

**Requirement & Design Specification**

**BOOK READING SYSTEM (BRS)**

**Version: 1.0**

**Author:**

**Nguyễn Quang Sự**

**MSSV:HE171345**

**Đoàn Thế Tuấn  
MSSV:HE173140**

Contents

[Record of Changes 2](#_gjdgxs)

[I. Overview 4](#_30j0zll)

[1. User Requirements 4](#_1fob9te)

[1.1 Actors 4](#_3znysh7)

[1.2 Use Cases 4](#_2et92p0)

[2. Overall Functionalities 5](#_tyjcwt)

[2.1 Screens Flow 5](#_3dy6vkm)

[2.2 Screen Descriptions 5](#_1t3h5sf)

[2.3 Screen Authorization 5](#_4d34og8)

[2.4 Non-UI Functions 6](#_2s8eyo1)

[3. System High Level Design 6](#_17dp8vu)

[3.1 Database Design 6](#_3rdcrjn)

[3.2 Code Packages 7](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.35nkun2)

[II. Requirement Specifications 8](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.1ksv4uv)

[1. <<Feature Name>> 8](#_44sinio)

[1.1 <<UseCaseCode\_UC Name>> 8](#_2jxsxqh)

[2. Common Functions 11](#_z337ya)

[2.1 UC-2\_Login System 11](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.3j2qqm3)

[3. Patron Feature 12](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.1y810tw)

[3.1 UC-5\_Order a Meal 12](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.4i7ojhp)

[3.2 UC-6\_Register for Payroll Deduction 13](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.2xcytpi)

[III. Design Specifications 14](#_1ci93xb)

[1. <<Feature Name>> 14](#_3whwml4)

[1.1 <<SubFeature Name>> 14](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.2bn6wsx)

[1.2 System Access 15](https://docs.google.com/document/d/15n6ifVM7e8FYEOOcAjSOMvkNHeyY9MUi/edit#heading=h.qsh70q)

[IV. Appendix 19](#_3as4poj)

[1. Assumptions & Dependencies 19](#_1pxezwc)

[2. Limitations & Exclusions 19](#_49x2ik5)

[3. Business Rules 19](#_2p2csry)

[4. .. 19](#_147n2zr)

# I. Overview

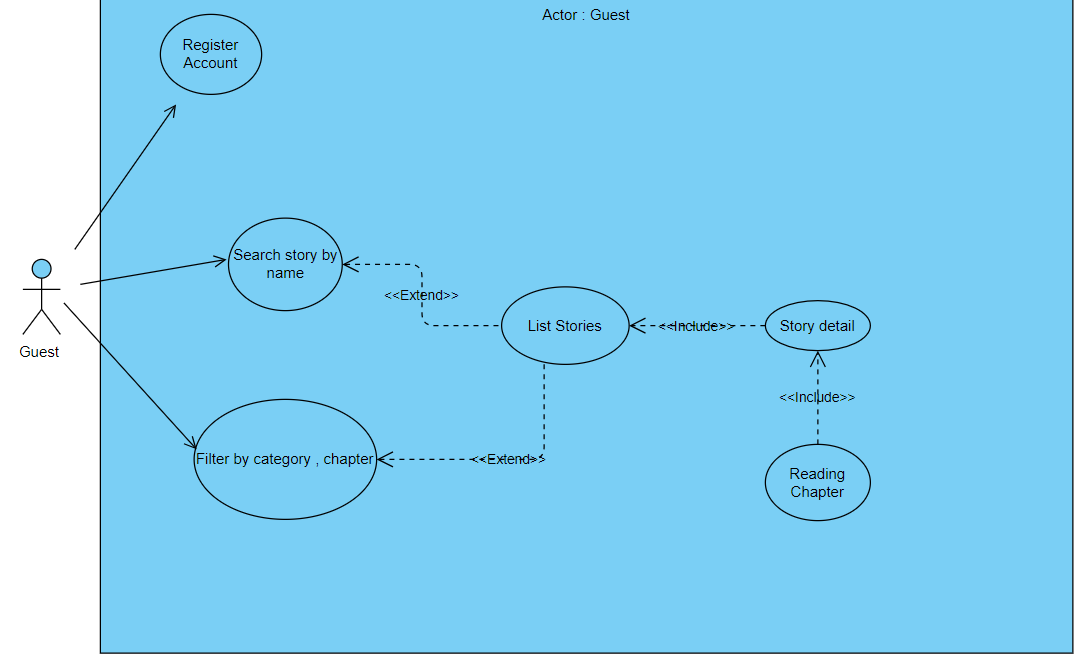
## 1. User Requirements

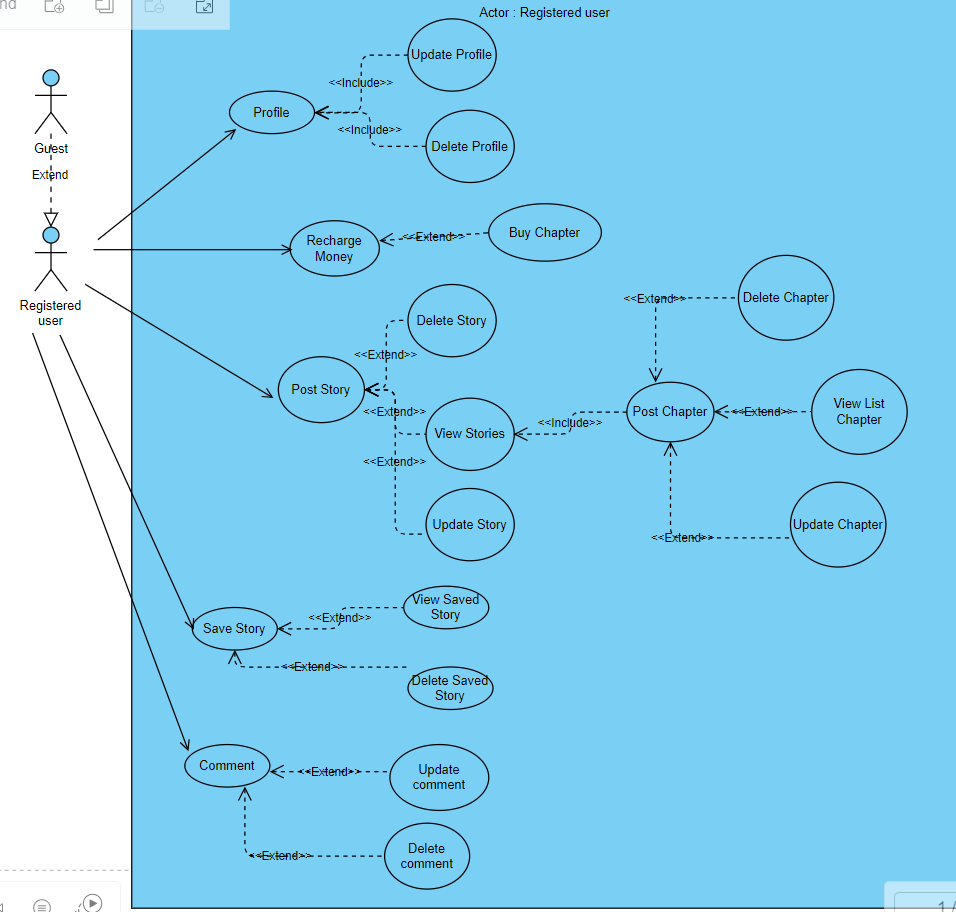
### 1.1 Actors

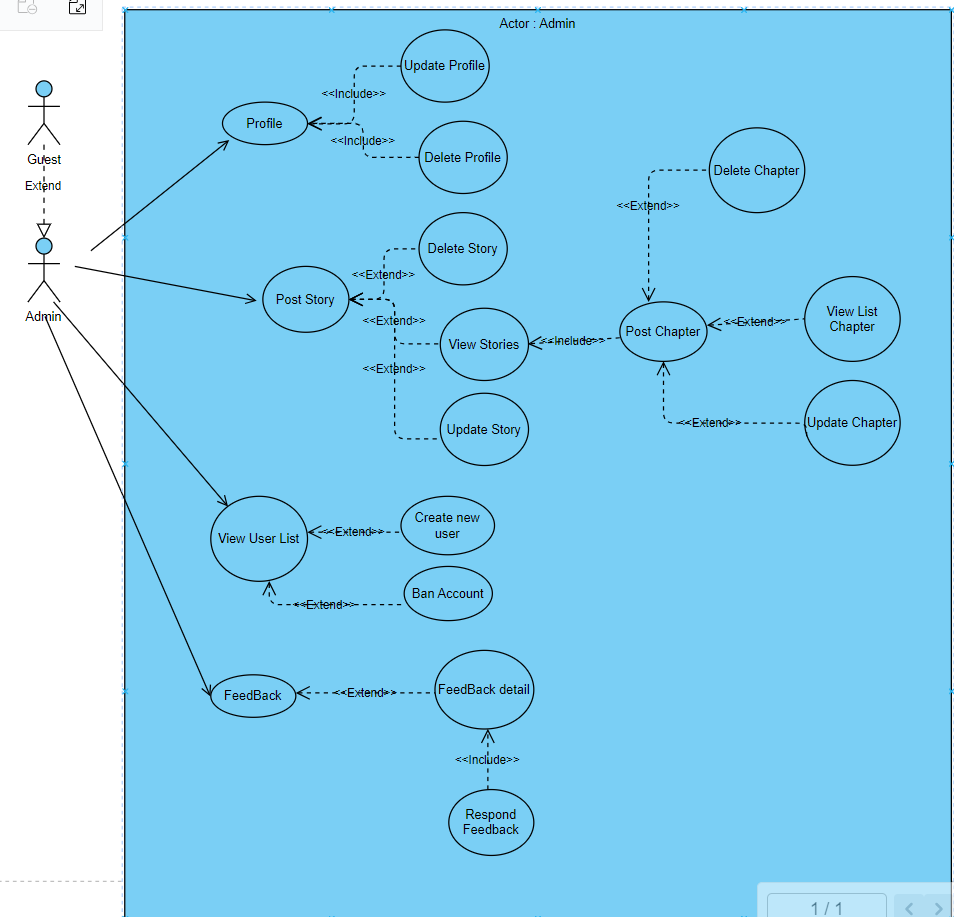
| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Admin | Creator Website , Manage account and view all stories |
| 2 | Registered user | Who have account can read and comment all stories |
| 3 | Guest | Who not have account but still can access the page and read stories |
|  |  |  |

