Requirement & Design Specification

**House Rental Management System (HRMS)**

**Version: 1.0**

– Hanoi, September 2023 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| V1.0 |  | A | MinhNTHE176001  HieuNDHE170151  LongNPHE170854  ThinhLHHE170034  AnhTQTHE172642 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[**Record of Changes 2**](#_heading=h.gjdgxs)

[**I. Overview 6**](#_heading=h.30j0zll)

[1. User Requirements 6](#_heading=h.1fob9te)

[1.1 Actors 6](#_heading=h.3znysh7)

[1.2 Use Cases 7](#_heading=h.2et92p0)

[2. Overall Functionalities 10](#_heading=h.tyjcwt)

[2.1 Screens Flow 10](#_heading=h.3dy6vkm)

[2.2 Screen Descriptions 10](#_heading=h.1t3h5sf)

[2.3 Screen Authorization 12](#_heading=h.4d34og8)

[2.4 Non-UI Functions 13](#_heading=h.2s8eyo1)

[3. System High Level Design 14](#_heading=h.17dp8vu)

[3.1 Database Design 14](#_heading=h.3rdcrjn)

[3.2 Code Packages 18](#_heading=h.35nkun2)

[**II. Requirement Specifications 21**](#_heading=h.1ksv4uv)

[1. User Registration 21](#_heading=h.z337ya)

[1.1 UC-1\_Register 21](#_heading=h.3j2qqm3)

[2. User Authentication 23](#_heading=h.nnniphymnxo6)

[2.1 UC-2\_Log In 23](#_heading=h.6wng838zfo9m)

[2.2 UC-3\_Log Out 25](#_heading=h.8s73hukigpm4)

[2.3 UC-4\_Forgot Password 26](#_heading=h.7futof6zblpj)

[3. User Profile Management 27](#_heading=h.nd8drtwb04di)

[3.1 UC-5\_Change Password 27](#_heading=h.5b9ax9t3lly)

[3.2 UC-6\_Edit Profile 29](#_heading=h.30ph5agat9x2)

[4. Property Listing Feature 31](#_heading=h.c75lddm9b62y)

[4.1 UC07\_View property list 31](#_heading=h.w7f60b1q31te)

[4.2 UC08\_View property detail 32](#_heading=h.ukk33p8cm9vk)

[4.3 UC09\_Search property 33](#_heading=h.1hrwteycndir)

[4.4 UC10\_Filter property 34](#_heading=h.rlkrzpwskhyo)

[5. Rental Application Feature 35](#_heading=h.dm2v34pjpdo3)

[5.1. UC11\_Create rental application 35](#_heading=h.2h2thh9al2qk)

[5.2. UC12\_Cancel rental application 36](#_heading=h.f9ucg1rmzrst)

[5.2. UC13\_View application history 37](#_heading=h.k2hbi7ljvc26)

[5.2. UC14\_View rental application 38](#_heading=h.eh0frjtc3l4z)

[5.3. UC15\_Accept/Reject tenant application 39](#_heading=h.jzgqora2rx9g)

[6. Property Management Feature 41](#_heading=h.vanihwucxg7j)

[6.1 UC16\_View posted properties 41](#_heading=h.et6zl1onb34a)

[6.2 UC17\_Post new rental property 42](#_heading=h.q5sk8tuwulg4)

[6.3 UC18\_Edit existing property 43](#_heading=h.v0v96m81qg5u)

[6.4 UC19\_Remove from property listing 45](#_heading=h.ndbe1xltrwix)

[7. Payment feature 47](#_heading=h.1r2ptrfd3frs)

[7.1. UC20\_Pay for post 47](#_heading=h.27it09tm6rlr)

[7.2. UC21\_View transaction history 48](#_heading=h.i886slffhljj)

[7.3. UC22\_Add point to Landlord 49](#_heading=h.jcfqwkfbpxsn)

[7.4. UC23\_View system transaction 50](#_heading=h.e151u6zdlqy3)

[8. Wishlist feature 51](#_heading=h.ol5ogc3epvo3)

[8.1. UC24\_Add to wishlist 51](#_heading=h.1jvb1ll6504d)

[8.2. UC25\_Remove from wishlist 52](#_heading=h.24xs972kob4u)

[8.3. UC26\_View wishlist 53](#_heading=h.82pacmwroyc2)

[9.Manage User Account 55](#_heading=h.kd3frwt9heig)

[9.1. UC40\_View account 55](#_heading=h.9r6kg2zeg9q7)

[9.2. UC33\_Add account 56](#_heading=h.d1d7zutrgqwf)

[9.3. UC34\_Disable account 57](#_heading=h.gly12glcl0jw)

[9.4. UC35\_Ban account 58](#_heading=h.6vkyuogt09ht)

[9.5. UC36\_Verify landlord account 59](#_heading=h.op6pe45bm4wz)

[10. Review feature 60](#_heading=h.c2vtfxbfc22i)

[10.1. UC27\_Send feedback 60](#_heading=h.f4hfsf2hc8f)

[10.2. UC28\_Rate property 61](#_heading=h.8v980xlul1cj)

[10.3. UC38\_View feedback 62](#_heading=h.lnw212848zig)

[11. Report feature 63](#_heading=h.jvrhyd16l0vi)

[11.1 . UC29\_Report 63](#_heading=h.cotpaok2818x)

[12. Promotion feature 64](#_heading=h.udro8uv89cj2)

[12.1. UC30\_Create promotion 64](#_heading=h.xspzoj9xy9vm)

[12.2. UC31\_Edit promotion 65](#_heading=h.1edtuo5t19pr)

[12.3. UC32\_Remove promotion 66](#_heading=h.ou40inrfuck7)

[13. UC39\_View homepage 67](#_heading=h.2046g34qer4r)

[10. Business template 68](#_heading=h.wsvfgjgbyg4d)

[**III. Design Specifications 71**](#_heading=h.1ci93xb)

[1. User Registration 71](#_heading=h.wh5v3gj7ii88)

[1.1 Register 71](#_heading=h.w1hnsct27w24)

[2. User Authentication 74](#_heading=h.xgq7b0m4m1wb)

[2.1 Log In 74](#_heading=h.iopy7ruajxq)

[2.2 Log Out 76](#_heading=h.nxsxmx3egakz)

[2.3 Forgot Password 76](#_heading=h.5p9p49au1kv7)

[3. User Authentication 78](#_heading=h.2wi8q13tkrnd)

[3.1 Reset Password 78](#_heading=h.l9zwkkwpwft9)

[3.2 Edit profile 79](#_heading=h.mh6ezijtq6ko)

[4. Property Listing Feature 80](#_heading=h.x1rfj0a65ens)

[4.1. View property list 80](#_heading=h.weumwqoyww4x)

[4.2. Search property 81](#_heading=h.ddkk38c6m0kn)

[4.3. Filter property 82](#_heading=h.dhotcyf3umaj)

[4.4. View property detail 84](#_heading=h.rqxkbeu6m48t)

[5. Rental Application Feature 86](#_heading=h.410fznu7zl2a)

[5.1. Application Processing 86](#_heading=h.49ywzslb9ttl)

[5.2. Application Management 93](#_heading=h.32c8t6un3tza)

[6. Property Management Feature 97](#_heading=h.je1ef6um7jtk)

[6.1. View posted properties 97](#_heading=h.xe8ov6v4tdvx)

[6.2. Post new rental property 100](#_heading=h.r2ob4naaqh6)

[6.3. Edit existing property 101](#_heading=h.jod57fvh2ikn)

[6.4. Remove from property listing 104](#_heading=h.62wef51r23l0)

[7. Manage Wishlist 105](#_heading=h.gouk9dxe2qel)

[8. Admin management 109](#_heading=h.5rsyi6ilhqyg)

[8.1. Manage account 109](#_heading=h.g4czbak1k3sb)

[8.2. Report management 110](#_heading=h.8wirez8y7a5o)

[9. Review 111](#_heading=h.qforb270wb2q)

[10. Payment feature 112](#_heading=h.npa36wwtf810)

[10.1. Pay for post 112](#_heading=h.pb4ixktt54q6)

[10.2. View transaction history 112](#_heading=h.lw1v51682a9h)

[10.3. View system transaction 113](#_heading=h.y71zcuya6wyz)

[10.4. Add point to Landlord 114](#_heading=h.wwh61floutcx)

[11. Promotion feature 115](#_heading=h.e2opv74ltpvl)

[**IV. Appendix 118**](#_heading=h.3as4poj)

[1. Assumptions & Dependencies 118](#_heading=h.1pxezwc)

[2. Limitations & Exclusions 118](#_heading=h.49x2ik5)

[3. Business Rules 119](#_heading=h.2p2csry)

# 

# I. Overview

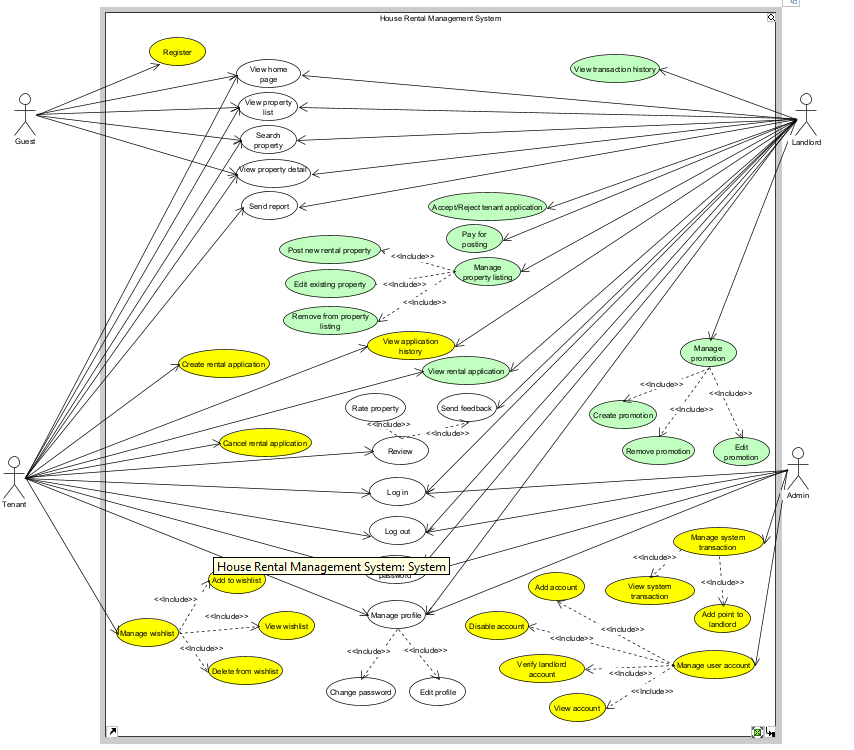
## 1. User Requirements

### 1.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator | An Administrator is responsible for overseeing and managing the entire House Rental Management System. |
| 2 | Landlord | An individual or entity who owns and manages rental properties listed on the platform |
| 3 | Tenant | An individual or entity looking for a rental property |
| 4 | Guest | A potential user who visits the platform without registering or logging in |

### 1.2 Use Cases

#### a. Diagram(s)



#### b. Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Feature** | **Use Case** | **Use Case Description** |
| 01 | User registration | Register | Creates a new account on the system |
| 02 | User authentication | Log in | Log in to the system |
| 03 | User authentication | Log out | Log out of the system |
| 04 | User authentication | Forget password | Set new password if user forgot old password |
| 05 | User profile management | Change password | Set new password |
| 06 | User profile management | Edit profile | Modify user personal information |
| 07 | Property listing | View property list | Display list of available rental properties |
| 08 | Property listing | View property detail | Display property detail |
| 09 | Property listing | Search property | Search property by criteria |
| 10 | Property listing | Filter property | Filter property list by criteria |
| 11 | Application processing | Create rental application | Tenant create rental application |
| 12 | Application processing | Cancel rental application | Tenant can cancel rental application if landlord does not accept or reject the application |
| 13 | Application management | View application history | Display rental application history |
| 14 | Application management | View rental application | Display rental application |
| 15 | Application processing | Accept/Reject rental application | Landlord accept or reject the rental application |
| 16 | Property management | View posted properties | Landlord view posted properties |
| 17 | Property management | Post new rental property | Post new rental property |
| 18 | Property management | Edit existing property | Edit existing property |
| 19 | Property management | Remove from property listing | Remove a property from listing  pay for post |
| 20 | Payment | Pay for post | Pay for each post over a period of time |
| 21 | Payment | View transaction history | Display all payment transaction of a landlord |
| 22 | Payment | Add point to landlord | Add points (system’s currency) to landlord account |
| 23 | Payment | View system transaction | Display all transaction in the system |
| 24 | Wishlist management | Add to wishlist | Add property to wishlist |
| 25 | Wishlist management | Remove from wishlist | Remove property from wishlist |
| 26 | Wishlist management | View wishlist | Display all properties in wishlist |
| 27 | Review feature | Send feedback | Send feedback to user |
| 28 | Review feature | Rate property | Rate a property |
| 29 | Report feature | Send report | Send a report to administrator |
| 30 | Promotion management | Create promotion | Create a promotion |
| 31 | Promotion management | Edit promotion | Edit a promotion |
| 32 | Promotion management | Remove promotion | Remove a promotion |
| 33 | User account management | Add account | Add an account |
| 34 | User account management | Disable account | Disable an account |
| 35 | User account management | Ban account | Ban an account |
| 36 | User account management | Verify landlord account | Approve or deny user’s request to become a landlord |
| 37 | Report feature | View and process report | Receive report from tenant/landlord and solve problem |
| 38 | Review feature | View feedback | View feedback of properties and tenant |
| 39 | Homepage | View homepage | View homepage of system. |
| 40 | Post management | Disable post | Disable post that violate the regulation |

## 2. Overall Functionalities

### 2.1 Screens Flow



### 2.2 Screen Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Login | Login page | For login action |
| 2 | Register | Register | Sign in an account |
| 3 | Forgot the password | Forgot password | Set new password when forgot the old password |
| 4 | View property list | View property list | All user can see the property list |
| 5 | View property list | View property detail | All user can see details about any specific property (price, rate, comment,...) |
| 6 | View property detail | Review | Vote and comment about the property, can only do when the tenant is already signed the contract with landlord |
| 7 | Manage application | Manage application | Manager application for 2 roles: tenant and landlord |
| 8 | Manage application | Approve/reject order | Function for landlord only, they can accept or reject tenant application |
| 9 | Manage application | View application detail | Function for landlord and tenant, they can view all the applications |
| 10 | Manage application | Cancel order | Function for tenant only, they can cancel their applications |
| 11 | Manage profile | Manage profile | Manage account for all roles |
| 12 | Manage account | Change password | Change account password |
| 13 | Send report | Send report | Report to the admin about bugs of web, or any trouble with the tenant/landlord |
| 14 | Payment | View payment history | Landlord can see what they have paid |
| 15 | Manage property listing | Manage property | Landlord manage all their posts |
| 16 | Manage property | Post new rental property | Landlord can use point to post |
| 17 | Manage property | Edit existing post | Landlord can edit information in their posts |
| 18 | Manage property | Manage promotion | Manage the promotion of each property |
| 19 | Manage promotion | Create promotion | Create promotion on specific property |
| 20 | Manage promotion | Edit promotion | Landlord can edit the promotions that have been released |
| 21 | View wishlist | View wishlist | Tenant can see all the list that they have added before |
| 22 | View wishlist | Create application | Create application to rental the property |
| 23 | Dashboard | Dashboard | Admin management system |
| 24 | Dashboard | Manage user account | Manage all the existed account |
| 25 | Dashboard | Manage system transaction | Manage all the transaction on platform |
| 26 | Manage user account | Ban account | Ban an user if they violate the system’s regulations |
| 27 | Add admin account | Add admin account | Promote an normal account to be an admin account |
| 28 | Manage system transaction | Add point to landlord | Admin can add points to landlord’s wallet for later usage |

### 2.3 Screen Authorization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Guest** | **Landlord** | **Tenant** | **Admin** |
| Login page |  | X | X | X |
| Register | X |  |  |  |
| Forgot password |  | X | X | X |
| View property list | X | X | X | X |
| View property detail | X | X | X | X |
| Review |  |  | X |  |
| Manage application |  | X | X |  |
| Approve/reject order |  | X |  |  |
| View application detail |  | X | X |  |
| Cancel order |  |  | X |  |
| Manage profile |  | X | X | X |
| Change password |  | X | X | X |
| Send report |  | X | X |  |
| View payment history |  | X |  |  |
| Manage property |  | X |  |  |
| Post new rental property |  | X |  |  |
| Edit existing post |  | X |  |  |
| Manage promotion |  | X |  |  |
| Create promotion |  | X |  |  |
| Edit promotion |  | X |  |  |
| View wishlist |  |  | X |  |
| Create application |  |  | X |  |
| Dashboard |  |  |  | X |
| Manage user account |  |  |  | X |
| Manage system transaction |  |  |  | X |
| Ban account |  |  |  | X |
| Add admin account |  |  |  | X |
| Add point to landlord |  |  |  | X |
| View account |  |  |  | X |

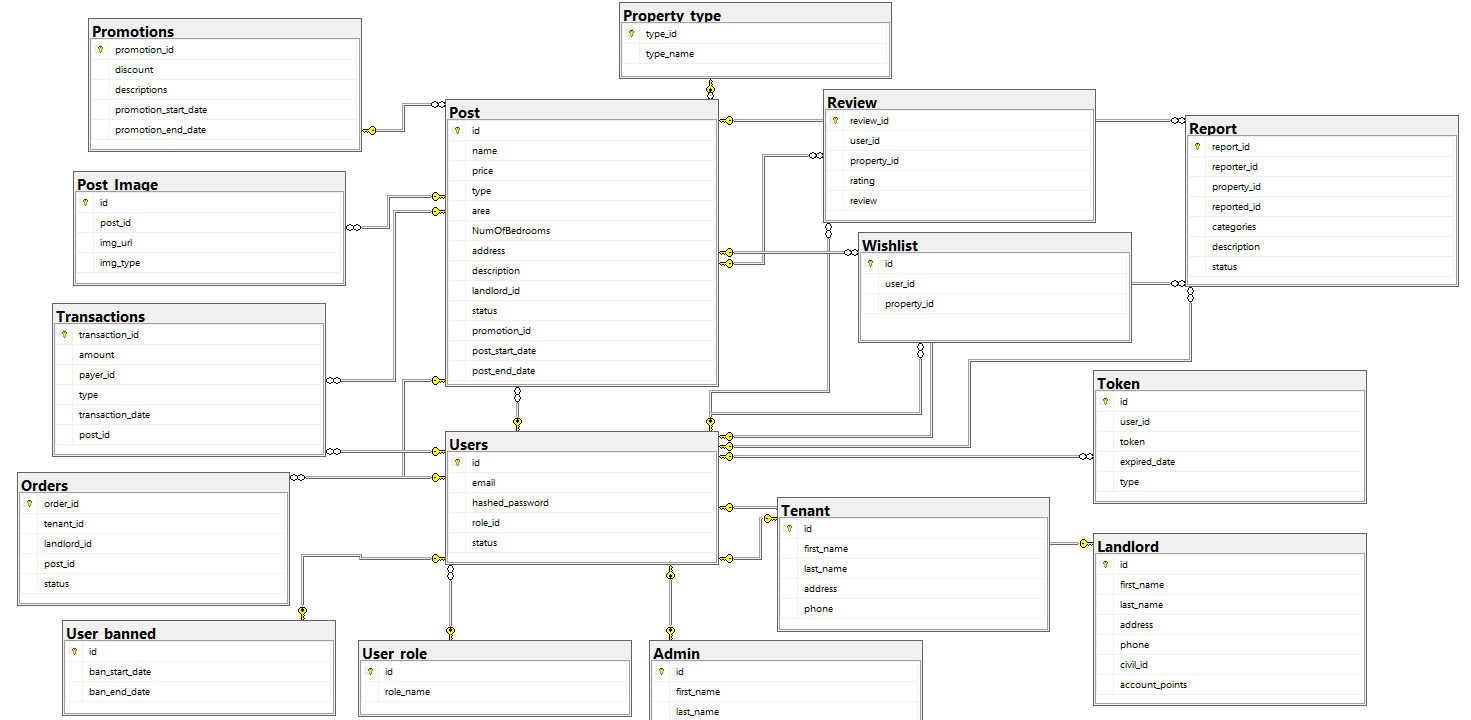
### 2.4 Non-UI Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | Friendly UI/UX |  | Friendly UI/UX lets users easily use this web. |
| 2 | Security | Reset password | Automatically send an email containing the verify code to the registered email for the account when resetting the password for the account. |
| 3 | Security | Register | Automatically send an email containing the verify code to the registered email for the account when signing in a new account. |
| 4 | Security | Register | The password and verify code of user must be encrypted by MD5 algorithm |

