

Requirement & Design Specification

**Global Assess Management System (GAMS)**

**Version: 1.0**

– Hanoi, August 2022 –

# Record of Changes

| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- | --- |
| V1.0 | 14/05/2024 | A | Nguyễn Chu Tứ | 1. Overview   1.User Requirements  1.1 Actor  1.2.b. Description |
| V1.0 | 18/05/2024 | A | Mai Ngọc Đạt | 1.2. a Use Case |
| V1.0 | 19/05/2024 | M | Mai Chí Thanh | 1.2.a Use Case |
| V1.0 | 20/05/2024 | A | Đào Minh Quân | 3. System High Level Design  3.1. Database Design |
| V1.0 | 22/05/2024 | A | Nguyễn Chu Tứ | 2.Overall Functionalities  2.2 Screen Descriptions |
| V1.0 | 23/05/2024 | A | Nguyễn Chu Tứ | 2.3 Screen Authorization |
| V1.0 | 24/05/2024 | A | Nguyễn Chu Tứ | II. Requirement Specfications  1.1 UC-1\_Register  1.1.a UC Specifications  1.1.b Business Rules |
| V1.0 | 25/05/2024 | A | Mai Chí Thanh | 2.Common Function  2.1. Login System   1. UC Specification |
| V1.0 | 26/05/2024 | A | Đào Minh Quân | 3. Patron Feature  3.1 Ordel a meal   1. UC Specifications 2. Business role |
| V1.0 | 26/05/2024 | A | Mai Ngọc Đạt | 3.2 Register for Payroll Deduction |
| V1.0 | 26/05/2024 | A | Nguyễn Chu Tứ | III. Screen Designs1.2 System Accessa. User Login |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[Record of Changes 2](#_gjdgxs)

[I. Overview 5](#_30j0zll)

[1. User Requirements 5](#_1fob9te)

[1.1 Actors 5](#_3znysh7)

[1.2 Use Cases 5](#_tyjcwt)

[2. Overall Functionalities 6](#_3dy6vkm)

[2.1 Screens Flow 6](#_1t3h5sf)

[2.2 Screen Descriptions 6](#_4d34og8)

[2.3 Screen Authorization 6](#_2s8eyo1)

[2.4 Non-UI Functions 7](#_17dp8vu)

[3. System High Level Design 7](#_3rdcrjn)

[3.1 Database Design 7](#_26in1rg)

[3.2 Code Packages 7](#_1ksv4uv)

[II. Requirement Specifications 9](#_44sinio)

[1. <<Feature Name>> 9](#_2jxsxqh)

[1.1 << UC Name>> 9](#_z337ya)

[2. Common Functions 11](#_3j2qqm3)

[2.1 Login System 11](#_1y810tw)

[3. Patron Feature 12](#_4i7ojhp)

[3.1 Order a Meal 12](#_2xcytpi)

[3.2 Register for Payroll Deduction 13](#_1ci93xb)

[III. Screen Designs 14](#_3whwml4)

[1. <<Feature Name>> 14](#_2bn6wsx)

[1.1 <<SubFeature Name>> 14](#_qsh70q)

[1.2 System Access 14](#_3as4poj)

[IV. Code Designs 17](#_1pxezwc)

[1. <Feature/Function Name1> 17](#_49x2ik5)

[1.1 Class Diagram 17](#_2p2csry)

[1.2 Sequence Diagram(s) 17](#_147n2zr)

[1.3 Database Queries 18](#_3o7alnk)

[2. <Feature/Function Name2> 18](#_23ckvvd)

[V. Appendix 18](#_ihv636)

[1. Assumptions & Dependencies 18](#_32hioqz)

[2. Limitations & Exclusions 18](#_1hmsyys)

[3. Business Rules 18](#_41mghml)

[4. .. 19](#_2grqrue)

# I. Overview

## 1. User Requirements

### 1.1 Actors

*[An actor is a person (or sometimes another software system or a hardware device) that interacts with the system to perform a use case. Following are some questions you might ask to help user representatives identify actors*

* *Who (or what) is notified when something occurs within the system?*
* *Who (or what) provides information or services to the system?*
* *Who (or what) helps the system respond to and complete a task?*

