

**CAPSTONE PROJECT REPORT**

**Report 3 – System Requirement Specification**

– Ho Chi Minh, October 2020 –

**Table of Contents**

[I. Project Report 3](#_30j0zll)

[1. Status Report 3](#_1fob9te)

[2. Team Involvements 3](#_3znysh7)

[3. Issues/Suggestions 3](#_2et92p0)

[II. System Requirement Specification 4](#_tyjcwt)

[1. Overall Description 4](#_3dy6vkm)

[1.1 Product Overview 4](#_1t3h5sf)

[1.2 Business Rules 5](#_4d34og8)

[2. User Requirements 6](#_2s8eyo1)

[2.1 Overview 6](#_17dp8vu)

[2.2 <<Feature Name 1 – i.e Order Meals>> 7](#_3rdcrjn)

[2.3 <<Feature name 2 – i.e: Meal Subscriptions>> 10](#_35nkun2)

[2.4 <<Next Feature Name..>> 11](#_44sinio)

[3. Functional Requirements 12](#_2jxsxqh)

[3.1 System Functional Overview 12](#_z337ya)

[3.2 <<Feature Name 1>> 13](#_3j2qqm3)

[3.3 <<Feature Name 2>> 14](#_1y810tw)

[4. Non-Functional Requirements 15](#_4i7ojhp)

[4.1 External Interfaces 15](#_2xcytpi)

[4.2 Quality Attributes 16](#_3as4poj)

[5. Other Requirements 18](#_32hioqz)

[5.1 Appendix1 - Messages List 18](#_1hmsyys)

[5.2 Appendix2 - … 18](#_41mghml)

# I. Project Report

## 1. Status Report

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

## 2. Team Involvements

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| 1 |  | KienNT |  |
| 2 |  | TuanTV |  |
| 3 |  | AnhLM |  |

