|  |
| --- |
| Passbook Management System  for  i.t apparel Limited |
| **Functional Specification**  Prepared by Global Technology Integrator Limited |
| Ref. No.:  Version: 0.1  Date: 27 July 2015 |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Issue Date** | **Update By** | **Sections, Pages and Text Revised** |
| 0.1 | 03 August 2015 | Jeffrey Ho |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Revision History 2](#_Toc426447856)

**[1](#_Toc426447857)****[Introduction](#_Toc426447857)** [6](#_Toc426447857)

**[2](#_Toc426447858)****[Functional Charts](#_Toc426447858)** [6](#_Toc426447858)

[2.1 Overview of PMS 6](#_Toc426447859)

[2.2 User Catalogue 7](#_Toc426447860)

[2.3 User Requirements 8](#_Toc426447861)

[2.5 Summary of Function List 8](#_Toc426447862)

[2.6 System Menu of PMS 8](#_Toc426447863)

**[3](#_Toc426447864)****[Function Description for Pass Management System](#_Toc426447864)** [10](#_Toc426447864)

[3.1 Home 10](#_Toc426447865)

[3.1.1 Login Screen 10](#_Toc426447866)

[3.1.2 Forget Password 12](#_Toc426447867)

[3.1.3 Reset Password 14](#_Toc426447868)

[3.1.4 Change Password 16](#_Toc426447869)

[3.1.5 PMS Home Screen 18](#_Toc426447870)

[3.2 Template 19](#_Toc426447871)

[3.2.1 Template Management - Search 19](#_Toc426447872)

[3.2.2 Template Management – Create 22](#_Toc426447873)

[3.2.3 Template Management – Update 26](#_Toc426447874)

[3.3 Pass Management 30](#_Toc426447875)

[3.3.1 Pass Management - Search 30](#_Toc426447876)

[3.3.2 Pass Management - Create 33](#_Toc426447877)

[3.4 Promotion 37](#_Toc426447878)

[3.4.1 Promotion Management - Search 37](#_Toc426447879)

[3.4.2 Promotion Management - Create 40](#_Toc426447880)

[3.4.3 Promotion Management - Update 43](#_Toc426447881)

[3.5 Location Based Service (LBS) 46](#_Toc426447882)

[3.5.1 LBS Management - Search 46](#_Toc426447883)

[3.5.2 LBS Management - Create 49](#_Toc426447884)

[3.5.3 LBS Management - Update 54](#_Toc426447885)

[3.6 Administrator 59](#_Toc426447886)

[3.6.1 User Management 59](#_Toc426447887)

[3.6.2 User Management – Create 61](#_Toc426447888)

[3.6.3 User Management – Update 63](#_Toc426447889)

[3.6.4 Role Management 65](#_Toc426447890)

[3.6.5 Role Management – Create 67](#_Toc426447891)

[3.6.6 User Management – Update 69](#_Toc426447892)

[3.6.7 Email Template Management 71](#_Toc426447893)

[3.6.8 Email Template Management – Create 73](#_Toc426447894)

[3.6.9 Email Template Management – Update 75](#_Toc426447895)

[3.6.10 Landing Page Management 77](#_Toc426447896)

[3.7 Landing Page 78](#_Toc426447897)

[3.7.1 Landing Page 78](#_Toc426447898)

**[4](#_Toc426447899)****[Web Services](#_Toc426447899)** [81](#_Toc426447899)

[4.1 Create Pass Web Service for Android Device 81](#_Toc426447900)

[4.2 Create Pass Web Service for Apple Device 83](#_Toc426447901)

[4.3 Sign Pass Web Service for Apple Device 85](#_Toc426447902)

[4.4 Push Notification Web Services 86](#_Toc426447903)

[4.5 Apple Passbook Server 87](#_Toc426447904)

[4.5.1 Apple Device Register Web Service 87](#_Toc426447905)

[4.5.2 Pass Update Web Services for Apple Device 89](#_Toc426447906)

[4.5.3 Pass Delete Web Services for Apple Device 91](#_Toc426447907)

**[5](#_Toc426447908)****[Batch Jobs](#_Toc426447908)** [92](#_Toc426447908)

[5.1 Update Bonus Point of Member Cards 92](#_Toc426447909)

[5.2 Update Spending of Primary Member Cards 93](#_Toc426447910)

[5.3 Email Notification (For Model Withdraw) 94](#_Toc426447911)

[5.4 Update Pass by General / Personal Promotion 95](#_Toc426447912)

[5.5 Update Pass for LBS Information 96](#_Toc426447913)

[5.6 Update Pass for Model Update 97](#_Toc426447914)

**[Appendix A – Database Schema](#_Toc426447915)** [98](#_Toc426447915)

**[Appendix B – Report Template](#_Toc426447916)** [99](#_Toc426447916)

1. **Introduction**

The I.T Passbook Management System (PMS) is developed for replacing the subscription of the service provided by 3rd party which is used for pass management in I.T.  The new system includes online functions for back office users to manage both membership and coupon passes usage and creation.  There are some schedule jobs built for updating the existing passes for facilitating normal operation and promotion purpose.

In addition, an online landing page for public customers is also provided in this system for achieving new passes.

1. **Functional Charts**

## 2.1 Overview of PMS

The functional chart for PMS are as follow:

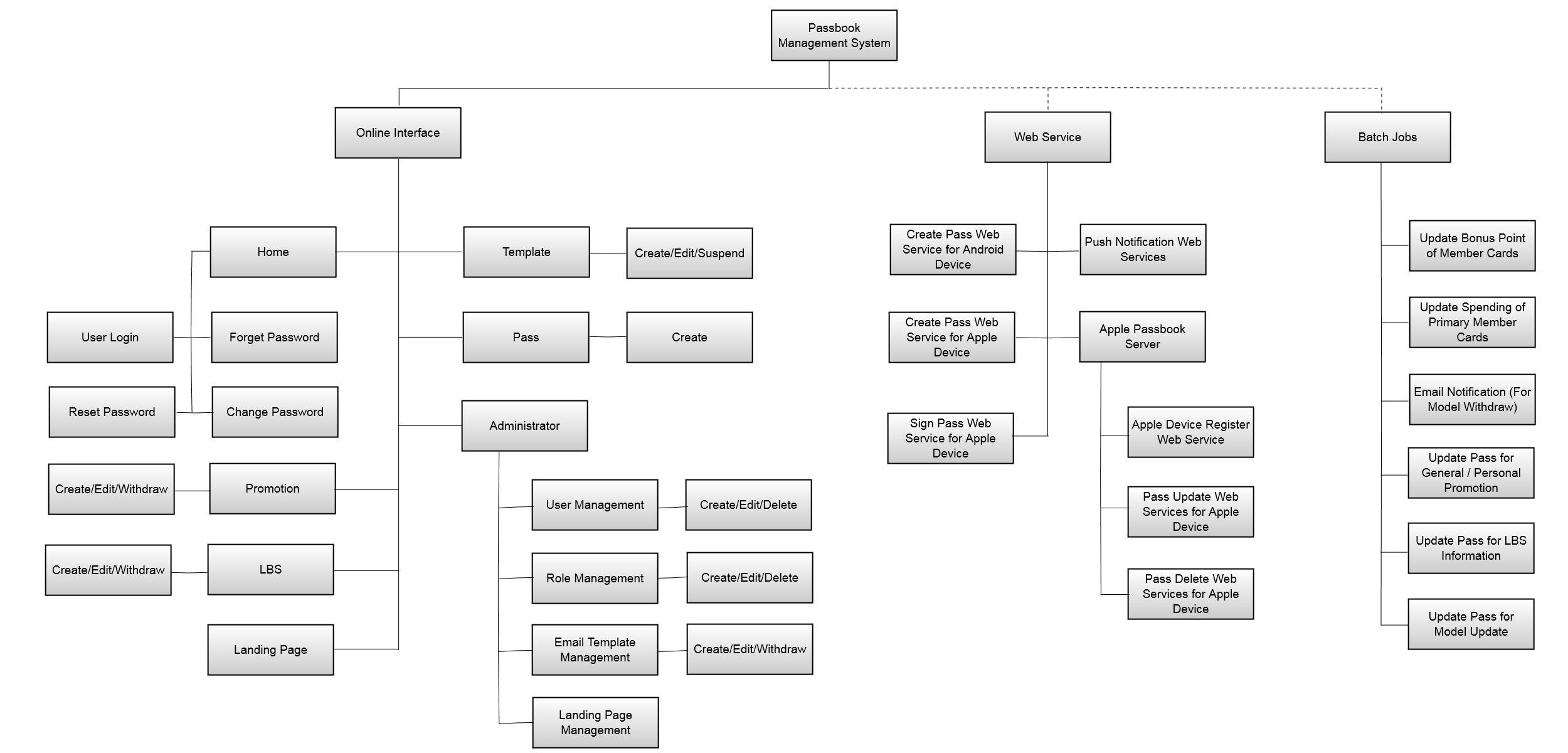


Figure 2.1

The system logical infrastructure for PMS are as follow:

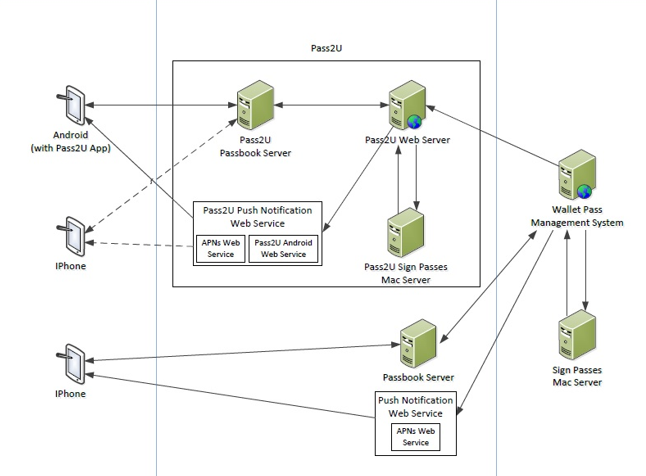


Figure 2.2

## 2.2 User Catalogue

## 2.3 User Requirements

* I.T Passbook Management System
  + Model Management (Template)
  + Passes Management
  + Passes Quota Release
  + Admin Management
* HTML Landing Page
* Web Services for Pass Creation
* Other backend batch jobs / services

**2.4 System Requirements**

## 2.5 Summary of Function List

**Summary of the System Requirements (PMS)**

**Summary of the System Requirements (Web Services)**

**Summary of the System Requirements (Batch Jobs)**

## 2.6 System Menu of PMS

* Template Management
* Pass Management
* Promotion Management
* LBS Management
* Administrator
  + User Management
  + Role Management
  + Email Template Management
  + Default Landing Page Management

1. **Function Description for Pass Management System**

## 3.1 Home

### 3.1.1 Login Screen

|  |  |
| --- | --- |
| **Function ID:** | FN-HME-001 |
| **Function Name:** | Login Screen |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface for users to login.

**Function Screen**

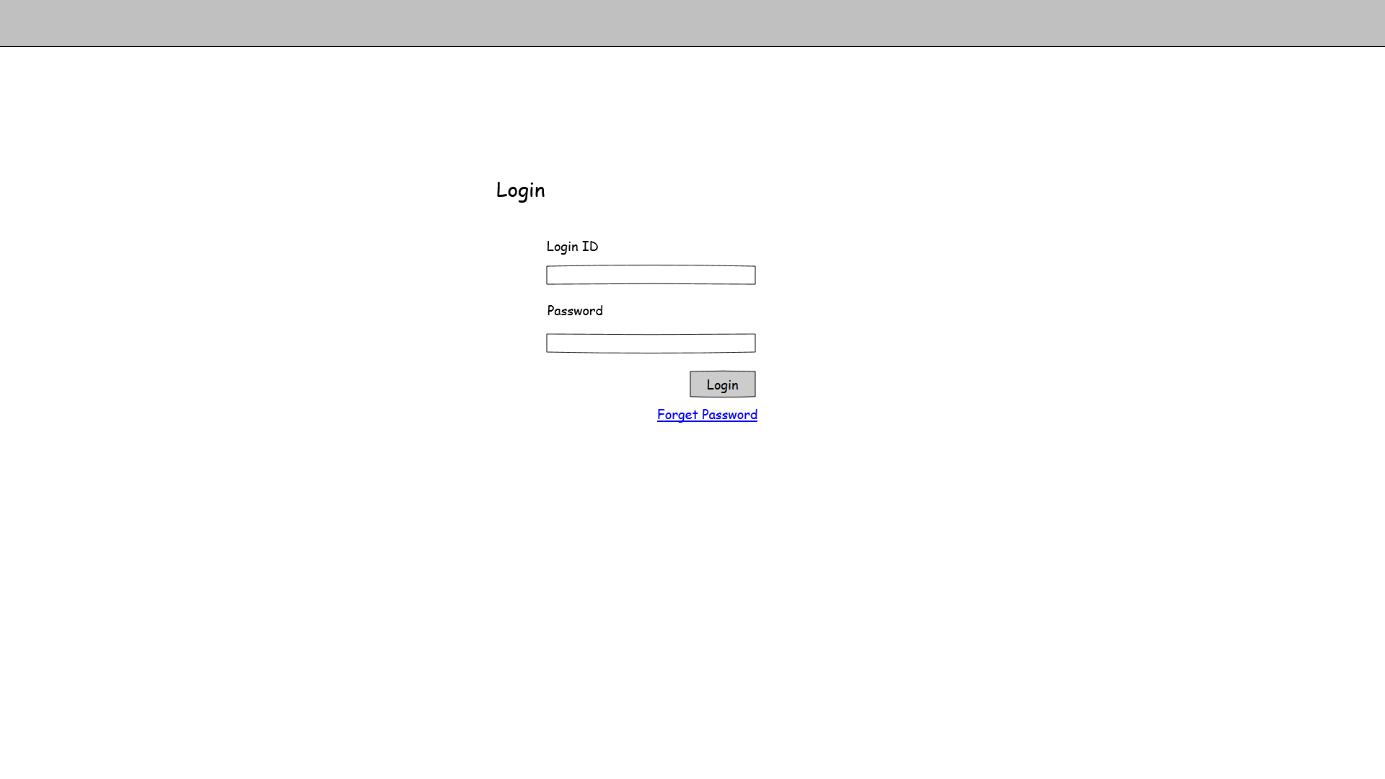


Figure 3.1.1 PMS-HME-001

**Function Details**

Main elements/actions of the graphical user interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| Login ID | Textbox | VARCHAR(30) | Y |  |
| Password | Textbox | VARCHAR(15) | Y | Should be 8-15 characters and case sensitive |
| Login | Button | - | - | User login and go to home screen PMS-HME-006 |
| Forget Password | Link button | - | - | Go to Screen PMS-HME-002 |

**Flow and Actions**

1. User input login ID and password

2. System checks login ID and password. If correct, login user and go to home page PMS-HME-006.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Password should be 8 -15 characters and case sensitive. If not, show error message “Invalid password”.

3. Login ID and password both exist and correct in user table. If not, show error message “Invalid login ID and password”.

4. If user is inactive, show error message “User account is inactive, please connect system administrator.”

### 3.1.2 Forget Password

|  |  |
| --- | --- |
| **Function ID:** | FN-HME-002 |
| **Function Name:** | Forget Password |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface for users to send password through email.

**Function Screen**

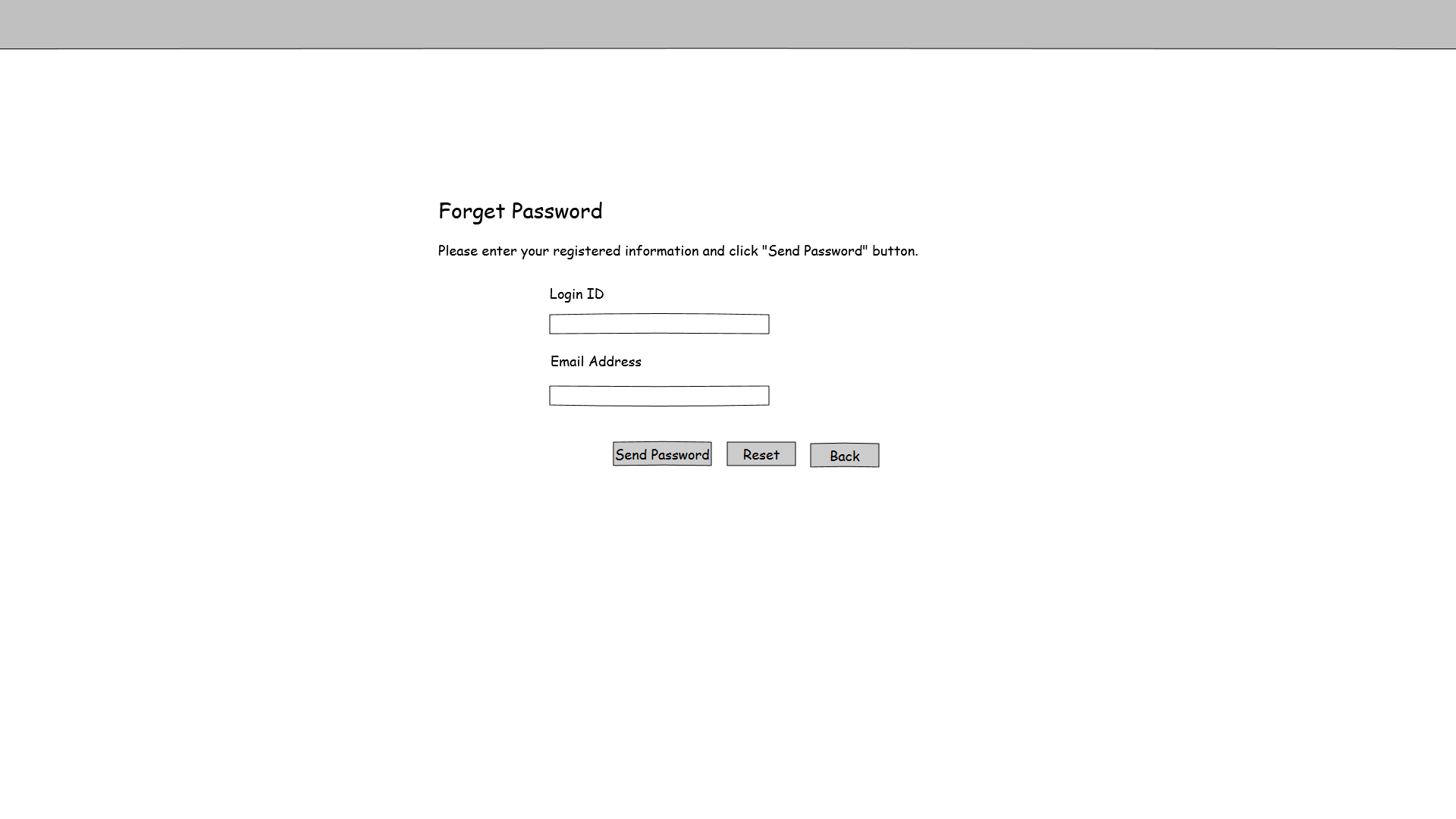


Figure 3.1.2a PMS-HMS-002



Figure 3.1.2b PMS-HME-003

**Function Details**

Main elements/actions of the graphical user interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| Login ID | Textbox | VARCHAR(30) | Y |  |
| Email Address | Textbox | VARCHAR(250) | Y | Check email format |
| Send Password | Button | - | - | Send email to user. Then go to screen PMS-HME-003 |
| Reset | Button | - | - | Clear all input fields |
| Back | Button | - | - | Back to Screen PMS-HME-001 |
| Home | Button | - | - | Go to Screen PMS-HME-001 |

**Flow and Actions**

1. User input login ID and email address

2. System checks login ID and email address. If correct, send a new password to user and go to confirm page PMS-HME-003.

3. Click Home button and go to login screen PMS-HME-001

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Check email format. Email address must have character “@”. Otherwise, show error message “Invalid email address”.

3. Login ID and email address both exist and correct in user table. If not, show error message “Invalid login ID and email address”.

### 3.1.3 Reset Password

|  |  |
| --- | --- |
| **Function ID:** | FN-HME-003 |
| **Function Name:** | Reset Password |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface for users to reset password.

**Function Screen**

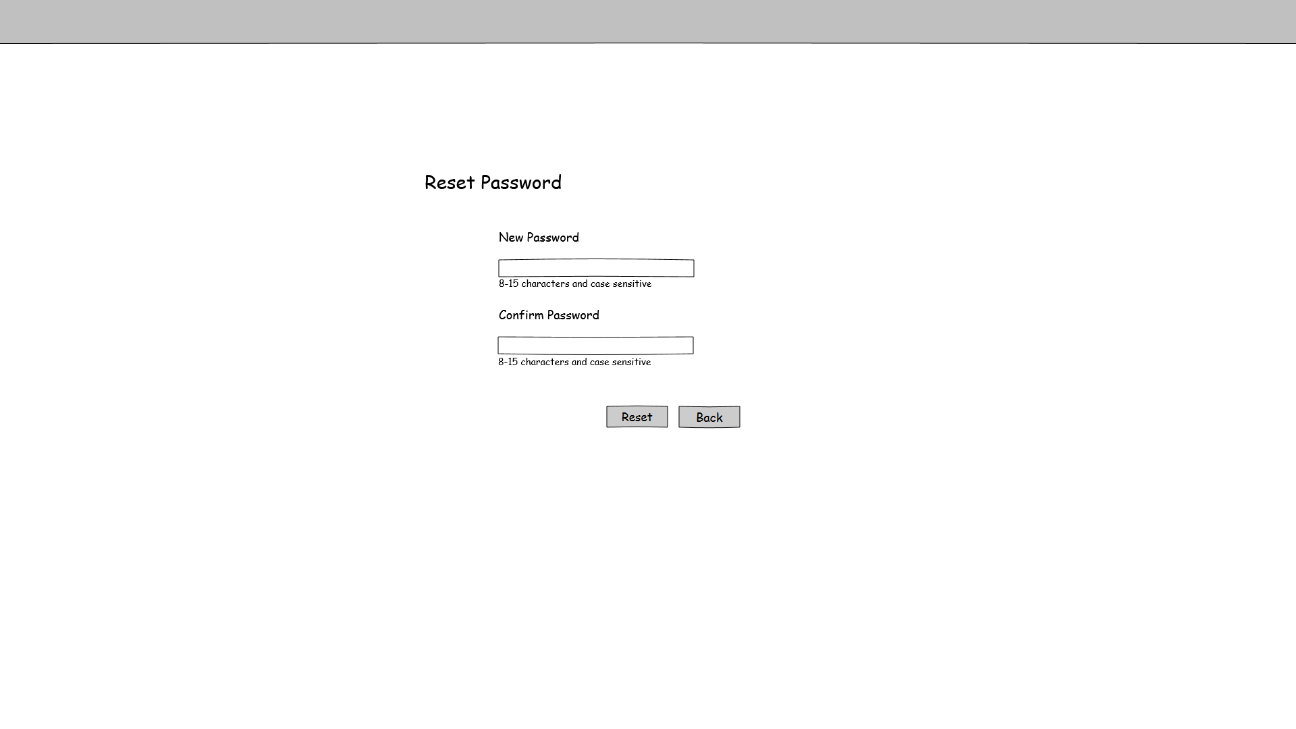


Figure 3.1.3 PMS-HME-003

**Function Details**

Main elements/actions of the graphical user interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| New Password | Textbox | VARCHAR(15) | Y | 8-15 characters, case sensitive |
| Confirm Password | Textbox | VARCHAR(15) | Y | 8-15 characters, case sensitive |
| Reset | Button | - | - | Reset password and login, then go to home screen PMS-HME-005 |
| Cancel | Button | - | - | Go to Screen PMS-HME-001 |

**Flow and Actions**

1. User input new password and confirm password

2. System checks new password and confirm password. If correct, system will save password to user table. Then login user and go to home page PMS-HME-005.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. New password and confirm password should be 8 -15 characters and case sensitive. If not, show error message “Invalid password”.

3. New password and confirm password must be the same. If they are different, show error message “Confirm Password is different from New Password. Please input password again.”

### 3.1.4 Change Password

|  |  |
| --- | --- |
| **Function ID:** | FN-HME-004 |
| **Function Name:** | Change Password |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface for users to change password.

**Function Screen**

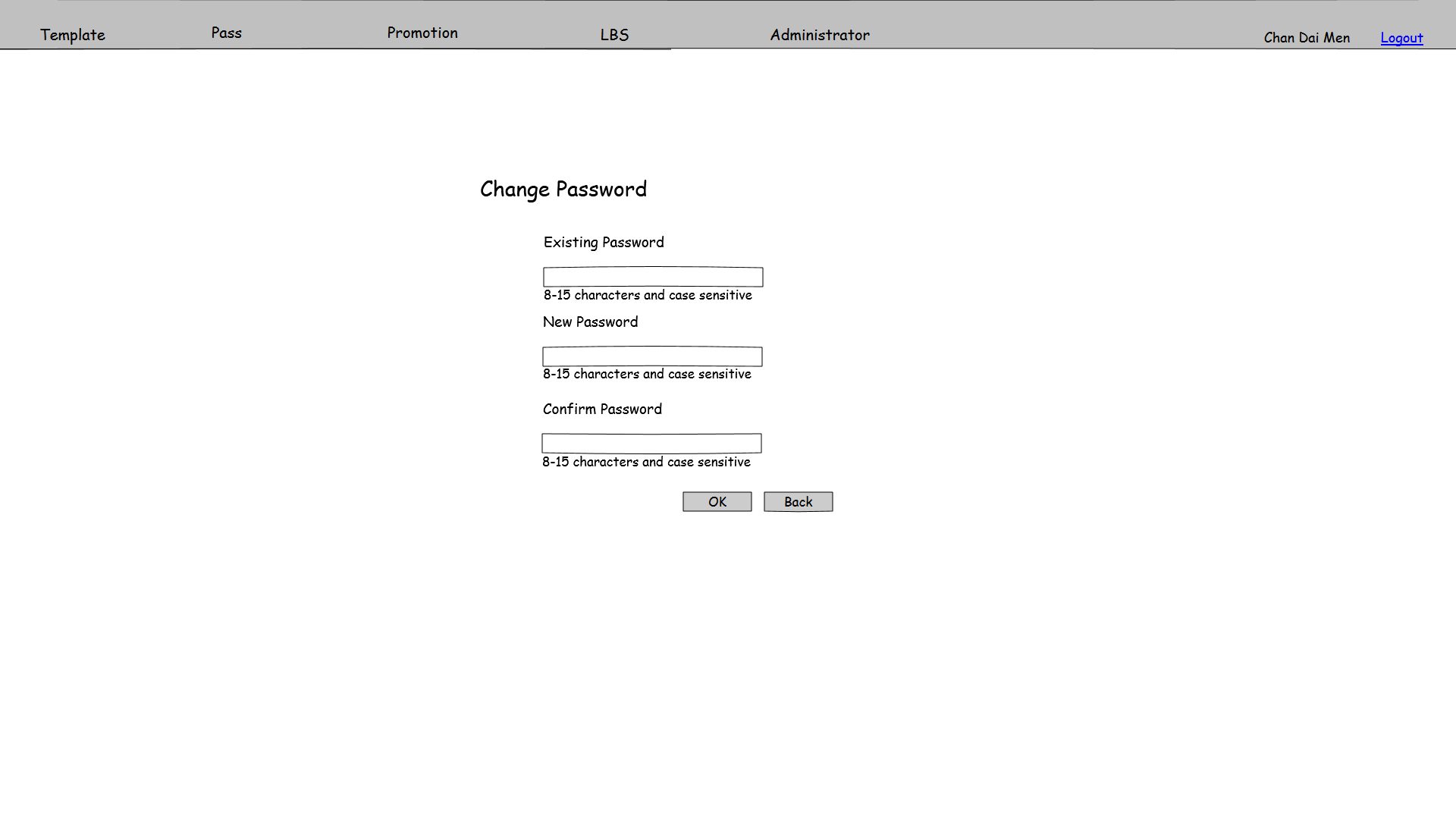


Figure 3.1.4 PMS-HME-004

**Function Details**

Main elements/actions of the graphical user interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| Existing Password | Textbox | VARCHAR(15) | Y | 8-15 characters, case sensitive |
| New Password | Textbox | VARCHAR(15) | Y | 8-15 characters, case sensitive |
| Confirm Password | Textbox | VARCHAR(15) | Y | 8-15 characters, case sensitive |
| OK | Button | - | - | Change password and go to login screen PMS-HME-001 |
| Back | Button | - | - | Back to Screen PMS-HME-005 |

**Flow and Actions**

1. User input existing password, new password and confirm password

2. When user click “OK” button, system checks existing password, if correct then check new password and confirm password. If correct, system will save password for user. Then go to login page PMS-HME-001.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Existing password, new password and confirm password should be 8 -15 characters and case sensitive. If not, show error message “Invalid password”.

