BBM487 -Software Engineering Laboratory

Library Book Loan System
System-Wide Requirements Specification

Group 4:
Ali Burak ÖZAKINCI - 21427162

Sabit ÇATALTAŞ - 21692544

Aishat Temitope OLOMOWEWE – 21603775

Library Book Loan System
System-Wide Requirements Specification

1. Introduction

The System Requirements Specification is a document of the system requirements for customers, end-users, and software developers. It contains what services the system should provide, the system properties, and the constraints on the operation and implementation of the system.

This document contains details about the requirements,use-cases that mentioned below and user interface prototypes.

2. System-Wide Functional Requirements

2.1 Auditing

All user interactions and book transactions will be stored as logs.An administrator process that supervised by a system manager will keep the audit trail.This is a requirement for data mining,examining the reasons for unexpected events on system. The database will have a functional audit trail on its own for backup and other operations.

2.2 Authentication

All registered and authenticated users have deemed to be accepted our policy statement.Unauthenticated users can only search for books.Even if they are not a registered user,system gives them a unique guest id and tracks their usage of program.Librarians and registered users can access their relative-privileged actions after logging in.Users also tagged with a unique id.System expects them to have unique usernames in registering process.

2.3 Licensing

The Library Book Loan System that is being developed is an open source project.Software is licensed with LGPL license.This means anyone can take this code and develop their own project based on this,the software respects all agreements of its license.There won't be any tracking or monitoring about the usage of our software in the future.

2.4 Reporting

All related logs must be reported when an unexpected event occurs.Backup logs and statistics must be reported every month.

2.5 Scheduling

The software has several scheduled events that must be done :
- Database backup – Second week of every month.
- Late Fine Check and notification for users – Everyday at 5pm.
- Notify users on book waiting list – When a book returns.

2.6 Security

- Multiple login sessions with a single account is prohitibed.
- Library users cannot modify book information.
- Librarians cannot modify user's late fine.
- All sensitive user info must be encrypted.

3. System Qualities

3.1 Usability

- **Ease of use** is an important quality for interface and user interactions.A user must be perform the action with minimum interaction (click or keystroke) with the interface.
- **Ease of learning** is related with the ease of use.The learning curve must be short.Provided interface must have enough elements to access and navigate easily but must not confuse the user with a complex structure.Learning period for users and librarians is expected to be 1 hours maximum.Simple interface design will also improve the **ease of remembering** quality of the system.
- **Task efficiency** will be provided by clear, familliar interface and short response time**.**Database operations must be done in short intervals in an optimized way to reduce waiting time.Besides that, confirm messages and status messages makes interface forgiving for users.
- **Subjective satisfaction** will be provided by periodic usage tests.A group of people who will use the system for the first system will test and give feedbacks about usability of the software.

3.2 Reliability

- **Accuracy :**
  o Late book calculations must be based checkout hour and date to current hour and date**.**
  o Book late fines must be calculated up to 2 decimal digit and the currency is TRY.
  e.g 3.15 TRY
- **Availability :**
  Library software that integrated to library must be working when the library is open.Scheduled backup and recovery operations will be performed at midnight to not interrupt working flow of library.Users won't be degraded because of any scheduled event during a time period.
  In case of failure on system or database, the problem must be inpected immediately by developer team and must be solved in maximum of 1 week.
  Backup cloud database will recover the user and book information in case of an unexpected failure on database.This also defines the **recoverability** for the system.

- **Frequency and severity of failures:**
  Critical failures :
  - o Loss of data can be examined and recovered by backup database.
  - o Inability of a function,must be reported and problem must be solved by maintanance team in max 10 days.

