

Requirement & Design Specification

Birthday party booking for Kids (BPBFK)

**Version: 1.0**

– HoChiMinh, March 2024 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| V1.0 | 15/3 | A | Use Case Diagram | Add Use Case Diagram |
| V1.0 | 15/3 | A | Database design | Add database design |
| V1.0 | 15/3 | A | Table Descriptions | Add Table Description |
| V1.0 | 15/3 | A | Use case template | Add Use Case Template |
| V1.0 | 15/3 | A | UC01 - Register | Add UC01 - Register |
| V1.0 | 15/3 | A | UC02 - Login | Add UC02 - Login |
| V1.0 | 15/3 | A | UC03 - Login with Google | Add UC03 - Login with Google |
| V1.0 | 15/3 | A | UC04 - Edit Profile | Add UC04 - Edit Profile |
| V1.0 | 15/3 | A | UC05 - Create Party | Add Create Party |
| V1.0 | 15/3 | A | UC06 - View Party Packages | Add UC06 - View Party Packages |
| V1.0 | 15/3 | A | UC07 - Update Party Package | Add UC07 - Update Party Package |
| V1.0 | 16/3 | A | UC08 - Delete Party Package | Add UC08 - Delete Party Package |
| V1.0 | 16/3 | A | UC09 - Create Service | Add UC09 - Create Service |
| V1.0 | 16/3 | A | UC10 - View Services | Add UC10 - View Services |
| V1.0 | 16/3 | M | Use Case Diagram | Edit Use Case Diagram |
| V1.0 | 16/3 | A | UC11 - View Parties | Add UC11 - View Parties |
| V1.0 | 16/3 | A | Usability | Add Usability |
| V1.0 | 16/3 | A | Reliability | Add Reliability |
| V1.0 | 16/3 | A | Performance | Add Performance |
| V1.0 | 16/3 | A | UC12 - View Party Detail | Add UC12 - View Party Detail |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Table of Contents

[Record of Changes 2](#_heading=h.30j0zll)

[I. Overview 4](#_heading=h.1fob9te)

[1. User Requirements 4](#_heading=h.3znysh7)

[1.1 Actors 4](#_heading=h.2et92p0)

[1.2 Use Cases 4](#_heading=h.tyjcwt)

[a. Diagram(s) 4](#_heading=h.kgcv8k)

[2. Overall Functionalities 5](#_heading=h.34g0dwd)

[2.1 Screens Flow 5](#_heading=h.1jlao46)

[2.2 Screen Descriptions 5](#_heading=h.43ky6rz)

[2.3 Screen Authorization 5](#_heading=h.2iq8gzs)

[2.4 Non-UI Functions 6](#_heading=h.xvir7l)

[3. System High Level Design 6](#_heading=h.3dy6vkm)

[3.1 Database Design 6](#_heading=h.4d34og8)

[II. Requirement Specifications 8](#_heading=h.17dp8vu)

[1. <<Feature Name>> 8](#_heading=h.3rdcrjn)

[1.1 <<UseCaseCode\_UC Name>> 8](#_heading=h.26in1rg)

[2. Common Functions 9](#_heading=h.lnxbz9)

[2.1 UC01 - Register 9](#_heading=h.35nkun2)

[III. Design Specifications 10](#_heading=h.4bvk7pj)

[1. <<Feature Name>> 10](#_heading=h.2r0uhxc)

[1.1 <<SubFeature Name>> 10](#_heading=h.1664s55)

[1.2 System Access 11](#_heading=h.3q5sasy)

[IV. Appendix 15](#_heading=h.3hv69ve)

[1. Assumptions & Dependencies 15](#_heading=h.1x0gk37)

[2. Limitations & Exclusions 15](#_heading=h.4h042r0)

[3. Business Rules 15](#_heading=h.2w5ecyt)

[4. .. 15](#_heading=h.25b2l0r)

