

**READMATE**

**Software Requirement Specification**

– Hanoi, March 2023 –

**Table of Contents**

[I. Project Report 3](#_heading=h.gjdgxs)

[1. Status Report 3](#_heading=h.30j0zll)

[2. Team Involvements 3](#_heading=h.1fob9te)

[3. Issues/Suggestions 3](#_heading=h.3znysh7)

[II. System Requirement Specification 4](#_heading=h.2et92p0)

[1. Overall Description 4](#_heading=h.tyjcwt)

[1.1 Product Overview 4](#_heading=h.3dy6vkm)

[1.2 Business Rules 5](#_heading=h.1t3h5sf)

[2. User Requirements 6](#_heading=h.4d34og8)

[2.1 Overview 6](#_heading=h.2s8eyo1)

[2.2 Read - ReadMate](#_heading=h.17dp8vu)

1. Read Book/Story
2. Search by Categories
3. View Book/Story Details
4. View current list favourite Book/Story on ReadMate
5. Comment on Book/Story
6. Rate Book/Story
7. View history Book/Story
8. Buy unique chapter of Book/Story
9. Create favourite list Book/Story (for readers)

[2.3 Login - ReadMate](#_heading=h.lnxbz9)

1. Login

[2.4](#_heading=h.1ksv4uv) Register - ReadMate

1. Register Account
2. Register for V.I.P account
3. Register as Author

2.5 Release - ReadMate

1. CRUD Book/Story (draft)
2. Validate Book/Story
3. Release, Reject Book/Story

2.6 Gift - ReadMate

a. Send gift (donate) for author

b. View gift information

2.7 Manage Account - ReadMate

1. Manage Account break the rules

[3. Functional Requirements 12](#_heading=h.44sinio)

[3.1 System Functional Overview](#_heading=h.2jxsxqh)

1. Screen Flow
2. Screen Details
3. Screen Authorization
4. Non-screen Functions
5. Entity Relationship Diagram

[3.2 Login](#_heading=h.z337ya)

1. Function Name: Submit Order

[3.3 Home](#_heading=h.3j2qqm3)

3.4 Library

3.5 Searching story

3.9 Writing story

3.14 Payment

3.17 Comment

3.18 Discussion [14](#_heading=h.3j2qqm3)

[4. Non-Functional Requirements 15](#_heading=h.1y810tw)

[4.1 External Interfaces](#_heading=h.4i7ojhp)

1. User Interfaces
2. Software Interfaces
3. Hardware Interfaces
4. Communications Interfaces

[4.2 Quality Attributes](#_heading=h.qsh70q)

1. Usability
2. Reliability
3. Performance
4. Dependability

d1. Security

d2. Safety

1. Supportability

[5. Other Requirements 18](#_heading=h.ihv636)

[5.1 Appendix1 - Messages List 18](#_heading=h.32hioqz)

[18](#_heading=h.41mghml)

# I. Project Report

## 1. Status Report

| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| --- | --- | --- | --- |
| 1 | Overall Description | Completed | Product Overview, Business Rules |
| 2 | User Requirements | Completed | Overview, Use case details |
| 3 | Functional Requirement | Completed | System Functional Overview, Function Description |
| 4 | Non-Functional Requirement | Completed | Non-Functional Overview |

## 2. Team Involvements

| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| --- | --- | --- | --- |
| 1 | Document | Group 6 | 1.1 |
| 2 | Business Rules | Nguyễn Bá Hoàn | II. 1.2 |
| 3 | User Requirements Overview | Nguyễn Bá Hoàn | II. 2.1 |
| 4 | Read - Read Book/Story use case | Nguyễn Bá Hoàn | II. 2.2a |
| 5 | Read - Search by Categories use case | Nguyễn Bá Hoàn | II. 2.2b |
| 6 | Read - View Book/Story Details use case | Nguyễn Bá Hoàn | II. 2.2c |
| 7 | Read - View current list favourite Book/Story on ReadMate use case | Nguyễn Bá Hoàn | II 2.2d |
| 8 | Read - Comment on Book/Story use case | Nguyễn Bá Hoàn | II 2.2e |
| 9 | Read - Rate Book/Story use case | Nguyễn Bá Hoàn | II 2.2f |
| 10 | Read - View history Book/Story use case | Nguyễn Bá Hoàn | II 2.2g |
| 11 | Read - Buy unique chapter of Book/Story use case | Nguyễn Bá Hoàn | II 2.2h |
| 12 | Read - Create favourite list Book/Story (for readers) use case | Nguyễn Bá Hoàn | II 2.2i |
| 13 | Login - Login use case | Nguyễn Đức Anh | II 2.3a |
| 14 | Register - Register Account use case | Nguyễn Đức Anh | II 2.4a |
| 15 | Register - Register for V.I.P account use case | Nguyễn Bá Hoàn | II 2.4b |
| 16 | Register - Register as Author use case | Nguyễn Bá Hoàn | II 2.4c |
| 17 | Release - CRUD Book/Story (draft) use case | Nguyễn Đức Anh | II 2.5a |
| 18 | Release - Validate Book/Story use case | Nguyễn Bá Hoàn | II 2.5b |
| 19 | Release - Release, reject Book/Story use case | Nguyễn Đức Anh | II 2.5c |
| 20 | Gift - Send gift (donate) for author use case | Nguyễn Bá Hoàn | II 2.6a |
| 21 | Gift - View gift information use case | Nguyễn Bá Hoàn | II 2.6b |
| 22 | Manage - Manage Account use case | Nguyễn Bá Hoàn | II 2.7a |
| 23 | System Functional Overview | Phạm Thị Ngọc Huyền | II 3.1 |
| 24 | Login functional descriptions | Phạm Thị Ngọc Huyền | II 3.2 |
| 25 | Home functional descriptions | Phạm Thị Ngọc Huyền | II 3.3 |
| 26 | Library functional descriptions | Phạm Thị Ngọc Huyền | II 3.4 |
| 27 | Searching Story functional descriptions | Phạm Thị Ngọc Huyền | II 3.5 |
| 28 | Writing Story functional descriptions | Phạm Thị Ngọc Huyền | II 3.6 |
| 29 | Reading Story functional descriptions | Phạm Thị Ngọc Huyền | II 3.7 |
| 30 | Payment functional descriptions | Phạm Thị Ngọc Huyền | II 3.8 |
| 31 | Comment functional descriptions | Phạm Thị Ngọc Huyền | II 3.9 |
| 32 | Discussion functional descriptions | Phạm Thị Ngọc Huyền | II 3.10 |
| 33 | Non functional External Interfaces - User interfaces | Nguyễn Đức Anh | II 4.1a |
| 34 | Non functional External Interfaces - Software Interfaces | Nguyễn Đức Anh | II 4.1b |
| 36 | Non functional External Interfaces - Communication Interfaces | Nguyễn Đức Anh | II 4.1d |
| 37 | Non functional - Usability | Đoàn Văn Đông | II 4.2a |
| 38 | Non functional - Reliability | Đoàn Văn Đông | II 4.2b |
| 39 | Non functional - Performance | Đoàn Văn Đông | II 4.2c |
| 40 | Non functional - Dependability (Security) | Đoàn Văn Đông | II 4.2d1 |
| 41 | Non functional - Dependability (Safety) | Đoàn Văn Đông | II 4.2d2 |
| 42 | Non functional - Supportability | Đoàn Văn Đông | II 4.2e |
|  |  |  |  |

## 3. Issues/Suggestions

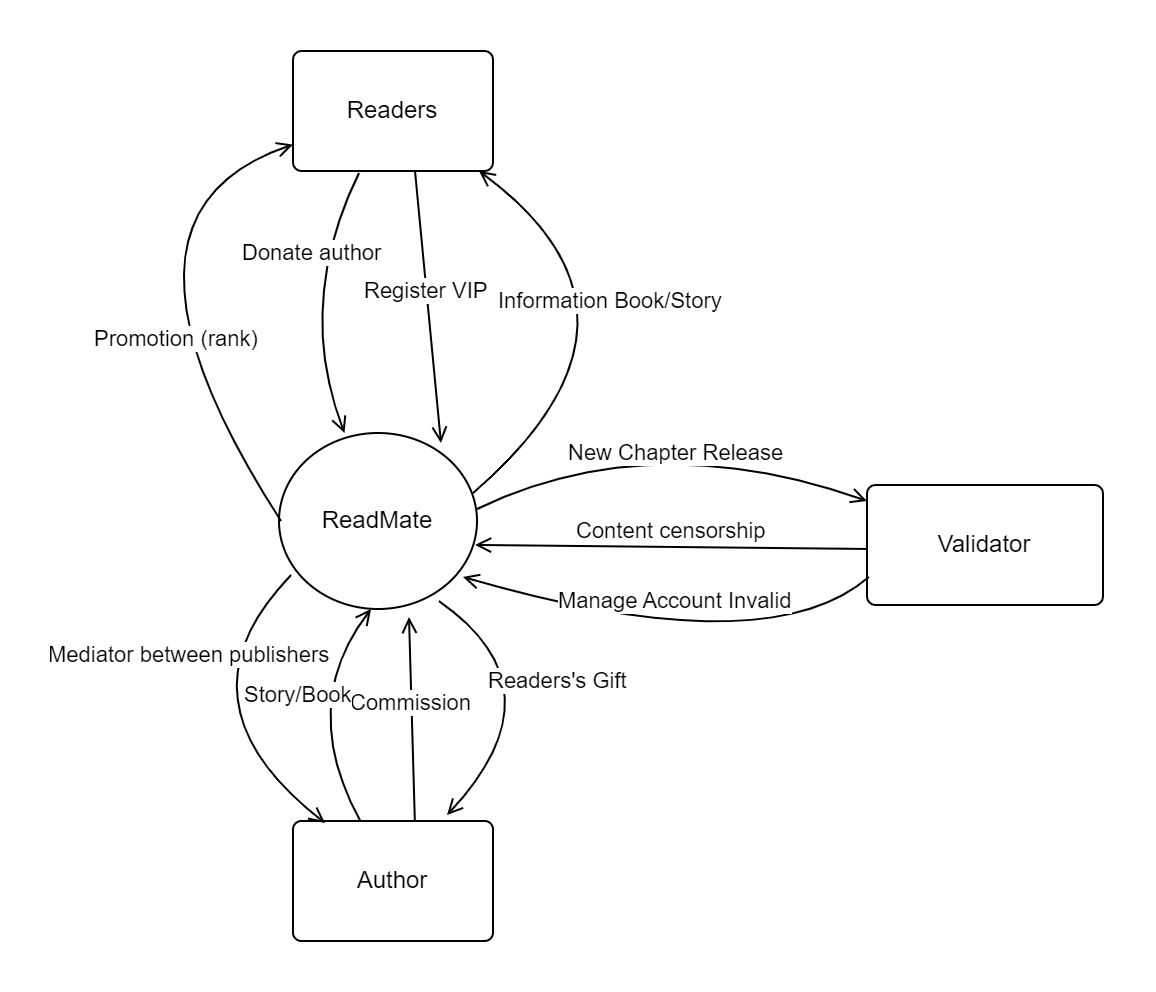
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| --- | --- | --- | --- |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. Software Requirement Specification

