

**Capstone Project Report**

**Report 3 – Software Requirement Specification**

– Hanoi, August 2019 –

**Table of Contents**

[I. Record of Changes 3](#_Toc116163308)

[II. Software Requirement Specification 4](#_Toc116163309)

[1. Product Overview 4](#_Toc116163310)

[2. User Requirements 5](#_Toc116163311)

[2.1 Actors 5](#_Toc116163312)

[2.2 Use Cases 6](#_Toc116163313)

[3. Functional Requirements 7](#_Toc116163314)

[3.1 System Functional Overview 7](#_Toc116163315)

[3.2 <<Feature Name 1>> 16](#_Toc116163316)

[3.3 <<Feature Name 2>> 19](#_Toc116163317)

[4. Non-Functional Requirements 20](#_Toc116163318)

[4.1 External Interfaces 20](#_Toc116163319)

[4.2 Quality Attributes 20](#_Toc116163320)

[5. Requirement Appendix 21](#_Toc116163321)

[5.1 Business Rules 21](#_Toc116163322)

[5.2 Common Requirements 21](#_Toc116163323)

[5.3 Application Messages List 21](#_Toc116163324)

[5.4 Other Requirements… 22](#_Toc116163325)

# I. Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 05/10/2022 | A | GiangNT | Screen Flows |
| 06/10/2022 | A | KienNT | Screen Description; Screen Authorization |
| 07/10/2022 | A | HuyenBN | Use Case Diagram |
| 07/10/2022 | A | KienNT | Use Case Description |
| 07/10/2022 | M | GiangNT | Use Case Description |
| 08/10/2022 | A | HuyenBN | Entity Relationship Diagram |
| 08/10/2022 | A | KienNT | Entity Descriptions |
| 08/10/2022 | M | GiangNT | Entity Descriptions |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Product Overview

*[Gives the overall description about the product with some introduction and the context diagram. The context diagram presents the boundary and connections between the system you’re developing and everything else in the universe. This identifies external entities (or terminators – software, hardware, human components, and other systems) outside the system that interface to it in some way, as well as data, control, and material flows between the terminators and the system.]*

<<Sample: The Cafeteria Ordering System is a new software system that replaces the current manual and telephone processes for ordering and picking up meals in the Process Impact cafeteria. The context diagram below illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately connecting to the Internet ordering services for several local restaurants and to credit and debit card authorization services.



>>

## 2. User Requirements

### 2.1 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 | Administrator |  |
| 2 | Menu Manager |  |
| 3 | … |  |

### 2.2 Use Cases

#### 2.2.1 Diagram(s)



#### 2.2.2 Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| 01 | Log In | Guest | Guest logs into the system |
| 02 | View List Of Available Houses | Guest | Guest views list of houses |
| 03 | Search Available Houses | Guest | Guest searches for houses by name or filters it by category or by map |
| 04 | View a House Detail | Guest | Guest views the detail information of house |
| 05 | Review a House | Student | Student rates and comments the house |
| 06 | Report Violated House | Student | Student reports the violated house |
| 07 | Log In with Email and Password | Admin | Admin logs into the system |
| 08 | Manage Staff’s Account | Admin | Admin views the list of all staff accounts and can be change active status of the staff |
| 09 | Change Password | Admin | Admin changes login password |
| 10 | Log In with Email and Password | Staff | Staff logs into the system |
| 11 | View Dashboard | Staff | Staff views list of available houses and list of landlord’s request. Staff can be search for houses & statistics by name or filter it by category at the same time can also accept or decline landlord’s request |
| 12 | View Profile | Staff | Staff views and updates the profile |
| 13 | Sign Up | Landlord | Landlord registers for an account |
| 14 | Log In | Landlord | Landlord logs into the system |
| 15 | Upload House Information | Landlord | Landlord imports information of the house by excel file |
| 16 | Manage Houses | Landlord | Landlord views list of rooms, detail information of room at the same time can also add new, update, delete, change status the room |
| 17 | View Profile | Landlord | Landlord views and updates the profile |

