

**Software Requirement Specification**

**Project Code: FMS**

**Document Code: FMS\_SRS\_FindMotel\_v1.4**

**Ho Chi Minh, November 202RECORD OF CHANGE**

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Effective Date** | **Changed Items** | **A\* M, D** | **Change Description** | **New Version** |
| 11/09/2021 | First release | A |  | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**SIGNATURE PAGE**

**ORIGINATOR:** Vo Quoc Bao 01-May-14

**REVIEWERS:**

**APPROVAL:**

Table of content

[1. FIND MOTEL 5](#_Toc87912555)

[1.1 Login 6](#_Toc87912556)

[1.1.1 Overview 6](#_Toc87912557)

[1.1.2 Actors 6](#_Toc87912558)

[1.1.3 Use-case diagrams 7](#_Toc87912559)

[1.1.4 Workflows 7](#_Toc87912560)

[1.1.5 External interfaces 8](#_Toc87912561)

[1.1.6 Business descriptions 9](#_Toc87912562)

[1.1.7 User-validation rules 10](#_Toc87912563)

[1.1.8 Related use-cases 10](#_Toc87912564)

[1.2 View motel 11](#_Toc87912565)

[1.2.1 Overview 11](#_Toc87912566)

[1.2.2 Actors 11](#_Toc87912567)

[1.2.3 Use – case diagram 11](#_Toc87912568)

[1.2.4 Workflows 11](#_Toc87912569)

[1.2.5 External interfaces 13](#_Toc87912570)

[1.3 Find motel 14](#_Toc87912571)

[1.3.1 Overview 14](#_Toc87912572)

[1.3.2 Actors 14](#_Toc87912573)

[1.3.3 Use-case diagram 14](#_Toc87912574)

[1.3.4 Workflows 14](#_Toc87912575)

[1.3.5 GUI 15](#_Toc87912576)

[1.4 Looking for roommates 16](#_Toc87912577)

[1.5 Post a motel 19](#_Toc87912578)

[1.6 Post a Review about Motel 22](#_Toc87912579)

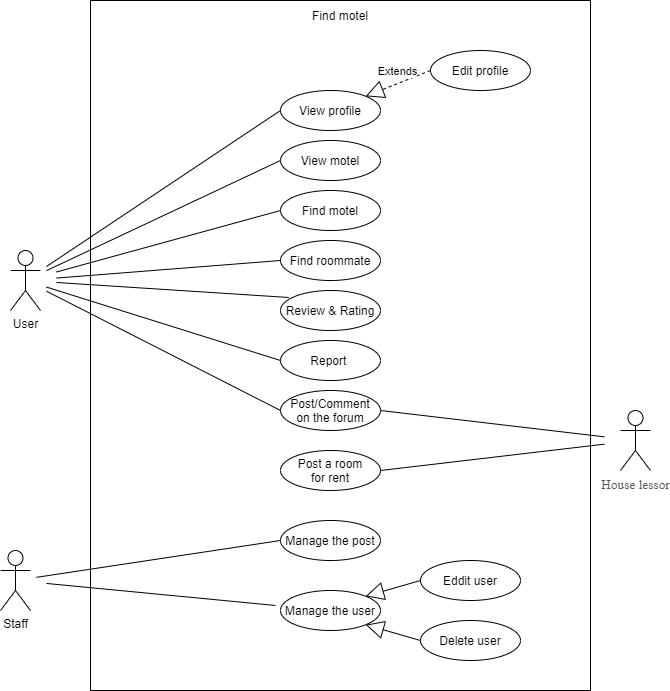
[1.7 Chat/Contact 22](#_Toc87912580)

[2. OTHERS REQUIREMENT (NOT YET) 22](#_Toc87912581)

# FIND MOTEL

Find Motel contains the following functionalities:

* Post a motel
* Attachment
* Motel List
* Motel View or Edit
* Find Motel and Roommate
* Review Motel
* Chat
* List News



## 1.1 Login

### 1.1.1 Overview

Required to login to use the app's functionality

### 1.1.2 Actors

Request everyone

### 1.1.3 Use-case diagrams



#### 1.1.3.1 Pre-condition

User must be logged in

#### 1.1.3.2 Post-condition

Users can use the functions of the application.