## 1. Overall Description

### 1.1 Product Overview

Readmate is a platform application that helps people to write their own stories, where anyone can become the author. The app will help people interact with, send gifts to the authors who love, promote your product and a lot of features will be developed later, the system will have an overview of the overview as follows:



### 1.2 Business Rules

| **ID** | **Rule Definition** |
| --- | --- |
| BR-01 | The works of the author need to approve the criteria before being able to publish |
| BR-02 | The price of the works will be set by the author |
| BR-03 | All copyright issues must be solved and handled quickly |
| BR-04 | Do not refund the amount the user has bought the work if there is no fault from the system and the author |
| BR-05 | Accounts that violate the rules more than 3 times will be permanently banned |
| BR-06 | Protect the privacy and security of users when processing personal and financial information. |
| BR-07 | Provide clear and prepaid prices to prevent any costs incurred for the reader. |
| BR-08 | Provide accurate proposals based on user interests and budgets, consider location, time and other related factors. |
| BR-09 | Provide high quality content and a lot of information to inspire and improve the experience of readers |
| BR-10 | Support users to solve any questions or concerns quickly and effectively. |
| BR-11 | Personal information of user is confidential and encrypted, only allowing information to see the consent of the user |
| BR-12 | Users need to ensure the rules of conduct when participating in the conversation, the user's account can be permanently locked if the user is reported. |
| BR-13 | The user chapters that buy from the app will not be returned in any form |
| BR-14 | Users need to ensure their personal information accurately (the application has a age limit for users 16+) |
| BR-15 | Users are responsible for themselves if they give personal information to strangers or fraud or appropriation of property. |
| BR-16 | Users need to upgrade to the VIP account to fully experience the feature that the app brings. |
| BR-17 | Users can use the app without registering and creating an account, but to use certain functions of the app, users will have to create an account. |
| BR-18 | All resources posted on the App must comply with the current law provisions, general regulations and regulations on the content of the app. |
| BR-19 | All standard works are licensed by the app to distribute them on the basis of paid. |
| BR-20 | Users are not allowed to copy, clone, sell, and sell services for any purpose, unless the actions have been clearly allowed by the owner. |
| BR-21 | Keywords search stories or books by names are not empty or only special characters, spaces. |
| BR-22 | User's user name and password when logging in must be filled with exactly (with a discrimination of uppercase) to be successfully logged in. |
| BR-23 | When registering an account, users must create a password with enough flower and usually the same digit with the length of not more than 255 characters |
| BR-24 | The original price information of the product is not blank or only spaces, and must be the number with a limit between 0 to 2147483648. |
| BR-25 | Price information of the product is limited to empty or only space, and must be the number with a limit between 0 to 2147483648. |
| BR-26 | Always update the situation of the book as completed, updated or delayed |
| BR-27 | High rankings can receive promotions created by the author of specific books |
| BR-28 | Description information, initial introduction about books are not blank or have special characters inappropriate |
| BR-29 | The author's author information is not blank or has special characters inappropriate |
| BR-30 | The information of the book's release time is not blank or has special characters inappropriate |
| BR-31 | Before releasing the latest chapter, the author must notify the specific time for the reader |
| BR-32 | When delayed, update the time to release books, the author must notify the user in advance |
| BR-33 | When creating a new chapter, the author must specify the title of that chapter |
| BR-34 | When creating a new product, do not leave the book genre, limit age, as well as notify the expected release time of that product |
| BR-35 | Before the product release, the author must ensure that his products do not violate the App internal rules and violate the law |
| BR-36 | When each chapter is posted, users are allowed to comment on that chapter |
| BR-37 | Users are allowed to vote star for that each book is released, rate from 1 - 5 |
| BR-38 | The type of stories and series that are most read will be displayed on the top of the page for users to easily identify |
| BR-39 | When editing or creating new products, the name of the classification is not allowed to leave blank or only spaces, and the number of characters must not exceed 255. |
| BR-40 | After the first part of the page will display information of the book's release schedule with specific time frames in the nearest time |
| BR-41 | The main content of the homepage screen will display a list of the latest series of books |
| BR-42 | Each set of books is displayed on the lists must display basic information about that series such as the total story chapter, the nearest update time |
| BR-43 | When the user comments on the books that use vulgar and vulgar words will be displayed after automatic censorship into \*\*\*\* |
| BR-44 | User negative comments may be reported by other users |
| BR-45 | The validator may directly prohibit the accounts that violate the rules more than 3 times |

## 2. User Requirements

- Easy Navigation: The app should have a user-friendly interface that is easy to navigate, allowing users to quickly find the books, articles or content they are interested in.

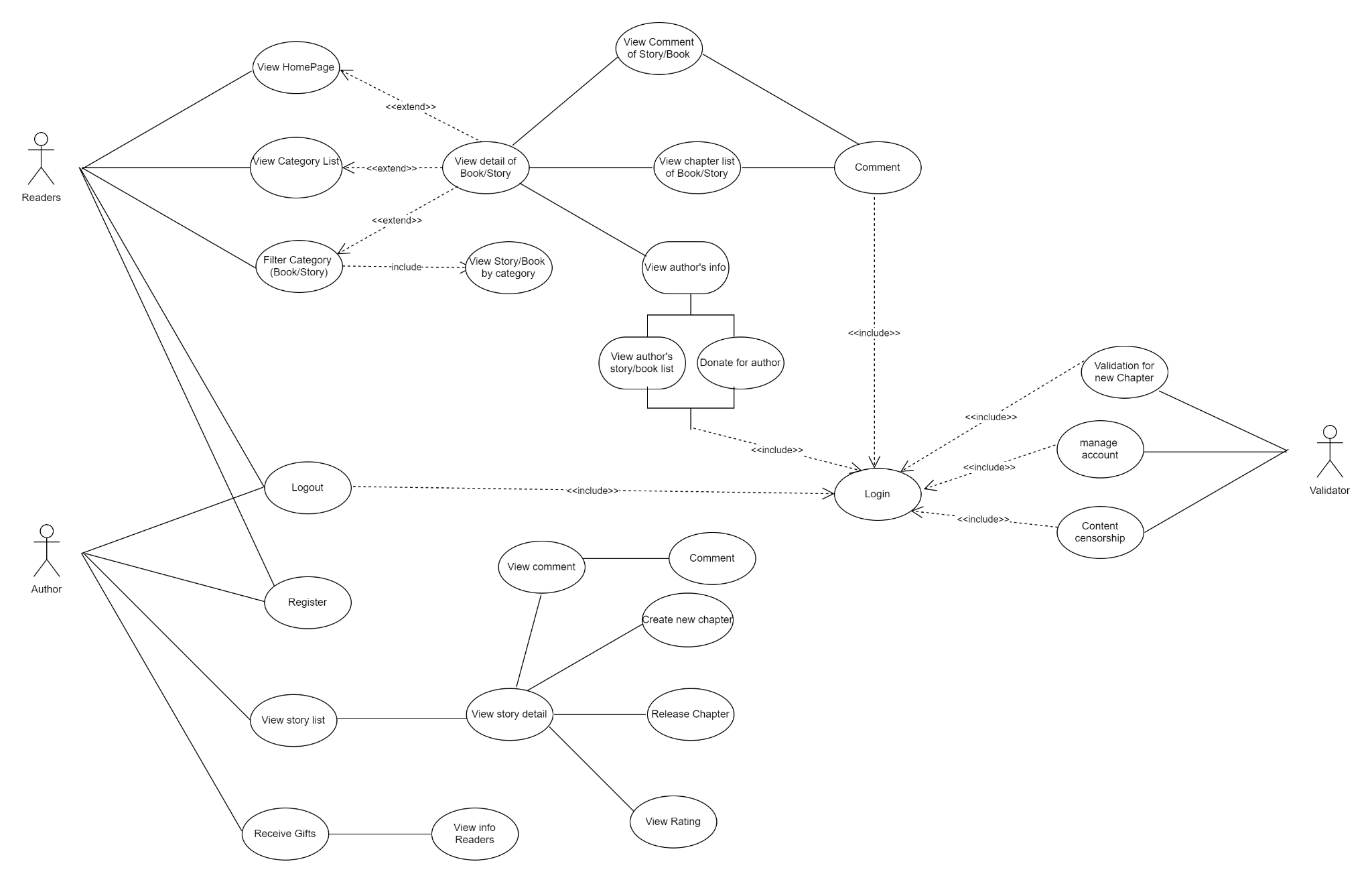
- Social Features: The app should allow users to connect with other readers and share book recommendations, reviews, and comments.

- Offline Access: The app should allow users to download content for offline reading when they don't have an internet connection

- Search Function: The app should have a search function that allows users to find specific books or articles based on keywords or author name.