### 1.2 Use Cases

#### **a. Diagram(s)**







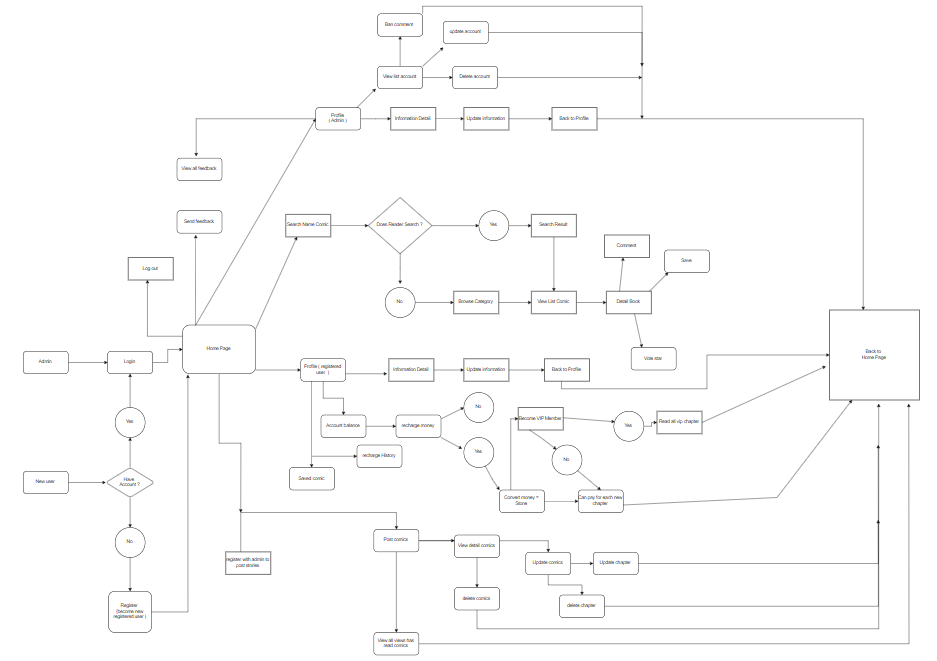
#### **b. Description(s)**

| **ID** | **Feature** | **Use Case** | **Use Case Description** |
| --- | --- | --- | --- |
| 01 | Register | Register with account, password, email (not verified) | Guests or those who do not have an account will register to have an account |
| 03 | Contact | Respond to problems with admin (Form) | If there is a problem, visitors can contact and report to the admin |
| 04 | search | search stories | Search for stories on the search bar |
| 05 | View category | See categories of stories | See the story genres available on the filter bar |
| 06 | stories list | All stories list | Visitors can see the entire list of stories on the Home page |
| 07 | Search filter | Get stories by filter (get by tag , author , ) | Reader can get stories by choose each filter |
| 08 | Read story details | Read story details | Read the story in detail |
| 09 | Log in | Log in with your registered account and password | The user will log in with the existing password account |
| 14 | Log out | Log out | Sign out of your account from the device |
| 15 | Post stories | Post stories | Registered users can request admin permission to post stories |
| 16 | Edit stories | Edit stories | Story posters can update and edit posted stories |
| 17 | Add chapter | Add chapters to published stories | Story publishers can add story chapters |
| 19 | Delete story | Delete story chapter | Story publishers can delete story chapters |
| 20 | save story | save story | registered user can save stories |
| 21 | Story saved | Story saved | Show the list of saved stories |
| 22 | view profile | View personal profile | Displays personal registered user information |
| 23 | Update Avatar | Update Avatar | Change the registered user avatar |
| 24 | change Password | change Password | Change registered user account password |
| 25 | Update profile | Update personal profile (name, gender, phone number, address) | Edit personal information |
| 33 | Reading history | View reading history | View story reading history |
| 34 | Rate the story | Rate the story | Vote for the following story |
| 35 | story report | story report | Readers can report the story if they find the story has misleading content |
| 40 | View readings | View readings (statistics, etc.) | registered user can see the number of people who have visited and read the story |
| 42 | View account list | View account list | Admin can view the list of users |
| 43 | Ban account | Ban account | Ban accounts when there is negative or false information or comments |
| 44 | View reports | View reports (From user) | Admin can view user-submitted report forms |
| 45 | Report response | Report response(From user) | Admin can reply and send back to the sender |
| 46 | List of story reports | See list of story reports | Admin can view the list of user-submitted report forms |
| 47 | Manage web visits | Manage web visits (statistics,...) | Admin can view web visits |

## 2. Overall Functionalities

### 

### 2.1 Screens Flow



### 2.2 Screen Descriptions

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | ViewHome | Homepage | View Comics list ,category |
| 2 | View list stories | Stories list | list of Stories |
| 3 | Stories’s detail | Stories’s detail | View detail of Stories |
| 4 | Filter | Filter by category | View Stories by category |
| 5 | Register | Register | Register an account |
| 6 | Login | Login | Login to system |
| 7 | Profile | registered user profile | View registered user’s profile |
| 8 | Change password | Change password | change Password |
| 9 | Update Profile | Update Profile | Update Profile( email, name , phone ,...) |
| 10 | Send Report | Send Report | Send Report for a story |
| 11 | Post stories | Post stories | The author can post Stories |
| 12 | Update Stories | Update Stories | The writer can Update Stories |
| 13 | View Account | View Account | Admin can view all profile registered user and their account |
| 14 | Manage stories | View all stories | Admin can view all stories |
| 15 | View report | View all report | Admin can view all reader-submitted responses |

### 

### 

### 2.3 Screen Authorization

| **#** | **Screen** | **Guest** | **Registered User** | **Admin** |
| --- | --- | --- | --- | --- |
|  | ViewHome | X | x | x |
|  | View list stories | x | x | x |
|  | Stories’s detail | x | x | x |
|  | Filter by category | x | x | x |
|  | Register | x | x |  |
|  | Login | x | x | x |
|  | Profile |  | x | x |
|  | Change password |  | x | x |
|  | Update Profile |  | x | x |
|  | Send Report |  | x |  |
|  | Post stories |  | x | x |
|  | Update Stories |  | x | x |
|  | View Account |  |  | x |
|  | Recharge money |  | x |  |
|  | Manage stories |  |  | x |
|  | View report |  |  | x |
|  |  |  |  |  |

### 2.4 Non-UI Functions

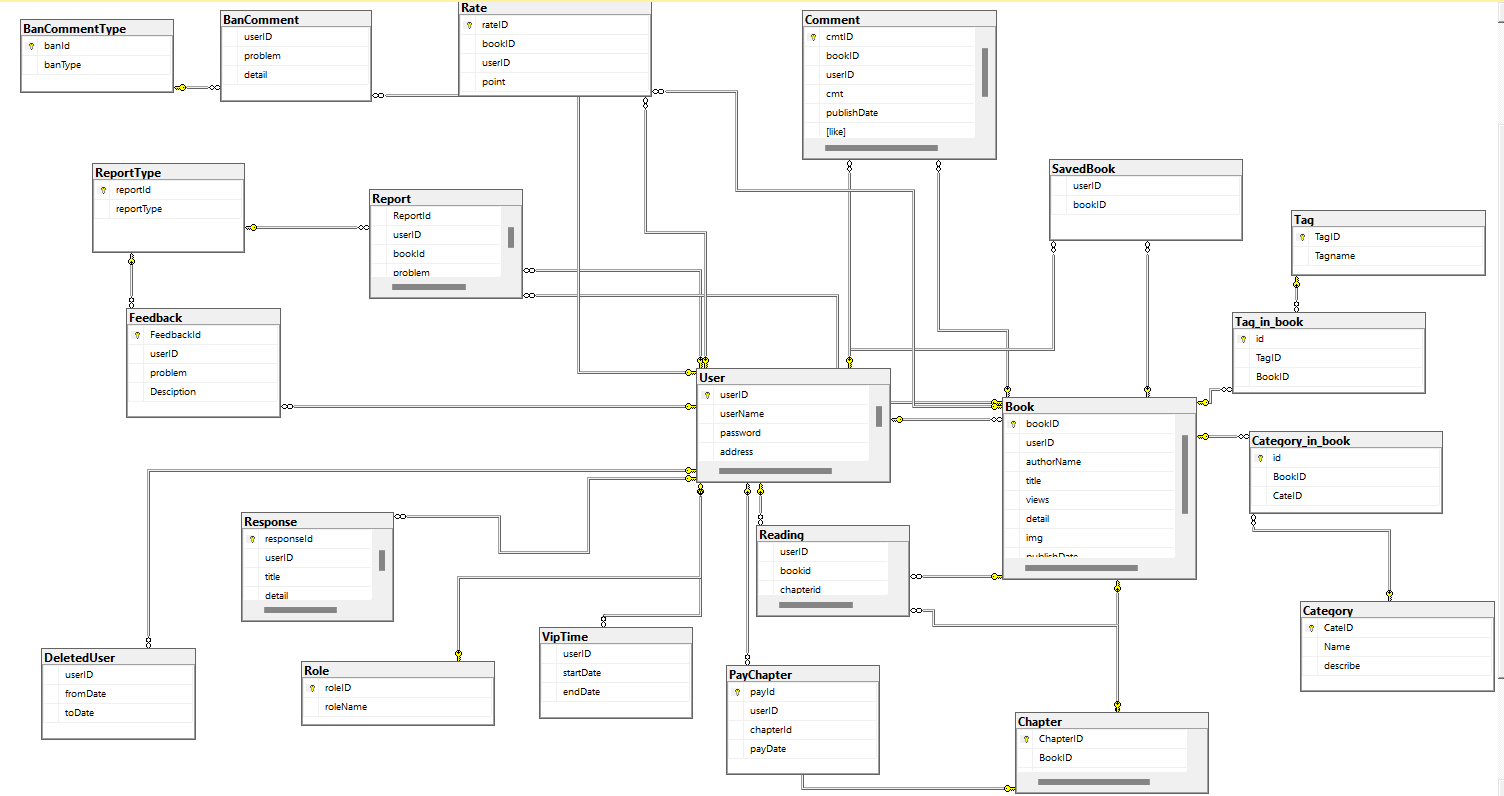
## 

|  | Features | System Function | Description |
| --- | --- | --- | --- |
| 1 | Common | Count view | Count the number of view |
|  |  |  |  |

## 

## 3. System High Level Design

### 3.1 Database Design



# II. Requirement Specifications

## 1. Public Function

### 1.1 PB-1\_Register System

#### *a. Functionalities*

**Functional Description Template**

| UC ID and Name: | **PB-1\_Register System** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: |  |
| Primary Actor: | Guest | Secondary Actors: | User |
| Trigger: | Guest click on Login button from the page header,then click on “Sign in” link.Or Guest accesses an authenticated feature (from a link or type the page URL directly into the address bar) | | |
| Description: | As an guest ,i want to be able to sign in the system so that i can interact with the book or feedback to admin | | |
| Preconditions: | Guest’s username must not exist in database | | |
| Postconditions: | Guest sign up successfully and auto login with that account  Guest’s information save to database | | |
| Normal Flow: | Guest access the Register screen  Guest type username,password then click on sign in button  System validate the username (see E1.1.1,E1.1.2)  System save guest’s information to database  System login with username and password  Return to the Homepage | | |
| Alternative Flows: |  | | |
| Exceptions: | E1.1.1 Guest’s username exist in database  1.Error Message show in screen  2.Guest cancel Register=>Go to homepage  3.Guest click Login => Go to login page  E.1.1.2 Guest’s password invalid  1.Error Message show in screen | | |
| Priority: | High ,Should Have | | |
| Frequency of Use: |  | | |
| Business Rules: | PB1\_BR | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| PB1\_BR | Invalid Register | Guest can’t sign in system if below case:   * His/her username is exist in database * His/her password is less than 6 character or greater than 18 |

### 1.2 PB-2\_View homepage

#### *a. Functionalities*

**Functional Description Template**

| UC ID and Name: | PB-2\_View homepage | | |
| --- | --- | --- | --- |
| Created By: | Su | Date Created: | 15/01/2024 |
| Primary Actor: | admin/registered user/guest | Secondary Actors: |  |
| Trigger: | User accesses the website. | | |
| Description: | This use case describes the process of viewing the homepage of the website. | | |
| Preconditions: | 1. The website is accessible. 2. The user is logged in (for registered users). 3. The user has appropriate permissions (for admins). 4. The Guest | | |
| Postconditions: | The homepage is displayed to the user. | | |
| Normal Flow: | 1. User opens the website. 2. System displays the homepage. | | |
| Alternative Flows: | If the user is not logged in and chooses to continue as a guest, a simplified version of the homepage may be displayed. | | |
| Exceptions: | If there is a system error, display an error message.  If the user authentication fails, prompt the user to re-enter credentials. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | PB\_2 | | |
| Other Information: | The homepage layout may be subject to periodic updates. | | |
| Assumptions: | 1. Users have a stable internet connection. 2. The application or website is operational during the user's access. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| PB\_2 | The homepage should be user-friendly and include relevant information and features. | The design and content of the homepage must prioritize user-friendliness, ensuring that it provides a positive and intuitive experience for all users. |