3. Existing password must match the password of login user. If not match, show error message “Please input the correct existing password”

4. New password and confirm password must be the same. If they are different, show error message “Confirm Password is different from New Password. Please input password again.”

### 3.1.5 PMS Home Screen

|  |  |
| --- | --- |
| **Function ID:** | FN-HME-005 |
| **Function Name:** | PMS Home Screen |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface of home screen. The system will show different menu items for different user role. The function screen shows all menu items.

In the Home screen, it will show the summary of pass issued for different member type. And clicking on the hyperlink of each records can popup an Excel file with the passes issued to corresponding member type.

**Function Screen**

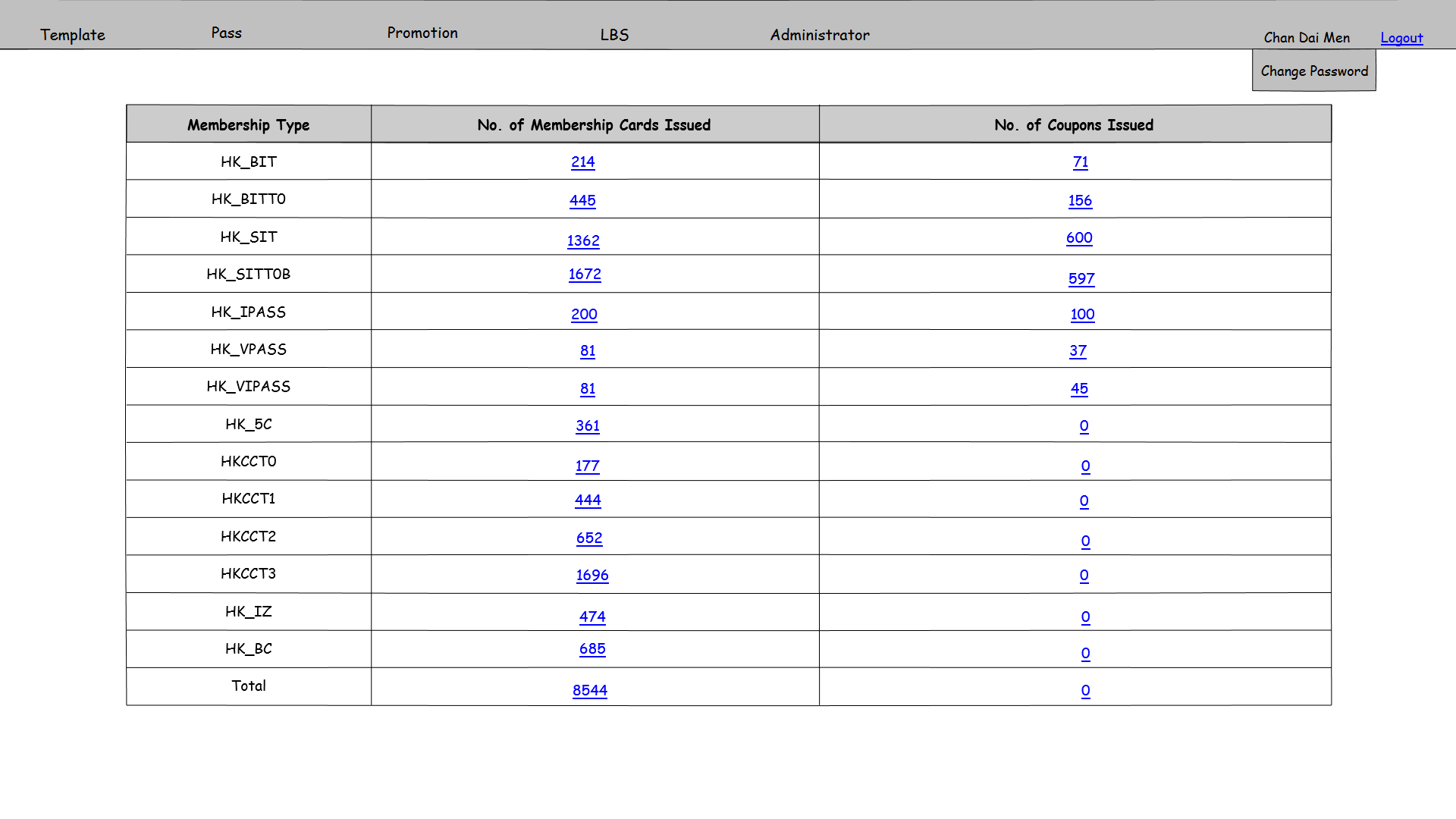


Figure 3.1.4 PMS-HME-004

**Function Details**

Main elements/actions of the graphical user interface.

**Flow and Actions**

1. Select the items listed on the menu to access corresponding functions.

2. Click “Change Password” and go to screen PMS-HME-004 Change Password.

**Form Validation**

N/A

## 3.2 Template

### 3.2.1 Template Management - Search

|  |  |
| --- | --- |
| **Function ID:** | FN-TEM-001 |
| **Function Name:** | Template management - Search |
| **Target User:** | All User |

**Function Description**

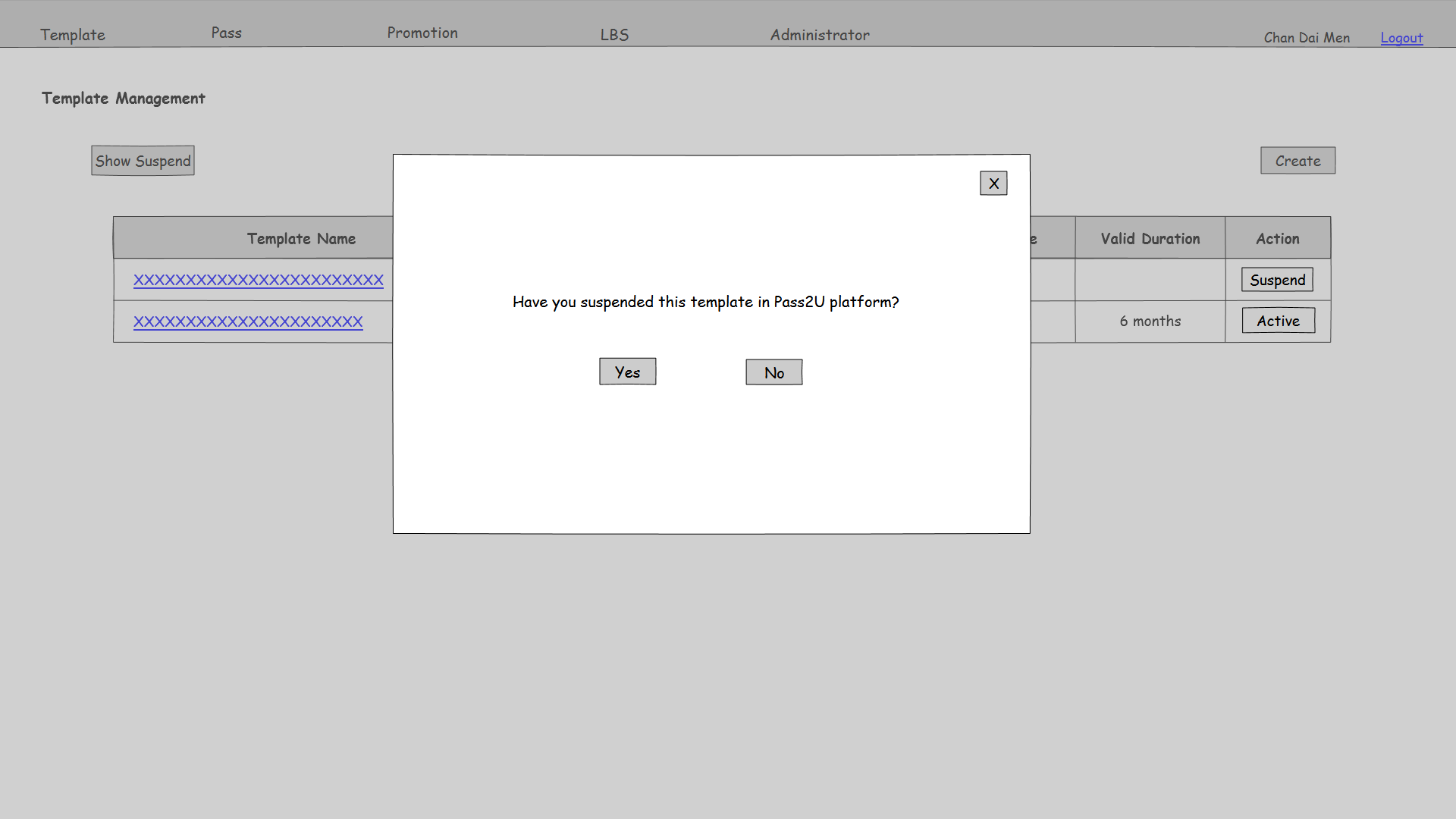
This function provides an interface to manage the pass templates. Users can create, update and suspend the templates in this screen.

**Function Screen**

Figure 3.2.1a PMS-TEM-001



Figure 3.2.1b PMS-TEM-001



**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Show Suspend | Button | - | - | Show All Template, including suspended, in the table. And then change the text to “Hidden Suspend” |
| Create | Button | - | - | Button for users to click on it and go to template create page PMS-TEM-002 |
| Template Name | Link button | - | - | Hyper link for users to click on it and go to template edit page PMS-TEM-003 |
| Template Type | Display | - | - |  |
| Issued Start Date | Display | - | - | Format is dd/mm/yyyy |
| Issued End Date | Display | - | - | Format is dd/mm/yyyy |
| Valid Duration | Display | - | - |  |
| / | Button | - | - | Prompt a confirm dialog as Figure 3.2.1b. If click “Yes”, update suspend/active status of the corresponding record |

**Flow and Actions**

1. User clicks button to suspend the template. A confirmation window will prompt out as Figure 3.2.1b shown. After clicking “Yes”, the template will be suspend and hidden and this button will change to “Active”. When Clicking “Active”, the template will be re-activated.



2. User can click “Show Suspend” button to show all template records in the table. After that, this button will change to “Hide Suspend”.

**Form Validation**

N/A

### 3.2.2 Template Management – Create

|  |  |
| --- | --- |
| **Function ID:** | FN-TEM-002 |
| **Function Name:** | Template Management – Create |
| **Target User:** | All Users |

**Function Description**

This function provides an interface to create a new template.

**Function Screen**

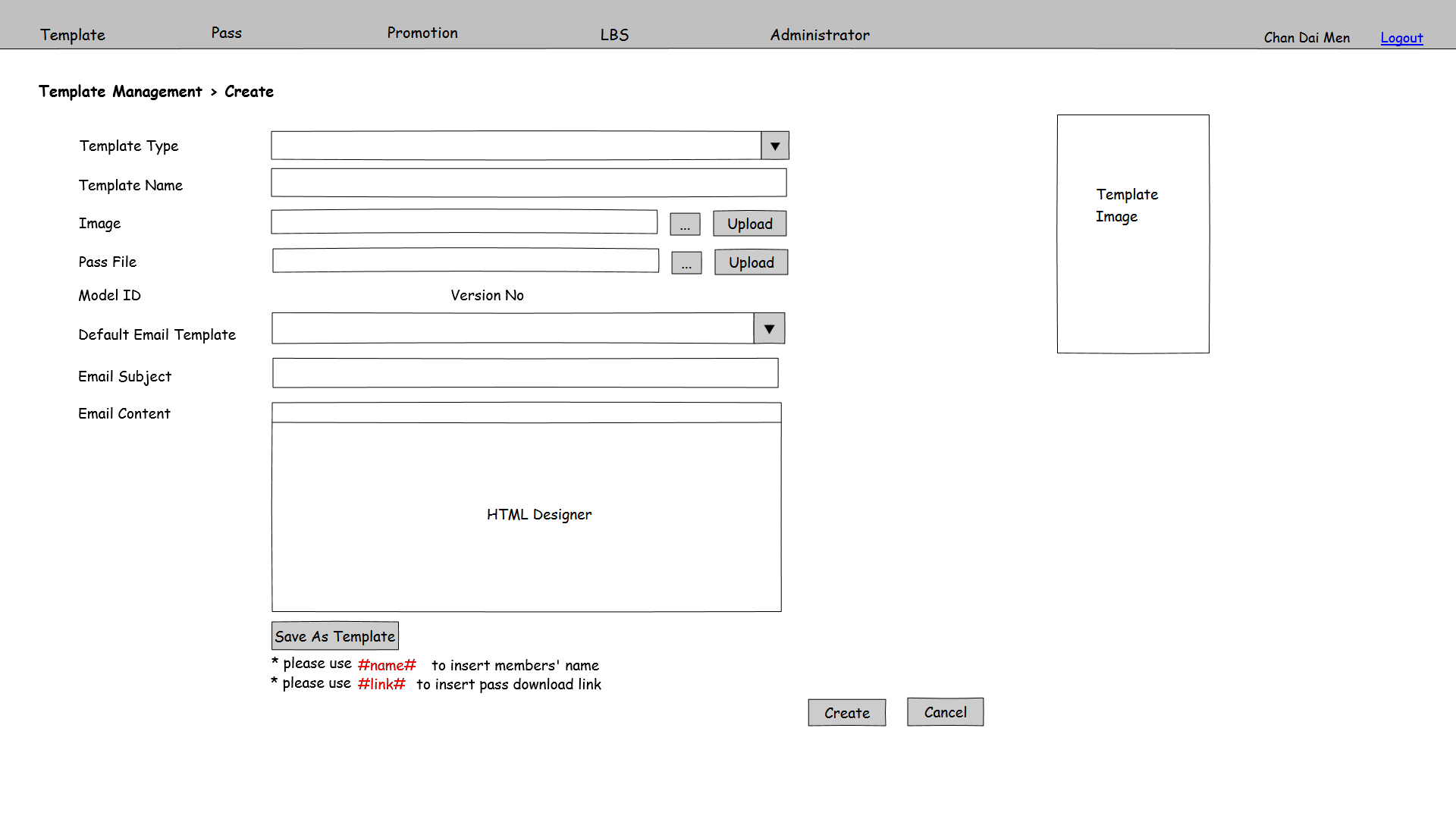


Figure 3.2.2a PMS-TEM-002

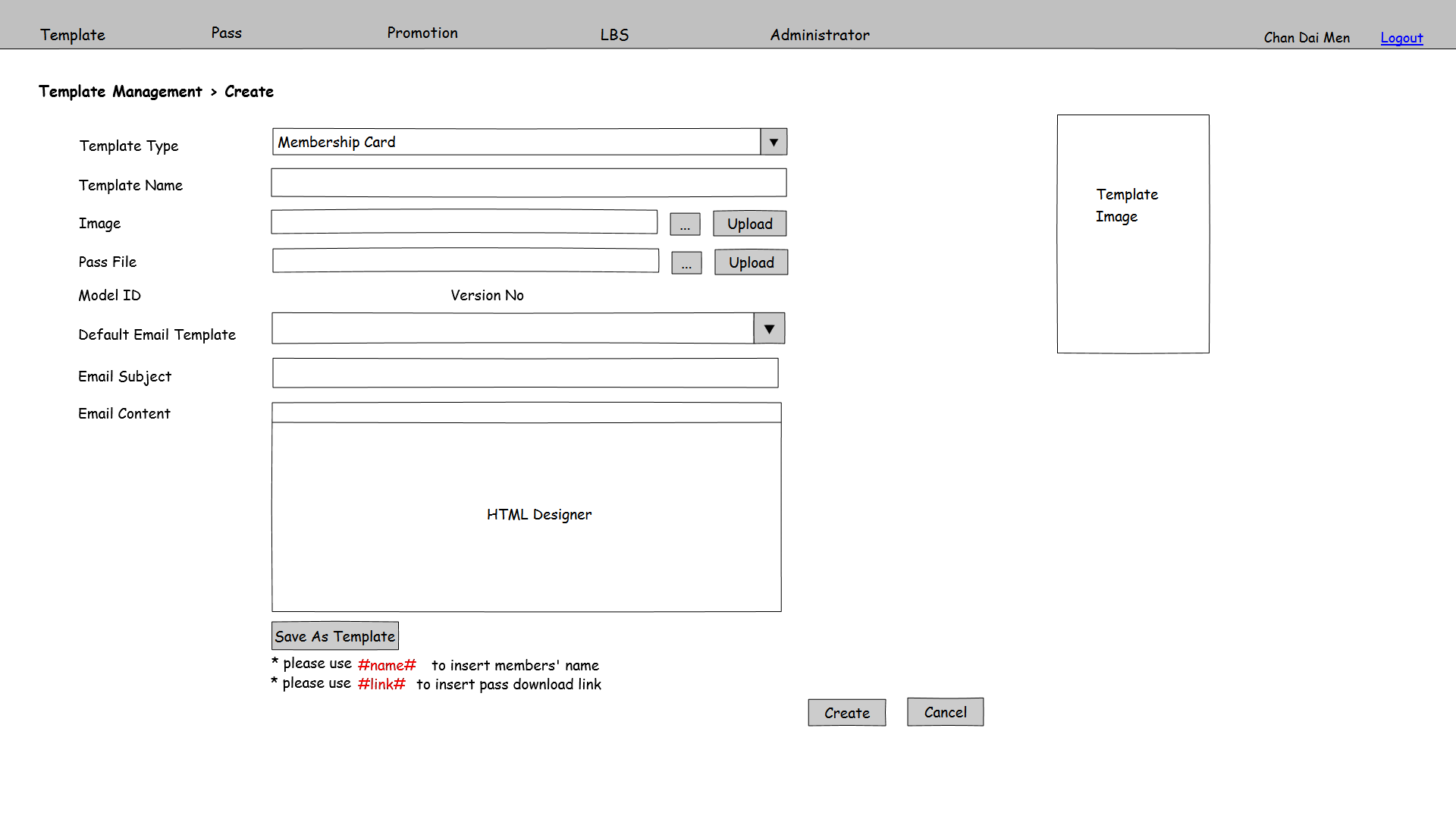


Figure 3.2.2b PMS-TEM-002

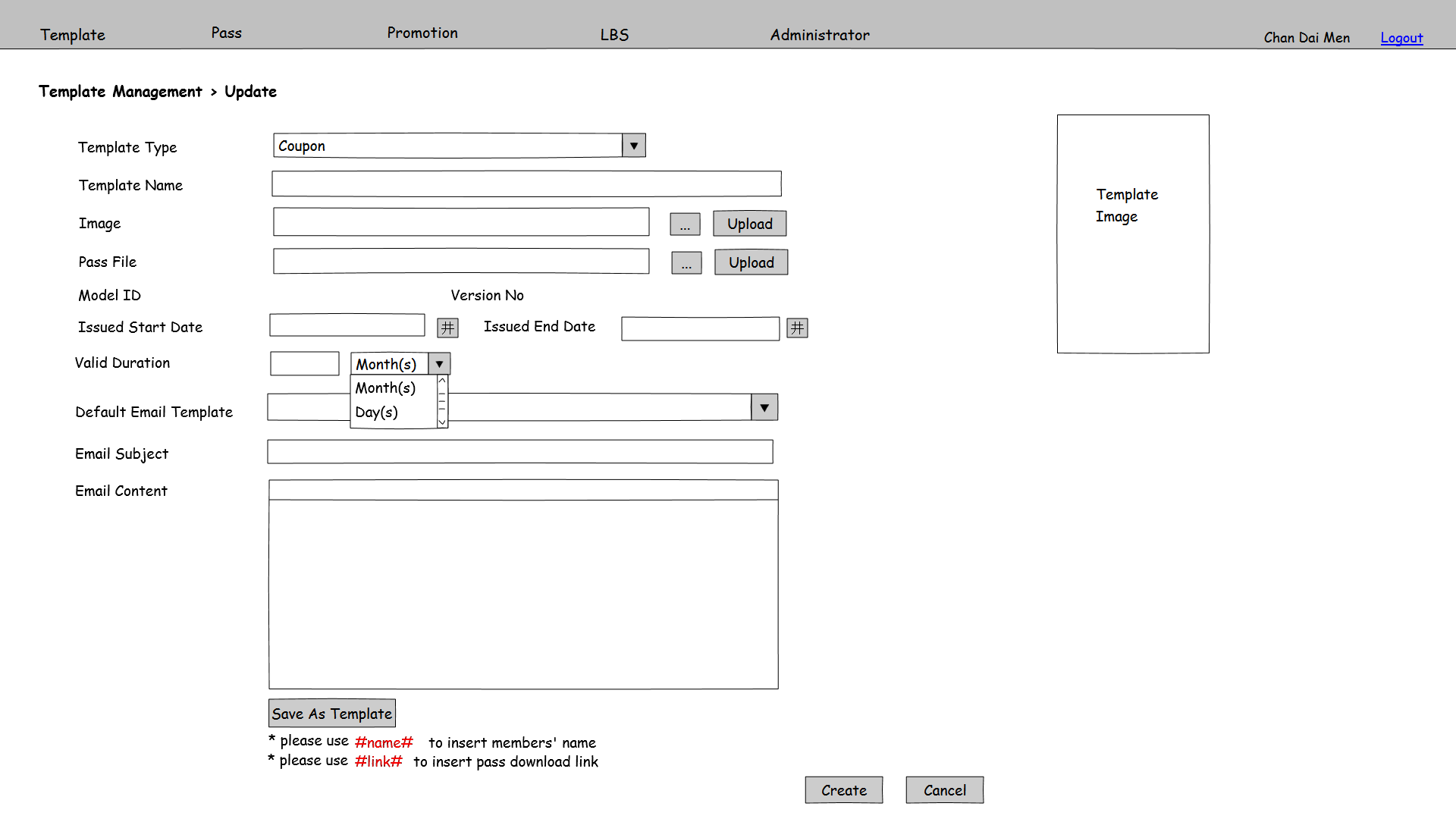


Figure 3.2.2c PMS-TEM-002

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Template Type | Pull down menu | - | Y | Selection list including “Member Card” and “Coupon” |
| Template Name | Textbox | NVARCHAR(30) | Y |  |
| Image | Textbox | VARCHAR(200) | Y |  |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Pkpass file | Textbox | VARCHAR(200) | Y |  |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Model ID | Display | - | - | Automatic show the pass model ID when pkpass file is uploaded |
| Version No | DIsplay | - | - | Automatic show the pass version no when pkpass file is uploaded |
| Issued Start Date | Textbox | Date time | Y | Enabled when template type is “Coupon” |
| Issued End Date | Textbox | Date time | Y | Enabled when template type is “Coupon” |
| Valid Duration | Textbox | Integer | Y | Enabled when template type is “Coupon” |
| Default Email Template | Pull down menu | - | N | Selection list gets from email template table. |
| Email Subject | Textbox | NVARCHAR(200) | Y | Automatic fill in content when Default Email Template is selected. |
| Email Content | HTML Designer | NVARCHAR(MAX) | Y | Use #name# to insert members’ name, user #link# to insert pass download link  Automatic fill in content when Default Email Template is selected. |
| Save As Template | Button | - | - | Save the inputted parameters as an email template.  Enable when user has Email Template edit function access right. |
| Create | Button | - | - | Save the inputted parameters as a template record. Launch batch job “Update Pass” and go to PMS-TEM-001 |
| Cancel | Button | - | - | Back to PMS-TEM-001 |

**Flow and Actions**

1. Refer to PMS-TEM-001, click on “Create” and enter this functions as figure 3.2.2a shows.
2. Select Template Type.
3. For user select Member Card, the Template Create screen will show as figure 3.2.2b.
4. For user select Coupon, the Template Create screen will show as figure 3.2.2c. The input fields of Issued Start Date, Issued End Date and Valid Duration will be shown.
5. Attach an image of latest update by using the and “Upload” button. The image will show at the Template Image field once the pass image is uploaded.



1. Upload a pass file through and “Upload” button. After pass file is uploaded, the Model ID and Version No fields will show the corresponding information in the uploaded pass file.



1. After “Default Email Template” is selected, the Email Subject and Email Content will be filled by contents defined in Email Template function.
2. After input all mandatory fields, press “Submit” button to save the template record and back to PMS-TEM-001.
3. If user has access right for Email Template Edit, “Save As Template” button will be enabled. User can edit the selected email template and save it. Otherwise, if user hasn’t access right for Email Template Edit, “Save As Template” button will be disabled.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Template Name is unique. Check duplicate on Template Name. If there is duplicate record in database, show error message “Template Name is already exists. Please input other Template Name.”

3. Issued Start Date must smaller or equal to Issued End Date. Otherwise, show error message “Invalid Issued Date. Issued Start Date must be smaller or equal to Issued End Date.”

4. File format used by pass file upload is pkpass. Otherwise, show error message “Invalid file format. File for pass file must be in pkpass format.”

5. File format used by Image upload is correct image format. And maximum file size is 500KB. Otherwise, show error message “Invalid file format. Image file must be in proper image format and size smaller than 500KB.”

6. Email Content must have “#link#” in order to send notification to update the pass. Otherwise, show error message “Missing #link# in Email Content”.

### 3.2.3 Template Management – Update

|  |  |
| --- | --- |
| **Function ID:** | FN-TEM-003 |
| **Function Name:** | Template Management – Update |
| **Target User:** | All Users |

**Function Description**

This function provides an interface to update an existing template.

