

Requirement & Design Specification

**Project management**

**Version: 1.0**

– Hanoi, August 2024 –

# Record of Changes

| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- | --- |
| V1.0 | 20/1 | A |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[Record of Changes 2](#_gjdgxs)

[I. Overview 4](#_30j0zll)

[1. User Requirements 4](#_1fob9te)

[1.1 Actors 4](#_3znysh7)

[1.2 Use Cases 4](#_2et92p0)

[2. Overall Functionalities 5](#_tyjcwt)

[2.1 Screens Flow 5](#_3dy6vkm)

[2.2 Screen Descriptions 5](#_1t3h5sf)

[2.3 Screen Authorization 5](#_4d34og8)

[2.4 Non-UI Functions 6](#_2s8eyo1)

[3. System High Level Design 6](#_17dp8vu)

[3.1 Database Design 6](#_3rdcrjn)

[3.2 Code Packages 7](#_35nkun2)

[II. Requirement Specifications 8](#_1ksv4uv)

[1. <<Feature Name>> 8](#_44sinio)

[1.1 <<UseCaseCode\_UC Name>> 8](#_2jxsxqh)

[2. Common Functions 11](#_z337ya)

[2.1 UC-2\_Login System 11](#_3j2qqm3)

[3. Patron Feature 12](#_1y810tw)

[3.1 UC-5\_Order a Meal 12](#_4i7ojhp)

[3.2 UC-6\_Register for Payroll Deduction 13](#_2xcytpi)

[III. Design Specifications 14](#_1ci93xb)

[1. <<Feature Name>> 14](#_3whwml4)

[1.1 <<SubFeature Name>> 14](#_2bn6wsx)

[1.2 System Access 15](#_qsh70q)

[IV. Appendix 19](#_3as4poj)

[1. Assumptions & Dependencies 19](#_1pxezwc)

[2. Limitations & Exclusions 19](#_49x2ik5)

[3. Business Rules 19](#_2p2csry)

[4. .. 19](#_147n2zr)