#### 1.1.3.3 Triggers

* Clicks on the “ĐĂNG NHẬP” button on the screen
* Clicks on the “Zalo icon” button on the screen
* Clicks on the “Facebook icon” button on the screen
* Clicks on the “Gmail icon” button on the screen

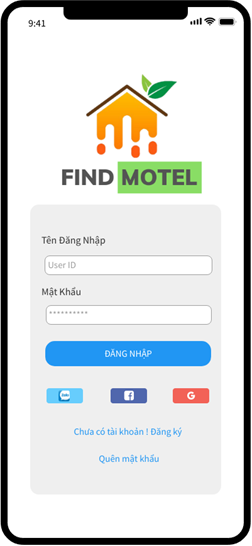
### 1.1.4 Workflows

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Actor** | **System** |
| 1. The user successfully logged in | 1.1. Go to this screen | 1.2. Display home interface of the application |
| 1.3 Users can use the functions of the application | 1.4 Implement user-selected functions |
| 2. User login failed | 2.1 Refer to step 1.1 | 2.2 Refer to step 1.2 |
| 2.3 Refer to step 1.3 | 2.4 Login failed more than 3 times  Send login notification to phone number or email to verified account |

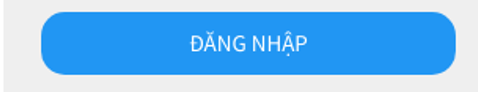
### 1.1.5 External interfaces

#### 1.1.5.1 Prototype:

Login



Button



#### 1.1.5.2 Screen description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Control type** | **Required** | **Data type** | **Default value** | **Description** |
| **Login Form** | | | | | | |
| 1 | Tên Đăng Nhập | Text field | Yes | Text | Blank | Username to login  The username must be under 25 characters |
| 2 | Mật Khẩu | Text field | Yes | Text | Blank | Password to login  At least 6 characters (and up to 35 characters) |
| 3 | Chưa có tài khoản! Đăng ký | Link | No | Text | Blank | Show register form an account for user who do not have an account |
| 4 | Quên mật khẩu | Link | No | Text | Blank | Show forgot password form for user to recover password |
| **Button** | | | | | | |
| 24 | ĐĂNG NHẬP | Button | Yes | N/A | N/A | Check the username and password in the database and go to the application interface |
| 25 | Zalo icon | Button | Yes | N/A | N/A | Connect via Zalo account to log in to the app |
| 26 | Facebook icon | Button | Yes | N/A | N/A | Connect via Facebook account to log in to the app |
| 27 | Gmail icon | Button | Yes | N/A | N/A | Connect via Gmail account to log in to the app |

### 1.1.6 Business descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Control Type** | **Description** |
| **Login Form** | | | |
| 1 | Tên Đăng Nhập | Text field | The username must be under 25 characters |
| 2 | Mật Khẩu | Text field | Password validation  Password must not contain spaces |
| **Button** | | | |
| 3 | Zalo icon | Button | Connect to API to perform login |
| 4 | Facebook icon | Button | Connect to API to perform login |
| 5 | Gmail icon | Button | Connect to API to perform login |

### 1.1.7 User-validation rules

|  |  |  |
| --- | --- | --- |
| **No** | **Validation rule** | **Message Code** |
| 1 | Login failed | ERR\_REQUIRED\_USERNAME  ERR\_REQUIRED\_PASSWORD  ERR\_REQUIRED\_CONNECT |
| 2 | Length of input fields are over limitation | ERR\_REQUIRED\_LENGTH |

### 1.1.8 Related use-cases

* Login

## 1.2 View motel

### 1.2.1 Overview

This function is described for viewing the rooms that are uploaded to the database system. On this screen, the user will have information about the motel room such as the address, size, and price.

### 1.2.2 Actors

All users have permission to view the motel

### 1.2.3 Use – case diagram



#### 1.2.3.1 Pre-condition

- The actor has successfully logged into the system.

#### 1.2.3.2 Post-condition

- A list of rooms will appear on the screen.

#### 1.2.3.3 Triggers

The user clicks on the article title to view images, prices, and sizes of the motel.

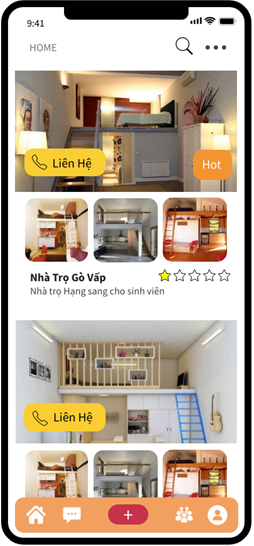
The user clicks on the “Liên hệ” button to view the contact information of the landlords.