## 3. Functional Requirements

### 3.1 System Functional Overview

#### 3.1.1 Screens Flow

Student



Figure 1-1: Screen Flow for User

Landlord



Figure 1-2: Screen Flow for Landlord

Staff



Figure 1-3: Screen Flow for Staff

Staff



Figure 1-3: Screen Flow for Staff

#### 3.1.2 Screen Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
|  | View List Houses | Home Page | View list of houses |
|  | Search & Filter House | Home Page | Search for houses by name or filter it by category |
|  | Interact with map | Home Page | Use the map to find the location of the house |
|  | View House Detail | House Details | Display detail information of house |
|  | View Room Detail | Room Details | Display detail information of room |
|  | Report House | Report House | User reports the house |
|  | Login | User Login | User logs into the system |
|  | Register | User Register | User registers for an account |
|  | Register | Register [Landlord] | Landlord registers for an account |
|  | Verify Information Of Landlord | Input Information [Landlord] | Get information of landlord |
|  | Verify Identity Card Of Landlord | Upload Identity Card Image [Landlord] | Get identity card of landlord |
|  | View List Houses | Dashboard [Landlord] | Landlord views list of houses |
|  | Search & Filter House | Dashboard [Landlord] | Landlord searché for houses by name or filters it by category |
|  | Upload House Information | Upload House Information [Landlord] | Landlord imports information of the house |
|  | Manage Rooms | House Detail [Landlord] | Display detail information of house |
|  | Search & Filter Room | House Detail [Landlord] | Landlord searché for rooms by name or filters it by category |
|  | Update House | Update House [Landlord] | Landlord updates the house |
|  | Delete House | Delete House [Landlord] | Landlord deletes the house |
|  | Update Room | Update Room [Landlord] | Landlord updates the room |
|  | Add Room | Add Room [Landlord] | Landlord adds new the room |
|  | Delete Room | Delete Room [Landlord] | Landlord deletes the room |
|  | Update Proflie | Update Proflie [Landlord] | Landlord updates the proflie |
|  | View List Houses & Statistics | Dashboard [Staff] | Staff view list of available houses |
|  | Search & Filter House & Statistic | Dashboard [Staff] | Staff searches for houses & statistics by name or filter it by category |
|  | Accept Or Decline Landlord’s Signup Request | List Of Landlord’s Signup Request [Staff] | Staff views list of landlord’s request and accepts or decline |
|  | Update Proflie | Update Proflie [Staff] | Staff updates the proflie |
|  | View List Accounts Of Landlords | Landlords List [Staff] | Staff views list of landlords |
|  | View Landlord Detail | Landlord Detail [Staff] | Staff views detail information of landlord |
|  | View List Of Reported Houses | List Of Reported Houses [Staff] | Staff views list of reported houses |
|  | View List Accounts Of Staffs | Dashboard [Admin] | Admin views the list of all staff accounts |
|  | Search & Filter Staff | Dashboard [Admin] | Admin searches for staffs by name or filter it by category |
|  | Change Password | Change Password [Admin] | Admin changes login password |
|  | Create Staff Account | Create Account [Admin] | Admin creates account for staff |
|  | Update Staff Account | Update Account [Admin] | Admin updates account of staff |
|  | Detele Staff Account | Detele Account [Admin] | Admin deletes account of staff |

