## chapter 3 ##

# 3 SRS

## 3.1 Overview

Reading has always been the simplest way to derive knowledge, seek advice or entertainment from a source. And today, with all the technological progress around, it’s one of the easiest things we can do to get yourself informed. We don’t need to wait months before we get that amazing book we really wanted in the morning mail, because we have e-books. Although books are great in general. E-books are also incredibly convenient. They are light and they fit in your pocket. You can bring thousands of pages with you everywhere with just logging into a website. They don’t tear apart and you can read them in the dark. What’s more, you don’t need to have a dedicated e-reader to read an e-book, you can do so from your laptop and smartphone. Online communities and socialmedia has mad tremendous hits in the last decades, people of all ages now spend so much time committing to websites to communicate and get in touch. A really big portion of those numerous folks are actually readers and want to share their reading experience. There comes ##app name ## to play it’s role which is gathering these segment in this platform. The website also provides books for it’s subscribers to download. Many other features are included one of them is that two subscribers can exchange their books. A blogging system is one of the features ## app name ## provides where subscribers can write their blogs and other subscribers can interact upon them. If you love reading and can’t get enough of it ## app name ## is your place to find books and share your experiences and interact with other readers.

## 3.2 Business Context

## app name ## system plays a role in books market where publishers post their books in ## app name ## platform and the subscribers of ## app name ## can purchase their favorite books. Advertisement system takes part in ## app name ## where Advertisers can post their ads in order to promote them for readers so they can reach out for them.

## 3.3 General Description

### 3.3.1 Product Functions

Person Actor Functions:

Register : user can create new account by entering his information (full name, username, password).

Login : user can login to the application by a valid username and password

Logout : user can logout from the application

edit profile : user can edit his profile information

search authors : user can search authors

shop : user can access a shop to discover books

Admin Actor Functions:

Manage groups : admin can add, modify delete groups and dedicate the appropriate authentication and authorization permissions

Manage users : admin can add, modify delete users of all types (reader, writer, bookstorer)

Manage blogs: admin can add, modify delete blogs

Manage Books: admin can add, modify delete books

Writer Actor Functions:

write post : writer can add a new blog post by filling add post form which includes (title, content)

Manage post : writer can edit or delete posts he wrote before

Book Storer Actor Functions:

Add book: bookstorer can add book

edit book: bookstorer can edit book he added before

Reader Actor Functions:

Read blogs : reader can read writers blogs on blogs tap

### 3.3.2 Similar System Information

Good Reads : Goodreads is a site for readers and book recommendations. Their mission is to help people find and share books they love.

Listy : Listy is a site that matches the joy of reading is sharing that joy with others. Their mission is to help people share and talk about books.

LibraryThing : LibraryThing is a site where users can design their libraries, LibraryThing also describe itself as "Facebook for books" where users can check out other peoples' libraries, see whose library is most similar to them and swap reading suggestions

AuthorsDen : AuthorsDen is a site focused on re-creating the historical face-to-face author and reader relationship where authors create, share, interact with each other and sell direct to their readers, but this time rather than taking place in a town square or temple it takes place online throughout the world.

### 3.3.3 Comparisons To Our System

Although these similar systems are very specific and strong at what they serve for example goodreads is one of the best book recommendation sites in the world, Listy is great for interacting with other readers, LibiraryThing is very helpful for organizing your books and AuthorsDen is very helpful for selling and purchasing books, These functionalities doesn’t exist in one place. ## app name ## combines most of these functionalities in one place so the readers and publishers don’t need to swap between platforms to do a specific task. Users can just sign up to ## app name ## and enjoy all of these functionalities.

------------------------------------------------------------------------------------------------------------------------------------------

## SCREEN SHOTS OF COMPARISON BETWEEN OUR SYSTEM AND THESE SYSTEMS ##

------------------------------------------------------------------------------------------------------------------------------------------

### 3.3.4 User Characteristics

Our System is easy to use and free to join for anyone who has the basic knowledge to navigate into a website to use blogging, chatting, swapping and purchasing books functionalities.

### 3.3.5 User Problem Statement

The difficulty of reaching out to latest books to purchase, finding people who share the same reading interests and joining very specific categorized blogging platform.

### 3.3.6 User Objectives

- Find books they are looking for.

- Find readers with same interests

- Write their blogs easily

- Purchase their books quickly

### 3.3.7 General Constraints

Any user in the system needs an internet access and English Knowledge.