## 3. Issues/Suggestions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. System Requirement Specification

## 1. Overall Description

### 1.1 Product Overview

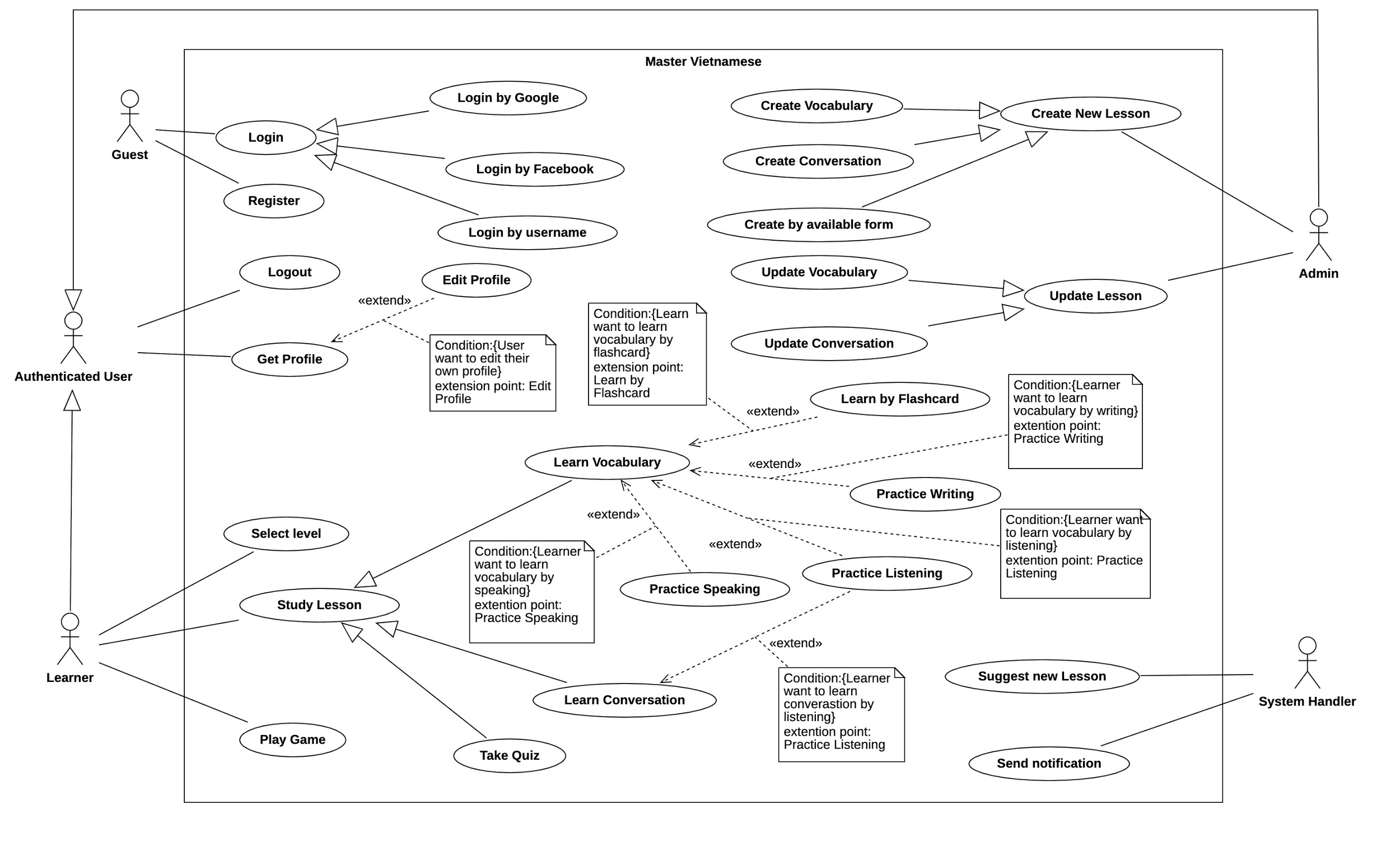
### 1.2 Business Rules

|  |  |
| --- | --- |
| **ID** | **Rule Definition** |
| BR-01 | Must login to the application |
| BR-02 | Must select level to access lesson |
| BR-03 | Must complete the current lesson to unlock next lesson |
| BR-04 | To complete a lesson, take a quiz |
| BR-05 | Estimate time of loading lesson to lesson screen about 3 to 5 seconds |
| BR-06 | Estimate time of uploading to server about 30 to 60 seconds |
| BR-07 | Must have an account |
| BR-08 | Must have a unique username |
| BR-09 | Must learn at least one lesson |
| BR-10 |  |

## 2. User Requirements

### 2.1 Overview

#### a. Use Case Diagram



#### b. System Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator |  |
| 2 | System Handler |  |
| 3 | Learner |  |
| 4 | Authenticated User |  |
| 5 | Guest |  |

#### c. Use Cases List

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Primary Actors** | **Secondary Actors** |
| 01 | Login | Guest |  |
| 02 | Sign up | Guest |  |
| 03 | Get Profile | Authenticated User |  |
| 04 | Logout | Authenticated User |  |
| 05 | Select Level | Learner |  |
| 06 | Play game | Learner |  |
| 07 | Study Lesson | Learner |  |
| 08 | Learn Vocabulary | Learner |  |
| 09 | Learn Conversation | Learner |  |
| 10 | Take quiz | Learner |  |
| 11 | Create New Lesson | Administrator |  |
| 12 | Create Vocabulary | Administrator |  |
| 13 | Create Conversation | Administrator |  |
| 14 | Create by available form | Administrator |  |
| 15 | Update Lesson | Administrator |  |
| 16 | Update Vocabulary | Administrator |  |
| 17 | Update Conversation | Administrator |  |

### 2.2 <<Feature Name 1 – >>

#### a. <<Use Case Name 1.1>>

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC - Select Level | | |
| Created By: | BaoHQ | Date Created: | Oct 20, 2020 |
| Primary Actor: | Learner | Secondary Actors: |  |
| Trigger: | Register button | | |
| Description: | List all levels after learner registered a new account. We have three levels that learner can choose:   * Beginner: for people that never learn Vietnamese * Intermediate: for people that have known about Vietnamese * Advance: for people know Vietnamese and want to improve Vietnamese skill | | |
| Preconditions: | 1. Learner must register new account and login to the application | | |
| Post-conditions: | 1. List all the lesson that match with the level | | |
| Normal Flow: | 1. Learner registered new account then choose level | | |
| Alternative Flows: | 1. Learner change their level by get into setting screen | | |
| Exceptions: | Login Fail, Invalid username, Invalid password, Invalid email | | |
| Priority: | High | | |
| Frequency of Use: |  | | |
| Business Rules: | BR-1 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. <<Use Case Name 1.2 – Login >>