## 3. System High Level Design

### 3.1 Database Design

#### a. Database Schema

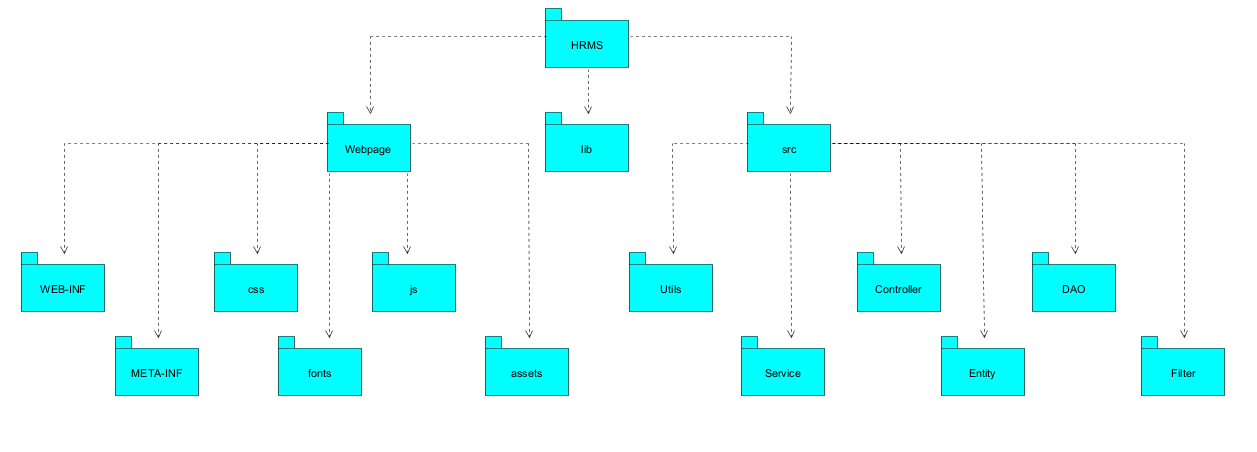


#### 

#### b. Table Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | Users | Store data about account information  - Primary keys:  + id: int  - Foreign keys:  + role\_id: int  - Columns:  + email: varchar(100)  + hashed\_password: varchar(80)  + status: int |
| 02 | User\_role | Store data about user’s role  - Primary keys:  + id: int  - Columns:  + role\_name: varchar(20) |
| 02 | Landlord | Store data about landlord’s information  - Primary keys:  + id: int  - Columns:  + first\_name: nvarchar(20)  + last\_name: nvarchar(20)  + address: nvarchar(255)  + phone: varchar(15)  + civil\_id: nvarchar(200)  + account\_points: int |
| 03 | Tenant | Store data about tenant’s information  - Primary keys:  + id: int  - Columns:  + first\_name: nvarchar(20)  + last\_name: nvarchar(20)  + address: varchar(255)  + phone: varchar(15) |
| 04 | Admin | Store data about admin’s information  - Primary keys:  + id: int  - Columns:  + first\_name: nvarchar(20)  + last\_name: nvarchar(20) |
| 05 | Orders | Store data about rental application  - Primary keys:  + order\_id: int  - Foreign keys:  + tenant\_id: int  + landlord\_id: int  + post\_id: int  - Columns:  + status: nvarchar(50) |
| 06 | Post\_Image | Store image of property  - Primary keys:  + id: int  - Foreign keys:  + post\_id: int  - Columns:  + img\_url: nvarchar(300)  + img\_type: nvarchar(20) |
| 07 | Post | Store data about property  - Primary keys:  + id: int  - Foreign keys:  + type: int  + landlord\_id: int  + promotion\_id: int  - Columns:  + name: nvarchar(255)  + price: int  + area: int  + NumOfBedrooms: int  + address: varchar(255)  + description: nvarchar(2000)  + status: int  + post\_start\_date: date  + post\_end\_date: date |
| 08 | Property\_type | Store data about property type  - Primary keys:  + type\_id: int  - Columns:  + type\_name: nvarchar(50) |
| 09 | Promotions | Store data about promotion  - Primary keys:  + promotion\_id: int  - Columns:  + discount: int  + descriptions: nvarchar(255)  + promotion\_start\_date: date  + promotion\_end\_date: date |
| 10 | Report | Store data about report  - Primary keys:  + report\_id: int  - Foreign keys:  + reporter\_id: int  + property\_id: int  + reported\_id: int  - Columns:  + categories: nvarchar(100)  + description: text  + status: nvarchar(20) |
| 11 | Review | Store data about review  - Primary keys:  + review\_id: int  - Foreign keys:  + user\_id: int  + property\_id: int  - Columns:  + rating: smallint  + review: nvarchar(300) |
| 12 | Transactions | Store data about transaction  - Primary keys:  + transaction\_id: int  - Foreign keys:  + payer\_id: int  + post\_id: int  - Columns:  + type: nvarchar(20)  + transaction\_date: date  + amount: decimal(19, 0) |
| 13 | Whislist | Store data about wishlist  - Primary keys:  + id: int  - Foreign keys:  + user\_id: int  + property\_id: int |
| 14 | User\_banned | Store data about user information has been banned  - Primary keys:  + id: int  - Columns:  + ban\_start\_date: datetime  + ban\_end\_date: datetime |
| 15 | Token | Store data about token  - Primary keys:  + id: int  - Columns:  + user\_id: int  + token: nvarchar(255)  + expired\_date: datetime  + type: int |

### 3.2 Code Packages



***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *Webpage* | *The Web Pages contains Web Pages files and .jsp file* |
| *02* | *src* | *The src contains all source code with .class, .java file* |
| *03* | *WEB-INF* | *The WEB-INF contains web.xml , directory is private on web, can access by clients* |
| *04* | *META-INF* | *The META-INF stores package and extension configuration data, including security, versioning, extension and services* |
| *05* | *css* | *The css contains css files* |
| *06* | *fonts* | *The fonts contains font files* |
| *07* | *js* | *The js contains javascript files* |
| *08* | *assets* | *The assets contains the resources file such as image, icon* |
| *09* | *lib* | *The lib contains libraries* |
| *10* | *utils* | *The utils consists of some repeatedly used functions that are commonly used in the project* |
| *11* | *controller* | *The user directory contains all controllers* |
| *12* | *service* | *The service contains source code to manage all api requests and navigation* |
| *13* | *entity* | *The entity contains all entities in project* |
| *14* | *dao* | *The dao contains all Data Access Field and manage all entities* |
| *15* | *filter* | *The filter contains all code to manage streams’ items* |

# II. Requirement Specifications

## 1. User Registration

### 1.1 UC-1\_Register

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC01-Register | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Guest | Secondary Actors: |  |
| Trigger: | The user visits the platform’s website and clicks on the “Sign Up” button. | | |
| Description: | Register is the process where a user creates a new account on the system. | | |
| Preconditions: | PRE-1. The user has access to the internet and the platform’s website.  PRE-2. The account with that email has not been activated. | | |
| Postconditions: | POST-1. A new user account is created in the platform’s database.  POST-2. A verification email is sent to the user’s provided email address.  POST-3. The user is redirected to their newly created account page. | | |
| Normal Flow: | **1.0 Register**  1. The user navigates to the platform’s website.  2. The platform’s homepage is displayed  3. The user clicks on the “Sign Up” button.  4. The registration page loads, displaying input fields for email, password,first name ,last name.  5. The user enters their valid email address.  6. The system check email format, which is valid  7. The user creates a strong password  8. The system validates the password as strong  9. The user selects their role as either "Tenant" or "Landlord."  10. The system records the user's selected role.  11. The user submits the registration form.  12. The system processes the registration information, creates a new user account, and sends a verification email to the provided email address.  13. The user checks their email and clicks the verification link.  14. The system verifies the email address, activates the user’s account, and redirects the user to their newly created account page.  17. The user explores their newly registered account.  18. The system displays the user’s account dashboard, allowing the user to access platform features. | | |
| Alternative Flows: | None | | |
| Exceptions: | 1.11.E1. If the email address is already associated with an activated account, the system displays an error message and prompts the user to log in instead of registering.  1.11.E2. If the email address is already associated with an unverified account with the same role, the system will notify that email is already used and go straight “forgot password” to take the new password. | | |
| Priority: | High | | |
| Frequency of Use: | Estimated to be many times per day. | | |
| Business Rules: | BR-01, BR-02,BR-03,BR-04, BR-05, BR-06, BR-07, BR-08 | | |
| Other Information: |  | | |
| Assumptions: | - The user has a valid and accessible email address  - The platform’s email system is functioning correctly for sending verification emails. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| BR-01 |  | Account’s email must be valid. |
| BR-02 |  | Users must enter password twice to confirm password when changing password or sign up. |
| BR-03 |  | Account’s password must be at least 8 characters in length, 1 letter and 1 digit. |
| BR-04 |  | Account’s password must not be stored as a string sequence. Instead it must be hashed using a secure hash algorithm. |
| BR-05 |  | Cannot register with an email that has already been activated. |
| BR-06 |  | When registering or changing the password, the user must enter the new password twice. |
| BR-07 |  | After registering or forgetting password, guests must have an active account with a valid link sent to email. |
| BR-08 |  | If the email is stored in a database but not activated, the system sends a link to change the password of the existing account to that email when there is another registration request. |

## 

## 2. User Authentication

### 2.1 UC-2\_Log In

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC02 - Log In | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Landlord, Tenant, Admin | Secondary Actors: |  |
| Trigger: | The user visits the platform’s website and clicks on the “LogIn” button. | | |
| Description: | Log in to the system. | | |
| Preconditions: | PRE-1. The user has access to the internet and the platform’s website.  PRE-2. The user account has been created and authorized. | | |
| Postconditions: | POST-1. The user is successfully logged into their account and can access their profile and platform features.  POST-2. If the user chooses to log out, they are logged out of their account. | | |
| Normal Flow: | **2.0 Log in**  1. The user navigates to the platform’s website.  2. The platform’s homepage is displayed.  3. The user clicks on the “Log In” button.  4. The login page loads, displaying input fields for email and password.  5. The user enters their registered email address.  6. The system checks the email format. which is valid.  7. The user enters their password.  8. The system validates the user’s email and password combination.  9. The user clicks the “LogIn” button.  10. The system verifies the user’s credentials. and if they are correct, the user is logged into their account.  11. The system redirects to the home page. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2.9.E1: The email and password combination is incorrect, the system displays an error message and prompts the user to try again or reset their password. | | |
| Priority: | High | | |
| Frequency of Use: | Estimated to be thousands of times per day. | | |
| Business Rules: | BR28 | | |
| Other Information: |  | | |
| Assumptions: | - The user has a registered account with a valid email address and password.  - The platform’s login system is functioning correctly. | | |

#### 

#### b. Business Rules

None

### 2.2 UC-3\_Log Out

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC03 - Log Out | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Landlord, Tenant, Admin | Secondary Actors: |  |
| Trigger: | The logged-in user clicks on the icon to log out. | | |
| Description: | Log out of the system. | | |
| Preconditions: | PRE-1. The user is currently logged into their account. | | |
| Postconditions: | POST-1. The user is successfully logged out of their account and can no longer access their profile and platform features until they log in again. | | |
| Normal Flow: | **3.0 Log out**  1. THe logged-in user clicks on the “Log Out” button or link within the platform.  2. The system logs the user out and redirects them to the platform’s homepage or a log-in page. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Estimated to be hundreds of times per day. | | |
| Business Rules: | BR-10 | | |
| Other Information: | After logging out, the system should clear any session data and cookies associated with the user’s session to ensure they are fully logged out. | | |
| Assumptions: | - The user is currently logged into their account.  - The platform’s logout functionality is functioning correctly. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| BR-9 |  | Users must be logged in to log out or edit information or change password. |

### 2.3 UC-4\_Forgot Password

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC04 - Forgot Password | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Landlord, Tenant, Admin | Secondary Actors: |  |
| Trigger: | The user clicks on the “Forgot Password” link on the login page after forgetting their password. | | |
| Description: | Set new password if user forgot old password | | |
| Preconditions: | PRE-1. The user has an existing account on the platform. | | |
| Postconditions: | POST-1. The user receives an email with a link to reset their password.  POST-2. The user receives a new password.  POST-3. The user is redirected to the login page. | | |
| Normal Flow: | **4.0 Forgot Password**  1. The user navigates to the platform’s login page.  2. The login page is displayed, including a “Forgot Password” link.  3. The user clicks on the “Forgot Password” link.  4. The system presents a form where the user can enter their registered email address.  5. The user enters their registered email address.  6. The system verifies that the provided email address is associated with a valid account.  7. The user submits the form to request a password reset.  8. The system generates a unique token for the user and sends an email to the provided email address with a link to reset the password.  9. The user checks their email and clicks the password reset link.  10. The system verifies the token and sends a new password.  11. The system updates the user’s password in the database.  12. The user is redirected to the login page and can log in with their new password.  16. The system presents the user with the login page. | | |
| Alternative Flows: | None | | |
| Exceptions: | 4.7.E1: If the provided email address is not associated with a valid account, the system displays an error message and prompts the user to try again.  4.9.E2: If the token is invalid or has expired, the system displays an error message and prompts the user to request another password reset email. | | |
| Priority: | Medium | | |
| Frequency of Use: | Estimated to be several times per week. | | |
| Business Rules: | BR-1,BR-3,BR4,BR-7, BR-8 | | |
| Other Information: | In case of an error during the email delivery of the password reset link, the system should provide an error message and offer the option to resend the email. | | |
| Assumptions: | - The user has a registered account with a valid email address.  - The platform’s email system is functioning correctly for sending password reset emails. | | |

#### b. Business Rules

None

## 3. User Profile Management

### 3.1 UC-5\_Change Password

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC05 - Change Password | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Landlord, Tenant, Admin | Secondary Actors: |  |
| Trigger: | The user clicks on the “Change password” link on the profile page. | | |
| Description: | Set new password if user decide to change password | | |
| Preconditions: | PRE-1. The user is currently logged into their account.  PRE-2. The user knows their current password. | | |
| Postconditions: | POST-1. The user's password is updated to the new one.  POST-2. The user is logged out and prompted to log in again with the new password. | | |
| Normal Flow: | **5.0 Change Password**  1.The logged-in user navigates to their account settings or profile page.  2. The user's account settings page is displayed.  3. The user clicks on the "Change Password" option.  4. The system presents a form for changing the password, including fields for the current password, the new password and the re-password.  5.The user enters their current password ,the new desired password and re-password.  6.The system validates the current password and ensures that the new password conforms to the platform's password requirements and the new password is the same as the re-password.  7. The user submits the form to change the password.  8. The system updates the user's password in the database with the new one.  9. The user is logged out automatically by the system.  10. The system logs the user out and displays a message instructing the user to log in again with the new password. | | |
| Alternative Flows: | None | | |
| Exceptions: | 5.5.E1: If the entered current password is incorrect, the system displays an error message and prompts the user to try again.  5.5.E3.If the new password and the re-password are not the same,the system displays an error message and prompts the user to try again.  5.7.E4: If there are validation errors in the new password (e.g., too short, not meeting complexity requirements), the system displays an error message and advises the user to choose a different password.  5.7.E5: If there is a system error during the password update process, the system displays an error message and advises the user to try again later. | | |
| Priority: | Medium | | |
| Frequency of Use: | Estimated to be several times per year. | | |
| Business Rules: | BR-8,BR-10 | | |
| Other Information: | In case of a password change error or system error, the system should provide an error message and possibly a recovery mechanism (e.g., reset password). Assumptions: | | |
| Assumptions: | - The user is currently logged into their account.  - The platform's password change functionality is functioning correctly. | | |

#### b. Business Rules

None

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| BR-10 |  | When changing password, the new password must not be the same as the current password. |

### 3.2 UC-6\_Edit Profile

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC06 - Edit profile | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Landlord, Tenant, Admin | Secondary Actors: | None |
| Trigger: | The logged-in user decides to edit their profile information. | | |
| Description: | Modify user personal information | | |
| Preconditions: | PRE-1. The user is currently logged into their account.  PRE-2. The user has an existing profile on the platform. | | |
| Postconditions: | POST-1. The user’s profile information is updated with the new data.  POST-2. The user is redirected to their updated profile page. | | |
| Normal Flow: | 6.0 Edit profile  1. The logged-in user navigates to their profile page.  2. The user’s profile page is displayed, showing their current profile information.  3. The user clicks on the “Edit profile” button.  4. The system loads an edit profile form pre-populated with the user's existing information.  5. The user makes changes to their profile information, such as updating their password, phone number.  6. The system validates the changes and ensures that the updated information conforms to the platform's requirements.  7. The user submits the edited profile information.  8. The system updates the user's profile information in the database with the new data.  9. The user is redirected to their updated profile page.  10. The system displays the user's profile with the updated information. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Estimate to be several times per week | | |
| Business Rules: | BR-9 | | |
| Other Information: |  | | |
| Assumptions: | - The user is currently logged into their account.  - The platform’s edit profile functionality is functioning correctly. | | |

#### 

#### b. Business Rules

None

## 4. Property Listing Feature

### 4.1 UC07\_View property list

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC07 - View property list | | |
| Created By: | LongNP | Date Created: | 23/9/2023 |
| Primary Actor: | Landlord, Tenant, Guest | Secondary Actors: | None |
| Trigger: | Landlord, Tenant and Guest click on “Property Listing” from the page header | | |
| Description: | Landlord, Tenant, Guest can access property list page to view list of property in the system. | | |
| Preconditions: | None | | |
| Postconditions: | The user is redirected to the property list page. | | |
| Normal Flow: | **7.0 Property List Page**  1. Landlord, Tenant and Guest click on “Property Listing” from the page header.  2. System redirect to property list page. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Estimate to be many times per day | | |
| Business Rules: | BR-21 | | |
| Other Information: | The system should ensure that property listings load quickly, even with a large number of listings. | | |
| Assumptions: | View property list functionality is functioning correctly. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-21 | Property Information | Each item in the "Property List" must provide detailed information about the property, including images, rental price, location, area, number of bedrooms, promotions. |