### 1.3 PB-3\_View category

#### *a. Functionalities*

**Functional Description Template**

| UC ID and Name: | PB-3\_View category | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: |  |
| Primary Actor: | Guest,User, Admin | Secondary Actors: | User |
| Trigger: | Guest click on Thể Loại button from the page header | | |
| Description: | As a guest,User,Admin I want to be able to see the types of stories I want | | |
| Preconditions: | Category must exist in database | | |
| Postconditions: | System show category in screen | | |
| Normal Flow: | Guest/User/Admin view homepage  Guest/User/Admin choose “Thể Loại”that want to see  System show category from database | | |
| Alternative Flows: |  | | |
| Exceptions: |  | | |
| Priority: | High ,Should Have | | |
| Frequency of Use: | Frequently used, especially by users exploring new content. | | |
| Business Rules: | PB3\_BR | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| PB3\_BR |  | Categories must be unique and should follow a standardized naming convention. |

### 1.5 PB-5\_Filter

#### *a. Functionalities*

**Functional Description Template**

| UC ID and Name: | PB-5\_Filter | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: |  |
| Primary Actor: | Guest,User, Admin | Secondary Actors: | User |
| Trigger: | Guest/User/Admin click on book them click on the writer's name in“tác gỉả”, category name in”Thể loại”or access by link address | | |
| Description: | As a Guest,User, Admin, I want to be able to search for stories by author and genre | | |
| Preconditions: | Author, category must exist in database | | |
| Postconditions: | Guest/User/Admin go to link address successfully  System show book filter in screen | | |
| Normal Flow: | Guest/User/Admin view homepage  System listed book in database according to each criterion  User choose book (See TU1.5.1,E1,5.1)  System show book’s detail from database  Guest/User click on author or category under image book(See TU1.5.2,E1,5.2)  System show list book from database | | |
| Alternative Flows: | TU1.5.1 Guest/User/Admin access by link address ,return step 4 in normal flow  TU1.5.2 Guest/User/Admin access by link address ,return step 6 in normal flow | | |
| Exceptions: | E1.5.1.Error list book show in screen | | |
| Priority: | medium,Should Have | | |
| Frequency of Use: |  | | |
| Business Rules: | PB5\_BR | | |
| Other Information: |  | | |
| Assumptions: | Users have a basic understanding of the authors and categories listed. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| PB5\_BR | Exactly | The book filter should prioritize displaying relevant and accurate results based on the selected author or category. |

### 1.6 PB-6\_**View book detail**

#### *a. Functionalities*

**Functional Description Template**

| UC ID and Name: | PB-6\_**View book detail** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: |  |
| Primary Actor: | Guest, user, admin | Secondary Actors: | User |
| Trigger: | Guest/User/admin click on book in home screen or access by link address | | |
| Description: | As an Guest/User/admin,i want to be able to see details about the book | | |
| Preconditions: | book detail must exist in database  Book is not banned | | |
| Postconditions: | Guest/User/admin go to link address successfully  System show book detail in screen | | |
| Normal Flow: | Guest/User/admin view homepage  System listed book in database according to each criterion  User choose book that want to read(See TU1.6.1,E1,6.1)  System show book’s detail from database | | |
| Alternative Flows: | TU1.6.1 Guest/User/admin access by link address ,return step 4 in normal flow | | |
| Exceptions: | E1.6.1.Error Message show in screen | | |
| Priority: | High ,must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | PB6\_BR | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| PB1\_BR | Exactly | The book's information should be clear and accurate |

### 1.7 PB-7\_Reading chapter

#### *a. Functionalities*

**Functional Description Template**

| UC ID and Name: | PB-7\_Reading chapter | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: |  |
| Primary Actor: | Guest,User | Secondary Actors: | User |
| Trigger: | Guest/User click on book them click on chapter or access by link address | | |
| Description: | As a Guest/User i want to able to reading chapter, so that i can .. | | |
| Preconditions: | Chapter must exist in database  Book is not banned | | |
| Postconditions: | Guest/User go to link address successfully  System show chapter in screen | | |
| Normal Flow: | Guest/User view homepage  System listed book in database according to each criterion  User choose book that want to read(See AF1.7.1,E1,7.1)  System show book’s detail from database  Guest/User click on chapter(See AF1.7.2,E.7.1)  System show chapter’s detail from database | | |
| Alternative Flows: | AF1.7.1 Guest/User access by link address ,return step 4 in normal flow  AF1.7.2 Guest/User access by link address ,return step 6 in normal flow | | |
| Exceptions: | E1.7.1 Book is vip-only  1.Show message on the screen  2.Guest/User cancel reading  3.System request user for login => go to login page  4.User login with common account => show message on the screen | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | PB7\_BR1,PB7\_BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| PB7\_BR1 | Access Book | Guest/User can’t read chapter if below case:   * Chapter vip * Book is banned * Account is banned |
| PB7\_BR2 | View | Chapter’s view is increase when Guest/User access to chapter and reset after 5p |

## 

## 2. Common Functions

## 2.1 CM-1\_Login System

#### *a. Functional Description*

| UC ID and Name: | **CM-1\_Login System** | | |
| --- | --- | --- | --- |
| Created By: | Su | Date Created: | 13/01/2024 |
| Primary Actor: | Customer | Secondary Actors: | None |
| Trigger: | User clicks Login button from the page header, or  User accesses an authenticated feature (from a link or type the page URL directly into the address bar) | | |
| Description: | As a user, I want to be able to log into the system so that I can use the system’s authenticated features and access my personalized account. | | |
| Preconditions: | User account has been created & authorized | | |
| Postconditions: | * User logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Flow | **2.0 Login System**  1. User accesses the User Login screen  2. User types in the login details or choo other login options (see 2.1 and 2.2)  3. User clicks the Login button  4. System validates the login details (see 2.0.E1)  5. System allows user to access  6. System tracks user’s success login to the Activity Log  7. System accesses the Home Page (or the previous calling page if any) | | |
| Alternative Flows: | ***2.1 Login with your registered account and password***  1. User chooses to login system using register  2. System redirects the user to the Login screen  3. User types in the register account details and chooses to login  4. Account validates user’s login information successfully and redirect him/her back to the system  5. Return to step 5 of normal flow.  ***2.2 Google Login***  1. User chooses to login system using Google account  2. System redirects the user to the Google’s Login screen  3. User types in the Google account details and chooses to login  4. Google validates user’s login information successfully and redirect him/her back to the system  5. Return to step 5 of normal flow. | | |
| Exceptions: | ***2.0 System can’t authenticate the user***  1. The Error Message screen is shown to the user  2. User cancels the logging in *Home Page*  3. User clicks “Forgot Password” *Reset Password*  4. User clicks “Register” link *=> Register User Account* | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1, FR2, FR3 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| FR1 | Password Encoding | User’s password must be encoded with MD5 hashing |
| FR2 | Invalid Logging In | User can’t be authenticated to login the system if below cases   * His/her logging-in details are incorrect * His/her account has not been verified * His/her account has been locked or blocked |
| FR3 | Account Locking | If user inputs wrong logging-in details 6 times continuously, his/her account would be locked in 30 minutes |

### 

### 2.2 CM-2\_Log out

#### *a. Functional Description*

| UC ID and Name: | **CM4\_Logout System** | | |
| --- | --- | --- | --- |
| Created By: | Quan | Date Created: | 17/01/2024 |
| Primary Actor: | Registered user , Admin | Secondary Actors: | None |
| Trigger: | User clicks Login button from the page header | | |
| Description: | As a user, when I logged into the system I can log out from the system and bring it to the original screen page. | | |
| Preconditions: | User account is being logged into the system. | | |
| Postconditions: | none | | |
| Normal Flow | **2.4 Logout System**  1.When users are in the system, the user chooses to log out the current account  2. User clicks the Logout button | | |
| Alternative Flows: | none | | |
| Exceptions: | none | | |
| Priority: | Must Have | | |
| Frequency of Use: | When you want to change from one account to another to log in | | |
| Business Rules: | FR1, FR2, FR3 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| FR1 | User Authentication | Users must be authenticated before accessing any resource. |
| FR2 | Logout Process | Users can only log out if they are currently logged in. |
| FR3 | Password Complexity | Passwords must meet certain complexity requirements. |

### 

### 2.3 CM-3\_View Profile

#### *a. Functional Description*

| UC ID and Name: | **CM-5\_Login System** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 16/Jun/2023 |
| Primary Actor: | User | Secondary Actors: | None |
| Trigger: | User clicks Profile button from the page header, or  User accesses an authenticated feature (from a link or type the page URL directly into the address bar) | | |
| Description: | As a user, I want to be able to view profile so that I can manage my account. | | |
| Preconditions: | User has been login to system | | |
| Postconditions: | * System show user’s information in screen | | |
| Normal Flow | 1. User login to system (See 2.1)  2. User click on view profile (See AF2.5.1)  4. System show user’s information on screen | | |
| Alternative Flows: | ***AF2.5.1 User access by address link*** | | |
| Exceptions: | ***2.5.E1 View profile without login***  1.System request to login =>go to login page  2.User cancel login => go to homepage | | |
| Priority: | Must Have | | |
| Frequency of Use: | None | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

-None

## 

### 2.4 CM-4\_Search Book

#### *a. Functional Description*

| UC ID and Name: | **CM-6 Search** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 18/01/2024 |
| Primary Actor: | Admin/Registered user  Guest | Secondary Actors: | None |
| Trigger: | Requests to search for a book. | | |
| Description: | Search for a book on the system with the name book. | | |
| Preconditions: | 1. The user must type the book name 2. The user must remember the name or letters of the book's title | | |
| Postconditions: | Displays all book with the same name or list names with similar words | | |
| Normal Flow | 1. The user entered the story name into the search bar. 2. The system redirects to the search page. 3. The system displays all books with the same name or characters corresponding to the word the user enters. | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Priority: | Medium | | |
| Frequency of Use: | 3000 user , 10 times per users, once per week | | |
| Business Rules: | 2.6\_HU1 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |  |
| --- | --- | --- | --- |
| 2.6\_HU1 | 1. The system can search for books by name | | |

### 2.5 CM-5\_Change Password

#### *a. Functional Description*

| UC ID and Name: | **CM-7\_Change Password** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 19/01/2024 |
| Primary Actor: | Admin/Registered user | Secondary Actors: | None |
| Trigger: | Registered User requests to change their password. | | |
| Description: | changing a password for an admin or registered user on the system. | | |
| Preconditions: | 1. The registered user must be logged into the system. 2. The registered user must know their current password. | | |
| Postconditions: | The registered user's password is successfully changed. | | |
| Normal Flow | 1. Registered User navigates to the change password section. 2. Registered User selects the option to change the password. 3. System prompts the registered user to enter their current password. 4. Registered User enters the current password. 5. System validates the current password. 6. System prompts the Registered user to enter a new password. 7. Registered User enters the new password. 8. System updates the registered user's password with the new one. | | |
| Alternative Flows: | None | | |
| Exceptions: | 1. If the entered current password is incorrect, the system notifies the user 2. If the new password does not meet the system's password policy requirements, the system notifies the user and prompts them to enter a valid password. | | |
| Priority: | Medium | | |
| Frequency of Use: | Registered Users may change their passwords periodically for security reasons, but it is not an everyday occurrence. | | |
| Business Rules: | 2.6\_HU1 | | |
| Other Information: | None | | |
| Assumptions: | 1. The system has a password change functionality. 2. The system enforces password policies for security purposes. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.6\_HU1 | The system has a password change functionality. | The system must provide a feature that allows users, both admins, and registered users, to change their passwords. |