### 3.3.4 User Characteristics

### 3.3.4.1 Person Actor

Register :

|  |  |
| --- | --- |
| Name | register |
| Actor | person |
| Preconditions | Open register form |
| Input | Username, Email, Password, Date of birth, Address, his Role (Reader, Writer, BookStorer) |
| Output | User is directed to site home page |
| Description | Any person can sign up to the application |
| Postconditions | Person have an account and redirected to home page |

Login :

|  |  |
| --- | --- |
| Name | Login |
| Actor | Person |
| Preconditions | Open login form, enter valid username and password |
| Input | Valid username, valid password |
| Output | User is directed to site home page if username and password is valid |
| Description | User must enter his username and password for authorizing that he has an account |
| Postconditions | User redirected to site home page |

Logout:

|  |  |
| --- | --- |
| Name | Logout |
| Actor | Person |
| Preconditions | Person must be logged in |
| Input | None |
| Output | Clear session id |
| Description | User logged out of the application |
| Postconditions | User redirected to site home page |

Edit Profile:

|  |  |
| --- | --- |
| Name | Edit profile |
| Actor | Person |
| Preconditions | User must be logged in |
| Input | Valid username or email or address |
| Output | Profile is updated |
| Description | Allow user to update his information |
| Postconditions | The updated information will be displayed in the profile and saved in the database |

### 3.3.4.1 writer Actor

Add post :

|  |  |
| --- | --- |
| Name | Add post |
| Actor | writer |
| Preconditions | writer must be logged in |
| Input | Title, Content |
| Output | Blog post is added to database. Writer is redirected to site home page. |
| Description | Allow writer to write the blog title and content in the add post form. |
| Postconditions | Blog added to database, writer is redirected to home page. |

Delete post :

|  |  |
| --- | --- |
| Name | Delete post |
| Actor | Writer |
| Preconditions | Writer is logged in, writer selects a specific blog post |
| Input | Select blog post title |
| Output | The blog post is deleted from database. |
| Description | Allow writer to delete a blog post he wrote before |
| Postconditions | User redirected home |

Edit post:

|  |  |
| --- | --- |
| Name | Edit post |
| Actor | Writer |
| Preconditions | Writer is logged in, writer selects a specific blog post |
| Input | Enter title, content |
| Output | The blog post is updated into the database. |
| Description | Allow writer to edit a blog post he wrote before |
| Postconditions | User redirected home |
|  |  |

Read post :

|  |  |
| --- | --- |
| Name | Read post |
| Actor | Writer |
| Preconditions | Writer must be logged in |
| Input | Enter the link to a specific blog post |
| Output | Blog post is viewed to writer |
| Description | Allow writer to read a specific blog post |
| Postconditions | Blog post is viewed. |

### 3.3.4.1 Reader Actor

Read post :

|  |  |
| --- | --- |
| Name | Read post |
| Actor | Reader |
| Preconditions | Reader must be logged in |
| Input | Enter the link to a specific blog post |
| Output | Blog post is viewed to reader |
| Description | Allow user to read a specific blog post |
| Postconditions | Blog post is viewed. |

### 3.3.4.1 BookStorer Actor

Add book:

|  |  |
| --- | --- |
| Name | Add book |
| Actor | BookStorer |
| Preconditions | BookStorer must be logged in |
| Input | Title, Content |
| Output | Book is added to database. Bookstorer is redirected to site home page. |
| Description | Allow bookstorer to add books into the site. |
| Postconditions | book added to database, bookstorer is redirected to home page. |

Delete book:

|  |  |
| --- | --- |
| Name | Delete post |
| Actor | bookstorer |
| Preconditions | bookstorer must be logged in |
| Input | Book id |
| Output | Book is deleted from database. bookstorer is redirected to site home page. |
| Description | Allow bookstorer to delete book he added before. |
| Postconditions | Book is deleted from database. bookstorer is redirected to site home page. |

### 3.3.4.1 Admin Actor

Add post :

|  |  |
| --- | --- |
| Name | Add post |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | Id, Title, Content |
| Output | Blog post is added to database. |
| Description | Allow writer to write the blog title and content in the add post form. |
| Postconditions | Blog added to database |

Delete post :

|  |  |
| --- | --- |
| Name | Delete post |
| Actor | Admin |
| Preconditions | Admin is logged in |
| Input | Select blog post id |
| Output | The blog post is deleted from database. |
| Description | Allow Admin to delete a blog post |
| Postconditions | The blog post is deleted from database. |

Edit post:

|  |  |
| --- | --- |
| Name | Edit post |
| Actor | Admin |
| Preconditions | Admin is logged in, select the post to edit |
| Input | Enter id, title, content |
| Output | The blog post is updated into the database. |
| Description | Allow Admin to edit a blog post he wrote before |
| Postconditions | The blog post is edited into database. |
|  |  |