3.3 Performance

- **Response times:**
  - o Any navigation action on home page between a user and the system shall have a maximum response time of 1 seconds.
  - o Search for books operation shall have a maximum response time of 3 seconds.This depends on the DBMS access time.
  - o Any database CRUD operations must be performed in 3 seconds maximum.To achieve that database management system must be optimized with appropriate indexing and cache mechanisms.This operations include librarians's user and book manipulations.
  - o Pay fine operation must be performed in maximum of 5 seconds.This depends on the external payment API.
- **Throughput:**
  - o Information flow capacity of system supports 200 online users and 500 transaction per second.
- **Start Time:**
  - o Average start-up time must be 1 minute.
- **Shutdown:**
  - o Average shutdown time must be 1 minute.It can be 5 min if it includes a backup opeation.

3.4 Supportability

- **Adaptability:**
  - o The software must adapt many environments and operating systems.The cross-platform feature helps that.
- **Configurability:**
  - o The system must be configurable for different library environments with different needs.Storage capacity,managing staff accounts,user account attributes and book attributes must be customizable.Maintanance team is responsible for these operations.
- **Installation:**
  - o The software installation only requires a Windows or Linux environment.

- **Level of Support:**
  - o Project team must provide all levels of support(L1,L2,L3,L4) depending on the situation.Library users can access the help desk by phone number of support.For the bigger problems the developer team must take care of the system in-place.
  - o The help desk support is not a part of customer's provided support directly.If a user calls for help and librarian can't find a solution,librarian redirects user to project team.
- **Maintainability:**
  - o Besides bug-fixing, a new maintanance release will be offered once a year.This is an obligation for keeping the software up-to-date to techological developments and needs.
- **Scalability:**
  - o Software must survive increasing user and book interaction over time without losing any significant performance aspect.

4. System Interfaces

4.1 User Interfaces

4.1.1 Interface Prototoypes for Use Cases

Graphical info provided in Appendix-B.The required interfaces are:

Interface Prototype	Use-case
Login Window	Login
Home for librarians	Menu for other use-cases
Home for registered users	Menu for other use-cases
Search for books tab on home window - Book info	Search for books-Pay Fine-Checkout
Book info for non-guest users	Add name to waiting list
Book info for guest users	Search for books
View my books tab on home window	Self return
Manipulate Book tab on librarian's home	Manipulate Book
Manipulate Users tab on librarian's home	Manipulate Customers

4.1.2 Look & Feel

- Project interface backgrounds should be white to get a clean and simple look.

- Push buttons should be in different color and shadowed.

- Instead of separate windows for each use case,a home window with tabs for use cases must be used.

4.1.3 Layout and Navigation Requirements

- Login window, book info window and home window are seperated windows.
- Home window must contain tabs to other functionalities "Search for books, View My Books,Home"
- Book info page should have "Add name to waiting list button" for non-guest users.

4.1.4 Consistency

- Tab based navigation and large buttons should be placed to improve predictibility and consistency.
- Window sizes must be middle-sized for home page and relatively small for login page.

4.1.5 User Personalization & Customization Requirements

- Librarians must have tabs to manipulate books and users.
- Users must have a button to joining to waiting list different than others on book info window.
- Users must have a tab for search,checkout and waiting list operations.
- Users must have a tab to view his/her books and return function.

4.2 Interfaces to External Systems or Devices

4.2.1 Software Interfaces

- The Library Book Loan System has 3 main components.System itself, Database Management System and Payment API.

- The database management will be based MongoDB.

- Python 3.5 will be used as programming language.

- PyQt5 will be used to design and implement interfaces.

- The connection between database and the system will be established using PyMongo.

- For money transactions an external payment system will be used.

4.2.2 Hardware Interfaces

There won't be a hardware interaface associated with project.

4.2.3 Communications Interfaces

There won't be a hardware interaface for communication for the normal release of software but in the future, software can be integrated to a mobile app to improve portability.

5. Business Rules

5.1 Book Loan

5.1.1 Max Books Allowed to Users

- Each member user can loan at most 4 books at any given time.

5.1.2 Availability to Loan Books

- If user has an unpaid fine,he can't loan any books until the user pays the fines.