### 2.6 CM-6\_Update Profile

#### *a. Functional Description*

| UC ID and Name: | **CM-8\_Update Profile** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Admin/Registered user | Secondary Actors: | None |
| Trigger: | User click button update profile | | |
| Description: | This use case describes the process of updating a user's profile information within the system. | | |
| Preconditions: | The user must be authenticated and logged into the system. | | |
| Postconditions: | The user's profile information is successfully updated in the system. | | |
| Normal Flow | 1. User accesses the profile. 2. User selects the option to edit profile information. 3. User modifies the desired fields (e.g., name, email, password). 4. User saves the changes. 5. System verifies the changes and updates the user's profile information. 6. System confirms successful update to the user. | | |
| Alternative Flows: | If the user cancels the edit, they are returned to the profile view without saving changes. | | |
| Exceptions: | 1. If there are errors in the input data (e.g., invalid email format, password too short), system prompts the user to correct them. 2. If the system encounters technical issues during the update process, it displays an error message and prompts the user to try again later. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | 2.8\_CM8\_1 | | |
| Other Information: | N/A | | |
| Assumptions: | 1. Users are familiar with the process of updating profile information. 2. The system is stable and accessible during the update process. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.8\_CM8\_1 | Users can only update their own profiles. | This rule stipulates that users are only permitted to update their own profile information within the system. They cannot modify profiles belonging to other users. |

### 

### 2.7 CM-7\_Story Report

#### *a. Functional Description*

| UC ID and Name: | **CM-9\_Story Report** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click on Report button on book’s detail screen  Fill all blank field  Click on submit button | | |
| Description: | As a user, i want to report a book’s problem to admin | | |
| Preconditions: | User must logged in | | |
| Postconditions: | Report save to database successful  Admin can view the user’s report | | |
| Normal Flow | User click on the book that want to report on homepage  System load Book’s detail on database(See CM9\_EX1)  User click on report button  User fill all blank field then click on submit button (See CM9\_EX2)  System save user’s report in database | | |
| Alternative Flows: | none | | |
| Exceptions: | CM9\_EX1 : User report while not logged in: Return to log in page  CM9\_EX2 : User do not fill field : Request user to fill all blank field | | |
| Priority: | Should have | | |
| Frequency of Use: |  | | |
| Business Rules: | 2.9\_BR1, 2.9\_BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.9\_BR1 | Authoriz.. | Only User can report the book |
| 2.9\_BR2 |  | Each user can only report a book one time |

### 2.8 CM-8\_Reading Story

#### *a. Functional Description*

| UC ID and Name: | **CM-8\_Reading Story** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click on Dashboard button on menu  View the reading history on User’s detail screen | | |
| Description: | As a user, i want to view my reading history | | |
| Preconditions: | User must logged in | | |
| Postconditions: | User can view their reading history from database | | |
| Normal Flow | User click on Dashboard button on menu  System load User’s detail and reading history on database(See CM10\_EX1) | | |
| Alternative Flows: | none | | |
| Exceptions: | CM9\_EX1 : User access screen while not logged in: Return to log in page | | |
| Priority: | Should have | | |
| Frequency of Use: |  | | |
| Business Rules: | 2.9\_BR1 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.9\_BR1 | Add reading history | After user view a book’s chapter,this book will save to table reading in database |

### 2.9 CM-9\_View posted stories

#### *a. Functional Description*

| UC ID and Name: | **CM-9\_View posted stories** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click view my posted books | | |
| Description: | This use case describes the process by which a registered user who has posted stories can view the list of books they have posted. | | |
| Preconditions: | 1. The user must be authenticated and logged into the system. 2. The user must have previously posted stories on the platform. | | |
| Postconditions: | The user successfully views the list of stories they have posted. | | |
| Normal Flow | 1. User navigates to the profile section of the application. 2. User clicks on the "View My Posted Stories" button. 3. System retrieves and displays the list of stories posted by the user. 4. User reviews the list of posted stories. | | |
| Alternative Flows: | If the user has not posted any stories, the system displays a message indicating that there are no posted stories to view. | | |
| Exceptions: | 1. If there are technical issues retrieving the posted stories data, the system displays an error message and prompts the user to try again later. 2. If the user's session expires or they are logged out, they are prompted to log in again before viewing their posted stories. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | 2.12\_CM12\_1 | | |
| Other Information: | N/A | | |
| Assumptions: | 1. Users are familiar with the process of navigating the application and accessing their profile/dashboard. 2. The system is stable and accessible during the process of retrieving and displaying posted stories. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.12\_CM12\_1 | The list of posted stories should be accurate and up-to-date. | This rule ensures that the list of posted stories displayed to the user is accurate and reflects the most recent updates. Any changes made to the user's posted stories should be immediately reflected in the displayed list to provide an up-to-date view of their posted stories. |
|  |  |  |

### 

### 2.10 CM-10\_Post story

#### *a. Functional Description*

| UC ID and Name: | **CM-10\_Post story** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click post story | | |
| Description: | This use case describes the process by which a registered user can post their story. | | |
| Preconditions: | 1. The user must be authenticated and logged into the system. 2. The user has a story ready to be posted. | | |
| Postconditions: | The user's story is successfully posted on the platform. | | |
| Normal Flow | 1. User navigates to the section of the application dedicated to posting stories. 2. User clicks on the "Post Story" button. 3. System presents a form or interface for the user to input their story details, including title, content, genre, etc. 4. User fills out the required fields with the relevant information. 5. User submits the story for posting. 6. System verifies the input data and ensures all required fields are filled. 7. System saves the posted story to the platform's database. 8. System confirms successful posting of the story to the user. | | |
| Alternative Flows: | If the user decides not to post the story after clicking "Post Story," they can navigate away from the posting interface without submitting. | | |
| Exceptions: | 1. If there are errors in the input data (e.g., missing required fields, invalid content format), the system prompts the user to correct them before proceeding. 2. If the system encounters technical issues during the posting process, it displays an error message and prompts the user to try again later. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | 2.13\_CM13\_1 , 2.13\_CM13\_2 | | |
| Other Information: | N/A | | |
| Assumptions: | 1. Users understand how to navigate the posting interface and provide the necessary information for their story. 2. The system is stable and accessible during the posting process. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.13\_CM13\_1 | Users can only post stories if they are registered and logged into the system. | This rule specifies that only registered users who are logged into the system can post stories. Unauthorized users or users who are not logged in cannot access the functionality to post stories. |
| 2.13\_CM13\_2 | Users should have the ability to edit or delete their posted stories after posting them. | This rule stipulates that users who have posted stories on the platform should have the ability to edit or delete their stories after they have been posted. This allows users to make changes or remove their stories if necessary, providing them with control over their own content. |

### 

### 2.11 CM-11\_Add chapter to stories

#### *a. Functional Description*

| UC ID and Name: | **CM-11\_Add chapter to stories** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click on Dashboard button on menu  User click ‘Stories in charge’  Click on add chapter | | |
| Description: | As a user, i want to write new chapters of the stories | | |
| Preconditions: | User must logged in | | |
| Postconditions: | Chapter save to database successful  User can view the chapter of stories | | |
| Normal Flow | User click on the dash board on homepage  System show the dashboard  User click ‘Stories in charge’  User click on add chapter  User fill all blank field then click on submit button (See CM14\_EX1)  System save chapter of stories in database | | |
| Alternative Flows: | none | | |
| Exceptions: | CM14\_EX1 : User do not fill field : Request user to fill all blank field | | |
| Priority: | Should have | | |
| Frequency of Use: |  | | |
| Business Rules: | 2.14\_BR1, 2.14\_BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.14\_BR1 | Authoriz.. | Only User’s stories can add chapter of stories |
| 2.14\_BR2 |  | user cannot add a chapter if there are no existing stories existing |

### 

### 2.12 CM-12\_Edit chapter

#### *a. Functional Description*

| UC ID and Name: | **CM-12\_Edit chapter** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click on Dashboard button on menu  User click ‘Stories in charge’  User click story chapter  Click on edit chapter | | |
| Description: | As a user, i want to edit chapters of the stories | | |
| Preconditions: | User must logged in | | |
| Postconditions: | Chapter edited save to database successful  User can view the chapter edited of stories | | |
| Normal Flow | User click on the dash board on homepage  System show the dashboard  User click ‘Stories in charge’  User click on edit chapter  User fill all blank field then click on submit button (See CM15\_EX1)  System save chapter edited of stories in database | | |
| Alternative Flows: | none | | |
| Exceptions: | CM14\_EX1 : User do not fill field : Request user to fill all blank field | | |
| Priority: | Should have | | |
| Frequency of Use: |  | | |
| Business Rules: | 2.15\_BR1, 2.15\_BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.15\_BR1 | Authoriz.. | Only User’s stories edit chapter of stories |
| 2.15\_BR2 |  | user cannot edit a chapter if there are no existing stories existing |

### 2.13 CM-13\_Delete chapter

#### *a. Functional Description*

| UC ID and Name: | **CM-13\_Delete chapter** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click on Dashboard button on menu  User click ‘Stories in charge’  User click story chapter  Click on delete chapter | | |
| Description: | As a user, i want to delete chapters of the stories | | |
| Preconditions: | User must logged in | | |
| Postconditions: | Chapter should remove to database successful  User can’t see the chapter removed of stories | | |
| Normal Flow | User click on the dash board on homepage  System show the dashboard  User click ‘Stories in charge’  User click on delete chapter  System delete chapter of stories in database | | |
| Alternative Flows: | none | | |
| Exceptions: | none | | |
| Priority: | Should have | | |
| Frequency of Use: |  | | |
| Business Rules: | 2.16\_BR1 2.16\_BR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.16\_BR1 | Authoriz.. | Only User’s stories delete chapter of stories |
| 2.16\_BR2 |  | user cannot delete a chapter if there are no existing stories existing |

#### 

#### 

### 2.14 CM-14\_Update story

#### *a. Functional Description*

| UC ID and Name: | **CM-14\_Update story** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click Update story | | |
| Description: | This use case describes the process by which a registered user can update their story. | | |
| Preconditions: | 1. The user must be authenticated and logged into the system. 2. The user must have previously posted a story. | | |
| Postconditions: | The user's story is successfully updated on the platform. | | |
| Normal Flow | 1. User navigates to the section of the application dedicated to managing their posted stories. 2. User selects the story they want to update. 3. User clicks on the "Update Story" button for the selected story. 4. System presents a form or interface pre-filled with the existing story details (e.g., title, content, genre). 5. User makes the desired modifications to the story details. 6. User submits the updated story. 7. System verifies the input data and ensures all required fields are filled. 8. System saves the updated story to the platform's database. 9. System confirms successful update of the story to the user. | | |
| Alternative Flows: | If the user decides not to update the story after selecting it, they can navigate away from the update interface without submitting changes. | | |
| Exceptions: | 1. If there are errors in the input data (e.g., missing required fields, invalid content format), the system prompts the user to correct them before proceeding with the update. 2. If the system encounters technical issues during the update process, it displays an error message and prompts the user to try again later. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | 2.20\_BR1 | | |
| Other Information: | N/A | | |
| Assumptions: | 1. Users understand how to navigate the update interface and provide the necessary modifications to their story. 2. The system is stable and accessible during the update process. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.20\_BR1 | Users can only update stories that they have previously posted on the platform. | This rule specifies that users are only allowed to update stories that they have previously posted on the platform. Users cannot update stories posted by other users. This ensures that users have control over the content they have contributed to the platform and cannot modify content that does not belong to them. |