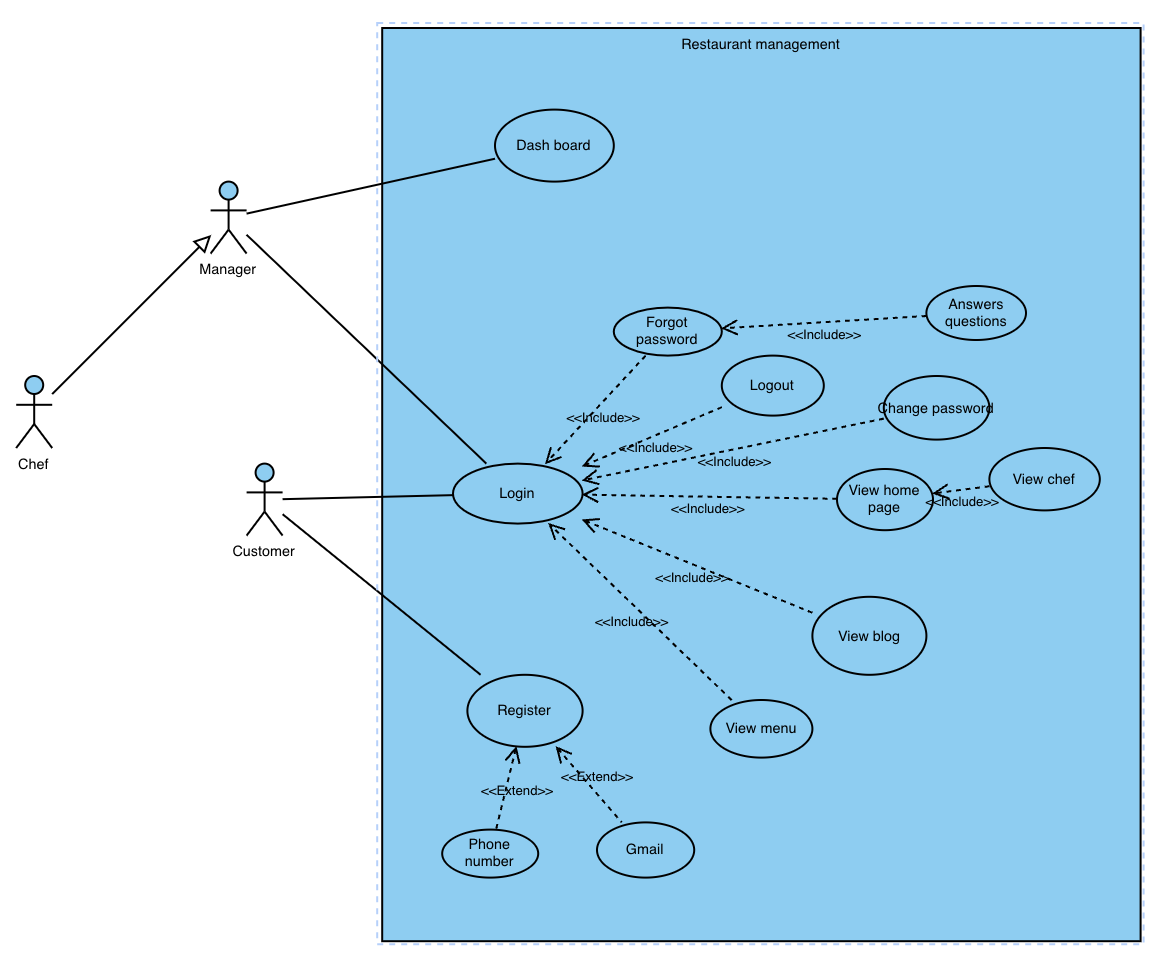
*This part gives the description of system actors, you can follow the table form as below]*

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Manager | The highest manager in the WMS system has the right to access and perform all system functions. |
| 2 | User | Customers can view profiles, book tables, view menus,... |

### 1.2 Use Cases

*[A use case (UC) describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value. The names of use cases are always written in the form of a verb followed by an object. Select strong, descriptive names to make it evident from the name that the use case will deliver something valuable for some user]*

#### a. Diagram(s)

*[Provide the UC diagram(s) to show the actor-UCs and UC-UC relationships like the sample below. You can have multiple UC diagrams for the system*

#### b. Descriptions

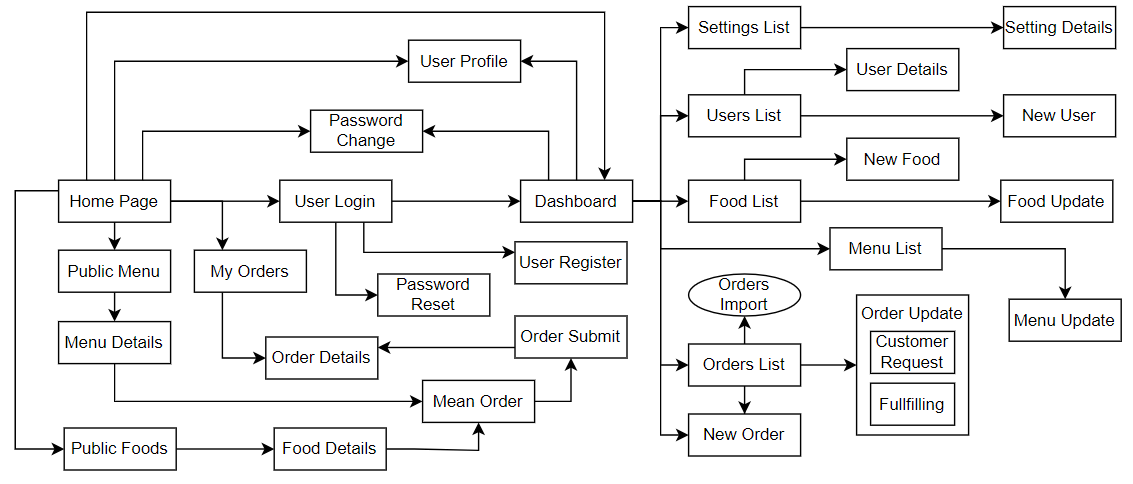
*This part describes the use cases, you can follow the table form as below]*

