Software Requirements Specification

for

OpenRead Web-Application

**Version 4.0**

**Prepared by**

**Anthony Gilrandy Theo- 2201734814**

**Dewi Puspita Tanurezal - 2201740546**

**Devita - 2201755131**

**Steven - 2201852132**

**Bina Nusantara University**

**2020**

Table of Contents

Table of Contents 2

Revision History 3

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.2.1 Visitor Use Case 3

2.2.2 Reader Use Case 4

2.2.3 Writer Use Case 6

2.3 User Classes and Characteristics 7

2.4 Operating Environment 8

2.5 User Documentations 8

2.6 Non-function requirements 8

3. External Interface Requirements 9

3.1 User Interfaces 9

3.2 Hardware Interfaces 10

3.3 Software Interfaces 11

3.4 Communications Interfaces 11

4. System Features 11

4.1 Register Account 11

4.2 Login 12

4.3 Search Story 12

4.4 Write Comments 12

4.5 Update Comments 13

4.6 Write Story 13

4.7 Edit Story 14

5. Other Nonfunctional Requirements 15

5.1 Performance Requirements 15

5.2 Safety Requirements 15

5.3 Security Requirements 15

5.4 Software Quality Attributes 15

5.5 Business Rules 15

6. Other Requirements 16

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Steven | 13-10-2020 | Added Product Function (2.2) | 1.1 |
| Steven | 14-10-2020 | Added System Features (Chapter 4) | 1.2 |
| Steven | 27-11-2020 | Revised System Features (4.10 & 4.11)   * Takedown Story * Ban User   Added New System Features (4.1 & 4.2)   * Register Account * Login | 2.0 |
| Steven | 29-11-2020 | Revised User Interfaces (3.1)   * Login Account * Side Menu   Added Nonfunctional Requirements (Chapter 5) | 3.0 |
| Steven | 11-12-2020 | Revised ERD   * Added Entity Rating * Added Entity Story\_Genre * Added Entity Genre * Minor changes to attribute’s name | 3.1 |
| Steven | 13-12-2020 | Revised ERD   * Removed Entity Complaint * Removed Entity Admin   Removed Admin Management  Removed Admin Use Case  Removed System Feature related to admin (4.9, 4.10, 4.11) | 4.0 |

# Introduction

## Purpose

The purpose of this document is to build a web-application for people to share and read stories.

## Document Conventions

|  |  |
| --- | --- |
| *DB* | *Database* |
| *ER* | *Entity Relationship* |

## Intended Audience and Reading Suggestions

This project is a prototype for storytelling website and it is restricted within the university premises. This project are made under the guidance of university professors. This project is useful for developer who want to make website storytelling and as well as the user of the website.

## Product Scope

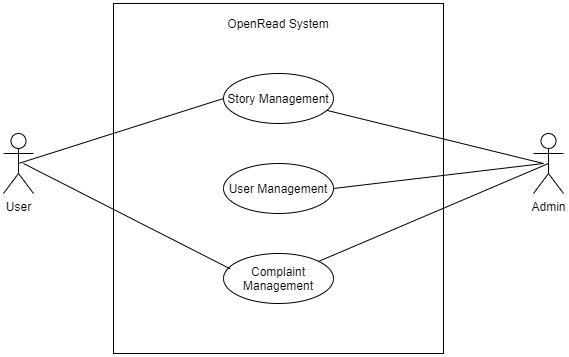
The purpose of OpenRead is that we provide a platform for people to read and share stories. We will have a database server that support to store writer’s stories as well as reader’s thoughts. Above all, We aim to create a community around stories for both amateur and established writers.