### 4.2 UC08\_View property detail

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC08 - View property detail | | |
| Created By: | MinhNT | Date Created: |  |
| Primary Actor: | Landlord, Tenant, Guest | Secondary Actors: | None |
| Trigger: | Landlord, Tenant and Guest click on the property. | | |
| Description: | Landlord, Tenant, Guest want access property detail page to view detailed information of a property in the system. | | |
| Preconditions: | None | | |
| Postconditions: | The user is redirected to the property detail page of chosen property. | | |
| Normal Flow: | **8.0 Property detail page**  1. Landlord, Tenant and Guest access the home page.  2. Landlord, Tenant and Guest click on a property that appears in the system.  3. System redirect to the property detail page of chosen property. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Estimate to be many times per day | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | It is assumed that the property being viewed is currently available for rent. | | |

#### b. Business Rules

None

### 4.3 UC09\_Search property

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC09 - Search property | | |
| Created By: | LongNP | Date Created: | 23/9/2023 |
| Primary Actor: | Landlord, Tenant, Guest | Secondary Actors: | None |
| Trigger: | Landlord , Tenant and Guest enter some keywords of property, choose property type and click on button search from the page header. | | |
| Description: | Landlord, Tenant, Guest want to access property lists that match with the input. | | |
| Preconditions: | None | | |
| Postconditions: | System shows list of properties. | | |
| Normal Flow: | **9.0 Search property**  1. Landlord, Tenant and Guest types keyword into search textbox, choose property type and click on icon search from the header page.  2. System shows list of properties that match input. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Estimate to be several times per day | | |
| Business Rules: | BR-21, BR-22 | | |
| Other Information: | None | | |
| Assumptions: | Users are expected to enter valid search criteria.  Search property functionality is functioning correctly. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-21 | Property Information | Each item in the "Property List" must provide detailed information about the property, including images, rental price, location, area, number of bedrooms, promotions. |
| BR-22 | List Property | The view list property functionality should support filters and display matching results effectively. |

### 4.4 UC10\_Filter property

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC10 - Filter property | | |
| Created By: | LongNP | Date Created: | 23/9/2023 |
| Primary Actor: | Landlord, Tenant, Guest | Secondary Actors: | None |
| Trigger: | Landlord , Tenant and Guest choose some criteria for the property and click on the “Filter” button. | | |
| Description: | Landlord, Tenant, Guest want to get a property list that meets the criteria. | | |
| Preconditions: | None. | | |
| Postconditions: | The system returns a property list that meets the criteria. | | |
| Normal Flow: | **10.0 Filter property**  1. Landlord, Tenant and Guest access property list page.  2. Landlord, Tenant and Guest choose some criteria that apply for the current property list.  3. Landlord, Tenant and Guest click on the “Filter” button to apply filters.  4. The system returns some properties from the current list meet the criteria. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Estimate to be several times per day | | |
| Business Rules: | BR-21, BR-22 | | |
| Other Information: | None | | |
| Assumptions: | Users are expected to select filter criteria for meaningful results.  Filter property functionality is functioning correctly. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-21 | Property Information | Each item in the "Property List" must provide detailed information about the property, including images, rental price, location, area, number of bedrooms. |
| BR-22 | List Property | The view list property functionality should support filters and display matching results effectively. |

## 5. Rental Application Feature

### 5.1. UC11\_Create rental application

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC11 - Create rental application | | |
| Created By: | MinhNT | Date Created: | 23/9/2023 |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | Tenant click on “Submit Request” from the property detail page to create a rental application for the current property. | | |
| Description: | Tenant want to send a rental application to Landlord to rent the property. | | |
| Preconditions: | The user has successfully logged into the system with role is Tenant | | |
| Postconditions: | The system creates an application for that property and sends it to Landlord. | | |
| Normal Flow: | **11.0 Create rental application**  1. Tenant choose a property in the system.  2. Tenant access property detail page of a property.  3. Tenant click on “Submit Request” from the property detail page.  4. The system creates an application for that property and sends it to Landlord. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Estimate to be several times per week | | |
| Business Rules: | None | | |
| Other Information: | The system ensures that the personal information is securely stored and transmitted. | | |
| Assumptions: | The user has reviewed the property details and is interested in applying for a specific property.  The system provides clear guidelines and requirements for the rental application process. | | |

#### b. Business Rules

None

### 5.2. UC12\_Cancel rental application

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC12 - Cancel rental application | | |
| Created By: | MinhNT | Date Created: | 23/9/2023 |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | Tenant click on “Cancel” from the application history. | | |
| Description: | Tenant want to cancel an application that is being processed by Landlord. | | |
| Preconditions: | The user has successfully logged into the system with role is Tenant | | |
| Postconditions: | The system sets the status of the application to cancel. | | |
| Normal Flow: | **12.0 Cancel rental application**  1. Tenant click on “Application History” from the page header.  2. System redirect to application history page.  3. Tenant click on “Cancel” of a particular application from the application history page.  4. The system set the status of that application to cancel. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | One time an application | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### 

#### b. Business Rules

None

### 5.2. UC13\_View application history

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC13 - View application history | | |
| Created By: | MinhNT | Date Created: | 24/9/2023 |
| Primary Actor: | Tenant, Landlord | Secondary Actors: | None |
| Trigger: | User requests to access their application history. | | |
| Description: | It provides a record of the application that the user submitted or received. | | |
| Preconditions: | PRE-1. The user must be logged into their account. | | |
| Postconditions: | POST-1. The user is presented with a list or summary of their rental application history.  POST-2. For each application entry, the user can view application status, property information and submission date. | | |
| Normal Flow: | **13.0 View application history**  1. The user selects “Application History” on the menu.  2. The system retrieves and displays a list of the rental applications. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Usually | | |
| Business Rules: | BR-17 | | |
| Other Information: | The system may offer search and filter options for users to quickly locate specific applications. | | |
| Assumptions: | None | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| BR-17 | Access to Applications | Users can only view the application associated with their own account |

### 5.2. UC14\_View rental application

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC14 - View rental application | | |
| Created By: | MinhNT | Date Created: | 24/9/2023 |
| Primary Actor: | Tenant, Landlord | Secondary Actors: | None |
| Trigger: | The user click on “Rental Application” on the Dashboard | | |
| Description: | Both tenants and landlords can access and review rental applications submitted by tenants. Tenants can view the status and details of their submitted applications, while landlords can review and assess tenant applications for their rental properties. | | |
| Preconditions: | PRE-1. The user attempting to view the rental application must be logged into the system.  PRE-2. There must be at least one rental application submitted for the property. | | |
| Postconditions: | POST-1. The rental application's details are displayed to the user.  POST-2. The user sees the current status and details of the rental application. | | |
| Normal Flow: | **14.0 View a tenant application**  1. The user navigates to the dashboard.  2. The user selects “Rental Application”.  3. The system presents a list of rental applications.  4. The user selects a particular rental application from the list.  5. The system displays the rental application details. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Several times a week | | |
| Business Rules: | BR-17 | | |
| Other Information: | Given the sensitive nature of the information in rental applications, data security and privacy measures should be in place to protect applicant information and comply with relevant data protection regulations. | | |
| Assumptions: | It is assumed that tenants have submitted their rental applications through the system. | | |

#### b. Business Rules

None

### 5.3. UC15\_Accept/Reject tenant application

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC15 - Accept/Reject tenant application | | |
| Created By: | MinhNT | Date Created: | 24/9/2023 |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | The Landlord needs to make a decision regarding the tenant's application | | |
| Description: | When the Tenant submits an application, the Landlord can access the application details then the Landlord has the option to accept or reject the application. | | |
| Preconditions: | PRE-1. The Landlord must log into their account.  PRE-2. There must be at least one tenant application submitted for the property with status “Processing”. | | |
| Postconditions: | POST-1. The tenant application's status changes. | | |
| Normal Flow: | **15.0 Accept or reject tenant application**  1. The user navigates to the dashboard.  2. The user selects “Tenant Application”.  3. The system presents a list of tenant applications.  4. The user selects a particular tenant application with status “Processing” from the list.  5. The system displays the tenant application details.  6. The user makes decisions(accept or reject). | | |
| Alternative Flows: | None | | |
| Exceptions: | **15.0.E1** **Tenant cancels application while Landlord views application detail and does not make decision**  1. The system displayed a message: The rental application was canceled.  2. The system reloads the application with status “Canceled”. | | |
| Priority: | High | | |
| Frequency of Use: | 1-2 times per application | | |
| Business Rules: | BR-17, BR-20 | | |
| Other Information: | When a decision is made (acceptance or rejection), the tenant should be automatically notified through the system. | | |
| Assumptions: | Tenant applications are assumed to be complete and accurate.  Landlords have predefined criteria and rental requirements against which they evaluate tenant applications. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| BR-20 | Equal Consideration | All applications should be evaluated fairly and objectively based on the established criteria. |

## 

## 6. Property Management Feature

### 6.1 UC16\_View posted properties

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC16 - View posted properties | | |
| Created By: | MinhNT | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | The Landlord wants to view the properties they have listed on the system. | | |
| Description: | The Landlord can view a list of rental properties that they have posted on the system. | | |
| Preconditions: | PRE-1. The Landlord must log into their account. | | |
| Postconditions: | POST-1. The landlord has successfully viewed the list of properties they have posted on the system. | | |
| Normal Flow: | **16.0 View posted properties**  1. The Landlord navigates to the "Manage Properties" section of the system.  2. The system displays a list of all the rental properties that the Landlord has posted on the system. | | |
| Alternative Flows: | None | | |
| Exceptions: | **16.0.E1** **The Landlord does not post any rental property**  1. The system displayed a message: There is no property. | | |
| Priority: | High | | |
| Frequency of Use: | Monthly | | |
| Business Rules: | BR-30 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-30 |  | The Landlord should be able to search the posted properties for ease of use. |

### 6.2 UC17\_Post new rental property

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC17 - Post new rental property | | |
| Created By: | HieuND | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | The Landlord creates a listing for a new rental property. | | |
| Description: | The Landlord adds a new rental property listing to the system. | | |
| Preconditions: | PRE-1. The Landlord must log into their account. | | |
| Postconditions: | POST-1. A new rental property listing has been added to the system. | | |
| Normal Flow: | **17.0 Post new rental property**  1. The Landlord navigates to the "Manage Properties" section of the system.  2. The system displays a list of all the rental properties that the Landlord has posted on the system.  3. The Landlord selects “Add Listing”.  4. The system presents a form for creating a new rental property listing.  5. The Landlord provides the following information about the rental property.  6. The Landlord submits the rental property listing to the system. (see 17.0.E1 and 17.0.E2)  7. The system validates the information and stores it in the database. | | |
| Alternative Flows: | None | | |
| Exceptions: | **17.0.E1** **The information in form is invalid**  1. The system emphasizes invalid fields.  2. The system displays the message: The information is invalid. Please try again.  **17.0.E2. There is not enough point to create a new post**  1. The system displays a message: Insufficient point. | | |
| Priority: | High | | |
| Frequency of Use: | Every few months | | |
| Business Rules: | BR-13, BR-14, BR-23 | | |
| Other Information: | None | | |
| Assumptions: | The platform's post new rental property functionality is functioning correctly. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-13 |  | Most input form must have validation |

|  |  |  |
| --- | --- | --- |
| BR-14 |  | When add new property ,.. , must fill all information in form with validation |

|  |  |  |
| --- | --- | --- |
| BR-23 |  | The system should ensure that the Landlord has enough points in their account to cover the posting fee before proceeding with publication. |

### 6.3 UC18\_Edit existing property

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC18 - Edit existing property | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | The Landlord who owns the property decides to update or modify the property listing information | | |
| Description: | The Landlord can update or modify the property listing information to ensure that property listings remain accurate and up-to-date, reflecting any changes in property details. | | |
| Preconditions: | PRE-1. The Landlord must log into their account.  PRE-2. The Landlord has access to the property listing they want to edit. | | |
| Postconditions: | POST-1. The property listing is updated with the new information.  POST-2. Any tenant viewing the property will see the updated details. | | |
| Normal Flow: | **18.0 Edit a property through Manage Properties**  1. The Landlord navigates to the "Manage Properties" section of the system. (see 18.1)  2. The system displays a list of all the rental properties that the landlord has posted on the system.  3. The Landlord selects the specific property listing they wish to edit.  4. The system presents an editable form with fields for property details.  5. The Landlord updates property information.  6. The Landlord submits the updated property listing. (see 18.0.E1)  7. The system validates the changes and updates the property listing in the database with the new information.  8. The system sends a notification to the actor confirming that the property listing has been successfully updated. | | |
| Alternative Flows: | **18.1 Edit a property through property details**  1.The Landlord searches for property by the feature “Search Property”.  2. The system displays a list of properties that match with search criteria.  3. The Landlord selects their own property that is needed to modify.  4. The system displays the property details.  5. The Landlord selects “Edit”.  6. Return to step 4 of normal flow. | | |
| Exceptions: | **18.0.E1** **The information in form is invalid**  1. The system emphasizes invalid fields.  2. The system displays the message: The information is invalid. Please try again. | | |
| Priority: | High | | |
| Frequency of Use: | Every few months | | |
| Business Rules: | BR-13, BR-14, BR-16 | | |
| Other Information: | None | | |
| Assumptions: | The platform’s edit existing property functionality is functioning correctly. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-16 |  | Price of property will change by the last edit. |

### 6.4 UC19\_Remove from property listing

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC19- Remove from property listing | | |
| Created By: | LongNP | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: |  | | |
| Description: | The Landlord removes a rental property listing from the system. Removing a property listing indicates that the property is no longer available for rent through the platform. | | |
| Preconditions: | PRE-1. The Landlord must log into their account.  PRE-2. The Landlord has access to the property listing management. | | |
| Postconditions: | POST-1. The property listing is no longer available for rent on the platform.  POST-2. The system retains a record of the removed listing for historical purposes. | | |
| Normal Flow: | **19.0 Remove a property through Manage Properties**  1. The Landlord navigates to the "Manage Properties" section of the system. (see 19.1)  2. The system displays a list of all the rental properties that the landlord has posted on the system.  3. The Landlord selects the specific property listing they want to remove.  4. The system presents the details of the selected property listing.  5. The Landlord confirms their intention to remove the property listing.  6. The system prompts the actor to provide a reason for removing the listing.  7. The Landlord confirms the removal by clicking a "Remove" button.  8. The system updates the property listing status to indicate that it has been removed from the platform. | | |
| Alternative Flows: | **19.1 Remove a property through property details**  1. The Landlord searches for property by the feature “Search Property”.  2. The system displays a list of properties that match with search criteria.  3.The Landlord selects their own property that is needed to modify.  4. The system displays the property details.  5. The Landlord selects “Remove”.  6. Return to step 5 of normal flow. | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Every few months | | |
| Business Rules: | BR-15 | | |
| Other Information: | None | | |
| Assumptions: | The platform’s Remove from property listing functionality is functioning correctly. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-15 |  | When deleting a product, provider,.., landlord must confirm that they want to delete that product. |

## 7. Payment feature

### 7.1. UC20\_Pay for post

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC20 - Pay for Post | | |
| Created By: | LongNP | Date Created: | 23/9/2023 |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | When Landlord submit a new house rental post | | |
| Description: | Landlord creates a house rental post on the website and making payment (point) for that post to be published | | |
| Preconditions: | The landlord is registered and logged into their account on the house rental website  The landlord has filled out all the necessary details and confirm their intent to publish the post  Landlord has enough points in account to cover the posting fee | | |
| Postconditions: | The landlord’s house rental post is published on the website and pay for posting fee | | |
| Normal Flow: | **20.0 Pay for Post**  1. The landlord logged in the system.  2. The landlord clicks “post property” at the top right corner and fills the form.  3. The landlord clicks “post property” button to submit form.  4. The website processes the payment by minus the point in landlord’s account.  5. The post is published. | | |
| Alternative Flows: | The landlord’s house rental post is not published and receives an error message. | | |
| Exceptions: | Payment failure because of insufficient points or other payment-related issues  Landlord decides to cancel the post | | |
| Priority: | High | | |
| Frequency of Use: | Usually | | |
| Business Rules: | BR-23 | | |
| Other Information: | None | | |
| Assumptions: | Payment functionality is functioning correctly. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-23 | Account Points | The system should ensure that the Landlord has enough points in their account to cover the posting fee before proceeding with publication. |

### 7.2. UC21\_View transaction history

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC21 - View transaction history | | |
| Created By: | LongNP | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | When Landlord chooses “View transaction history” option in profile page | | |
| Description: | Landlord view their transaction history to review past payment and related details | | |
| Preconditions: | The landlord is registered and logged into their account on the house rental website | | |
| Postconditions: | The landlord can view a detailed transaction history including payment dates, type, amounts and post | | |
| Normal Flow: | **21.0 View transaction history**  1. The landlord clicks on "View Transaction History" in profile page  2. The website displays a list of transaction history including payment dates, type, amounts and post.  3. The landlord can filter transactions by date, type. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | BR-24 | | |
| Other Information: | None | | |
| Assumptions: | Internet connection is available | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-24 | Transaction history | The Landlord should be able to filter and search the transaction history for ease of use. |

### 7.3. UC22\_Add point to Landlord

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC22 - Add point to Landlord | | |
| Created By: | AnhTQT | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | Landlord |
| Trigger: | When Admin chooses “Add point” option | | |
| Description: | Admin adding points to a landlord's account on the house rental website, which may be used for posting new house | | |
| Preconditions: | The admin is logged into their admin account and has authorization to add points to landlord accounts. | | |
| Postconditions: | The specified number of points is added to the landlord's account. The landlord receives a notification when points are successfully added. | | |
| Normal Flow: | **22.0 Add point to landlord**  1. The landlord navigates to their account settings and click on “Manage Landlords”  2. The admin selects the specific landlord account to whom they want to add points.  3. The admin specifies number of points to add  4. The admin provides landlord with description for adding point | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | The platform’s add point functionality is functioning correctly | | |

#### b. Business Rules

None

### 7.4. UC23\_View system transaction

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC23 - View system transaction | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | When Admin chooses “View system transaction ” option | | |
| Description: | Admin views system transactions on the house rental website | | |
| Preconditions: | The admin is logged into their admin account and has authorization to view system transaction | | |
| Postconditions: | The admin can view detailed all transaction including detail like date, amount, type, and related information | | |
| Normal Flow: | **23.0 View system transaction**  1. The admin navigates to the admin dashboard and choose "System Transactions" section  2. The website displays a list of system transactions including detail like date, amount, type, and related information  3. The admin can view system transactions by filter like date, amount, type | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | The platform’s view system transaction functionality is functioning correctly. | | |