### 

### 2.15 CM-15\_Delete story

#### *a. Functional Description*

| UC ID and Name: | **CM-15\_Delete story** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 22/02/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User click button Delete story | | |
| Description: | This use case describes the process by which a registered user can delete their story. | | |
| Preconditions: | 1. The user must be authenticated and logged into the system. 2. The user must have previously posted a story. | | |
| Postconditions: | The user's story is successfully deleted their story. | | |
| Normal Flow | 1. User navigates to the section of the application dedicated to managing their posted stories. 2. User selects the story they want to delete. 3. User clicks on the "Delete Story" button for the selected story. 4. System prompts the user to confirm the deletion action. 5. User confirms the deletion action. 6. System deletes the selected story from the platform's database. 7. System confirms successful deletion of the story to the user. | | |
| Alternative Flows: | If the user decides not to delete the story after selecting it, they can cancel the deletion action, returning to the previous state without deleting the story. | | |
| Exceptions: | 1. If the system encounters technical issues during the deletion process, it displays an error message and prompts the user to try again later. 2. If the user's session expires or they are logged out during the deletion process, they are prompted to log in again before proceeding with the deletion. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | 2.21\_BR1 , 2.21\_BR2 | | |
| Other Information: | N/A | | |
| Assumptions: | 1. Users understand the consequences of deleting their story and are sure of their decision. 2. The system is stable and accessible during the deletion process. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.21\_BR1 | Users can only delete stories that they have previously posted . | This rule specifies that users are only allowed to delete stories that they have previously posted on the platform. Users cannot delete stories posted by other users. |
| 2.21\_BR2 | Deletion of a story should be irreversible and permanently remove the story from the platform. | This rule ensures that once a user deletes a story from the platform, the deletion action is irreversible and permanently removes the story from the platform's database. |

### 

### 

### 2.16 CM-16\_Send FeedBack

#### *a. Functional Description*

| UC ID and Name: | **CM-16\_Send FeedBack** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 10/03/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | User wants to provide feedback to the system. | | |
| Description: | This use case describes the process of a registered user sending feedback to the system. | | |
| Preconditions: | The user must be registered and logged into the system. | | |
| Postconditions: | The feedback is submitted to the system. | | |
| Normal Flow | 1. User navigates to the "Send Feedback" section . 2. User selects the type of feedback (e.g., suggestion, complaint, compliment). 3. User enters the feedback message. 4. User submits the feedback. 5. System acknowledges the receipt of the feedback and stores it for processing. | | |
| Alternative Flows: | None | | |
| Exceptions: | If the user's session expires during the feedback submission process, they are redirected to log in again before continuing. | | |
| Priority: | Medium | | |
| Frequency of Use: | Low to Medium | | |
| Business Rules: | 2.27\_BR1 , 2.27\_BR2 | | |
| Other Information: | Feedback may be reviewed and processed by system administrators . | | |
| Assumptions: | The system provides a dedicated interface or form for users to submit feedback.  Feedback submission process is secure and user information is handled confidentially. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.27\_BR1 | Feedback message length may be limited to a certain number of characters. | This rule specifies that the length of feedback messages may be restricted to a specific number of characters. This limitation can help ensure that feedback messages are concise and to the point. |
| 2.27\_BR2 | Inappropriate or offensive language in feedback messages may be filtered or flagged for moderation. | This rule states that any feedback messages containing inappropriate or offensive language may be subjected to filtering or flagged for moderation. This ensures that the feedback provided remains respectful and suitable for all users. |

### 2.17 CM-17\_Rate The Story

#### *a. Functional Description*

| UC ID and Name: | **CM-17\_Rate the story** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 11/03/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | The registered user wants to rate a story they have read. | | |
| Description: | This use case describes the process where a registered user rates a story they have read on the platform. | | |
| Preconditions: | The registered user has read a story on the platform. | | |
| Postconditions: | The user's rating is successfully recorded for the story. | | |
| Normal Flow | 1. The registered user accesses the story they want to rate on the platform.  2.The user selects the option to rate the story.  3.The system displays a rating interface for the user to provide their rating.  4.The user provides a rating for the story.  5.The system records the user's rating for the story.  6.The use case ends. | | |
| Alternative Flows: | If the user decides not to rate the story, they can navigate away from the rating interface, and the use case ends. | | |
| Exceptions: | If the system encounters an error while recording the user's rating, an error message is displayed, and the user is prompted to try again. | | |
| Priority: | Medium | | |
| Frequency of Use: | Moderate | | |
| Business Rules: | 2.28\_BR1, 2.28\_BR2 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.28\_BR1 | Registered users are allowed to rate stories they have read. | Registered users are permitted to rate stories they have read on the platform. |
| 2.28\_BR2 | Ratings must be recorded accurately and securely in the system. | Ratings provided by users must be accurately recorded and securely stored within the system. |

#### 

#### 

### 2.18 CM-18\_View list of Feedback

#### *a. Functional Description*

| UC ID and Name: | **CM-18\_View list of Feedback** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 11/03/2024 |
| Primary Actor: | Registered user | Secondary Actors: | None |
| Trigger: | The registered user wants to view their payment history on the platform. | | |
| Description: | This use case describes the process where a registered user accesses and views their payment history on the platform. | | |
| Preconditions: | The registered user has made one or more payments on the platform. | | |
| Postconditions: | The user successfully views their payment history. | | |
| Normal Flow | 1. The registered user logs into their account on the platform.  2.The user navigates to the payment history section.  3.The system retrieves and displays the user's payment history, including transaction dates, amounts, and status.  4.The user reviews their payment history.  5.The use case ends. | | |
| Alternative Flows: | If the user has not made any payments yet, the system displays a message indicating that there is no payment history available, and the use case ends. | | |
| Exceptions: | If the system encounters an error while retrieving the user's payment history, an error message is displayed, and the user is prompted to try again. | | |
| Priority: | Medium | | |
| Frequency of Use: | Moderate | | |
| Business Rules: | 2.30\_BR1, 2.30\_BR2 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 2.30\_BR1 | Only administrators are allowed to view the list of feedback. | Only administrators have the permission to access and view the list of feedback provided by users. |
| 2.30\_BR2 | Feedback must be displayed accurately and securely. | The feedback displayed to the administrator must be accurate and securely stored within the system. |

#### 

#### 

# 3. Admin Functions

### 3.1 AD-1 View list account

#### *a. Functional Description*

| UC ID and Name: | **AD-1\_View list account** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 1/03/2024 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin click on Dashboard button on menu  Admin click ‘Quản lý tài khoản’ | | |
| Description: | As a admin, I want to see a list of user accounts | | |
| Preconditions: | The Admin must logged into the system. | | |
| Postconditions: | The Admin successfully views the list of register user’s account | | |
| Normal Flow | Admin click on the dash board on homepage  System show the dashboard  Admin click ‘Quản lý tài khoản’  System views the list of register user’s account | | |
| Alternative Flows: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | 3.1\_AD1\_1, 3.1\_AD1\_2 | | |
| Other Information: | N/A | | |
| Assumptions: | Users are familiar with the process of navigating the application and accessing their profile/dashboard.  The system is stable and accessible during the process of retrieving and displaying posted stories. | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 3.1\_AD1\_1 | Admin Access | Only users with admin privileges can access the 'Quản lý tài khoản' functionality. |
| 3.1\_AD1\_2 | User Authentication | The admin must be authenticated and logged into the system to access the user account list. |

# 

### 3.2 AD-2 Authorization

#### *a. Functional Description*

| UC ID and Name: | **AD-2\_Authorization** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 11/03/2024 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | The system needs to determine a user's access rights to specific features or resources. | | |
| Description: | This use case describes the process of authenticating and determining a user's access rights to features or resources within the system. | | |
| Preconditions: | The admin is logged into the system. | | |
| Postconditions: | The system identifies and grants access rights to the user based on their role and permissions. | | |
| Normal Flow | 1.The admin logs into the system.  2.The system verifies the admin's login information.  3.The system determines the admin's role and permissions based on the login information and system configuration.  4.The system grants the admin access rights to specific features or resources based on the determined role and permissions.  5.The admin can access the permitted features or resources.  6.The use case ends. | | |
| Alternative Flows: | If the admin's login information is invalid, the system denies authentication and prompts the admin to provide correct login information. | | |
| Exceptions: | If the system encounters an issue during authentication or determining access rights, an error message is displayed, and the admin cannot access the feature or resource. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | 3.2\_BR1, 3.2\_BR2 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 3.2\_BR1 | Users must have valid credentials and appropriate permissions to access features or resources. | The system must ensure that only authenticated users with valid credentials and appropriate permissions are granted access to features or resources. |
| 3.2\_BR2 | Access rights granted to users must adhere to the principle of least privilege. | Access rights granted to users must adhere to the principle of least privilege, ensuring that users only have access to the features or resources necessary for their role or function within the system. |

### 

### 3.3 AD-3 Add account from Admin

#### *a. Functional Description*

| UC ID and Name: | **AD-3\_Add account from admin** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 5/03/2024 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Admin click on Dashboard button on menu  Admin click ‘Quản lý tài khoản’ Admin click ‘Tạo tài khoản mới’ | | |
| Description: | As a admin, I want to create new accounts can set role | | |
| Preconditions: | The Admin must logged into the system. | | |
| Postconditions: | The Admin successfully create new accounts can set role | | |
| Normal Flow | Admin click on the dash board on homepage  System show the dashboard  Admin click ‘Quản lý tài khoản’  System views the list of register user’s account  Admin click ‘Tạo tài khoản mới’ to ban account  Admin create a new account sucessfull | | |
| Alternative Flows: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | 3.5\_AD5\_1, 3.5\_AD5\_2 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 3.5\_AD5\_1 | Role Assignment | Admins can assign roles to new accounts during the creation process. |
| 3.5\_AD5\_2 | Unique Username | Usernames for new accounts must be unique across the system. |

### 3.4 AD-4 ViewReport

#### *a. Functional Description*

| UC ID and Name: | **AD-4 ViewReport** | | |
| --- | --- | --- | --- |
| Created By: |  | Date Created: | 11/03/2024 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Administrators receive reports from users and can view reports and response reports | | |
| Description: | This use case describes the process by which an administrator views reports and the administrator performs the function of responding to those reports. | | |
| Preconditions: | Administrators receive reports from users to check reports about stories. | | |
| Postconditions: | The user's account is successfully restored by the admin. | | |
| Normal Flow | 1.The administrator receives the report request and accesses the system's administration interface.  2. Administrator sees all view report of user report.  3.The administrator starts checking the sections that users report and can then respond to users.  4.The system displays the admin's response to the reports.  6. Use case ends. | | |
| Alternative Flows: | If administrators cannot identify the error the user reports want, they can ask the user for additional information to support the reporting process. | | |
| Exceptions: | If the system encounters an error during the account restoration process, an error message is displayed, and the admin may need to retry or seek assistance from technical support. | | |
| Priority: | Medium | | |
| Frequency of Use: | Low | | |
| Business Rules: | 3.3\_BR1, 3.3\_BR2 | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### *b. Business Rules*