## References

* IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
* Roger S. Pressman. (2015). Software engineering: a practitioners approach. 08. McGraw - Hill Higher Education. NY.
* Ian Sommerville. (2016). Software Engineering TENTH edition. Pearson
* https://www.cse.msu.edu/~chengb/RE-491/Papers/SRS-BECS-2007.pdf

# Overall Description

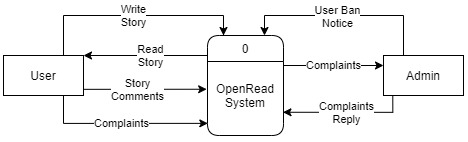
## Product Perspective



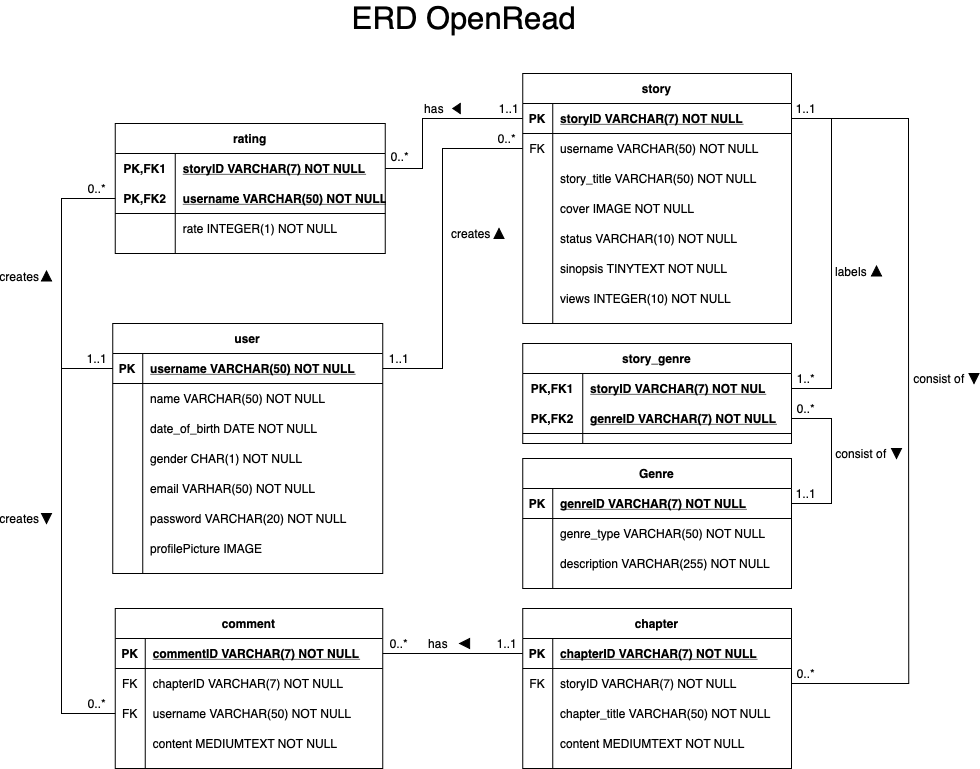
OpenRead is a platform as a website for amateur or professional writers to create stories that can be read by readers without the need to go through a publisher. Readers can search stories by genre and title to find whatever story they want to read. Writers can create stories, determine its genre and divide it into several chapters.

## Product Functions

Data Flow Context Diagram:



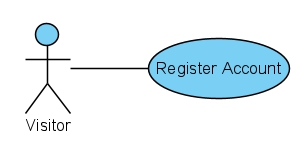
Entity-Relationship model :



### Visitor Use Case

**Use case : Register Account**

**Diagram:**



**Brief Description:**

The user will register their information and create their account. This action can be done on register page.

**Initial Step-by-Step Description:**

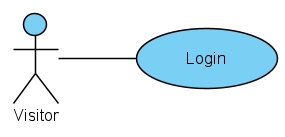
Before this use case can be initiated, the user has already accessed the register page.

1. The visitor inputs their username, email, password, and date of birth.
2. The visitor presses Sign Up button.
3. If the register is successful, the visitor will be logged in and redirected to the home page.
4. If one of the inputs is failed to be validated, the visitor must correct one of the fields before the account can be created.

**Xref:** Section 4.1, Register Account

**Use case : Login**

**Diagram:**



**Brief Description:**

The visitor can login based on the user credentials that he/she has registered. After he/she has login, his/her status will change to User.

**Initial Step-By-Step Description**

Before this use case can be initiated, the reader has already register and his/her account details is recorded in the database.