### 2.1 Overview

#### a. Use Case Diagram

*[*

#### b. System Actors

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Readers | Read and use basic services of the application such as buying chapter, commenting, product evaluation, gift giving to the author, ... etc |
| 2 | Author | Release products, receiving gifts from readers, commenting, viewing rate, ... etc |
| 3 | Validator | Validate content books, management of accounts that break the rules |

#### c. Use Cases List

#### 

| **ID** | **Use Case** | **Primary Actors** | **Secondary Actors** |
| --- | --- | --- | --- |
| 01 | Read Book/Story | Readers | System |
| 02 | Search by Categories | Readers | System |
| 03 | View Book/Story Details | Readers | System |
| 04 | View current list favourite Book/Story on ReadMate | Readers, Author | System |
| 05 | Register account | Readers, Author | System |
| 06 | Login | Readers, Author, Validator | System |
| 07 | Register for V.I.P account | Readers | System |
| 08 | Register as Author | User | System |
| 09 | CRUD Book/Story (draft) | Author | System |
| 10 | Validate Book/Story | Validator | System |
| 11 | Release, Reject Book/Story | Validator | System |
| 12 | Comment on Book/Story | Readers, Author | System |
| 13 | Rate Book/Story | Readers | System |
| 14 | Send Gift (Donate) for Author | Readers | Author |
| 15 | View Gift Information | Author | System |
| 16 | Buy Unique Chapter of Book/Story | Readers | System |
| 17 | View history Book/Story | Readers | System |
| 18 | Create favourite list of Book/Story (for readers) | Readers | System |
| 19 | Manage account break the rules | Validator | Readers, System |

### 

### 2.2 Read - ReadMate

#### a. Read Book/Story

| UC ID and Name: | **001 - Read Book/Story** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 08/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Trigger: | User selects a book/story to read from the app library | | |
| Description: | This use case describes the process of reading a book/story using the ReadMate feature. The user selects a book/story from the app library, and the app displays the content on the user's screen. The user can then read the content and navigate through the pages using the app's controls | | |
| Preconditions: | 1. The user can have a registered account and be logged into the app. (not required) 2. The user must have access to the app's library of books/stories. 3. The selected book/story must be available and accessible to the user. | | |
| Post-conditions: | 1. The selected book/story has been opened and is displayed on the user's screen. 2. The app records the user's progress in reading the book/story | | |
| Normal Flow: | 1. User selects a book/story from the app library. 2. The app retrieves the selected book/story from the library. 3. The app displays the book/story content on the user's screen. 4. The user reads the book/story content and navigates through the pages using the app's controls. 5. The user finishes reading the book/story | | |
| Alternative Flows: | User searches for a specific book/story in the app library.  The selected book/story is not available or accessible to the user.  The user adjusts the font size or background colour of the app to make reading easier.  The user bookmarks a specific page for later reference | | |
| Exceptions: | The selected book/story is not available or accessible to the user. The app displays an error message and prompts the user to select another book/story | | |
| Priority: | High | | |
| Frequency of Use: | Daily | | |
| Business Rules: | BR-21, BR-24, BR-25, BR-28, BR-29 | | |
| Other Information: | The app should allow the user to highlight and take notes while reading.  If the app loses network connectivity or experiences a timeout during use case execution, the app should save the user's progress and restore it when connectivity is restored.  If the use case execution fails for some unanticipated or systemic reason, the user should be prompted to save their progress and exit the app.  The app assumes that the user has a basic understanding of how to use a mobile app and navigate through its controls | | |
| Assumptions: | none | | |

#### b. Search by Categories

| ID and Name: | **002 - Search by Categories** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 09/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process of searching for books or stories by category. The user can input a category and the system will return a list of books or stories that belong to that category. | | |
| Trigger: | The user wants to search for books or stories by category | | |
| Preconditions: | The user has access to the system.  The user has entered the search function.  The system has a database of books or stories with assigned categories | | |
| Post-conditions: | The system displays a list of books or stories that belong to the searched category.  The system allows the user to view the details of the selected book or story | | |
| Normal Flow: | 1. The user enters the search function. 2. The system presents the search bar. 3. The user inputs the desired category. 4. The system searches the database for books or stories that belong to the searched category. 5. The system presents a list of books or stories that belong to the searched category. 6. The user selects a book or story from the list. 7. The system displays the details of the selected book or story | | |
| Alternative Flows: | If there are no books or stories found that belong to the searched category, the system displays a message saying "No results found".  If the user wants to refine the search, the user can input a new category or refine the search with additional filters. | | |
| Exceptions: | The system encounters an error while searching the database. The system displays an error message and prompts the user to try again.  The user inputs an invalid or non-existent category. The system displays an error message and prompts the user to try again. | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times a day | | |
| Business Rules: | BR-21, BR-26, BR-28 | | |
| Other Information: | 1. The system allows the user to filter the search results by additional criteria such as author or publication date. 2. The system allows the user to sort the search results by relevance, popularity or publication date. 3. The system allows the user to save books or stories to a reading list for later use | | |
| Assumptions: | The user has a basic understanding of the system's categories and how they are assigned to books or stories | | |

#### 

#### 

#### 

#### 

#### 

#### c. View Book/Story Details

#### 

| ID and Name: | **003 - View Book/Story Details** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 09/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case allows a user to view the details of a selected book or story | | |
| Trigger: | User selects a book/story to view its details | | |
| Preconditions: | The user must have a book or story selected to view its details. | | |
| Post-conditions: | The user can view the book/story details including the title, author, cover image, summary, and other relevant information | | |
| Normal Flow: | 1. User selects a book/story to view its details.  2. System displays the book/story details including the title, author, cover image, summary, and other relevant information.  3. Users can scroll through the details and view any additional information or media associated with the book/story.  4. Users can select to add the book/story to their reading list or library, share the book/story with friends, or return to the main screen | | |
| Alternative Flows: | If the selected book/story is not available or cannot be found  System displays an error message indicating the book/story cannot be found.  User can select to return to the main screen | | |
| Exceptions: | If there is a system error or technical issue:  System displays an error message indicating the problem.  User can try again later or contact customer support for assistance | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | BR-28, BR-29, BR-30 | | |
| Other Information: | Book/story details may include reviews, ratings, genre, publication date, length, language, and other relevant information.  Users may have the option to purchase the book/story or access it through a subscription service.  Users may be able to read a sample or preview of the book/story before making a purchase.  Users may have the option to leave a review or rating of the book/story after reading it.  If the user is accessing the Reading App through a mobile device, the book/story details may be optimised for smaller screens.  If the user has a slow internet connection or limited data plan, the book/story details may be optimised for faster loading times | | |
| Assumptions: | none | | |

#### 

#### d. View current list favourite Book/Story on ReadMate

#### 

| ID and Name: | **004 - View current list favourite Book/Story on ReadMate** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 10/03/23 |
| Primary Actor: | Readers, Author | Secondary Actors: | ReadMate System |
| Description: | This use case describes the steps taken by a user to view their list of favourite books/stories within the app | | |
| Trigger: | User selects the "View Favourites" option in the app's main menu | | |
| Preconditions: | The user has already marked one or more books/stories as "favourites."  The app is open and the user is logged in. | | |
| Post-conditions: | The user is presented with a list of their favourite books/stories.  The user can select a book/story from the list to view more details or read | | |
| Normal Flow: | The user selects the "View Favourites" option in the app's main menu.  The app retrieves the list of favourite books/stories for the user.  The app displays the list of favourite books/stories.  The user selects a book/story from the list to view more details or read.  The app displays the details of the selected book/story | | |
| Alternative Flows: | If the user selects a book/story that has been removed from current favourites list since their last visit, the app displays a message informing them of this and removes the book/story from the list | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times per day. | | |
| Business Rules: | BR-26, BR-28, BR-29, BR-30 | | |
| Other Information: | The list of favourite books/stories should be sorted by the date they were marked as favourites, with the most recent at the top.  The list should include the book/story title and author, as well as a thumbnail image | | |
| Assumptions: | None | | |

#### 

#### e. Comment on Book/Story

#### 

| ID and Name: | **012 - Comment on Book/Story** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 11/03/23 |
| Primary Actor: | Readers, Author | Secondary Actors: | ReadMate System |
| Description: | This use case allows users to leave comments on books/stories they have read. Comments can provide feedback to the author or publisher, as well as recommendations or warnings for other readers. Users can also view comments left by other readers | | |
| Trigger: | User selects the "Comment" option on the book/story detail page | | |
| Preconditions: | Users must be logged in to the app.  User must have read at least one book/story | | |
| Post-conditions: | User's comment is saved and added to the book/story's comment section | | |
| Normal Flow: | User selects the book/story they want to leave a comment on from their list of read books/stories.  User scrolls to the comment section of the book/story detail page.  User selects the "Add Comment" button.  System prompts the user to enter their comment.  Users type in their comment and select the "Submit" button.  System saves the comment and adds it to the book/story's comment section.  User can view their comment and comments left by other readers | | |
| Alternative Flows: | If a user tries to submit an empty comment, the system displays an error message and prompts the user to enter a valid comment.  If a user tries to submit a comment that exceeds the maximum character limit, the system displays an error message and prompts the user to shorten their comment.  If the system is unable to save the comment, system displays an error message and prompts user to try again later | | |
| Exceptions: | If the user is not logged in to the app, the system prompts the user to log in before they can leave a comment.  If user has not read any books/stories, system displays an error message and prompts user to read a book/story before they can leave a comment | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | BR-36, BR-43, BR-44, BR-45 | | |
| Other Information: | None | | |
| Assumptions: | The app has a character limit for comments to prevent abuse or spam.The app has a reporting system for inappropriate comments.Users have a clear understanding of how to leave comments and view comments left by others | | |