#### 3.1.3 Screen Authorization

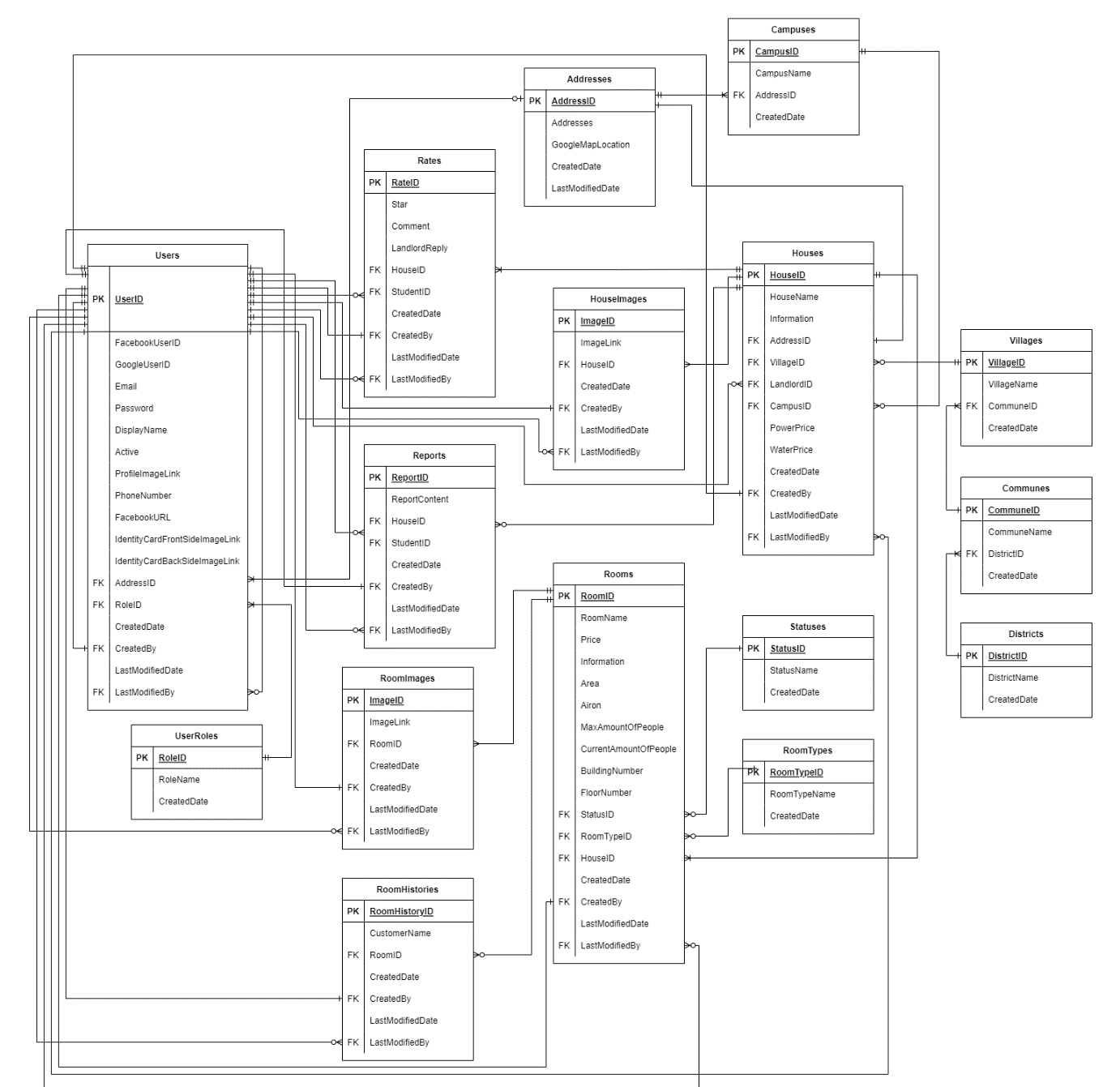
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen** | **Guest** | **Student** | **Landlord** | **Staff** | **Admin** |
| Home Page | X | X | X | X | X |
| House Details | X | X | X | X | X |
| Room Details | X | X | X | X | X |
| Report House |  | X |  |  |  |
| User Login |  | X | X | X | X |
| User Register | X |  |  |  |  |
| Register [Landlord] |  |  | X |  |  |
| Input Information [Landlord] |  |  | X |  |  |
| Upload Identity Card Image [Landlord] |  |  | X |  |  |
| Dashboard [Landlord] |  |  | X |  |  |
| Upload House Information [Landlord] |  |  | X |  |  |
| House Detail [Landlord] |  |  | X |  |  |
| Update House [Landlord] |  |  | X |  |  |
| Delete House [Landlord] |  |  | X |  |  |
| Update Room [Landlord] |  |  | X |  |  |
| Add Room [Landlord] |  |  | X |  |  |
| Delete Room [Landlord] |  |  | X |  |  |
| Update Proflie [Landlord] |  |  | X |  |  |
| Dashboard [Staff] |  |  |  | X |  |
| List Of Landlord’s Signup Request [Staff] |  |  |  | X |  |
| Update Proflie [Staff] |  |  |  | X |  |
| Landlords List [Staff] |  |  |  | X |  |
| Landlord Detail [Staff] |  |  |  | X |  |
| List Of Reported Houses [Staff] |  |  |  | X |  |
| Dashboard [Admin] |  |  |  |  | X |
| Change Password [Admin] |  |  |  |  | X |
| Create Account [Admin] |  |  |  |  | X |
| Update Account [Admin] |  |  |  |  | X |
| Detele Account [Admin] |  |  |  |  | X |