| **ID** | **Feature** | **Use Case** | **Use Case Description** |
| --- | --- | --- | --- |
| 01 | Authentication | User Login | Allows users to log into the system. |
| 02 | Authentication | User Forgot password | Allows users to retrieve their password if they forget it. |
| 03 | Authentication | User logout | Allows users to log out of the system. |
| 04 | Common | View profile user | Users can view their profile by clicking the personal icon. |
| 05 | Common | edit user profile | Users can modify their personal information. |
| 06 | Common | change password | Users can change their password. |
| 07 | Manager Features | View list user | Administrators can view a list of all users. |
| 08 | Manager Features | New user | The administrator wants to create a new user on the system. |
| 09 | Manager Features | User detail | Admin reviews user information, reads and knows user information |
| 10 | Manager Features | update user | Admin can edit user description. |
| 11 | Manager Features | Block user | Admin can block and unblock user in system. |
| 12 |  |  |  |
| 13 |  |  |  |

## 2. Overall Functionalities

### 2.1 Screens Flow

*[This part shows the system screens and the relationship among screens. You can draw the Screens Flow for the system in the form of diagram as below. Please note that beside the normal flat screen, we might have the oval notation for pop-up screen (Orders Import) or a screen with multiple information tab (Order Update), etc. You may also use text or background format for different visuality purpose]*



### 2.2 Screen Descriptions

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Home | Home page | Introduce the website and highlight its features, display featured dishes and special offers |
| 2 | Home | Admin home | Home page for management. You can filter dishes and drinks by certain categories and manage user accounts |
| 3 | Login | Login | There are 2 fields username and password. There is a link to register a new user, forgot password |
| 4 | Login | User register | Registration screen to register a new user account. The following fields are available: Username, Password, Confirm Password, Full Name, Mobile Phone, Email, City, Address. |
| 5 | Login | Forgot password | Forgot Password for users who have forgotten their password. There is a field to enter email. After entering and validating email, Code will be sent to user email and after entering code, User can enter new password and confirm password. |
| 6 | Profile | User Profile | Update user profile. The updated fields will be Full Name, Mobile, City, Address |
| 7 | Customer Features | View Menu | Customers can view all the dishes on the restaurant's menu |
| 8 | Customer Features | View Home page | Người dùng có thể xem thông tin chung về khách hàng |
| 9 |  |  |  |

### 2.3 Screen Authorization

| **Screen** | **User** | **Manager** |  | **…** |
| --- | --- | --- | --- | --- |
| Home page | X | X |  |  |
| Admin home |  | X |  |  |
| User login | X | X |  |  |
| Forgot password | X | X |  |  |
| User profile | X | X |  |  |
| Dish list page | X | X |  |  |
| Dish detail page | X | X |  |  |
| set table list | X | X |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

### 2.4 Non-UI Functions

*[Provide the descriptions for the functions which have no UI (or not screens), i.e batch/cron job, service, API, etc.]*

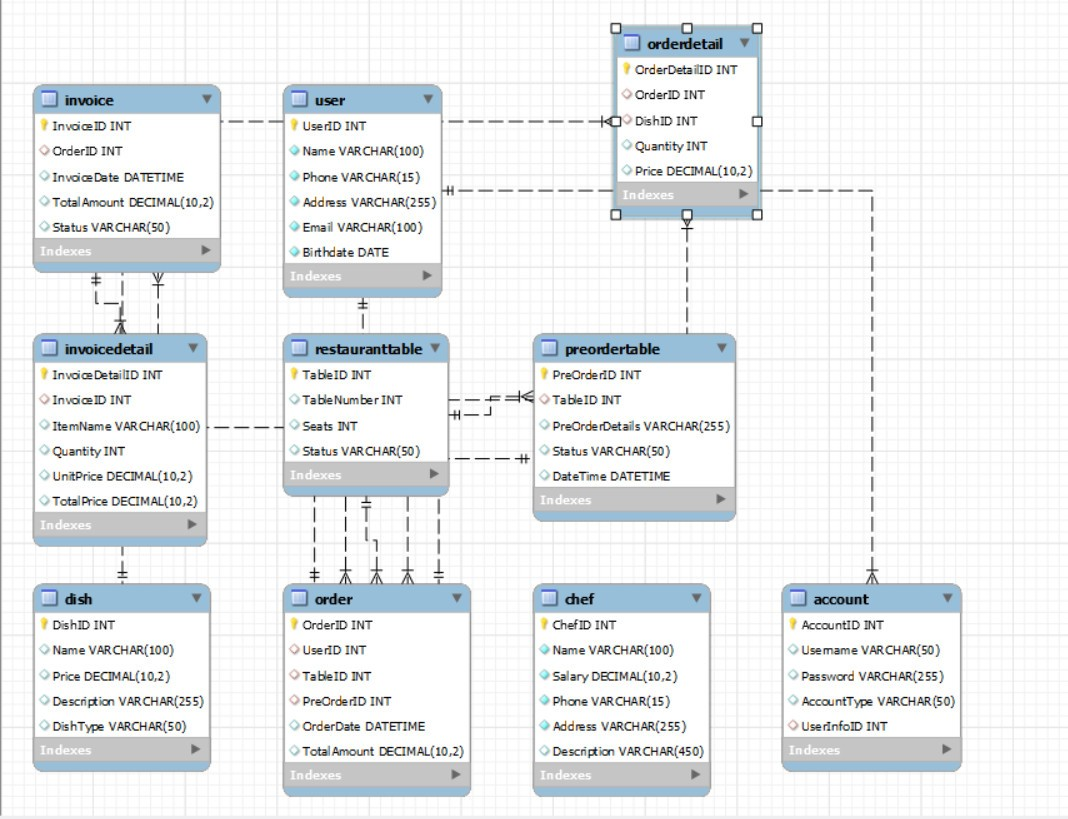
| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | <<Feature Name>> | <<Function Name1>> | <<Function Name1 Description>> |
| 2 | … |  |  |