Read post :

|  |  |
| --- | --- |
| Name | Read post |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | Enter the link to a specific blog post |
| Output | Blog post is viewed to Admin |
| Description | Allow Admin to read a specific blog post |
| Postconditions | Blog post is viewed. |

Add book:

|  |  |
| --- | --- |
| Name | Add book |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | Id, Title, Book |
| Output | Book is added to database. |
| Description | Allow Admin to add books into the site. |
| Postconditions | book added to database |

Delete book:

|  |  |
| --- | --- |
| Name | Delete post |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | Book id |
| Output | Book is deleted from database. |
| Description | Allow Admin to delete book from database. |
| Postconditions | Book is deleted from database. |

Delete user :

|  |  |
| --- | --- |
| Name | Add user |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | User id |
| Output | User is deleted from database and all user activities associated with it |
| Description | Allow admin to delete a specific user from database |
| Postconditions | All User data is deleted from database and all user activities associated with it |

Add User:

|  |  |
| --- | --- |
| Name | Add user |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | Username, Email, Password, Date of birth, Address, his Role (Reader, Writer, BookStorer) |
| Output | New user is added to database |
| Description | Admin can add many user as he wants and new user can login into system by the information that admin provided |
| Postconditions | New user is added to database |

Edit user:

|  |  |
| --- | --- |
| Name | Edit profile |
| Actor | Admin |
| Preconditions | Admin must be logged in |
| Input | User id, Valid username or email or address |
| Output | Profile is updated in the database |
| Description | Allow admin to update any user information |
| Postconditions | The updated information is saved into the database |

## Template ##

Function4 :

|  |  |
| --- | --- |
| Name |  |
| Actor |  |
| Preconditions |  |
| Input |  |
| Output |  |
| Description |  |
| Postconditions |  |

## END TEMPLATE ##

### 3.5 Interface Requirements

### 3.5.1 Admin Interfaces

Described in details in the system interface chapter 8.

### 3.5.2 User Interfaces

Described in details in the system interface chapter 8.

### 3.5.3 CLI

The command-line interface is not available in our application

### 3.5.4 API

In this section all the external package that is used is listed

* Django
  + Description: Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
  + usage: it’s used to create the application and it takes care of running the application server.
* Redis
  + Description: Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker.
  + Usage: it’s used to run server for Django channels to handle chat between users
* ## some more ##

### 3.5.5 Diagnostics or Rom

## Database general description ##

### 3.5.6 Hardware interfaces

The project doesn’t include hardware devices.

### 3.5.7 Communication interfaces

Network interface are not available in our application.

## Network interface might be included later ##

### 3.5.8 Software Interfaces

Describes any remaining software interfaces not included above.

### 3.6 Performance Requirements

There is no performance requirement in this system because the server request and response is depended on the end user internet connection.

### 3.7 Design Constraints

The application requires an internet connection to interact with the system API.

### 3.8 Other non-functional attributes

### 3.8.1 security

Users log in, and authentication function authenticates the user name and password, making sure the user is already registered in the system database.

1. Passwords are saved encrypted in the database
2. System is protected against vulnerabilities such as SQL injection attacks

Users can read other users contents but they cannot modify, edit anything except their personal information.

System has many different users types and every user has access constraints

### 3.8.2 Binary Compatibility

This system will be compatible with any computer that has a web browser and internet connection.

### 3.8.3 Reliability

The system should be accessible at any time.

### 3.8.4 Maintainability

## will be filled later ##

### 3.8.5 Portability

This application is Web-based which makes it compatible with any device. Independent on the OS-system in which users communicate with. The end user interface is fully portable and any system using web browser should be able to use the features of the application.

### 3.8.6 Extensibility

The system is designed and documented in way that anybody with understanding of web development can be able to extend the system.

### 3.8.7 Re-usability

The system is well-documented for new administrators to change content as needed. Also the system is designed in a way that administrators can modify content without having to modify source code.

## 3.8.8 Resource Utilization

The system is accessible from any device with an active internet connection.

The system requires an active server with adequate hard drive space and available memory.

## 3.8.9 Serviceability

The system comes with in-depth user manual that guide the user with detailed steps of interactions with the system starting with signing up and covering all other aspects of functionalities that the users can act upon the system.

The user manual also includes specific detailed instructions for administrators to guide them on how to operate the content of the system and how to apply modifications on the system.

## 3.9 Primitive Class Diagram

## 3.9.1 Simple Class Diagram

## ATTACH SCREENSHOT OF CLASS DIAGRAM ##

## 3.9.2 Inheritance Relationships

## ATTACH SCREENSHOTS OF INHERTINACE FROM SIMPLE CLASSDIAGRAM ##