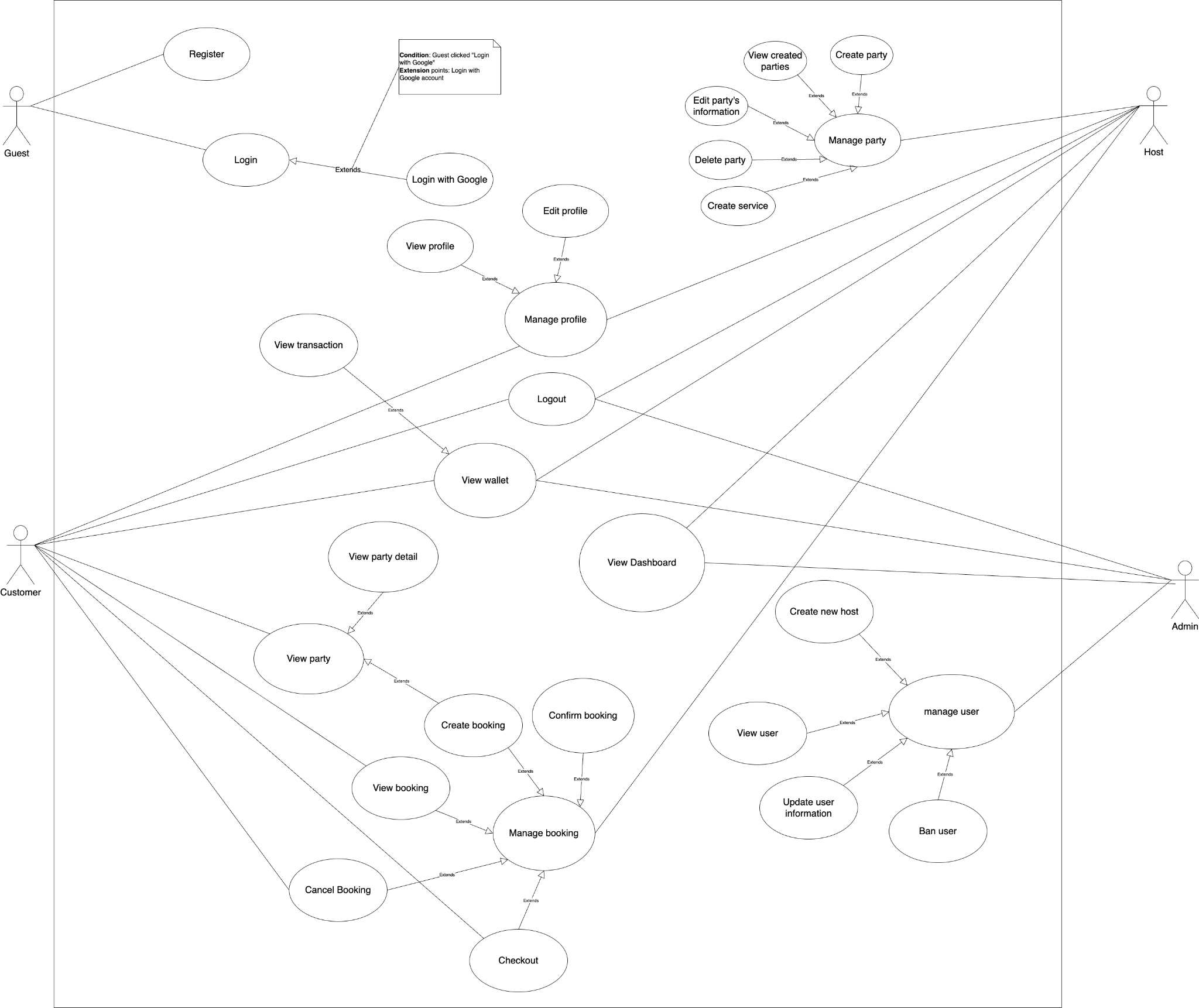
# I. Overview

## 1. User Requirements

### 1.1 Actors

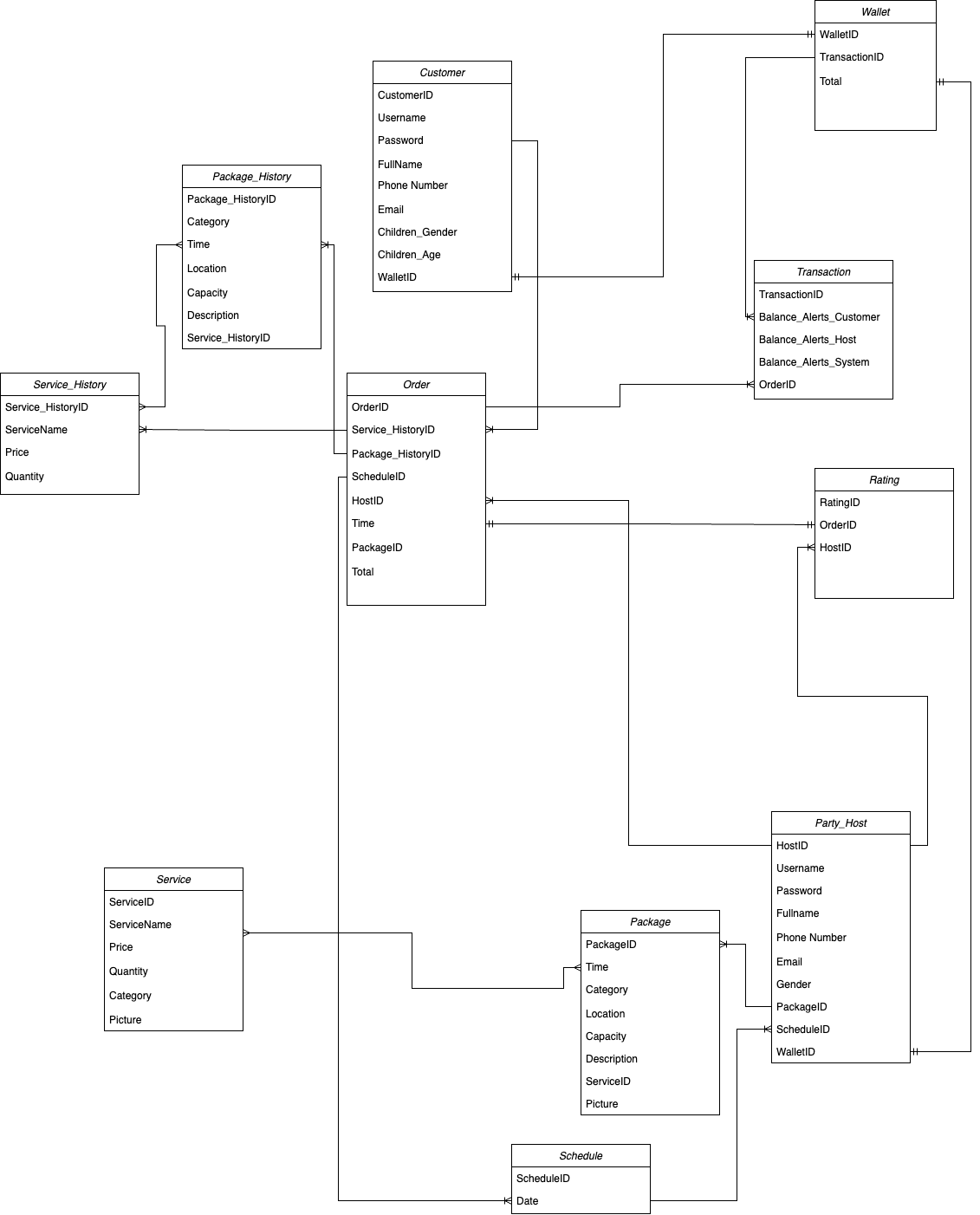
|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Admin | The person who manages the overall system, including user accounts, party packages, venue availability, and bookings. - Can add, edit, and delete information about party packages, venues, and system settings. - May access reports and manage financial transactions. |
| 2 | Host | The person who can create new party packages, be provided with an account from the admin, create, delete, and edit their own party packages and manage financial transactions. |
| 3 | Customer | The person who who can view and book a party for a specific period of time and can manage financial transactions. |
| 4 | Guest | The person who visits the website. - May not have a direct role in the system. - Can view the home page and preview party packages without logging in |