#### b. Business Rules

None

## 8. Wishlist feature

### 8.1. UC24\_Add to wishlist

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC024 - Add to wishlist | | |
| Created By: | AnhTQTHE172642 | Date Created: |  |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | The tenant decides to add a product to their wishlist for future consideration or purchase. | | |
| Description: | Tenant adds a property on the property list for consideration. | | |
| Preconditions: | PRE-1: The tenant is currently logged into their account | | |
| Postconditions: | POST-1: The selected property is added to the tenant’s wishlist | | |
| Normal Flow: | **24.0 Add to wishlist**  1. The tenant clicks the “Add to Wishlist” button on the property.  2. The system adds the property to the tenant’s wishlist | | |
| Alternative Flows: | None | | |
| Exceptions: | 24.0.E1: If there is a system error during the process of adding the property to the wishlist, the system displays an error message and advises the user to try again later. | | |
| Priority: | Low | | |
| Frequency of Use: | Estimated to be several times per day. | | |
| Business Rules: | BR-29 | | |
| Other Information: | None | | |
| Assumptions: | - The tenant is currently logged into their account.  - The platform’s “Add to Wishlist” functionality is functioning correctly. | | |

#### b. Business Rules

#### 

|  |  |  |
| --- | --- | --- |
| BR-29 | Wishlist action | Tenants can add, remove and view products to their wishlist for future consideration or purchase. |

### 8.2. UC25\_Remove from wishlist

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC025 - Remove from wishlist | | |
| Created By: | AnhTQTHE172642 | Date Created: |  |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | The tenant decides to remove a property from their wishlist. | | |
| Description: | Tenant remove a property from the property list | | |
| Preconditions: | PRE-1: The tenant is currently logged into their account.  PRE-2: The tenant has one or more properties in their wishlist | | |
| Postconditions: | POST-1: The selected property is removed from the user’s wishlist. | | |
| Normal Flow: | **25.0 Remove from wishlist**  1. The tenant clicks the “Remove” button.  2. The system removes the selected property from the user’s wishlist. | | |
| Alternative Flows: | None | | |
| Exceptions: | 25.0.E1: If there is a system error during the process of removing the property from the wishlist, the system displays an error message and advises the user to try again later. | | |
| Priority: | Low | | |
| Frequency of Use: | Estimated to be several times per day. | | |
| Business Rules: | BR-29 | | |
| Other Information: | None | | |
| Assumptions: | - The user is currently logged into their account.  - The platform's "Remove from Wishlist" functionality is functioning correctly. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-29 | Wishlist action | Tenants can add, remove and view products to their wishlist for future consideration or purchase. |

### 

### 8.3. UC26\_View wishlist

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC026 - View wishlist | | |
| Created By: |  | Date Created: | 20/09/2023 |
| Primary Actor: | AnhTQTHE172642 | Secondary Actors: | None |
| Trigger: | The tenant decides to view their wishlist to see the products they've saved for future consideration or purchase. | | |
| Description: | Display all properties in wishlist | | |
| Preconditions: | PRE-1: The tenant is currently logged into their account. | | |
| Postconditions: | POST-1: The tenant is presented with a list of products in their wishlist. | | |
| Normal Flow: | **26.0 View wishlist**  1. The tenant clicks on the “Wishlist” icon.  2. The system displays the tenant’s wishlist, showing a list of properties they have save.  3. The tenant scrolls through the wishlist to view the products they've saved.  4. The system allows the tenant to scroll through the wishlist and view the property details, including images, names, prices, and descriptions. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Low | | |
| Frequency of Use: | Estimated to be several times per day. | | |
| Business Rules: | BR-29 | | |
| Other Information: | None | | |
| Assumptions: | - The user is currently logged into their account.  - The platform's "View Wishlist" functionality is functioning correctly. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-29 | Wishlist action | Tenants can add, remove and view products to their wishlist for future consideration or purchase. |

## 9.Manage User Account

### 9.1. UC40\_View account

#### a. Functionalities

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC037 - View account | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin clicks on “User account management” on the menu and chooses “Account”. | | |
| Description: | Admin view all the information of any account on the system. | | |
| Preconditions: | The user must be logged into the system as an admin. | | |
| Postconditions: | Every account information show up | | |
| Normal Flow: | **37.0 View account**  1. Admin Action: Admin choose “User account management” on the menu.  2. System Response: The user account management screen displayed.  3. Admin Action: Admin clicks the “Account” button.  4. System Response: All the accounts displayed. | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Priority: | High | | |
| Frequency of Use: | Rarely | | |
| Business Rules: |  | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

*None*

### 9.2. UC33\_Add account

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC033 - Add account | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin clicks on “User account management” on the menu and chooses “Add account”. | | |
| Description: | Admin wants to add another admin account to the system. | | |
| Preconditions: | The user must be logged into the system as an admin. | | |
| Postconditions: | Another admin account is added to the system. | | |
| Normal Flow: | **33.0 Add an account**  1. Admin Action: Admin choose “User account management” on the menu.  2. System Response: The user account management screen displayed.  3. Admin Action: Admin clicks the “Add account” button.  4. System Response: The add account screen displayed.  5. User Action: Admin submits the additional account form.  6. System Response: The system processes the additional account information, creates a new admin account, and sends a verification email to the provided email address. | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Priority: | High | | |
| Frequency of Use: | Rarely | | |
| Business Rules: | BR-26 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-26 | Admin account | Admin cannot add an tenant/landlord account to an admin account |

### 9.3. UC34\_Disable account

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC034 - Disable account | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin clicks on “User account management” on the menu and chooses “Disable account”. | | |
| Description: | Admin wants to disable an user account. | | |
| Preconditions: | PRE-1. The user must be logged into the system as an admin.  PRE-2. The account that the admin wants to disable must exist. | | |
| Postconditions: | That account is disabled. | | |
| Normal Flow: | **34.0 Disable account**  1. Admin Action: Admin choose “User account management” on the menu.  2. System Response: The user account management screen displayed.  3. Admin Action: Admin select an account on the list and click “Disable”.  4. System Response: The system disables the selected account's features and hides it. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | BR-27 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-27 | Disable account | Only one highest admin can disable admin account |

### 9.4. UC35\_Ban account

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC035 - Ban account | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin clicks on “User account management” on the menu and chooses “Ban account”. | | |
| Description: | Admin can ban an user account for an amount of time because they violate the regulations | | |
| Preconditions: | PRE-1. The user must be logged into the system as an admin.  PRE-2. The account that the admin wants to ban must exist. | | |
| Postconditions: | That account has been banned. | | |
| Normal Flow: | **35.0 Ban account**  1. Admin Action: Admin choose “User account management” on the menu.  2. System Response: The user account management screen displayed.  3. Admin Action: Admin select an account on the list and click “Ban”.  4. System Response: The system does not allow the selected account to log in an amount of time. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | None | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

*None*

### 9.5. UC36\_Verify landlord account

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC036 - Verify landlord account | | |
| Created By: | ThinhLH | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin clicks on “User account management” on the menu and chooses “Verify”. | | |
| Description: | Admin wants to verify an account that is waiting to be verified as a landlord. | | |
| Preconditions: | The user must be logged into the system as an admin. | | |
| Postconditions: | That account has been verified with the landlord's authority. | | |
| Normal Flow: | **36.0 Verify landlord account**  1. Admin Action: Admin choose “User account management” on the menu.  2. System Response: The user account management screen displayed.  3. Admin Action: Admin select an account on the waiting verify list and click “Verify”.  4. System Response: The system adds landlord permissions to the selected account. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Must Have | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

*None*

## 10. Review feature

### 10.1. UC27\_Send feedback

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC027 - Send feedback | | |
| Created By: | ThinhLH | Date Created: | 20/09/2023 |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | When Tenant clicks on “Send feedback” button | | |
| Description: | Allow tenant to send feedback about property | | |
| Preconditions: | The tenant is registered and logged into their account and they have already signed contract with landlord | | |
| Postconditions: | The feedback will be sent to landlord and show on their review box | | |
| Normal Flow: | **27.0 Send feedback**  1. The tenant logins and browses property posts on the website.  2. The tenant can see others' feedback. Add new comments and edit comments as long as they have already signed a contract with that landlord. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | BR-19, BR-25 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-19 | Authenticity in feedback | Tenant can only send feedback or rate when they are already signed contract with landlord |
| BR-25 | Comment regulations | User cannot comment or chat with obscene words, these word would automatically change to “\*” |

### 10.2. UC28\_Rate property

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC028 - Rate property | | |
| Created By: | ThinhLH | Date Created: | 20/09/2023 |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | When Tenant clicks on “Rate” button | | |
| Description: | From 1 to 5, tenant can give scores to a property once | | |
| Preconditions: | The tenant is registered and logged into their account and they have already signed contract with landlord on the property they about to rate. | | |
| Postconditions: | The scores will be showed on the review box on specific property | | |
| Normal Flow: | **28.0 Rate property**  1. The tenant login and browses the property post that they want to rate  2. The tenant can rate and edit their rate  3. The tenant can see others rate and see the average rate of that property | | |
| Alternative Flows: | None | | |
| Exceptions: | The tenant can not send rate if they have not signed contract with the owner of the property that they want to rate | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | BR-19 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-19 | Authenticity in feedback | Tenant/landlord can only send feedback or rate when they are already signed contract with tenant/landlord |

### 10.3. UC38\_View feedback

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC038 - View feedback | | |
| Created By: | ThinhLH | Date Created: | 20/09/2023 |
| Primary Actor: | Tenant | Secondary Actors: | None |
| Trigger: | When Tenant clicks on “View feedback” button | | |
| Description: | Allow tenant to see feedback about property | | |
| Preconditions: | The tenant is registered and logged into their account | | |
| Postconditions: | The feedback will be showed to the screen | | |
| Normal Flow: | **38.0 View feedback**  1. The tenant logins and browses property posts on the website.  2. The tenant click on “view feedback” button and they can see all the feedback. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Low | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

*None*

## 11. Report feature

### 11.1 . UC29\_Report

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC029 - Report | | |
| Created By: | ThinhLH |  |  |
| Primary Actor: | Tenant or landlord |  |  |
| Trigger: | When Tenant clicks on “Report” button | | |
| Description: | The user can send a report to the administrator about any trouble while using this system, including web bugs, partners,... | | |
| Preconditions: | The user is registered and logged into their account | | |
| Postconditions: | The report will be sent directly to administrator | | |
| Normal Flow: | **29.0 Send report**  1. The user logins and use the report button to send report to administrator  2. Administrator receives and processes report  3. After the problem be solved, a notification will be sent to user | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Usually | | |
| Business Rules: | BR-25 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-25 | Comment regulations | Users cannot comment or chat with obscene words, these word would automatically change to “\*” |

## 12. Promotion feature

### 12.1. UC30\_Create promotion

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC030 - Create promotion | | |
| Created By: | AnhTQTHE172642 |  |  |
| Primary Actor: | Landlord |  |  |
| Trigger: | When Tenant clicks on “Promotion” button | | |
| Description: | The landlord can add promotion for their each property | | |
| Preconditions: | The landlord is registered and logged into their account. They have had at least one post | | |
| Postconditions: | The information of promotion is shows in specific property on property list | | |
| Normal Flow: | **30.0 Create promotion**  1. The landlord logins and browses to Manage Property and press “Manage promotion” to manage their Promotion.  2. The landlord can add, edit and delete promotion on their each property.  3. After the promotion added, the information about promotion will be added on the post, change the price of that property. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Sometimes | | |
| Business Rules: | BR-31 | | |
| Other Information: | The pricing and discount policies should be configurable and enforced by the Pricing System. | | |
| Assumptions: | It is assumed that the marketing manager has proper authorization to create and manage promotions. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-31 | Promotions condition | Promotions should adhere to predefined pricing and discount policies |

### 

### 12.2. UC31\_Edit promotion

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC031 - Edit promotion | | |
| Created By: | AnhTQTHE172642 | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | The landlord clicks on the “Promotion” button on their post and chooses “Edit”. | | |
| Description: | The landlord wants to change an existing promotion information. | | |
| Preconditions: | PRE-1. The user must be logged into the system as a landlord.  PRE-2. The promotion they want to edit must have already been created. | | |
| Postconditions: | The promotion has been successfully edited and appears according to the post. | | |
| Normal Flow: | **31.0 Edit promotion**  1. Landlord Action: The landlord navigates to their posts list page.  2. System Response: The landlord’s posts list page is displayed, showing their posts.  3. Landlord Action: The landlord selects one of their posts that has a promotion and clicks on the “Promotion” button.  4. System Response: The promotion management screen displayed.  5. Landlord Action: The landlord edits the promotion information then clicks the ”Submit edit” button.  6. System Response: The system validates the changes and ensures that the updated information conforms to the platform's requirements. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | BR-31 | | |
| Other Information: | The pricing and discount policies should be configurable and enforced by the Pricing System. | | |
| Assumptions: | It is assumed that the marketing manager has proper authorization to create and manage promotions. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-31 | Promotions condition | Promotions should adhere to predefined pricing and discount policies |

### 12.3. UC32\_Remove promotion

#### a. Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC032 - Remove promotion | | |
| Created By: | AnhTQTHE172642 | Date Created: |  |
| Primary Actor: | Landlord | Secondary Actors: | None |
| Trigger: | The landlord clicks on the “Promotion” button on their post. | | |
| Description: | The landlord wants to remove an existing promotion. | | |
| Preconditions: | PRE-1. The user must be logged into the system as a landlord.  PRE-2. The promotion they want to edit must have already been created. | | |
| Postconditions: | The promotion has been successfully removed. | | |
| Normal Flow: | **32.0 Remove promotion**  1. Landlord Action: The landlord navigates to their posts list page.  2. System Response: The landlord’s posts list page is displayed, showing their posts.  3. Landlord Action: The landlord selects one of their posts that has a promotion and clicks on the “Promotion” button.  4. System Response: The promotion management screen displayed.  5. Landlord Action: The landlord clicks the ”Remove” button.  6. System Response: The system validates the changes and ensures that the updated information conforms to the platform's requirements. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | BR-31 | | |
| Other Information: | The pricing and discount policies should be configurable and enforced by the Pricing System. | | |
| Assumptions: | It is assumed that the marketing manager has proper authorization to create and manage promotions. | | |

#### 

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| BR-31 | Promotions condition | Promotions should adhere to predefined pricing and discount policies |

### 

## 13. UC39\_View homepage

#### a. Functionalities

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC39 - View homepage | | |
| Created By: | AnhTQT | Date Created: | 20/09/2023 |
| Primary Actor: | Guest, Tenant, Landlord | Secondary Actors: | None |
| Trigger: | When users access the website address or click on the “Home” button. | | |
| Description: | Allow users to see the homepage. | | |
| Preconditions: | None | | |
| Postconditions: | The homepage is displayed on the screen. | | |
| Normal Flow: | **39.0 View homepage**  1. The user accesses the website address or after logging in to the website or after pressing the “Home” button.  2. The system displays the homepage screen. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### 

## 14. UC40\_Post management feature

#### a. Functionalities

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC40 - Disable post | | |
| Created By: | LongNP | Date Created: | 20/09/2023 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | When the administrator accesses the "Manage Post" page within the admin section and clicks on the "Status" column of a post that wants to disable. | | |
| Description: | Allow Admin to disable post if it violate the regulations | | |
| Preconditions: | Admin logged in the system | | |
| Postconditions: | Post is disabled | | |
| Normal Flow: | **40.0** Disable post  1. Admin logged in the system  2. Admin accesses the "Manage Post" page within the admin section  3. Admin clicks on "Status" column of the post they want to disable. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Sometimes | | |
| Business Rules: |  | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### 

#### b. Business Rules

None

## 10. Business template

Provide the business rules those are applied only to the use case

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| BR-01 |  | Account’s email must be valid. |
| BR-02 |  | Users must enter password twice to confirm password when changing password or sign up. |
| BR-03 |  | Account’s password must be at least 8 characters in length, 1 letter and 1 digit. |
| BR-04 |  | Account’s password must not be stored as a string sequence. Instead it must be hashed using a secure hash algorithm. |
| BR-05 |  | Cannot register with an email that has already been activated. |
| BR-06 |  | When registering or changing the password, the user must enter the new password twice. |
| BR-07 |  | After registering or forgetting password, guests must have an active account with a valid link sent to email. |
| BR-08 |  | If the email is stored in a database but not activated, the system sends a link to change the password of the existing account to that email when there is another registration request. |
| BR-09 |  | When changing password, the new password must not be the same as the current password. |
| BR-10 |  | Users must be logged in to log out or edit information or change password. |
| BR-11 |  | If a user wants to search, the user must fill in the search bar. |
| BR-12 |  | Date updated must after date created |
| BR-13 |  | Most input form must have validation |
| BR-14 |  | When add new property ,.. , must fill all information in form with validation |
| BR-15 |  | When deleting a product, provider,.., landlord must confirm that they want to delete that product. |
| BR-16 |  | Price of property will change by the last edit. |
| BR-17 | Access to Applications | Users can only view the application associated with their own account. They cannot access or view the application of other users. |
| BR-18 |  | Tenant can only review properties when they have occupied. |
| BR-19 | Authenticity in feedback | Tenant/landlord can only send feedback or rate when they are already signed contract with tenant/landlord |
| BR-20 | Equal Consideration | All applications should be evaluated fairly and objectively based on the established criteria. |
| BR-21 | Property Information | Each item in the "Property List" must provide detailed information about the property, including images, rental price, location, area, number of bedrooms. |
| BR-22 | List Property | The view list property functionality should support filters and display matching results effectively. |
| BR-23 | Account Points | The system should ensure that the Landlord has enough points in their account to cover the posting fee before proceeding with publication. |
| BR-24 | Transaction history | The Landlord should be able to filter and search the transaction history for ease of use. |
| BR-25 | Comment regulations | Users cannot comment or chat with obscene words, these word would automatically change to “\*” |
| BR-26 | Admin account | Admin cannot add an tenant/landlord account to an admin account |
| BR-27 | Disable account | Only one highest admin can disable admin account |
| BR-28 |  | Allow user to remember account in 1 hour without login again |
| BR-29 | Wishlist action | Tenants can add, remove and view products to their wishlist for future consideration or purchase. |
| BR-30 |  | The Landlord should be able to search the posted properties for ease of use. |
| BR-31 | Promotions condition | Promotions should adhere to predefined pricing and discount policies |

## 

# III. Design Specifications

## 1. User Registration

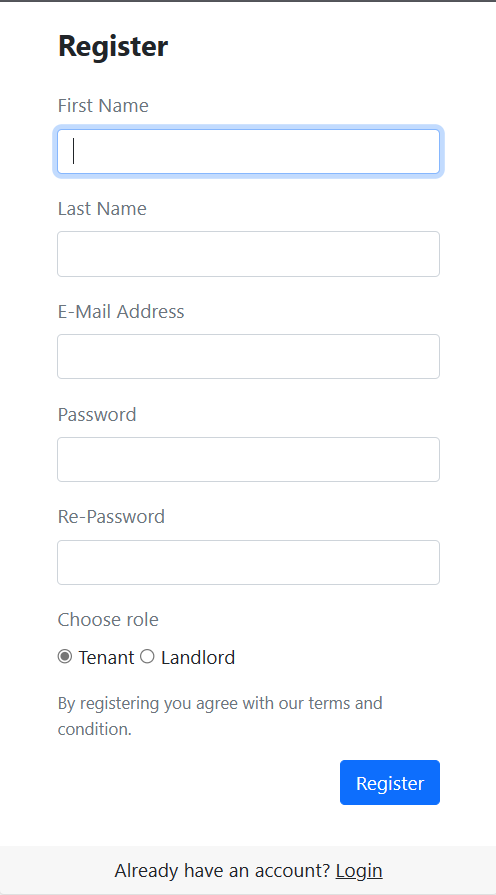
### 1.1 Register

1. User registration

This screen allows user to sign up for a new account in the system.