**Function Screen**

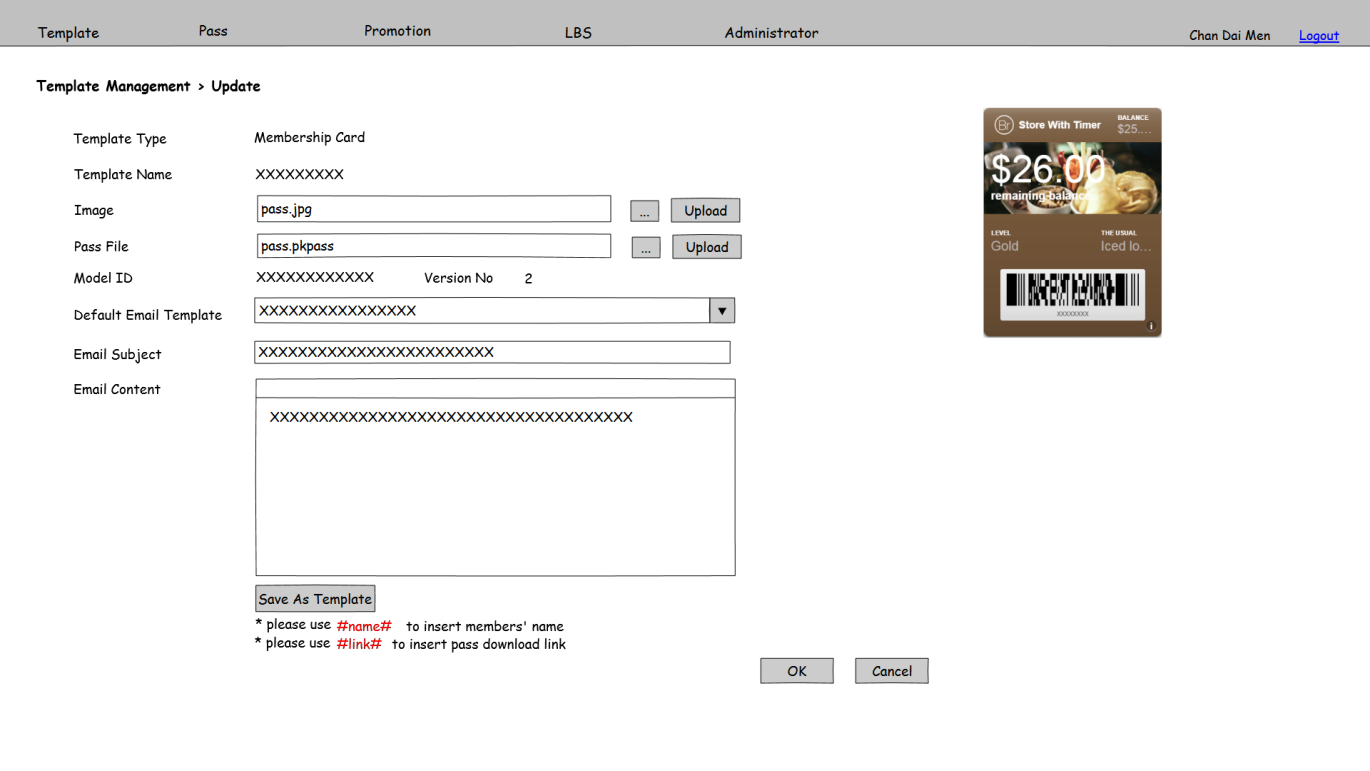


Figure 3.2.3a PMS-TEM-003

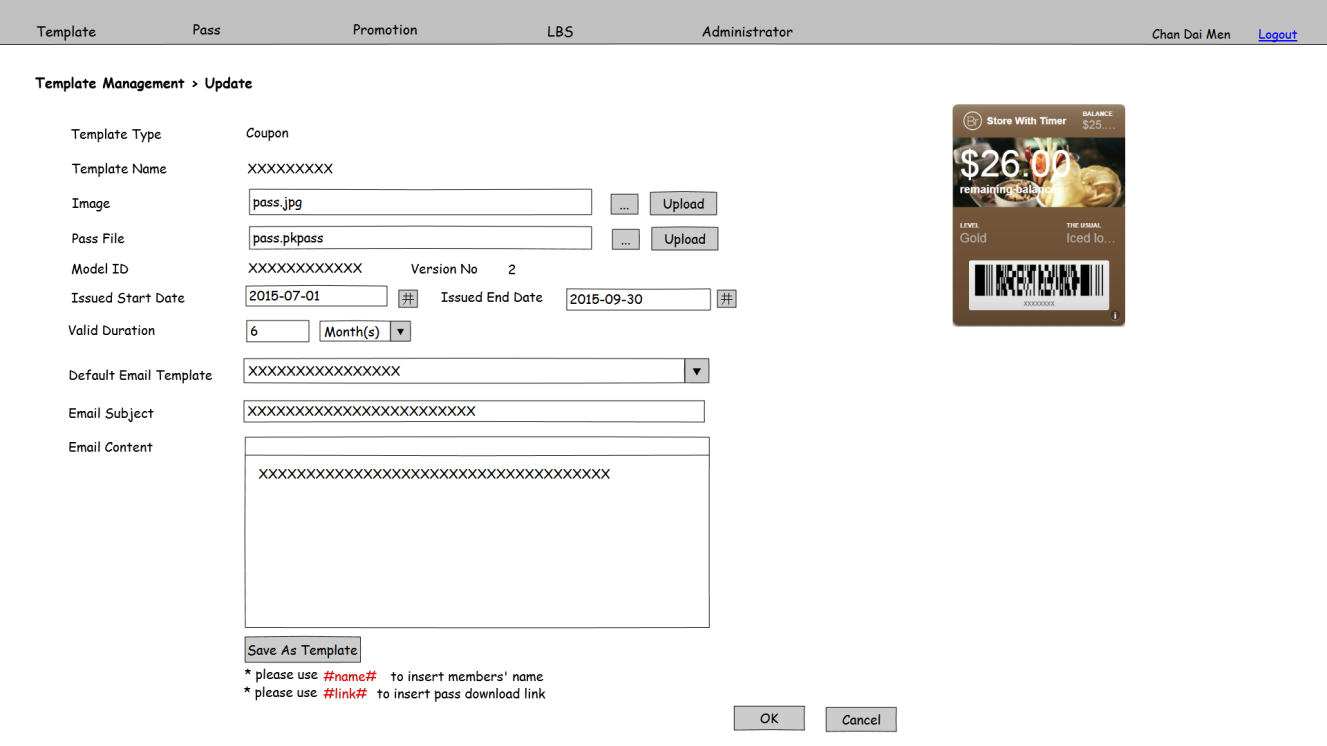


Figure 3.2.3b PMS-TEM-003

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Template Type | Display | - | - |  |
| Template Name | Display | - | - |  |
| Image | Textbox | VARCHAR(200) | Y |  |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Pkpass file | Textbox | VARCHAR(200) | Y |  |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Model ID | Display | - | - | Automatic show the pass model ID when pkpass file is uploaded |
| Version No | DIsplay | - | - | Automatic show the pass version no when pkpass file is uploaded |
| Issued Start Date | Textbox | Date time | Y | Enabled when template type is “Coupon” |
| Issued End Date | Textbox | Date time | Y | Enabled when template type is “Coupon” |
| Valid Duration | Textbox | Integer | Y | Enabled when template type is “Coupon” |
| Default Email Template | Pull down menu | - | - | Selection list gets from email template table. |
| Email Subject | Textbox | NVARCHAR(200) | Y | Automatic fill in content when Default Email Template is selected. |
| Email Content | HTML Designer | NVARCHAR(MAX) | Y | Use #name# to insert members’ name, user #link# to insert pass download link  Automatic fill in content when Default Email Template is selected. |
| Save As Template | Button | - | - | Save the inputted parameters as an email template.  Enable when user has Email Template edit function access right. |
| OK | Button | - | - | Save the inputted parameters as a template record. Launch batch job “Update Pass” and go to PMS-TEM-001 |
| Cancel | Button | - | - | Back to PMS-TEM-001 |

**Flow and Actions**

1. Refer to PMS-TEM-001, click on hyperlink of search result.

2. For Template Type is Member Card, the Template Update screen will show as figure 3.2.3a.

For Template Type is Coupon, the Template Update screen will show as figure 3.2.3b. Issued Start Date, Issued End Date and Valid Duration will show as an input field.

3. Attach an image of latest update by using the and “Upload” button. The image will show at the Template Image field once the pass image is uploaded.



4. Upload a pass file through and “Upload” button. After pass file is uploaded, the Model ID and Version No fields will show the corresponding information in the uploaded pass file.



5. Inputted the parameter fields.

6. After “Default Email Template” is selected, the Email Subject and Email Content will be filled by contents defined in Email Template function.

7. After input all mandatory fields, press “Submit” button to update the template record and back to PMS-TEM-001.

8. If user has access right for Email Template Edit, “Save As Template” button will be enabled. User can edit the selected email template and save it. Otherwise, if user hasn’t access right for Email Template Edit, “Save As Template” button will be disabled.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Issued Start Date must smaller or equal to Issued End Date. Otherwise, show error message “Invalid Issued Date. Issued Start Date must be smaller or equal to Issued End Date.”

3. File format used by pass file upload is pkpass. Otherwise, show error message “Invalid file format. File for pass file must be in pkpass format.”

4. File format used by Image upload is correct image format. And maximum file size is 500KB. Otherwise, show error message “Invalid file format. Image file must be in proper image format and size smaller than 500KB.”

5. Email Content must have “#link#” in order to send notification to update the pass. Otherwise, show error message “Missing #link# in Email Content”.

6. The model ID of uploaded pkpass file must the same as the selected Template. And the version number of uploaded pkpass file should larger or equal to the selected Template. Otherwise, show error message “Invalid pkpass file. Please check model ID or version no. of uploaded pkpass.”

## 3.3 Pass Management

### 3.3.1 Pass Management - Search

|  |  |
| --- | --- |
| **Function ID:** | FN-PAS-001 |
| **Function Name:** | Pass Management - Search |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface for users to manage passes. Users can search and create passes in this screen.

**Function Screen**



Figure 3.3.1 PMS-PAS-001

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Template Name | Textbox | NVARCHAR(30) | N |  |
| Template Type | Pull down menu | - | N | Selection choices including “All”, “Member card”, “Coupon”  Default value is “All” |
| Campaign ID | Textbox | VARCHAR(15) | N |  |
| Template Status | Pull down menu | - | N | Selection choices including “All”, “Active”, “Suspend”  Default value is “All” |
| Target Member List | Textbox | VARCHAR (200) | N | User can input member IDs with comma delimiter. |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Member Email | Textbox | VARCHAR(250) | N | Check email format |
| Member Mobile | Textbox | VARCHAR(15) | N |  |
| Bonus From | Textbox | Integer | N |  |
| Bonus To | Textbox | Integer | N |  |
| Spending From | Textbox | Integer | N |  |
| Spending To | Textbox | Integer | N |  |
| Issued Date From | Textbox | Date time | N | Format is dd/mm/yyyy |
| Issued Date To | Textbox | Date time | N | Format is dd/mm/yyyy |
| Issued By | Textbox | VARCHAR(30) | N |  |
| Create | Button | - | - | Button for users to click on it and go to pass create page PMS-PAS-002 |
| Search | Button | - | - | Search by the input parameters. The result list will order by Template Name |
| Cancel | Button | - | - | Clear all input fields |
| Template Name | Display | - | - |  |
| Template Type | Display | - | - |  |
| Template Status | Display | - | - |  |
| Member ID | Display | - | - |  |
| Email | Display | - | - |  |
| Mobile No | Display | - | - |  |
| Campaign ID | Display | - | - |  |
| Bonus/Spending | Display | - | - |  |
| Issued Date | Display | - | - | Format is dd/mm/yyyy |
| Issued By | Display | - | - |  |

**Flow and Actions**

1. User input search criteria(s) and press search

2. System searches the results and shows the searching result to users. The searching result will order by Template Name, Template Type and Issued Date ascending.

**Form Validation**

1. Issued Date From must smaller or equal to Issued Date To. Otherwise, show error message “Invalid Issued Date. Issued Date From must be smaller or equal to Issued Date To.”

2. File format used by Target Member List upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for Target Member List must be csv (comma delimiter) format.”

### 3.3.2 Pass Management - Create

|  |  |
| --- | --- |
| **Function ID:** | FN-PAS-002 |
| **Function Name:** | Pass Management – Create |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface for users to create passes and send email to the target member list for pass creation.

**Function Screen**



Figure 3.3.2a PMS-PAS-002

Figure 3.3.2b PMS-PAS-002

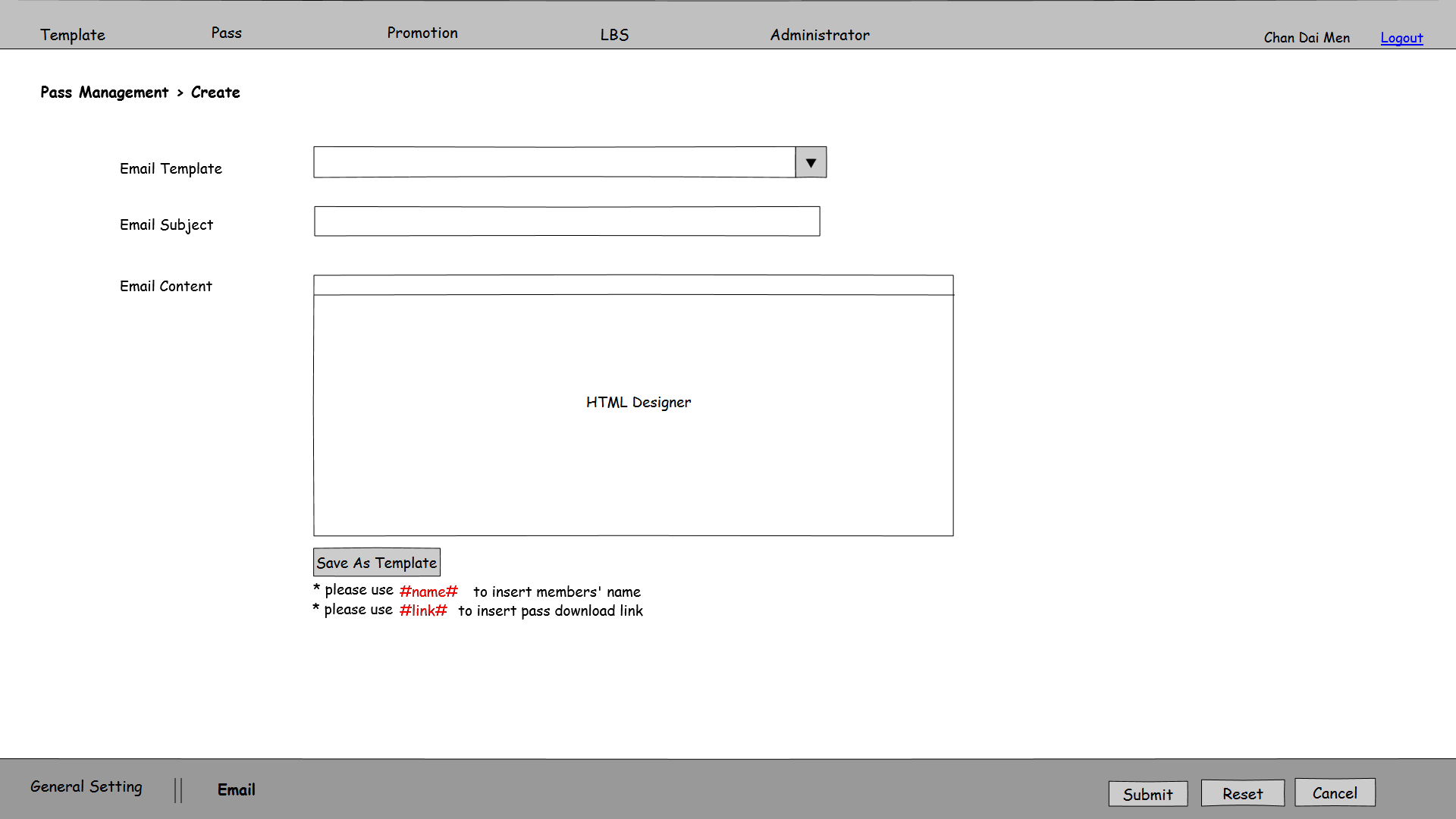
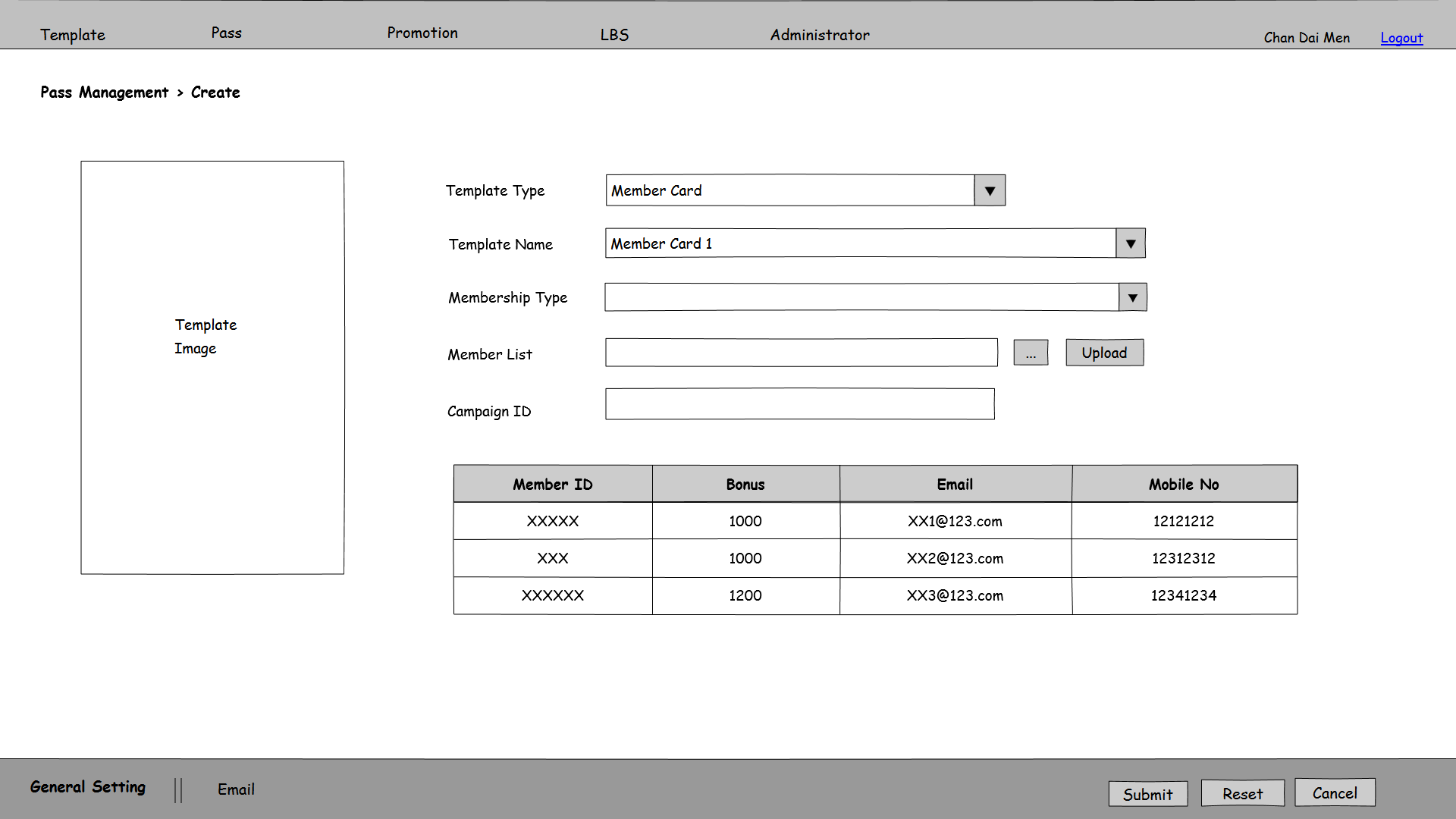


Figure 3.3.2c PMS-PAS-002

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Template Type | Pull down Menu | - | Y | Selection list including “Member Card” and “Coupon” |
| Template Name | Pull down Menu | - | Y | Get the Selection list when Template Type is selected. |
| Template Image | Image | - | - | Show the template image when template name is selected. |
| Membership Type | Pull down menu | - | N | Get the Selection list as the home page shown. |
| Member List | Textbox | VARCHAR(200) | N |  |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Campaign ID | Textbox | VARCHAR(15) | N |  |
| Email Template | Pull down menu | - | N | Selection list gets from email template table. |
| Email Subject | Textbox | NVARCHAR(200) | Y | Automatic fill in content when Default Email Template is selected. |
| Email Content | HTML Designer | NVARCHAR(MAX) | Y | Use #name# to insert members’ name, user #link# to insert pass download link  Automatic fill in content when Default Email Template is selected. |
| Save As Template | Button | - | - | Save the inputted parameters as an email template.  Enable when user has Email Template edit function access right. |
| Submit | Button | - | - | Save the inputted parameters as a pass record. Send the email with download link to the members in member list and back to PMS-PAS-001 |
| Reset | Button | - | - | Clear all inputted fields |
| Cancel | Button | - | - | Back to PMS-PAS-001 |

**Flow and Actions**

1. Refer to PMS-PAS-001, click on “Create” and enter this functions as figure 3.3.2a shown.

2. Select Template Type, and the Template Name field will show the corresponding template. And there will be a table including the fields required by the template displayed below as figure 3.3.2b shown. The columns of table shown depend on the template selected.

3. Select Template Name, and the template image will show in “Template Image” area.

4. Upload a member list file through and “Upload” button. Row checking for upload member list will be done after upload member list file. The records which are missing values or has incorrect values will be shown in the table in figure 3.3.2b



5. Click on “Email” and can go to Email tab as figure 3.3.2c shows. After “Email Template” is selected, the Email Subject and Email Content will be filled by contents defined in Email Template function. If required, user can edit the selected email template and save it for this batch of passes.

6. If user has access right for Email Template Edit, the “Save As Template” button should be enabled.

7. After input all mandatory fields, press “Submit” button to save the pass record and send email with the pass download link to target member list. After that, back to PMS-PAS-001 screen.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. File format used by Member List upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for Member List must be in csv format.”

3. Member list for pass creation must be selected by “Member List Type” or uploaded by “Member List”.

3. If the table in figure 3.3.2 has records, which means there are invalid records in uploaded member list, the system will show error message “Invalid uploaded member list.”.

4. Email Content must have “#link#” in order to send notification to update the pass. Otherwise, show error message “Missing #link# in Email Content”.

**3.4 Promotion**

### 3.4.1 Promotion Management - Search

|  |  |
| --- | --- |
| **Function ID:** | FN-PRO-001 |
| **Function Name:** | Promotion Management - Search |
| **Target User:** | All users |

**Function Description**

This function provides an interface for users to manage promotion. Users can search, create, edit and withdraw promotion in this screen.

**Function Screen**

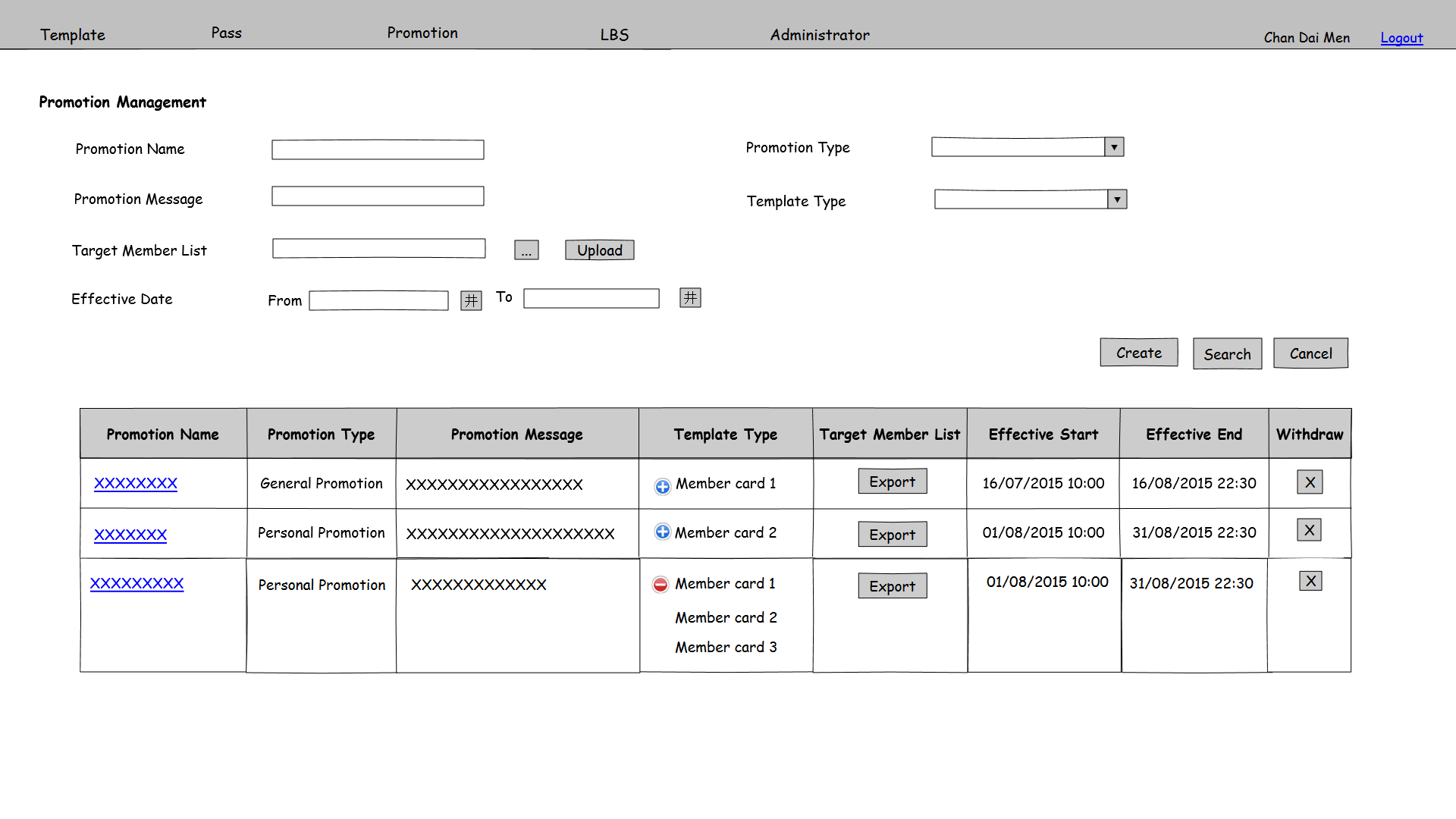


Figure 3.4.1 PMS-PRO-001

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Promotion Name | Textbox | VARCHAR(50) | N |  |
| Promotion Type | Pull down menu | - | N | Selection choices including “All”, “General Promotion”, “Personal Promotion”  Default value is “All” |
| Promotion Message | Textbox | NVARCHAR(45) | N |  |
| Template Type | Pull down menu | - | N | Selection choices get from Template table  Suspend template will not show.  Default value is “All” |
| Target Member List | Textbox | VARCHAR (200) | N | User can input member IDs with comma delimiter. |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Effective Date From | Textbox | Date time | N | Format is dd/mm/yyyy |
| Effective Date To | Textbox | Date time | N | Format is dd/mm/yyyy |
| Create | Button | - | - | Button for users to click on it and go to promotion create page PMS-PRO-002 |
| Search | Button | - | - | Search by the input parameters. The result list will order by Name |
| Cancel | Button | - | - | Clear all input fields |
| Promotion Name | Display | - | - | Hyper link for users to click on it and go to LBS edit page PMS-PRO-003 |
| Promotion Type | Display | - | - |  |
| Promotion Message | Display | - | - |  |
| Template Type | Display | - | - | Click on and this column will show all the template type have relationship with this record. |
| Export | Button | - | - | Export the members list that have relationship with this record. |
| Effective Start Date | Display | - | - |  |
| Effective End Date | Display | - | - |  |
| Withdraw | Button | - | - | Enable only before the record is effective.  Prompt a confirm dialog. If click “OK”, mark delete the corresponding record from promotion table |

**Flow and Actions**

1. User input search criteria(s) and press “Search”.

2. System searches the results and shows the searching result to users. The searching result will order by Promotion Name, Promotion Type and Effective Date ascending.