## 3. System High Level Design

### 3.1 Database Design

*[Provide the tables relationship like example below]*

#### a. Database Schema



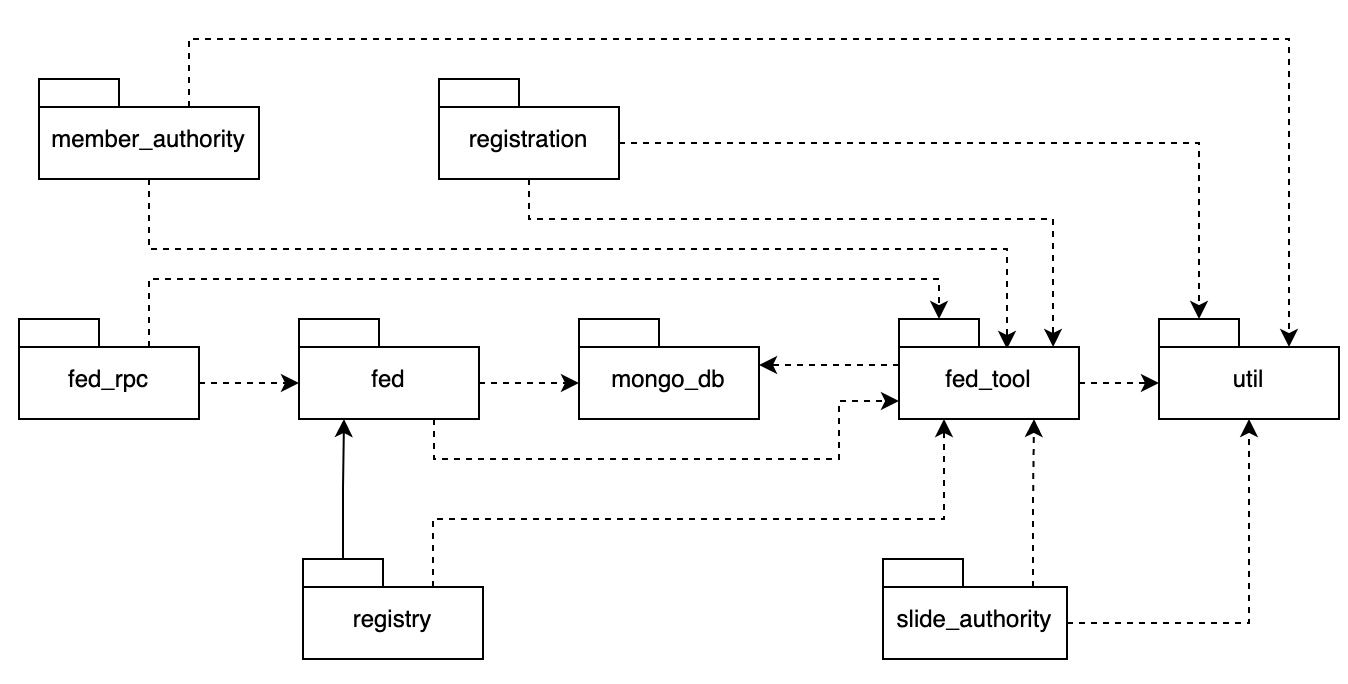
#### b. Table Descriptions

#### 

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *Account* | *Stores information about users.*  *- Primary keys: UserID*  *- Foreign keys: None* |
| *02* | *User* | *Stores account details and references the User table via UserInfoID.*  *Primary keys: AccountID*  *Foreign keys: UserInfoID (references User(UserID))* |
| *03* | *RestaurantTable* | *Stores information about the restaurant's tables.*  *Primary keys: TableID*  *Foreign keys: None* |
| *04* | *Chef* | *Stores information about chefs.*  *Primary keys: ChefID*  *Foreign keys: None* |
| *05* | *Dish* | *Stores information about dishes available in the restaurant.*  *Primary keys: DishID*  *Foreign keys: None* |
| *06* | *PreOrderTable* | *Manages pre-orders and references the RestaurantTable table via TableID.*  *Primary keys: PreOrderID*  *Foreign keys: TableID (references RestaurantTable(TableID))* |
| *07* | *Order* | *Manages orders, referencing the User, Chef, RestaurantTable, and PreOrderTable tables.*  *Primary keys: OrderID*  *Foreign keys: UserID (references User(UserID))*  *ChefID (references Chef(ChefID))*  *TableID (references RestaurantTable(TableID))*  *PreOrderID (references PreOrderTable(PreOrderID))* |
| *08* | *OrderDetail* | *Stores details of each order, including dishes ordered and their quantities.*  *Primary keys: OrderDetailID*  *Foreign keys:*  *OrderID (references Order(OrderID))*  *DishID (references Dish(DishID))* |
| *09* | *Invoice* | *Stores invoice details, referencing the Order table.*  *Primary keys: InvoiceID*  *Foreign keys:*  *OrderID (references Order(OrderID))* |
| *10* | *InvoiceDetail* | *Stores detailed information about each item in an invoice, referencing the Invoice table.*  *Primary keys: InvoiceDetailID*  *Foreign keys:*  *InvoiceID (references Invoice(InvoiceID))* |

### 3.2 Code Packages

*[Provide the package diagram for each sub-system. The content of this section including the overall package diagram, the explanation, package and class naming conventions in each package. Please see the sample & description table format below]*



***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *member\_authority* | *<Description of the package: purpose, contents,..>* |
| *02* | *registration* | *<Description of the package: purpose, contents,..>* |
| *03* | *…* |  |

# II. Requirement Specifications

## 1.Register

### 1.1 UC-1\_Register

#### a. UC Specifications

Provide the functional description for the use cases using the template/guides below

**Functional Description Template**

| UC Name: | UC1\_Register | | |
| --- | --- | --- | --- |
| Primary Actor: | User | Secondary Actors: | None |
| Created By: | TuNC | Date | 20/5/2024 |
| Trigger: | When user want to have a account to login website | | |
| Description: | As user, I want create an account to login website so that i can use some service in website | | |
| Preconditions: | Just click into register page | | |
| Postconditions: | * Success: return to LoginPage * Failure: The system reports an error corresponding to the error | | |
| Normal Flow: | **1.0 Register**  1. User accesses the User Login screen  2. User switch to Register or click don't have account  3.User enters some basic information  4. User clicks the Register button  5. System validates check the email and send a link to your email  6. Go email message and click the link to active account  7. System transfer to login page with note “Your account has been verified. You can log in now!” | | |