#### 

#### f. Rate Book/Story

#### 

| ID and Name: | **013 - Rate Book/Story** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 12/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process of allowing the user to rate a book/story they have read on the app. The rating will be used to provide a collective rating for the book/story, which can be viewed by other users of the app | | |
| Trigger: | User selects the option to rate a book/story on the app | | |
| Preconditions: | Users must be logged in to the app.  User must have read the book/story they want to rate | | |
| Post-conditions: | The rating is saved and added to the book/story's collective rating.  The user's rating is added to their personal rating history | | |
| Normal Flow: | User selects the book/story they want to rate.  User selects the rating they want to give (e.g., 1-5 stars).  System confirms the rating has been saved.  System updates the book/story's collective rating with the new rating.  User's personal rating history is updated with the new rating | | |
| Alternative Flows: | If the user has not read the book/story they want to rate, the system displays an error message and does not allow the rating to be submitted | | |
| Exceptions: | If the rating cannot be saved due to a technical issue, the system displays an error message and asks the user to try again later | | |
| Priority: | Medium | | |
| Frequency of Use: | Depends on user activity | | |
| Business Rules: | BR-37 | | |
| Other Information: | Users can only rate a book/story once.  Users can view the collective rating for a book/story before rating it themselves.  The rating system should be intuitive and user-friendly.  The rating system should prevent fraudulent ratings (e.g., spam ratings) to maintain the integrity of the collective rating | | |
| Assumptions: | None | | |

#### 

#### g. View history Book/Story

#### 

| ID and Name: | **017 - View history Book/Story** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 13/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case allows users to view their history of previously read books/stories in the app. It enables users to revisit and access the details of books/stories they have read before | | |
| Trigger: | User selects the "View History" button in the app | | |
| Preconditions: | Users must be logged in to the app.  User must have previously read and finished at least one book/story | | |
| Post-conditions: | The app displays a list of books/stories previously read by the user.  The user can access the details of each book/story in the list | | |
| Normal Flow: | User selects the "View History" button in the app.  The app retrieves and displays a list of books/stories previously read by the user.  The user selects a book/story from the list.  The app displays the details of the selected book/story, including the title, author, genre, rating, comments, and the date it was read | | |
| Alternative Flows: | If the user has not previously read any books/stories, the app displays a message informing the user that there is no history available | | |
| Exceptions: | If the app encounters an error while retrieving the history, it displays an error message and prompts the user to try again later | | |
| Priority: | Medium | | |
| Frequency of Use: | This use case is expected to be used frequently by users who want to revisit and access the details of books/stories they have read before | | |
| Business Rules: | BR-10, BR-29, BR-30 | | |
| Other Information: | Users can delete a book/story from their history.  The app automatically adds a book/story to the user's history after it has been read completely.  The app stores the history of read books/stories locally on the user's device | | |
| Assumptions: | None | | |

#### 

#### h. Buy unique chapter of Book/Story

#### 

| ID and Name: | **016 - Buy unique chapter of Book/Story** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 14/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case allows a user to purchase a unique chapter of a book/story instead of buying the entire book/story. This feature is especially useful for readers who only want to read specific chapters and do not want to purchase the entire book/story | | |
| Trigger: | User selects a specific chapter of a book/story to purchase. | | |
| Preconditions: | The user must be logged into their account  The book/story containing the chapter must be available for purchase  The user must have a valid payment method saved in their account. | | |
| Post-conditions: | The user's payment is processed and confirmed  The purchased chapter is added to the user's account for future access. | | |
| Normal Flow: | User selects a book/story from their library  User navigates to the book/story details page  User selects a specific chapter to purchase  System prompts user to confirm the purchase and displays the cost  User confirms the purchase  System processes the payment using the user's saved payment method  System confirms the payment and adds the purchased chapter to the user's library. | | |
| Alternative Flows: | User cancels the purchase: System cancels the purchase and returns the user to the book/story details page  Payment processing fails: System displays an error message to the user  User is prompted to try again or select a different payment method | | |
| Exceptions: | E1: User is not logged in  System prompts the user to log in or create an account  E2: Book/story is not available for purchase  System displays an error message to the user  User is returned to their library  E3: User does not have a valid payment method saved in their account  System prompts the user to add a payment method  User is redirected to their account settings page  E4: User's payment is declined  System displays an error message to the user  Users are prompted to try again or select a different payment method. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | BR-10, BR-11, BR-13, BR-27 | | |
| Other Information: | The cost of each chapter may vary depending on the book/story and the number of chapters already purchased by the user  The purchased chapter can be accessed at any time from the user's library  Users may purchase multiple chapters at once and the system will process all payments in a single transaction  If the purchase is successful, the user will receive a confirmation email with details of the transaction and the purchased chapter | | |
| Assumptions: | None | | |

#### 

#### i. Create favourite list Book/Story (for readers)

#### 

| ID and Name: | **018 - Create favourite list Book/Story (for readers)** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 14/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case allows users to create a new favourite list for books or stories they want to keep track of | | |
| Trigger: | The user selects the option to create a new favourite list for books or stories | | |
| Preconditions: | The user must have a registered account on the Reading App.  The user must be logged in to their account.  The Reading App must have books or stories available for the user to add to their favourite list | | |
| Post-conditions: | A new favourite list is created with the specified name.  The user can add books or stories to their favourite list.  The new favourite list is saved to the user's account | | |
| Normal Flow: | The user opens the Reading App.  The user navigates to their profile or settings section.  The user selects the option to create a new favourite list.  The Reading App prompts the user to enter a name for the new favourite list.  The user enters a name for the new favourite list and confirms.  The Reading App creates the new favourite list and displays it to the user.  The user can now add books or stories to the new favourite list by selecting the option to add to favourites and choosing the newly created list | | |
| Alternative Flows: | If the user cancels the creation of the new favourite list, the Reading App returns the user to their profile or settings section | | |
| Exceptions: | If the user is not logged in, the ReadMate prompts the user to log in or register before creating a new favourite list | | |
| Priority: | Medium | | |
| Frequency of Use: | This use case is expected to be used frequently by users who want to keep track of their favourite books or stories | | |
| Business Rules: | BR-26, BR-28 | | |
| Other Information: | The ReadMate should allow users to create multiple favourite lists.  Users should be able to view and edit their favourite lists at any time.  The ReadMate should provide users with the option to delete a favourite list or remove books or stories from a favourite list.  Users should be able to search for books or stories to add to their favourite list | | |
| Assumptions: | Users have the ability to add books/stories to their favourite list, and that this feature is visible and accessible within the app.  Users are able to easily navigate to the "Create favourite list" option within the app and understand how to use it.  The app is able to store and retrieve users' favourite lists, and that these lists persist across app sessions.  Users have an unlimited number of favourite lists they can create and edit.  Users have the ability to rename or delete their favourite lists as needed | | |

#### 

### 2.3 Login - ReadMate

#### a. Login

| ID and Name: | **UC-006 Login** | | |
| --- | --- | --- | --- |
| Created By: | Duc Anh | Date Created: | 10/03/2023 |
| Primary Actor: | Readers, Author, Validator | Secondary Actors: | ReadMate System |
| Description: | This use case allows a registered user to log in to the system to access their account and perform various actions such as | | |
| Trigger: | User clicks on the login button on the system's login page. | | |
| Preconditions: | Users are registered with the system.  User has an active internet connection.  System is up and running. | | |
| Postconditions: | Users are granted access to their account.  Users are presented with their account dashboard. | | |
| Normal Flow: | User enters their registered email address and password on the login page.  System validates the user's credentials.  System grants the user access to their account.  System redirects the user to their account dashboard. | | |
| Alternative Flows: | None | | |
| Exceptions: | System displays an error message informing the user that their email address or password is incorrect and prompts them to try again.  System displays an error message informing the user that there was an error processing their login request and prompts them to try again later. | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times per day per user. | | |
| Business Rules: |  | | |
| Other Information: | None | | |
| Assumptions: | The system is running on a secure and reliable server. | | |

### 

### 2.4 Register - ReadMate

#### a. Register Account

#### 

| ID and Name: | **UC-005  *Register Account*** | | |
| --- | --- | --- | --- |
| Created By: | Duc Anh | Date Created: | 10/03/2023 |
| Primary Actor: | Readers, Author | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process of a user creating a new account in the system. The outcome of this use case is the creation of a new user account with the entered user information | | |
| Trigger: | User clicks on the login button on the system's login page. | | |
| Preconditions: | The user is not already registered on the system.  The user has access to a valid email address. | | |
| Postconditions: | A new user account is created with the entered information.  The user is logged into the system and taken to the home page. | | |
| Normal Flow: | 1. The user navigates to the registration page and clicks the "Register" button. 2. The system displays the registration form. 3. The user enters their desired username, email address, and password into the registration form. 4. The user clicks the "Submit" button. 5. The system validates the user's information and confirms that the email address is valid and has not been used to register another account. 6. The system creates a new user profile with the user's information. 7. The system redirects the user to the login page and displays a message indicating that their account has been created. | | |
| Alternative Flows: | None | | |
| Exceptions: | 1. System displays an error message informing the user that their email address or password is incorrect and prompts them to try again.  2. System displays an error message informing the user that there was an error processing their login request and prompts them to try again later. | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times per day per user. | | |
| Business Rules: | BR-06, BR-14, BR-15, BR-23 | | |
| Other Information: | The system must store user passwords securely and protect against brute force attacks.  The system must provide a "forgot password" feature to allow users to reset their passwords if needed.  If login fails due to a systemic reason such as loss of network connectivity, the system should display an error message informing the user that there was an error processing their request and prompt them to try again later.  If login results in a durable state change such as updating the user's last login timestamp in the database, the change should be completed correctly. | | |
| Assumptions: | The user has access to a valid email address. | | |