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-01 - Login** | | |
| Created By: | TuongDM | Date Created: | Oct 20, 2020 |
| Primary Actor: | Guest | Secondary Actors: |  |
| Description: | Users login the lesson system through the mobile application.  There are three ways to login to the application:   * Login by username and password * Login by facebook * Login by google | | |
| Trigger: | Login button, Facebook button, Google button | | |
| Preconditions: | PRE-1. Learner must sign up an account  PRE-2. Learner must have Facebook account  PRE-3. Learner must have Google account | | |
| Post-conditions: | Success: Navigate to home screen.  Fail: Notify a message “login failed” to screen | | |
| Normal Flow: | 1. Enter username into username field 2. Enter password into password field 3. Press Login button 4. Guest can choose access the system with Facebook by Facebook button 5. Guest can choose access the system with Google by Google button | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Incorrect username or password | | |
| Priority: | High | | |
| Frequency of Use: | Whenever guests want to login to the system. | | |
| Business Rules: | BR-07 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### c. <<Next Use Case Name 1.3 - Sign Up>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-02 - Sign Up Account** | | |
| Created By: | TuongDM | Date Created: | Oct 20, 202 |
| Primary Actor: | Guest | Secondary Actors: |  |
| Description: | Guest register an account | | |
| Trigger: | Sign up button | | |
| Preconditions: | Guest does not have an account | | |
| Post-conditions: | Success: An account will be create and save into database  Fail: Unable to sign up an account | | |
| Normal Flow: | 1. Enter username into username field 2. Enter email into email field 3. Enter password into password field 4. Enter confirm password into confirm password field | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot connect to database.  Username is already existed. | | |
| Priority: | High | | |
| Frequency of Use: | Whenever guest want to create new account | | |
| Business Rules: | BR-08 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### d. <<Next Use Case Name 1.4 - Get Profile>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-03 - Get Profile** | | |
| Created By: | AnhND | Date Created: | Oct 22, 2020 |
| Primary Actor: | Authenticated User | Secondary Actors: |  |
| Description: | Get all authenticate user profile include:   * Username * Progress * Profile picture * Setting | | |
| Trigger: | Profile tab in navigation bar | | |
| Preconditions: | PRE-1. Learner must sign in with their account  PRE-2. Learner must sign in with their Facebook account  PRE-3. Learner must sign in with their Google account | | |
| Post-conditions: | Success: Show all profiles of authenticated user.  Fail: Cannot get profiles of authenticated user. | | |
| Normal Flow: | 1. Enter username into username field 2. Enter password into password field 3. Press Login button 4. Guest can choose access the system with Facebook by Facebook button 5. Guest can choose access the system with Google by Google button 6. Authenticated User press on Profile tab. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot connect to database to get user profile. | | |
| Priority: | High | | |
| Frequency of Use: | Whenever user click on profile tab | | |
| Business Rules: | BR1 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### e. <<Next Use Case Name 1.5 - Logout>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-04 - Logout** | | |
| Created By: | AnhND | Date Created: | Oct 22, 2020 |
| Primary Actor: | Authenticated User | Secondary Actors: |  |
| Description: | Authenticated User logout of the system | | |
| Trigger: | Authenticated user press the logout button in the profile tab to logout their account. | | |
| Preconditions: | PRE-1. Guest must sign in with their account  PRE-2. Guest must sign in with their Facebook account  PRE-3. Guest must sign in with their Google account | | |
| Post-conditions: | Success: Back to login screen  Fail: | | |
| Normal Flow: | 1. Authenticated User press on Profile tab. 2. Authenticated User press Logout button. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot logout of the system | | |
| Priority: | High | | |
| Frequency of Use: | Whenever authenticated user want to logout | | |
| Business Rules: | BR1 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### f. <<Next Use Case Name 1.7 - Study Lesson>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-05 - Study Lesson** | | |
| Created By: | AnhND | Date Created: | Oct 22, 2020 |
| Primary Actor: | Learner | Secondary Actors: |  |
| Description: | List all the lesson to lesson tab screen for learner base on the level | | |
| Trigger: | After learner login, all the lessons will show. | | |
| Preconditions: | PRE-1. Must login to the system  PRE-2. Must choose the level | | |
| Post-conditions: | Success: List all the lessons that match with level.  Fail: Fail to load lessons to list on the lesson tab screen. | | |
| Normal Flow: | 1. Login to the system. 2. Navigate to lesson screen and load all the lesson | | |
| Alternative Flows: | 1. Learner presses on suggest new lesson notification. | | |
| Exceptions: | Cannot connect to database. | | |
| Priority: | High | | |
| Frequency of Use: | Whenever learners login to the system or re-open the application. | | |
| Business Rules: | BR1,BR2, BR3 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***g.<<Use Case Name 1.8 - Learn Vocabulary>>***