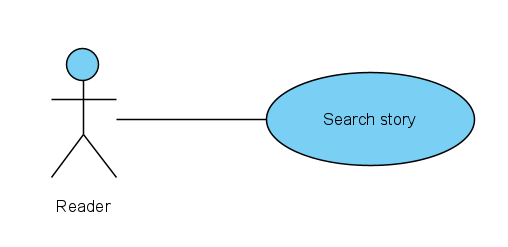
1. The visitor enters his/her username/emails and passwords
2. The system will validate whether the credentials match.
3. The visitor logins successfully.
4. The status of the visitor change to User.

**Xref:** Section 4.2, Login

### Reader Use Case

**Use case: Search story**

**Diagram:**

****

**Brief description:**

The reader searches a story they want to read. They can search by the title of the story or by the author name.

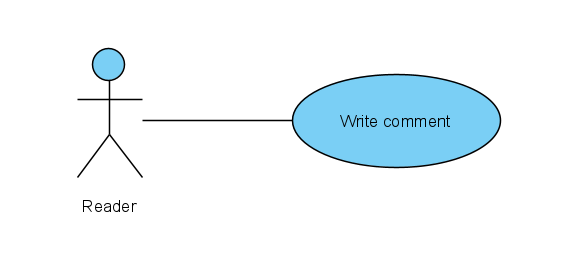
**Initial step-by-step description:**

Before this use case initiated, the reader has accessed the home page.

1. The reader clicks the search button at the top of the website home page.
2. The reader writes down the name of the story or the author name specifically.
3. The system will find the story or author name related to what the reader search for.
4. If there are any story or author name related, the system will show the story. If there are not, the system will write “The story or author name not found” and show another story available.

**Xref:** Section 4.3, Search Story

**Use Case: Write Comments**

**Diagram:**

**Brief Description:**

The reader can comment his/her thoughts from other people’s stories, this action can be done on the comment section.

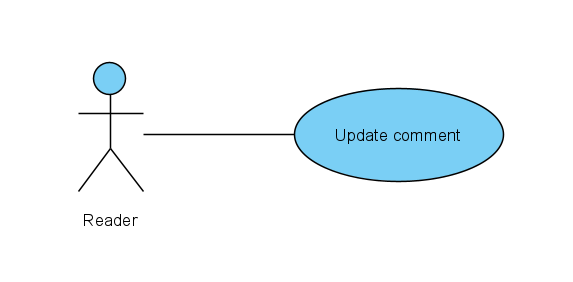
**Initial Step-By-Step Description:**

Before this use case can be initiated, the reader has already accessed a story page.

1. The reader selects add button near comment section.
2. The reader types his/her comments.
3. The reader posts.
4. The system asks for the reader to review his/her comments to make sure it doesn’t contain any inappropriate content.
5. The reader chooses to continue/review.
6. The system reads the comment and posts it at the comment section.

**Xref:** Section 4.4, Write Comments

**Use Case: Update Comments**

**Diagram:**

**Brief Description:**

The reader can edit his/her comment that he/she posted; this action can be done on the comment section.

**Initial Step-By-Step Description:**

Before this use case can be initiated, the reader has already access a story page and posted a comment at the exact same story page.

1. The reader selects the pencil button of the comment he/she wishes to update.
2. The reader updates his/her comments.
3. The reader posts.
4. The system asks for the reader to review his/her comments to make sure it doesn’t contain any inappropriate content.
5. The reader chooses to continue/review.
6. The system reads the comment and renews it at the comment section.

**Xref:** Section 4.5, Update Comment

### Writer Use Case

**Use Case: Write Story**

**Diagram**

Diagram

Description automatically generated

**Brief Description:**

The writer will create story they want to write. This action can be done on writer story page.

**Initial Step-by-Step Description:**

Before this use case can be initiated, the writer has already accessed the writer story page.

1. The writer selects the create story button.
2. The writer enters story title and genre.
3. After entering title and genre, system automatically adds one new chapter to the story.
4. Writer enters title for the chapter and its content.
5. The writer reviews the story title, chosen genre, chapter and its content.
6. The writer chooses to publish the story.
7. The system publishes to story on to the website.