# I. Overview

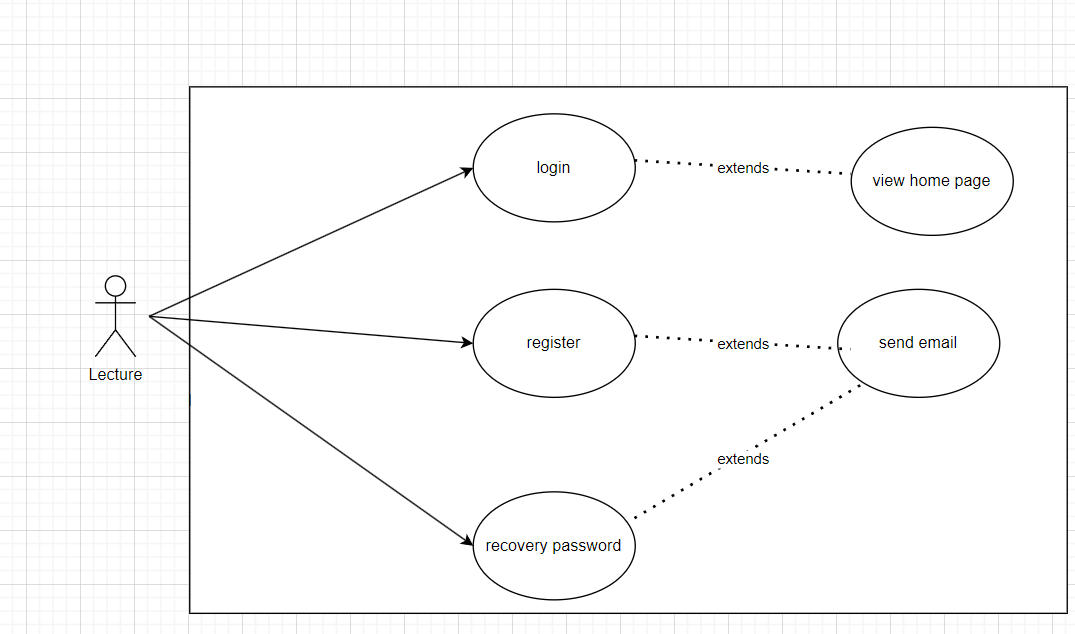
## 1. User Requirements

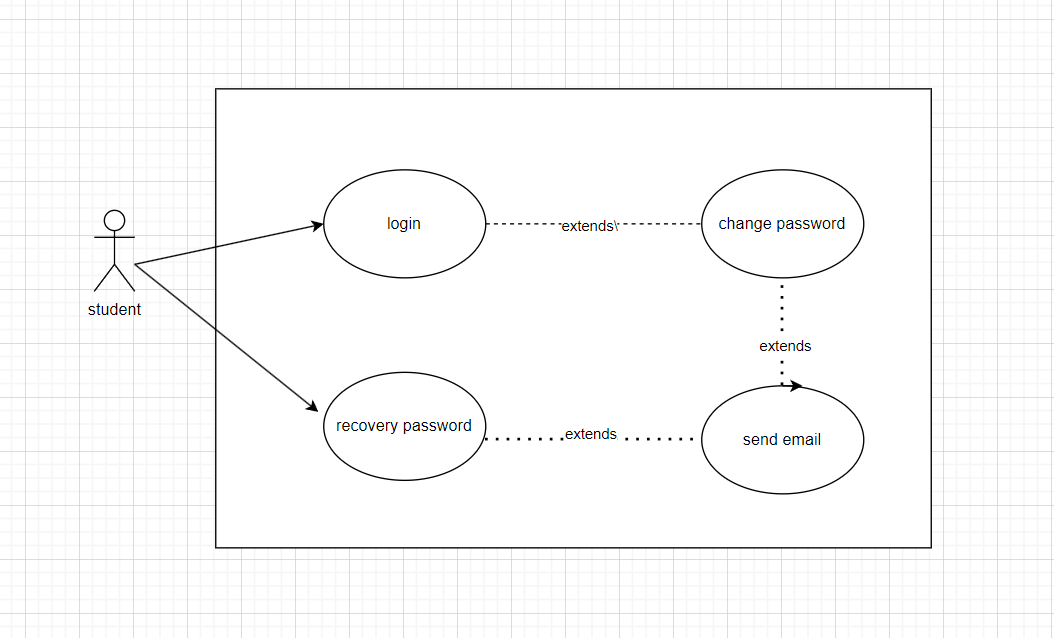
### 1.1 Actors

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Student | Use System for manage SWP Project |
| 2 | Lecture | People use the system to view the progress of projects and grade students |

### 1.2 Use Cases

#### a. Diagram(s)





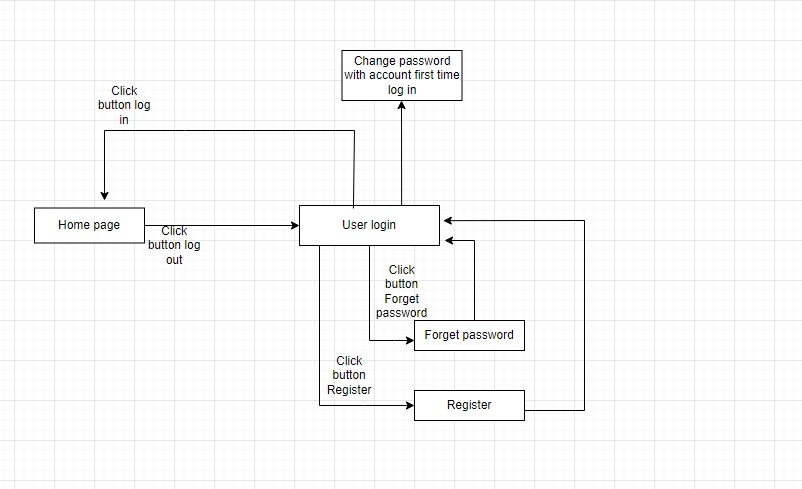
#### b. Descriptions

| **ID** | **Feature** | **Use Case** | **Use Case Description** |
| --- | --- | --- | --- |
| 01 | Login | Login to Home page | Enter email, password to confirm account, forward to Home page |
| 02 | Register | Register a lecturer's account | Enter email, user name, password to create a new account |
| 03 | Forgot Password | Reset for a new password | Enter email then the system will send to their gmail a random password. Users return to the Change password page to change the new password. |
| 04 | Change Password | Get a new password | Enter email then the system will send to their gmail a random password. Reload page and enter a new password. |

## 

## 2. Overall Functionalities

### 2.1 Screens Flow



### 2.2 Screen Descriptions

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Order Meals | Create Order | <<Screen Brief description>> |
| 2 | Order Meals | Change Order |  |
| 3 | .. |  |  |

### 2.3 Screen Authorization

| **Screen** | **Role\_Student** | **Role-Lecture** |
| --- | --- | --- |
| Login | X | X |
| Forgot Password | X | X |
| Register account |  | X |
| Change password | X | X |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| … |  |  |  |  |