### 

#### b. Register for V.I.P account

#### 

#### 

| ID and Name: | **007 - Register for V.I.P account** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 10/03/23 |
| Primary Actor: | Readers | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process of registering for a V.I.P account in the app, which provides access to premium features such as exclusive content, early access to new releases, and personalised recommendations | | |
| Trigger: | User selects the "Register for V.I.P Account" option in the app | | |
| Preconditions: | The user must have an active internet connection.  The user must not already have a V.I.P account with the app.  The user must be willing to provide personal information such as name, email address, and payment information | | |
| Post-conditions: | The user has successfully registered for a V.I.P account.  The user's payment information has been securely stored in the app's database | | |
| Normal Flow: | The user selects the "Register for V.I.P Account" option in the app.  The app displays a form for the user to enter their personal information.  The user enters their name, email address, and payment information.  The app validates the user's payment information and displays a confirmation message.  The app stores the user's payment information securely in the app's database.  The app grants the user access to V.I.P features | | |
| Alternative Flows: | If the user's payment information is invalid, the app displays an error message and prompts the user to enter valid payment information.  If the user decides not to register for a V.I.P account, they can choose to cancel the registration process and return to the app's main menu. | | |
| Exceptions: | If the app encounters an error while processing the user's payment information, it will display an error message and prompt the user to try again later | | |
| Priority: | Medium | | |
| Frequency of Use: | This use case is expected to be used frequently by users who want to use full option on ReadMate | | |
| Business Rules: | BR-04, BR-04, BR-07, BR-16 | | |
| Other Information: | None | | |
| Assumptions: | The user is willing to provide personal information such as name, email address, and payment information.  The app is able to securely store the user's payment information.  The app is able to validate the user's payment information.  The user understands that V.I.P features are only available to registered V.I.P members | | |

#### 

#### c. Register as Author

| ID and Name: | **008 - Register as Author** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 11/03/23 |
| Primary Actor: | User | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process for a user to register as an author on the reading app. After registering, the user will be able to create and publish their own stories on the app | | |
| Trigger: | A user wants to register as an author on the ReadMate | | |
| Preconditions: | The user has access to the internet.  The user has not already registered as an author on the reading app.  The user has a valid email address | | |
| Post-conditions: | The user has successfully registered as an author on the reading app.  The user can access the author dashboard to manage their stories | | |
| Normal Flow: | The user selects the "Register as Author" option on the reading app.  The app prompts the user to enter their email address and password.  The user enters their email address and password and clicks on the "Register" button.  The app sends a verification link to the user's email address.  The user clicks on the verification link and is directed back to the app.  The app confirms the verification and prompts the user to fill out their author profile, including their name, bio, profile picture, and social media links.  The user fills out their author profile and clicks on the "Save" button.  The app confirms the user's registration as an author and redirects them to the author dashboard | | |
| Alternative Flows: | The user does not receive the verification email within a reasonable amount of time.  The user can request for the verification email to be resent.  The user checks their spam folder.  The user contacts the app's support team for assistance | | |
| Exceptions: | If the user's email address is already registered as an author, the app will display an error message and prompt the user to login instead.  If the user enters an invalid email address or password, the app will display an error message and prompt the user to try again | | |
| Priority: | Low | | |
| Frequency of Use: | This use case is expected to be used frequently by users who want to become author | | |
| Business Rules: | BR-04, BR-07, BR-16 | | |
| Other Information: | None | | |
| Assumptions: | The user has a valid email address that they can access.  The user has read and agreed to the terms and conditions for becoming an author on the reading app.  The user has the necessary rights and permissions to publish the content they create on the reading app | | |

#### 

#### 

### 2.5 Release - ReadMate

#### a. CRUD Book/Story (draft)

#### 

| ID and Name: | **009  *CRUD Book/Story*** | | |
| --- | --- | --- | --- |
| Created By: | Duc Anh | Date Created: |  |
| Primary Actor: | Author | Secondary Actor: | ReadMate System |
| Description: | This use case allows the author to perform CRUD (Create, Read, Update, Delete) operations on books/stories in the system. | | |
| Trigger: | Author selects the "Manage Books/Stories" option from the author dashboard. | | |
| Preconditions: | The author is logged in to the system.  The author has appropriate permissions to manage books/stories. | | |
| Postconditions: | For Create operation: a new book/story is sended to the Validator.  For Read operation: the author is presented with a list of books/stories in the system.  For Update operation: the updated book/story information is saved in the system.  For Delete operation: the selected book/story is removed from the system. | | |
| Normal Flow: | Author selects the "Manage Books/Stories" option from the author dashboard.  System presents a list of books/stories in the system.  Author selects the desired book/story from the list.  System presents the details of the selected book/story.  Author can choose to:  a. Edit the book/story information and save changes (Update operation).  b. Remove the book/story from the system (Delete operation).  If the author chooses to add a new book/story, they select the "Add New Book/Story" option.  Author enters the book/story information and saves it (Create operation). | | |
| Alternative Flows: | If there are no books/stories in the system, the system displays a message indicating there are no books/stories to manage. The author can choose to add a new book/story (step 6 of normal flow).  If the author chooses to cancel any changes they made during the update or create operations, the system discards the changes and returns to step 2 of the normal flow.  If the author enters invalid data during the create or update operation, the system displays an error message and returns to the corresponding step in the normal flow. | | |
| Exceptions: | If there is a database error while performing any of the CRUD operations, the system displays an error message and does not perform the operation. | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times per day. | | |
| Business Rules: | BR-01, BR-02, BR-03, BR-24, BR-25 | | |
| Other Information: | The system should be fast and responsive when performing CRUD operations on books/stories.  The Validator checks to ensure that the book/story information is valid.  If the use case execution fails for some unanticipated or systemic reason, the system should display an error message and provide a way for the administrator to try the operation again.  If the use case results in a durable state change in a database or the outside world, the change is completed correctly or rolled back in case of an exception. | | |
| Assumptions: | None | | |

### 

#### b. Validate Book/Story

#### 

| ID and Name: | **010 - Validate Book/Story** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 15/03/23 |
| Primary Actor: | Validator | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process of validating the content of a book/story to ensure that it meets the quality standards set by the platform. The validator is responsible for reviewing the book/story and either approving or rejecting it | | |
| Trigger: | A validator logs in and navigates to the "Validate Content" page | | |
| Preconditions: | The validator is logged into their account.  The book/story has been submitted for validation | | |
| Post-conditions: | The validator has approved or rejected the book/story.  The author of the book/story is notified of the validator's decision | | |
| Normal Flow: | The validator navigates to the "Validate Content" page.  The validator selects the book/story they wish to review.  The validator reads through the book/story and checks for any spelling, grammar, or formatting errors.  If the book/story meets the quality standards, the validator approves it and submits it for publishing.  If the book/story does not meet the quality standards, the validator rejects it and provides feedback to the author.  The author is notified of the validator's decision | | |
| Alternative Flows: | If the validator is unsure about the quality of the content, they can request a second opinion from another validator.  If the validator finds minor errors in the content, they can request that the author make corrections and resubmit the book/story for validation | | |
| Exceptions: | If the book/story contains offensive or inappropriate content, the validator must reject it and report it to the platform administrators | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | BR-08, BR-09 | | |
| Other Information: | None | | |
| Assumptions | None | | |

#### 

#### 

### 

#### c. Release, Reject Book/Story

#### 

| ID and Name: | **011- *Release, Reject Book/Story*** | | |
| --- | --- | --- | --- |
| Created By: | Duc Anh | Date Created: | 15/03/2023 |
| Primary Actor: | Validator | Secondary Actor: | System |
| Description: | This use case describes the process for a Validator to release or reject a book/story for publishing. The Validator is responsible for reviewing the book/story and deciding whether it is suitable for publication. If the book/story meets the necessary requirements, the Validator will release it for publishing. If the book/story fails to meet the necessary requirements, the Validator will reject it. | | |
| Trigger: | Validator has reviewed the book/story and needs to either approve or reject it. | | |
| Preconditions: | The Validator must be logged into the system  The Validator must have access to the book/story to be reviewed  The book/story must have been submitted for review by the Author | | |
| Postconditions: | If the Validator releases the book/story, it will be available for publishing  If the Validator rejects the book/story, it will not be available for publishing  and system will send a notification to the author of it | | |
| Normal Flow: | The Validator logs into the system  The Validator accesses the book/story to be reviewed  The Validator reads through the book/story to assess its quality and suitability for publishing  If the Validator deems the book/story suitable for publishing:  a. The Validator selects the "Release" option  b. The system notifies the Author that the book/story has been released for publishing  If the Validator deems the book/story unsuitable for publishing:  a. The Validator selects the "Reject" option  b. The system prompts the Validator to provide a reason for rejection  c. The Validator enters the reason for rejection  d. The system notifies the Author that the book/story has been rejected with the reason provided | | |
| Alternative Flows: | none | | |
| Exceptions: | The system displays an error message and prompts the Validator to try again later. | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times per day. | | |
| Business Rules: | BR-24, BR-25, BR-26, BR-30 | | |
| Other Information: | If the use case execution fails for some unanticipated or systemic reason, the system will display an error message and prompt the Validator to try again later.  If the book/story is rejected, it will not be deleted from the system, but it will not be available for publishing until the necessary revisions have been made.  If the book/story is released for publishing, the system will generate a unique identifier for it.  The system will store the reason for rejection in the database.  The system will track the status of the book/story (released or rejected) in the database. | | |
| Assumptions: | The system has already been developed and is functional.  The Validator has been trained to assess the quality and suitability of books/stories | | |

### 

### 2.6 Gift - ReadMate

#### a. Send gift (donate) for author

#### 

| ID and Name: | **014 - Send gift(donate) for author** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 15/03/23 |
| Primary Actor: | Readers | Secondary Actors: | Author |
| Description: | This use case describes the process of a user sending a gift (donation) to an author on the reading app. The user can choose to send a monetary donation to an author as a way of showing appreciation for their work | | |
| Trigger: | User wants to send a gift (donation) to an author | | |
| Preconditions: | Users must be logged into their account.  The author must have an account on the reading app and have enabled the gift (donation) feature | | |
| Post-conditions: | The author will receive the gift (donation) amount in their account on the reading app.  The user's payment information will be securely stored for future donations | | |
| Normal Flow: | User selects an author they wish to send a gift (donation) to.  User navigates to the author's profile page.  User selects the "Send Gift" button.  User selects the amount they wish to donate.  User enters their payment information (credit card, PayPal, etc.).  User confirms the payment amount and details.  Payment is processed and the author receives the gift (donation) amount in their account on the reading app.  User receives a confirmation message that the payment was successful | | |
| Alternative Flows: | User selects a pre-set donation amount.  User enters a custom donation amount | | |
| Exceptions: | If the user's payment is declined, the user will receive an error message and the transaction will not go through.  If the author has not enabled the gift (donation) feature, the user will not be able to send a gift (donation) to that author.  If the user's account is suspended or banned, they will not be able to send gifts (donations) on the reading app | | |
| Priority: | High | | |
| Frequency of Use: | This use case is expected to be used frequently by users who want donate fir their favourite author | | |
| Business Rules: | BR-08, BR-09 | | |
| Other Information: | None | | |
| Assumptions: | The user has a valid payment method and will not experience any issues when processing the payment.  The user is sending the gift (donation) out of appreciation for the author's work and not for any other reasons.  The author will receive the gift (donation) amount in a timely manner.  The reading app has a secure payment processing system in place to protect the user's payment information | | |