**Xref:** Section 4.7, Write Story

**Use Case: Edit Story**

**Diagram**

Diagram

Description automatically generated

**Brief Description:**

The writer will edit story they have created. This action can be done on writer story page.

**Initial Step-by-Step Description:**

Before this use case can be initiated, the writer has already accessed the writer story page and has created at least one story.

1. The writer selects the edit story on the story he/she wants to edit.
2. The writer updates title or genre of the story.
3. The writer chooses a chapter he/she wants to edit.
4. The writer updates title or content of the chapter.
5. The writer reviews the story title, chosen genre, chapter and its content.
6. The writer chooses to accept the story’s edit.
7. The system republishes the story on to the website.

**Xref:** Section 4.8, Edit Story

## User Classes and Characteristics

There are 2 basic users – User

* User is both the reader and the writer. They are allowed to read stories as well as uploading stories for people to read. The user should be able to do the following functions:

1. Upload stories
2. Read stories
3. Comment on stories

## Operating Environment

The software will operate with all modern operating systems that has access to a browser (google chrome, firefox, safari, internet explorer), while the server will be operating under the Linux operating systems.

## User Documentation

For user documentation and information, head to section 4 (System Features) to understand further about the system. There will also be a user manual attached.

## Non-functional requirements

* The website must be compatible with google chrome, firefox, and internet explorer.
* The database’s query will run on the database administrator’s PC and will contain an SQL database. SQL Server Management Studio is already installed on the PC and it has a Windows operating system as well.