### 1.2 Use Cases Diagram

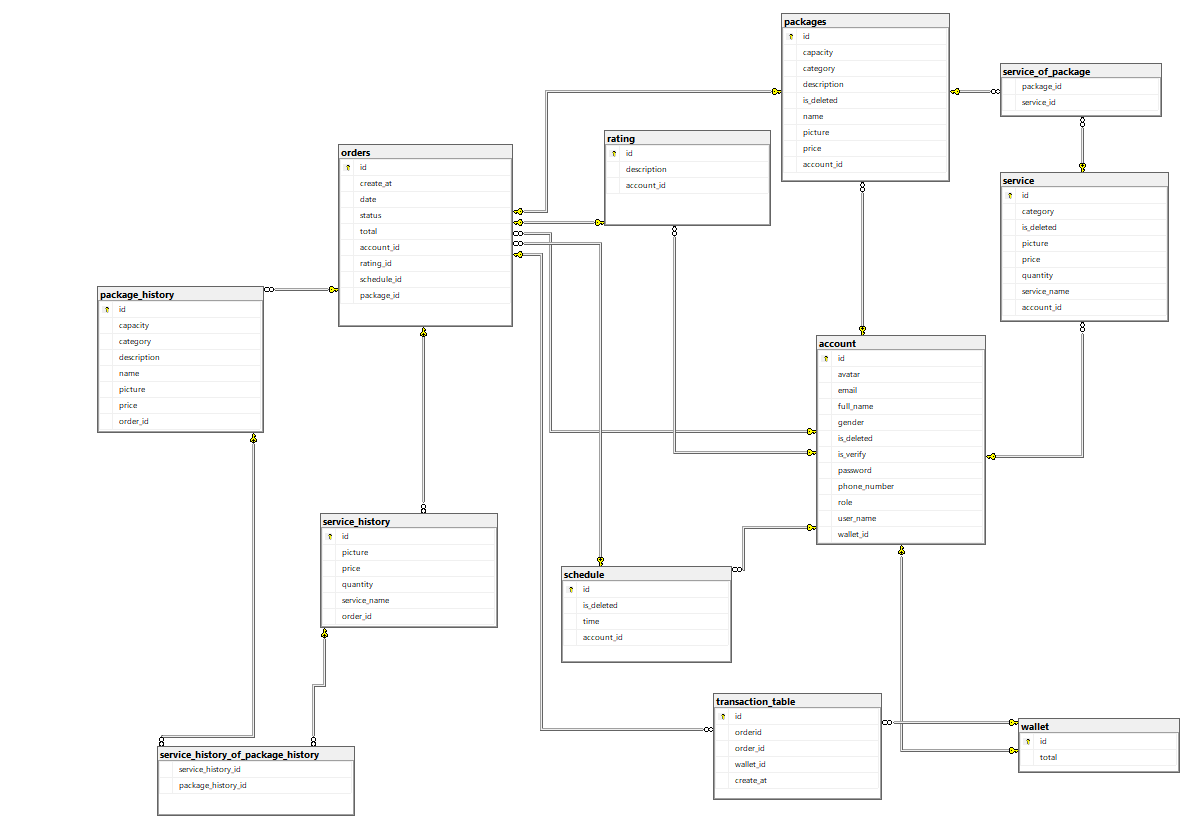


## 3. System High Level Design

### 3.1 ERD



### 3.2 Database Design



### 3.3 Table Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | Account | * Id (Number: 1 – max characters) * Username (Text: 2 – 30 characters) * Password (Text: 8 – max characters) * Role (Admin/Host/Customer) * Full name (Text: 2 – max characters) * Gender (Male/Female) * Phone number (Text: 9 – 12 characters) * Avatar (Link) * Wallet\_ID (Number: 1 – max characters) * Is\_Verify (True/False) * Is\_Deleted (True/False) |
| 02 | Order | * Id (Number: 1 – max characters) * Create\_at (Time) * Date (Time) * Status (Number: 1 characters) * Total (Number: 1 – max characters) * Account\_ID (Number: 1 – max characters) * Rating\_ID (Number: 1 – max characters) * Schedule\_ID (Number: 1 – max characters) * Package\_ID (Number: 1 – max characters) |
| 03 | Package | * Id (Number: 1 – max characters) * Name (Text: 1 – 255 characters) * Price (Number: 1 – max characters) * Picture (Link) * Capacity (Number: 1 – 255 characters) * Category (Number: 1 – 10 characters) * Description (Text: 1 – 255 characters) * Is\_Deleted (True/False) * Account\_ID (Number: 1 – max characters) |
| 04 | Package\_History | * Id (Number: 1 – max characters) * Name (Text: 1 – 255 characters) * Price (Number: 1 – max characters) * Picture (Link) * Capacity (Number: 1 – 255 characters) * Category (Number: 1 – 10 characters) * Description (Text: 1 – 255 characters) * Is\_Deleted (True/False) * Order\_ID (Number: 1 – max characters) |
| 05 | Service | * Id (Number: 1 – max characters) * Name (Text: 1 – 255 characters) * Price (Number: 1 – max characters) * Category (Number: 1 – 10 characters) * Picture (Link) * Is\_Deleted (True/False) * Quantity (Number: 1 – 255 characters) * Account\_ID (Number: 1 – max characters) |
| 06 | Service\_History | * Id (Number: 1 – max characters) * Name (Text: 1 – 255 characters) * Price (Number: 1 – max characters) * Picture (link) * Quantity (Number: 1 – 255 characters) * Order\_ID (Number: 1 – max characters) |
| 07 | Rating | * Id (Number: 1 – max characters) * Description (Text: 1 – max characters) * Account\_ID (Number: 1 – max characters) |
| 08 | Schedule | * Id (Number: 1 – max characters) * Time (Time) * Is\_Deleted (True/False) * Account\_ID (Number: 1 – max characters) |
| 09 | Wallet | * Id (Number: 1 – max characters) * Total (Number: 1 – max characters) |
| 10 | Transaction | * Id (Number: 1 – max characters) * Order\_ID (Number: 1 – max characters) * Wallet\_ID (Number: 1 – max characters) * Create\_at (Time) |

## 3. Non - Functional Requirement

### 3.1 Usability

- All text, label and message should be uniformly written in Vietnamese.

- The application should be friendly and easy for users to use without training.

- The web application for admin should require no more than 2 days of training of use.