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| 3.6\_BR1 | The account must be granted admin rights to use the function | Users must have admin rights to view the report list and respond |
| 3.6\_BR2 |  | Users must be logged in to be able to use the report |

# 

# III. Design Specifications

## 1. Public

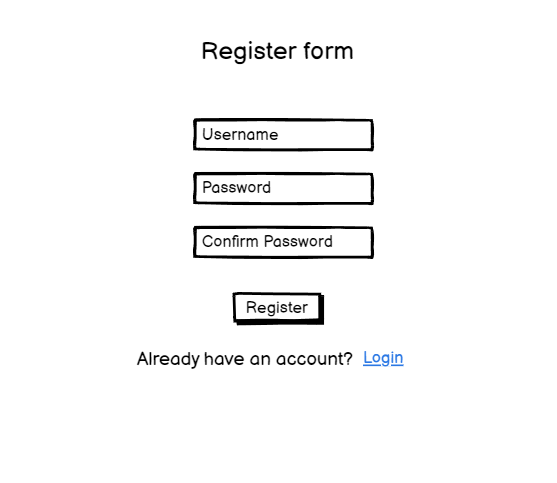
#### *1.1. Register*

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

Related use cases:

* [PB-1\_Register System](#_2jxsxqh)

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Username\* | Text Box | This is for guest to input valid username for register |
| Password\* | Password Box | This is for guest to input password for register |
| Confirm password\* | Button | This is for guest to re-input password for register |
| Sign up | Button | User clicks to redirect to the User Register page for registering new user account to access the system |
| Login | Hyperlink | User clicks to redirect to the Login page for login |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | Verify username |
| User | C | Create new user’s information |

***SQL Commands:***

1/ Verify username

select \* from [User]

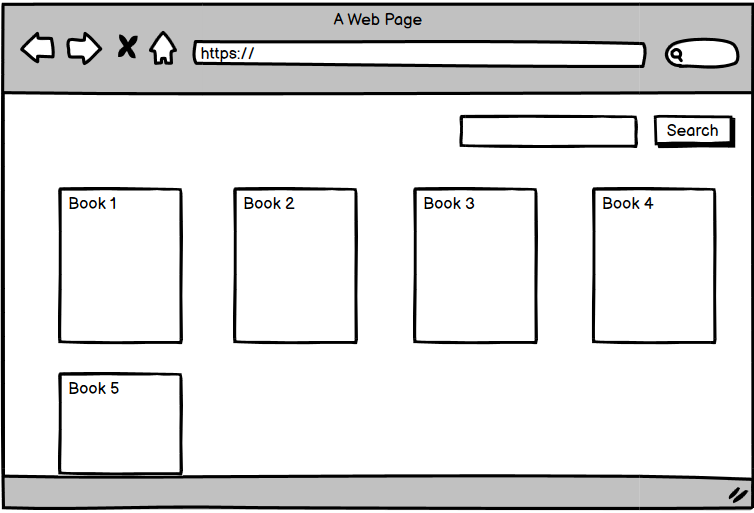
where username = ?

2/ Create new user

insert into [User] values(?,?,?,?)

#### *1.2. View homepage*

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thể Loại | Dropdown-list | View all category |
| Theo Số Chương | Dropdown-list | Filter book by chapter |
| Tìm kiếm | Text Box | Search book by title |
| Truyện hot | Button | View book most views |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Category | R | Show category on screen |

***SQL Commands:***

1/ Show category on screen

select \* from Category

2/ Show all Book by View

select top 10 \* from Book\n"

+ " order by [Views] desc

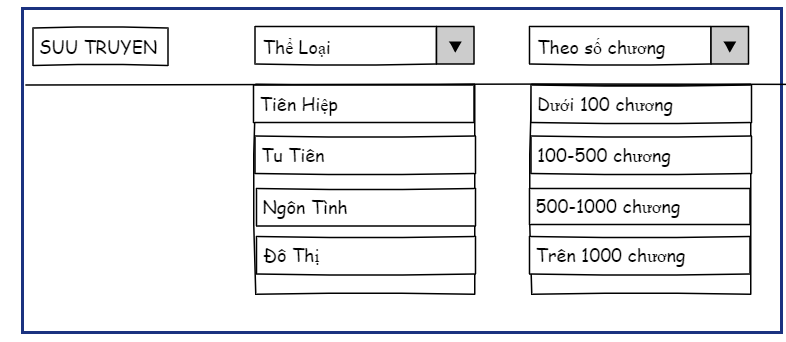
3/ Show book by View

select top 10 \* from Book\n"

+ " order by [Views] desc

#### *1.3. View category*

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thể Loại | Dropdown-list | View all category |
| Theo Số Chương | Dropdown-list | Filter book by chapter |
| Tìm kiếm | Text Box | Search book by title |
| Truyện hot | Button | View book most views |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Category | R | Show category on screen |

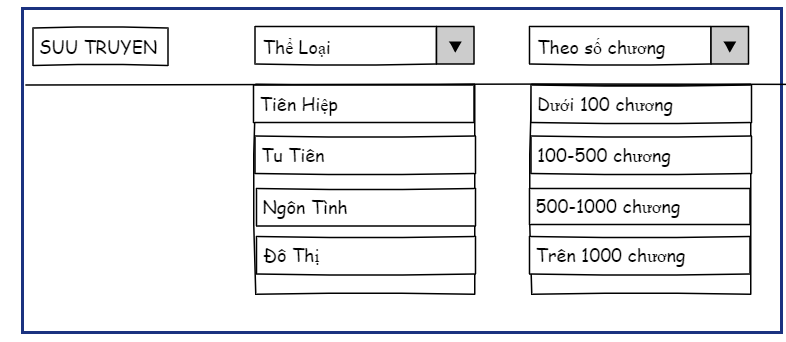
***SQL Commands:***

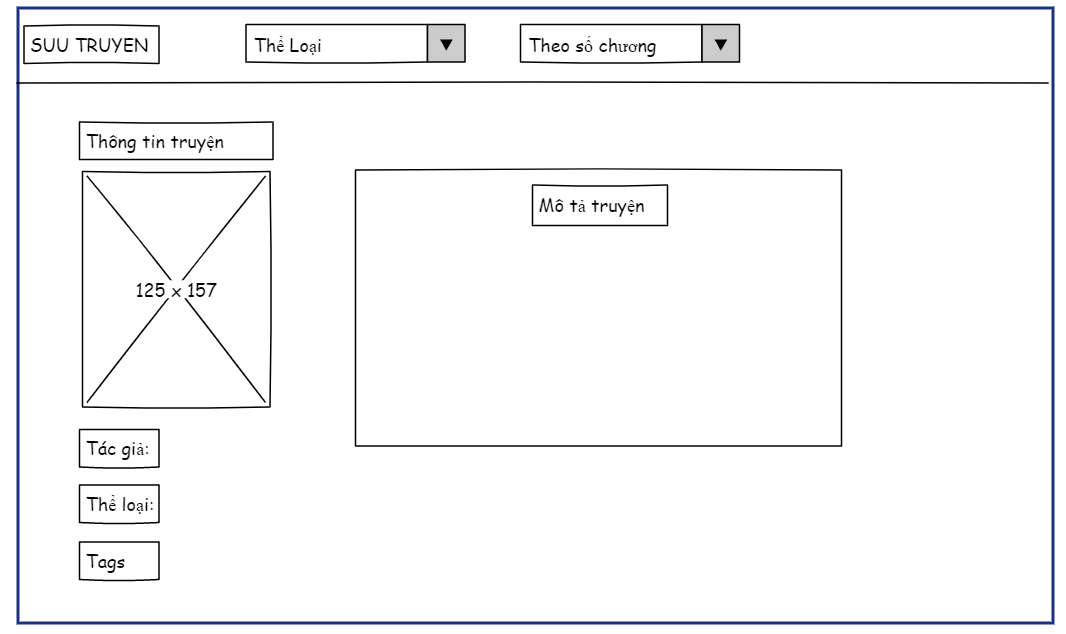
1/ Show category on screen

select \* from Category

#### *1.4. Filter*

##### **UI Design**





| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thể Loại | Dropdown-list | View all category |
| Theo Số Chương | Dropdown-list | Filter book by chapter |
| Tìm kiếm | Text Box | Search book by title |
| Tác giả | Button | View author of book |
| Thể loại | Button | View category of book |
| Tags | Button | View tags of book |
| Mô tả thể loại | Text | describe of each category |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Category | R | Show category on screen |
| Tag | R | Show tag name |
| Tag\_in\_book | R | Take tagid |
| Book | C | Show author on screen |

***SQL Commands:***

1/ filter category on screen

select \* from Category

2/ filter by book’s author

select \* from Book

where [authorName] COLLATE Latin1\_General\_CI\_AI = ?

3/ filter by chapter number

select b.bookID,b.creator,b.authorName,b.title,b.[views],b.detail,b.img,b.publishDate,b.[status],

count(b.bookID) as number of chapter

from Chapter c join Book b on c.BookID = b.bookID

group by b.bookID,b.creator,b.authorName,b.title,b.[views],b.detail,b.img,b.publishDate,b.[status]

having (count(b.bookID) >? and count(b.bookID) <?)

4,filter by tags

select \* from Book b join Tag\_in\_book cib on b.bookID = cib.BookID

join Tag c on c.TagID = cib.TagID\

where c.TagID= ?

5/Filter by category

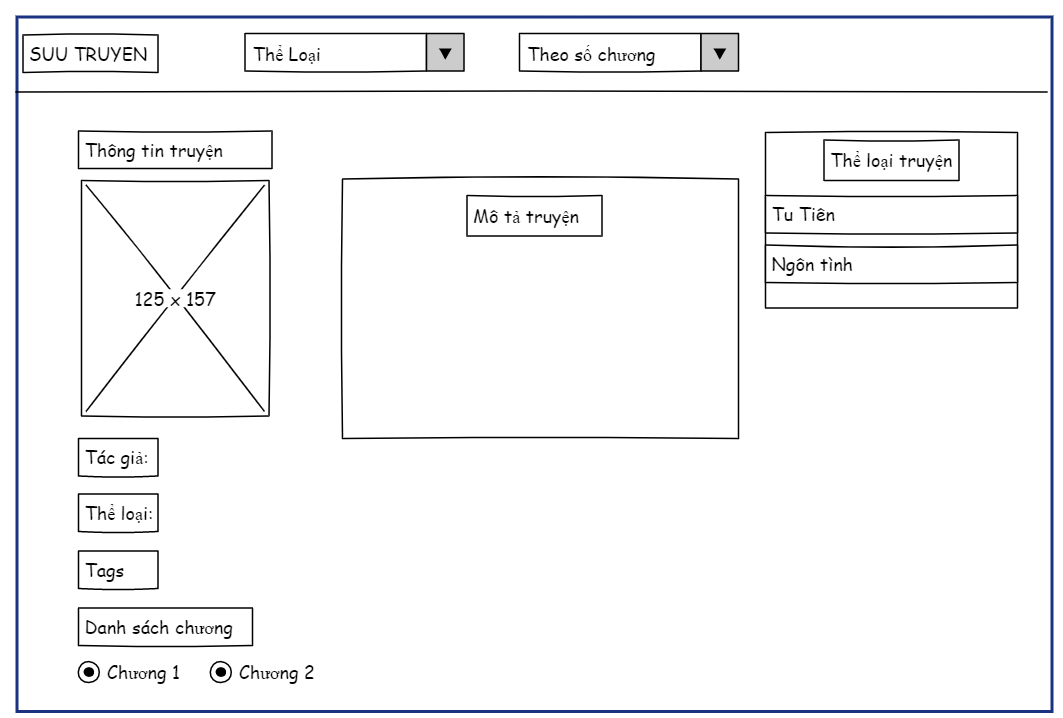
select \* from Book b join Category\_in\_book cib on b.bookID = cib.BookID

join Category c on c.CateID = cib.CateID

where c.CateID= ?