| Alternative Flows: |  | | |
| Exceptions: | ***1.0.E1 System not accept user information***  1. The Error Message screen is shown to the user | | |
| Business Rules: | BR1, BR2 | | |
| Priority: | High , Must Have | | |
| Frequency of Use: | Always | | |

#### b. Business Rules

Provide the business rules those are applied only to the use case

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| BR1 | Password | Password must have characters, including capital letters, numbers, special characters, and no spaces. |
| BR2 | Email unique | Email must not exist in database |

## 2. Common Functions

### 2.1 Login System

#### a. UC Specification

| UC Name: | **UC-2\_Login System** | | |
| --- | --- | --- | --- |
| Primary Actor: | Customer | Secondary Actors: | None |
| Created By: | ThanhMC | Date | 20/5/2024 |
| Trigger: | The Customer indicates they want to log into the system. | | |
| Description: | *The Customer accesses the Login System from the website or mobile application, enters credentials, and logs into the system. The Manager can monitor and manage customer login sessions.* | | |
| Preconditions: | User account has been created & authorized | | |
| Postconditions: | * The Customer has a registered account. * The system is operational and ready to accept login requests. | | |
| Normal Flow | **Logging into the system**  1. The Customer accesses the login page.  2. The system displays the login interface.  3. The Customer enters their username and password (see exceptions E1 and E2).  4. The Customer clicks the "Login" button.  5. The system verifies the customer's credentials (see exception E3).  6. If the credentials are valid, the system logs the customer in and redirects them to the main page.  7. The system stores the login session information  8. The system sends a successful login notification email to the customer  9. The Manager can access and view the login session information. | | |
| Alternative Flows: | ***AF1. Forgot Password***  1.The Customer clicks on the "Forgot Password" link  2.The system prompts the customer to enter their registered email address  3.The system sends an email with password recovery instructions to the customer  4. Google validates user’s login information successfully and redirect him/her back to the system  5. Return to step 5 of normal flow.  ***AF2. Facebook Login***  1. User chooses to login system using Facebook account  2. System redirects the user to the Facebook’s Login screen  3. User types in the Facebook account details and chooses to login  4.The Customer follows the instructions in the email to reset their password.  5. Return to step 1 of the normal flow. | | |
| Exceptions: | ***E1. Invalid Username or Password***  1. The system displays a message "Your account has been locked. Please contact the manager for more details”.  2. The Customer contacts the manager to unlock their account.  **E2. Account Locked**  1. The system displays a message "Your account has been locked. Please contact the manager for more details."  2.The Customer contacts the manager to unlock their account.  **E3. System Error**  1. The system displays an error message "The system is currently experiencing issues. Please try again later.  2.The Customer attempts to log in again later. | | |
| Business Rules: | 1.All login credentials must be encrypted before being sent over the network  2.The Manager has the authority to lock or unlock customer accounts | | |