### 2.4 Non-UI Functions

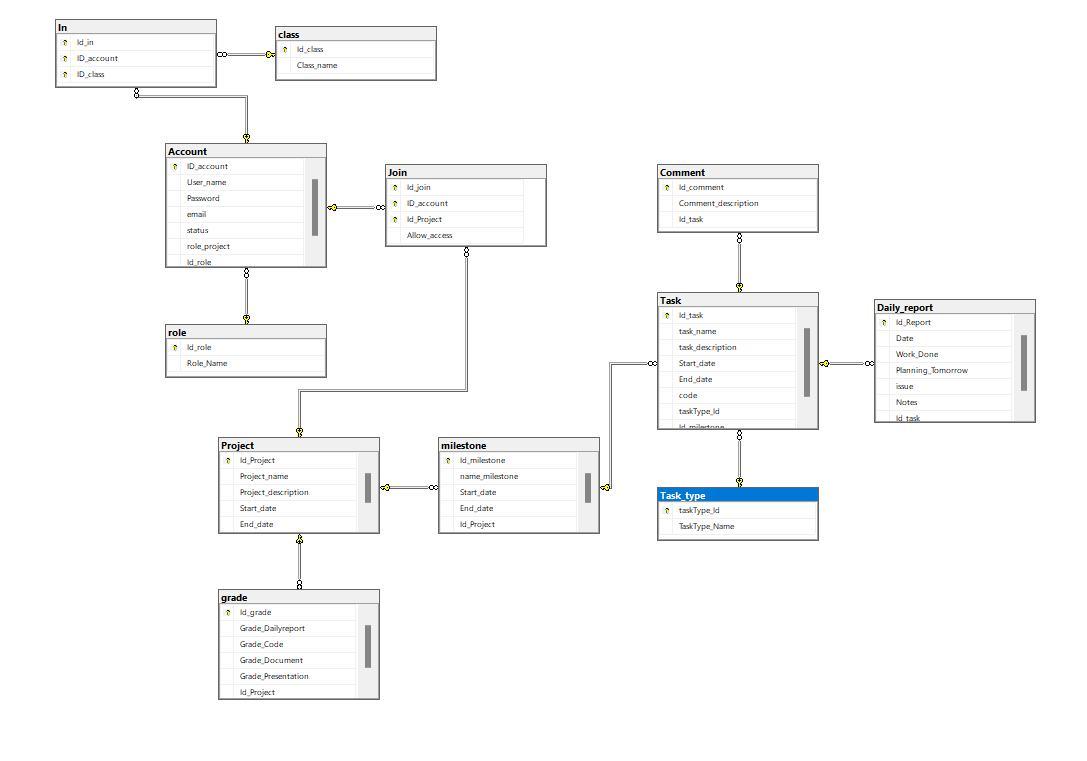
| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Send Email | Send email, generate password | Send email for student |
| 2 | Encryption password | Encryption password | Encryption password (request key only admin had) |