#### *1.5. View book’s detail*

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thể Loại | Dropdown-list | View all category |
| Theo Số Chương | Dropdown-list | Filter book by chapter |
| Tìm kiếm | Text Box | Search book by title |
| Tác giả | Button | View author of book |
| Thể loại | Button | View category of book |
| Tags | Button | View tags of book |
| Giới thiệu về truyện | Text | Introducing the book |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Category | R | Show category on screen |
| Book | C | Show list book on screen |
| Rate | R | Show point rate, rate times |

***SQL Commands:***

1/ Show book’s detail on screen

select \* from Book where bookID = ?

2/ Show category on screen

select \* from Category

3/Show list book on screen

select \* from Book b join Category\_in\_book cib on b.bookID = cib.BookID\n"

+ " join Category c on c.CateID = cib.CateID\n"

+ "where c.CateID= ?

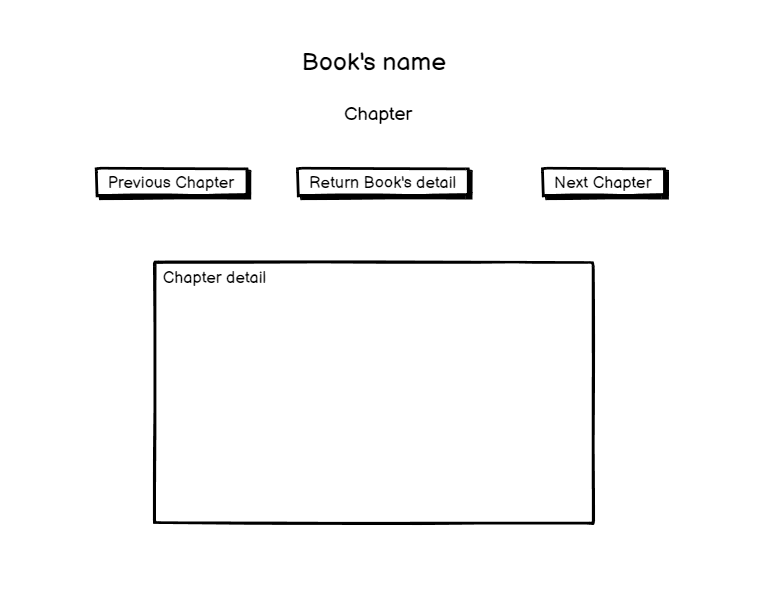
4, Show point rate:

String sql = "select AVG(point) from Rate where bookID = ?";

5, Show rate times  
 String sql = "select count(point) from Rate where bookID = ?";

#### *1.6. Reading chapter*

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thể Loại | Dropdown-list | View all category |
| Theo Số Chương | Dropdown-list | Filter book by chapter |
| Tìm kiếm | Text Box | Search book by title |
| Chương trước | Button | View previous chapter |
| Chương sau | Button | View next chapter |
| Description | Text Area  String (200) | Description of the setting |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Chapter | R | Show chapter’s detail on screen |
| Book | C | Show book’s title |
| PayChapter | R | Check Vip chapter Payed |

***SQL Commands:***

1/ Show chapter’s detail on screen

select \* from Chapter where BookID = ?

2/ Show book’s title

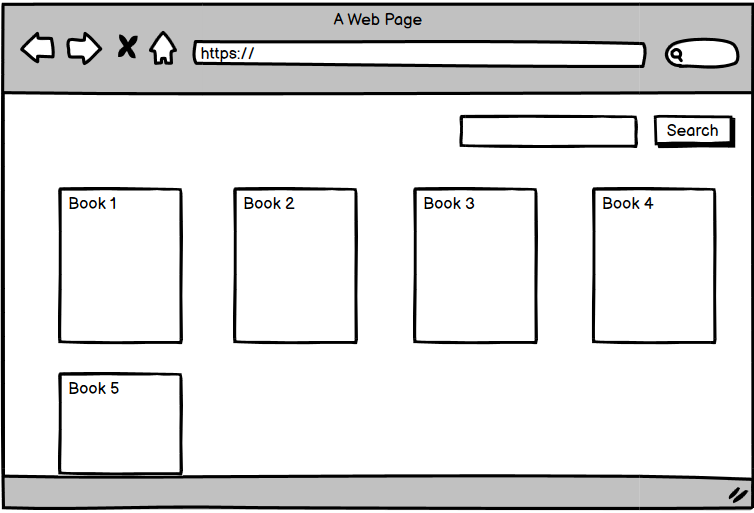
select \* from Book where bookID = ?

3/Check paychapter

select \* from PayChapter where userID = ? and chapterId = ?

#### *1.7. Search Book*

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thể Loại | Dropdown-list | View all category |
| Theo Số Chương | Dropdown-list | Filter book by chapter |
| Tìm kiếm | Text Box | Search book by title |
| Tác giả | Button | View author of book |
| Thể loại | Button | View category of book |
| Tags | Button | View tags of book |
| Giới thiệu về truyện | Text | Introducing the book |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Book | R | Show list book on screen |

***SQL Commands:***

1/ Show book on screen

String sql = "SELECT \*FROM Book Where title COLLATE Latin1\_General\_CI\_AI LIKE ?";

## 2. Common

### 1 Login

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Tên đăng nhập | text | input user name |
| Mật khẩu | text | input |
| Login | Submit | Submit account |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | check name |
| User | R | check pass |

***SQL Commands:***

String sql = "select \* from [User] where [username] = ? and [password] = ?";

### 

### 2 Logout System

##### **UI Design**

### 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Enter Log out | button | Exit the current account |

##### **Database Access**

none

***SQL Commands:***

none

### 

### 3 User Profile

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Category | Dropdown-list | View all category |
| By Chapter Number | Dropdown-list | Filter book by chapter |
| Search | Text Box | Search book by title |
| Email | Text | User’s email |
| Transaction | Text | User’s transaction |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | Get user’s information |

***SQL Commands:***

1/ Get user’s information

select \* from [User] where [username] = ? and [password] = ?

### 4. Change Password

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Current password | password | Enter old password |
| new password | password | Enter new Password |
| confirm password | password | Confirm new Password |
| Home | button | Back to home page |
| Change Password | button | Change password successful |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | U | Get user’s ID and change password |

***SQL Commands:***

1/ Get user’s information

update [User] set [password] = ?

where [userID] = ?

### 5. Update Profile

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Profile Image | text | Enter link image |
| Profile Image | img | Show image |
| Email | email | Enter your email |
| Address | text | Enter your address |
| Phone | text | Enter your phone number |
| Save | button | Save your change update profile |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | U | Get user’s ID and Update Profile |

***SQL Commands:***

1/Update user’s information

update [User] set [email] = ?,

[address] = ?,

[phone] = ?,

[avatar] = ?

where userID = ?

### 6. View my story

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Name story | text | Enter name story |
| Image | img | Show story image |
| Author name | text | Enter author story |
| Detail | text area | Enter story summary |
| Publish Date | date | Enter publish date story |
| Delete | button | Delete a story |
| Update | button | Update detail story |
| Search | search | Search story name |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Book | R | Get creatorID |
| User | R | List all book by userID |

***SQL Commands:***

1/ Get All Book by userID

select \* from [Book] b join [User] u on

b.creator = u.userID

where u.userID = ?

### 7. Post story

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Title | text | Enter Title Story |
| Author Name | text | Enter the author's name |
| Category | checkbox | Select story category |
| Publish Date | Date | Enter the story's publication date |
| Image | text | Enter the story image link |
| Detail | text area | Enter story summary |
| Save New Comic | button | Post New comic |
| Cancel | button | Remove all field |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Book | D | Delete Book by BookID |

***SQL Commands:***

1/ Insert into Book

INSERT INTO Book (title, authorName, publishDate, img, detail, status, creator)

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

2/ Insert into Category\_in\_Book

DECLARE @BookID INT;

SET @BookID = (SELECT BookID FROM Book WHERE BookID = ?);

INSERT INTO Category\_in\_book (BookID, CateID)

SELECT @BookID, CateID

FROM (

VALUES ?

) AS CateIDList(CateID);

### 8. Delete Story

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Name story | text | Enter name story |
| Image | img | Show story image |
| Author name | text | Enter author story |
| Detail | text area | Enter story summary |
| Publish Date | date | Enter publish date story |
| Delete | button | Delete a story |
| Update | button | Update detail story |
| Search | search | Search story name |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Book | D | Get BookID and Delete Book |
| Category\_in\_book | D | Get BookID and Delete Book |

***SQL Commands:***

1/ Delete Book and Category\_in\_book by BookID

DELETE FROM Category\_in\_book WHERE BookID = ?

DELETE FROM Book WHERE BookID = ?

### 9. Update Story

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Title | text | Update Title Story |
| Author Name | text | Update the author's name |
| Category | checkbox | Select new story category |
| Publish Date | Date | Update the story's publication date |
| Image | text | Update the story image link |
| Detail | text area | Update story summary |
| Update Comic | button | Update comic |
| Cancel | button | Cancel Update |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Book | U | Get BookID and Update Story |
| Category\_in\_book | U | Get BookID and Update Category Story |

***SQL Commands:***

1/ Update Book :

UPDATE [Book] SET title = ?, authorName = ?, publishDate = ?, img = ?, detail = ?, [status] = ? where [bookID] = ?

2/ Update Category:

DELETE FROM Category\_in\_book WHERE BookID = ?

INSERT INTO Category\_in\_book (BookID, CateID) VALUES (?, ?)

### 10. View chapter of stories

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Chương số | text | number chapter, chapter's name |
| User name | text | name of user |
| Sửa | button | Update chapter |
| Kết quả | button | view chapter details |
| Xóa | button | delete chapter |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Chapter | U | select value from table Chapter |
| User | U | Select user name from table User |

***SQL Commands:***

1/ View chapter:

Select \* from [Chapter]

2/ View user name:

Select userName from [User] where userID=?