3. User click button to withdraw the promotion only before the record is effective. Then a confirmation window will prompt. After clicking “Yes”, the promotion will be withdraw and the record will be marked as deleted.



**Form Validation**

1. Effective Date From must smaller or equal to Effective Date To. Otherwise, show error message “Invalid Effective Date. Effective Date From must be smaller or equal to Effective Date To.”

2. File format used by Target Member List upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for Target Member List must be csv (comma delimiter) format.”

### 3.4.2 Promotion Management - Create

|  |  |
| --- | --- |
| **Function ID:** | FN-PRO-002 |
| **Function Name:** | Promotion Management - Create |
| **Target User:** | All users |

**Function Description**

This function provides an interface to create a new promotion.

**Function Screen**

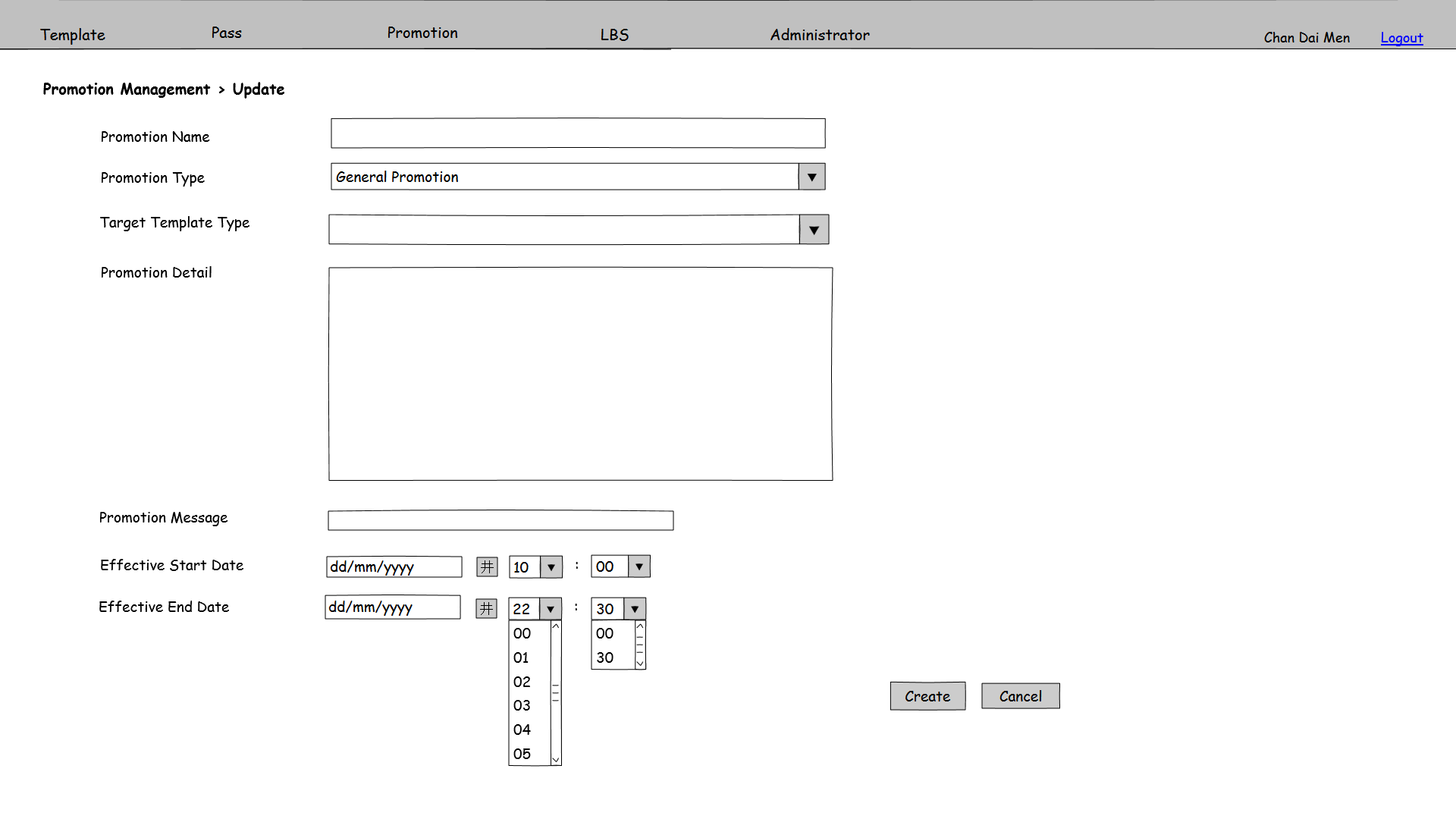


Figure 3.4.2a PMS-PRO-002

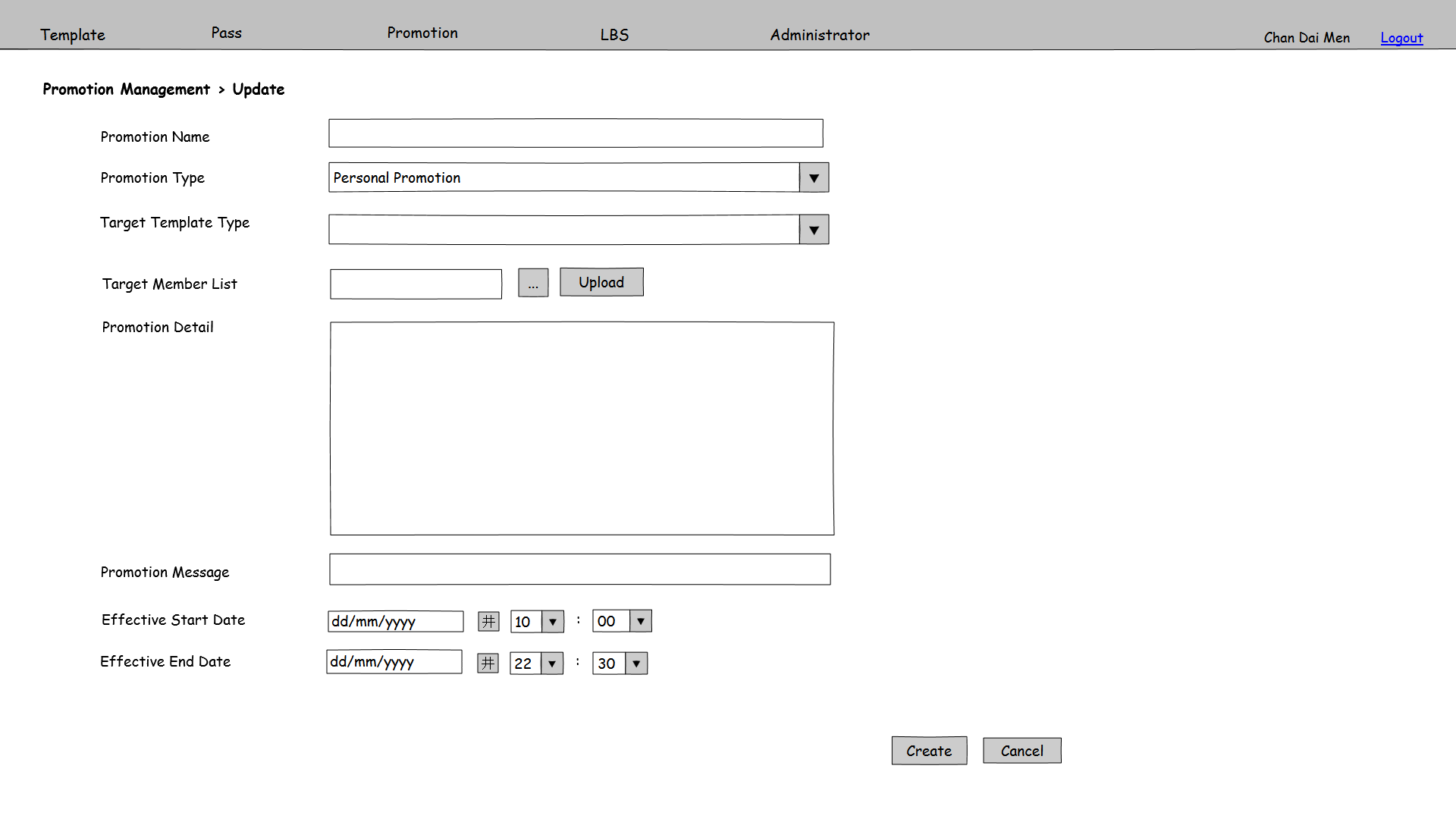


Figure 3.4.2b PMS-PRO-002

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Promotion Name | Textbox | VARCHAR(50) | Y |  |
| Promotion Type | Pull down menu | - | Y | Selection list including “General Promotion” and “Personal Promotion” |
| Target Template Type | Pull down menu | - | Y | Selection list gets from template table  Suspend template will not show.  Multiple select |
| Target Member List | Textbox | VARCHAR(200) | Y | Enabled when promotion type is “Personal Promotion” |
|  | Button | - | - | Enabled when promotion type is “Personal Promotion” |
| Upload | Button | - | - | Enabled when promotion type is “Personal Promotion” |
| Promotion Detail | Text Area | NVARCHAR(MAX) | Y |  |
| Promotion Message | Textbox | NVARCHAR(45) | Y |  |
| Effective Start Date | Textbox | Date time | Y |  |
| Effective End Date | Textbox | Date time | Y |  |
| OK | Button | - | - | Save the inputted parameters as a promotion record. Launch batch job “Update Pass for promotion information” and go to PMS-PRO-001 |
| Cancel | Button | - | - | Back to PMS-PRO-001 |

**Flow and Actions**

1. Refer to PMS-PRO-001, click on “Create” and enter this functions as figure 3.4.2a shown.

2. Select Promotion Type,

For user select General Promotion, the Template Create screen will show as figure 3.4.2b. And the target member list for this promotion will be all members.

3. After input all mandatory fields, press “Create” button to save the promotion record and back to PMS-PRO-001

For user select Personal Promotion, the Template Create screen will show as figure 3.4.2c. Target Member List will show as an input field.

3. Upload a Target Member List through and “Upload” button.



4. After input all mandatory fields, press “Create” button to save the promotion record and back to PMS-PRO-001.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Promotion Name is unique. Check duplicate on Promotion Name. If there is duplicate record in database, show error message “Promotion Name is already exists. Please input other Promotion Name.”

3. Effective Start Date must smaller than Effective End Date. Otherwise, show error message “Invalid Effective Date. Effective Start Date must be smaller than Effective End Date.”

4. File format used by target member list upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for target member list must be in csv (comma delimiter) format.”

### 3.4.3 Promotion Management - Update

|  |  |
| --- | --- |
| **Function ID:** | FN-PRO-003 |
| **Function Name:** | Promotion Management - Update |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface to update an existing promotion.

**Function Screen**

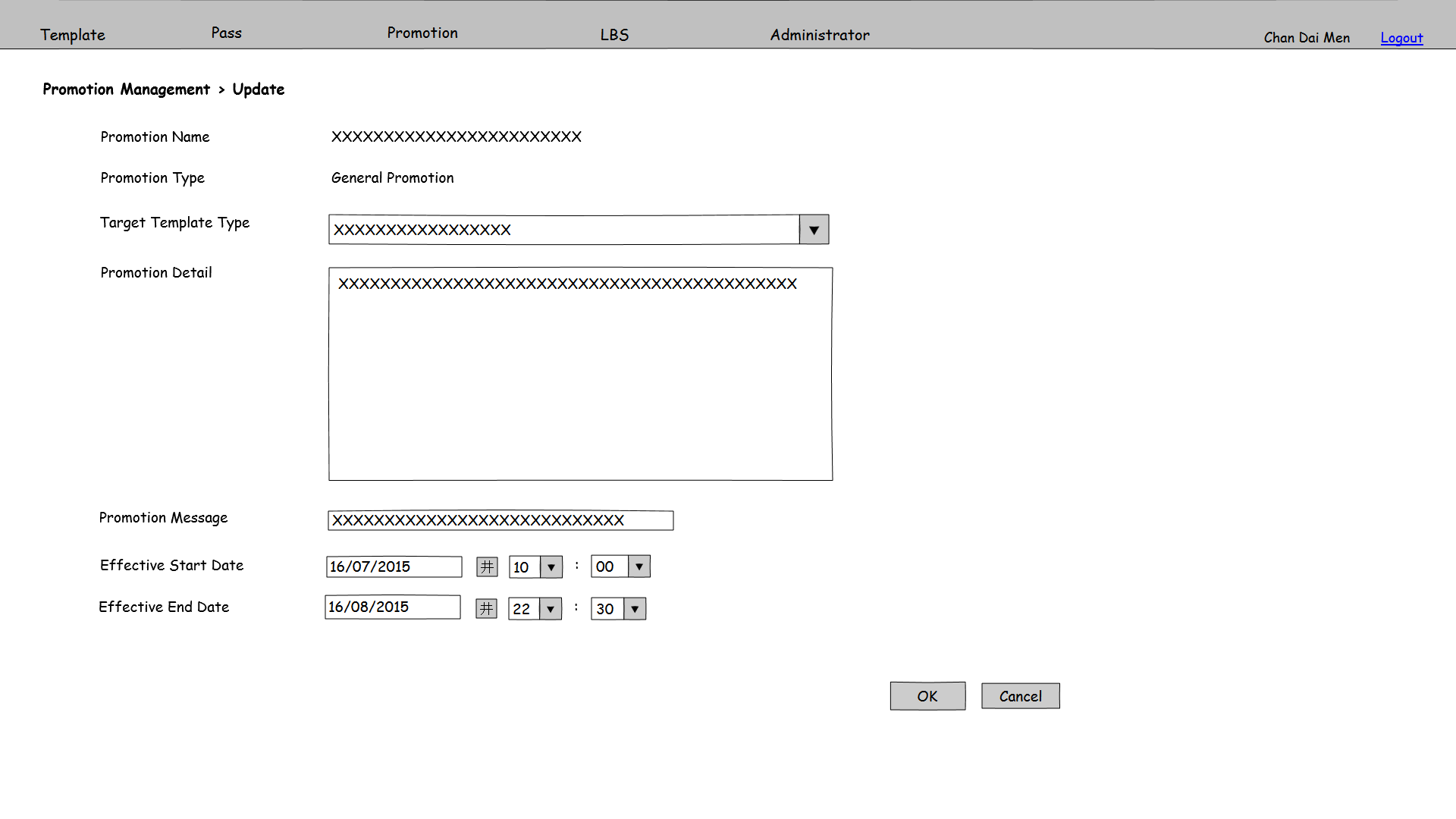


Figure 3.4.3a PMS-PRO-003

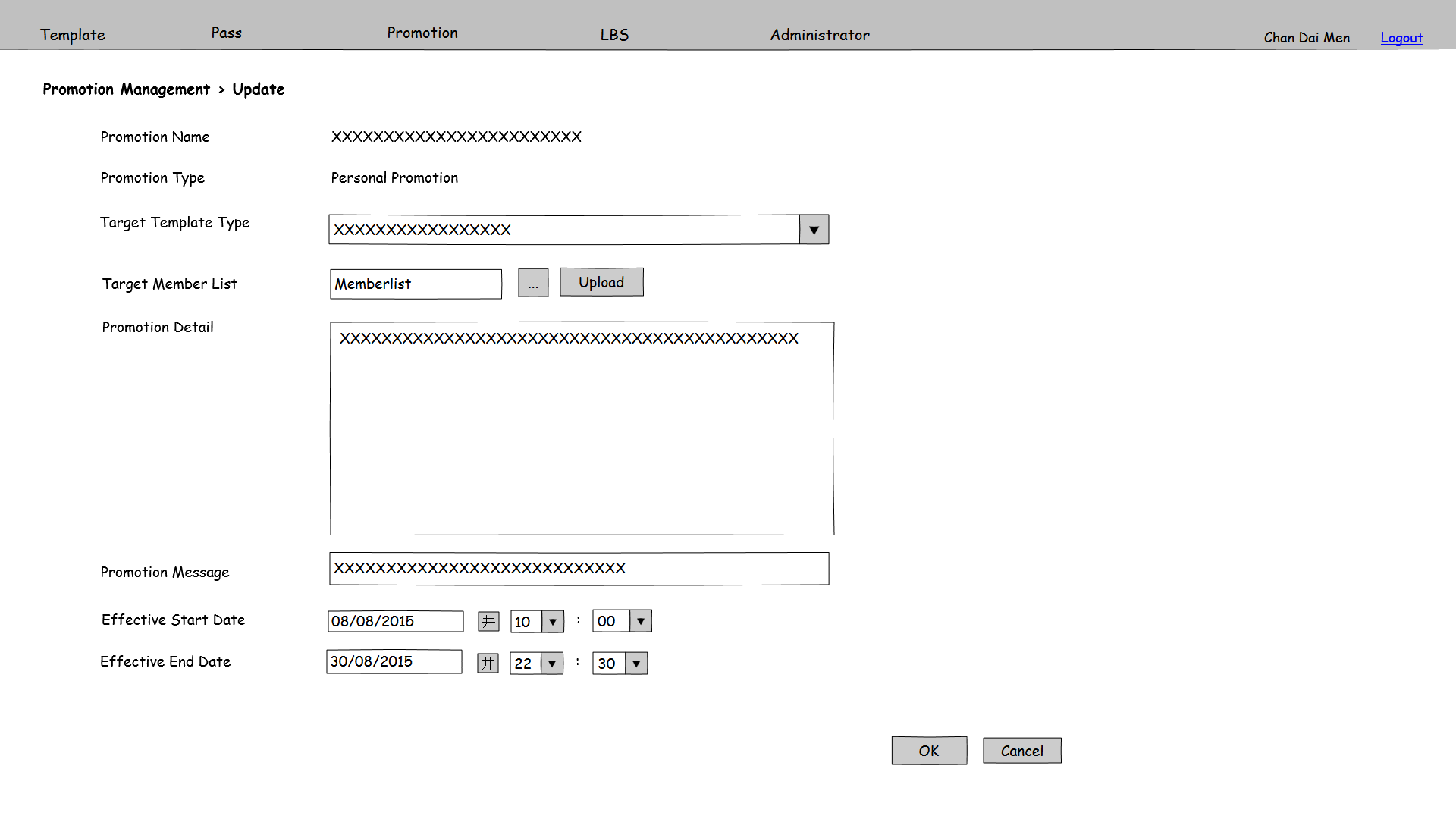


Figure 3.4.3b PMS-PRO-003

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Promotion Name | Display | - | - |  |
| Promotion Type | Display | - | - |  |
| Target Template Type | Pull down menu | - | Y | Selection list gets from template table  Suspend template will not show.  Enable only before the Promotion becomes effective.  Multiple select |
| Target Member List | Textbox | VARCHAR(200) | Y | Enable when promotion type is “Personal Promotion”  Enable only before the promotion becomes effective. |
|  | Button | - | - | Enable when promotion type is “Personal Promotion”  Enable only before the promotion becomes effective. |
| Upload | Button | - | - | Enable when promotion type is “Personal Promotion”  Enable only before the promotion becomes effective. |
| Promotion Detail | Text Area | NVARCHAR(MAX) | Y |  |
| Promotion Message | Textbox | NVARCHAR(45) | Y |  |
| Effective Start Date | Textbox | Date time | Y |  |
| Effective End Date | Textbox | Date time | Y |  |
| OK | Button | - | - | Save the inputted parameters as a promotion record. Launch batch job “Update Pass for promotion information” and go to PMS-PRO-001 |
| Cancel | Button | - | - | Back to PMS-PRO-001 |

**Flow and Actions**

1. Refer to PMS-PRO-001, click on hyperlink of search result.

If “General Promotion” is selected, the Template Create screen will show as figure 3.4.3a. And the target member list for this promotion will be all members.

2. Inputted the parameter fields

3. After input all mandatory fields, press “OK” button to update the promotion record and back to PMS-PRO-001

If “Personal Promotion” is selected, the Template Create screen will show as figure 3.4.3b. Target Member List will show as an input field.

2. Upload a Target Member List through and “Upload” button.



3. Inputted the parameter fields.

4. After input all mandatory fields, press “OK” button to update the promotion record and back to PMS-PRO-001.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Effective Start Date must smaller than Effective End Date. Otherwise, show error message “Invalid Effective Date. Effective Start Date must be smaller than Effective End Date.”

3. File format used by target member list upload is in csv (comma delimiter). Otherwise, show error message “Invalid file format. File for target member list must be in csv (comma delimiter) format.”

## 3.5 Location Based Service (LBS)

### 3.5.1 LBS Management - Search

**Function ID:** FN-LBS-001

**Function Name:** LBS Management - Search

**Target User:** All users

**Function Description**

This function provides an interface for users to manage LBS. Users can search, create, edit and withdraw LBS in this screen.

**Function Screen**

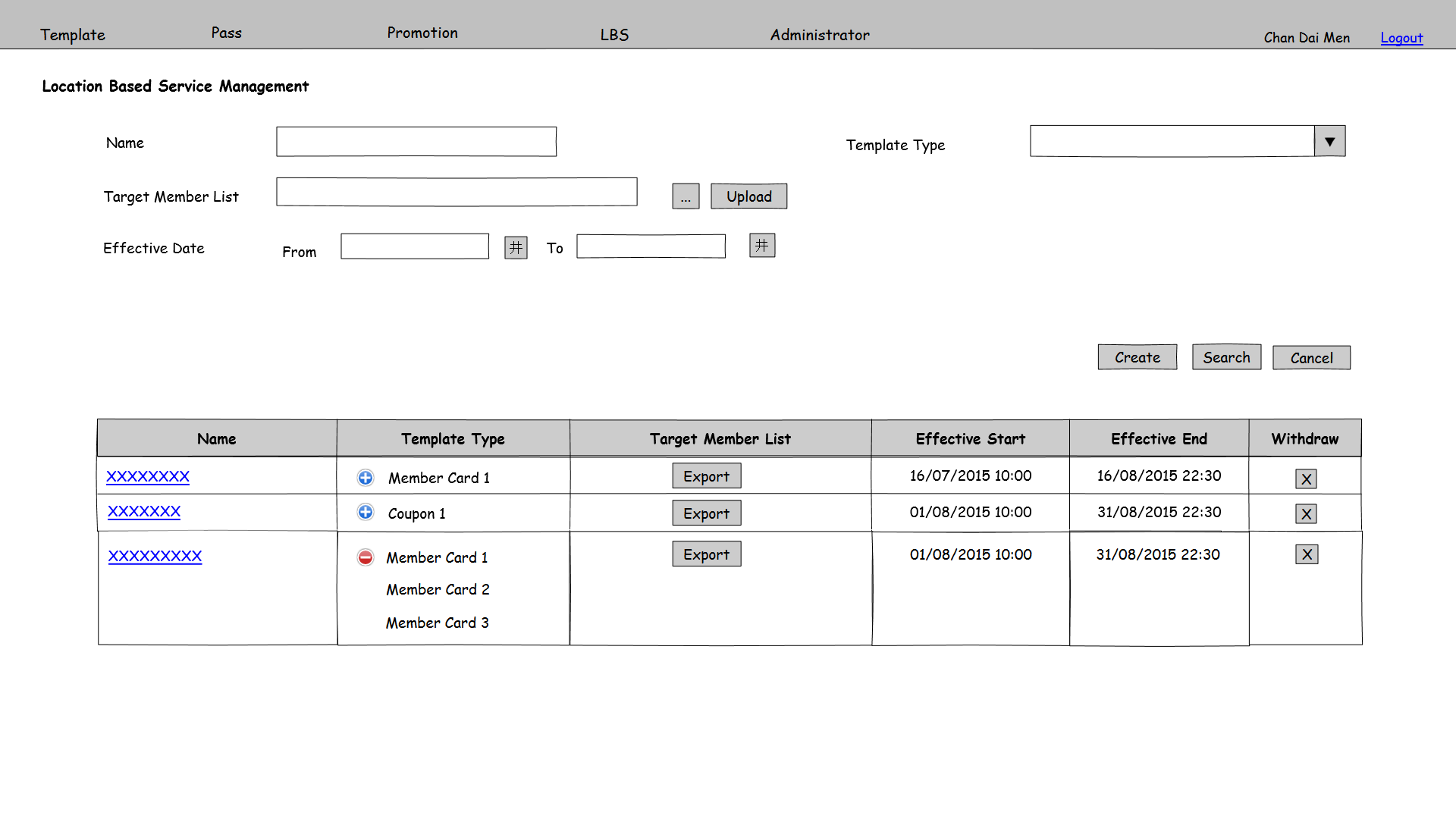


Figure 3.5.1 PMS-LBS-001

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Name | Textbox | VARCHAR(50) | N |  |
| Template Type | Pull down menu | - | N | Selection choices get from Template table  Suspend template will not show.  Default value is “All” |
| Target Member List | Textbox | VARCHAR (200) | N | User can input member IDs with comma delimiter. |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Effective Date From | Textbox | Date time | N | Format is dd/mm/yyyy |
| Effective Date To | Textbox | Date time | N | Format is dd/mm/yyyy |
| Create | Button | - | - | Button for users to click on it and go to LBS create page PMS-LBS-002 |
| Search | Button | - | - | Search by the input parameters. The result list will order by Name |
| Cancel | Button | - | - | Clear all input fields |
| Name | Display | - | - | Hyper link for users to click on it and go to LBS edit page PMS-LBS-003 |
| Template Type | Display | - | - | Click on and this column will show all the template type have relationship with this record. |
| Export | Button | - | - | Export the members list that have relationship with this record. |
| Effective Start Date | Display | - | - | Format is dd/mm/yyyy hh:mm |
| Effective End Date | Display | - | - | Format is dd/mm/yyyy hh:mm |
| Withdraw | Button | - | - | Enable only before the record is effective.  Prompt a confirm dialog. If click “OK”, mark delete the corresponding record from LBS table |

**Flow and Actions**

1. User input search criteria(s) and click “Search” button.

2. System searches the results and shows the searching result to users. The searching result will order by LBS Name and Effective Date ascending.

3. User can click button to withdraw the LBS only before effective. Then a confirmation window will prompt. After clicking “Yes”, the LBS will be withdraw and the record will be marked delete.



**Form Validation**

1. Effective Date From must smaller or equal to Effective Date To. Otherwise, show error message “Invalid Effective Date. Effective Date From must be smaller or equal to Effective Date To.”

2. File format used by Target Member List upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for Target Member List must be in csv (comma delimiter) format.”

### 3.5.2 LBS Management - Create

|  |  |
| --- | --- |
| **Function ID:** | FN-LBS-002 |
| **Function Name:** | LBS Management – Create |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface to create a new LBS.

