| 6 | Get list of home-cook | Home-cook management | This is the screen where a list of available home-cook is displayed. The system will query the database for all home-cook in the system and show it on the screen |
| 7 | Create a home-cook account | Create Home-cook | This is a pop-up screen where admin can enter information of a home-cook. The system will check the input data and create a new home-cook record in the database |
| 8 | Disable a home-cook account | Confirmation of disability | This is a pop-up box to confirm whether to disable the selected account or not. The system will then validate the user’s choice and decide to perform the action or not |
| 9 | Get list of orders | Order List | This is a screen where a list of orders belong to the current user would be displayed. The system will query the database for all orders that belong to a user before displaying it on the screen |
| 10 | Get detailed order | Order Details | This is a screen where detailed information of a specific order would be displayed. The system will query the database for information of the corresponding order before displaying it on the screen |
| 11 | Cancel an order | Confirmation of Cancelation | This is a pop-up confirmation box of canceling an order. The system will check the user’s choice before deciding whether to take the action or not |
| 11 | Get a dish details | Dish Details | This is a pop-up screen where detailed information of a dish would be displayed. The system will query the database for detailed information of the corresponding dish before showing it on the screen |
| 12 | Get a list of created dishes | Dish management | This is a screen where a list of created dishes by a specific home-cook would be displayed. The system will query the database for the dishes belongs to the corresponding home-cook before showing on the screen |
| 13 | Create a dish | Create Dishes | This is a pop-up screen where home-cook can enter information of a dish. The system will then validate the input data before creating a new record in the database |
| 14 | Edit a dish | Edit Dishes | This is a pop-up screen where home-cook can edit the information of a specific dish. The system will then validate the input data before updating the existed dish with new information |
| 15 | Place an order | Order Review, Delivery Information Confirmation | These are the screen where users can confirm the information of their order before entering the delivery information. The system will then validate the input data before creating a new order record in the database |
| 16 | Get a list of Menus | Menu Management | This is the screen where a list of menus created by the corresponding home-cook would be displayed. The system will query the database for all menus belong to the corresponding home-cook before showing it on the screen |
| 17 | Edit a Menu | Edit Menu | This is a page where home-cook can edit their menus. The system will validate the input data before updating the corresponding record in the database |
| 18 | Delete a Menu | Delete Menu | This is a pop-up confirmation box whether to delete the Menu or not. The system will verify the user’s choice before deciding to perform the action or not |
| 19 | Create a Menu | Create Menu | This is a page where Home-cook can enter information of a menu. The system will then validate the input data before creating a new record in the database |

### User authorization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Unauthenticated user** | **Customer** | **Admin** | **Home-cook** |
| Login | X |  |  |  |
| Register | X |  |  |  |
| Landing Page | X |  |  |  |
| Customer Dashboard |  | X |  |  |
| Admin Dashboard |  |  | X |  |
| Home-cook Dashboard |  |  |  | X |
| User Profile |  | X | X | X |
| Edit Profile |  | X | X | X |
| Home-cook management |  |  | X |  |
| Create a home-cook |  |  | X |  |
| Confirmation of disability |  |  | X |  |
| Order List |  | X |  | X |
| Order Details |  | X |  | X |
| Dish Details |  | X |  |  |
| Dish Management |  |  |  | X |
| Create Dish |  |  |  | X |
| Edit Dish |  |  |  | X |
| Order Review |  | X |  |  |
| Delivery Information Confirmation |  | X |  |  |
| Menu Management |  |  |  | X |
| Delete Menu |  |  |  | X |
| Edit Menu |  |  |  | X |
| Create Menu |  |  |  | X |

### Non-screen functions

## Common requirements

* Users must login before placing an order.
* The customers must pay before the delivery
* A menu must contain at least three dishes.
* User must place their order before 5:00 p.m everyday
* The order would be delivered by the next day
* If home-cooks want to cancel an order, they must cancel the order at least 12h before delivery time

# Function requirements

## Security

### Authentication

* Function trigger: The function is triggered when the user clicks on the “Login” button
* Function description: The function is used to verify that if one user can access the resources on the system or not.
* Function detail:
* Data validation:
* Email field must not be empty; the input data must be in email format.
* Password field must not be empty; the input data must have the length between 6-12 characters.
* Data process: After the user clicked on “Login” button, the program will query the database for the corresponding username and password. If both match that of the database, the user will be logged in successfully, otherwise, failed

### Authorization

* Function trigger: The function is triggered when customer authenticate themselves successfully.
* Function description: The function is used to check the user’s role before redirecting them to appropriate dashboard so that they can access appropriate resources on the system.
* Function details
* Data validation: None
* Data processing: After authenticated successfully, the system will query for the user’s role in the database. Then, the user will be redirected to corresponding dashboard page.