# External Interface Requirements

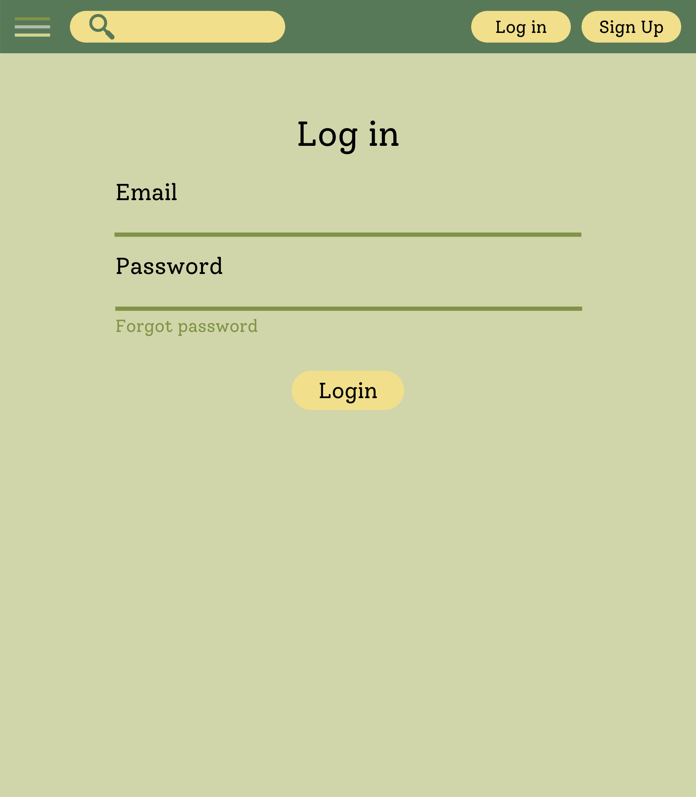
## User Interfaces

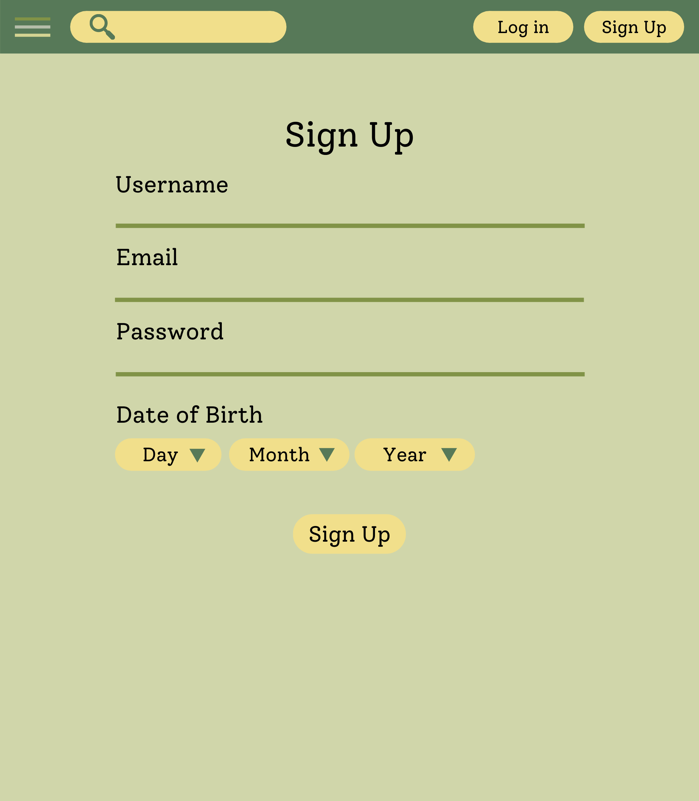
Welcome Page Side Menu

**

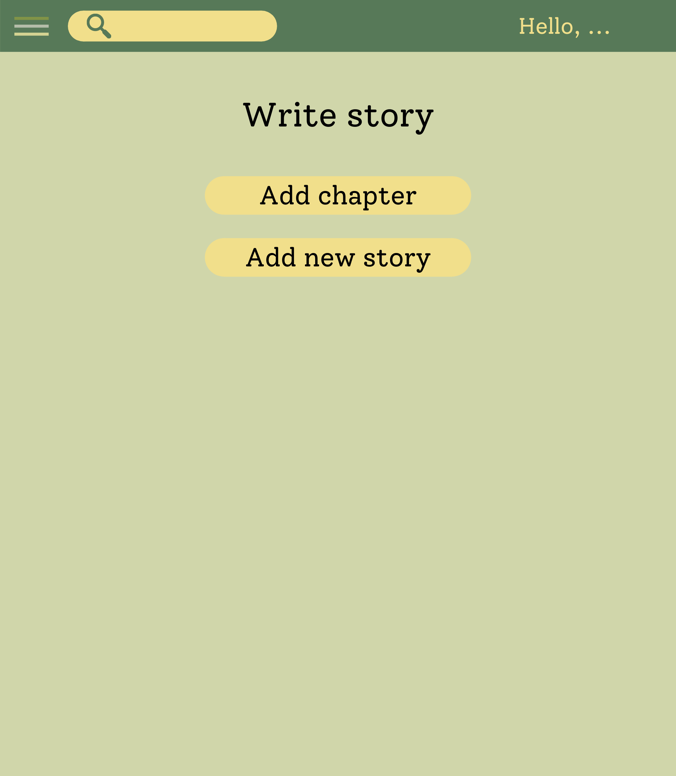
**

Sign Up Page Log In Page

**

**

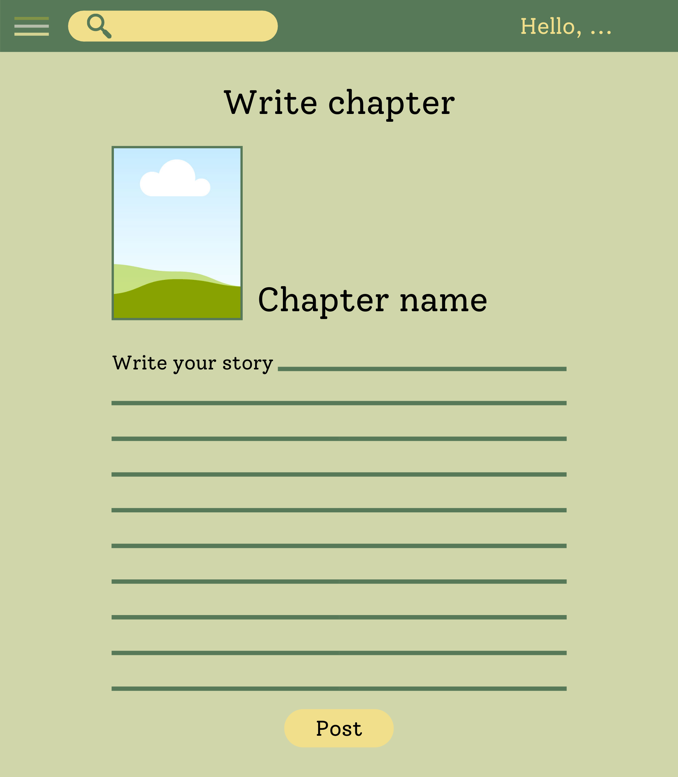
Read Story Page Write Story Page

**

**

Write Chapter Page Write New Story Page

**

**

## Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

This website supports all types of web browsers. We are using simple electronic forms for writing story, writing comment, reading story, etc.

* 1. **Register Account**

|  |  |
| --- | --- |
| **Use Case Name** | **Register Account** |
| **XRef** | Section 2.2.1, Register Account |
| **Trigger** | The user presses “register” button at the navbar or opens register page. |
| **Preconditions** | 1. The user must not be logged in 2. There is no existing username or email on the database |
| **Basic Paths** | 1. The visitor inputs their username, email, password, and date of birth. 2. The visitor presses Sign Up button. 3. If the register is successful, the visitor will be logged in and redirected to the home page. |
| **Alternative Paths** | – |
| **Postconditions** | The visitor’s account is created. |
| **Exception Paths** | If there is an existing account with same username input or email input, the register will fail and the visitor must input another username or email. |
| **Other** | None |

* 1. **Login**

|  |  |
| --- | --- |
| **Use Case Name** | **Login** |
| **XRef** | Section 2.2.1, Login |
| **Trigger** | The visitor selects the login button |
| **Preconditions** | 1. The visitor has registered. |
| **Basic Paths** | 1. The visitor enters his/her username/emails and passwords 2. The system will validate whether the credentials match. |
| **Alternative Paths** | - |
| **Postconditions** | Visitor is logged in and status changed to User |