#### 

#### b. View gift information

| ID and Name: | **015 - View gift information** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 16/03/23 |
| Primary Actor: | Author | Secondary Actors: | ReadMate System |
| Description: | This use case describes the process of viewing the information of gifts/donations received by the author from readers. The information includes the sender's name, amount, date of donation, and any message attached | | |
| Trigger: | The author wants to view the information of the gifts/donations received | | |
| Preconditions: | The author must have a logged account on the platform.  The author must have received at least one gift/donation from a reader. | | |
| Post-conditions: | The author can view the information of the gift/donation received.  The information will be displayed on the author's account dashboard | | |
| Normal Flow: | The author logs in to their account on the platform.  The author navigates to the "Gift Information" tab on their account dashboard.  The system retrieves and displays a list of all the gifts/donations received by the author.  The author can view the sender's name, amount, date of donation, and any message attached to the gift/donation | | |
| Alternative Flows: | N/A | | |
| Exceptions: | If the author has not received any gifts/donations, the system displays a message informing them of the same.  If the author's account is inactive or blocked, they will not be able to view the gift information | | |
| Priority: | Medium | | |
| Frequency of Use: | Depends on the number of gifts/donations received by the author | | |
| Business Rules: | BR-08, BR-09, BR-11 | | |
| Other Information: | N/A | | |
| Assumptions: | The gifts/donations received by the author are legitimate and not fraudulent.  The system accurately stores and retrieves the gift/donation information | | |

#### 

### 2.7 Manage Account - ReadMate

#### a. Manage account break the rules

#### 

| ID and Name: | **019 - Manage account break the rules** | | |
| --- | --- | --- | --- |
| Created By: | Hoàn | Date Created: | 17/03/23 |
| Primary Actor: | Validator | Secondary Actors: | Readers, ReadMate System |
| Description: | This use case involves the platform's ability to manage user accounts that violate community guidelines. In this case, a user has posted a toxic comment, and the platform must take appropriate action to manage the user account | | |
| Trigger: | A user has posted a toxic comment that violates the platform's community guidelines more than 3 times. | | |
| Preconditions: | The platform must have a system in place to detect and flag toxic comments.  The user must have an account on the platform.  The platform must have clearly defined community guidelines outlining what is considered toxic behaviour | | |
| Post-conditions: | The user's comment has been removed from the platform.  The user's account has been flagged for toxic behaviour.  The user may be temporarily or permanently banned from the platform, depending on the severity of their behaviour | | |
| Normal Flow: | The platform detects a toxic comment posted by a user.  The platform removes the comment from the platform.  The platform flags the user's account for toxic behaviour.  The platform sends a warning to the user explaining the violation and the consequences of further toxic behaviour.  The user has the opportunity to appeal the warning and provide a justification for their behaviour.  If the user does not appeal or their appeal is unsuccessful, the platform may take additional actions such as temporarily or permanently banning the user from the platform | | |
| Alternative Flows: | If the toxic comment is particularly severe or violates multiple community guidelines, the platform may skip the warning and immediately ban the user from the platform.  If the user appeals the warning and provides a valid justification for their behaviour, the platform may remove the flag on their account and restore their ability to post on the platform. | | |
| Exceptions: | If the platform fails to detect the toxic comment or remove it from the platform, the toxic behaviour may continue unchecked.  If the platform mistakenly flags a user account for toxic behaviour, innocent users may be penalised unfairly | | |
| Priority: | High | | |
| Frequency of Use: | Depends on the frequency of toxic behaviour on the ReadMate | | |
| Business Rules: | BR-43, BR-44, BR-45 | | |
| Other Information: | The platform must have a clear process for reviewing and addressing user appeals.  The platform should have a clear escalation path for particularly severe or repeated instances of toxic behaviour.  The platform must be transparent with users about the consequences of toxic behaviour and the process for managing user accounts that violate community guidelines.  The platform should have measures in place to prevent users from creating multiple accounts to circumvent penalties for toxic behaviour | | |
| Assumptions: | The platform has a clear definition of what constitutes toxic behaviour and has communicated this to users through their community guidelines.  The platform has sufficient resources to review and address toxic behaviour in a timely manner.  Users will abide by the platform's community guidelines once they have been communicated clearly | | |