## 

## 3. System High Level Design

### 3.1 Database Design

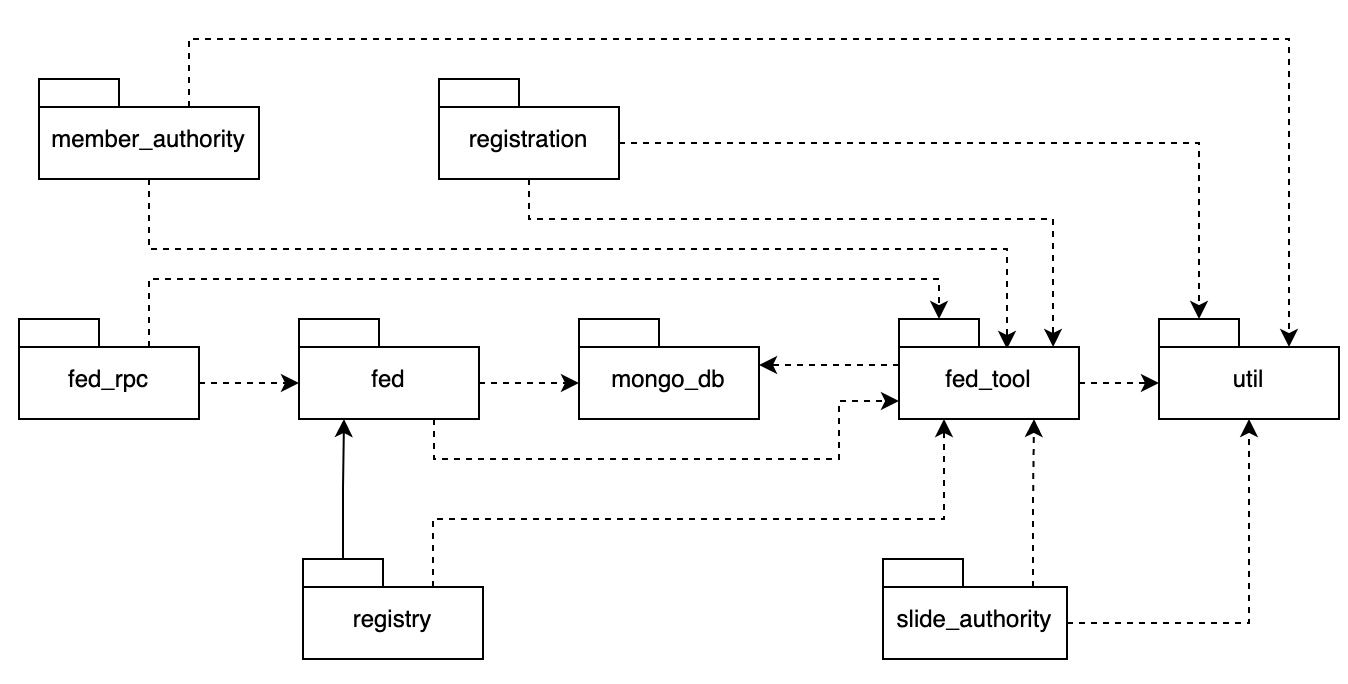
#### a. Database Schema



#### b. Table Descriptions

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *Account* | *- Primary keys: Id\_Account*  *- Foreign keys: id\_role*  *- Main function: Show detail of member* |
| *02* | *Role* | *- Primary keys: id\_role*  *- Main function: Divide tasks to each team member* |
| *03* | *Project* | *- Primary keys:,id\_project*  *- Main function: Show detailed content of each project* |
| *04* | *Join* | *- Primary keys: Id\_Account, id\_join ,id\_project*  *- Foreign keys: id\_project Id\_Account*  *- Main function: Who are join with that project* |
| *05* | *Grade* | *- Primary keys: Id\_grade*  *- Foreign keys: id\_project*  *- Main function: Show grade of each project* |
| *06* | *Daily report* | *- Primary keys: Id\_report*  *- Foreign keys: id\_project*  *- Main function: Show report of each task* |
| *07* | *Task* | *- Primary keys: Id\_task*  *- Foreign keys: id\_tasktype*  *- Main function: Show detailed content of each task* |
| *08* | *Milestone* | *- Primary keys: Id\_milestone, Id\_task, Id\_project*  *- Foreign keys: Id\_task, Id\_project*  *- Main function: Show detail of each milestone: start date, and date, name* |
| *09* | *Task\_type* | *- Primary keys: Id\_type*  *- Main function: Show issue of each task: Bug, error, failure,....* |
| *10* | *Comment* | *- Primary keys: Id\_commnet*  *- Foreign keys: Id\_task*  *- Main function: Show comment for each task* |
| *11* | *Daily report* | *- Primary keys: Id\_report*  *- Foreign keys: Id\_task*  *- Main function: Show process of each task for lecture and team member follow* |
| *12* | *In* | *- Primary keys: Id\_in*  *- Foreign keys: Id\_account, Id\_class*  *- Main function: Show student’s class* |

### 3.2 Code Packages



***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *Member\_authority* | *<Description of the package>* |
| *02* | *registration* | *<Description of the package>* |
| *03* | *…* |  |

# II. Requirement Specifications

## 1. <<Feature Name>>

### 1.1 <<UseCaseCode\_UC Name>>

#### a. Functionalities

**Functional Description Template**

| UC ID and Name: | login | | |
| --- | --- | --- | --- |
|  |  | | |
| Created By: | Trung, Minh | Date Created: | 4/1/2024 |
| Primary Actor: | Lecture ,Student | Secondary Actors: |  |
| Trigger: | None | | |
| Description: | As a user, I want to be able to log into the system so that I can use the system’s authenticated features, access my personalized account. | | |
| Preconditions: | Had account fpt | | |
| Postconditions: | Go to home page | | |
| Normal Flow: | Enter UserName and Password then click login button | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High (Medium, Low), Must Have (Should Have, Could Have),.. | | |
| Frequency of Use: | Daily | | |
| Business Rules: | Only mail fpt | | |
| Other Information: | None | | |
| Assumptions: | None | | |