| **Exception Paths** | If the visitor enters incorrect password or the username/email he/she enters didn’t exist in the database, the login process failed and the system will ask the visitor to enter again his/her credentials. |
| **Other** | None |

## Search Story

|  |  |
| --- | --- |
| **Use case name** | **Search story** |
| **Xref** | Section 2.2.2, Search Story |
| **Trigger** | The reader clicks the search button |
| **Preconditions** | The reader has accessed the website home page |
| **Basic paths** | 1. The reader writes down the name of the story or the author name specifically 2. The system will find the story or author name related to what the reader search for. 3. If there are any story or author name related, the system will show the story. If there aren’t the system will write “The story or author name not found” and show another story available |
| **Alternative paths** | None |
| **Postconditions** | Show the story |
| **Exception paths** | The reader can abandon the operation at any time |
| **Other** | None |

## Write Comments

|  |  |
| --- | --- |
| **Use Case Name** | **Write Comment** |
| **XRef** | Section 2.2.2, Write Comments |
| **Trigger** | The reader selects add button |
| **Preconditions** | The reader is on a story page and the reader had logged in. |
| **Basic Paths** | 1. The reader types his/her comments. 2. The reader posts. 3. The system asks for the reader to review his/her comments to make sure it doesn’t contain any inappropriate content. 4. The reader chooses to continue/review. 5. The system reads the comment and posts it at the comment section. |
| **Alternative Paths** | In step 4, if the reader chooses to review, the reader can edit his/her comments and goes back to step 2. |
| **Postconditions** | Comment is posted. |
| **Exception Paths** | The reader can abandon writing comments anytime. |
| **Other** | None |