UC01\_Registration

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| First Name | Text Box String | This allows the user to input their first name. |
| Last Name | Text Box String | This allows the user to input their last name. |
| E-Mail Address | Text Box  String | This allows the user to input their email. |
| Password | Password Box | This allows the user to input their password. |
| Re-Password | Password Box | This forces the user to input their password once again to confirm their password. |
| Role | Radio input(Tenant/ Landlord) | This allows user to choose a role to become in the system. |
| Register | Button | Click to register account |
| Login | Hyperlink | Click to redirect to login page |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Tenant | C | Add information for new tenants. |
| Landlord | C | Add information for the landlords. |
| Token | CR | Verify the account by email. |
| Users | CRU | Check if there is an existed account or not, add new user , change status of user. |

##### SQL Commands

1/ Check if there is email that is associated with an activated account.

select \* from Users where email = ? and status != ?

2/ Check if there is email that is associated with a registered account with the same role.

select \* from Users where email = ? and role\_id = ?

3/ Insert user in database

INSERT INTO [dbo].[Users] ([email] ,[hashed\_password] ,[role\_id] ,[status])

VALUES (?,?,?,?)

4/ Insert information for user

INSERT INTO [dbo].[Landlord]

([id] ,[first\_name] ,[last\_name] ,[address] ,[phone] ,[civil\_id] ,[account\_points]) VALUES (?,?,?,?,?,?,?)

INSERT INTO [dbo].[Tenant]

([id] ,[first\_name] ,[last\_name] ,[address] ,[phone] )

VALUES (?,?,?,?,?,?,?)

5/Create a token for that account.

INSERT INTO [dbo].[Token] ([user\_id],[token] ,[expired\_date] ,[type])

VALUES (?,?,?,?)

6/ Get token of account

Select \* from Token where token = ?

7/ Get User of token

Select \* from Users where id = ?

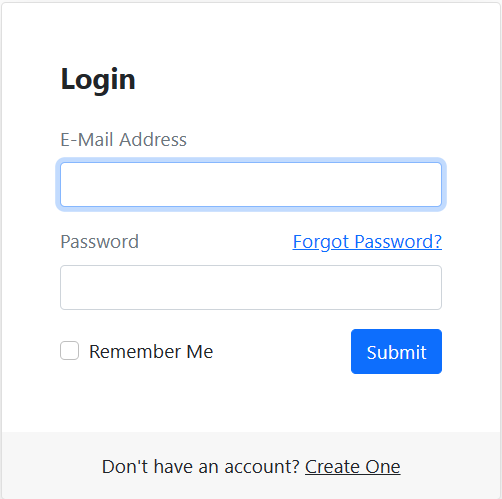
8/ Update status of User

Update Users set status = ? where id = ?

## 2. User Authentication

### 2.1 Log In

##### 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 |
| Create one | Hyperlink | 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 |
| Remember me | Check box | This allows the user to authorize in a time without login again. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Users | RU | Verify Email & Password information |
| User\_Banned | RD | Check if the user is banned or not. |

##### SQL Commands

1/ Verify username & password information.

SELECT \* FROM Users WHERE email = ? AND hashed\_password=? and status = ?

2/ Get information if User is banned.

Select \* from User\_Banned where id =?

3/ Update status of User if ban is time up.

Update Users set status = ? where id = ?

4/ Delete information ban if ban is time up.

Delete User\_Banned where id = ?

### 2.2 Log Out

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Logout | Button | User click to log out the account. |
| Logout | Hyperlink | This is for user to log out of their current  account.. |

##### Database Access

None.

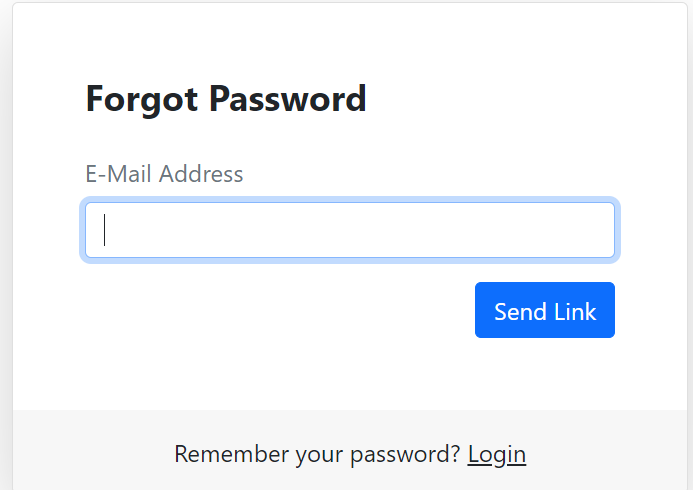
##### 

##### SQL Commands

None

### 2.3 Forgot Password

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Email | Text Box | This is for user to input their email that used to register. |
| Send Link | Button | Click this to get a password reset link in the email that entered. |
| Login | HyperLink | Redirect user to login page |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Users | U | Change new password when landlord or tenant forgot password |
| Token | CR | Create a new token and verify email when the user forgot the password. |

##### SQL Commands

1/ Create a new token for that account

INSERT INTO [dbo].[Token] ([email],[token] ,[expired\_date] ,[type])

VALUES (?,?,?,?)

2/ Get information of a token

Select \*from Token where token = ?

3/ Get User of a token

select \* from Users where id = ?

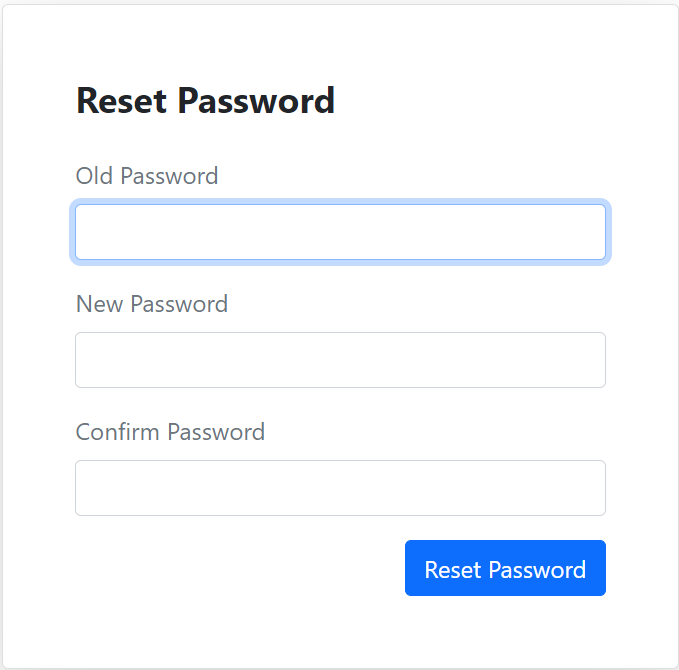
4/ Change password

Update Users set hashed\_password = ? where id = ?

## 3. User Authentication

### 3.1 Reset Password

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Old Password | Password Box | This is for user to input their password of the current logged in account. |
| New Password | Password Box | This is for user to input their new password. |
| Confirm Password | Password Box | This is for user to input their new password once again. |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Users | U | Change new password when user forgot password |

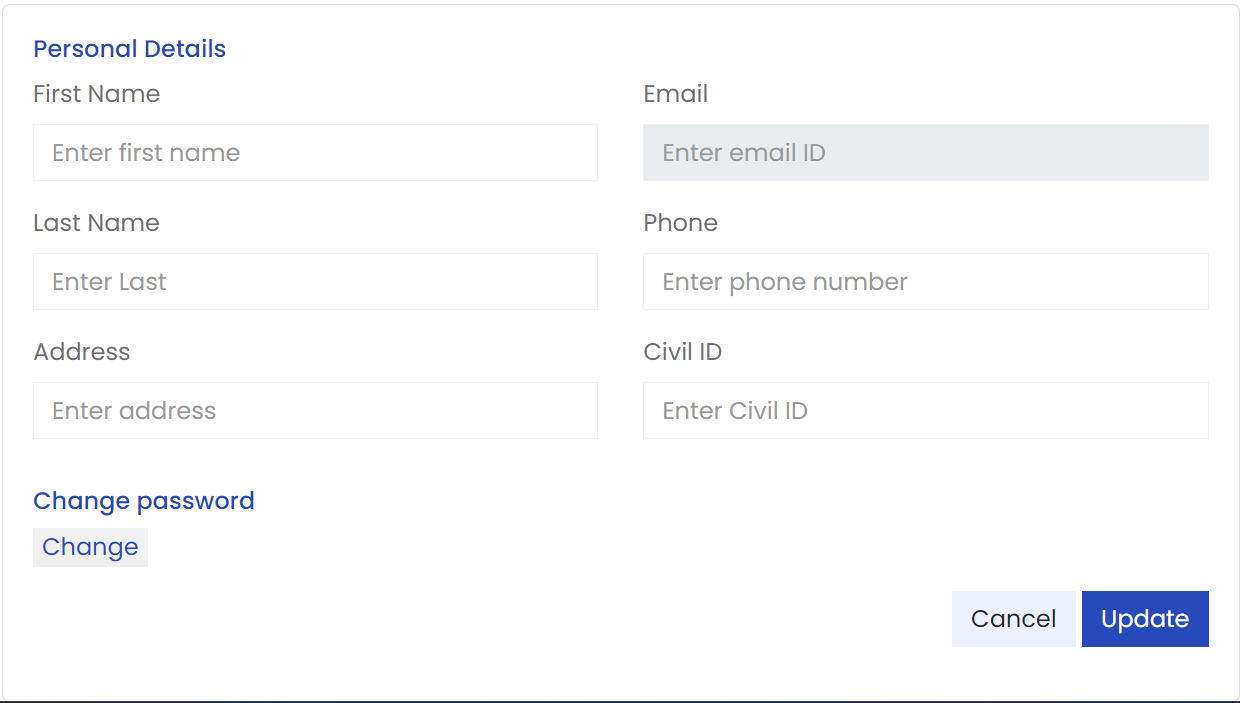
##### SQL Commands

1/ Change password

Update Users set hashed\_password = ? where id = ?

### 3.2 Edit profile

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| First Name | Text Box | This is for user to change their first name. |
| Last Name | Text Box | This is for user to change their last name. |
| Email | Text Box | This is for user view their email that is used for this account. |
| Phone | Text Box | This is for user to change their phone number. |
| Address | Text Box | This is for user to change their address number. |
| Civil ID | Text Box | This is for user to change their civil id. |
| Change | Button | Click to redirect the user to change the password page. |
| Cancel | Button | Click to reset information that the user has just entered. |
| Update | Button | Click to update user information. |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Tenant | U | Change information when the tenant wants to update information. |
| Landlord | U | Change information when the landlord wants to update information. |

##### SQL Commands

1/ Update information for Landlord

Update Landlord set first\_name = ?, last\_name = ?, address = ? , phone = ?, civil\_id = ? where id = ?

2/ Update information for Tenant

Update Tenant set first\_name = ?, last\_name = ?, address = ? , phone = ? where id =?

3/ Update information for Admin

Update Admin set first\_name = ?, last\_name = ? where id = ?

## 4. Property Listing Feature

### 4.1. View property list



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Property listing | Hyperlink | Redirect user to property listing page |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get information about property |
| Property\_type | R | Get information about property type |
| Post\_Image | R | Get thumbnail image of the property |

##### 

##### SQL Commands

1.Get information about property

select \* from Post

2.Get information about property type

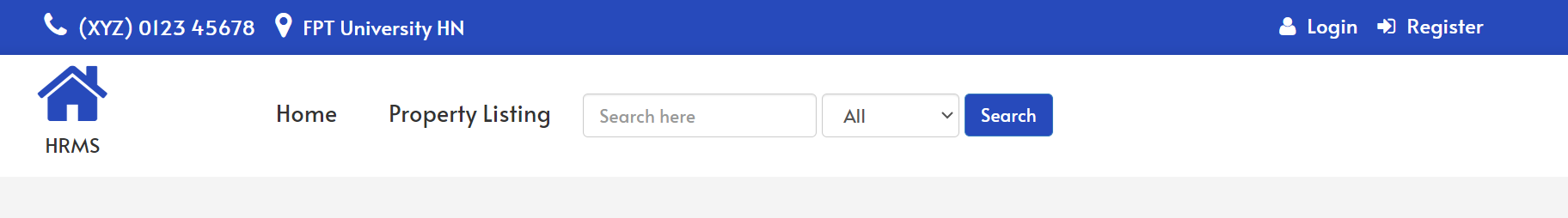
select \* from Property\_type

3.Get thumbnail image of the property

select \* from Post\_Image where post\_id = 1

### 4.2. Search property

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Search bar | Text Box | Enter the name of the property you want to search for. |
| Property type | Combo Box Single-Choice | Value: All, Nha tro, Chung cu  Allow user to choose type of property |
| Search | Button | Click this to search property |

##### 

##### Database Access

##### 

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get information about property |
| Property\_type | R | Get information about property type |
| Post\_Image | R | Get thumbnail image of the property |

##### 

##### SQL Commands

1.Get information about property

select \* from Post where name like ?

2.Get information about property type

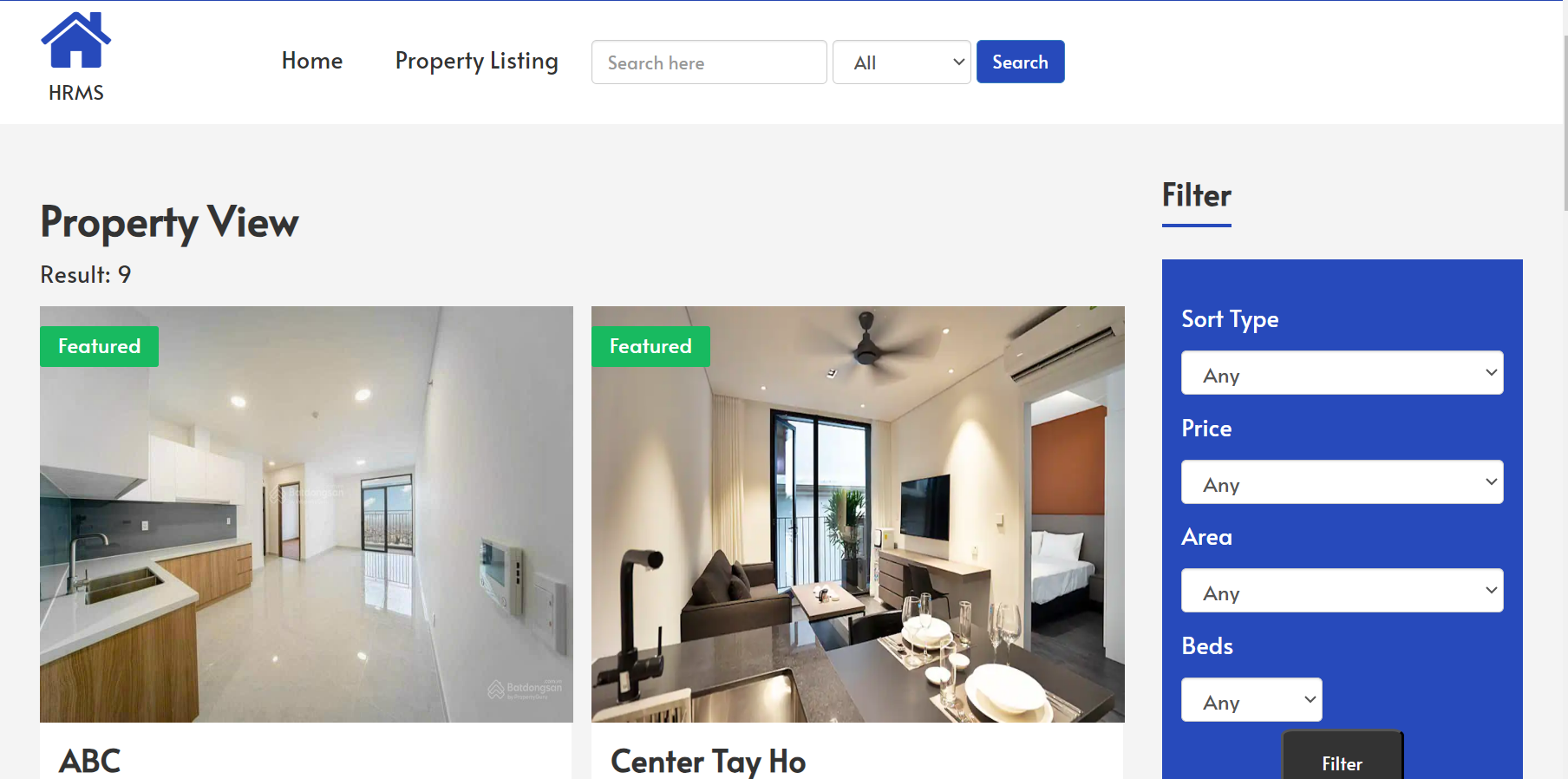
select \* from Property\_type where type\_id = ?

3.Get thumbnail image of the property

select \* from Post p join Post\_Image pi on p.id = pi.post\_id where img\_type = 1 and p.name like ?

### 4.3. Filter property

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Sort Type | Combo Box Single-Choice | Value: Any, Area small to large, Area large to small, Price low to high, Price high to low  Allow user to choose sort type |
| Price | Combo Box Single-Choice | Value: Any, <1000000 VND, 1000000-2000000 VND, 2000000-3000000 VND, >3000000 VND  This is for user to choose price range |
| Area | Combo Box Single-Choice | Value: Any, <1000000 VND, <15m2, 15-20m2  , 20-30m2, >30m2  This is for user to choose area range |
| Beds | Combo Box Single-Choice | Value: Any, number of bedrooms  Allow user to choose number of bedrooms |
| Filter | Button | Click this to submit criteria |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get information about property and number of bedrooms |
| Property\_type | R | Get information about property type |
| Promotions | R | Get promotion about property |
| Post\_Image | R | Get thumbnail image of the property |

##### 

##### SQL Commands

1.Get information about property

select \* from Post where name like ? and NumOfBedrooms = ?

and area between ? and ?

and price between ? and ?

order by ?

2.Get number of bedrooms

select distinct(NumOfBedrooms) from Post

3.Get information about property type

select \* from Property\_type where type\_id = ?

4.Get thumbnail image of the property

select \* from Post p join Post\_Image in p.id = pi.post\_id where img\_type = 1 and p.name like ?

### 4.4. View property detail

##### UI Design

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property details and recent properties |
| Landlord | R | Get information of landlord |
| Post\_Image | R | Get images of properties |
| Review | R | Get reviews of property |
| Tenant | R | Get name of reviewers |

##### 

##### SQL Commands

1. Get property details and recent properties

SELECT \* FROM [dbo].[Post] WHERE [id] = ?