#### 

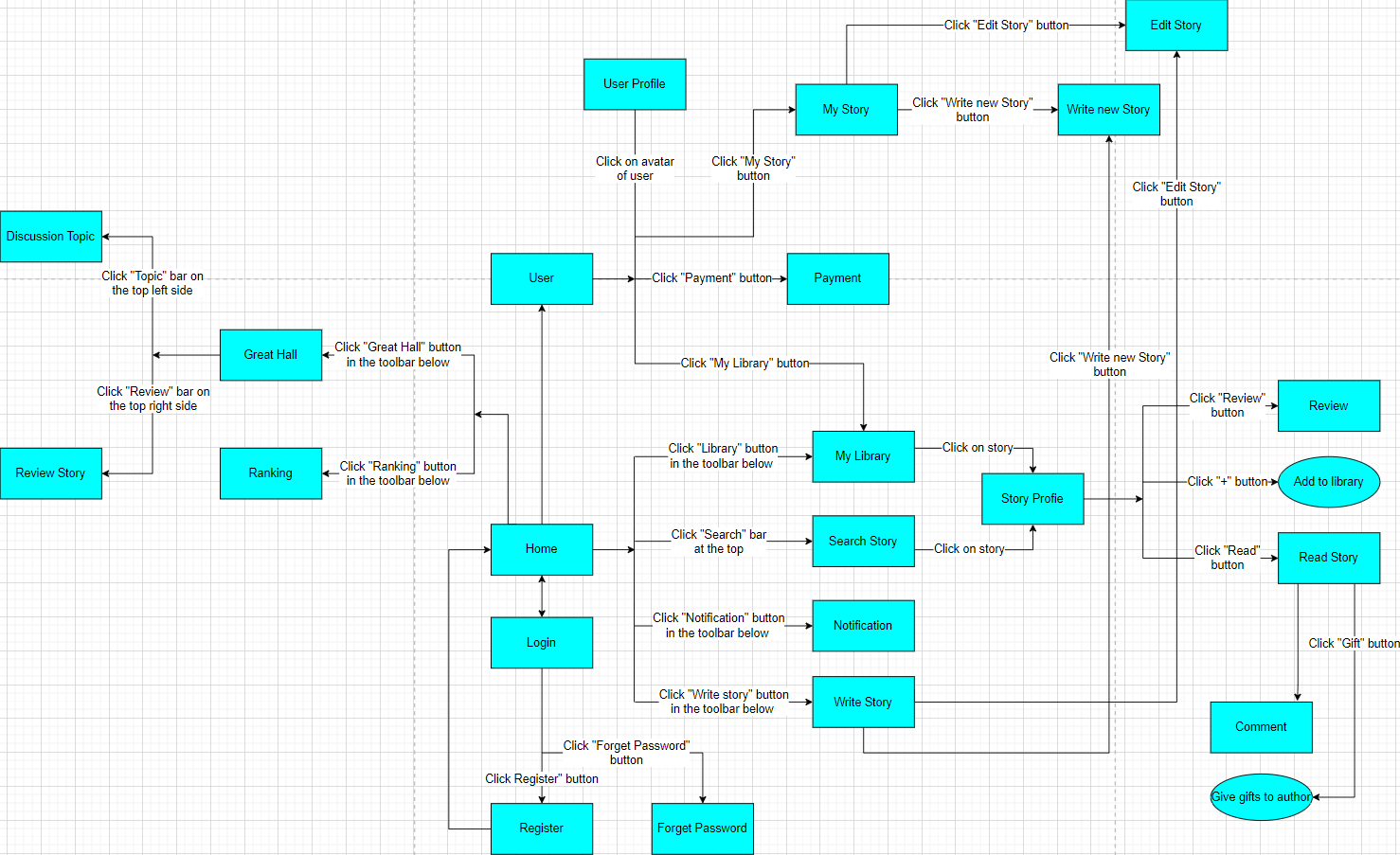
### 

#### 

## 3. Functional Requirements

### 3.1 System Functional Overview

#### a. Screen Flow



#### b. Screen Details

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Home | Home | Including nominated stories, new completed stories, updated stories, newly reviewed stories, high rating stories, highly read stories, highly commented stories,... |
| 2 | Login | Login | Users can log in with their ReadMate account, Google account, Facebook account,... |
| 3 | Register | Register | Users need to enter their username, email, display name, password and confirm password to register for a ReadMate account. |
| 4 | User | User | The “User” screen includes basic information about the user, links to the User Profile, Library, Payment, Setting, Content Policy, Privacy Policy, Terms of Use,... |
| 5 | Library | My Library | Allows users to add favourite stories to the library. Users can divide stories into many compartments according to personal preferences (can be set to private or public to other users). Besides, users can also keep track of the stories they have added to their library. |
| 6 | Searching story | Search Story | Users can search for stories by title, or search for stories by tags such as rating, status, genre, context,.... |
| 7 | Notification | Notification | Notify users of stories/authors they follow with new updates or have readers comment on their stories/give them gifts,... |
| 8 | List of user’s stories | My Story | Displays a list of stories that the user has written, allows users to add new stories, edit/delete stories. |
| 9 | Writing story | Write new Story | Provide a text editor that allows users to compose right on the ReadMate application. |
| 10 | Editing story | Edit Story | Allows users to edit the story profile, change the VIP price of each story chapter or edit the content of each story chapter. |
| 11 | Reading story | - Story Profile  - Read Story | - At the "Story profile" screen, users can read story information, reviews, summaries, table of contents and add stories to the library if they like. Besides, users will also be nominated for the same stories. author and story in the same genre.  - At the “Read Story” screen, users can read each story chapter in order, or select a story chapter to read in the table of contents. Users can change the font, font size, background colour and reading (vertical/horizontal scrolling). In addition, users can like chapters and bookmarks. |
| 12 | Payment | Payment | Allows users to deposit money into the application, this money can be used to buy VIP chapters or give gifts to the author. |
| 13 | Give gift to authors | Gift | Users can use the money paid into the application to buy gifts for their favourite authors. The application provides many gifts with different values. |
| 14 | Comment | Comment | Allow users to comment below each chapter |
| 15 | Discussion | Great Hall:  -Topic  - Review | “The Great Hall” includes discussion topics and reviews of stories, filtered by story genre. Users are free to discuss with others the topics or stories reviewed here. |
| 16 | Ranking stories | Ranking | Rank all stories by 3 criteria: genre, status and time period |
| 17 | Reset password when forgot password | Forgot Password | Users must enter an email address for the system to send an email with instructions to reset the account. |

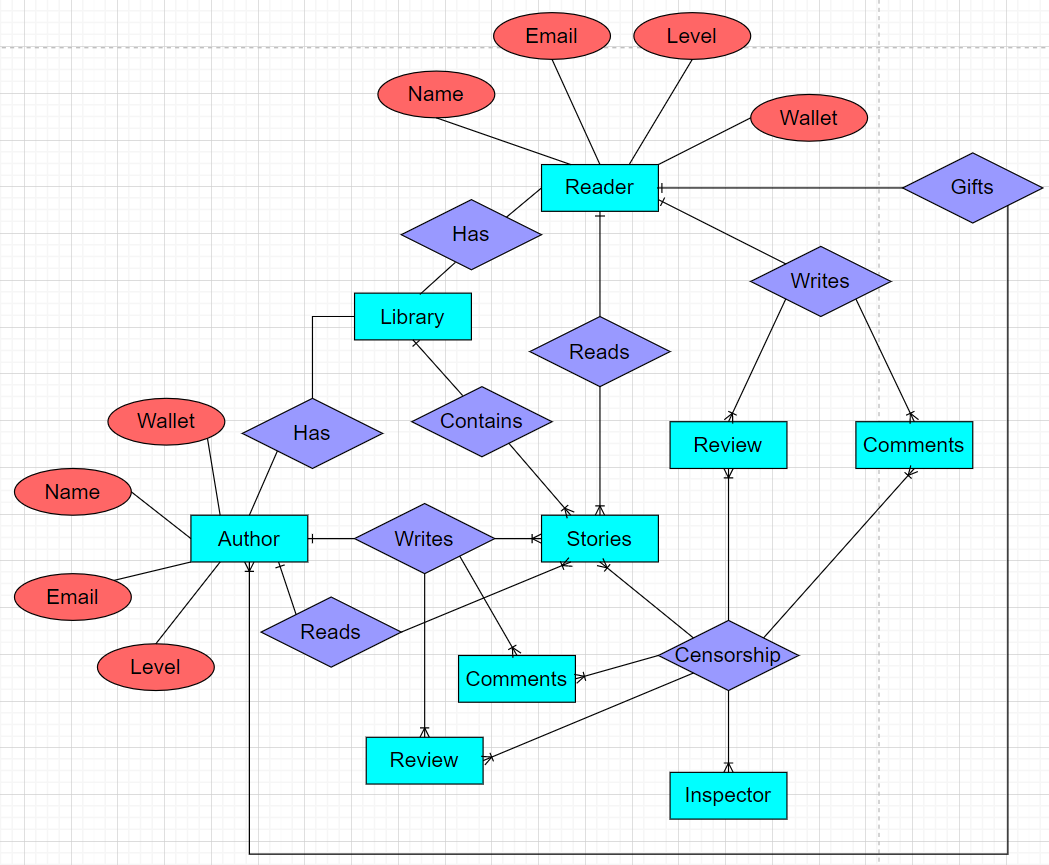
#### c. Screen Authorization

| **Screen** | **Readers** | **Author** | **Validator** |
| --- | --- | --- | --- |
| Home Page | X | X |  |
| Specific Book/Story | X | X |  |
| Login | X | X |  |
| Register | X | X |  |
| Register V.I.P Account | X | X |  |
| Add New Book/Story |  | X |  |
| Update Book/Story |  | X |  |
| Delete Book/Story |  | X |  |
| List Account Banned |  |  | X |
| Validate content for Book/Story |  |  | X |
| Validate content for Comment |  |  | X |
| Validate content for Review |  |  | X |
| Reset Password | X | X |  |
|  |  |  |  |

#### d. Non-Screen Functions

| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Recommend | Personalised Recommendations | A function that suggests books based on the user's reading history and preferences |
| 2 | Social | Social Sharing | A feature that allows users to share their favourite books and reading progress with friends and family on social media |
| 3 | Display Style | Customizable Themes | A feature that enables users to change the app's background colour, font size, and other visual settings for a personalised reading experience |
| 4 | Take Note | Highlighting and Note-taking | A tool that allows users to highlight text and take notes while reading, making it easier to remember important details |
| 5 | Challenge | Reading Challenges | A feature that encourages users to set goals and challenges for themselves, such as reading a certain number of books in a month or a year |

#### e. Entity Relationship Diagram



**Entities List**

| **#** | **Entity** | **Description** |
| --- | --- | --- |
| 1 | Author | Author includes name, email, level, wallet. Author reads Stories, writes Stories, writes Comments and Reviews. Author has a Library. |
| 2 | Reader | Reader includes name, email, level, wallet. Reader reads Stories, writes Comments and Reviews. Reader has a Library. |
| 3 | Inspector | Inspector censorship Stories, Comments and Reviews |
| 4 | Stories | Stories written by the Author, moderated by the Inspector. Stories are contained in the Library. Readers and Authors both read Stories. |
| 5 | Library | Reader and Author both have Libraries. The Library contains Stories. |
| 6 | Comments | Reader and Author both write Comments. Comments are moderated by Inspector. |
| 7 | Reviews | Reader and Author both write Reviews. Reviews are moderated by Inspector. |

### 3.2 Login

* *Function Name: Login*

*\_ Function Trigger: This function is activated when the user clicks the "Login" button on the Login screen or Login with Google/Facebook.*

*\_ Function Description:*

*Actors/Roles: The main actor for this function is the user who entered the correct username and password.*

*\_ Purpose: The purpose of this function is to require the user to log in to be able to use other functions of the application.*

*\_ Interface: The interface for this function includes the login screen*

*\_ Data Processing: Data processing related to this function includes authentication of account information.*

### 3.3 Home

* *Function Name: Home*

*\_ Function Trigger: This function is displayed by default when opening the app*

*\_ Function Description:*

*Actors/Roles: Default home page of the app*

*\_ Purpose: Display the list of nominated stories, new completed stories, updated stories, newly reviewed stories, high rating stories, highly read stories, highly commented stories,...*

*\_ Interface: The interface for this function includes the home screen*

*\_ Data Processing: The data processing associated with this function includes transferring data from the database.*

### 3.4 Library

* *Function Name: Library*

*Function Trigger: This function is activated when the user clicks the "My Library" button on the Home or User screen.*

*\_ Function Description:*

*Actors/Roles: List of favourite user stories*

*\_ Purpose: The purpose of this function is to allow users to follow and read their favourite stories*

*\_ Interface: The interface for this function includes the Library screen*

*\_ Data Processing: The data processing related to this function includes a database of stories.*

### 3.5 Searching story

* *Function Name: Searching story*

*Function Trigger: This function is activated when the user clicks on the search bar with the Home screen*

*\_ Function Description:*

*Actors/Roles: Search for stories by title, or search for stories by tags such as rating, status, genre, context,....*

*\_ Purpose: The purpose of this function is to allow users to find stories by interest*

*\_ Interface: The interface for this function includes the Search screen*

*\_ Data Processing: The data processing related to this function includes a database of stories, filtered by title, or search for stories by tags such as rating, status, genre, context,....*

### 3.6 Writing story

* *Function Name: Writing story*

*Function Trigger: This function is activated when the user clicks the Write New Story button in the Write Story and My Story screen*

*\_ Function Description:*

*Actors/Roles: The main actor for this functionality is the user's account which must be the author account*

*\_ Purpose: The purpose of this function is to allow users to compose stories right in the application.*

*\_ Interface: The interface for this function includes a professional text editor*

*\_ Data Processing: The data processing associated with this function includes adding data to the database of stories.*

### 3.7 Reading story

* *Function Name: Reading story*

*Function Trigger: This function is activated when the user clicks on a story*

*\_ Function Description:*

*\_ Purpose: The purpose of this function is to allow users to read stories*

*\_ Interface: The interface for this function includes the story profile screen and the screen of each chapter*

*\_ Data Processing: The data processing related to this function includes a database of stories.*

### 3.8 Payment

* *Function Name: Payment*

*Function Trigger: This function is activated when the user has a bank account or other types of e-wallets associated with the app.*

*\_ Function Description:*

*\_ Actor/Role: The main actor for this function is that the user must have a bank account or other e-wallets associated with the app*

*\_ Purpose: The purpose of this function is to allow users to recharge the app to pay for activities such as buying VIP story chapters, giving gifts to the author.*

*\_ Interface: The interface for this functionality includes both screens and non-screen components, such as backend processing of transaction data and communication with third-party payment processors where applicable .*