## Update Comments

|  |  |
| --- | --- |
| **Use Case Name** | **Update Comment** |
| **XRef** | Section 2.2.2, Update Comment |
| **Trigger** | The reader selects pencil button |
| **Preconditions** | The reader is on a story page and the reader had logged in. |
| **Basic Paths** | 1. The reader updates his/her comments. 2. The reader posts. 3. The system asks for the reader to review his/her comments to make sure it doesn’t contain any inappropriate content. 4. The reader chooses to continue/review. 5. The system reads the comment and renews it at the comment section. |
| **Alternative Paths** | In step 4, if the reader chooses to review, the reader can edit his/her comments and goes back to step 2. |
| **Postconditions** | Comment is edited and renewed. |
| **Exception Paths** | The reader can abandon updating comments anytime. |
| **Other** | None |

## Write Story

|  |  |
| --- | --- |
| **Use Case Name** | **Write Story** |
| **XRef** | Section 2.2.3, Write Story |
| **Trigger** | The writer selects create story button |
| **Preconditions** | The writer is on the create story page and writer has logged in |
| **Basic Paths** | 1. The writer selects the create story button 2. The writer enters story title and genre 3. After entering title and genre, system automatically adds one new chapter to the story. 4. Writer enters title for the chapter and its content. 5. The writer reviews the story title, chosen genre, chapter and its content. 6. The writer chooses to publish the story. 7. The system publishes to story on to the website. |
| **Alternative Paths** | In step 5, if the writer chooses to review the story’s information, the writer can edit his/her decision and goes back to step 2.  In step 5, if the writer chooses to review the chapter’s information or content, the writer can edit his/her decision and goes back to step 4. |
| **Postconditions** | Story is published on the website |
| **Exception Paths** | The writer can cancel story creating at any time |
| **Other** | None |

## Edit Story

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Story** |
| **XRef** | Section 2.2.3, Edit Story |
| **Trigger** | The writer selects create story button |
| **Preconditions** | The writer is on the create story page and writer has logged in |
| **Basic Paths** | 1. The writer selects the edit story on the story he/she wants to edit. 2. The writer updates title or genre of the story. 3. The writer chooses a chapter he/she wants to edit. 4. The writer updates title or content of the chapter. 5. The writer reviews the story title, chosen genre, chapter and its content. 6. The writer chooses to accept the story’s edit. 7. The system republishes the story on to the website. |
| **Alternative Paths** | In step 3, if the writer chooses to create a new chapter into the story, the system will provide a new chapter form.   1. Writer enters title for the chapter and its content. Goes to step 5.   In step 5, if the writer chooses to review the story’s information, the writer can edit his/her decision and goes back to step 2.  In step 5, if the writer chooses to review the chapter’s information or content, the writer can edit his/her decision and goes back to step 4. |
| **Postconditions** | Story is republished on the website |
| **Exception Paths** | The writer can cancel story editing at any time |

# Other Nonfunctional Requirements

## Performance Requirements

* OpenRead can be accessed via browser with respond time less than 5 second.
* The database will be gone through the normalization process so there are no redundant data.
* The application is designed to handle lots of user simultaneously.
* The application can process records up to 100 for every transaction.

## Safety Requirements

* There will be regular database backups to ensure that user data, both stories and others, are safe.
* When there is an error in inserting, updating, and deleting transactions, we can rollback.
* Implement an exception handling system so as not to break the application when an error occurs.

## Security Requirements

* User’s data will be stored properly and classified, especially for password data will be encrypted
* Provide an information statement to the user, that this system is based on an online / website that is safe, secure, reliable stores and guarantees member data will be safe
* Each feature can be used by users according to their privileges
* User input data will be validated and sanitized before being entered into the database.

## Software Quality Attributes

* Correctness: the application is made according to the needs where this application can be a container for expressing user ideas in the form of written works.
* Efficiency: Application efficient in terms of coding and efficient in terms of operation by the user
* Integrity: The application has an integrated security system to prevent unauthorized abuse.
* Flexibility: Applications can be modified to suit the user's wishes in the future.
* Maintainability: The application is easy to do maintenance or changes if there is a problem with the application.
* Reliability: the application can be used without failure.
* Testability: Applications have been tested to ensure that they meet the requirements
* Usability: the application can be used easily because the appearance of the application is neat and simple.

## Business Rules

To make the OpenRead application easier and more convenient to use, we will update the system according to user suggestions.

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*