## 3. Patron Feature

### 3.1 Order a Meal

#### a. UC Specification

| UC Name: | **UC-3\_Reserve a Table** | | |
| --- | --- | --- | --- |
| Primary Actor: | Customer | Secondary Actors: | Restaurant Reservation System, Restaurant Staff |
| Description: | A Customer accesses the Restaurant Reservation System from a website or mobile app, views available tables for a specific date and time, and reserves a table for dining. | | |
| Trigger: | A Customer indicates that they want to reserve a table. | | |
| Preconditions: | Customer is logged into the Reservation System (if required).  Restaurant has available tables for the specified date and time. | | |
| Postconditions: | Reservation is stored in the Reservation System with a status of “Confirmed”.  Restaurant staff is notified of the new reservation. | | |
| Normal Flow: | 1. **Reserve a Table** 2. Customer requests to view available tables for a specific date and time (see exceptions E1 and E2). 3. Reservation System displays available tables and time slots. 4. Customer selects a table and time slot. 5. Customer provides reservation details (e.g., number of guests, special requests). 6. Customer confirms the reservation. 7. Reservation System displays the reservation summary. 8. Customer confirms the reservation summary. 9. Reservation System confirms the reservation. 10. Reservation System notifies Restaurant Staff of the new reservation. | | |
| Alternative Flows: | 1. AF1. Modify Reservation 3. Customer requests to modify an existing reservation. 4. Return to step 1 of the normal flow. | | |
| Exceptions: | E1. No available tables for the selected date and time  -Reservation System informs the Customer that no tables are available for the specified date and time.  -If Customer cancels the reservation process, then the Reservation System terminates the use case.  -Else if Customer requests another date and time, then the Reservation -System restarts the use case.  E2. Selected table and time slot become unavailable during the booking process  -Reservation System informs the Customer that the selected table and time -slot are no longer available.  -Customer selects a different table and time slot.  -Return to step 2 of the normal flow. | | |
| Business Rules: | **BR-1, BR-2, BR-3, BR-4, BR-5, BR-6** | | |

#### b. Business Rules

None

### 3.2 Register for Payroll Deduction

a.*a. UC Specification*

| UC Name: | **UC-4\_ Display Menu Items** | | |
| --- | --- | --- | --- |
| Primary Actor: | Patron | Secondary Actors: | Restaurant Inventory System |
| Description: | The customer wants to view the available menu items on the restaurant's ordering system. | | |
| Trigger: | The customer requests to view the menu from the ordering system interface. | | |
| Preconditions: | 1.The customer has accessed the restaurant's ordering system.  2.The menu items are available from the restaurant's inventory system. | | |
| Postconditions: | 1.The menu items are displayed to the customer.  2.Detailed information about each menu item, including prices and daily specials, is shown. | | |
| Normal Flow: | **1.Customers request to see the menu:**  - The customer selects a date from the calendar or chooses the current date.  **2.The restaurant ordering system displays the available menu items and daily specials:**  - The system retrieves menu information from the restaurant's system.  - The menu is displayed in a list or grid format with the following details:  - Name of the dish  - Image of the dish  - Short description  - Price of each item  - Daily specials (if any)  **3.The customer selects a dish to view detailed information (if desired):**  - The customer clicks on the name or image of the dish.  - The system displays detailed information about the dish, including ingredients, price, and special notes (e.g., vegetarian, gluten-free). | | |
| Alternative Flows: | **AF1: The customer changes the date to view the menu:**  **\*The customer selects a different date from the calendar:**  - The system updates and displays the menu for the newly selected date.  - Return to step 2 of the normal flow. | | |
| Exceptions: | **E1: No menu is available for the selected date:**  **1. The system informs the customer that no menu is available for the selected date.**  **2. The customer chooses a different date or cancels the menu viewing process.** | | |
| Business Rules: | **OG-1:** Only display items that are in stock.  **OG-2:** Highlight daily specials at the top of the menu. | | |

#### b. Business Rules

None

# III. Screen Designs

## 1. <<Feature Name>>

### 1.1 <<SubFeature Name>>

#### a. <<Screen/Function Name>>

*[Provide brief description of the screen/function here and other details as in the sub-sections]*

*[This is to describe the UI layout (Mockup prototype) & descriptions for screen fields/components]*

<<Mockup prototype>>

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| ***Field Group Name*** | | |
| <<Field-Name>> | <<Type, Length>> | <<Field description & data initializing design>> |

##### Database Access

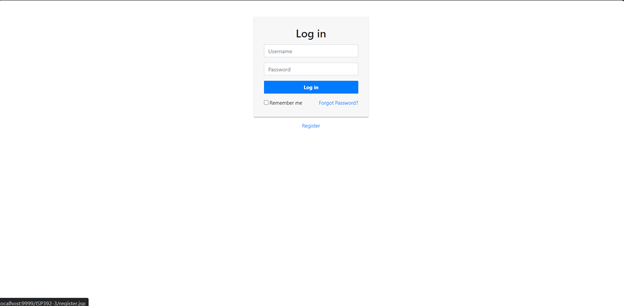
*[Provide the design description for the screen/function to access the database here: what table the screen/function would access, which transactions does it make (C-Create, R-Read, U-Update, or D-Delete), and how/purpose of the access (by providing Description and SQL commands)]*

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| <<Table Name>> | <<transaction(s)>> | <<Table access description: purpose, how,…>> |
| .. |  |  |

### 1.2 System Access

#### a. User Login

This screen allows user to be authenticated to the system screens/functionalities.