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-06 - Study Lesson** | | |
| Created By: | CuongHX | Date Created: | Oct 22, 2020 |
| Primary Actor: | Learner | Secondary Actors: |  |
| Description: | List all the functions to learn vocabulary. There are two functions to learn vocabulary:   * Learn by Flashcard * Learn by writing | | |
| Trigger: | Learner | | |
| Preconditions: | PRE-1. Guest must sign in with their account  PRE-2. Guest must sign in with their Facebook account  PRE-3. Guest must sign in with their Google account | | |
| Post-conditions: | N/A | | |
| Normal Flow: | 1. After login and choose level, learner choose the lesson 2. Learner click on the vocabulary button 3. Choose learn by flashcard or learn writing | | |
| Alternative Flows: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Low | | |
| Frequency of Use: |  | | |
| Business Rules: | BR1, BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### h. <<Next Use Case Name 1.9 - Learn Conversation>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-07 - Learn Conversation** | | |
| Created By: | BaoHQ | Date Created: | Oct 22, 2020 |
| Primary Actor: | Learner | Secondary Actors: |  |
| Description: | List all conversations in this lesson. Students can practice listening in conversation. | | |
| Trigger: | Learners click on the conversation button in lesson detail. | | |
| Preconditions: | PRE-1. Learner must login to the application  PRE-2. Learner must choose the level  PRE-3. Learner must choose the lesson | | |
| Post-conditions: | Success: List all the conversations to the screen for students.  Fail: Nothing to load conversation onto the screen. | | |
| Normal Flow: | 1. After login and choose level, learner choose the lesson 2. Learner click on the conversation button 3. To listen the conversation learner click on the speaker icon | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Fail to load conversation for Learner | | |
| Priority: | High | | |
| Frequency of Use: |  | | |
| Business Rules: | BR1,BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### i. <<Next Use Case Name 1.10 - Take Quiz>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-08 - Take Quiz** | | |
| Created By: | BaoHQ | Date Created: | Oct 23, 2020 |
| Primary Actor: | Learner | Secondary Actors: |  |
| Description: | Every lesson has a quiz and learner can take one or more quizzes depending on the lesson. | | |
| Trigger: | Learners click on the quiz button in lesson detail. | | |
| Preconditions: | PRE-1. Learner must login to the application  PRE-2. Learner must choose the level  PRE-3. Learner must choose the lesson  PRE-4. Learner must start quiz | | |
| Post-conditions: | Success: Show question and answer for learner and learner can take quiz and after learner finish there quiz the system will show score  Fail: Cannot load questions and options for Learner to take quiz. | | |
| Normal Flow: | 1. After login and choose level, learner choose the lesson 2. Learner click on the quiz button 3. System will show quiz for the learner question by question 4. After learner done, system will show final result 5. If the final result reached the required to open next lesson, the next lesson will unlock 6. If the final result has not reached the required, next lesson will not unlock | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Learner cancel the quiz while taking the quiz  Cannot load quiz to the screen | | |
| Priority: | High | | |
| Frequency of Use: | Whenever a learner wants to take a quiz. | | |
| Business Rules: | BR1, BR2, BR3 | | |
| Other Information: | If a learner cancels while taking a quiz, this quiz will reset and all quiz progress will be shut down and the result will be not saved. | | |
| Assumptions: | N/A | | |

#### j. <<Next Use Case Name 1.7 - Create New Lesson>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-09 - Create New Lesson** | | |
| Created By: | BaoHQ | Date Created: | Oct 23, 2020 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Description: | For administrator to create new lesson to the system and add it to database | | |
| Trigger: | Administrator click on the create new lesson button on the web admin dashboard. | | |
| Preconditions: | PRE-1. Administrator must login to the web admin dashboard  PRE-2. Admin choose the create new lesson button | | |
| Post-conditions: | Success: Show all methods that will create new lesson.  Fail: Create a new lesson failed. | | |
| Normal Flow: | 1. After login, administrator chooses to create the new lesson. 2. Choose method that will create new lesson. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Error from server that cannot create new lesson. | | |
| Priority: | Medium | | |
| Frequency of Use: | Whenever administrator wants to create a new lesson. | | |
| Business Rules: | BR-01, BR-06 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### 