### 11. Add Chapter

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Chương số | text | Insert number chapter |
| Tiêu đề | text | Insert the chapter's name |
| Nội dung | text area | Insert story’s content |
| save a new chapter | button | Insert chapter |
| Cancel | button | Cancel Update |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Chapter | R | Select least numberchap, list chap |
| Chapter | U | Insert value in table Chapter |

***SQL Commands:***

String sql = "select top 1\* from Chapter\n"

+ "where BookID = ?\n"

+ "Order BY NumberChapter Desc";

String sql = "select \* from Chapter where NumberChapter = ? and BookID=?";

Insert Chapter:

String sql = "INSERT INTO Chapter VALUES(?,?,?,?,?,?)";

### 12. Edit Chapter

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Chương số | text | Update number chapter |
| Tiêu đề | text | Update the chapter's name |
| Nội dung | text area | Update story’s content |
| Save | button | Update chapter |
| Cancel | button | Cancel Update |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Chapter | R | Select value of chapter |
| Chapter | U | Update value in table Chapter |

***SQL Commands:***

String sql = "select \* from Chapter where ChapterID=?";

Update Chapter:

String sql = "update Chapter\n"

+ "set [BookID] = ?,\n"

+ "[NumberChapter] = ?,\n"

+ "ChapterName = ?,\n"

+ "Contents\_1 = ?,\n"

+ "Contents\_2 = ?\n"

+ "where ChapterID = ?";

### 13. Delete Chapter

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Chương số | text | number chapter, chapter's name |
| User name | text | name of user |
| Sửa | button | Update chapter |
| Kết quả | button | view chapter details |
| Xóa | button | delete chapter |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Reading | D | Delete chapterid in table Reading |
| Chapter | D | Delete ChapterID in table Chapter |

***SQL Commands:***

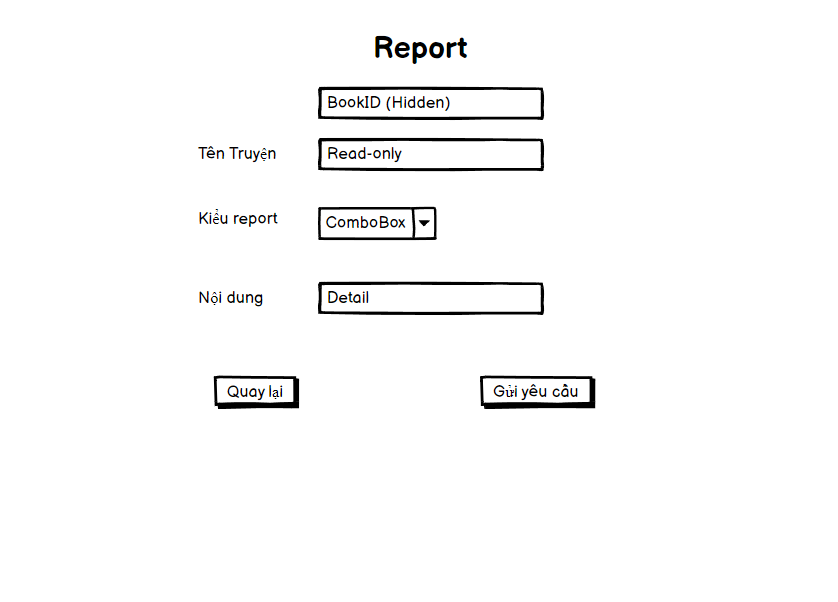
1/ Delete Chapter:

String sql= "DELETE FROM Chapter WHERE ChapterID = ?"

String sql= "DELETE FROM Reading WHERE chapterid = ?"

### 14. Story Report

##### **UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| id | text(hidden) | Book’s id |
| Tên truyện | text(Read only) | Insert book’s name |
| Loại báo cáo | dropdown list | Choose type of report |
| Nội dung | text | Insert report’s detail |
| Gửi | Button | Submit |
| Quay Lại | Button | Cancel |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Report | C | Insert value in table Report |
| Report Type | R | Read report type for Report |
| Chapter | R | Check exist chapter in book |

***SQL Commands:***

1/ Insert Report:

INSERT INTO Report VALUES (?, ?, ?, ?)

2/Report Type

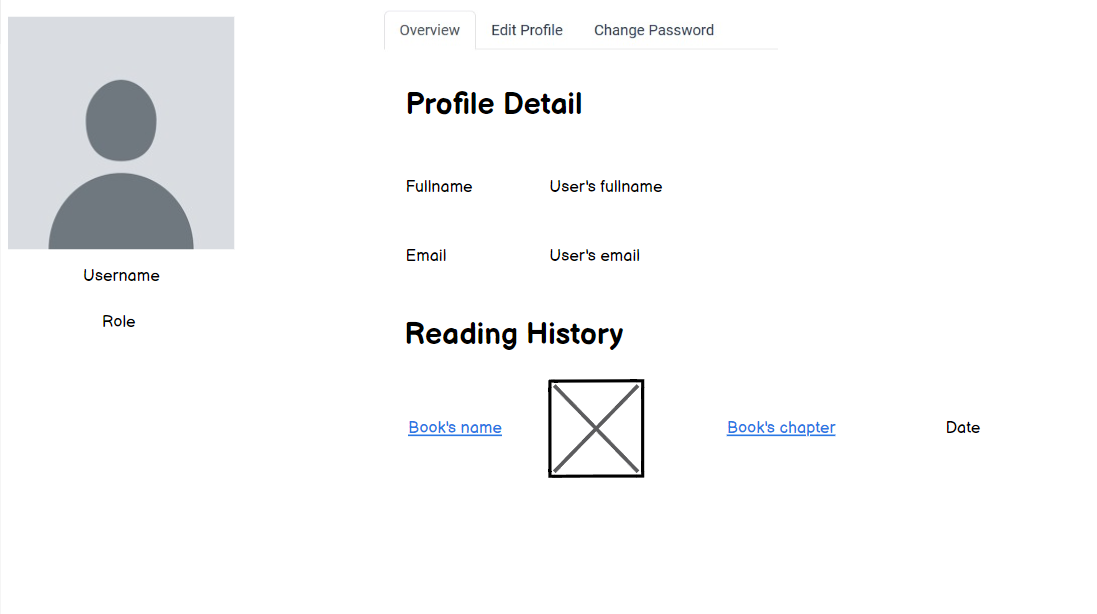
select \* from ReportType

3/Chapter

select \* from Chapter where bookID = ?

### 15. Reading History

##### **UI Design**

******

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| id | text(hidden) | Book’s id |
| Tên truyện | text | Insert book’s name |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Reading | R | Read value in table Reading |
| Insert | C | Insert new value to Reading |
| Reading | U | Update last time Reading |

***SQL Commands:***

1/ Insert Report:

select \* from Reading where userID = ?

2/Insert Reading

insert into Reading values(?,?,?,?)

3/Update Reading

update Reading set chapterid = ?, readingDate = ? where userid = ? and bookid = ?

### 16. Rate the story

##### **UI Design**

****

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Thần Mộ | text | The name of the story needs to vote |

##### 

##### 

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Rate | C | Create vote for that |
| Rate | R | Reading value |

***SQL Commands:***

1/ Add Rate

insert into Rate values (?,?,?)

2/Read rate

select \* from Rate where userID = ? and bookID = ?

## 

## 3. Admin

### 1. Views list user’s account

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| ID | text | View id user |
| Tên tài khoản | text | View username of user |
| Mật khẩu | password | View password of user |
| Email | text | View email of user |
| Chức năng | button | Ban user’account to comment in book |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | View ID, username, password, email |

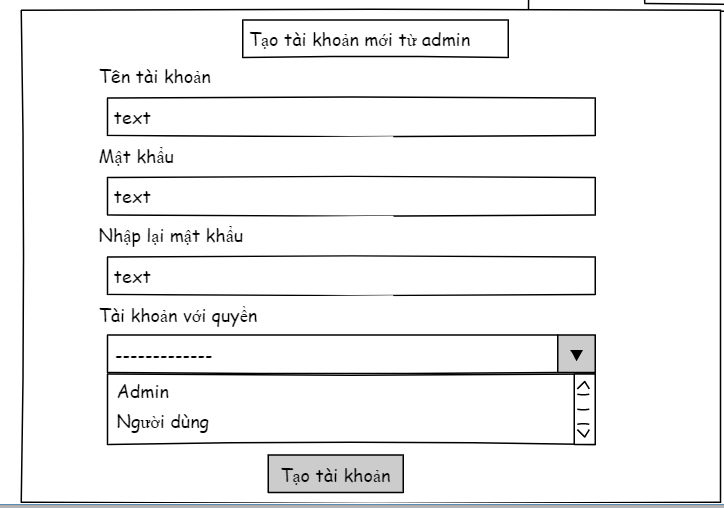
***SQL Commands:***

String sql = "select \* from [User] where [roleID]!=4";

### 2. Add account from admin

##### **UI Design**

# 



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Tên tài khoản | text | Input username of user |
| Mật khẩu | password | Input password of user |
| Nhập lại mật khẩu | password | Input repassword of user |
| Tài khoản với quyền | combobox | list role user and admin |
| Tạo tài khoản | button | submit to create new account |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | Check exit username |
| User | C | create ID, username, password, role |

***SQL Commands:***

String sql = "insert into [User] values(?,?,?,?,?,?,?,?)";

String sql = "select \* from [User] where username = ?";

### 3. Edit account

##### **UI Design**

# 

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Tên tài khoản | text | username of user |
| Mật khẩu | password | password of user |
| Email | text | email of user |
| Tài khoản với quyền | combobox | list role user and admin |
| Sửa tài khoản | button | submit to edit account |

##### **Database Access**

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | R | list ID, username, password, email, phone, role |
| User | U | Update role |

***SQL Commands:***

String query = "select \* from [User]\n"

+ "where userID = ?";

String query = "update [User]\n"

+ "set [roleID] = ?\n"

+ "where userID = ?";

### 4. View Feedback:

##### **UI Design**

##### **Database Access**

***SQL Commands:***

select f.\*,rt.reportType,u.userName from Feedback f "

+ "left join [User] u on f.userID = u.userID "

+ "join ReportType rt on rt.reportId = f.problem

# V. Appendix

## 1. Assumptions & Dependencies

AS-1: Readers will have access to devices with internet connectivity and appropriate web browsers to access the online reading platform.

AS-2: The online reading platform will support a large number of concurrent users accessing and reading content simultaneously without significant performance degradation.

AS-3: Readers will have basic literacy skills and familiarity with navigating web interfaces to effectively use the online reading platform.

DE-1: The online reading platform must be compatible with modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

DE-2: Payment processing services must be integrated into the platform to facilitate transactions for purchasing premium content or subscriptions.

DE-3: Content management systems or databases must be available to store and manage a vast library of digital books, novels, and other literary works.

DE-4: Reliable hosting services with sufficient bandwidth and storage capacity are required to host the online reading platform and its associated content.

DE-5: Collaboration with publishers or authors may be necessary to acquire licenses or permissions for distributing copyrighted material on the platform.

## 2. Limitations & Exclusions

*[Identify any product features or characteristics that a stakeholder might anticipate, but which are not planned to be included in the new product]*

LE-1: The online reading platform will not provide offline reading capabilities or downloadable content for access without an internet connection.

LE-2: Social networking features such as user-to-user messaging, friend lists, or community forums will not be implemented in the initial version of the platform.

LE-3: While the platform may offer recommendations based on reading preferences, personalized content suggestions or algorithmic recommendations will not be available in the first release.

LE-4: The online reading platform will not support the creation or publication of user-generated content, such as original stories or fan fiction.

LE-5: The platform will not offer interactive elements or gamification features within the reading experience, such as quizzes, puzzles, or mini-games embedded within the text.

## 3. Business Rules

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

<<Sample

| **ID** | **Category** | **Rule Definition** |
| --- | --- | --- |
| BR-01 | User | Users must register an account with valid email addresses and unique usernames. |
| BR-02 | Content | Only authorized publishers or authors can upload content to the platform. |
| BR-03 | Payment | Users must provide valid payment information to purchase premium content or subscribe to premium services. |
| BR-04 | Security | User passwords must be securely stored and encrypted in the database to protect against unauthorized access. |
| BR-05 | Privacy | The platform must comply with relevant data protection regulations, ensuring the privacy of user information. |
| BR-06 | Quality | Uploaded content must undergo a review process to ensure it meets quality standards and is free of errors. |
| BR-07 | Access | Users must have uninterrupted access to purchased content for the duration of their subscription period. |
| BR-08 | Support | Customer support services must be available to address user inquiries, technical issues, and complaints. |
| BR-09 | Licensing | The platform must acquire appropriate licenses or permissions for distributing copyrighted material. |
| BR-10 | Compliance | The platform must adhere to legal requirements regarding content censorship, hate speech, and obscenity. |
| BR-11 |  |  |
|  |  |  |
|  |  |  |

## 