Catalin Bent

URS

Individual

Table of Contents

[1. Introduction 2](#_Toc128953137)

[Desktop Application Requirements 2](#_Toc128953138)

[Website Application Requirements 2](#_Toc128953139)

[2. Non-Functional Requirements 2](#_Toc128953140)

[Maintainable and extendable 2](#_Toc128953141)

[• Bug free solution 2](#_Toc128953142)

[• Secure software 2](#_Toc128953143)

[3. Use Cases 3](#_Toc128953144)

[UC-01: Log In 3](#_Toc128953145)

[UC-02: Log Out 3](#_Toc128953146)

[UC-03: Add Book 3](#_Toc128953147)

[UC-04: Edit book 4](#_Toc128953148)

[UC-05: Deactivate Books 4](#_Toc128953149)

[UC-08: Log in 5](#_Toc128953150)

[UC-09: Sign Out 5](#_Toc128953151)

[UC-10: Sign Up 6](#_Toc128953152)

[UC-11: Add Books to “Favorite List” 6](#_Toc128953153)

[UC-12: Remove Items from Shopping Cart 6](#_Toc128953154)

[UC-13: See account info 7](#_Toc128953155)

[UC-14: Edit account information 7](#_Toc128953156)

[UC-15: Add Review 8](#_Toc128953157)

# Introduction

This project is made so that users can see and rate their favorite book but also to just explore and see other books that they might like to read. The base feature of this website is rating and adding books to a personal list so that you can save them.

## Desktop Application Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **A user should be able to** | **Target actor** | **Use Cases** | **Priority** |
| FR-100 | Log In | Employee | UC-01 | Must Have |
| FR-101 | Log Out | Employee | UC-02 | Must Have |
| FR-102 | Add Books | Employee | UC-03 | Must Have |
| FR-103 | Edit Books | Employee | UC-04 | Must Have |
| FR-104 | Deactivate Books | Employee | UC-05 | Must Have |

## Website Application Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **A user should be able to** | **Target actor** | **Use Cases** | **Priority** |
| FR-200 | Log In | User | UC-08 | Must Have |
| FR-201 | Log Out | User | UC-09 | Must Have |
| FR-202 | Sign Up | User | UC-10 | Must Have |
| FR-203 | Add Books to “Favorite List” | User | UC-11 | Must Have |
| FR-204 | Remove Books from “Favorite List” | User | UC-12 | Must Have |
| FR-205 | See Account Info | User | UC-13 | Must Have |
| FR-206 | Edit Account Info | User | UC-14 | Must Have |
| FR-207 | Add Review | User | UC-15 | Must Have |

# Non-Functional Requirements

## Maintainable and extendable

Proper OO principles must be applied to ensure good maintainability and extensibility of the

code base.

## • Bug free solution

Appropriate testing techniques must be used when implementing the solution to ensure

proper functioning.

## • Secure software

Only authorized people may make use of the solution and can only access data they are

authorized for. Passwords and user input must also be handled appropriately.

# Use Cases

## UC-01: Log In

Requirement: FR-100

Short Description: An employee logs in

Actor: Employee

Pre-conditions: No Employee is logged in

Post-conditions: The employee is logged in

Basic Flow:

1. The actor goes to log in
2. The system requests log in credentials
3. The actor fills in log in credentials and confirms
4. The system checks the credentials and notifies the customer that he has logged in

Alternative Flow:

3a. The actor fills in the wrong credentials

.1. The system notifies the actor that the credentials are wrong

## UC-02: Log Out

Requirement: FR-101

Description: An employee logs out

Actor: Employee

Pre-condition: Employee is logged in

Post-condition: The employee is logged out

Basic Flow:

1. The actor goes to log out
2. The system notifies the actor that he has successfully logged out
3. End of use case

## UC-03: Add Book

Requirement: FR-102

Description: An employee wants to add a book to the store

Actor: Employee

Pre-conditions: Employee is logged in

Post-condition: A book is added

Basic Flow:

1. The actor goes to add an book
2. The system requests the book information
3. The actor fills in all the required information
4. The system adds the book and confirms that the book has been added

Alternative flow:

3a. The actor forgets to fill in certain parts

.1. The system tells the actor that not everything has been filled

.2. The actor is sent to step 2 of the use case

## UC-04: Edit book

Requirement: FR-103

Description: An employee wants to edit a book

Actor: Employee

Pre-conditions: Employee is logged in

Post-conditions: A book is edited

Basic Flow:

1. The actor goes to edit the information of a book
2. The system requests the new book information
3. The actor fills in all the required information
4. The system edits the book information and confirms that the book has been edited

Alternative flow:

4a. The actor forgets to fill in certain parts

.1. The system notifies the actor that not everything has been filled

.2. The actor is sent to the fourth step of the use case

## UC-05: Deactivate Books

Requirement: FR-104

Description: An employee wants to deactivate a book

Actor: Employee

Preconditions: Employee is logged in

Post-condition: The book is deactivated

Basic Flow:

1. The actor goes to deactivate a book
2. The system deactivates the book and notifies the actor

## UC-08: Log in

Requirement: FR-200

Short Description: A user logs in

Actor: user

Pre-conditions: No user is logged in

Post-condition: user is logged in

Basic Flow:

1. The actor goes to log in
2. The system requests log in credentials
3. The actor fills in log in credentials and confirms
4. The system checks the credentials and notifies the customer that he has logged in

Alternative Flow:

3a. The actor fills in the wrong credentials

.1. The system notifies the actor that the credentials are wrong

## UC-09: Sign Out

Requirement: FR-201

Description: A user logs out

Actor: Customer

Preconditions: user is logged in

Post-condition: The user is logged out

Basic Flow:

1. The actor goes to log out
2. The system notifies the actor that he has successfully logged out
3. End of use case

## UC-10: Sign Up

Requirement: FR-202

Description: A user wants to sign up

Actor: user

Pre-conditions: user is not logged in

Post-condition: An account is created

Basic flow:

1. The actor goes to sign up and fills in the required information
2. The system saves the data and notifies the actor that the account was created successfully

Alternative flow:

3a. The actor does not fill in the required information

.1. The system notifies the user that he did not complete the required information

## UC-11: Add Books to “Favorite List”

Requirement: FR-203

Description: A user wants to add an item to his “Favorite List”

Actor: User

Pre-conditions: User is logged in

Post-condition: A book is added to the shopping cart

Basic flow:

1. The actor goes to add a book
2. The system puts the book in the “Favorite List”

Alternative flow:

3a. The book is not available because it is deactivated

.1. The system notifies the customer that the book is not available

## UC-12: Remove Items from Shopping Cart

Requirement: FR-204

Description: A user wants to remove an item to his shopping cart

Actor: User

Pre-conditions: User is logged in

Post-condition: A book is removed from the “Favorite List”

Basic flow:

* 1. The actor goes to remove an item from the “Favorite List”
  2. The system removes the book

## UC-13: See account info

Requirement: FR-205

Description: A user wants to see the account info

Actor: User

Pre-conditions: User is logged in

Post-condition: Account information is shown

Basic flow:

1. The actor goes to see the account information
2. The system shows the actor the account information

## UC-14: Edit account information

Requirement: FR-206

Description: A user wants to edit his account information

Actor: User

Pre-condition: User is logged in

Post-condition: Account information is edited

Basic flow:

1. The actor goes to see the account information
2. The system shows the actor the account information
3. The actor inputs the new information
4. The system saves the new information and notifies that the user has edited the information successfully

Alternative flow:

4a. The actor leaves a textbox empty

.1. The system notifies the customer that he must fill in everything

## UC-15: Add Review

Requirement: FR-207

Description: A user wants to add a review

Actor: User

Pre-conditions: User is logged in

Post-condition: A review is added

Basic flow:

1. The actor goes to add a review
2. The system saves the review and displays it