##### 

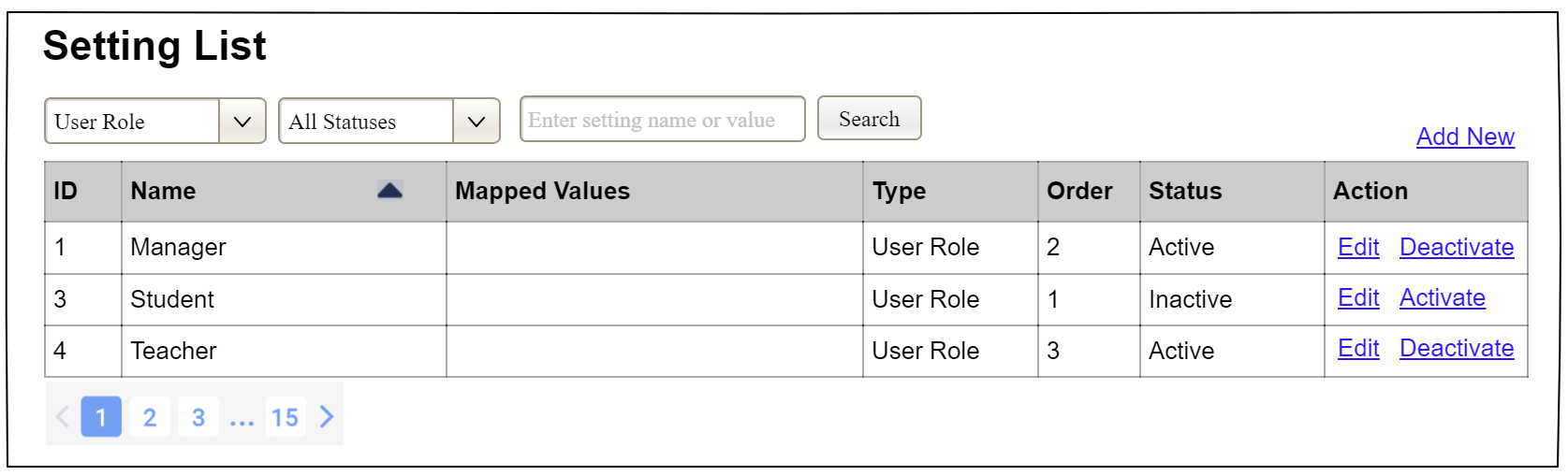
| Field Name | Field Type | Description |
| --- | --- | --- |
| User name | Text Box | This is for user to input valid user name for login |
| Password | Password Box | This is for user to input password for login |
| Forgot Password? | Button | User clicks to redirect to the User Register page for registering new user account to access the system |
| Register | Hyperlink | User clicks to redirect to the Password Reset page for resetting his/her forgot password |

##### 

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | Enter UserName & Password information |
| Setting, User | R | Specify the authorizations of the user |

#### b. Setting List

##### This is for the administrator to view the list of current system settings. On the screen, s/he can also activate or deactivate (change status) of a specific setting.