| UC ID and Name: | Register | | |
| --- | --- | --- | --- |
|  |  | | |
| Created By: | Tùng, Bình | Date Created: | 22/1/2024 |
| Primary Actor: | Student | Secondary Actors: |  |
| Trigger: | None | | |
| Description: | *As a user, I want to be able to register information with the system, from there the system will store personal data such as email and password.* | | |
| Preconditions: | Had account fpt | | |
| Postconditions: | Go to login page | | |
| Normal Flow: | Enter the rigister button so enter the your email and password | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High (Medium, Low), Must Have (Should Have, Could Have),.. | | |
| Frequency of Use: | Daily | | |
| Business Rules: | Only mail fpt | | |
| Other Information: | None | | |
| Assumptions: | None | | |

| UC ID and Name: | Recovery password | | |
| --- | --- | --- | --- |
|  |  | | |
| Created By: | Tùng, Linh | Date Created: | 22/1/2021 |
| Primary Actor: | Lecture ,Student | Secondary Actors: |  |
| Trigger: | None | | |
| Description: | As a user, I would like to be able to support the function of changing a forgotten password using the information registered with the system. | | |
| Preconditions: | Had account fpt | | |
| Postconditions: | Go to forgot page | | |
| Normal Flow: | Entered email has been registered in the system | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High (Medium, Low), Must Have (Should Have, Could Have),.. | | |
| Frequency of Use: | Daily | | |
| Business Rules: | Only mail fpt | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### b. Business Rules

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

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| FR1 | Password Encoding | User’s password must be encoded with MD5 hashing |

## 2. Common Functions

### 2.1 UC-2\_Login System

#### a. Functional Description

| UC ID and Name: | **UC-2\_Login System** | | |
| --- | --- | --- | --- |
| Created By: | Trung, Minh | Date Created: | 8/1/2024 |
| Primary Actor: | Lecture,Student | Secondary Actors: | None |
| Trigger: | None | | |
| Description: | As a student, I want to be able to log into the system so that I can use the system’s authenticated features and access my personalized account. | | |
| Preconditions: | User account has been created & authorized | | |
| Postconditions: | * Student logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Flow | **2.0 Login System**  1. Student accesses the Student Login screen  2. Student types in the login details or choose other login options  3. Student clicks the Login button  4. System validates the login details  5. System allows Student to access  6. System tracks Student’s success login to the Activity Log  7. System accesses the Home Page (or the previous calling page if any) | | |
| Exceptions: | ***E1 System can’t authenticate the user***  1. The Error Message screen is shown to the Student  2. Student cancels the logging in *=> UC stops, change to UC-1\_View Home Page*  3. Student clicks “Forgot Password?” link *=> change to UC-3\_Reset Password*  4. Student clicks “Register” link *=> change to UC-4\_Register User Account* | | |
| Priority: | Must Have | | |
| Frequency of Use: | Whenever Student want to enter the system | | |

#### 

| UC ID and Name: | **UC-2\_Register System** | | |
| --- | --- | --- | --- |
| Created By: | Tùng, Bình | Date Created: | 13/1/2024 |
| Primary Actor: | Lecture | Secondary Actors: | None |
| Trigger: | None | | |
| Description: | As a lecture, I want to be able to have a account so that I can use the system’s authenticated features and access my personalized account. | | |
| Preconditions: | User account has been created & authorized | | |
| Postconditions: | * Lecture create a account * The system verify into the Activity Log | | |
| Normal Flow | **2.0 Register System**  1. Lecture accesses the Lecture Register screen  2. Lecture types in the register details  3. Lecture clicks the Register button  4. System validates the register details  5. System allows Lecture to access  6. System tracks Lecture ’s success register to the Activity Log  7. System accesses the Home Page (or the previous calling page if any) | | |
| Exceptions: | ***E1 System can’t authenticate the user***  1. The Error Message screen is shown to the Lecture  2. Lecture cancels the signing in *=> UC stops, change to UC-1\_View Home Page* | | |
| Priority: | Must Have | | |
| Frequency of Use: | Whenever Lecture want to create account to login the system | | |