### 1.2.4 Workflows

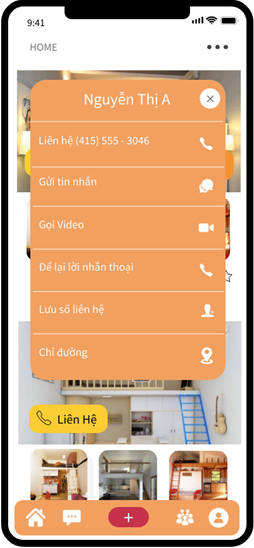
|  |  |  |
| --- | --- | --- |
| **Scenario** | **Actor** | **System** |
| 1. View motel details | 1.1. User clicks on the article title | 1.2. Show motel details including images, videos, prices, room sizes. |
| 1.3 User clicks “←” to back to the home screen | 1.4 Back to home screen |
| 2. View the contact information of the landlords | 2.1 User clicks “Liên hệ” button | 2.2 A dialog box will appear with the contact information of the landlord.  Including phone number, message, video call, leave a voice message, save contact number, and directions to the motel. |
| 2.3 User clicks the phone number in the the dialog box | 2.4 Call the phone number of the landlord. |
| 2.5 User clicks “Gửi tin nhắn” in the the dialog box | 2.6 Open the messaging interface on the application and send a message to the landlord. |
| 2.7 User clicks “Gọi video” in the the dialog box | 2.8 Implement a connection to the landlord account and conduct a video call. |
| 2.9 User clicks “Để lại lời nhắn thoại” in the the dialog box | 2.10 Appear the record button and send it to the landlord via in-app message. |
| 2.11 User clicks “Lưu số liên hệ” in the the dialog box | 2.12 Get the phone number from the database and save it to the phone of the user. |
| 2.13 User clicks “Chỉ đường” in the the dialog box | 2.14 Display the interface directions to the motel.  Find the way from the user location to the motel location. |
| 3. Close dialog box | 3.1 User clicks “X” to back to the home screen | 3.2 Close dialog box  Return to the current screen |

### 1.2.5 External interfaces

#### 1.2.5.1 Prototype



**Figure: Motel details screen**



**Figure: Contact screen**

#### 1.2.5.2 Screen Description

This function uses two screens: View the details motel and contact the landlord

## 1.3 Find motel

### 1.3.1 Overview

This screen displays information that allows users to enter what they are looking for

### 1.3.2 Actors

Everyone can watch

### 1.3.3 Use-case diagram



#### 1.3.3.1 Pre-condition

User must be logged-in.

#### 1.3.3.2 Post-condition

List of hostels by user input

#### 1.3.3.3 Triggers

* Users enter the content to search according to their own needs
* Click the "Submit" button to display the accommodation information as entered

### 1.3.4 Workflows

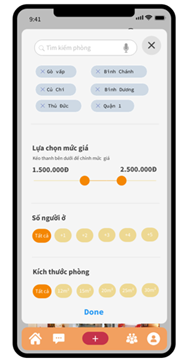
#### 1.3.4.1 Scenario 1 (Fixed Query - Filter)

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. The user enters the information to search (A list of motels appears according to the user's entered requirements) | 2. List of inns according to the request entered by the user   * Innkeeper's name * Phone number * Send Message * Video call * Leave a voice message * Save contact * Direct |
| 3. User enters information related to searching defects in the **Filter** section. |  |
| 4. User click on the button “Submit” | 5. List of inns corresponding to the entered content |

### 1.3.5 GUI

#### 1.3.5.1 Prototype

Filter



#### 1.3.5.2 Screen Description

##### 1.3.5.2.1 Zone 1 - filter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field  Name** | **Control Type** | **Data Type** | **Default Value** | **Required** | **Functional  Description/(\*)ToolTip** |
| 1 | Address | Text field | Text | Yes | Yes | The user enters the address to find the inn to perform the search function of the application |
| 2 | Price | Combo box | Text | All | No | Choose the desired price, in accordance with the existing costs of the user |

## 1.4 Looking for roommates

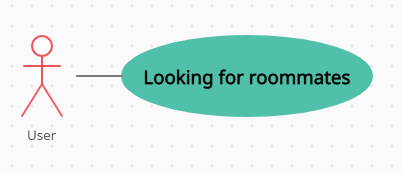
* + 1. **Overview**

With this function, users can find more people to live with. This function is available to all users

* + 1. **Actors**

All user

* + 1. **Use-case diagram**



* + - 1. **Pre-condition**

User must be logged-in the system.