*\_ Data Processing: Data processing related to this function includes verifying transaction information, calculating any applicable taxes, saving transaction orders in the database and sending notifications via email. email to customers and appropriate parties within the organisation.*

### 3.9 Comment

* *Function Name: Comment*

*Function Trigger: This function is activated when the user clicks on the comment bar below the story profile or in each story chapter.*

*\_ Function Description:*

*\_ Actor/Role: The main actor for this function is that the user must be on the screen of that story or that chapter.*

*\_ Purpose: The purpose of this function is to allow users to comment on stories*

*\_ Interface: The interface for this function includes both the Story Profile screen and the Story Reading screen*

*\_ Data Processing: The data processing related to this function includes adding data to the database of stories*

### 3.10 Discussion

* *Function Name: Discussion*

*Function Trigger: This function is activated when the user clicks on the "Great Hall" button, where Topic and Review are included.*

*\_ Function Description:*

*\_ Purpose: The purpose of this function is to allow users to freely discuss with others the topics or stories reviewed here.*

*\_ Interface: The interface for this function includes the Great Hall screen*

*\_ Data Processing: The data processing related to this function includes a database of stories, filtered by story genre.*

## 4. Non-Functional Requirements

### 4.1 External Interfaces

The user interface should follow standard design principles and include easy-to-use navigation, clear labelling, and consistent layout. It should also be designed to be compatible with different devices and screen sizes. A separate user interface specification document should be created to provide more detailed information about the design and functionality of each interfac*e*

#### a. User Interfaces

UI-1: Home Screen - This screen serves as the app's main interface and allows users to browse and select books to read. It may include elements such as book covers, categories, search bar, and menu buttons.

UI-2: Reading Screen - This interface is used when the user opens a book to read. It may include elements such as the book title, author, progress bar, bookmark button, and navigation buttons.

UI-3 Settings Screen - This interface allows users to customise the app's settings, such as font size, background colour, and reading preferences.

UI-4 Library Screen - This interface provides users with a list of their downloaded and purchased books. It may include elements such as book covers, title, author, and options to delete or sort the library.

UI-5 Purchase Screen - This interface allows users to purchase new books within the app. It may include elements such as book covers, prices, reviews, and payment options.

#### b. Software Interfaces

SI-1: Database - The reading app will need to interface with a database to store information about books, user accounts, and reading progress. The database will need to be able to handle high volumes of data and provide fast search and retrieval.

SI-2: Payment Gateway - To facilitate in-app purchases, the reading app will need to interface with a payment gateway. The payment gateway should support a variety of payment methods and be secure and reliable.

SI-3: Email Service - The reading app may interface with an email service to send users notifications and updates. The email service should be secure and reliable, with high delivery rates and low latency.

#### c. Hardware Interfaces

no definition

#### d. Communications Interfaces

CI-1: The COS shall send an email or text message (based on user account settings) to the Patron to confirm acceptance of an order, price, and delivery instructions.

CI-2: The COS shall send an email or text message (based on user account settings) to the Patron to report any problems with the meal order or delivery.

### 4.2 Quality Attributes

#### a. Usability

*Training time: This requirement specifies how long it should take for a typical user and a power user to become proficient in using the system. For example, a requirement may specify that a typical user should be able to complete basic tasks within 1 hour of training, while a power user should be able to complete advanced tasks within 2 hours of training.*

*Task completion time: This requirement specifies measurable task times for typical tasks that users will perform with the system. For example, a requirement may specify that a user should be able to complete a task such as creating a new document in under 2 minutes.*

*Usability standards: This requirement specifies that the system should conform to common usability standards, such as IBM's CUA Standards or Microsoft's GUI Standards. These standards provide guidelines on how user interfaces should be designed to ensure that they are intuitive and easy to use.*

*User feedback: This requirement specifies that the system should provide clear and concise feedback to the user when an action is completed, or an error occurs. The feedback should be designed to help the user understand what has happened and what they need to do next.*

*Navigation: This requirement specifies that the system should be easy to navigate, with clear and intuitive menus, buttons, and other controls. Users should be able to quickly find the features and functions they need and move between them with ease.*

#### b. Reliability

*Availability: This requirement specifies the percentage of time the system should be available for use, as well as any maintenance windows or degraded mode operations. For example, a requirement may specify that the system should be available 99.9% of the time during business hours.*

*Mean Time Between Failures (MTBF): This requirement specifies how long the system should operate before experiencing a failure. MTBF is usually specified in hours but can also be specified in days, months, or years. For example, a requirement may specify that the system should have an MTBF of 10,000 hours.*

*Mean Time To Repair (MTTR): This requirement specifies how long it should take to repair the system once a failure has occurred. For example, a requirement may specify that the system should have an MTTR of no more than 4 hours.*

*Accuracy: This requirement specifies the precision and accuracy required in the system's output according to a known standard. For example, a requirement may specify that the system should provide output with a resolution of 0.01 units and an accuracy of +/- 1%.*

*Maximum Bugs or Defect Rate: This requirement specifies the maximum number of bugs or defects allowed in the system and can be expressed in terms of bugs per thousand lines of code or bugs per function point. For example, a requirement may specify that the system should have a maximum defect rate of 0.1 bugs/KLOC.*

*Bug or Defect Rate: This requirement categorises bugs or defects in terms of minor, significant, and critical bugs, and defines what is meant by a critical bug. For example, a requirement may specify that a critical bug is one that causes complete loss of data or prevents access to certain parts of the system's functionality.*

#### c. Performance

*Response time: This requirement specifies the average and maximum response times for transactions in the system. For example, a requirement may specify that the system should have an average response time of 2 seconds and a maximum response time of 5 seconds.*

*Throughput: This requirement specifies the number of transactions or events that the system can process per second. For example, a requirement may specify that the system should be able to handle 1,000 transactions per second.*

*Capacity: This requirement specifies the maximum number of customers or transactions that the system can accommodate at any given time. For example, a requirement may specify that the system should be able to handle 10,000 simultaneous users.*

*Degradation modes: This requirement specifies acceptable modes of operation when the system has been degraded or is experiencing issues. For example, a requirement may specify that the system should continue to operate in a limited capacity even if certain components fail.*

*Resource utilisation: This requirement specifies the acceptable levels of resource utilisation, such as memory, disk, and communications. For example, a requirement may specify that the system should not use more than 80% of available memory or network bandwidth.*

#### d. Dependability

*Dependability is a critical aspect of software engineering, and it encompasses various characteristics that ensure the software's reliability, security, and safety. For instance, reliable software should be able to function correctly and consistently under different operating conditions. This means that it should be free from errors, bugs, and other faults that can cause system failure or unexpected behaviour.*

*On the other hand, secure software should protect against unauthorised access, malicious attacks, and other cyber threats that can compromise the integrity, confidentiality, and availability of data and resources. This involves implementing robust security measures such as authentication, access control, encryption, and intrusion detection/prevention mechanisms.*

*Similarly, safe software should not pose any physical or economic risk to users, operators, or the environment. This means that it should adhere to relevant safety standards and regulations and avoid any hazardous situations that could result in injury, damage, or loss.*

##### d1. Security

*Access control: This requirement specifies who can access the system and what actions they can perform. For example, a requirement may specify that only authorised personnel with valid credentials can log into the system and that they only have access to the data and functions that are necessary for their job.*

*Data protection: This requirement specifies how sensitive data is protected from unauthorised access or use. For example, a requirement may specify that all user data must be encrypted both in transit and at rest, and that data backups must be stored in secure offsite locations.*

*Authentication and authorization: This requirement specifies how users are authenticated and authorised to access the system. For example, a requirement may specify that multi-factor authentication is required for all users accessing the system remotely.*

*Security policies and regulations: This requirement specifies any security policies or regulations that the system must conform to. For example, a requirement may specify that the system must comply with the General Data Protection Regulation (GDPR) or Payment Card Industry Data Security Standards (PCI DSS).*

*Physical security: This requirement specifies how physical access to the system is controlled and monitored. For example, a requirement may specify that the server room must be secured with biometric locks and monitored by security cameras 24/7.*

##### d2. Safety

*Hazardous conditions: This requirement specifies any hazardous conditions that may arise from the use of the product and how they should be avoided or mitigated. For example, a requirement may specify that the system must have built-in safety features to prevent electrical shocks or overheating.*

*Emergency procedures: This requirement specifies the emergency procedures to be followed in the event of a system failure or other incident. For example, a requirement may specify that the system must have an emergency shutdown procedure that can be activated in case of a catastrophic failure.*

*Safety certifications and regulations: This requirement specifies any safety certifications or regulatory compliance requirements that the system must meet. For example, a requirement may specify that the system must comply with Occupational Safety and Health Administration (OSHA) regulations.*

*User training: This requirement specifies the training that users must receive before operating the system, including any safety-related training. For example, a requirement may specify that all users of the system must receive training on how to use the system safely and effectively.*

*System maintenance: This requirement specifies the maintenance procedures required to ensure that the system remains safe and reliable over time. For example, a requirement may specify that the system must be inspected and maintained regularly by qualified personnel.*

#### e. Supportability

*Coding standards: This requirement specifies the coding standards that must be followed by developers when writing code for the system. For example, a requirement may specify that all code should be written in accordance with the C# coding standard.*

*Naming conventions: This requirement specifies the naming conventions to be used for variables, functions, and other elements of the system. For example, a requirement may specify that all function names should begin with a verb and use camel case notation.*

*Class libraries: This requirement specifies any class libraries or software components that should be used in the system. For example, a requirement may specify that the system should make use of the Microsoft .NET Framework class library.*

*Maintenance access: This requirement specifies the level of access required for maintenance personnel to perform updates and modifications to the system. For example, a requirement may specify that maintenance personnel should have root-level access to the server hosting the system.*

*Maintenance utilities: This requirement specifies the maintenance utilities that should be provided with the system to aid in maintenance tasks. For example, a requirement may specify that the system should include a backup and restore utility for data files.*

### 