## 3. Patron Feature

### 3.1 UC-5\_Order a Meal

#### a. Functional Description

| ID and Name: | **UC-5 Order a Meal** | | |
| --- | --- | --- | --- |
| Created By: | Prithvi Raj | Date Created: | 10/4/13 |
| Primary Actor: | Patron | Secondary Actors: | Cafeteria Inventory System |
| Description: | A Patron accesses the Cafeteria Ordering System from the corporate intranet or from home, views the menu for a specific date if desired, selects food items, and places an order for a meal to be delivered to a specified location within a specified 15-minute time window. | | |
| Trigger: | A Patron indicates that he wants to order a meal | | |
| Preconditions: | PRE-1. Patron is logged into COS.  PRE-2. Patron is registered for meal payments by payroll deduction. | | |
| Postconditions: | POST-1. Meal order is stored in COS with a status of “Accepted”.  POST-2. Inventory of available food items is updated to reflect items in this order.  POST-3. Remaining delivery capacity for the requested time window is updated. | | |
| Normal Flow: | **5.0 Order a Single Meal**   1. Patron asks to view menu for a specific date. (see 5.0.E1, 5.0.E2) 2. COS displays menu of available food items and the daily special. 3. Patron selects one or more food items from menu. (see 5.1) 4. Patron indicates that meal order is complete. (see 5.2) 5. COS displays ordered menu items, individual prices, and total price, including taxes and delivery charge. 6. Patron either confirms meal order (continue normal flow) or requests to modify meal order (return to step 2). 7. COS displays available delivery times for the delivery date. 8. Patron selects a delivery time and specifies the delivery location. 9. Patron specifies payment method. 10. COS confirms acceptance of the order. 11. COS sends Patron an email message confirming order details, price, and delivery instructions. 12. COS stores order, sends food item information to Cafeteria Inventory System, and updates available delivery times. | | |
| Alternative Flows: | **5.1 Order multiple identical meals**   1. Patron requests a specified number of identical meals. (see 5.1.E1) 2. Return to step 4 of normal flow.   **5.2 Order multiple meals**   1. Patron asks to order another meal. 2. Return to step 1 of normal flow. | | |
| Exceptions: | **5.0.E1 Requested date is today and current time is after today’s order cutoff time**  1. COS informs Patron that it’s too late to place an order for today.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests another date, then COS restarts use case.  **5.0.E2 No delivery times left**  1. COS informs Patron that no delivery times are available for the meal date.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests to pick the order up at the cafeteria, then continue with normal flow, but skip steps 7 and 8.  **5.1.E1 Insufficient inventory to fulfill multiple meal order**  1. COS informs Patron of the maximum number of identical meals he can order, based on current available inventory.  2a. If Patron modifies number of meals ordered, then Return to step 4 of normal flow.  2b. Else if Patron cancels the meal ordering process, then COS terminates use case. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 300 users, average of one usage per day. Peak usage load for this use case is between 9:00 A.M. and 10:00 A.M. local time. | | |
| Business Rules: | BR-1, BR-2, BR-3, BR-4, BR-11, BR-12, BR-33 | | |
| Other Information: | 1. Patron shall be able to cancel the meal ordering process at any time prior to confirming it. 2. Patron shall be able to view all meals he ordered within the previous six months and repeat one of those meals as the new order, provided that all food items are available on the menu for the requested delivery date. (Priority = M) 3. The default date is the current date if the Patron is using the system before today’s order cutoff time. Otherwise, the default date is the next day that the cafeteria is open. | | |
| Assumptions: | Assume that 15 percent of Patrons will order the daily special (source: previous 6 months of cafeteria data). | | |

#### b. Business Rules

None

### 3.2 UC-6\_Register for Payroll Deduction

#### a. Functional Description

| ID and Name: | **UC-6 Register for Payroll Deduction** | | |
| --- | --- | --- | --- |
| Created By: | Nancy Anderson | Date Created: | 9/15/13 |
| Primary Actor: | Patron | Secondary Actors: | Payroll System |
| Description: | Cafeteria patrons who use the COS and have meals delivered must be registered for payroll deduction. For noncash purchases made through the COS, the cafeteria will issue a payment request to the Payroll System, which will deduct the meal costs from the next scheduled employee payday direct deposit. | | |
| Trigger: | Patron requests to register for payroll deduction, or Patron says yes when COS asks if he wants to register | | |
| Preconditions: | PRE-1. Patron is logged into COS. | | |
| Postconditions: | POST-2. Patron is registered for payroll deduction. | | |
| Normal Flow: | **6.0 Register for Payroll Deduction**   1. COS asks Payroll System if Patron is eligible to register for payroll deduction. 2. Payroll System confirms that Patron is eligible to register for payroll deduction. 3. COS asks Patron to confirm his desire to register for payroll deduction. 4. If so, COS asks Payroll System to establish payroll deduction for Patron. 5. Payroll System confirms that payroll deduction is established. 6. COS informs Patron that payroll deduction is established. | | |
| Alternative Flows: | None | | |
| Exceptions: | 6.0.E1 Patron is not eligible for payroll deduction  6.0.E2 Patron is already enrolled for payroll deduction | | |
| Priority: | High | | |
| Business Rules: | BR-86 and BR-88 govern an employee’s eligibility to enroll for payroll deduction. | | |
| Other Information: | Expect high frequency of executing this use case within first 2 weeks after system is released. | | |