5.1.3 Loan Period

- A loaned book must be returned within 40 days.

5.2 Fine Policy

5.2.1 Fine Condition

- If the user does not return the loaned book on time,the fine process starts.Inıtial value of late fine is 5 TRY.

5.2.2 Calculate Fine

- For every day the book not returned after legal time,it increases by 1.5 TRY
- Fine is calculated by : $5 + (\text{Late days}) * (1.5)$

6. System Constraints

- Python 3.5 will be used to implement the project.
- Software will be operation on Linux and Windows.

7. System Compliance

7.1 Licensing Requirements

- The software is LGPL licensed. Open-sourced for every types of use.

7.2 Legal, Copyright, and Other Notices

- Non-commercial and commercial redistribution or modification of code is allowed.

7.3 Applicable Standards

- Pep8 Python style guide will be followed on implementation.

- IEEE standards will be used for software design process.

8. System Documentation

- Application will have an 'HELP' section for users.
- A user manual documentation will be provided to customers and users.
- User manual will include detailed usage info for every UI element and functionality.

APPENDIX A
USE CASES

Use Case 1 : Login

Actors	Librarian and library user
Pre-Condition	1. User is not logged in. 2. User is registered
Post-Condition	1. User is logged in.
Main (Happy) Path	1.User enters his/her username and password and presses the Login button. 2.System sends login info to DBMS and checks if valid then show main page of application. 3.User will be able to use several registered-user features.
Alternative Path	1.If the username or password is invalid, the system will show a message indicating it. 2.User will be able enter the login info again.

Use Case 2 : Logout

Actors	Librarian and library user
Pre-Condition	User is logged in.
Post-Condition	User is not logged in.
Main (Happy) Path	1.User pushes the logout button to send a request to end his/her session. 2.System checks if any ongoing operation that can be interrupted by logout operation. 3.System ends user session and user logs out.
Alternative Path	1.If there is an ongoing operation that will be interrupted by the login operation,system sends an confirm message to user. 2.If user does not confirm this message,logout request cancels and his/her session continues.

Use Case 3 : View my books

Actors	Library user
Pre-Condition	User is logged in.
Post-Condition	User reviewed the list of loaned books.
Main (Happy) Path	1.User clicks 'View My Books' button to send a request. 2.System sends user info to DBMS. 3.DBMS queries items associated with the given user and send to system. 4.System renders and sends the list items to user. 5.User reviews the list of the books loaned.
Alternative Path	1.If there aren't any books associated with the given user,the user will see an empty list.

Use Case 4 : Search for books

Actors	Library user,guest user
Pre-Condition	-
Post-Condition	User reviewed the books he/she searched.
Main (Happy) Path	1.On the search page after clicking 'Search for books' button,user fills the search form and submits. 2.System converts submitted info to search query and sends to DBMS. 3.DBMS queries items that fits search criterias and returns to system. 4.System renders and sends results to user. 5.User reviews the list of the books searched.
Alternative Path	1.If there aren't any books with given attributes,system shows an error message to user and redirects to search page. 2.User will be able enter the search info again.

Use Case 5 : Issue late fine

Actors	System agent
Pre-Condition	-
Post-Condition	1.User is notified if there is a late fine issue with his/her account.
Main (Happy) Path	1.System agent starts a periodic fine status check on system. 2.System requests user's transaction info from DBMS. 3.DBMS searchs for the needed info and returns to system. 4.System checks loan and return times and updates book's status. 5.System calculates fines for books with "LATE" statuses. 6.User gets a notification about the late-fine situtation.
Alternative Path	1.If there aren't any books with "LATE" status associated with the user account,system turns to initial state and waits for another fine check session.