* + - 1. **Post-condition**

Functional interface Find roommates

* + - 1. **Triggers**
    - Users click on the "Send" button to search for information
    1. **Workflows**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User selects the function "Find a roommate" | 2. List of rooms to find roommates |
| 3. Click on any room | 4. The information of the person who needs to find a roommate appears |
| 5. User clicks "Contact" | 6. Ask the user to enter the information and wait for the host to accept it |

* + 1. **External interfaces**
       1. **Prototype**

****

* + - 1. **Screen descriptions**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Control Type** | **Required** | **Data type** | **Default value** | **Description** |
| **Defect detail** | | | | | | |
| 1 | Room photo | image | Yes | Text | Current Value | Room description pictures |
| 2 | Contact | button | No | Date | Blank | Contact the room owner |
| 3 | Full name | Text field | Yes | Text | Current Value | Name of person who wants to share |
| 4 | Photo of identity card | image | Yes | Date | Current Value | Informative photo that proves the person wants to share |
| 5 | Confirm | button | Yes | Text | Current Value | Room owner confirms roommate for roommate |

* + 1. **Business descriptions**

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Description** |
| **Defect detail** | | |
| 1 | Image | Not more than 300KB |
| 2 | Full name | Same with ID card information and no longer than 50 characters |

## 1.5 Post a motel

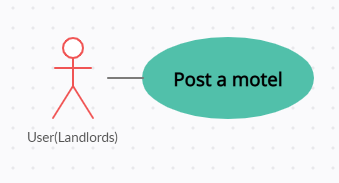
* + 1. **Overview**

This function allows the innkeeper to post inn information to the app

* + 1. **Actors**

Only for innkeepers

* + 1. **Use - case diagram**



* + - 1. ***Pre-condition***

- Log in to your landlord account

* + - 1. ***Post-condition***

An interface that allows the landlord to fill in the inn information appears

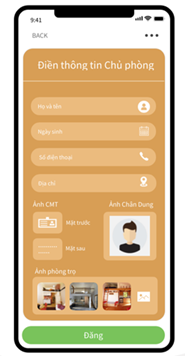
* + - 1. **Triggers**

Press the "Submit" button and wait for the system manager to confirm

* + 1. **Workflows** 
       1. ***Scenario – Batch update***

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1.Click on the function interface “ **Post a motel**” | 2. A form will appear asking the user to enter information |
| 2. Press the "Send" button | 3. Wait for the management system to confirm |
| 4. User change some necessary fields of the selected defects |  |
| 5. User clicks on **Save** button | 6. Save all modifications and return to List Defect screen. |

* + 1. **External interfaces**
       1. ***Prototype***



* + - 1. ***Screen Description***

Each defect is listed here with following information:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Control type** | **Data type** | **Default value** | **Require** | **Functional Description/(\*) Tool tips** |
| 1 | Fill in the owner's informationI | Label | Text | Copy from previous | Yes | Information of the innkeeper |
| 2 | First and last name | Text box | Text | Copy from previous | Yes | Name and surname of the innkeeper |
| 3 | Date of birth | datetime | date | Copy from previous | No | Innkeeper's date of birth  - format dd/mm/yy |
| 4 | Phone number | Text box | Text | Copy from previous | Yes | Landlord's phone number |
| 5 | Address | Text box | Text | Copy from previous | Yes | Address of the inn |
| 6 | Photo of identity card | image | Text | Copy from previous | Yes | Photos and information of the innkeeper |
| 7 | Hostel photo | image | Text | Copy from previous | Yes | Room photo posted |

* + 1. **Business Description**
       1. ***Data fields***

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Business rules** |
| 1 | First and last name | First and last name must not exceed 40 characters |
| 2 | Date of birth | Must enter the same identity card |
| 3 | Phone number | Number sequence of 10 characters |
| 4 | Address | No longer than 60 characters |
| 5 | Images | Not more than 300KB |

## 1.6 Post a Review about Motel

## 1.7 Chat/Contact

# OTHERS REQUIREMENT (NOT YET)

* 1. **Header of page**
     1. **Overview**

Always display Header in all screens.