#### 3.1.4 Non-Screen Functions

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

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

#### 3.1.5 Entity Relationship Diagram



**Entities Description**

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | Addresses | Used to store the address of User, House & Campus |
| 2 | Campuses | Used to store Campuses of FPT university |
| 3 | UserRoles | Used to store the role of each User |
| 4 | Users | Used to store Users in the system |
| 5 | Districts | Used to store Districts around the school |
| 6 | Communes | Used to store Communes around the school |
| 7 | Villages | Used to store Villages around the school |
| 8 | Houses | Used to store Houses |
| 9 | Statuses | Used to store Statuses of Room |
| 10 | RoomTypes | Used to store Types of Room |
| 11 | Rooms | Used to store Rooms |
| 12 | Rates | Used to store Rating & Comments of User |
| 13 | ImagesOfHouse | Used to store Images of House |
| 14 | ImagesOfRoom | Used to store Images of Room |
| 15 | Reports | Used to store student's Reports for House |
| 16 | RoomHistories | Used to store Histories of people staying in the House, for the landlord to voluntarily add if there is a need to manage & monitor |

### 3.2 <<Feature Name 1>>

#### 3.2.1 <<Function Name 1>>

*[A function can be a screen or a non-screen function (listed in the part 3.1.5 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: mock-up prototype of the screen, sample below is for Manage Products screen*