Use Case 6 : Pay Fine

Actors	Library User
Pre-Condition	1.User is logged in. 2.User have loaned but not returned books.
Post-Condition	1.User paid the fine. 2.The book's status on user account updated.
Main (Happy) Path	1.User selects a book from list to pay it's fine. 2.System retrieves the book status from DBMS. 3.System returns the book's fine amount to user and asks for a confirmation about payment. 4.User provides payment information. 5.System sends user and payment info to payment API. 6.Paymet API performs the action. 7.System updates book's status on given account. 8.User receives receipt info.
Alternative Path	1.If there aren't any books with "LATE" status associated with the user account,system redirects user to "View my books" page. 2.If the payment operation didn't performed successfuly,program waits for correct payment information from user.

Use Case 7 : Add Name To Book Waiting List

Actors	Library User
Pre-Condition	1.User is logged in. 2.The book to be loaned unavailable at request time.
Post-Condition	1.User added to the waiting list of the book.
Main (Happy) Path	1.User selects the unavailable book to be loaned. 2.User clicks "Add to waiting list" button on book info or search for books tab. 3.System forwards book and user info to DBMS 4.DBMS updates book's waiting list attribute. 5.User gets an confirmation message and waits until book is available.
Alternative Path	-

Use Case 8 : Self Checkout

Actors	Library User
Pre-Condition	1.User is logged in. 2.The book is pyhsically available to loan.
Post-Condition	1.User loaned the book.
Main (Happy) Path	1.User selects the book from the list after search and pushes "Checkout" button. 2.User scans the barcode of the book to be loaned. 3.System sends a search query to DBMS to check if the user has unpaid fines. 4.DBMS updates book's status and user account. 5.User loans the book.
Alternative Path	1.If user has any unpaid fines,he/she won't be able to loan any book until he/she pay the fines. 2.User returns to self checkout page.

Use Case 9: Self Return

Actors	Library User
Pre-Condition	1.User is logged in. 2.The book is pyhsically available to return.
Post-Condition	1.User returned the book.
Main (Happy) Path	1.User selects the book to be returned from "View My Books" section. 2.System waits for barcode input. 3.User scans the barcode of the book to be returned. 4.System retrieves information about the selected and barcode scanned book.Compares if they are the same. 5.DBMS updates the user account and book status. 6.System sends an "Operation completed" indicator message.a 7.User leaves the book to the library.
Alternative Path	1.If the selected book and barcode scanned book does not match,return operation will not be completed.

Use Case 10 : Notify of Book Being Late

Actors	System agent
Pre-Condition	-
Post-Condition	1.User is notified if there is a book available to loan that user waiting for.
Main (Happy) Path	1.System agent starts an available book check when a book is returned. 2.System checks returned book's waiting list. 3.System sends a notification to user that book is available.
Alternative Path	1.If there aren't any users in returned book's waiting list,book stays at available status.

Use Case 11 : Manipulate User (Create – Read – Update -Delete)

Actors	Librarian
Pre-Condition	1.Librarian is logged in. 2.The user account to act on must be exist for non-create operations.
Post-Condition	1.Librarian performed an CRUD operation on user account.
Main (Happy) Path	1.1.To delete,update or read a user,librarian navigates the "Manipulate User" tab on home window. 1.2.Librarian searchs for the user. 1.3.System sends form info to DBMS as search query. 1.4.Librarian gets result list. 1.5.Librarian selects user and action. 1.6.System performs the action. 1.7.DBMS updates. 1.8.System shows "Operation successful" message to librarian. 2.1.To create a user,librarian pushes the "Create User" button. 2.2.Librarian enters user info. 2.3.System sends info to DBMS. 2.4.DBMS checks if username is unique. 2.5. System shows "Operation successful" message to librarian.
Alternative Path	1.If the username of user to be created is not unique,it fails. 2.User returns to "Manipulate User" tab. 3.User reenters form info.