#### k. <<Next Use Case Name 1.7 - Create Vocabulary>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-10 - Create Vocabulary** | | |
| Created By: | BaoHQ | Date Created: | Oct 23, 2020 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Description: | For administrator to create new vocabulary to the system and add it to database | | |
| Trigger: | Administrator click on the create new lesson button on the web admin dashboard. And choose method create vocabulary | | |
| Preconditions: | PRE-1. Administrator must login to the web admin dashboard  PRE-2. Administrator choose the create new lesson button  PRE-3. Administrator choose the create vocabulary method | | |
| Post-conditions: | Success: Show all methods that will create new lessons.  Fail: Create a new vocabulary failed. | | |
| Normal Flow: | 1. After login, the administrator chooses to create the new lesson. 2. Choose the create vocabulary method. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Error from server that cannot create vocabulary | | |
| Priority: | Medium | | |
| Frequency of Use: | Whenever administrator wants to create a vocabulary. | | |
| Business Rules: | BR-01, BR-06 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### j. <<Next Use Case Name 1.7 - Play Game>>

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-09 - Play Game** | | |
| Created By: | AnhND | Date Created: | Oct 23, 2020 |
| Primary Actor: | Learner | Secondary Actors: |  |
| Description: | Learner play games about Vietnamese language to practice Vietnamese language skills. | | |
| Trigger: | Learner choose the game tab menu. | | |
| Preconditions: | PRE-1. Guest must login to the mobile application.  PRE-2. Learner must learn at least one lesson. | | |
| Post-conditions: | Success: Show game screen.  Fail: Leaner does not start any lesson. | | |
| Normal Flow: | 1. Learner choose the game tab menu. 2. Learner choose a game to play. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Learner does not start any lesson. | | |
| Priority: | Medium | | |
| Frequency of Use: | Whenever learner wants to play a game. | | |
| Business Rules: | BR1, BR6 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### 

### 2.3 <<Feature name 2 – i.e: Meal Subscriptions>>

#### a. <<Use Case Name 2.1 – i.e Register for Payroll Deduction>>

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-05 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: | **5.0 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: | 5.0.E1 Patron is not eligible for payroll deduction  5.0.E2 Patron is already enrolled for payroll deduction | | |
| Priority: | High | | |
| Frequency of Use: |  | | |
| 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. | | |
| Assumptions: |  | | |