#### b. Business Rules

None

# III. Design Specifications

## 1. <<Feature Name>>

### 1.1 <<SubFeature Name>>

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

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

##### UI Design

*[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>> | <<Field type>> | <<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,…>> |
| .. |  |  |

***SQL Commands***

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

### 1.2 System Access

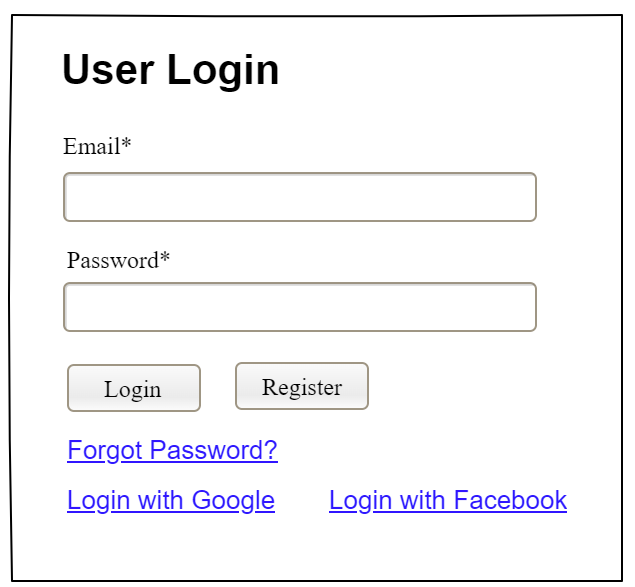
#### a. User Login

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

Related use cases:

* [UC02\_Login System](#_3j2qqm3)

##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Email\* | Text Box | This is for user to input valid email address for logging in |
| Password\* | Password Box | This is for user to input password for logging in |
| Login | Button | User clicks to authenticate him/herself into the system with provided email & password |
| Register | Button | User clicks to redirect to the User Register page for registering new user account to access the system |
| Forgot Password? | Hyperlink | User clicks to redirect to the Password Reset page for resetting his/her forgot password |
| Login with Google | Hyperlink | Allow user to login with his/her Google account |
| Login with Facebook | Hyperlink | Allow user to login with his/her Facebook account |

##### Database Access

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

***SQL Commands:***

1/ Verify UserName & Password information

SELECT user\_id, full\_name, email, image\_url, status

FROM user WHERE user\_name = ? AND password = ?

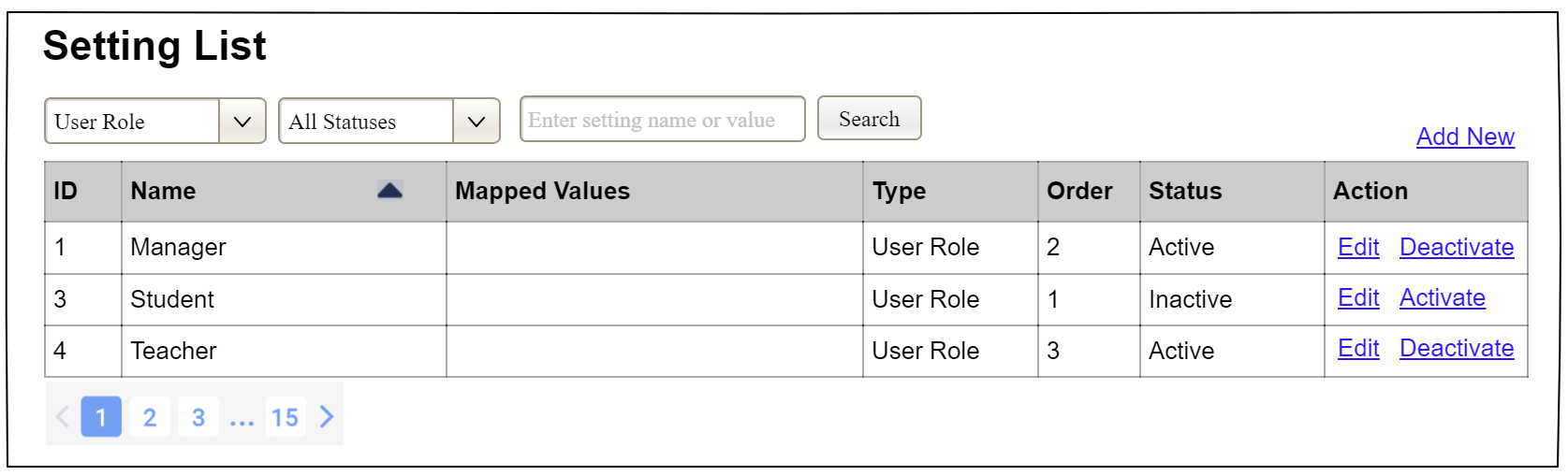
2/ Specify the authorizations of the logged-in user