Use Case 12 : Manipulate Book (Create – Read – Update -Delete)

Actors	Librarian
Pre-Condition	1.Librarian is logged in. 2.The book record to act on must be exist for non-create operations.
Post-Condition	1.Librarian performed an CRUD operation on book record.
Main (Happy) Path	1.1.To delete,update or read a book record,librarian navigates the "Manipulate Book" tab on home window. 1.2.Librarian searchs for the book. 1.3.System sends form info to DBMS as search query. 1.4.Librarian gets result list. 1.5.Librarian selects book and action. 1.6.System performs the action. 1.7.DBMS updates. 1.8.System shows "Operation successful" message to librarian. 2.1.To create a book,librarian pushes the "Create Book" button. 2.2.Librarian enters book info. 2.3.System sends info to DBMS. 2.4.DBMS checks if book id and title is unique. 2.5. System shows "Operation successful" message to librarian.
Alternative Path	1.If the title and id of book to be created is not unique,it fails. 2.User returns to "Manipulate Book" tab. 3.User reenters form info.

APPENDIX B
USER INTERFACE
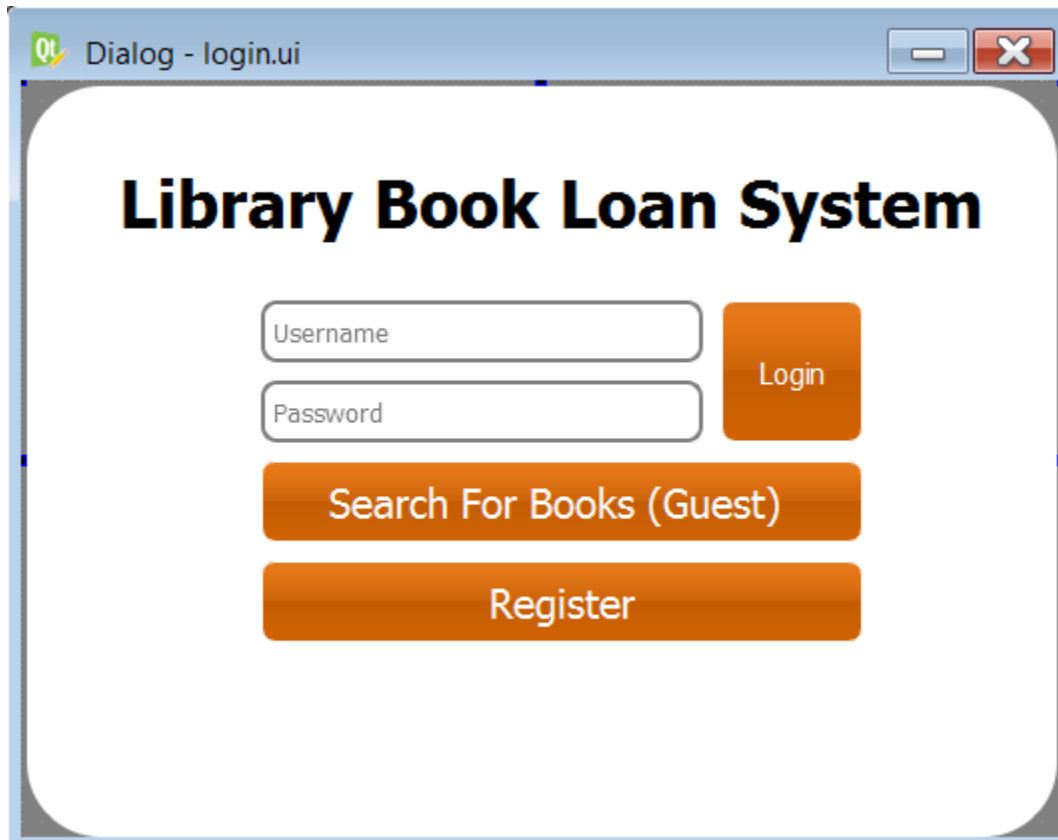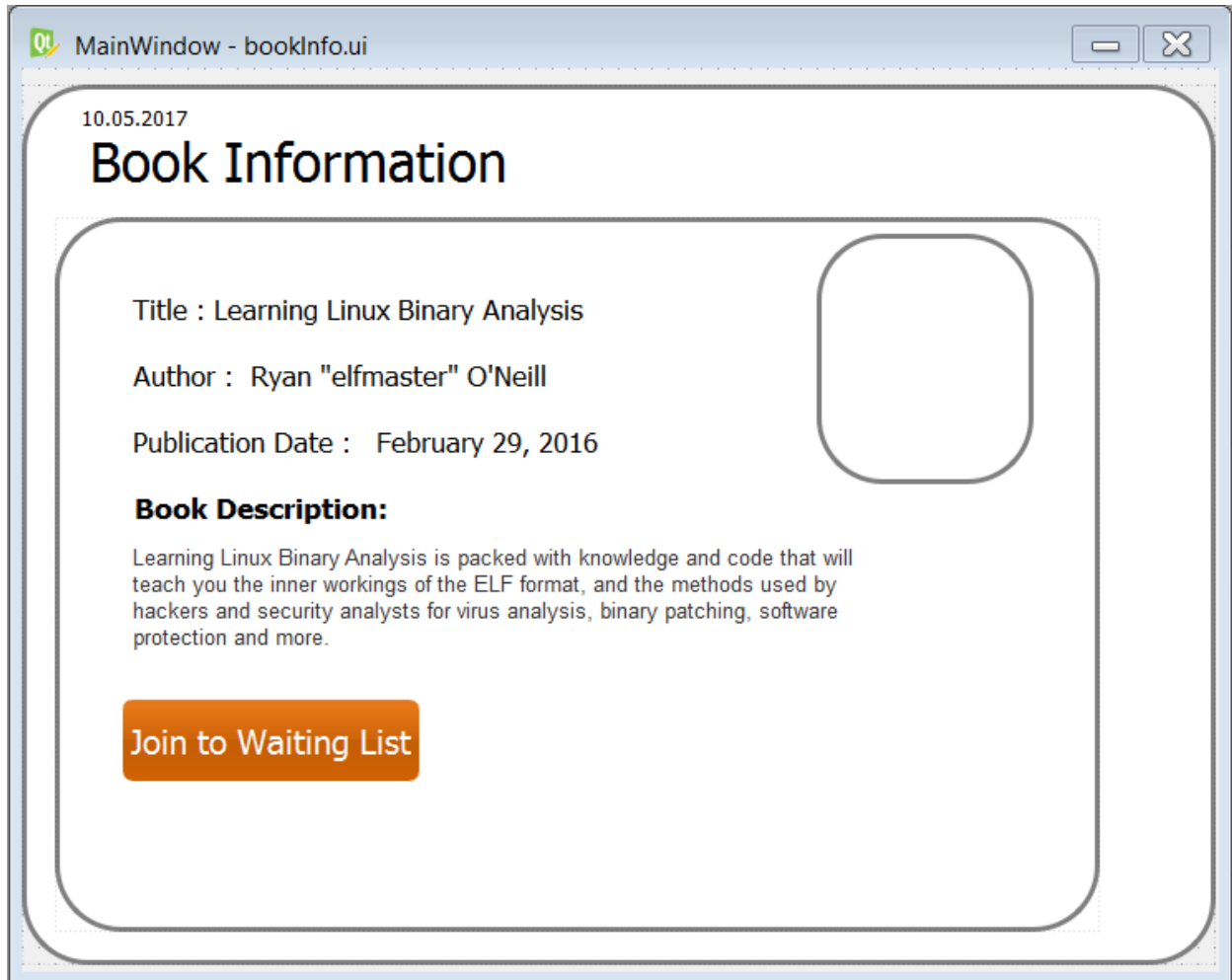

Login UI

Related Use Cases :