**

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

*]*

#### 3.2.1. Homepage

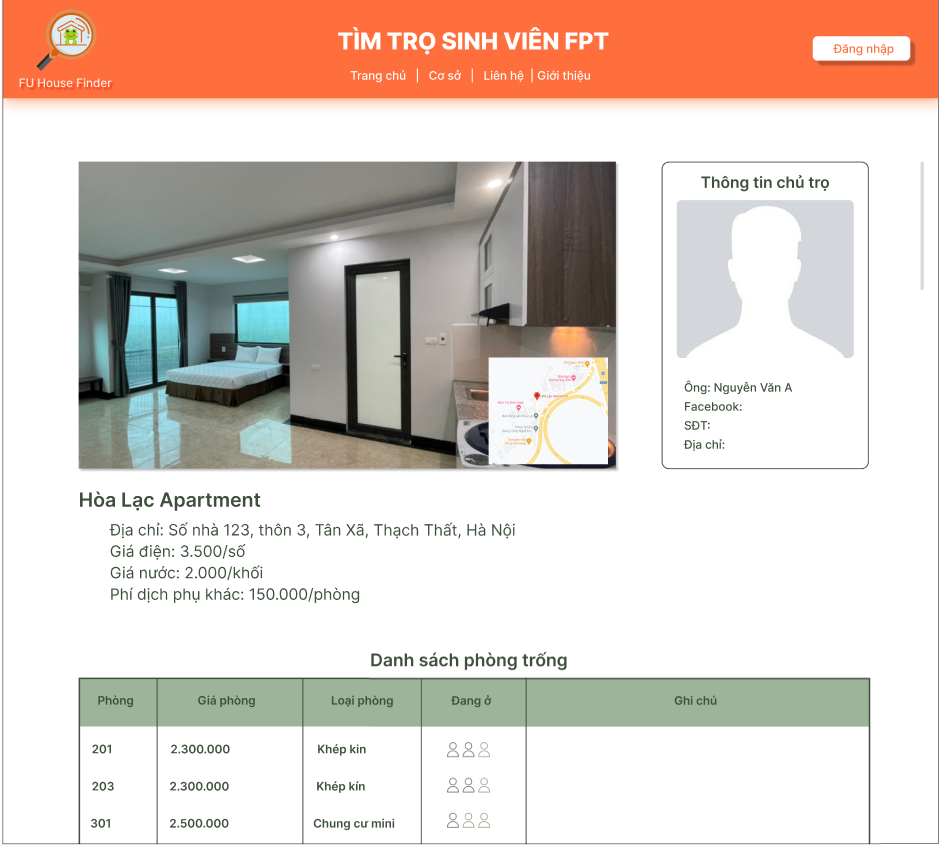
* Function trigger: User accesses the website
* Function description:
  + Role: Guest, Student
  + Purpose: User could view list of available houses and filter the results
* Screen layout:



#### 3.2.2. Login/Signup

* Function trigger: User accesses the website
* Function description:
  + Role: Guest
  + Purpose: Guest could sign up for an account or log in to access more functions
* Screen layout: 

#### 3.2.3. House Detail

* Function trigger: User accesses the website
* Function description:
  + Role: Guest, student
  + Purpose: View a house detail information (landlord’s information, available rooms, etc.)
* Screen layout: 

#### 3.2.4. Room Detail

* Function trigger: User accesses the website
* Function description:
  + Role: Guest, student
  + Purpose: View a room detail information (images, price, etc.)
* Screen layout: 

#### 3.2.5. Report House

* Function trigger: Guest has logged in as a student
* Function description:
  + Role: Student
  + Purpose: Student uses to report an unusual house
* Screen layout:



#### 3.2.2 <<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/system elements.]*

### 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]*

#### 4.2.1 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]*

#### 4.2.2 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.]*

#### 4.2.3 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*

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

#### 4.2.4 …

## 5. Requirement Appendix

*[Provide business rules, common requirements, or other extra requirements information here]*

### 5.1 Business Rules

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

|  |  |
| --- | --- |
| ID | Rule Definition |
| BR-01 | Delivery time windows are 15 minutes, beginning on each quarter hour. |
| BR-02 | Deliveries must be completed between 10:00 A.M. and 2:00 P.M. local time, inclusive. |
| BR-03 | All meals in a single order must be delivered to the same location. |
| BR-04 | All meals in a single order must be paid for by using the same payment method. |
| BR-11 | If an order is to be delivered, the patron must pay by payroll deduction. |
| BR-12 | 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. |
| BR-24 | Only cafeteria employees who are designated as Menu Managers by the Cafeteria Manager can create, modify, or delete cafeteria menus. |
| BR-33 | Network transmissions that involve financial information or personally identifiable information require 256-bit encryption. |
| BR-86 | Only regular employees can register for payroll deduction for any company purchase. |
| BR-88 | An employee can register for payroll deduction payment of cafeteria meals if no more than 40 percent of his gross pay is currently being deducted for other reasons. |

### 5.2 Common Requirements

*[Fill all the common requirements here..]*

### 5.3 Application Messages List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In line | There is not any search result | *No search results.* |
| 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.* |

### 5.4 Other Requirements…