### 3.2 Reliability

**- Availability:**

* The website should be available almost all the time, at least 99.99% of the time when people usually use it.
* If there's any planned maintenance, users will be told about it 2 days ahead, and it will be done when fewer people are using the website.

**- Mean Time Between Failures (MTBF):**

* The website should work without any major problems for a long time, around 5,000 hours.
* The important things on the website, like visualizing products and ordering online, should work even better and last around 7,500 hours.

**- Mean Time To Repair (MTTR):**

If something goes wrong, we promise to fix it within 2 hours. During that time, some features might work a bit slower but won't stop completely.

**- Accuracy:**

Pictures and product details on the website will look good, with clear images (1080p) and accurate product information (within 2% of the real thing).

**- Maximum Bugs or Defect Rate:**

* The website's code should have very few mistakes, no more than 0.5 bugs for every thousand lines of code.
* Each function on the website should have even fewer mistakes, not more than 0.1 bugs.

**- Bugs or Defect Rate:**

* If there are any mistakes, we'll sort them by how bad they are: small, medium, or big.
* Big mistakes (critical bugs) that could lose data or make parts of the website unusable should be fixed super quickly, within 2 days. Smaller mistakes can take up to a week to fix. We're aiming for zero big mistakes.

### 3.3 Performance

**- Response time:**

* Search and booking transactions: Maximum response time is 400 milliseconds.
* Login transaction: Minimum response time is 200 milliseconds

**- Throughput:** Transactions per second (TPS): 100

**- Capacity**: Number of customers: 10,000

**- Degradation mode:** In the event of loss of part or all of the infrastructure, the system must be able to maintain at least 50% capacity.

**- Resource utilization:**

* Memory: Maximum is 5 GB
* Disk: Minimum is 100 GB
* Networking: Minimum is 50 Mbps

# III. Design Specifications

## 1. <<Feature Name>>

### 1.1 <<SubFeature Name>>

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

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

##### UI Design

*[This is to describe the UI layout (Mockup prototype) & descriptions for screen fields/components]*

<<Mockup prototype>>

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| ***Field Group Name*** | | |
| <<Field-Name>> | <<Field type>> | <<Field description & data initializing design>> |

##### Database Access

*[Provide the design description for the screen/function to access the database here: what table the screen/function would access, which transactions does it make (C-Create, R-Read, U-Update, or D-Delete), and how/purpose of the access (by providing Description and SQL commands)]*

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| <<Table Name>> | <<transaction(s)>> | <<Table access description: purpose, how,…>> |
| .. |  |  |