#### b. <<Next Use Case Name 2.x>>

*[Use Case Description in the same format as above]*

### 2.4 <<Next Feature Name..>>

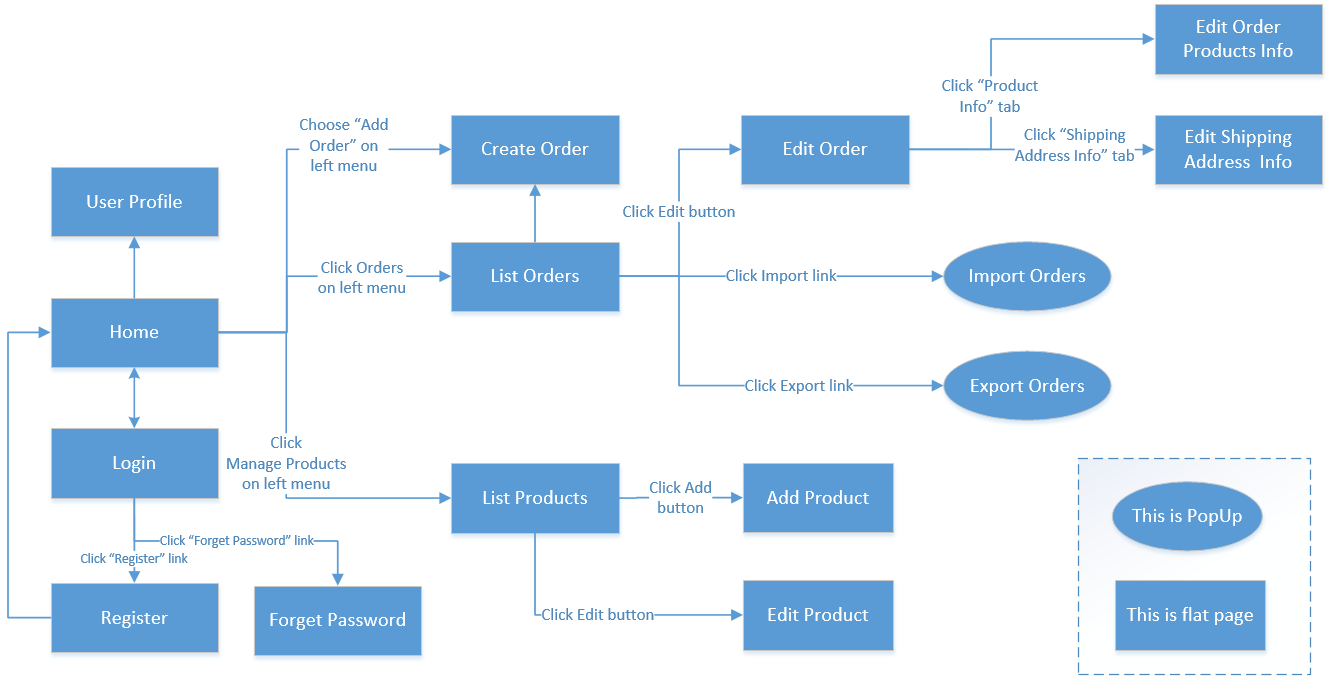
…

## 3. Functional Requirements

### 3.1 System Functional Overview

#### a. Screen Flow

*[This part show the system screens and the relationship among screens. You can draw the Screens Flow for the system in the form of diagram as below]*



#### b. Screen Details

*[Provide the descriptions for the screens in the Screens Flow above]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Feature** | **Screen** | **Description** |
| 01 | Order Meals | Create Order | <<Screen Brief description>> |
| 02 | Order Meals | Change Order |  |
| 03 | .. |  |  |

#### c. Screen Authorization

*[Provide the system roles authorization to the system features (down to screens, and event to the screen activities if applicable) in the table form as below]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen** | **Role1** | **Role2** | **Role3** | **Role4** | **RoleX** |
| <<Screen Name1>> | X |  |  | X | X |
| <<Screen Activity>> |  |  |  | X | X |
| <<Screen Name2>> | X |  |  | X |  |
| Query All Data | X |  |  |  |  |
| Query Own Data |  |  |  | X |  |
| Query Managed Data |  |  |  | X |  |
| Add New Data |  |  |  | X | X |
| Update All Data |  |  |  |  | X |
| Update Own Data |  |  |  |  | X |
| Update Managed Data |  |  |  |  | X |
| Delete Data |  |  |  |  |  |
| … |  |  |  |  |  |

In which:

* Role1: <<role1 description>>
* Role2: <<role2 description>>
* …

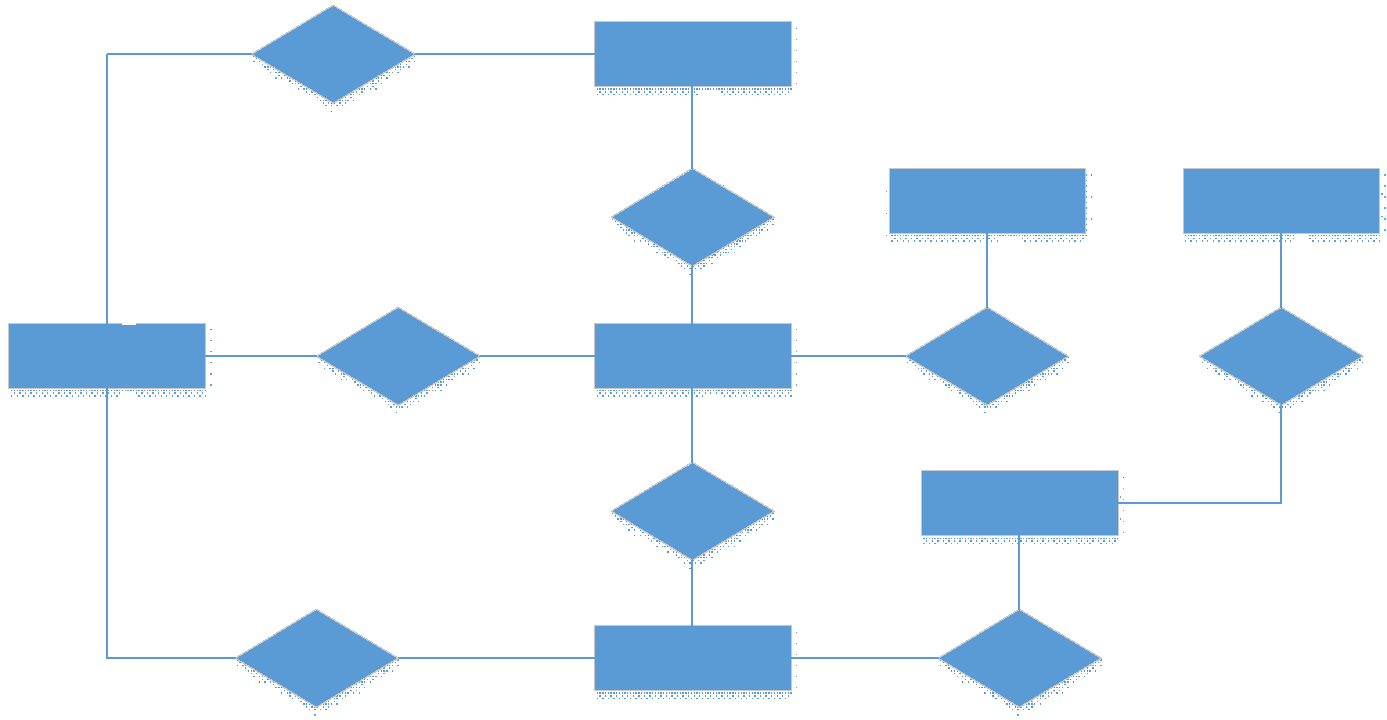
#### d. Non-Screen Functions

*[Provide the descriptions for the non-screen system functions, i.e batch/cron job, service, API, etc.]*

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

#### e. Entity Relationship Diagram

*[Provide the entity relationship diagram and the entity descriptions in the table format as below]*



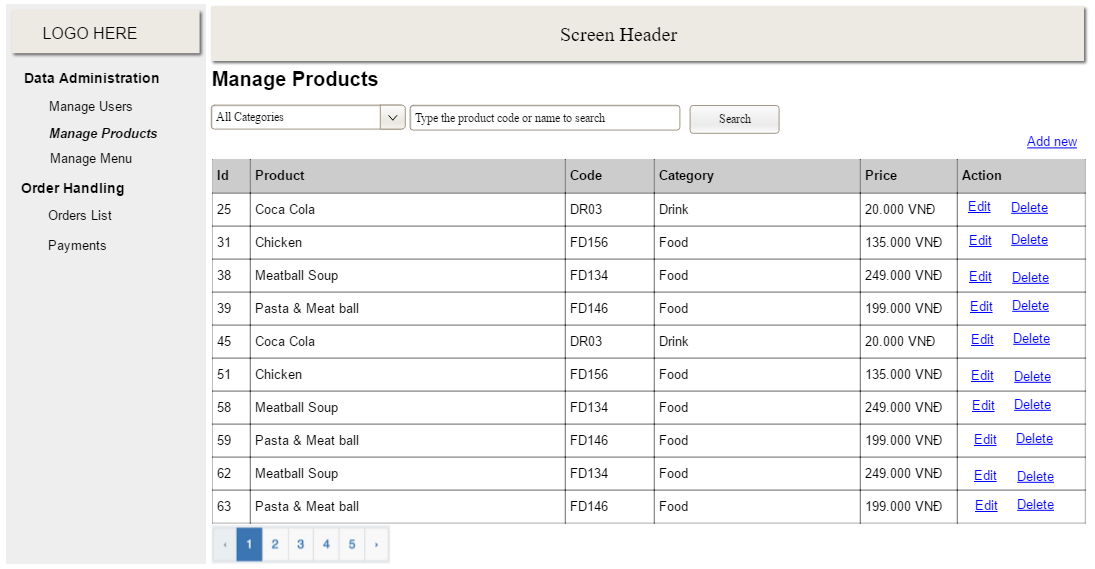
|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | User |  |
| 2 | Meal |  |
| 3 | Meal Subscription |  |
| 4 | … |  |

### 3.2 <<Feature Name 1>>

#### a. <<Function Name 1>>

*[A function can be a screen or a non-screen function (listed in the part 5.1 above). In this part, you need to provide the details on the related function, focus on mentioning below information*

* *Function trigger: how this function is triggered (navigation path, a timing frequency, etc.*
* *Function description: actors/roles, purpose, interface, data processing, etc.*
* *Screen layout: mockup prototype of the screen, sample below is for Manage Products screen*

**

* *Function Details: provide explanation for the data, validation, functionalities (for both normal cases and abnormal cases), etc. of the function so that the reader can image how it work.*

*]*

#### b. <<Function Name 2>>

…

### 3.3 <<Feature Name 2>>

…

## 4. Non-Functional Requirements

### 4.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.]*

#### a. User Interfaces

*[Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.]*

UI-1: The Cafeteria Ordering System screen displays shall conform to the Process Impact Internet Application User Interface Standard, Version 2.0 [3].