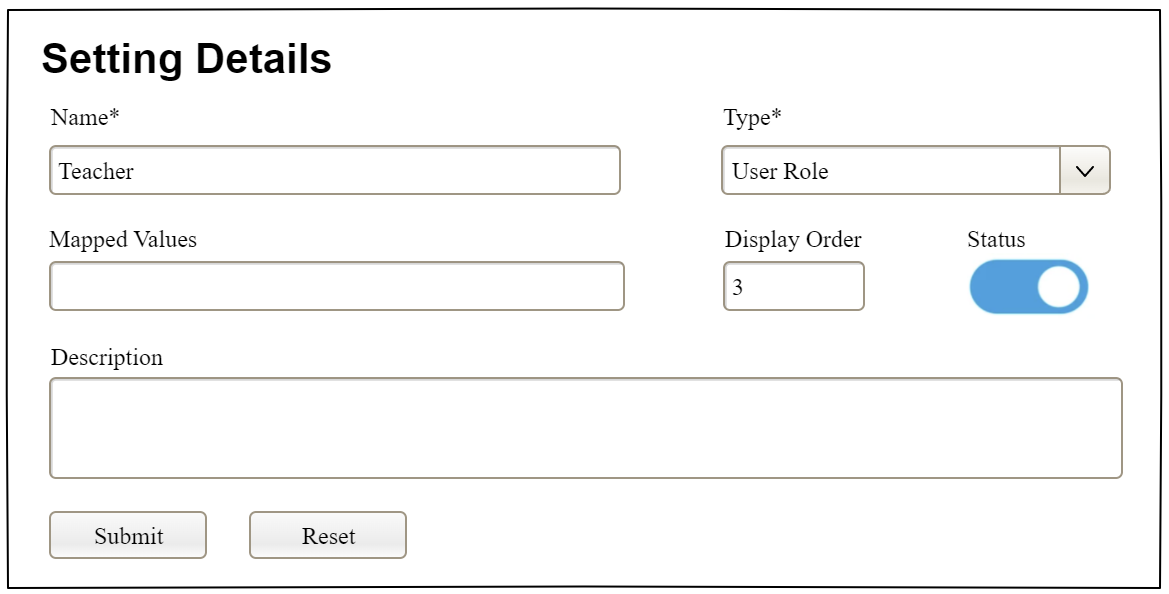
| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| ***Filter/Search Fields*** | | |
| Setting Type |  | Filled with the list of current active setting types  Allow to filter the list by setting type;  Default value is “All Types” |
| Setting Status |  | Values: All Statuses (default), Active, and Inactive  Allow to filter the list by status  Default value: “All Statuses” |
| Search Phase | String (30) | Allow to search using the name or map values  Default value: blank |
| Search |  | Click to refresh the list with the defined filter(s) and search phrase. |
| Add New |  | Click to open the Setting Details page for adding new setting (master data) |
| ***Data Table*** | | |
| ID | Integer | Auto-increased identifier of the setting |
| Name | Text | Name of the setting |
| Mapped Values | Text | Supplementary information for the setting |
| Type | Text | Type of the setting |
| Order | Integer | Display order of the setting: the order of the setting type, displayed among the list of settings with the same type |
| ***Action Links*** | | |
| Edit | icon | Click to open the Setting Details page for updating the relevant setting (master data) |
| Activate | icon | Shown when the data status is inactive. This is to activate the relevant setting (master data) |
| Deactivate | icon | Shown when the data status is active. This is to deactivate the relevant setting (master data) |

##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Setting | RU | Query the list of current settings from the database  Update status of a specific setting |

#### c. Setting Details

##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Name\* | Text Box  String (20) | Name of the setting |
| Type\* | Combo Box  (Single Choice) | Type of the setting, filled with the list of setting types  Default value: the first type in the list |
| Mapped Values | Text Box  String (50) | Supplementary information for the setting (if any) |
| Order | Text Box  Integer (>=0) | Display order of the setting: the order of the setting type, displayed among the list of settings with the same type |
| Status | On/Off button | Status of the setting: Active or Inactive  Default value: Active |
| Description | Text Area  String (200) | Description of the setting |
| Submit | Button | Click to store new or updated setting details |
| Reset | Button | Click to reset the changes use has made on the screen fields back to the initial values when the screen is loaded |

##### Database Access

…

# IV. Code Designs

## 1. <Feature/Function Name1>

*[Provide the detailed design for the function <Feature/Function Name1>. It includes class diagram and sequence diagram(s)]*

### 1.1 Class Diagram

*[This part presents the class diagram for the relevant feature]*



### 1.2 Sequence Diagram(s)

*[Provide the sequence diagram(s) for the feature/function, see the sample below. Sequence diagrams with similar message flow can be omitted, but you need to mention the appropriate reference to the similar sequence diagram]*



### 1.3 Database Queries

*[Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]*

## 2. <Feature/Function Name2>

…

# V. Appendix

## 1. Assumptions & Dependencies

*[Record any assumptions that were made when conceiving the project and writing this vision and scope document. Note any major dependencies the project must rely upon for success, such as specific technologies, third-party vendors, development partners, or other business relationships.]*

<<Sample:

AS-1: Systems with appropriate user interfaces will be available for cafeteria employees to process the expected volume of meals ordered.

AS-2: Cafeteria staff and vehicles will be available to deliver all meals for specified delivery time slots within 15 minutes of the requested delivery time.

DE-1: If a restaurant has its own on-line ordering system, the Cafeteria Ordering System must be able to communicate with it bi-directionally.

>>

## 2. Limitations & Exclusions

*[Identify any product features or characteristics that a stakeholder might anticipate, but which are not planned to be included in the new product]*

## 3. Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

<<Sample

| **ID** | **Category** | **Rule Definition** |
| --- | --- | --- |
| BR-01 | Constraints | Delivery time windows are 15 minutes, beginning on each quarter hour |
| BR-02 | Constraints | Deliveries must be completed between 10:00 A.M. and 2:00 P.M. local time, inclusive. |
| BR-03 | Facts | All meals in a single order must be delivered to the same location. |
| BR-04 | Facts | All meals in a single order must be paid for by using the same payment method. |
| BR-11 | Constraints | If an order is to be delivered, the patron must pay by payroll deduction. |
| BR-12 | Computations | Order price is calculated as the sum of each food item price times the quantity of that food item ordered, plus applicable sales tax, plus a delivery charge if a meal is delivered outside the free delivery zone. |
|  |  |  |

>>

## 4. ..