* + 1. **External interfaces**
       1. **Prototype:**



* + - 1. **Screen description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field name** | **Control type** | **Data type** | **Description** |
| 1 | Project Environment | Menu | Text |  |
| 2 | Assign User | Menu | Text | Submenu of “Project Environment” menu |
| 3 | Work Product | Menu | Text | Submenu of “Project Environment” menu |
| 4 | Module | Menu | Text | Submenu of “Project Environment” menu |
| 5 | Setup Environment | Menu | Text | Submenu of “Project Environment” menu |
| 6 | Mange Defect | Menu | Text |  |
| 7 | Manage Views | Menu | Text | Submenu of “Mange Defect” menu  Go to Home page |
| 8 | New Defect | Menu | Text | Submenu of “Mange Defect” menu  Go to Add New defect screen |
| 9 | Manage Planned Defect | Menu | Text | Submenu of “Mange Defect” menu |
| 10 | Project Summary | Menu | Text | Submenu of “Mange Defect” menu |
| 11 | Import Defect | Menu | Text | Submenu of “Mange Defect” menu |
| 12 | Logout | Menu | Text | Logout system, go to Login screen |
| 13 | Help | Menu | Text | Download & open file “Guideline\_Defect Management System.doc” |
| 14 | Defect management system <screen name> | Label | Text |  |
| 15 | User | Label | Text | Value = type of login user |
| 16 | Group | Label | Text | Value = group of login user |
| 17 | Login Date | Label | Text | Value = today  Date format = mm/dd/yy |
| 18 | Position | Label | Text | Role of login user |
| 19 | Project | Combo box | Text | List of project which login user is assigned  Default value = the current project |
| 20 | Status | Combo box | Text | Status of project  Default value = the current status of project |

* 1. **Data Format**
     1. **Date/Time Format**

Default date format is mm/dd/yy For example: 05/29/14

Default time format is hh:mm:ss For example: 07:23:55

* + 1. **Number Format**

No zero at prefix

File size: xx.x KB For example: 192.3 KB

Available size: xx KB For example: 456 KB

Order number has no comma separated For example: 12345

Decimal places of % is 1 For example: 25.7%

* 1. **Order of reference data**
     1. **QC Activity**

10-Unit test (Type of Activity = Test)

11-Integration test (Type of Activity = Test)

12-System test (Type of Activity = Test)

13-Acceptance test (Type of Activity = Test)

14-Regression test (Type of Activity = Test)

15-After release test (Type of Activity = Test)

16-Other test (Type of Activity = Test)

17-Prototype test (Type of Activity = Test)

20-Document review (Type of Activity = Review)

21-Code review (Type of Activity = Review)

22-After release review (Type of Activity = Review)

23-Prototype review (Type of Activity = Review)

24-Other review (Type of Activity = Review)

30-Quality gate inspection (Type of Activity = Inspection)

31-Final inspection (Type of Activity = Inspection)

32Other inspection (Type of Activity = Inspection)

33-UT inspection (Type of Activity = Inspection)

40-Baseline audit (Type of Activity = Audit)

41-Other audit (Type of Activity = Audit)

* + 1. **Defect Origin**

Administration

Coding

Collaborator Management

Configuration Management

Contract Management

Correction

Customer Support

Deployment

Design

Document Control

Facility Management

IS Management

Internal Audit

Management Review

Other

Prevention

Process Improvement

Project Management

Quality Control

Quality Planning

Recruitment

Requirement

Retirement

Staff Management

Student Management

Subcontract Management

Technology management

Test

Training

* + 1. **Type of Activity**

1. Review

2. Test

3. Inspection

4. Audit

* + 1. **Stage Injected / Stage Detected**

<blank>

1-Initiation

2-Definition

3-Solution

4-Construction

5-Transition

6-Termination

Definition&Solution

Initiation&Definition

Release 1

Release 2

Release 3

Release 4

Release 5

Release 6

Solution&Construction

* + 1. **Priority**

<blank>

1-Immediately

2-High

3-Normal

4-Low

* + 1. **Work product**

Acceptance note  
Architectural design  
Audit program  
Audit record  
Audit report  
Baseline Audit  
Baseline report  
CM Plan  
Change request  
Coding convention  
Configuration Status Report  
Contract  
Customer Satisfaction Survey  
DAR Report  
DP Plan  
DP report  
Database  
Deployment package  
Design prototype  
Detailed design  
Final inspection  
Handover note  
IP  
IP database  
Installation manual  
Integration test case  
Integration test plan  
Integration test report  
Meeting minutes  
Others  
PCB  
PQA report  
Pilot plan  
Pilot record  
Pilot report  
Plan  
Process assets  
Process database  
Program  
Project assests  
Project database  
Project plan  
Project record  
Project report  
Proposal  
Prototype  
QA report  
QA review record  
QDS  
Quality control report  
Quality gate record  
Record  
Release note  
Report  
Requirement prototype  
Resource and environment  
Review TestCase  
Review record  
SQA report  
SRS  
Service Level Agreement  
Software module  
Software package  
Support Diary  
System description  
System test case  
System test plan  
System test report  
Test data  
Test plan  
Test script  
Traceability matrix  
Training course  
Training material  
Training records  
URD  
Unit test case  
Unit test plan  
Unit test report  
Use case  
User manual and WO