**Function Screen**

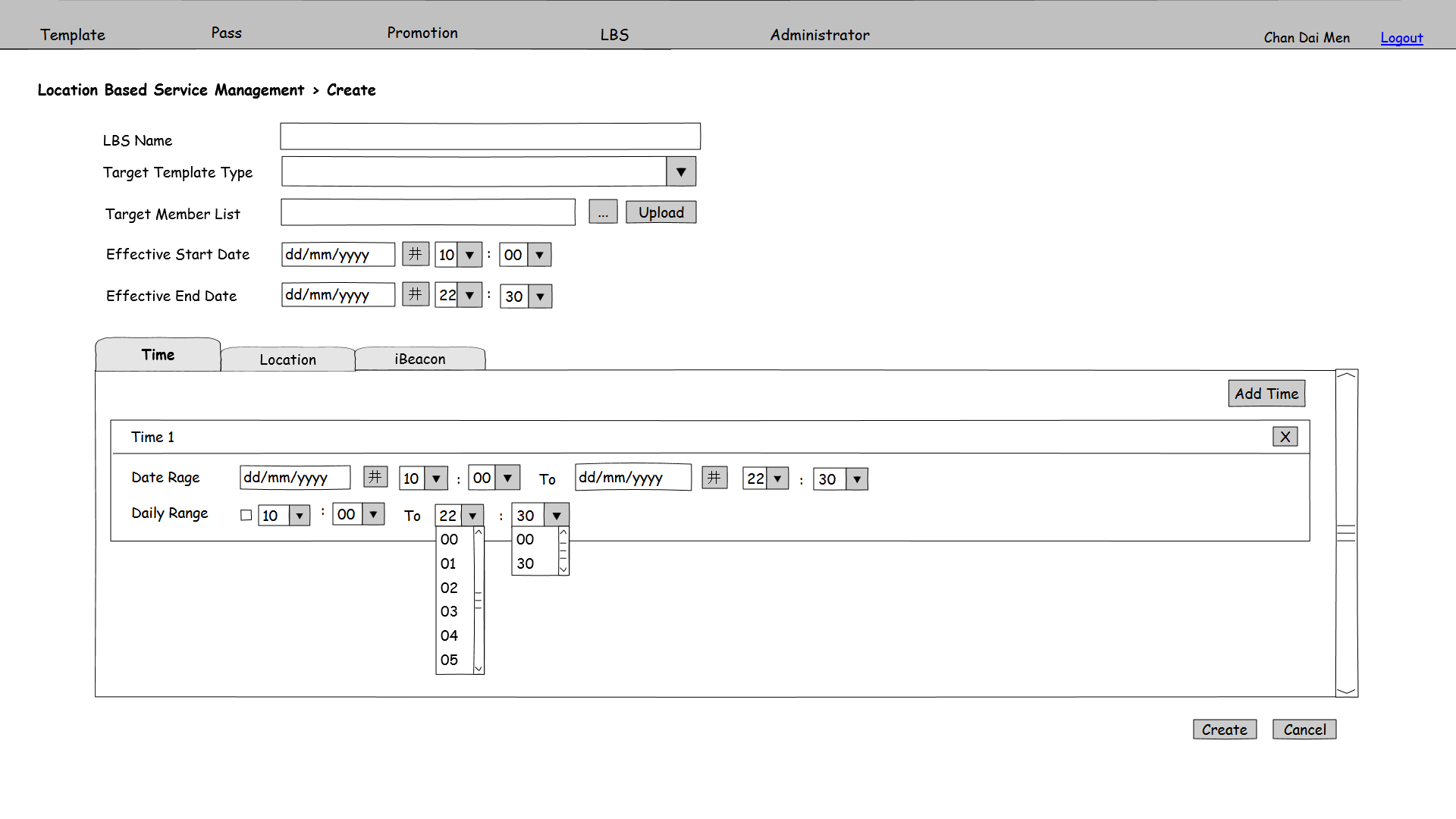


Figure 3.5.2a PMS-LBS-002

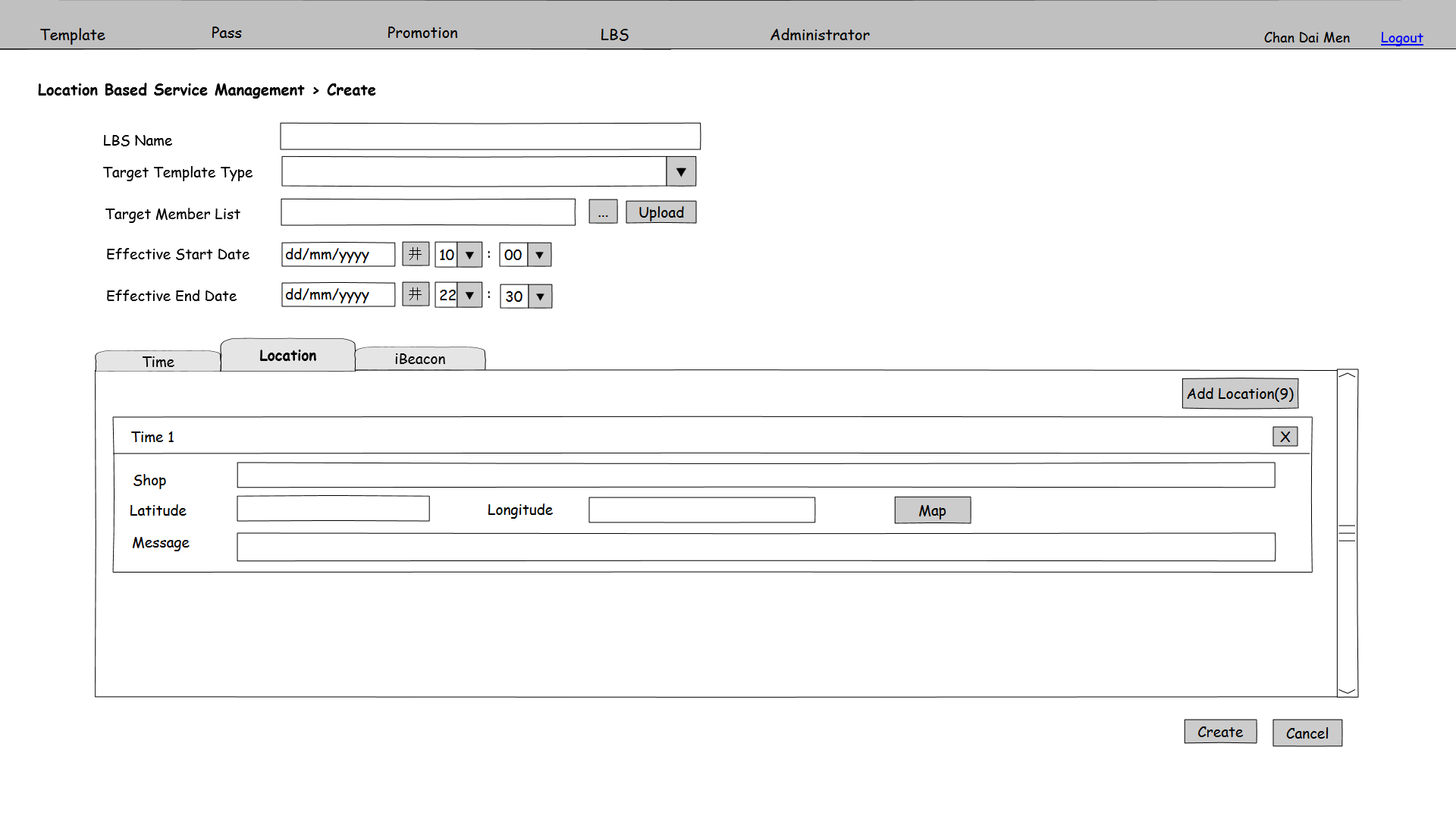


Figure 3.5.2b PMS-LBS-002



Figure 3.5.2c PMS-LBS-003

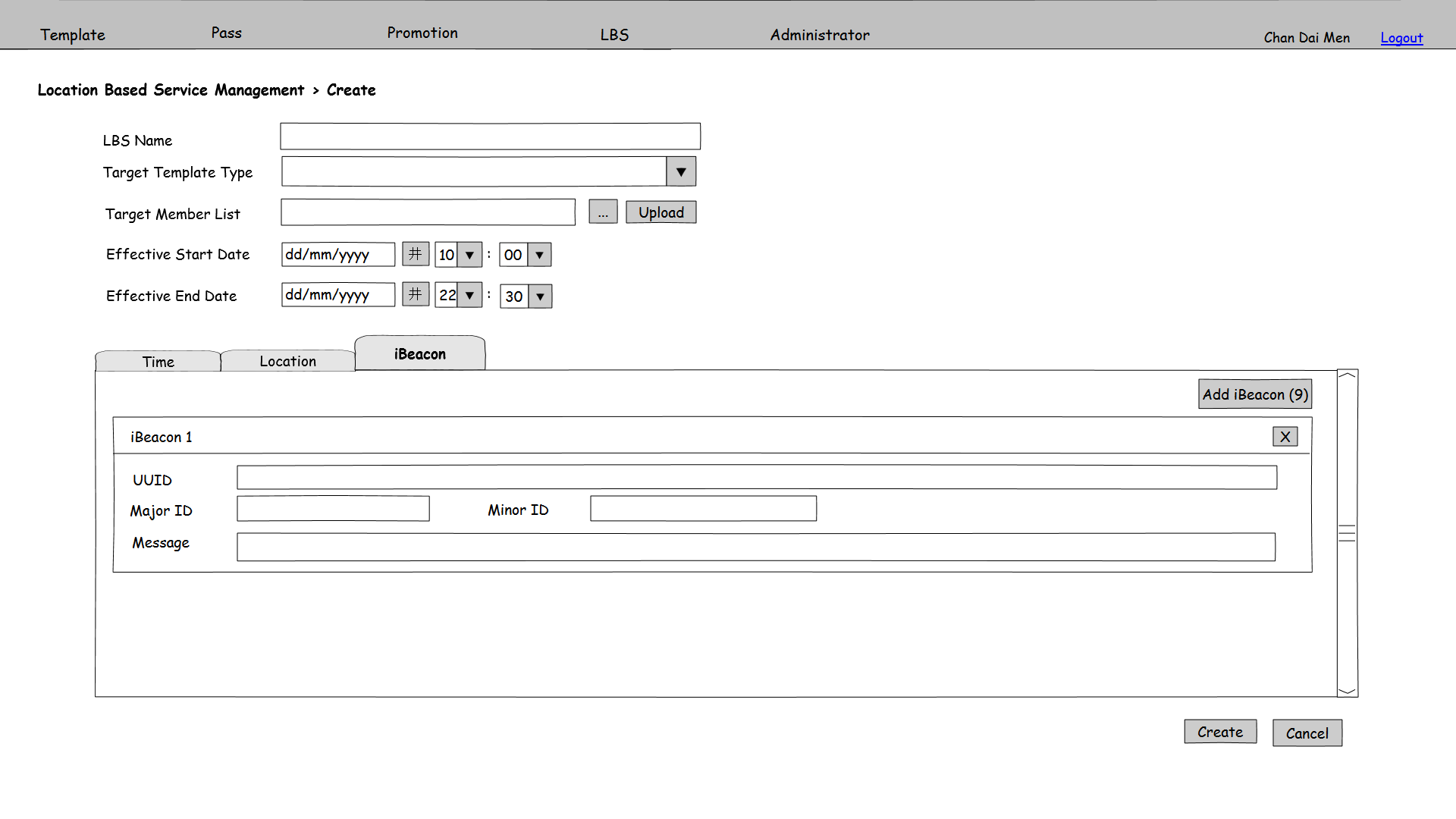


Figure 3.5.2d PMS-LBS-002

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| LBS Name | Textbox | VARCHAR(50) | Y |  |
| Target Template Type | Pull down menu | - | Y | Selection list gets from template table Suspend template will not show.  Multiple select |
| Target Member List | Textbox | VARCHAR(200) | Y |  |
|  | Button | - | - |  |
| Upload | Button | - | - |  |
| Effective Start Date | Textbox | Date time | Y |  |
| Effective End Date | Textbox | Date time | Y |  |
| Create | Button | - | - | Save the inputted parameters as a LBS record. Launch batch job “Update Pass for LBS information” and go to PMS-LBS-001 |
| Cancel | Button | - | - | Back to PMS-LBS-001 |
| In Time tab | | | | |
| Add Time | Button | - | - | User can click on it to add time information for this LBS. |
|  | Button | - | - | User can delete the existing time information by clicking on this button. |
| Date Range From | Textbox | Date time | N |  |
| Date Range To | Textbox | Date time | N |  |
| Daily Range | Checkbox | - | - |  |
| Daily Range From | Textbox | Date time | N | Enable when Daily Range checkbox is checked. |
| Daily Range To | Textbox | Date time | N | Enable when Daily Range checkbox is checked. |
| In Location tab | | | | |
| Add Location | Button | - | - | User can click on it to add location information for this LBS.  The maximum number of Locations is 10. |
|  | Button | - | - | User can delete the existing location information by clicking on this button. |
| Shop | Textbox | NVARCHAR(200) | N |  |
| Longitude | Textbox | Double | N |  |
| Latitude | Textbox | Double | N |  |
| Map | Button | - | - | User can click on it and prompt a window for Google map search, as figure3.5.2c shown. |
| Message | Textbox | NVARCHAR(45) | N |  |
| In iBeacon tab | | | | |
| Add iBeacon | Button | - | - | User can click on it to add iBeacon information for this LBS.  The maximum number of iBeacon is 10. |
|  | Button | - | - | User can delete the existing iBeacon information by clicking on this button. |
| UUID | Textbox | NVARCHAR(255) | N |  |
| Major ID | Textbox | Integer | N |  |
| Minor ID | Textbox | Integer | N |  |
| Message | Textbox | NVARCHAR(45) | N |  |

**Flow and Actions**

1. Refer to PMS-LBS-001, click on “Create” and enter this functions as figure 3.5.2a shown.

2. Inputted the parameter fields.

3. In Time tab, user can add relevant time range for the pass. Click “Add Time” to add a new time range and to delete time range.



4. Click Location and go to location tab. The screen is as 3.5.2b shown. Click “Add Location” to add a new location and to delete an existing one. The maximum number of locations can save to a LBS is 10.



If user want to search a location of a shop, click on “Map” button and a window prompt with Google Map search function for user to select a location as figure 3.5.2c shown. After select a shop location and click “OK”, the coordination of this shop will automatic fill in shop name, Latitude and Longitude.

5. Click iBeacon and go to iBeacon tab. The screen is as 3.5.2d shown. Click “Add iBeacon” to add a new iBeacon and to delete an existing one. The maximum number of iBeacon can save to a LBS is 10.



6. After input all mandatory fields, press “Create” button to save the LBS record and back to PMS-LBS-001.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. LBS Name is unique. Check duplicate on LBS Name. If there is duplicate record in database, show error message “LBS Name is already exists. Please input other LBS Name.”

3. Effective Start Date must smaller Effective End Date. Otherwise, show error message “Invalid Effective Date. Effective Start Date must be smaller Effective End Date.”

4. File format used by target member list upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for target member list must be csv (comma delimiter) format.”

5. At least 1 location or 1 iBeacon must be added to the LBS. Otherwise, show error message “At least 1 location or 1 iBeacon must be added.”

### 3.5.3 LBS Management - Update

|  |  |
| --- | --- |
| **Function ID:** | FN-LBS-003 |
| **Function Name:** | LBS Management - Update |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface to update an existing LBS.

**Function Screen**

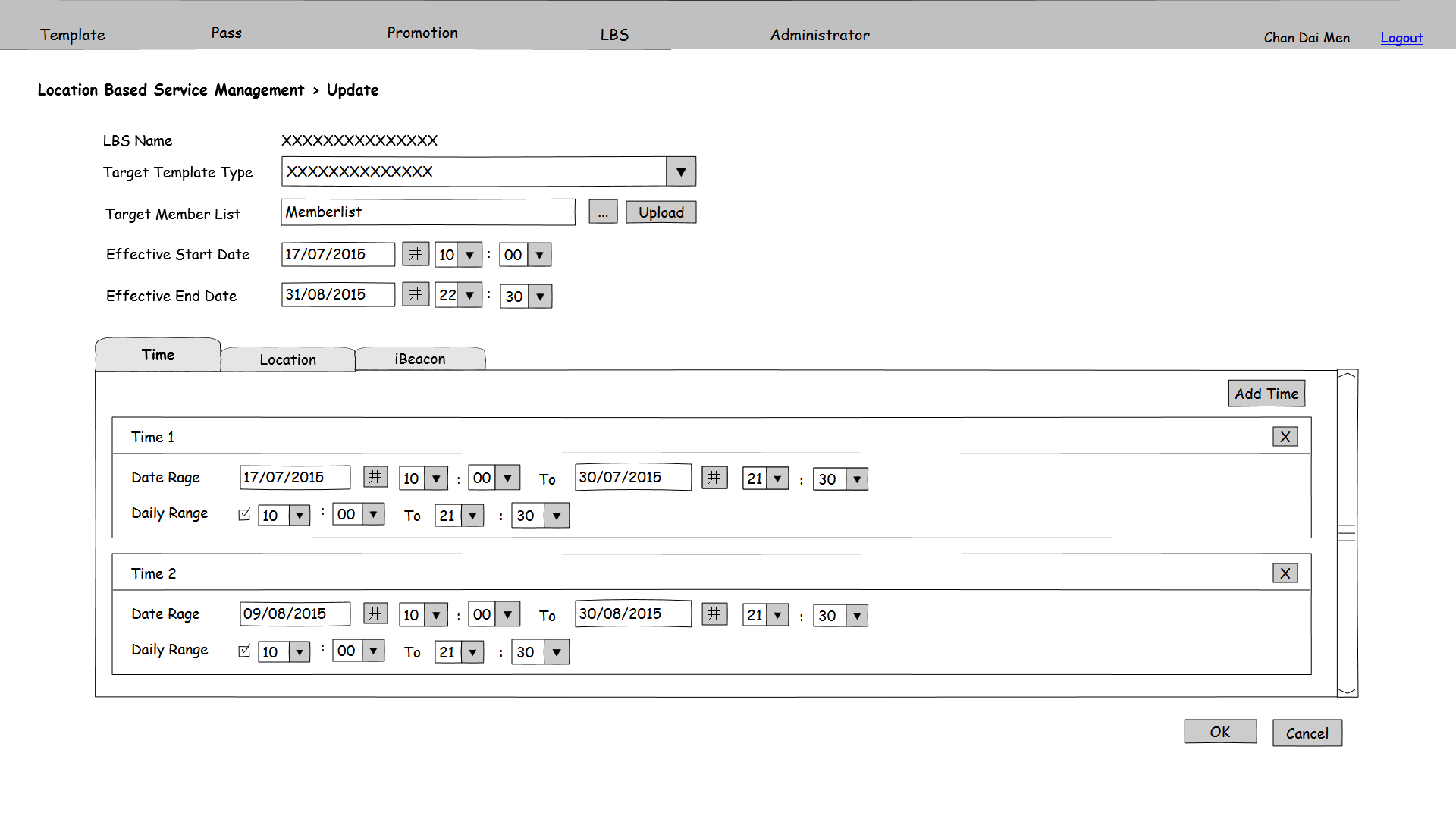


Figure 3.5.3a PMS-LBS-003

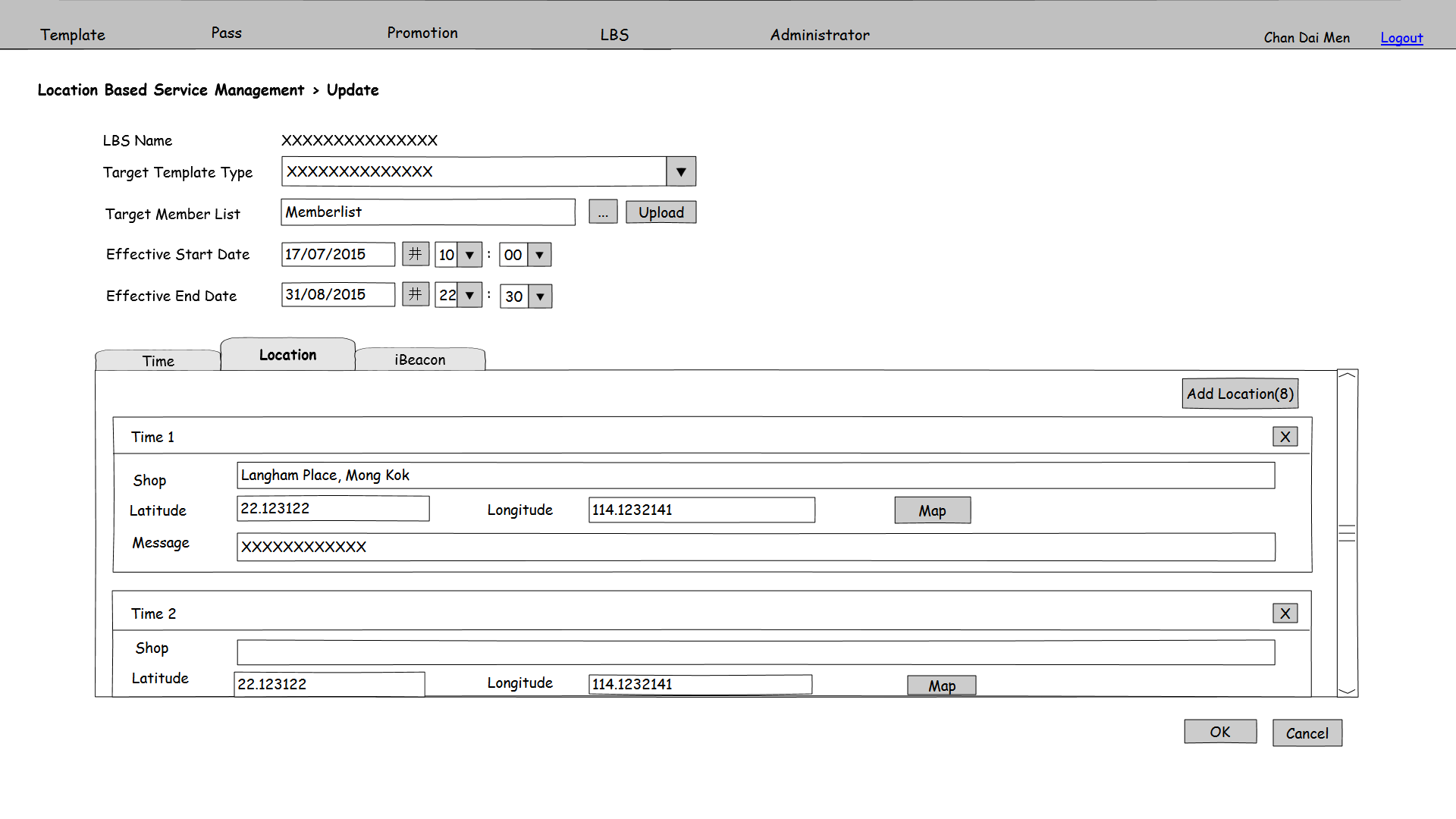
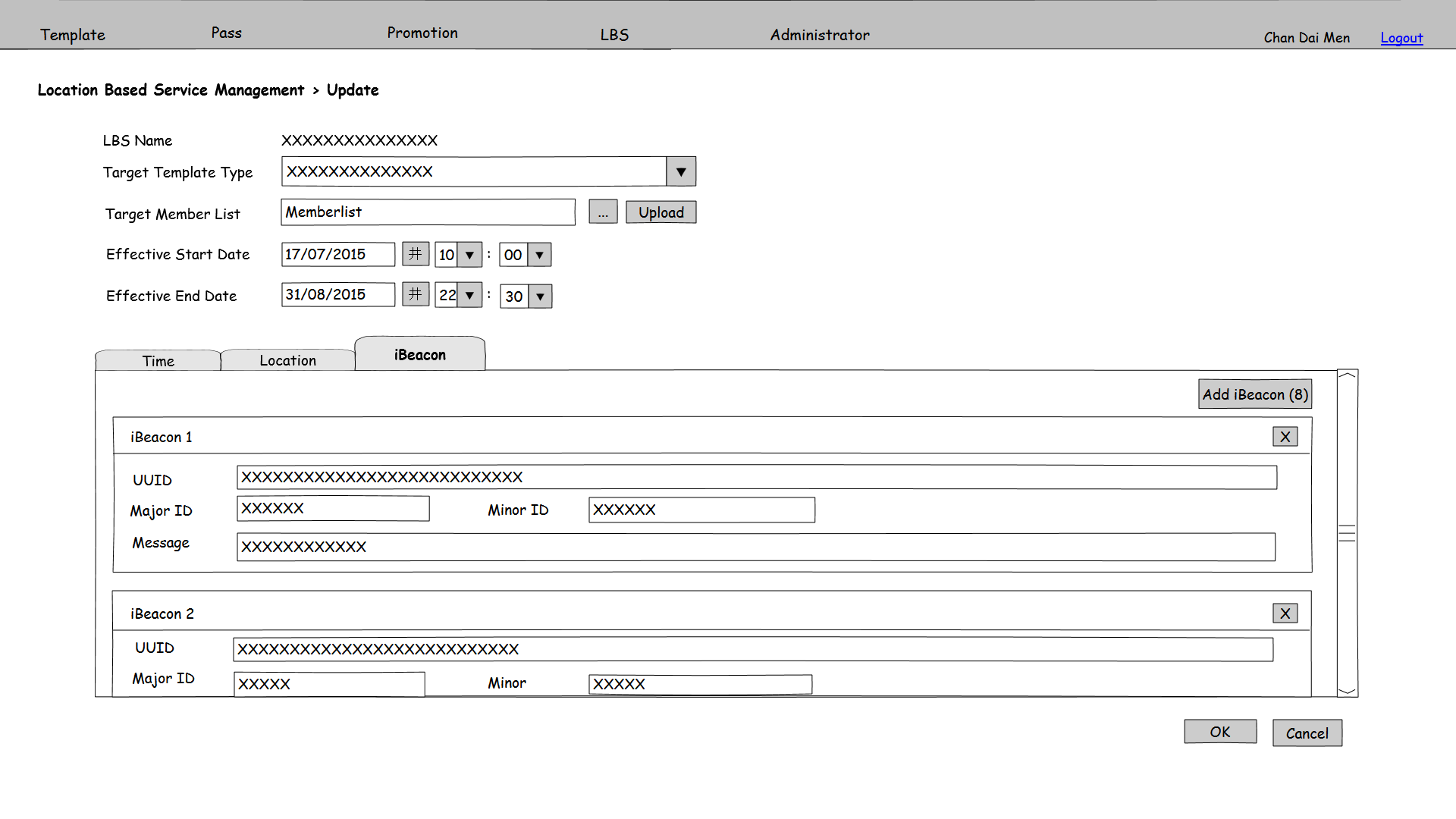


Figure 3.5.3b PMS-LBS-003

Figure 3.5.3c PMS-LBS-003



**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| LBS Name | Display | - | - |  |
| Target Template Type | Pull down menu | - | Y | Selection list gets from template table  Suspend template will not show.  Enable only before the LBS becomes effective.  Multiple select |
| Target Member List | Textbox | VARCHAR(200) | Y | Enable only before the LBS becomes effective. |
|  | Button | - | - | Enable only before the LBS becomes effective. |
| Upload | Button | - | - | Enable only before the LBS becomes effective. |
| Effective Start Date | Textbox | Date time | Y |  |
| Effective End Date | Textbox | Date time | Y |  |
| OK | Button | - | - | Save the inputted parameters as a LBS record. Launch batch job “Update Pass for LBS information” and go to PMS-LBS-001 |
| Cancel | Button | - | - | Back to PMS-LBS-001 |
| In Time tab | | | | |
| Add Time | Button | - | - | User can click on it to add time information for this LBS. |
|  | Button | - | - | User can delete the existing time information by clicking on this button. |
| Date Range From | Textbox | Date time | N |  |
| Date Range To | Textbox | Date time | N |  |
| Daily Range | Checkbox | - | - |  |
| Daily Range From | Textbox | Date time | N | Enable when Daily Range checkbox is checked. |
| Daily Range To | Textbox | Date time | N | Enable when Daily Range checkbox is checked. |
| In Location tab | | | | |
| Add Location | Button | - | - | User can click on it to add location information for this LBS.  The maximum number of Locations is 10. |
|  | Button | - | - | User can delete the existing location information by clicking on this button. |
| Shop | Textbox | NVARCHAR(200) | N |  |
| Longitude | Textbox | Double | N |  |
| Latitude | Textbox | Double | N |  |
| Map | Button | - | - | User can click on it and prompt a window for Google map search, like figure3.5.2c. |
| Message | Textbox | NVARCHAR(45) | N |  |
| In iBeacon tab | | | | |
| Add iBeacon | Button | - | - | User can click on it to add iBeacon information for this LBS.  The maximum number of iBeacon is 10. |
|  | Button | - | - | User can delete the existing iBeacon information by clicking on this button. |
| UUID | Textbox | NVARCHAR(255) | N |  |
| Major ID | Textbox | Integer | N |  |
| Minor ID | Textbox | Integer | N |  |
| Message | Textbox | NVARCHAR(45) | N |  |

**Flow and Actions**

1. Refer to PMS-LBS-001, click on hyperlink of search result.

2. Inputted the parameter fields.

3. In Time tab, user can add relevant time range for the pass. Click “Add Time” to add a new time range and to delete time range.