SELECT TOP ? [id], [name], [price], [area], [NumOfBedrooms], [address]

FROM [dbo].[Post]

WHERE [status] = 1

ORDER BY [post\_start\_date] DESC

2. Get information of landlord

SELECT \* FROM [dbo].[Landlord] WHERE [id] = ?

3. Get images of properties

SELECT \* FROM [dbo].[Post\_Image] WHERE post\_id = ?

SELECT [img\_url] FROM [dbo].[Post\_Image] WHERE [post\_id] = ? AND [img\_type] = 1

4. Get reviews of property

SELECT [user\_id], [rating], [review] FROM [dbo].[Review] WHERE [property\_id] = ?

5. Get name of reviewers

SELECT [first\_name], [last\_name] FROM [dbo].[Tenant] WHERE [id] = ?

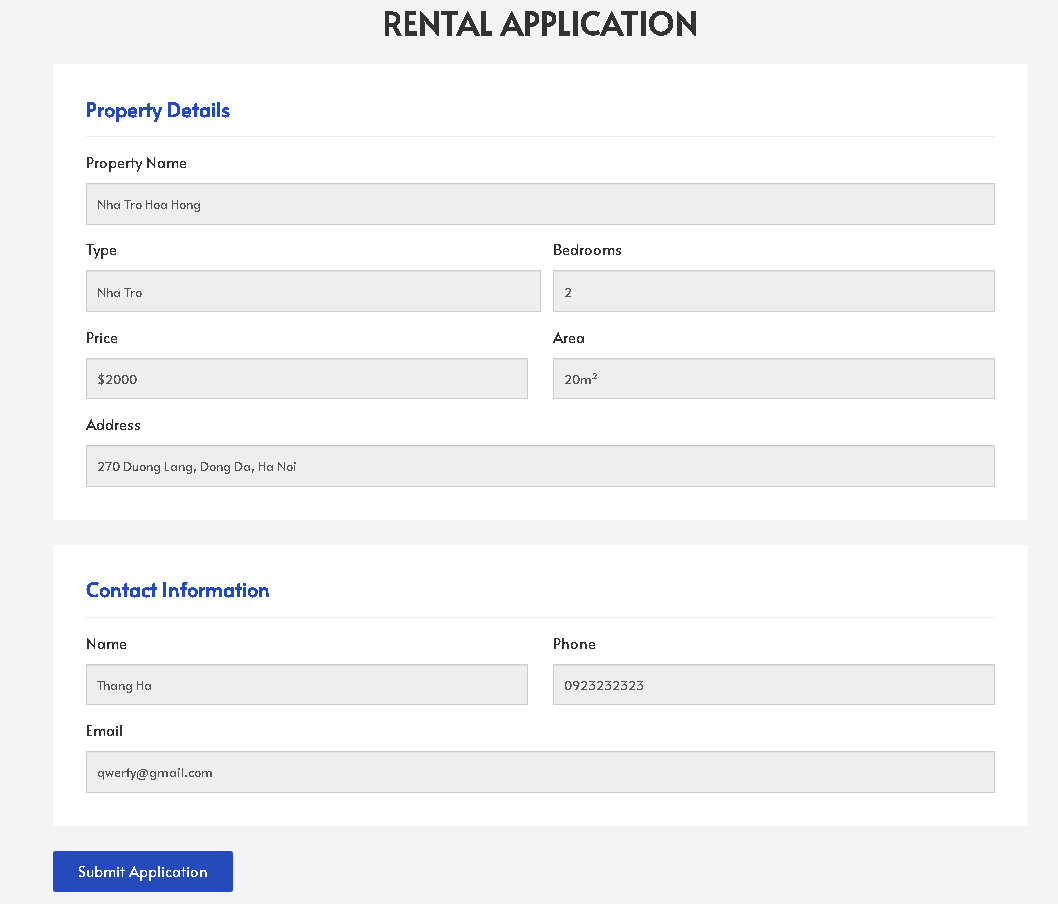
### 

## 5. Rental Application Feature

### 5.1. Application Processing

#### a. Create rental application

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| ***Property Details*** | | |
| Property Name | Text Box | Name of property. This field can not modify. |
| Type | Text Box | Type of property. This field can not modify. |
| Bedrooms | Number | Amount of bedrooms. This field can not modify. |
| Price | Text Box | Rental price of the property. This field can not modify. |
| Area | Text Box | Area of property. This field can not modify. |
| ***Contact Information*** | | |
| Name | Text Box | Full name of user. This field can not modify. |
| Phone | Text Box | Phone number of user. This field can not modify. |
| Email | Text Box | Email of user. This field can not modify. |
| Submit Application | Button | User clicks to submit and send the rental application to the owner of the property. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Orders | C | Create new rental application |

##### SQL Commands

1. Get property information

SELECT [name], [price], [type\_name], [NumOfBedrooms], [area], [address]

FROM [dbo].[Post] p

JOIN [dbo].[Property\_type] pt ON p.type = pt.type\_id

WHERE id = ?

2. Get tenant information

SELECT [id], [email], [first\_name], [last\_name], [phone] FROM [dbo].[Tenant] WHERE id = ?

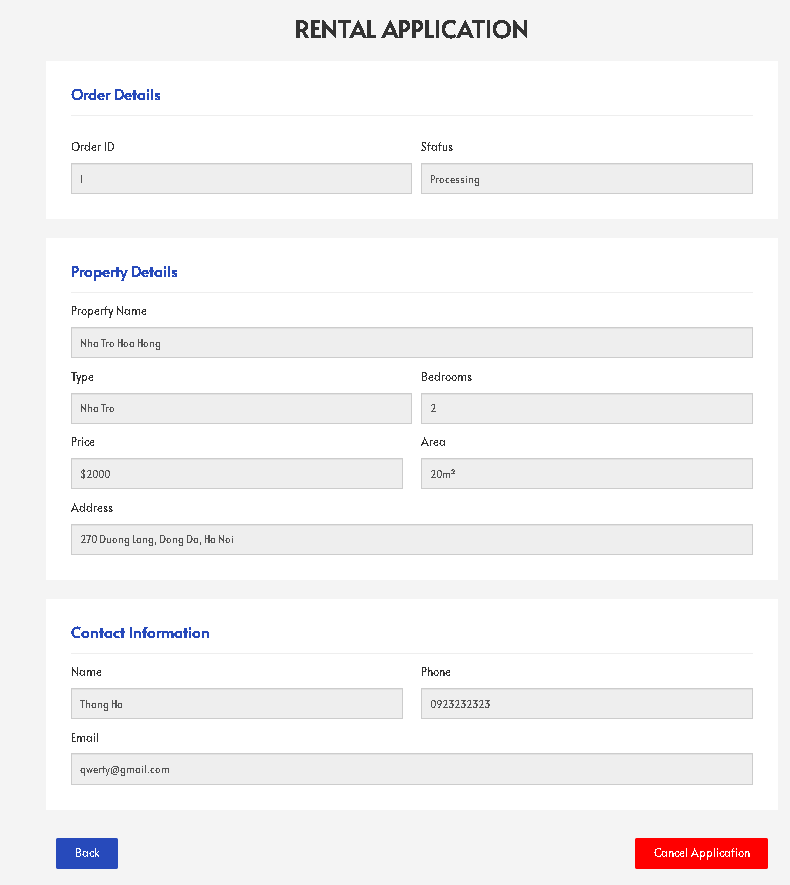
3. Create new rental application

INSERT INTO [dbo].[Orders] ([tenant\_id], [landlord\_id], [post\_id], [status])

VALUES (?, ?, ?, ‘Processing’)

#### b. Cancel rental application

##### UI Setting



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| ***Order Details*** | | |
| Order ID | Number | ID number of rental application. This field can not modify. |
| Status | Text Box | Status of rental application. This field can not modify. |
| ***Property Details*** | | |
| Property Name | Text Box | Name of property. This field can not modify. |
| Type | Text Box | Type of property. This field can not modify. |
| Bedrooms | Number | Amount of bedrooms. This field can not modify. |
| Price | Text Box | Rental price of the property. This field can not modify. |
| Area | Text Box | Area of property. This field can not modify. |
| ***Contact Information*** | | |
| Name | Text Box | Full name of user. This field can not modify. |
| Phone | Text Box | Phone number of user. This field can not modify. |
| Email | Text Box | Email of user. This field can not modify. |
| Back | Button | User clicks to go back to rental application history. |
| Cancel Application | Button | User clicks to cancel the rental application. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Orders | RU | Get information and change the status of the rental application. |

##### SQL Commands

1. Get property information

SELECT [name], [price], [type\_name], [NumOfBedrooms], [area], [address]

FROM [dbo].[Orders] o

JOIN [dbo].[Post] p ON o.post\_id = p.id

JOIN [dbo].[Property\_type] pt ON p.type = pt.type\_id

WHERE o.order\_id = ?

2. Get tenant information

SELECT [email], [first\_name], [last\_name], [phone]

FROM [dbo].[Orders] o

JOIN [dbo].[Tenant] t ON o.tenant\_id = t.id

WHERE o.order\_id = ?

3. Get information and change the status of the rental application

SELECT [order\_id], [status]

FROM [dbo].[Orders]

WHERE order\_id = ?

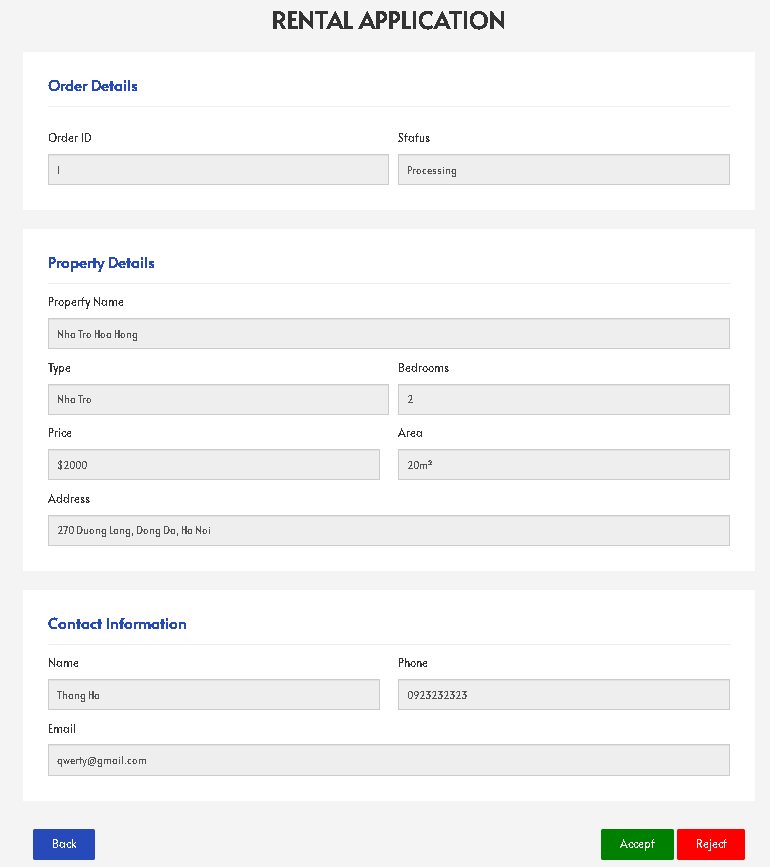
UPDATE [dbo].[Orders]

SET [status] = 'Canceled'

WHERE [order\_id] = ?

#### c. Accept/Reject tenant application

##### UI Setting



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| ***Order Details*** | | |
| Order ID | Number | ID number of rental application. This field can not modify. |
| Status | Text Box | Status of rental application. This field can not modify. |
| ***Property Details*** | | |
| Property Name | Text Box | Name of property. This field can not modify. |
| Type | Text Box | Type of property. This field can not modify. |
| Bedrooms | Number | Amount of bedrooms. This field can not modify. |
| Price | Text Box | Rental price of the property. This field can not modify. |
| Area | Text Box | Area of property. This field can not modify. |
| ***Contact Information*** | | |
| Name | Text Box | Full name of user. This field can not modify. |
| Phone | Text Box | Phone number of user. This field can not modify. |
| Email | Text Box | Email of user. This field can not modify. |
| Back | Button | User clicks to go back to rental application history. |
| Accept | Button | User clicks to accept the rental application. |
| Reject | Button | User clicks to reject the rental application. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Orders | RU | Get information and change the status of the rental application. |

##### SQL Commands

1. Get property information

SELECT [name], [price], [type\_name], [NumOfBedrooms], [area], [address]

FROM [dbo].[Orders] o

JOIN [dbo].[Post] p ON o.post\_id = p.id

JOIN [dbo].[Property\_type] pt ON p.type = pt.type\_id

WHERE o.order\_id = ?

2. Get tenant information

SELECT [email], [first\_name], [last\_name], [phone]

FROM [dbo].[Orders] o

JOIN [dbo].[Tenant] t ON o.tenant\_id = t.id

WHERE o.order\_id = ?

3. Get information and change the status of the rental application

SELECT [order\_id], [status]

FROM [dbo].[Orders]

WHERE order\_id = ?

UPDATE [dbo].[Orders]

SET [status] = 'Accepted'

WHERE [order\_id] = ?

UPDATE [dbo].[Orders]

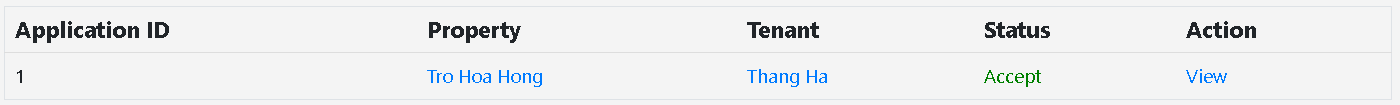
SET [status] = 'Rejected'

WHERE [order\_id] = ?

### 5.2. Application Management

#### a. View application history

##### UI Setting



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Application ID | Number | ID number of rental application |
| Property | HyperLink | Link to property of rental application |
| Tenant | HyperLink | Link to tenant of rental application |
| Status | Text Box | Status of rental application |
| Action | HyperLink | Actions that can be taken with the rental application |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Orders | R | Get the rental application information |

##### SQL Commands

1. Get property information

SELECT [id], [name]

FROM [dbo].[Orders] o

JOIN [dbo].[Post] p ON o.post\_id = p.id

2. Get tenant information

SELECT [id], [first\_name], [last\_name]

FROM [dbo].[Orders] o

JOIN [dbo].[Tenant] t ON o.tenant\_id = t.id

3. Get the rental application information

SELECT [order\_id], [status]

FROM [dbo].[Orders]

#### b. View tenant application

##### UI Setting

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| ***Order Details*** | | |
| Order ID | Number | ID number of rental application. This field can not modify. |
| Status | Text Box | Status of rental application. This field can not modify. |
| ***Property Details*** | | |
| Property Name | Text Box | Name of property. This field can not modify. |
| Type | Text Box | Type of property. This field can not modify. |
| Bedrooms | Number | Amount of bedrooms. This field can not modify. |
| Price | Text Box | Rental price of the property. This field can not modify. |
| Area | Text Box | Area of property. This field can not modify. |
| ***Contact Information*** | | |
| Name | Text Box | Full name of user. This field can not modify. |
| Phone | Text Box | Phone number of user. This field can not modify. |
| Email | Text Box | Email of user. This field can not modify. |
| Back | Button | User clicks to go back to rental application history. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Orders | R | Get the rental application information |

##### SQL Commands

1. Get property information

SELECT [name], [price], [type\_name], [NumOfBedrooms], [area], [address]

FROM [dbo].[Orders] o

JOIN [dbo].[Post] p ON o.post\_id = p.id

JOIN [dbo].[Property\_type] pt ON p.type = pt.type\_id

WHERE o.order\_id = ?

2. Get tenant information

SELECT [email], [first\_name], [last\_name], [phone]

FROM [dbo].[Orders] o

JOIN [dbo].[Tenant] t ON o.tenant\_id = t.id

WHERE o.order\_id = ?

3. Get the rental application information

SELECT [order\_id], [status]

FROM [dbo].[Orders]

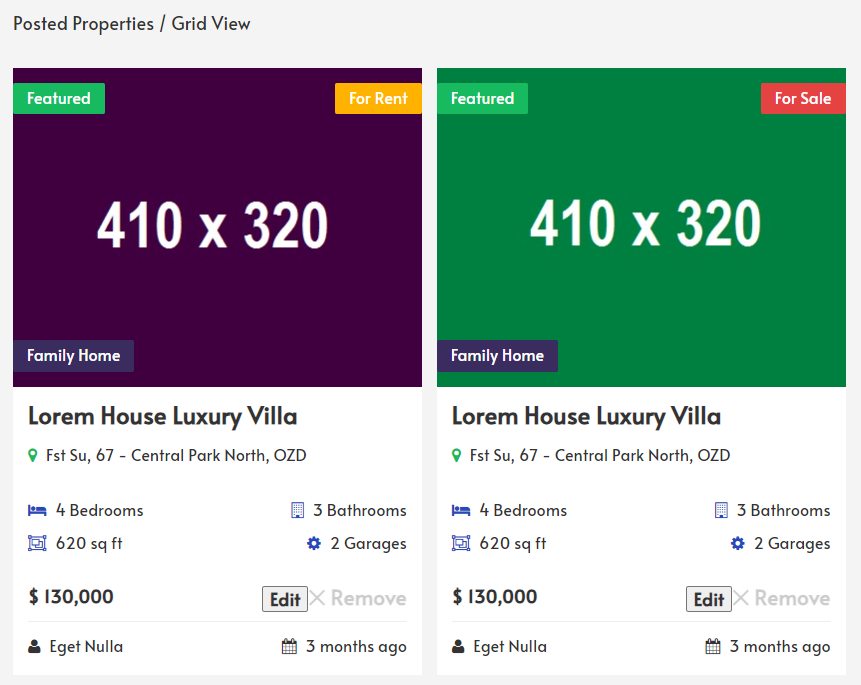
WHERE order\_id = ?

## 6. Property Management Feature

### 6.1. View posted properties

##### UI Design





|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Search bar | Text Box | Enter the name of the property you want to search for. |
| Property type | Combo Box Single-Choice | Value: Motel, Apartment  Allow user to choose type of property |
| Search | Button | Click to search property |
| Edit | Button | Click to edit property |
| Remove | Button | Click to remove property |

##### 

##### Database Access

##### 

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get information about property’s name, number of bedrooms. |
| Property\_type | R | Get information about property type. |
| Promotions | R | Get a promotion about property. |
| Post\_Image | R | Get thumbnail image of the property |

##### 

##### SQL Commands

1. Get information about property

SELECT \* FROM Post

1. Get information about property type

SELECT \* FROM Property\_type

1. Get information about promotions of property

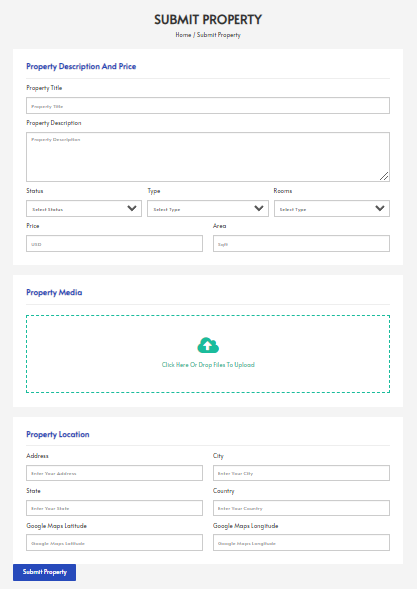
SELECT \* FROM Promotions

1. Get thumbnail image of the property

SELECT \* FROM Post\_Image WHERE post\_id = ?