***SQL Commands***

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

### 1.2 System Access

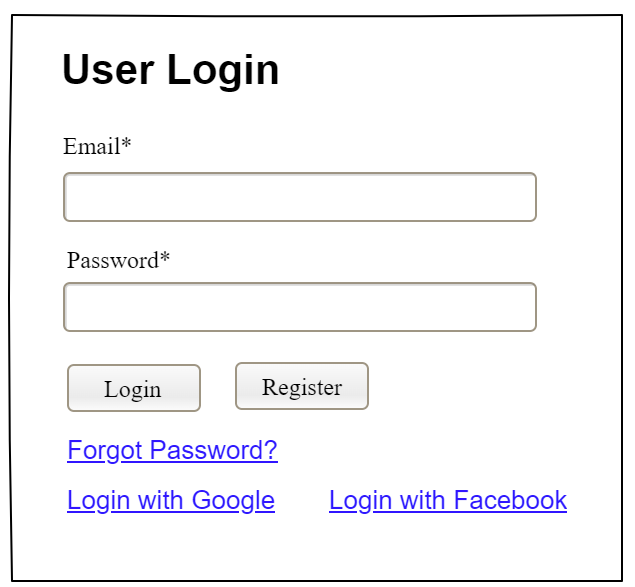
#### a. User Login

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

Related use cases:

* [UC02\_Login System](#_heading=h.35nkun2)

##### UI Design



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

##### Database Access

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

***SQL Commands:***

1/ Verify UserName & Password information

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

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

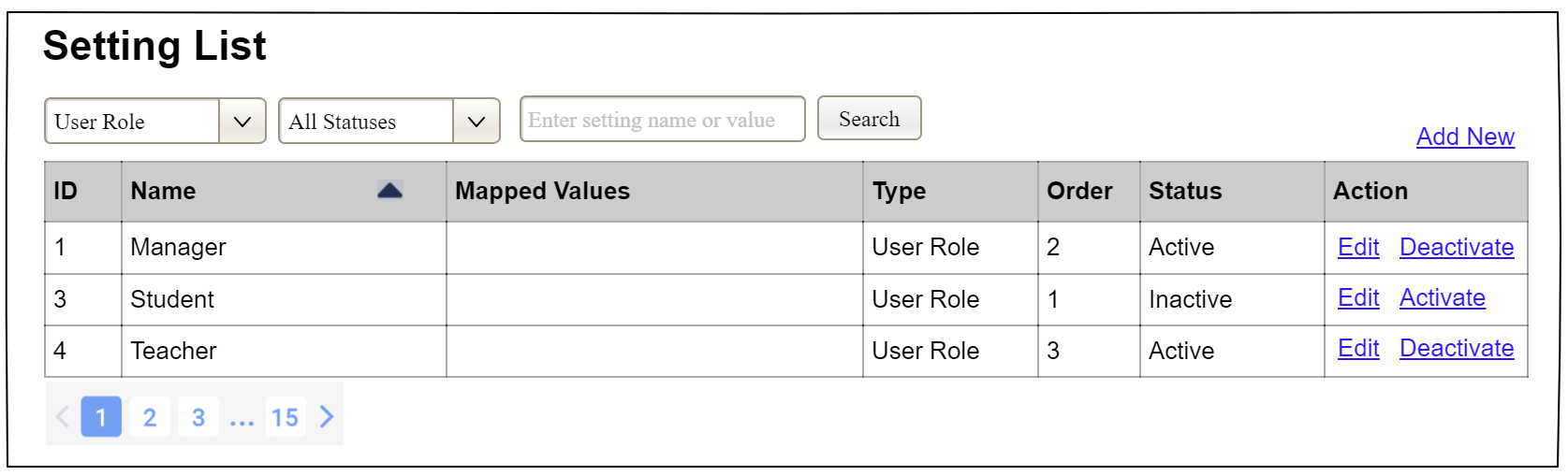
2/ Specify the authorizations of the logged-in user