4. Click Location and go to location tab. The screen is as 3.5.2b shown. Click “Add Location” to add a new location and to delete an existing one. The maximum number of location can save to a LBS is 10.



If user want search a location of a shop, click on “Map” button and prompt a window with Google Map search function for user to select a location as figure 3.5.2c shown. After select a shop location and click “OK”, the coordination of this shop will automatic fill in shop name, Latitude and Longitude.

5. Click iBeacon and go to iBeacon tab. The screen is as 3.5.3c shows. Click “Add iBeacon” to add a new iBeacon and to delete an existing one. The maximum number of iBeacon can save to a LBS is 10.



6. After input all mandatory fields, press “OK” button to update the LBS record and back to PMS-LBS-001.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Effective Start Date must smaller Effective End Date. Otherwise, show error message “Invalid Effective Date. Effective Start Date must be smaller Effective End Date.”

3. File format used by target member list upload is csv (comma delimiter). Otherwise, show error message “Invalid file format. File for target member list must be csv (comma delimiter) format.”

4. At least 1 location or 1 iBeacon must be added to the LBS. Otherwise, show error message “At least 1 location or 1 iBeacon must be added.”

## 3.6 Administrator

### 3.6.1 User Management

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-001 |
| **Function Name:** | User Management |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to manage users. Users can search, create, edit, active, inactive and delete users in this screen.

**Function Screen**

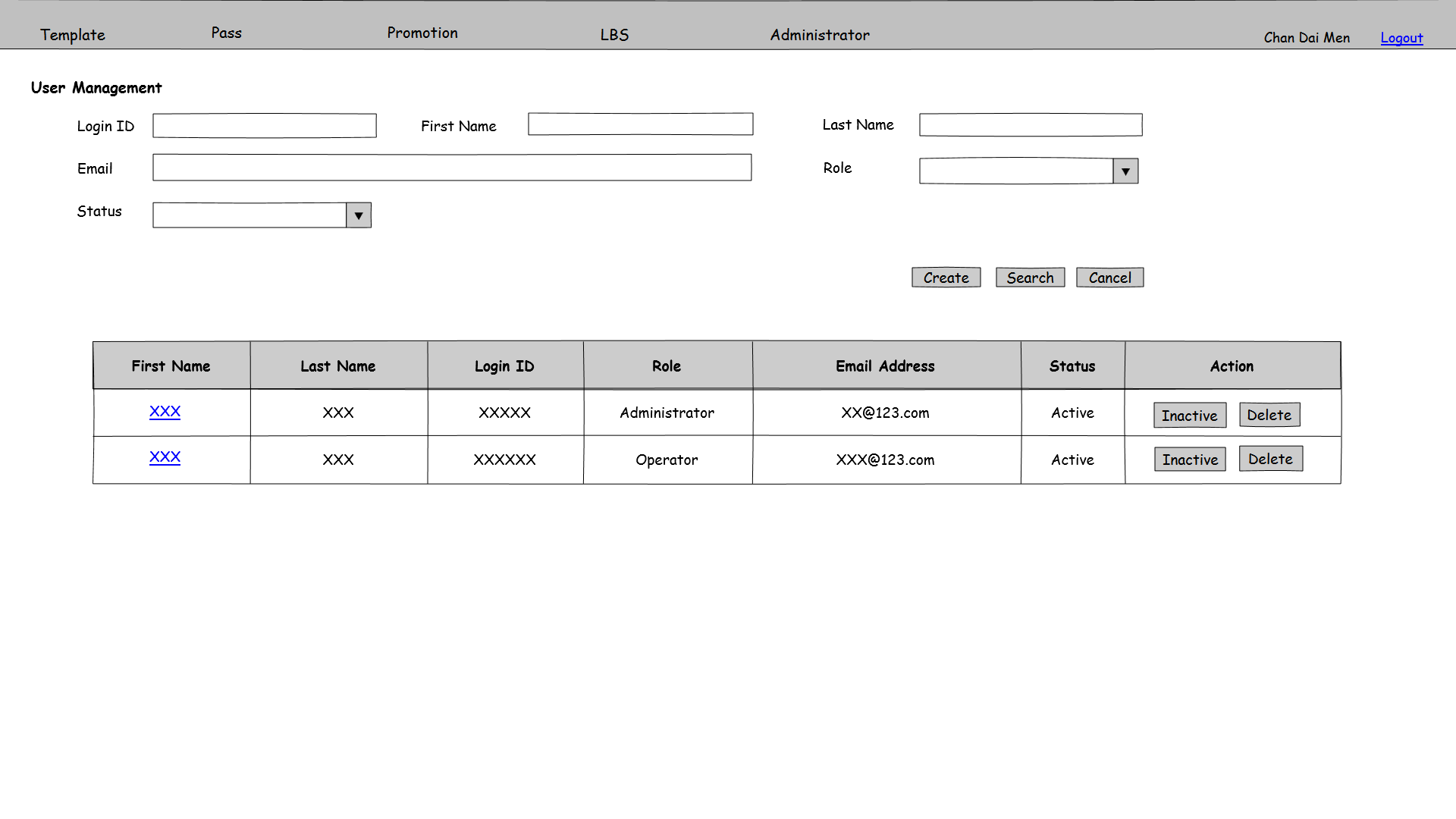


Figure 3.6.1 PMS-UAD-001

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Login ID | Textbox | VARCHAR(30) | N |  |
| First Name | Textbox | VARCHAR (50) | N |  |
| Last Name | Textbox | VARCHAR (50) | N |  |
| Email | Textbox | VARCHAR (200) | N |  |
| Role | Pull down menu | - | N | Selection choices get from user role table  Default value is “All” |
| Status | Pull down menu | - | N | Selection choices are “All”, “Active”, “Inactive”  Default value is “All” |
| Create | Button | - | - | Button for users to click on it and go to user create page PMS-UAD-002 |
| Search | Button | - | - | Search by the input parameters. The result list will order by First Name and Last Name |
| Cancel | Button | - | - | Clear all input fields |
| First Name | Link button | - | - | Hyper link for users to click on it and go to user edit page PMS-UAD-003 |
| Last Name | Display | - | - |  |
| Login ID | Display | - | - |  |
| Role | Display | - | - |  |
| Email Address | Display | - | - |  |
| Status | Display | - | - |  |
| Delete | Button | - | - | Prompt a confirm dialog. If click “OK”, delete the corresponding record from user table |
| Inactive/Active | Button | - | - | Change the user status to active/inactive and the button change to Active/Inactive |

**Flow and Actions**

1. User input search criteria(s) and press search.

2. System searches the results and shows the searching result to users. The searching result will order by first name and last name in ascending.

**Form Validation**

N/A

### 3.6.2 User Management – Create

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-002 |
| **Function Name:** | User Management - Create |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to create a new user.

**Function Screen**

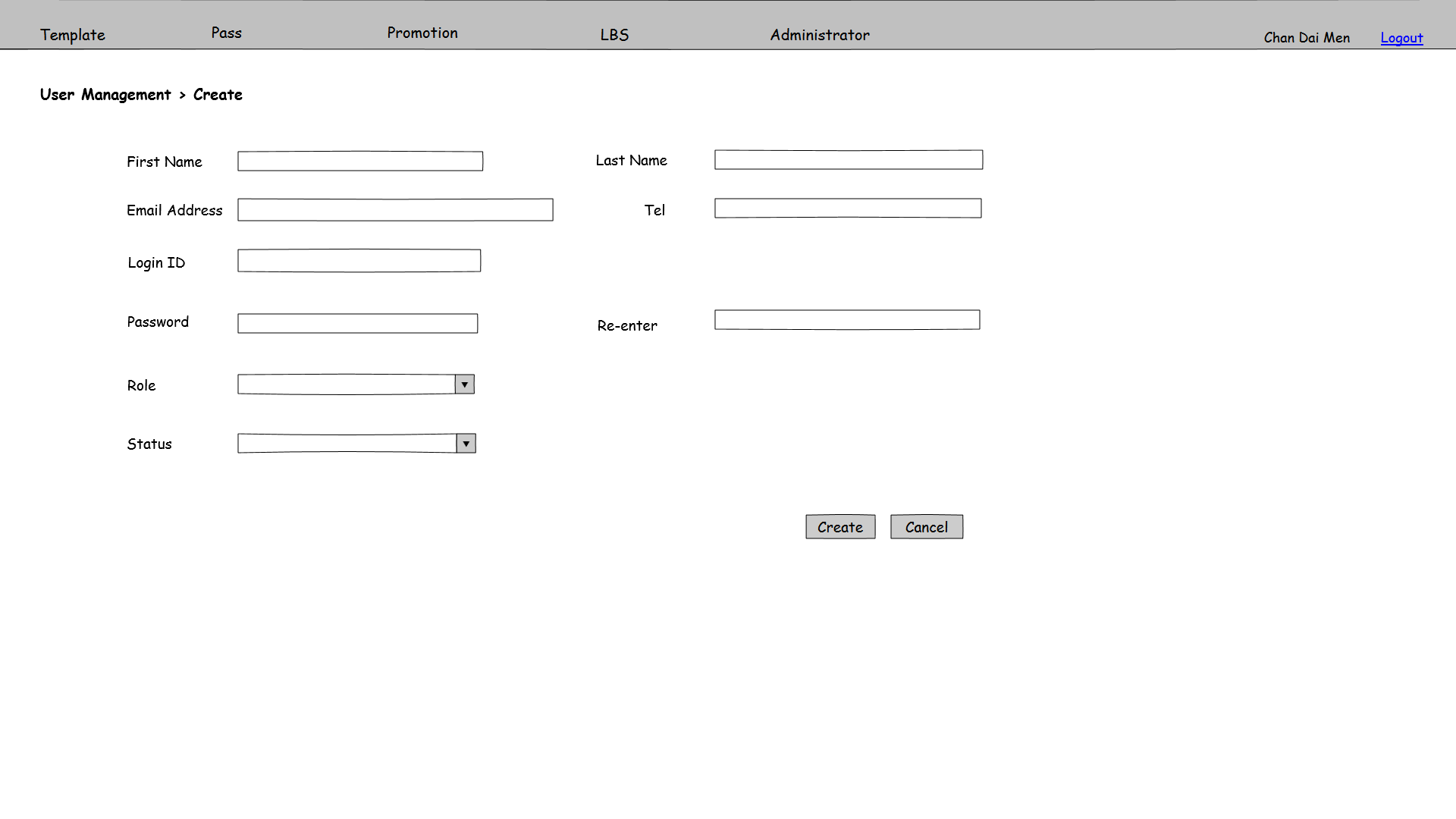


Figure 3.6.2 PMS-UAD-002

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| First Name | Textbox | VARCHAR(50) | Y |  |
| Last Name | Textbox | VARCHAR(50) | Y |  |
| Email | Textbox | VARCHAR(200) | Y | Check email format |
| Tel | Textbox | VARCHAR(15) | N |  |
| Login ID | Textbox | VARCHAR(30) | Y |  |
| Password | Textbox | VARCHAR(15) | Y | 8-15 characters and case sensitive |
| Re-enter | Textbox | VARCHAR(15) | Y | 8-15 characters and case sensitive  Must the same as Password field |
| Role | Pull down menu | - | Y | Selection list gets from user role table |
| Status | Pull down menu | - | Y | Selection choices are “Active”, “Inactive”  Default value is “Inactive” |
| New | Button | - | - | Save the inputted parameters as an user record and go to PMS-UAD-001 |
| Cancel | Button | - | - | Back to PMS-UAD-001 |

**Flow and Actions**

1. Refer to PMS-UAD-001, click on “Create”

2. Inputted the parameter fields in this screen

3. After input all mandatory fields, press “Create” button to save the user admin record and back to PMS-UAD-001

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Login ID is unique. Check duplicate on Login ID. If there is duplicate record in database, show error message “Login ID already exists. Please input other Login ID.”

3. Password and Re-enter fields are 8-15 characters and case sensitive. If not, show error message “Invalid password”

4. Password and Re-enter fields should be the same with each other. If they are different, show error message “Re-enter password is different from Password. Please input password again.”

5. Check email format. Email address must have character “@”. Otherwise, show error message “Invalid email address”.

### 3.6.3 User Management – Update

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-003 |
| **Function Name:** | User Management - Update |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to modify a user.

**Function Screen**

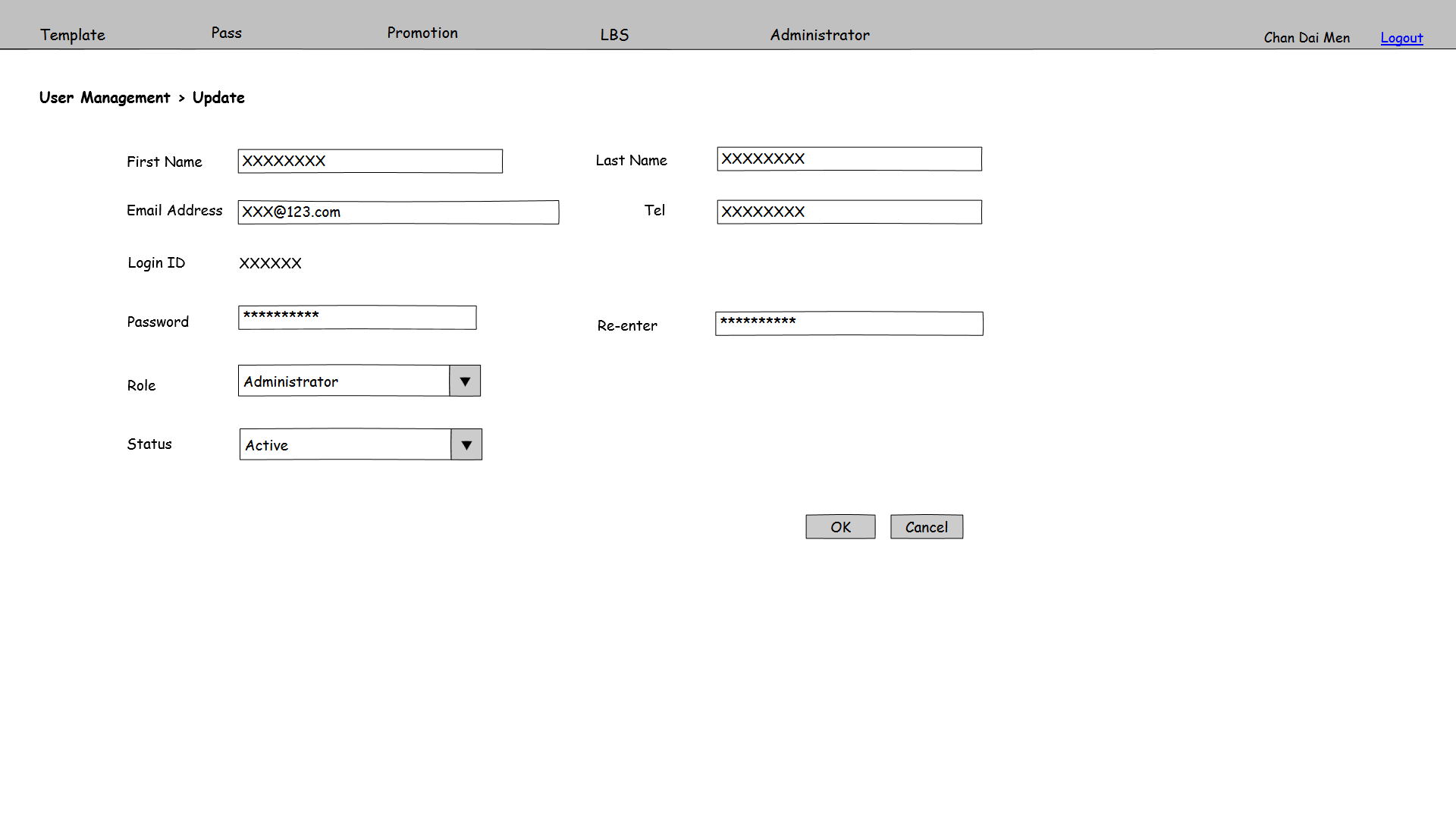


Figure 3.6.3 PMS-UAD-003

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| First Name | Textbox | VARCHAR(50) | Y |  |
| Last Name | Textbox | VARCHAR(50) | Y |  |
| Email | Textbox | VARCHAR(200) | Y | Check email format |
| Tel | Textbox | VARCHAR(15) | N |  |
| Login ID | Display | - | - |  |
| Password | Textbox | VARCHAR(15) | Y | 8-15 characters and case sensitive |
| Re-enter | Textbox | VARCHAR(15) | Y | 8-15 characters and case sensitive  Must the same as Password field |
| Role | Pull down menu | - | Y | Selection list gets from user role table |
| Status | Pull down menu | - | Y | Selection choices are “Active”, “Inactive”  Default value is “Inactive” |
| OK | Button | - | - | Update this record with inputted parameters and go to PMS-UAD-001 |
| Cancel | Button | - | - | Back to PMS-UAD-001 |

**Flow and Actions**

1. Refer to PMS-UAD-001, click on the hyperlink of search result

2. Inputted the parameter fields in this screen

3. After input all mandatory fields, press “OK” button to save the user admin record and back to PMS-UAD-001

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Password and Re-enter fields are 8-15 characters and case sensitive. If not, show error message “Invalid password”

3. Password and Re-enter fields should be the same with each other. If they are different, show error message “Re-enter password is different from Password. Please input password again.”

4. Check email format. Email address must have character “@”. Otherwise, show error message “Invalid email address”.

### 3.6.4 Role Management

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-004 |
| **Function Name:** | Role Management |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to search user role. Users can search, create, edit and delete roles in this screen.

**Function Screen**

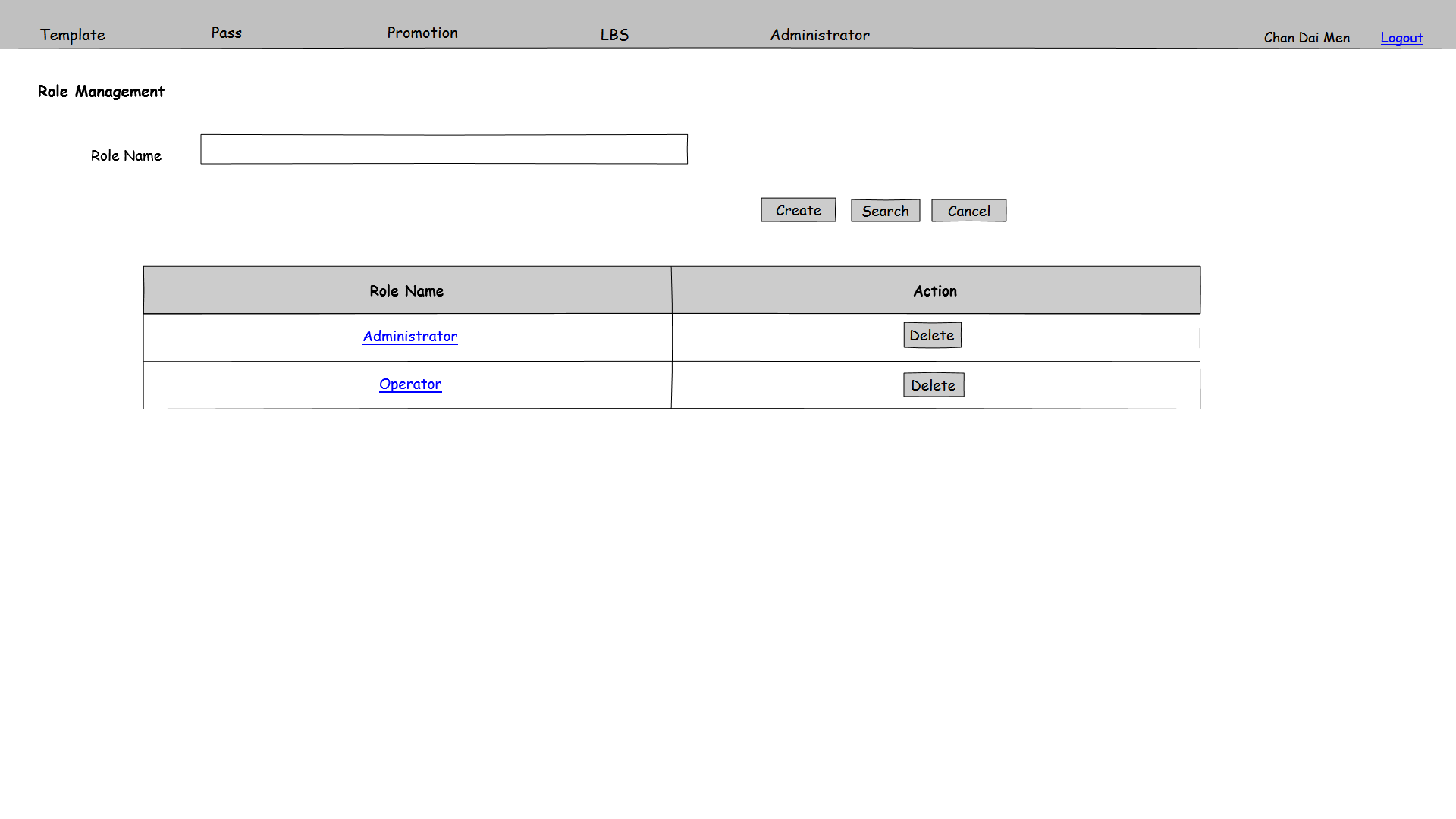


Figure 3.6.4 PMS-UAD-004

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Role Name | Textbox | VARCHAR(50) | N |  |
| Create | Button | - | - | Button for users to click on it and go to user role create page PMS-UAD-005 |
| Search | Button | - | - | Search by the input parameters. The result list will order by Role Name |
| Cancel | Button | - | - | Clear all input fields |
| Role Name | Link button | - | - | Hyper link for users to click on it and go to user role edit page PMS-UAD-006 |
| Delete | Button | - | - | Prompt a confirm dialog. If click “OK”, delete the corresponding record from role table |

**Flow and Actions**

1. User input search criteria(s) and press “Search” button.

2. System searches the results and shows the searching result to users. The searching result will order by role name in ascending.

**Form Validation**

1. If user role has already been assigned to user, can’t not delete this user role and shows error message “Users assigned to this role. Cannot delete.”

### 3.6.5 Role Management – Create

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-005 |
| **Function Name:** | Role Management - Create |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to create a new role.

**Function Screen**

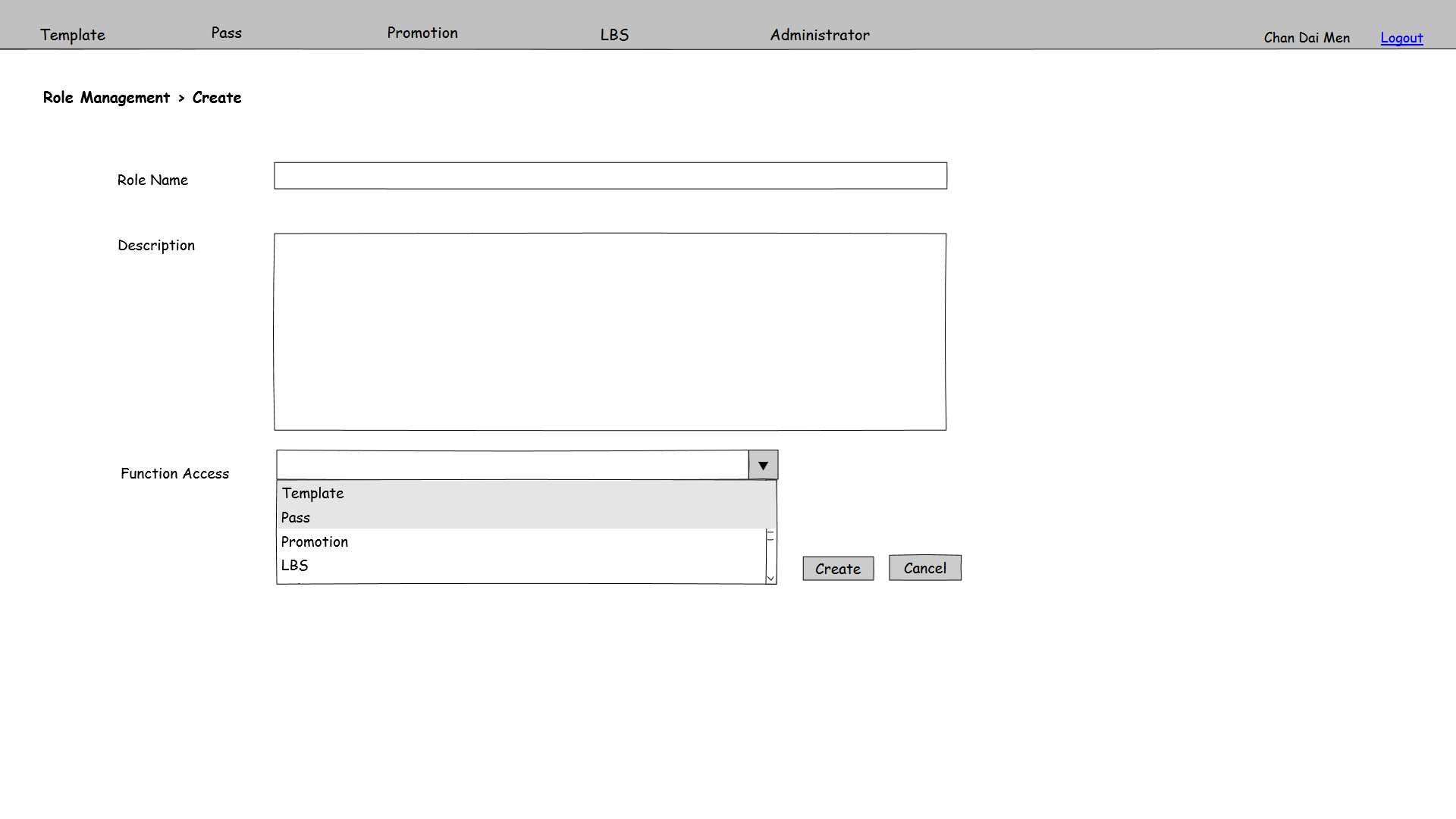


Figure 3.6.5 PMS-UAD-005

**Function Details**

Main elements/actions of the graphical user interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| Role Name | Textbox | VARCHAR(50) | Y |  |
| Description | Textbox | VARCHAR(200) | N |  |
| Function Access | Dropdown | - | - | Choose the function access, multiple choose, Selection list gets from function access table |
| Create | Button | - | - | Save the inputted parameters as a role record and go to PMS-UAD-004 |
| Cancel | Button | - | - | Back to PMS-UAD-004 |

**Flow and Actions**

1. Refer to PMS-UAD-004, click on “Create”

2. Inputted the parameter fields in this screen

3. After input all mandatory fields, press “Create” button to save the user role record and back to PMS-UAD-004

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Role Name is unique. Check duplicate on Role Name. If there is duplicate record in database, show error message “Role Name already exists. Please input other Role Name.”