### 6.2. Post new rental property

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Property title | Text Box | Enter the title of the property. |
| Property description | Text Box | Enter the description of the property. |
| Status | Combo Box Single-Choice | Value: Available, Unavailable  Allow Landlord to choose the status of the property. |
| Type | Combo Box Single-Choice | Value: Motel, Apartment  Allow User to choose type of property. |
| Rooms | Number | Amount of rooms. This field can not modify. |
| Price | Text Box | Rental price of the property. This field can not modify. |
| Area | Text Box | Area of property. This field can not modify. |
| Property location | Text Box | Location of property. This field can not modify. |
| Submit property | Button | The Landlord clicks to submit property. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Landlord | R | Get information about the landlord's point. |
| Post | C | Create new post property. |

##### SQL Commands

1. Get information about the landlord's point.

SELECT [account\_points]

FROM [dbo].[Landlord]

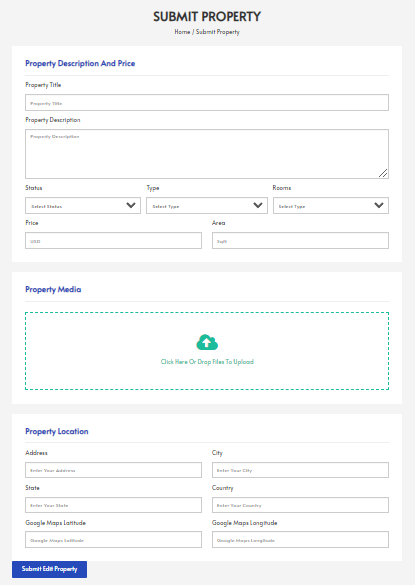
2.

INSERT INTO [dbo].[Post]

VALUES ?

### 6.3. Edit existing property

##### UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Property title | Text Box | Enter the title of the property. |
| Property description | Text Box | Enter the description of the property. |
| Status | Combo Box Single-Choice | Value: Available, Unavailable  Allow Landlord to choose the status of the property. |
| Type | Combo Box Single-Choice | Value: Motel, Apartment  Allow User to choose type of property. |
| Rooms | Number | Amount of rooms. This field can not modify. |
| Price | Text Box | Rental price of the property. This field can not modify. |
| Area | Text Box | Area of property. This field can not modify. |
| Property location | Text Box | Location of property. This field can not modify. |
| Submit Edited Property | Button | The Landlord clicks to submit edited property. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | U | Update new information about property. |

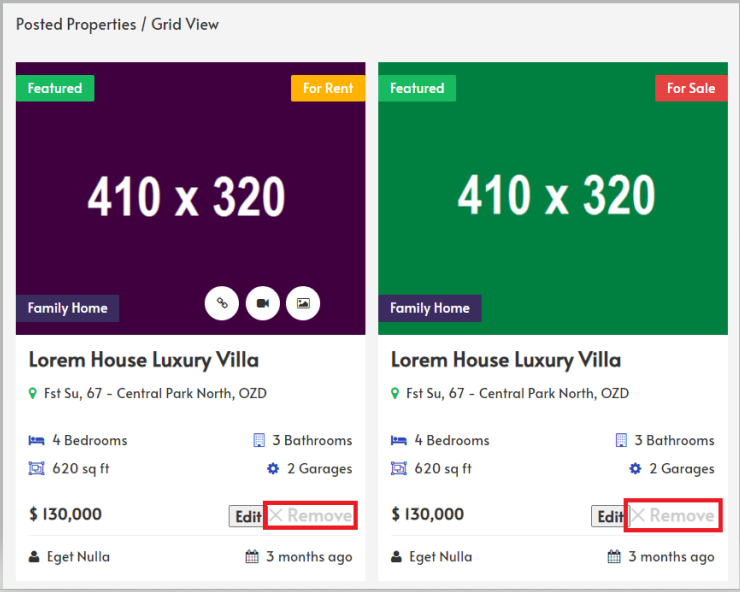
##### SQL Commands

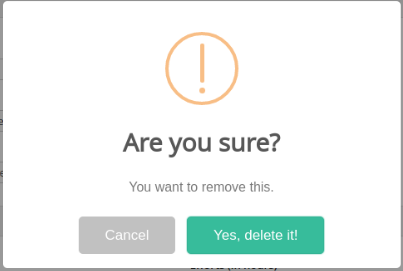
UPDATE [dbo].[Post]

SET ? WHERE id = ?

### 6.4. Remove from property listing

##### UI Design





|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Remove | Button | Click to remove property. |
| Cancel | Button | Click to cancel removing property. |
| Yes, remove it! | Button | Click to confirm remove property. |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | U | Update status of post property into disable. |

##### SQL Commands

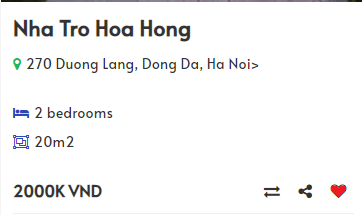
UPDATE [dbo].[Post]

SET [status] = Disable WHERE id = ?

## 7. Manage Wishlist

a. Add to Wishlist

**UI Design**



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Heart icon | button | After click this button, the property will add to wishlist |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Wishlist | RU | Get property information |

##### SQL Commands

1. Get property information

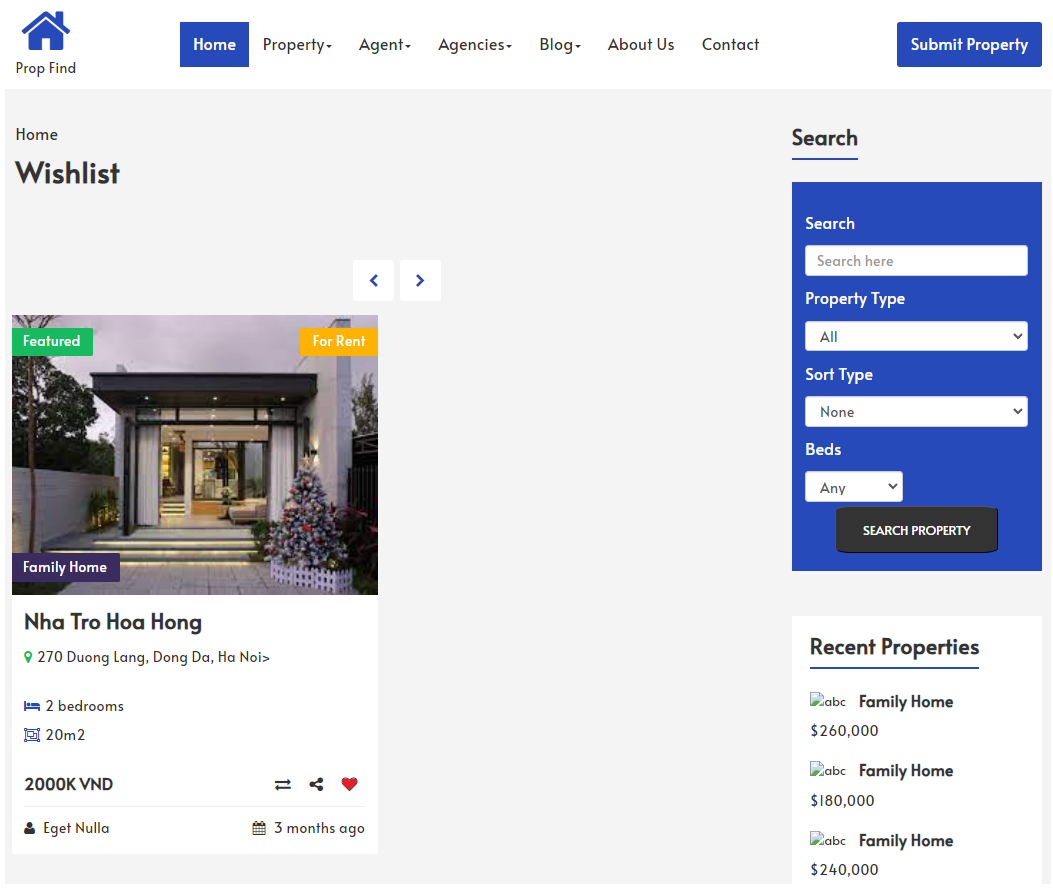
select \* from Post, Wishlist where Post.id = Wishlist.property\_id

2. Get tenant information

select \* from Tenant, Wishlist where Tenant.id = Wishlist.user\_id

b. View Wishlist

**UI Design**



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Wishlist | RU | Get property information |

##### SQL Commands

1. Get property information

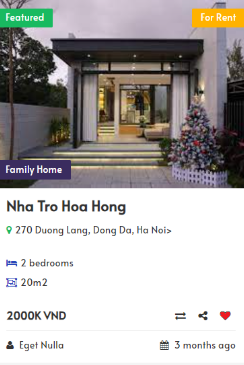
select \* from Post, Wishlist where Post.id = Wishlist.property\_id

2. Get tenant information

select \* from Tenant, Wishlist where Tenant.id = Wishlist.user\_id

c. Delete from Wishlist

UI Design



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Heart icon | button | After click this button, the property will remove from wishlist |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Tenant | R | Get tenant information |
| Wishlist | RU | Get property information |

##### SQL Commands

1. Get property information

select \* from Post, Wishlist where Post.id = Wishlist.property\_id

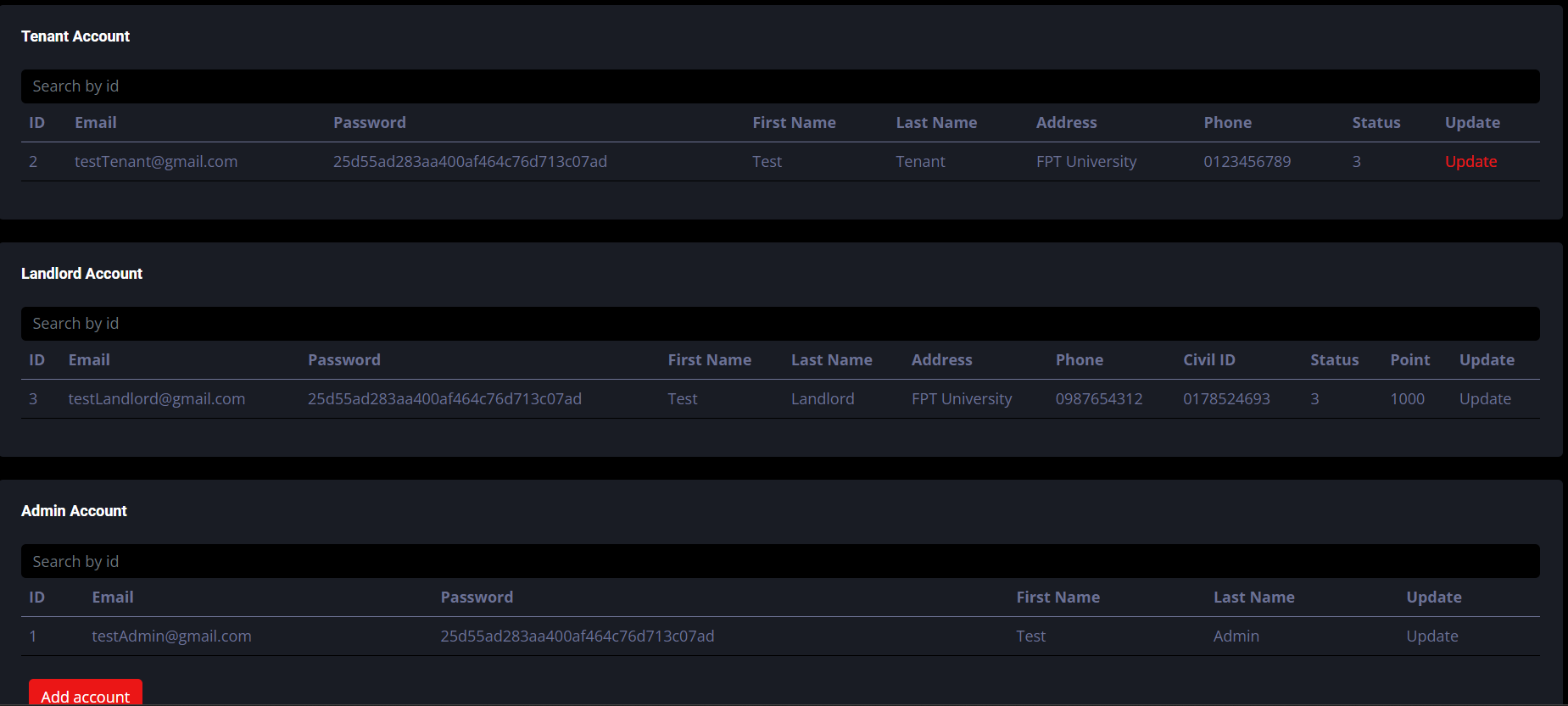
2. Get tenant information

select \* from Tenant, Wishlist where Tenant.id = Wishlist.user\_id

## 8. Admin management

### 8.1. Manage account

##### UI Design

****

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Disable account | Button | Deactivate landlord/tenant account indefinitely |
| Ban account | Button | Ban landlord/tenant account in a specific amount of time |
| Verify landlord account | Button | Approved landlord account |
| Add admin account | Hyperlink | Create an admin account |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Tenant table | R | Get all information about Tenant |
| Landlord table | R | Get all information about Landlord |
| Admin table | R | Get all information about Admin |

##### SQL Commands

**1.**

SELECT U.id, U.email, U.hashed\_password, A.first\_name, A.last\_name

FROM Users U

INNER JOIN Admin A ON U.id = A.id

**2.**

SELECT U.id, U.email, U.hashed\_password,

L.first\_name, L.last\_name, L.address,

L.phone, L.civil\_id, U.status, L.account\_points

FROM Users U

INNER JOIN Landlord L ON U.id = L.id

**3.**

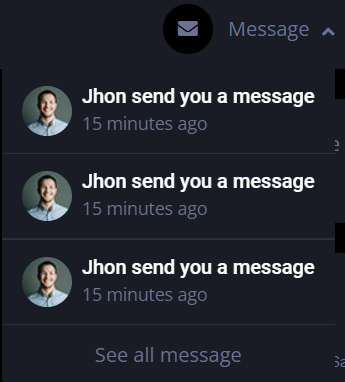
SELECT U.id, U.email, U.hashed\_password, T.first\_name, T.last\_name, T.address, T.phone, U.status

FROM Users U

INNER JOIN Tenant T ON U.id = T.id

### 8.2. Report management

##### UI Design

****

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Write message | Box | User can write message |
| Send message | Button | User can send message to admin |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Message | CR | Get or create message |

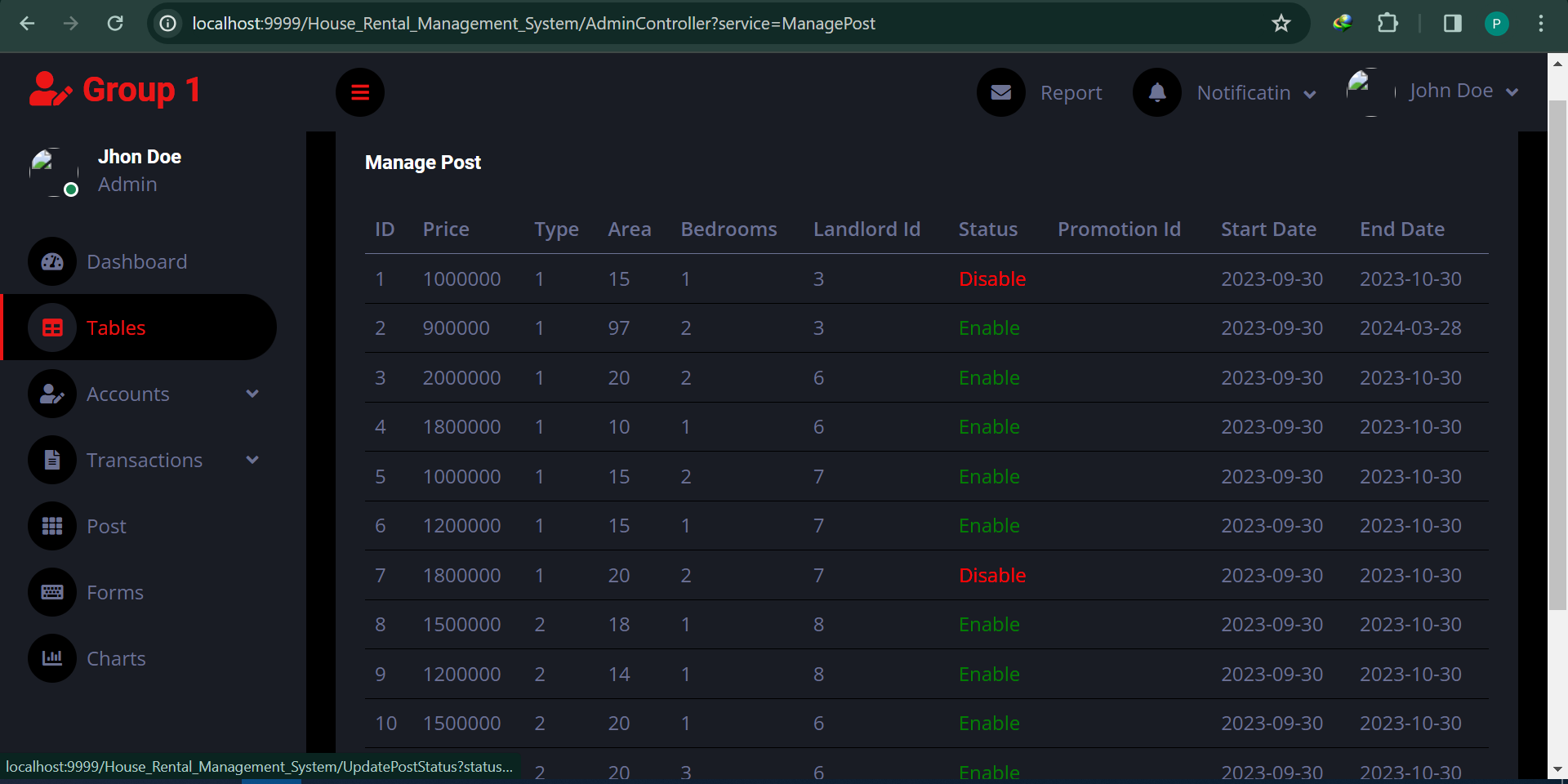
##### SQL Commands

Select\*from report

insert Report values ()

### 8.3. Manage post

##### UI Design

****

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Status | Hyperlink | Click this to update post’s status to Enable or Disable |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | U | Update status of post |
| Order | U | Reject all order related to post that have been disabled |

##### SQL Commands

1.Update status of post

UPDATE [dbo].[Post] SET [status] = ? WHERE id = ?

2.Reject all order related to post that have been disabled

update Orders set status = 2 where post\_id = ?