* + 1. **Severity**

1-Fatal

2-Serious

3-Medium

4-Cosmetic

* + 1. **Module / Module Code**

<Blank>

<List of module which got from another Project management tool>

* + 1. **Type**

01-Functionality (Other)  
011-Req misunderstanding  
012-Feature missing  
013-Coding logic  
014-Business logic  
02-User Interface  
03-Performance  
04-Design issue  
05-Coding standard  
06-Document  
07-Data & Database integrity  
08-Security & Access Control  
09-Portability  
10-Other  
11-Tools

* + 1. **Defect owner, Assigned To,**

All member of project are ordered by alphabet :

<Blank>

<Member 1>

…..

<Member n>

* + 1. **Status**

1-Error

2-Assigned

3-Pending

4-Tested

5-Accepted

6-Canceled

* 1. **Timeout requirement**

All screens have timeout of 1 hour

After timeout, user has to re-login, and does his/her previous step by step once again.

* 1. **Message list**

|  |  |
| --- | --- |
| **Message Code** | **English Content** |
| ERR\_FORMAT\_DATE\_CREATED | Created Date is invalid |
| ERR\_FORMAT\_DATE\_CREATED\_FROM | "From Created Date" is invalid. |
| ERR\_FORMAT\_DATE\_CREATED\_TO | "To Created Date" is invalid. |
| ERR\_FORMAT\_DATE\_FIXED | "Fixed Date" is invalid. |
| ERR\_FORMAT\_DATE\_FIXED\_FROM | "From Fixed Date" is invalid. |
| ERR\_FORMAT\_DATE\_FIXED\_TO | "To Fixed Date" is invalid. |
| ERR\_FORMAT\_DATE\_DUE | "Deadline" is invalid. |
| ERR\_REQUIRED\_SEVERITY | "Severity" is required |
| ERR\_REQUIRED\_TYPE | "Type" is required |
| ERR\_REQUIRED\_TYPE\_ACTIVITY | "Type of Activity" is required |
| ERR\_REQUIRED\_QC\_ACTIVITY | "QC Activity" is required |
| ERR\_REQUIRED\_ORIGIN | "Defect Origin" is required |
| ERR\_REQUIRED\_WP | "Work Product" is required |
| ERR\_REQUIRED\_CREATED\_DATE | "Created Date" is required |
| ERR\_REQUIRED\_TITLE | "Title" is required |
| ERR\_REQUIRED\_DESCRIPTION | "Description" is required |
| ERR\_REQUIRED\_STATUS | Ignore this message because status cannot be blank |
| ERR\_REQUIRED\_DEFECT\_ORIGIN | Ignore this message because defect origin cannot be blank |
| ERR\_DEFECT\_ID\_INVALID | Defect ID must be positive number |
| ERR\_REQUIRED\_ATTACHMENT\_FILE | Please select at least one file to attach |
| ERR\_FILE\_OVER\_AVAILABLE\_SIZE | Attachment size exceeds available space |
| ERR\_FILE\_OVER\_MAX\_FILE\_SIZE | Attach size exceeded |
| ERR\_EXISTED\_FILE\_NAME | The attached file already exists! |
| ERR\_CONSTRAINT\_DATE\_CREATED\_FROMTO | "From Created Date" cannot be later than "To Created Date" |
| ERR\_CONSTRAINT\_DATE\_FIXED\_FROMTO | "From Fixed Date" cannot be later than "To Fixed Date" |
| ERR\_CONSTRAINT\_DATE\_CREATED\_TODAY | "Created Date" cannot be later than today |
| ERR\_CONSTRAINT\_DATE\_CREATED\_DUE | "Created Date" cannot be later than "Deadline" |
| ERR\_CONSTRAINT\_DATE\_CREATED\_FIXED | "Created Date" cannot be later than fixed date |
| ERR\_CONSTRAINT\_DATE\_FIXED\_TODAY | "Fixed Date" cannot be later than today |
| ALT\_PAGE\_INVALID | Your number is out of range |
| ALT\_REQUIRED\_DEFECT | Defect(s) is required |
| ALT\_REQUIRED\_EXPORT\_LIST | Some items must be selected |