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

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

#### b. Setting List

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| ***Search Fields*** | | |
| Setting Type | Combo Box  Single-Choice | Filled with the list of current active setting types  Allow to filter the list by setting type;  Default value is “All Types” |
| Setting Status | Combo Box  Single-Choice | Values: All Statuses (default), Active, and Inactive  Allow to filter the list by status  Default value: “All Statuses” |
| Search Phase | Text Box  String (30) | Allow to search using the name or map values  Default value: blank |
| Search | Button | Click to refresh the list with the defined filter(s) and search phrase. |
| Add New | Hyperlink | Click to open the Setting Details page for adding new setting (master data) |
| ***Data Table*** | | |
| ID | Integer | Auto-increased identifier of the setting |
| Name | Text | Name of the setting |
| Mapped Values | Text | Supplementary information for the setting |
| Type | Text | Type of the setting |
| Order | Integer | Display order of the setting: the order of the setting type, displayed among the list of settings with the same type |
| ***Data Actions*** | | |
| Edit | icon | Click to open the Setting Details page for updating the relevant setting (master data) |
| Activate | icon | Shown when the data status is inactive. This is to activate the relevant setting (master data) |
| Deactivate | Ion | Shown when the data status is active. This is to deactivate the relevant setting (master data) |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Setting | RU | Query the list of current settings from the database  Update status of a specific setting |

***SQL Commands:***

1/ Query the list of current settings from the database

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

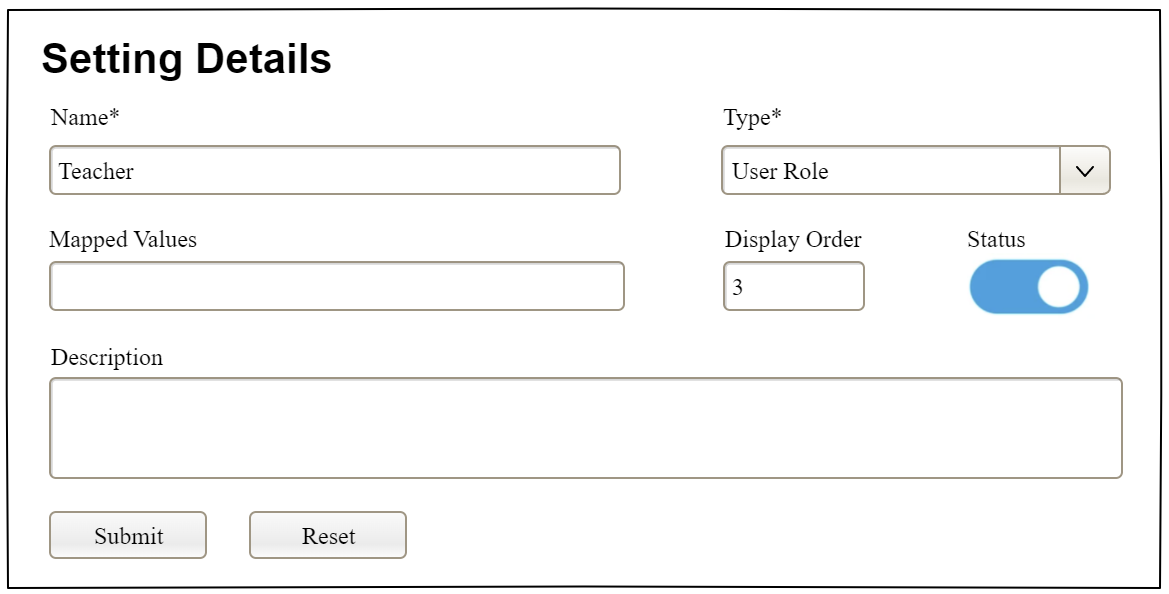
FROM setting WHERE (setting\_type = ?) AND (status = ?) AND (setting\_name LIKE ?)

2/ Update status of a specific setting

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

#### c. Setting Details

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Name\* | Text Box  String (20) | Name of the setting |
| Type\* | Combo Box  (Single Choice) | Type of the setting, filled with the list of setting types  Default value: the first type in the list |
| Mapped Values | Text Box  String (50) | Supplementary information for the setting (if any) |
| Order | Text Box  Integer (>=0) | Display order of the setting: the order of the setting type, displayed among the list of settings with the same type |
| Status | On/Off button | Status of the setting: Active or Inactive  Default value: Active |
| Description | Text Area  String (200) | Description of the setting |
| Submit | Button | Click to store new or updated setting details |
| Reset | Button | Click to reset the changes use has made on the screen fields back to the initial values when the screen is loaded |

##### Database Access

…

## 