## 9. Review

##### UI Design

****

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| View feedback | Button | Show feedback about the property or tenant |
| Send feedback | Button | Send feedback about the property or tenant |
| Rate property | Button | Rate from scale 1 to 5 about the property |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Comment | CRUD | User can create or edit, delete their comment |
| Rate | CRUD | Tenant can see all rate, create or edit, and delete their rate |

##### 

##### SQL Commands

Select\*from review

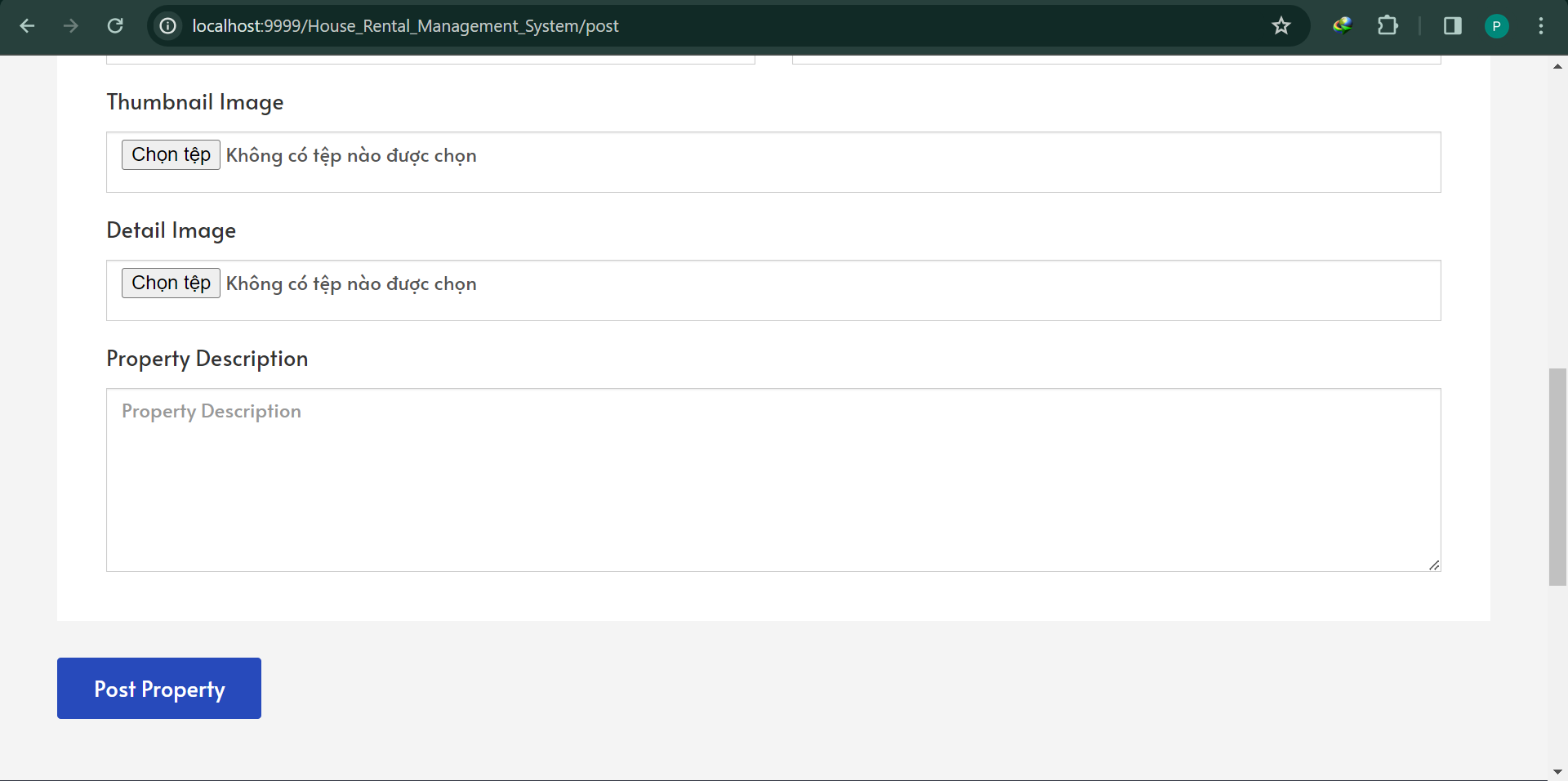
select\*from rate

insert feedback values ()

insert rate values ()

## 10. Payment feature

### 10.1. Pay for post



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Post Property | button | After click this button, the form will be send to Admin and system process the payment by minus the point in landlord’s account |

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Landlord | U | Minus point in landlord’s account |
| Transactions | C | Insert data into transactions table |

##### 

##### SQL Commands

1.Minus point in landlord’s account

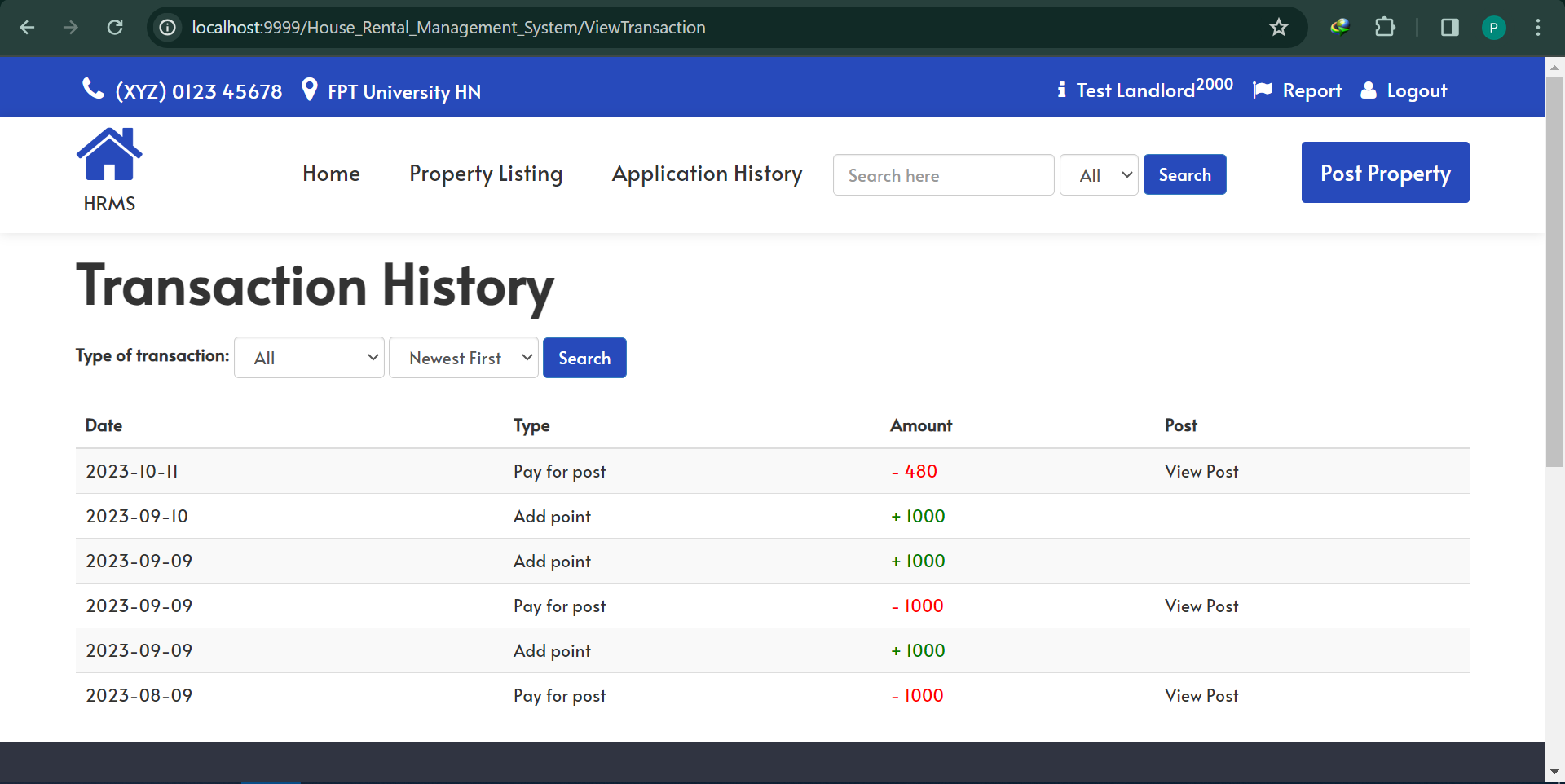
update [Landlord] set account\_points = account\_points - ? where [id] = ?

2.Insert data into transactions table

insert into [Transactions] (amount, payer\_id, type,transaction\_date, post\_id, receiver\_id)

values (?,?,1, getdate(), ?, null)

### 10.2. View transaction history



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Type of transaction | Combo Box Single-Choice | Value: All, Add point, Pay for post  Allow user to choose type of transaction |
| Sort type (date) | Combo Box Single-Choice | Value: News first, Oldest first  Allow user to choose type to sort |
| View post | Hyperlink | Click this to redirect to post has been paid |
| Search | Button | Click this to search |

##### 

Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Landlord | R | Get information about payer |
| Transaction | R | Get information about transactions |
| Post | R | Get information about posts have been paid |

##### 

##### SQL Commands

1.Get information about payer

select \* from Landlord where id = ?

2.Get information about transactions

select \* from Transactions where payer\_id = ?

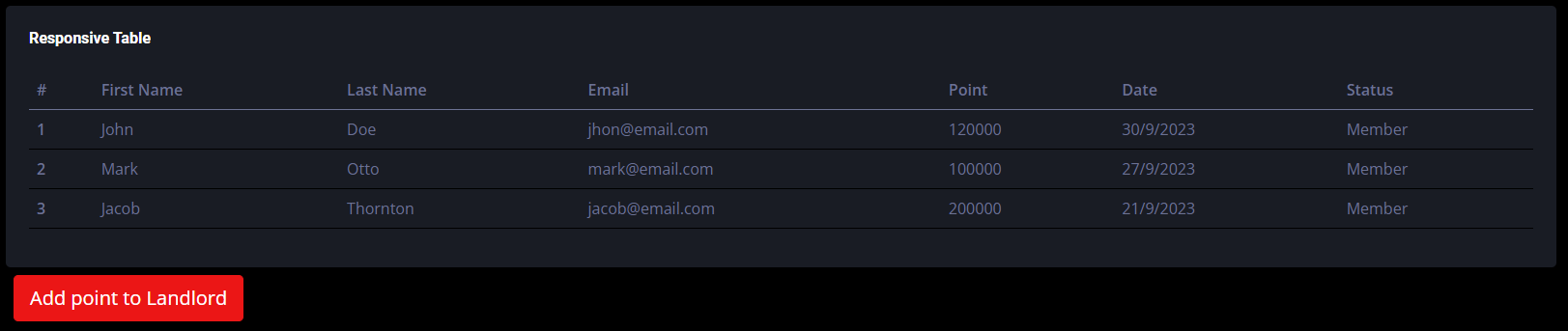
3.Get information about posts have been paid

select \* from Post where id = ?

### 

### 10.3. View system transaction

##### UI Design



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Landlord | R | Get information about payer’s id. |
| Transaction | R | Get information about transactions. |
| Post | R | Get information about posts that have been paid. |

##### 

##### SQL Commands

SELECT \* FROM [Transactions] tr

join [Landlord] ll ON tr.payer\_id = ll.id

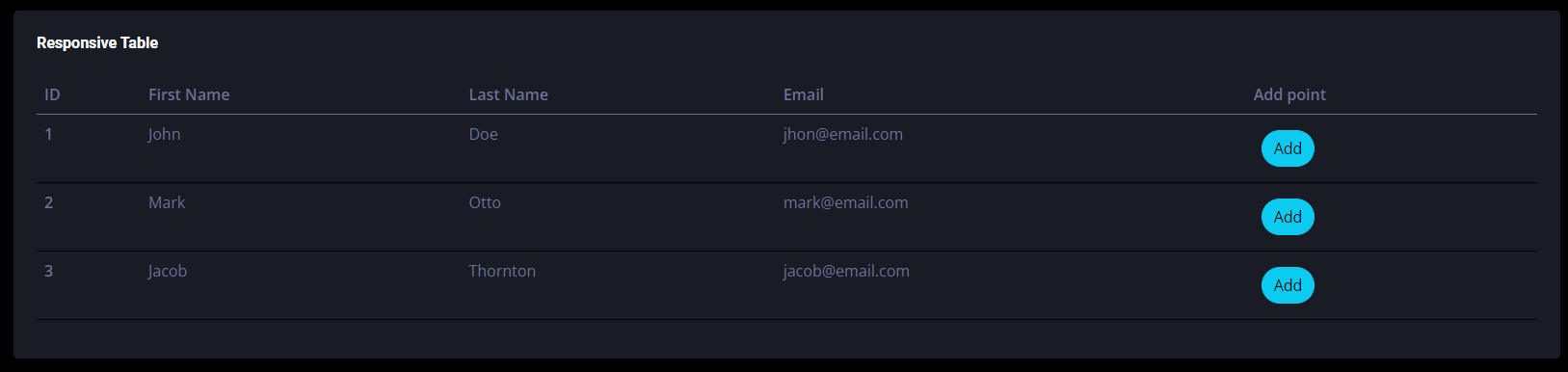
join [Post] p ON p.id = tr.post\_id

WHERE type = ?

ORDER BY ?

### 10.4. Add point to Landlord

##### UI Design



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Landlord | R | Get information about landlords. |
| Landlord | U | Update Landlord’s point. |
| Transactions | C | Create a new transaction. |

##### 

##### SQL Commands

1. Get information about landlords.

SELECT [id],[first\_name],[last\_name]

FROM [Landlord]

1. Update Landlord’s point.

UPDATE [Landlord] SET [accout\_points] = ?

1. Create a new transaction.

INSERT INTO [dbo].[Transactions]

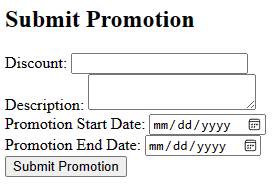
VALUES ?

## 

## 11. Promotion feature

#### a. Create a promotion

##### UI Setting



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Discount | Text box | Price reductions or deductions applied to properties |
| Description | Text box | Information about promotion of |
| Promotion Start Date | Date | Time to start promotion |
| Promotion End Date | Date | Time to close promotion |
| Submit Promotion | HyperLink | Actions that can be submit to save information |

##### 

##### Database Access

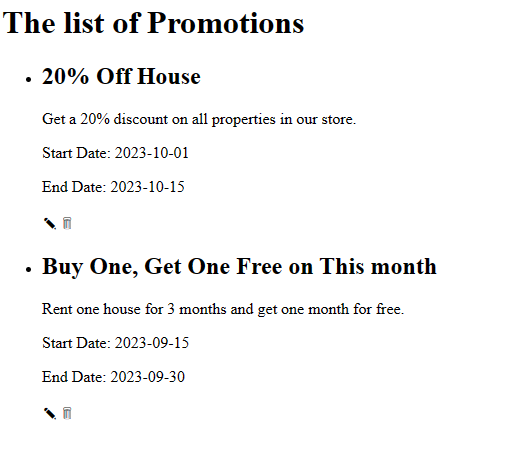
|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Promotions | RU | Get promotion information and update promotion |

##### SQL Commands

1. SELECT \* FROM Promotions

#### b. Edit a promotion

##### UI Setting



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Edit button | Button | Clicks to edit the promotion |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Promotions | RU | Get promotion information and update promotion |

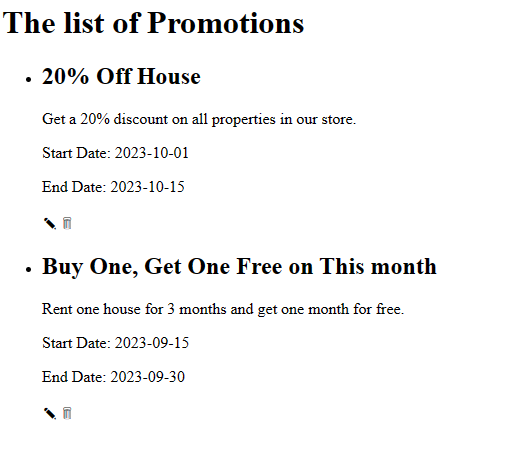
##### SQL Commands

1. SELECT \* FROM Promotions

2. UPDATE Promotions SET discount = 'New Discount Value', description = 'New Description', promotion\_start\_date = 'New Start Date', promotion\_end\_date = 'New End Date' WHERE promotion\_id =? ;

#### c. Remove a promotion

##### UI Setting



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Remove button | Button | Clicks to remove the promotion |

##### 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Post | R | Get property information |
| Promotions | RU | Get promotion information and update promotion |

##### SQL Commands

1. SELECT \* FROM Promotions

2. UPDATE Promotions SET discount = 'New Discount Value', description = 'New Description', promotion\_start\_date = 'New Start Date', promotion\_end\_date = 'New End Date' WHERE promotion\_id =? ;

# IV. Appendix

## 1. Assumptions & Dependencies

AS-1: Users are assumed to have reliable internet connectivity to access and use the system.

AS-2: It is assumed that the data provided by users is accurate and truthful. The system relies on accurate information for decision-making.

AS-3: The system assumes compliance with local and national laws and regulations related to rental agreements, data privacy, and fair housing.

AS-4: Landlords are assumed to have legal ownership or authorization to list properties on the platform.

AS-5: User authentication and access control mechanisms are assumed to be effective in safeguarding user accounts and data.

DE-1: The implementation of security protocols and encryption mechanisms is crucial to protect user data.

DE-2: Legal advice or consultation may be necessary to ensure the system complies with local and national rental laws and regulations.

DE-3: A responsive customer support team is essential to address user inquiries and issues promptly.

DE-4: The accuracy and availability of property listings depend on landlords actively managing their listings.

## 2. Limitations & Exclusions

LI-1: The system's property listings and services may be limited to specific geographic regions, cities and may not cover all areas.

LI-2: After a user tries to log in the system fails 3 or more times, that device can be locked in some time to try to log in one more time.

LI-3: While the system strives for accuracy, it may rely on user-submitted data which may not always be accurate or up-to-date.

LI-4: The system may not verify the physical condition or existence of listed properties, and users should exercise due diligence before renting.

LI-5: The system does not provide legal advice. Users should seek legal counsel for specific rental agreements or dispute-related issues.

LI-6: Users are responsible for keeping their account credentials secure and ensuring the accuracy of information they provide.

EX-1: The system is exclusively designed for rental property management and does not facilitate real estate sales transactions.

EX-2: The system may focus on residential properties and exclude commercial real estate.

EX-3: While the system may facilitate communication, it does not provide formal arbitration services for tenant-landlord disputes.

EX-4: The system may not provide information about environmental factors that may affect a property, such as pollution or flood risk.

## 3. Business Rules

BR-S1: The system should have high availability and minimal downtime, with scheduled maintenance communicated to users in advance.

BR-S2: The system and all user activities must comply with local laws and regulations related to rental properties, fair housing, and data protection.

BR-S3: Disputes between tenants and landlords should be resolved through a structured process, which may include mediation or arbitration if necessary.