SELECT mapped\_values FROM setting WHERE setting\_id = ?

SELECT setting\_name, mapped\_values FROM setting WHERE setting\_id IN (?)

#### b. Setting List

##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| ***Search Fields*** | | |
| Setting Type | Combo Box  Single-Choice | Filled with the list of current active setting types  Allow to filter the list by setting type;  Default value is “All Types” |
| Setting Status | Combo Box  Single-Choice | Values: All Statuses (default), Active, and Inactive  Allow to filter the list by status  Default value: “All Statuses” |
| Search Phase | Text Box  String (30) | Allow to search using the name or map values  Default value: blank |
| Search | Button | Click to refresh the list with the defined filter(s) and search phrase. |
| Add New | Hyperlink | 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 |
| ***Data Actions*** | | |
| 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 | Ion | 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 |

***SQL Commands:***

1/ Query the list of current settings from the database

SELECT setting\_id, setting\_name, mapped\_values, type\_id, display\_order, status

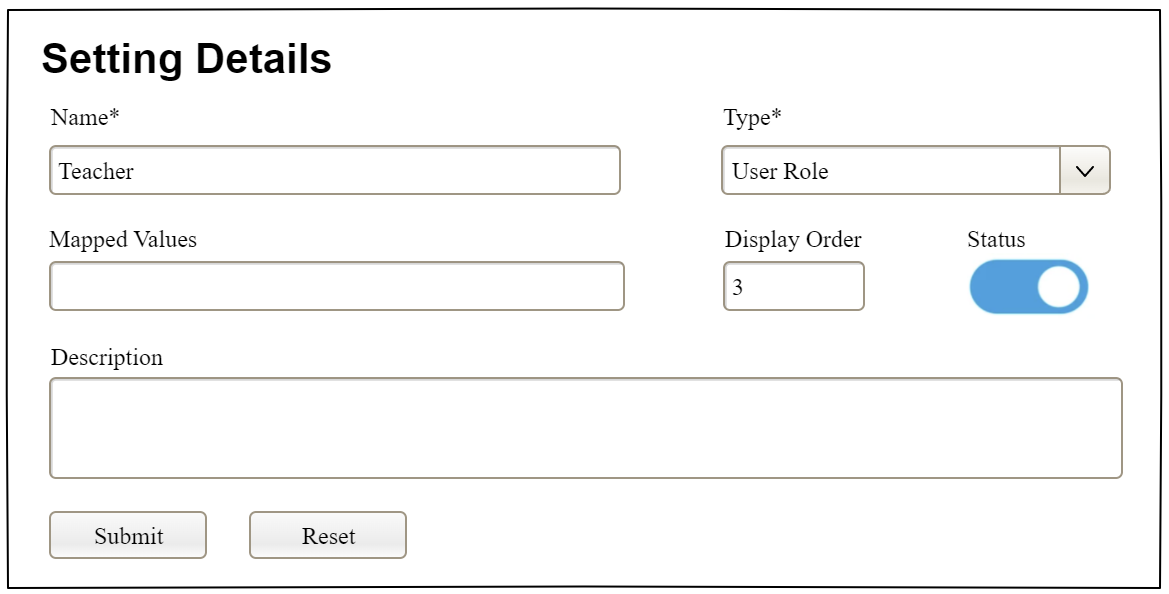
FROM setting WHERE (setting\_type = ?) AND (status = ?) AND (setting\_name LIKE ?)

2/ Update status of a specific setting

UPDATE setting SET status = ? WHERE setting\_id = ?

#### 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. 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. ..