3. User must choose at least one item from Function Access table. If not, show error message “At least one function must be chosen.”

### 3.6.6 User Management – Update

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-006 |
| **Function Name:** | User Management - Update |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to modify a role.

**Function Screen**

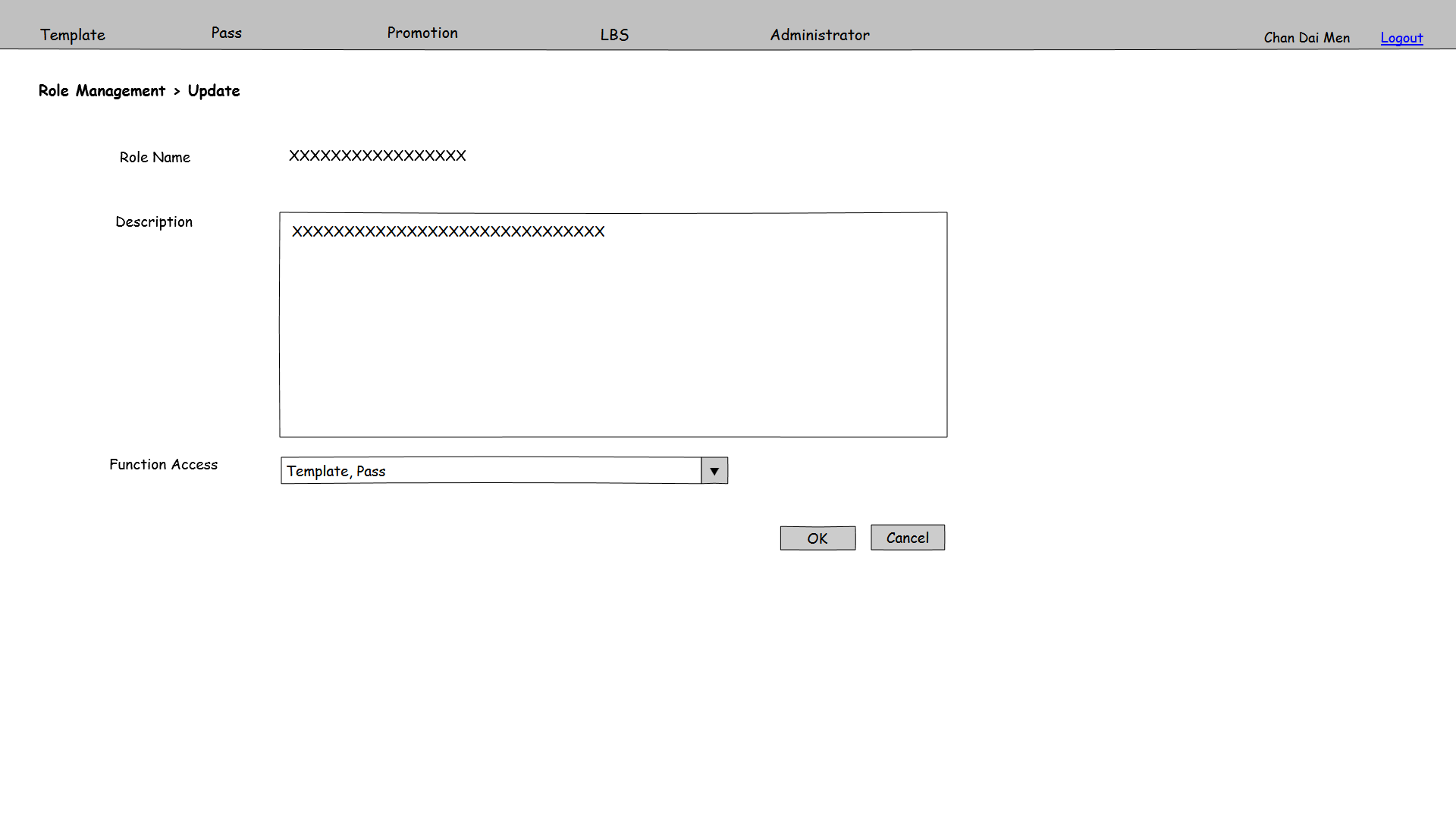


Figure 3.6.6 PMS-UAD-006

**Function Details**

Main elements/actions of the graphical user interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| Role Name | Display | - | - |  |
| Description | Textbox | VARCHAR(200) | N |  |
| Function Access | Dropdown | - | - | Choose the function access, multiple choose, Selection list gets from function access table |
| OK | Button | - | - | Save the inputted parameters as a role record and go to PMS-UAD-004 |
| Cancel | Button | - | - | Back to PMS-UAD-004 |

**Flow and Actions**

1. Refer to PMS-UAD-004, click on the hyperlink of search result

2. Inputted the parameter fields in this screen

3. After input all mandatory fields, press “OK” button to update this user role record and back to PMS-UAD-004

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. User must choose at least one item from Function Access table. If not, show error message “At least one function must be chosen.”

### 3.6.7 Email Template Management

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-007 |
| **Function Name:** | Email Template Management - Search |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

The function provides an interface to manage the email template. Users can create, edit and withdraw email template in this screen.

**Function Screen**

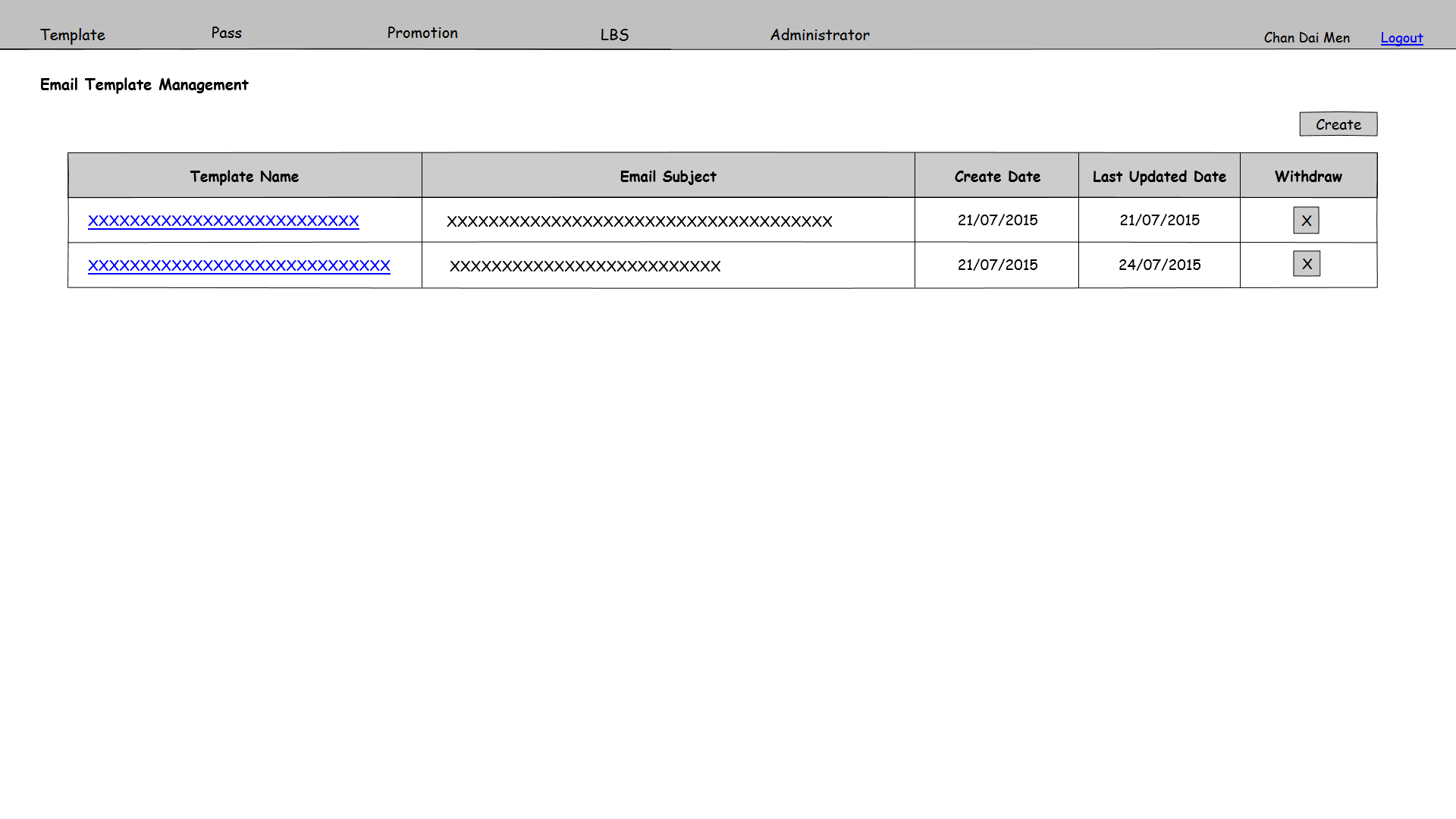


Figure 3.6.7 PMS-UAD-007

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Create | Button | - | - | Button for users to click on it and go to email template create page PMS-UAD-008 |
| Template Name | Link Button | - | - | Hyper link for users to click on it and go to user role edit page PMS-UAD-006 |
| Email Subject | Display | - | - |  |
| Create Date | Display | - | - | Format is dd/mm/yyyy |
| Last Updated Date | Display | - | - | Format is dd/mm/yyyy |
|  | Button | - | - | Prompt a confirm dialog. If click “OK”, Withdraw the corresponding record from email template table |

**Flow and Actions**

1. User click button to withdraw the email template. Then a confirmation window will prompt. After clicking “Yes”, the email template will be withdraw and the record will be deleted.



**Form Validation**

1. If click on email template using by pass templates, show error message “This email template can’t be withdraw because it is used by pass templates. “



### 3.6.8 Email Template Management – Create

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-008 |
| **Function Name:** | Email Template Management - Create |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to create an email template

**Function Screen**

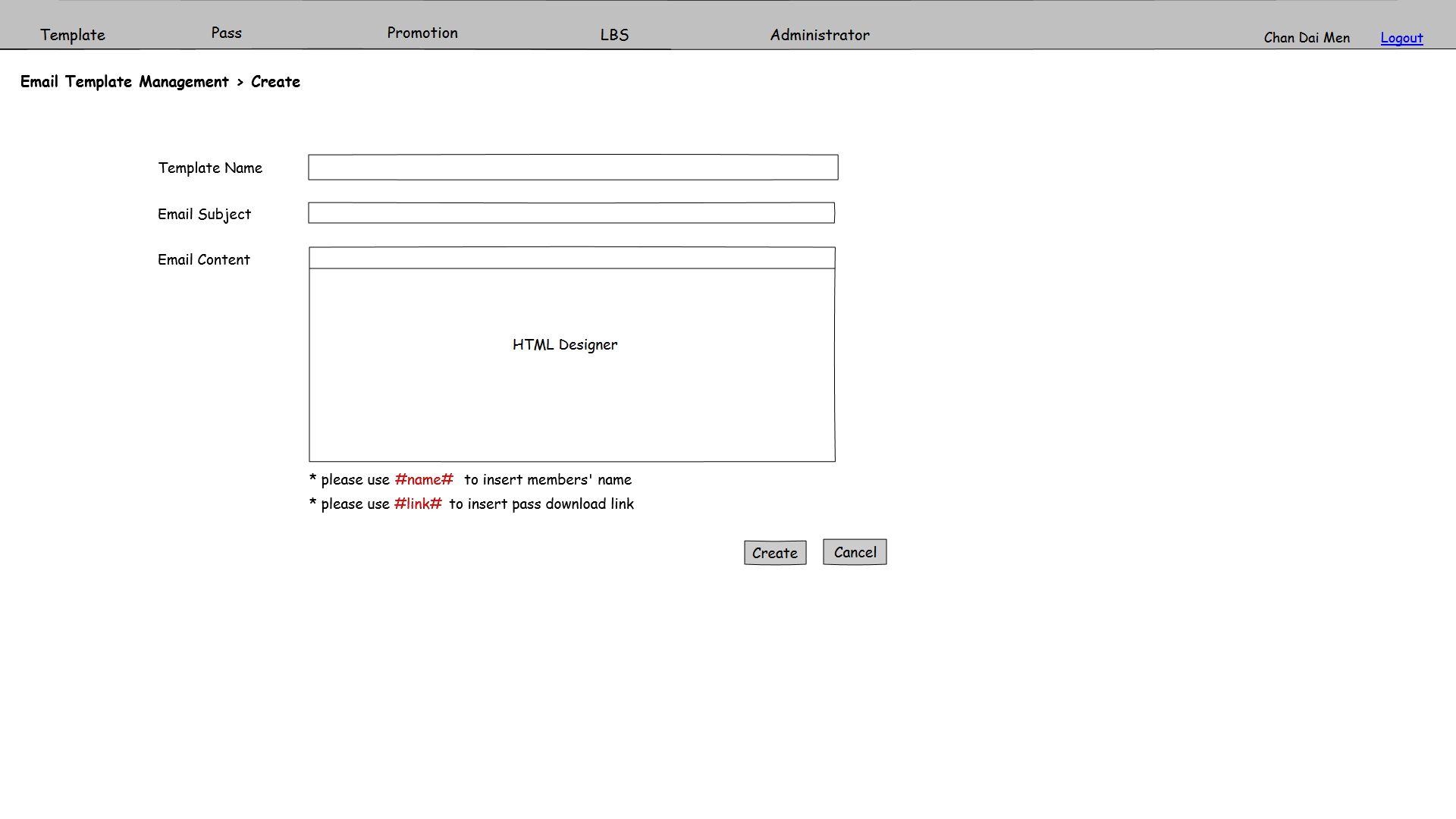


Figure 3.6.8 PMS-UAD-008

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Template Name | Textbox | VARCHAR(50) | Y |  |
| Email Subject | Textbox | NVARCHAR(200) | Y |  |
| Email Content | HTML Designer | NVARCHAR(MAX) | Y | Use #name# to insert members’ name, user #link# to insert pass download link |
| Create | Button | - | - | Save the inputted parameters as an email template and go to PMS-UAD-007 |
| Cancel | Button | - | - | Back to PMS-UAD-007 |

**Flow and Actions**

1. Refer to PMS-UAD-007, click on “Create”

2. Inputted the parameter fields in this screen

3. After input all mandatory fields, press “Create” button to save the email template record and back to PMS-UAD-007.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Template Name is unique. Check duplicate on Template Name. If there is duplicate record in database, show error message “Template Name already exists. Please input other Template Name.”

3. Email Content must have “#link#” in order to send notification to update the pass. Otherwise, show error message “Missing #link# in Email Content”.

### 3.6.9 Email Template Management – Update

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-009 |
| **Function Name:** | Email Template Management - Update |
| **Target User:** | PMS Administrator |
|  |  |

**Function Description**

This function provides an interface to modify an existing email template.

**Function Screen**

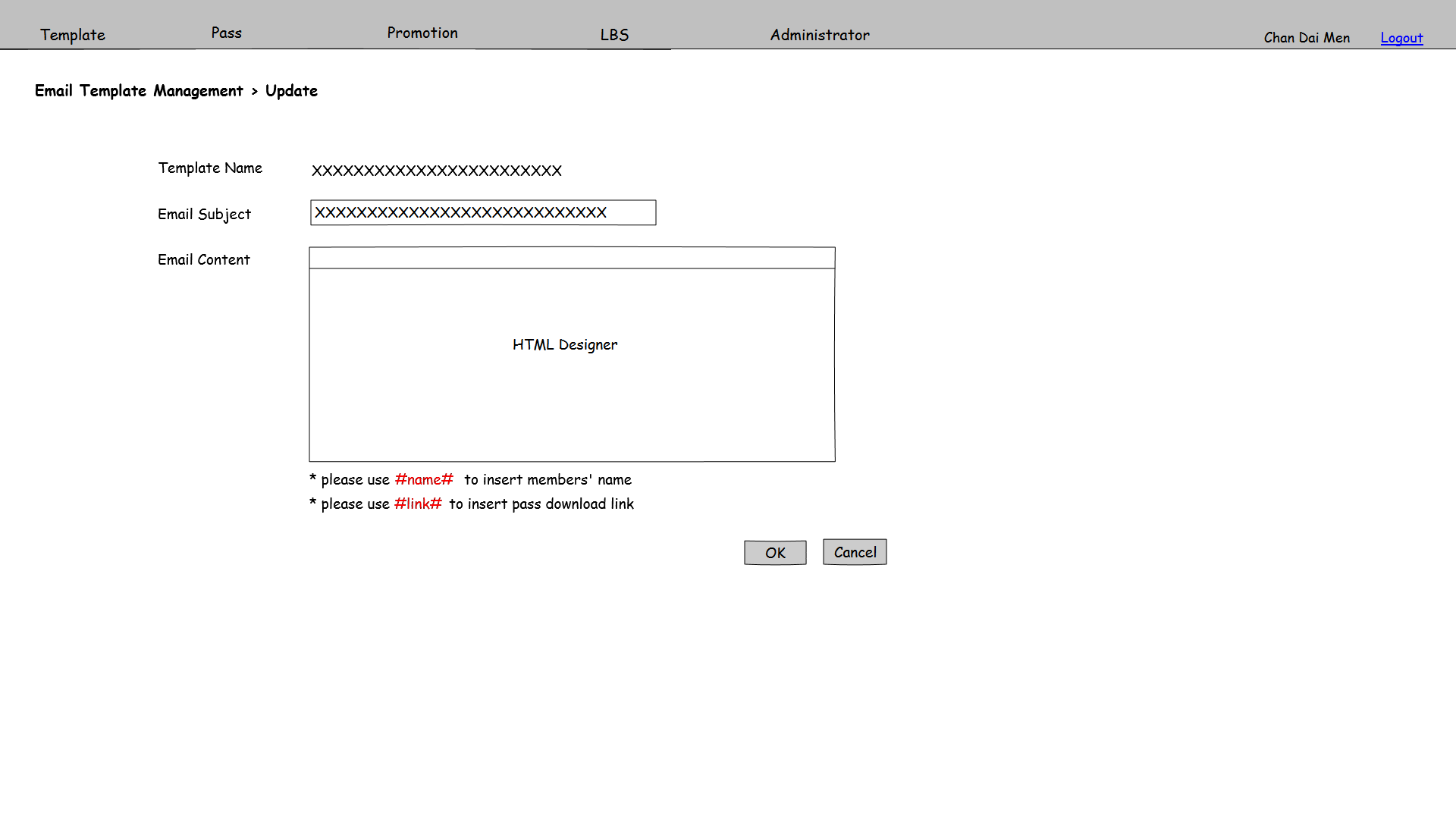


Figure 3.6.9 PMS-UAD-009

**Function Details**

Main elements/actions of the graphical user interface.

| **Field** | **Mode** | **Data Type** | **Mandatory** | **Remarks** |
| --- | --- | --- | --- | --- |
| Template Name | Display | - | - |  |
| Email Subject | Textbox | NVARCHAR(200) | Y |  |
| Email Content | HTML Designer | NVARCHAR(MAX) | Y | Use #name# to insert members’ name, user #link# to insert pass download link |
| OK | Button | - | - | Save the inputted parameters as an email template and go to PMS-UAD-007 |
| Cancel | Button | - | - | Back to PMS-UAD-007 |

**Flow and Actions**

1. Refer to PMS-UAD-007, click on hyperlink on search result

2. Inputted the parameter fields in this screen

3. After input all mandatory fields, press “OK” button to save the email template record and back to PMS-UAD-007

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.
2. Email Content must have “#link#” in order to send notification to update the pass. Otherwise, show error message “Missing #link# in Email Content”.

### 3.6.10 Landing Page Management

|  |  |
| --- | --- |
| **Function ID:** | FN-UAD-010 |
| **Function Name:** | Landing Page Management |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface to modify the pass download landing page.

**Function Screen**

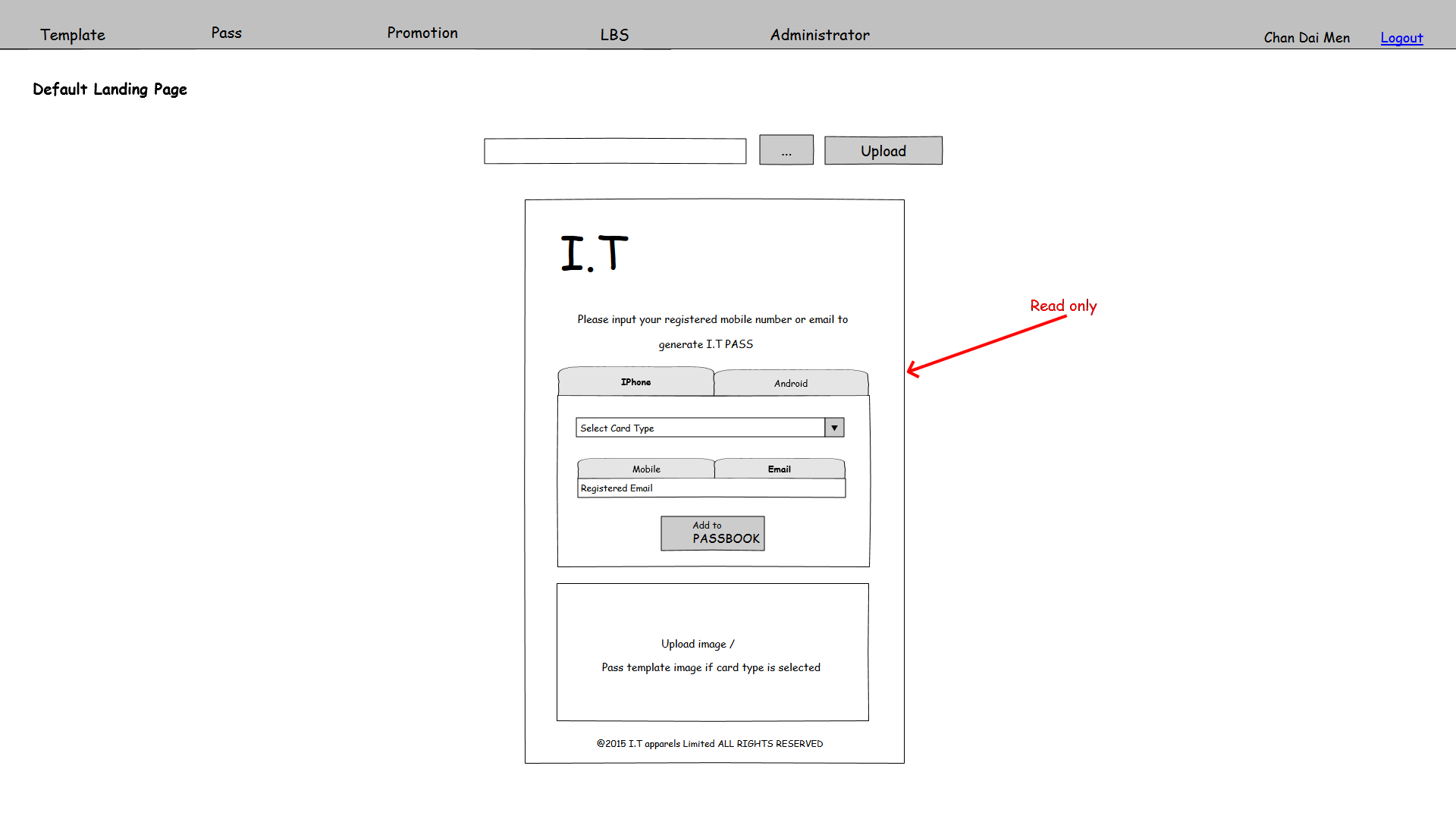


Figure 3.6.10 PMS-UAD-010

**Function Details**

Main elements/actions of the graphical user interface.

**Flow and Actions**

1. Attach an image of default landing page by using the and “Upload” button. The image will show at the landing page Image field once the pass image is uploaded.



**Form Validation**

1. The file input in upload field should be in image type and maximum file size is in 500KB. Otherwise, show error message “Invalid file format. Image file must be in proper image format and size smaller than 500KB.”

## 3.7 Landing Page

### 3.7.1 Landing Page

|  |  |
| --- | --- |
| **Function ID:** | FN-PLP-001 |
| **Function Name:** | Landing Page |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides an interface of the pass download landing page.