UI-2: The system shall provide a help link from each displayed webpage to explain how to use that page.

UI-3: The webpages shall permit complete navigation and food item selection by using the keyboard alone, in addition to using mouse and keyboard combinations.

#### b. Software Interfaces

*[Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the inte-component communications. Identify data that will be exchanged between or shared across software components. Specify non-functional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.]*

SI-1: Cafeteria Inventory System

SI-1.1: The COS shall transmit the quantities of food items ordered to the Cafeteria Inventory System through a programmatic interface.

SI-1.2: The COS shall poll the Cafeteria Inventory System to determine whether a requested food item is available.

SI-1.3: When the Cafeteria Inventory System notifies the COS that a specific food item is no longer available, the COS shall remove that food item from the menu for the current date.

SI-2: Payroll System

The COS shall communicate with the Payroll System through a programmatic interface for the following operations:

SI-2.1: To allow a Patron to register and unregister for payroll deduction.

SI-2.2: To inquire whether a Patron is registered for payroll deduction.

SI-2.3: To inquire whether a Patron is eligible to register for payroll deduction.

SI-2.4: To submit a payment request for a purchased meal.

SI-2.5: To reverse all or part of a previous charge because a patron rejected a meal or wasn’t satisfied with it, or because the meal was not delivered per the confirmed delivery instructions.

#### c. Hardware Interfaces

*[Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document]*

No hardware interfaces have been identified.

#### d. Communications Interfaces

*[State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.]*

CI-1: The COS shall send an email or text message (based on user account settings) to the Patron to confirm acceptance of an order, price, and delivery instructions.

CI-2: The COS shall send an email or text message (based on user account settings) to the Patron to report any problems with the meal order or delivery.

### 4.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification. Some of the possible attributes are provided with the guide/descriptions are mentioned here]*

#### a. Usability

*[This section includes all those requirements that affect usability. For example, specify the required training time for a normal users and a power user to become productive at particular operations specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]*

#### b. Reliability

*[Requirements for reliability of the system should be specified here. Some suggestions follow:*

*Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.*

*Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.*

*Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?*

*Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.*

*Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).*

*Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]*

#### c. Performance

*[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.*

*Response time for a transaction (average, maximum)*

*Throughput, for example, transactions per second*

*Capacity, for example, the number of customers or transactions the system can accommodate*

*Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)*

*Resource utilization, such as memory, disk, communications, and so forth.]*

#### d. Dependability

*[Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system]*

##### d1. Security

*[Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.]*

##### d2. Safety

*[Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.]*

#### e. Supportability

*[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]*

#### f. Design Constraints

*[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]*

#### g. Support Documents

*[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]*

#### h. Purchased Components

*[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]*

## 5. Other Requirements

*[Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. ]*

### 5.1 Appendix1 - Messages List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In line | There is not any search result | *No search result.* |
| 2 | MSG02 | In red, under the text box | Input-required fields are empty | *The \* field is required.* |
| 3 | MSG03 | Toast message | Updating asset(s) information successfully | *Update asset(s) successfully.* |
| 4 | MSG04 | Toast message | Adding new asset successfully | *Add asset successfully.* |
| 5 | MSG05 | Toast message | Confirming email of asset hand-over is sent successfully | *A confirmation email has been sent to {email\_address}.* |
| 6 | MSG06 | Toast message | Resetting asset information successfully | *Return asset(s) successfully.* |
| 7 | MSG07 | Toast message | Deleting asset information successfully | *Delete asset(s) successfully.* |
| 8 | MSG08 | In red, under the text box | Input value length > max length | *Exceed max length of {max\_length}.* |
| 9 | MSG09 | In line | Username or password is not correct when clicking sign-in | *Incorrrect user name or password. Please check again.* |

### 5.2 Appendix2 - …