## Register

* Function trigger: The function is trigger when the user clicks on “Register” link on the Login screen.
* Function description: The function is used to create a new user record in the database, so that the user can access the resources in the system.
* Function details:
* Data validation:
* Email field must not be empty; the input data must be in email format.
* Password field must not be empty; the input data must have the length between 6-12 characters.
* Full name field must not be empty; the input data must contain no digit and special character.
* Address field must not be empty.
* The City/District field must not be empty; the user must choose one item from the dropdown list.
* DoB field must not be empty; the user must choose a specific day in the past.
* Phone number field must not be empty; the input data must be 10 digits long.
* Data processing: When the user clicks on the “Register” button, the system will then validate the input data. If the inputs are valid, a new user record will be created in the database, and the user will be redirected to Customer Dashboard page. Else, the user must correct the input before proceeding.

## Profile management

### Get user’s profile

* Function trigger: The function is triggered when user clicks on the “View Profile” button on the dashboard.
* Function description: The function is used to query the database for the information of the corresponding user before processing it and display it.
* Function details
* Data validation: None
* Data processing: The system will query for personal information of the corresponding user from the database and processing it.

### Update use’s profile

* Function trigger: The function is triggered when customer clicks on the “Edit” button on User’s Profile screen
* Function description: The function is used to edit user’s information, validating it before updating the existed corresponding record in the database.
* Function details
* Data validation:
* Email field must not be blanked. The input data must be in email format.
* Full name field must not be empty; the input data must contain no digit and special character.
* Address field must not be empty.
* The City/District field must not be empty; the user must choose one item from the dropdown list.
* DoB field must not be empty; the user must choose a specific day in the past.
* Phone number field must not be empty; the input data must be 10 digits long.
* Data processing: Then user clicks on the “Edit” button, the system will validate the input data. If the data validated successfully, the system will update the corresponding record in the database with the new data. Otherwise, the user must return and correct the input data.

## Customer’s order management

### Create a new order

* Function trigger: The function is triggered when customer clicks on the “Order” button on their dashboard screen.
* Function description: The function is used to create a new order record in the database.
* Function details
* Data validation:
* The cart must not be empty.
* Receiver name field must not be blanked. The input data must contain only upper and lower case characters. No digit or special character is allowed.
* Phone number field must not be blanked. The input data must be 10 digits long.
* Address field must not be blanked.
* Data processing: After the user clicks on the “Order” button, the system will validate the input information. If the input is valid, the system will create a new order record in the database. Otherwise, the user have to correct the input information before proceeding with the next step

### Cancel an order

* Function trigger: The function is triggered when customer click the “Cancel” button on the detailed order.
* Function description: The function is used to update the status of the corresponding order record from the database when the user confirms to perform the action.
* Function details
* Data validation:
* The user clicks on the “Yes” button on the pop-up window.
* Data processing: The system will verify the user choice. If the user chooses “Yes”, the system will then update the status of the corresponding order record in the database. Otherwise, the system would do nothing.

### Get a list of orders

* Function trigger: The function is triggered when customer clicks on the “Order management” button on the dashboard page.
* Function description: The function is used to query for a list of order created by the corresponding customer.
* Function details
* Data validation: None
* Data processing: The system will query the database for the order that belongs to corresponding user

### Get an order’s detail

* Function trigger: The function is triggered when customer clicks on the “Detail” button on the order.
* Function description: The function is used to query for detail information of a specific order which belongs to the corresponding user from the database.
* Function details
* Data validation: None
* Data processing: The system will query for all information related to the specific order from the database before processing it and displaying it to the screen

## User’s dishes management

### Get a list of available dishes

* Function trigger: The function is triggered when customer logged in and redirected to their dashboard successfully
* Function description: The function is used to query the database and get all of the available dishes in the day and processing it before displaying.
* Function details
* Data validation: None
* Data processing: The system will query the database for all of the available dish and display it to the screen

### Get detailed information of a dish

* Function trigger: The function is triggered when customer clicks on a dish in the list displayed on the screen
* Function description: The function is used to query the detail information of the dish before displaying it to the screen
* Function details
* Data validation: None
* Data processing: The system will query the database for all the information related to the dish before displaying it.

## Home-cook’s dishes management

### Create a new dish

* Function trigger: The function is triggered when home-cook clicks on the “Create Dish” button on the Create Dish screen.
* Function description: The function is used to create a new dish record in the database.
* Function details:
* Data validation:
* The field “Price” must not be blanked. The input data must be number greater than 0. No character or special character is allowed.
* The field “Name” must not be blanked. The input data must contain character and/or digit. No special character is allowed.
* The field “Image” must not be empty. The user must upload an image of the dish.
* The field “Description” must not be blanked.
* Data processing: The system will validate the input data before creating a new record in the database. If the input values are successfully validated, the system will create a new record of the dish in the database. Otherwise, the system will throw an error and the user must correct the data before proceeding.