**Function Screen**

Figure 3.7.1a PMS-PLP-001

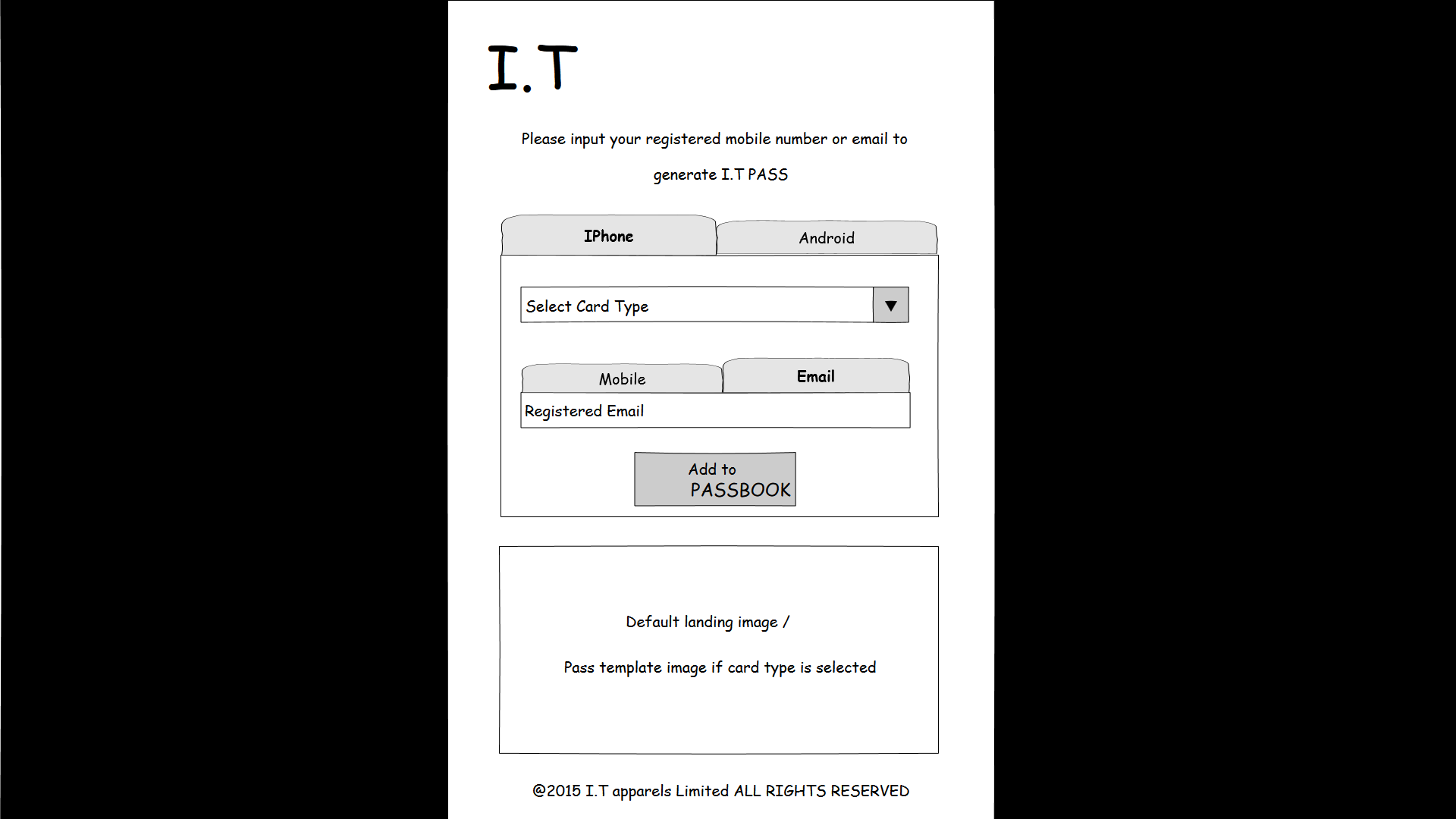


Figure 3.7.1b PMS-PLP-001

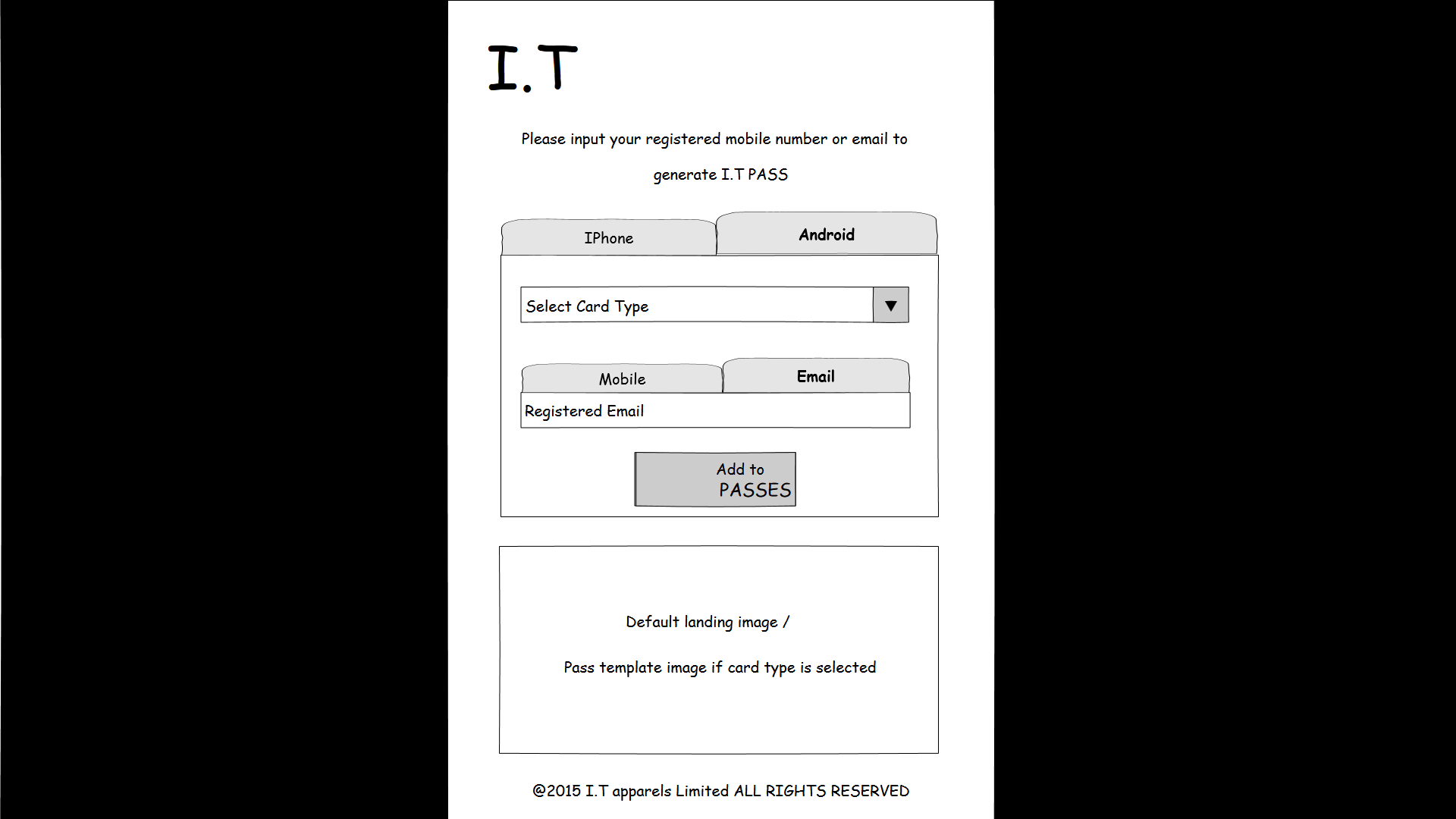
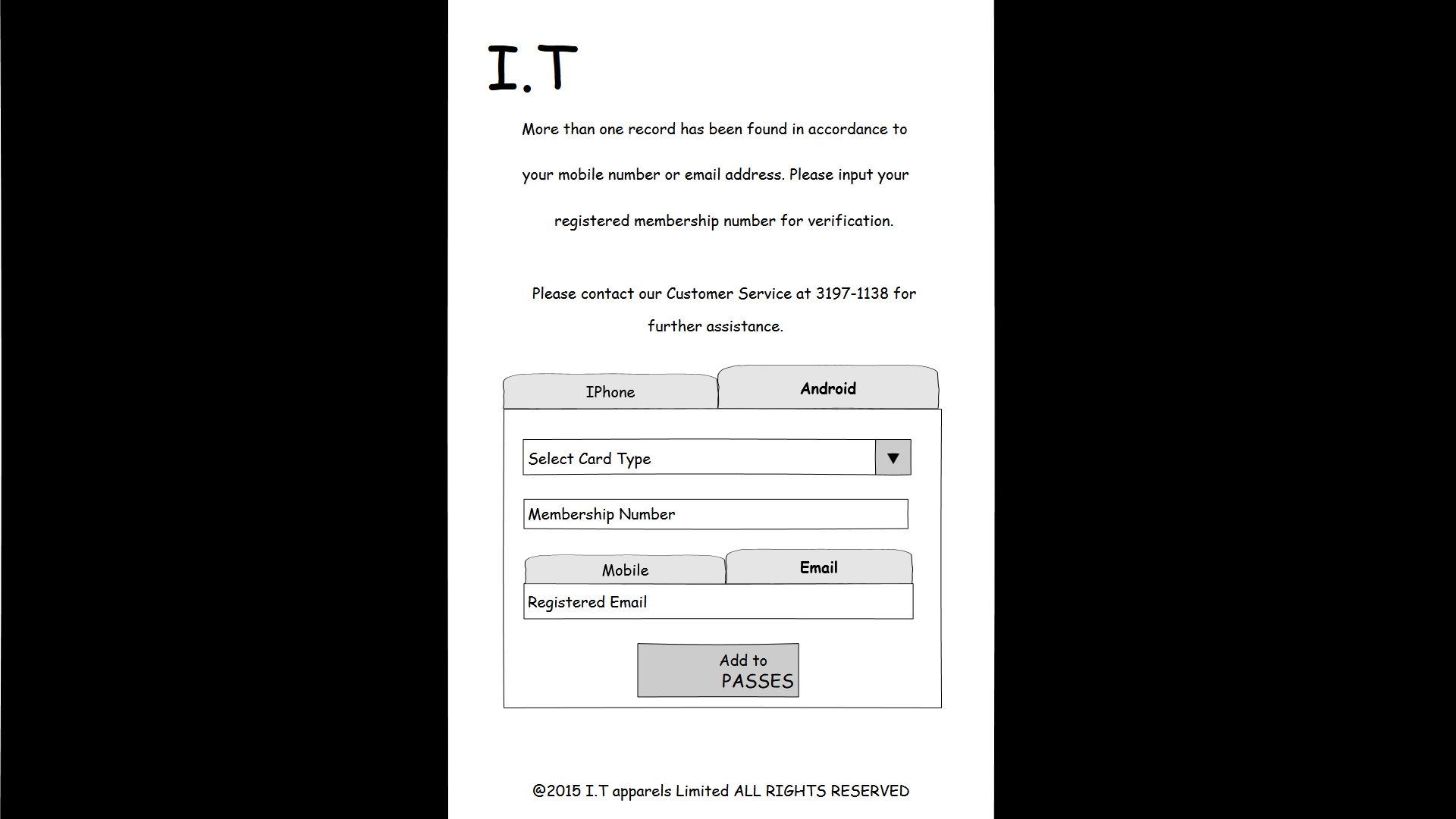
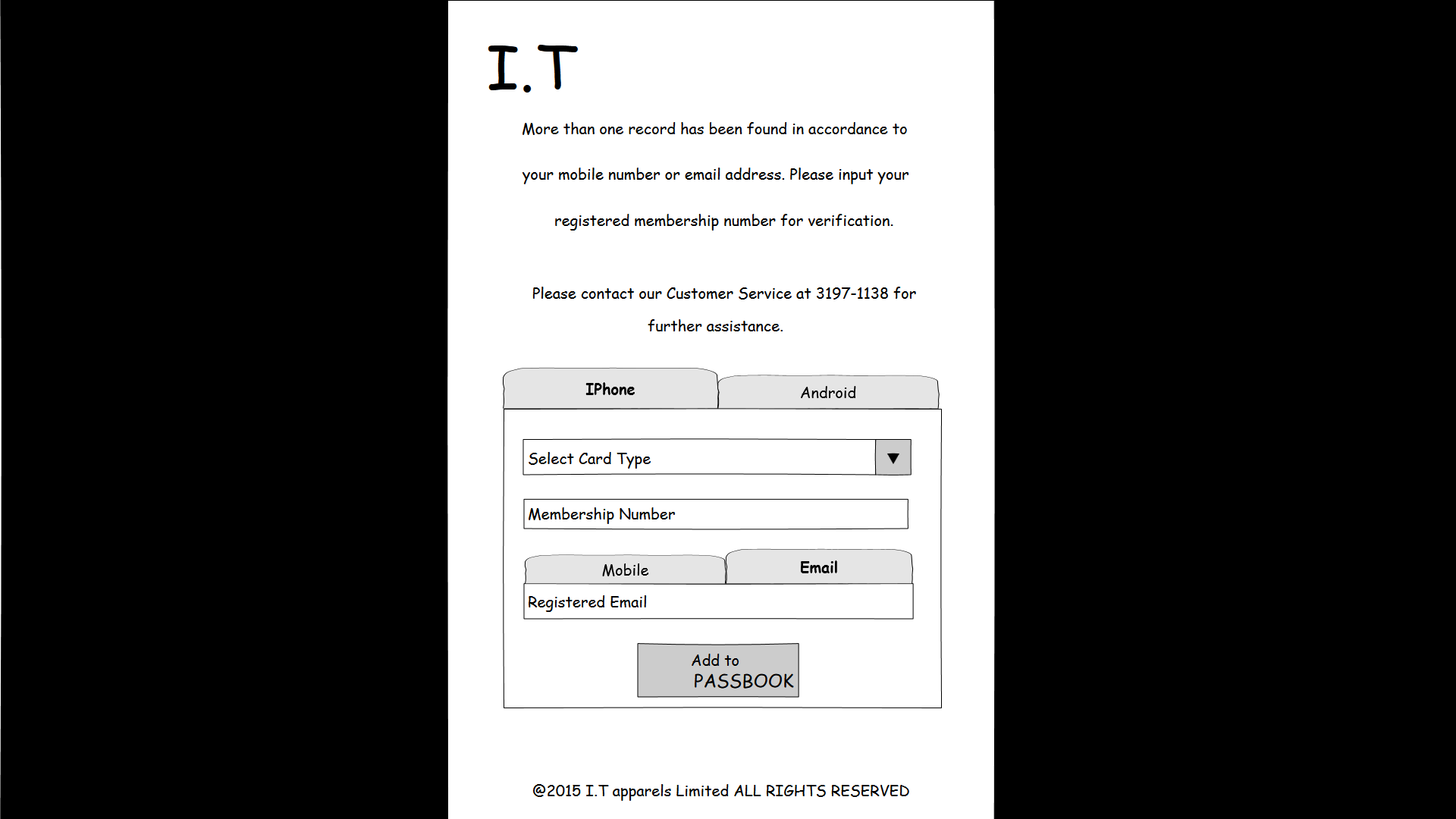


Figure 3.7.1c PMS-PLP-002Figure 3.7.1d PMS-PLP-002



**Function Details**

Main elements/actions of the graphical user interface.

**Flow and Actions**

1. Select target device.

For user select IPhone, the landing page screen will show as figure 3.7.1a.

For user select Android, the landing page screen will show as figure 3.7.1b.

2. Select card type, and the pass template image will show in the image area.

3. Input registered mobile no or registered email.

4. Click on the button “Add to PASSBOOR” or “Add to PASSES”. Pass download will start after click button.

5. If more than one record has been found in accordance to the mobile number or email address inputted, it will turn to screen as Figure 3.7.1c or Figure 3.7.1d shown. User should input membership number before click on the button “Add to PASSBOOR” or “Add to PASSES”.

**Form Validation**

1. All mandatory fields must be filled. Otherwise, show error message “Missing XXX”, which XXX stands for the mandatory field without input.

2. Check email format. Email address must have character “@”. Otherwise, show error message “Invalid email address”.

3. Registered Mobile No or Email Address must match with CRM record.

1. **Web Services**

## 4.1 Create Pass Web Service for Android Device

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-001 |
| **Function Name:** | Create Pass Web Service for Android Device |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides a web service for other functions to create Android device pass. It will use Pass2U APIs to create passes with members’ information.

**Function Details**

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| GET | <webServiceURL>/api/CreatePassForAndroid/<TemplateName> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | TemplateName | string |
| Payload | Passes data in JSON format. It is dynamic and decided by template.  An example is:  {  “Pass1”: {  “Bonus”: “1000”,  “Next Level”: “500”  }  “Pass2”: {  “Bonus”: “1200”,  “Next Level”: “300”  }  } | string |

**Response**

| **Status** | **Response** |
| --- | --- |
| 200 | Success  Return HTTP status 200 with a payload of the pass data (pkpass file) |
| 400 | Network Error. |
| 401 | Request is not authorized |
| 402 | Invalid Model ID |
| 403 | Invalid pass data |
| 404 | URL can’t not be found |

**Flow and Actions**

1. Get the Pass.JSON file of template with template name in request URL parameters.
2. Find the model ID in the Pass.JSON file for Android passes creation.
3. Use Pass2U API – ‘Create A Pass Distribution” to create passes with URL payload.
4. Use Pass2U API – “Get A .pkpass File” to return the created passes.

**Form Validation**

N/A

## 4.2 Create Pass Web Service for Apple Device

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-002 |
| **Function Name:** | Create Pass Web Service for Apple Device |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides a web service for other functions to create Apple device pass. It will use function 4.1.3 to create passes with members’ information.

**Function Details**

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| GET | <webServiceURL>/api/CreatePassForApple/<TemplateName> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | TemplateName | string |
| Payload | Passes data in JSON format. It is dynamic and decided by template.  An example is:  {  “Pass1”: {  “Bonus”: “1000”,  “Next Level”: “500”  }  “Pass2”: {  “Bonus”: “1200”,  “Next Level”: “300”  }  } | string |

**Response**

| **Status** | **Response** |
| --- | --- |
| 200 | Success  Return HTTP status 200 with a payload of the pass data (pkpass file) |
| 400 | Network Error. |
| 401 | Request is not authorized |
| 402 | Invalid Model ID |
| 403 | Invalid pass data |
| 404 | URL can’t not be found |

**Flow and Actions**

1. Get the Pass.JSON file of template with template name in request URL parameters.
2. Use URL payload to create pass folders (including Pass.JSON and image in template) for every pass.
3. Sign passes using “Sign Pass Web Service for Apple Device” and return the pkpass files.

**Form Validation**

N/A

## 4.3 Sign Pass Web Service for Apple Device

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-003 |
| **Function Name:** | Sign Pass Web Service for Apple Device |
| **Target User:** | All users |
|  |  |

**Function Description**

This function provides a web service on Mac OS for function 4.1.2 to sign pass file and get the pkpass file.

**Function Details**

**Flow and Actions**

N/A

**Form Validation**

N/A

## 4.4 Push Notification Web Services

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-001 |
| **Function Name:** | Push Notification Web Services |
| **Target User:** | All User |

**Function Description**

This function provides a web service for other functions when there are updates on passes. After updating function (updating bonus point, spending, promotion, LBS) started, this function will push a notification to APNs for telling the passes that should be updated to communicate with the Apple Pass Server.

**Function Details**

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| POST | <webServiceURL>/api/PushNotification/<pushToken> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | pushToken | string |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | Success. The request has sent to APNs. |
| 401 | Authentication failed. |
| 402 | Fail to connect with APNs. Invalid pushToken. |
| 403 | Fail to connect with APNs. Network error. |

**Flow and Actions**

1. When the pass is updated, the system will send this request to APNs.

**Form Validation**

N/A

## 4.5 Apple Passbook Server

### 4.5.1 Apple Device Register Web Service

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-005 |
| **Function Name:** | Apple Device Register Web Service |
| **Target User:** | All Users |

**Function Description**

This function will provide a web service for Apple device register when the user open the pass. When the pass holder open the pass first time, this function will be called by the Apple device and send the device information that needed for pass update to the database.

**Function Details**

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| POST | <webServiceURL>/<version>/devices/<deviceLibraryIdentifier>/registrations/<passTypeIdentifier>/<SerialNumber> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| HEAD | authorizationToken | string |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | version | string |
| URL\_PARAM | deviceLibraryIdentifier | string |
| URL\_PARAM | passTypeIdentifier | string |
| URL\_PARAM | serialNumber | string |
| POST payload | pushToken | string |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | Device already registered, save the pass and device relationship into the database. |
| 201 | Success, save the device information and relationship of pass and device into the database. |
| 401 | Request is not authorized |
| 400 | Invalid database connection |
| 404 | URL can’t not be found |

**Flow and Actions**

1. When the pass holders click and open a pass on Apple device, the Passbook app will send this request to Apple Passbook Server.
2. When the Apple Passbook Server receives this request, it will save the device and pass records into database and return a corresponding response to the Passbook app which sends this request.

**Form Validation**

N/A

### 4.5.2 Pass Update Web Services for Apple Device

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-006 |
| **Function Name:** | Pass Update Web Services for Apple Device |
| **Target User:** | All Users |

**Function Description**

This function will provide a web service for pass update. When the pass information is updated, the update functions (updating bonus, spending, promotion or LBS) will push a notification to APNs (Function 4.4) and tell the device that is registered in Function 4.5.1 to update the corresponding pass. Then the device will call this web service to communicate with database to update to latest pkpass.

**Function Details**

1) Getting the Serial Numbers for Passes Associated with a Device

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| GET | <webServiceURL>/<version>/devices/<deviceLibraryIdentifier>/registrations/<passTypeIdentifier>?passesUpdatedSince=<tag> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | version | string |
| URL\_PARAM | deviceLibraryIdentifier | string |
| URL\_PARAM | passTypeIdentifier | string |
| URL\_PARAM | tag | string |

**Response**

| **Status** | **Response** |
| --- | --- |
| 200 | Matching passes are found.  Return HTTP status 200 with a JSON dictionary with lastUpdated (string, current modification tag) and serialNumbers (array of strings, the serial numbers of the matching passes)  An example response is:  {  “lastUpdated”: “29/07/2015 10:00:00”,  “serialNumbers”: [“uP23jsajs123”, “09xji123Kla1”, “nm456hj891df”]  } |
| 204 | No matching passes |
| 300 | The pass is the latest version. No need to update. |
| 400 | Invalid database connection |
| 404 | URL can’t not be found |

2) Getting the Latest version of a Pass

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| GET | <webServiceURL>/<version>/passes/<passTypeIdentifier>/<serialNumber> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| HEAD | authorizationToken | string |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | version | string |
| URL\_PARAM | passTypeIdentifier | string |
| URL\_PARAM | serialNumber | string |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | Success  Return HTTP status 200 with a payload of the pass data (pkpass file) |
| 400 | Invalid database connection |
| 401 | Request is not authorized |
| 404 | URL can’t not be found |

**Flow and Actions**

1. When the Apple device receive a update notification sent from “Push Notification Web Service” from APNs, the Passbook app will send a request to get the serial numbers for passes associated with this device.
2. The Apple Passbook server will return a list of passes with the serial numbers received from the device which last update tag is larger than tag sent in the request.
3. When the Apple device received this response, the Passbook app will compare the LastUpdated with last update date in each pass. If LastUpdated is larger than last update date, the Passbook app will send a request to get the latest version of a pass
4. The Apple Passbook Server will return a pkpass file to the apple device if it receive a request from Apple device sent in step 3.

**Form Validation**

N/A

### 4.5.3 Pass Delete Web Services for Apple Device

|  |  |
| --- | --- |
| **Function ID:** | FN-WSS-007 |
| **Function Name:** | Pass Delete Web Services for Apple Device |
| **Target User:** | All Users |

**Function Description**

This function will provide a web service for pass delete. When the pass is deleted by pass holder, the device will call this web service to tell database to delete the pass information.

**Function Details**

Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| DELETE | <webServiceURL>/<version>/devices/<deviceLibraryIdentifier>/registrations/  <passTypeIdentifier>/<SerialNumber> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Parameters** | **Values** |
| HEAD | authorizationToken | string |
| URL\_PARAM | webServiceURL | string |
| URL\_PARAM | version | string |
| URL\_PARAM | deviceLibraryIdentifier | string |
| URL\_PARAM | passTypeIdentifier | string |
| URL\_PARAM | serialNumber | string |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | Success, delete the pass and device relationship in the database. |
| 401 | Request is not authorized |
| 400 | Invalid database connection |
| 404 | URL can’t not be found |

**Flow and Actions**

1. When the pass holders delete a pass on Apple device, the Passbook app will send this request to Apple Passbook Server.
2. When the Apple Passbook Server receives this request, it will mark the device and pass records into as deleted in database and return a corresponding response to the Passbook app which sends this request.

**Form Validation**

N/A

1. **Batch Jobs**

## 5.1 Update Bonus Point of Member Cards

|  |  |
| --- | --- |
| **Function ID:** | FN-BJS-001 |
| **Function Name:** | Update Bonus Point for Member Cards |
| **Run Mode:** | Daily |
|  |  |

**Function Description**

This function will run once a day to update the bonus point for all member cards. It will get the latest bonus point and call the corresponding update web services (Apple or Android device) for member cards updating.

**Function Details**

Input Parameters

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| Member ID | string |
| Bonus | Integer |

**Flow and Actions**

1. After the bonus points are updated in CRM, this batch job will start for updating bonus point of member cards.
2. Get the member ID and bonus list from CRM. Then the batch job will search the passes in detail for every member in member card type.
3. According to different devices registered for the passes, the batch job will call the corresponding update web service.

For Apple device, call Push Notification Web Service for Apple Device

For Android device, call Update Passes Web Service for Android Device

**Form Validation**

N/A

## 5.2 Update Spending of Primary Member Cards

|  |  |
| --- | --- |
| **Function ID:** | FN-BJS-002 |
| **Function Name:** | Update Spending of Primary Member Cards |
| **Run Mode:** | Daily |
|  |  |

**Function Description**

This function will run once a day to update the spending for all primary member cards. It will get the latest spending and call the corresponding update web services (Apple or Android device) for primary member cards updating.

**Function Details**

Input Parameters

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| Member ID | string |
| Spending | Double |

**Flow and Actions**

1. After the spending are updated in CRM, this batch job will start for updating spending of member cards.
2. Get the member ID and spending list from CRM. Then the batch job will search the passes in detail for every member in member card type.
3. According to different devices registered for the passes, the batch job will call the corresponding update web service.

For Apple device, call Push Notification Web Service for Apple Device

For Android device, call Update Passes Web Service for Android Device

**Form Validation**

N/A

## 5.3 Email Notification (For Model Withdraw)

|  |  |
| --- | --- |
| **Function ID:** | FN-BJS-003 |
| **Function Name:** | Email Notification (For Model Withdraw) |
| **Run Mode:** | Ad-hoc |
|  |  |

**Function Description**

When the model is withdraw, the system will send an email notification to all users with passes created by this model. The email notification will tell the pass holders that the pass is invalid and should be deleted from Passbook.

**Function Details**

Input Parameters

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| Model Name | string |

**Flow and Actions**

1. After the model is withdraw, this batch job will start for sending an email notification.
2. This batch job will search the templates with the model ID. Then it will get the passes list (with email) needed withdraw.
3. Send email notifications to all users for telling that the pass is invalid and should be deleted from Passbook.

**Form Validation**

N/A

## 5.4 Update Pass by General / Personal Promotion

|  |  |
| --- | --- |
| **Function ID:** | FN-BJS-004 |
| **Function Name:** | Update Pass by General / Personal Promotion |
| **Run Mode:** | Ad-hoc |

**Function Description**

When clicking “OK” in Create or Update Promotion function (Function 3.4.2 or 3.4.3), the system will call this function to update the passes selected in PMS screen by inputted fields.

**Function Details**

Input Parameters

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| Promotion Name | string |

**Flow and Actions**

1. After the promotion is created or updated, this batch job will start for updating backfields of passes.
2. This batch job will search the templates that using this promotion. Then it will get the passes list needed update with the templates’ detail.
3. According to different devices registered for the passes, the batch job will call the corresponding update web service.

For Apple device, call Push Notification Web Service for Apple Device

For Android device, call Update Passes Web Service for Android Device

**Form Validation**

N/A

## 5.5 Update Pass for LBS Information

|  |  |
| --- | --- |
| **Function ID:** | FN-BJS-005 |
| **Function Name:** | Update Pass for LBS Information |
| **Run Mode:** | Ad-hoc |

**Function Description**

When clicking “OK” in Create or Update LBS function (Function 3.5.2 or 3.5.3), the system will call this function to update the passes selected in PMS screen by inputted fields.

**Function Details**

Input Parameters

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| LBS Name | string |

**Flow and Actions**

1. After the LBS is created or updated, this batch job will start for updating backfields of passes.
2. This batch job will search the templates that using this LBS. Then it will get the passes list needed update with the templates’ detail.
3. According to different devices registered for the passes, the batch job will call the corresponding update web service.

For Apple device, call Push Notification Web Service for Apple Device

For Android device, call Update Passes Web Service for Android Device

**Form Validation**

N/A

## 5.6 Update Pass for Model Update

|  |  |
| --- | --- |
| **Function ID:** | FN-BJS-006 |
| **Function Name:** | Update Pass for Model Update |
| **Run Mode:** | Ad-hoc |

**Function Description**

When clicking “OK” in Update Template function (Function 3.2.3), the system will call this function to update the passes selected in PMS screen by inputted fields.

**Function Details**

Input Parameters

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| Template Name | string |

**Flow and Actions**

1. After the template is updated, this batch job will start for updating layout of passes.
2. This batch job will search the templates that using this template. Then it will get the passes list needed update with the templates’ detail.
3. According to different devices registered for the passes, the batch job will call the corresponding update web service.

For Apple device, call Push Notification Web Service for Apple Device

For Android device, call Update Passes Web Service for Android Device

**Form Validation**

N/A

**Appendix A – Database Schema**

For database schema, please refer to

**Appendix B – Report Template**

For report template, please refer to