### Update a dish

* Function trigger: The function is triggered when user clicks on the “Edit dish” button on the screen.
* Function description: The function is used to validate the input information before updating the existed information of a dish in the database.
* Function details:
* Data validation:
* The field “Price” must not be blanked. The input data must be number greater than 0. No character or special character is allowed.
* The field “Name” must not be blanked. The input data must contain character and/or digit. No special character is allowed.
* The field “Image” must not be empty. The user must upload an image of the dish.
* The field “Description” must not be blanked.
* Data processing: The system will validate the input data before perform the operation. If success, the system will update the information of the corresponding dish in the database. If failed, the system will throw an error and the user must correct the information before proceeding.

### Get a list of dished

* Function trigger: The function is triggered when user click on the “Dish management” on the dashboard screen
* Function description: The function is used to query for all of the dishes which belongs to the corresponding user in the database.
* Function details:
* Data validation: None
* Data processing: The system will query and process the data of all of the dishes which belongs to the corresponding user

## Menu management

### Get a list of menu

* Function trigger: The function is triggered when user clicks on the “Menu management” button on the home-cook dashboard.
* Function description: The function is used to query the database for a list of created dishes by the corresponding home-cook.
* Function details:
* Data validation: None
* Data processing: The system will query the database for a list of created dishes by the corresponding home-cook.

### Create a menu

* Function trigger: The function is triggered when user clicks on the “Create dish” button on the dashboard.
* Function description: The function is used to create a new dish record in the database
* Function details:
* Data validation:
* The “Name” field must not be blanked.
* The “Dishes” field must have at least 3 dishes
* Data processing: The system will validate the input data. If the information is validated successfully, the system will create a new record in the database. Otherwise, the system will throw an error and the user must correct the data before proceeding.

### Update a menu

* Function trigger: The function is triggered when customer clicks on the “Edit” button on the menu in the dashboard.
* Function description: The function is used to update the record of the corresponding menu in the database.
* Function details:
* Data validation:
* The “Name” field must not be blanked.
* The “Dishes” field must have at least 3 dishes
* Data processing: The system will validate the input data. If the validation is successful, the system will update the information of the corresponding menu in the database. If fails, the system will throw an error and the user must correct the input data before proceeding

## Home-cook’s order management

### Get a list of new order

* Function trigger: The function is triggered when home-cook clicks on the “Upcoming orders” button on the dashboard screen.
* Function description: The function is used to query the database for a list of incoming order which the home-cook should complete.
* Function details:
* Data validation: None
* Data processing: The system will access the database and get all of the information about the orders which the home-cook should complete.

### Get a list of completed order

* Function trigger: The function is triggered when home-cook clicks on the “Completed orders” button on the dashboard screen.
* Function description: The function is used to query the database for a list of completed order which belongs to the corresponding home-cook.
* Function details:
* Data validation: None
* Data processing: The system will access the database and get all of the information about the orders which the home-cook has finished.

### Cancel an order

* Function trigger: The function is triggered when home-cook click the “Cancel” button on the detailed order.
* Function description: The function is used to update the status of the corresponding order record from the database when the home-cook confirms to perform the action.
* Function details
* Data validation:
* The user clicks on the “Yes” button on the pop-up window.
* Data processing: The system will verify the user choice. If the user chooses “Yes”, the system will then update the status of the corresponding order record in the database. Otherwise, the system would do nothing.

### Get an order’s detail

* Function trigger: The function is triggered when customer clicks on the “Detail” button on the order.
* Function description: The function is used to query for detail information of a specific order which belongs to the corresponding home-cook from the database.
* Function details
* Data validation: None
* Data processing: The system will query for all information related to the specific order from the database before processing it and displaying it to the screen

## Home-cook management

### Create a home-cook account

* Function trigger: The function is trigger when the user clicks on “Create home-cook account” link on the dashboard.
* Function description: The function is used to create a new home-cook record in the database, so that home-cook can access the resources in the system.
* Function details:
* Data validation:
* Email field must not be empty; the input data must be in email format.
* Password field must not be empty; the input data must have the length between 6-12 characters.
* Full name field must not be empty; the input data must contain no digit and special character.
* Address field must not be empty.
* The City/District field must not be empty; the user must choose one item from the dropdown list.
* DoB field must not be empty; the user must choose a specific day in the past.
* Phone number field must not be empty; the input data must be 10 digits long.

Data processing: When the user clicks on the “Create” button, the system will then validate the input data. If the inputs are valid, a new user record will be created in the database. Else, the admin must correct the input before proceeding.

### Get a list of home-cook account

* Function trigger: The function is triggered when user clicks on the “Home-cook Management” button on the home-cook dashboard.
* Function description: The function is used to query the database for a list of created home-cook in the system.
* Function details:
* Data validation: None

Data processing: The system will query the database for a list of created home-cook in the system.

### Deactivate/Activate a home-cook account

* Function trigger: The function is triggered when admin clicks on the “Disable” button on the screen
* Function description: The function is used to change the state of a home-cook account from active to inactive, and update the status in the database
* Function details
* Data validation: None
* Data processing: The system will update the status of the corresponding home-cook account in the database.

## 