- Login
- Search For Books
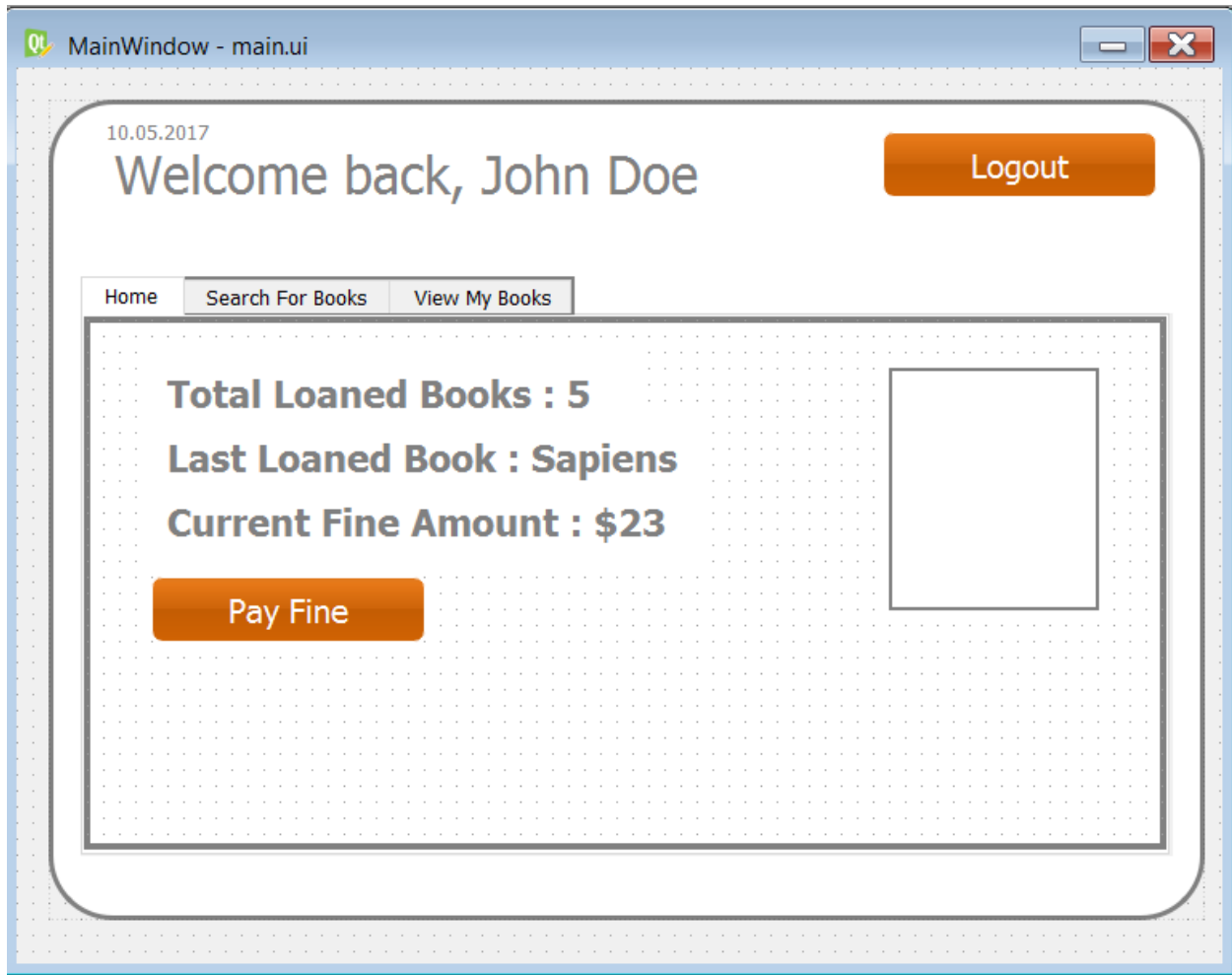
Home for member user

Related Use Cases :
- Self return
- Search for books
- Self checkout
- View my books
- Pay fine
- Logout

Book Info for member user

Related Use Cases :

- Add to waiting list
- Search for books
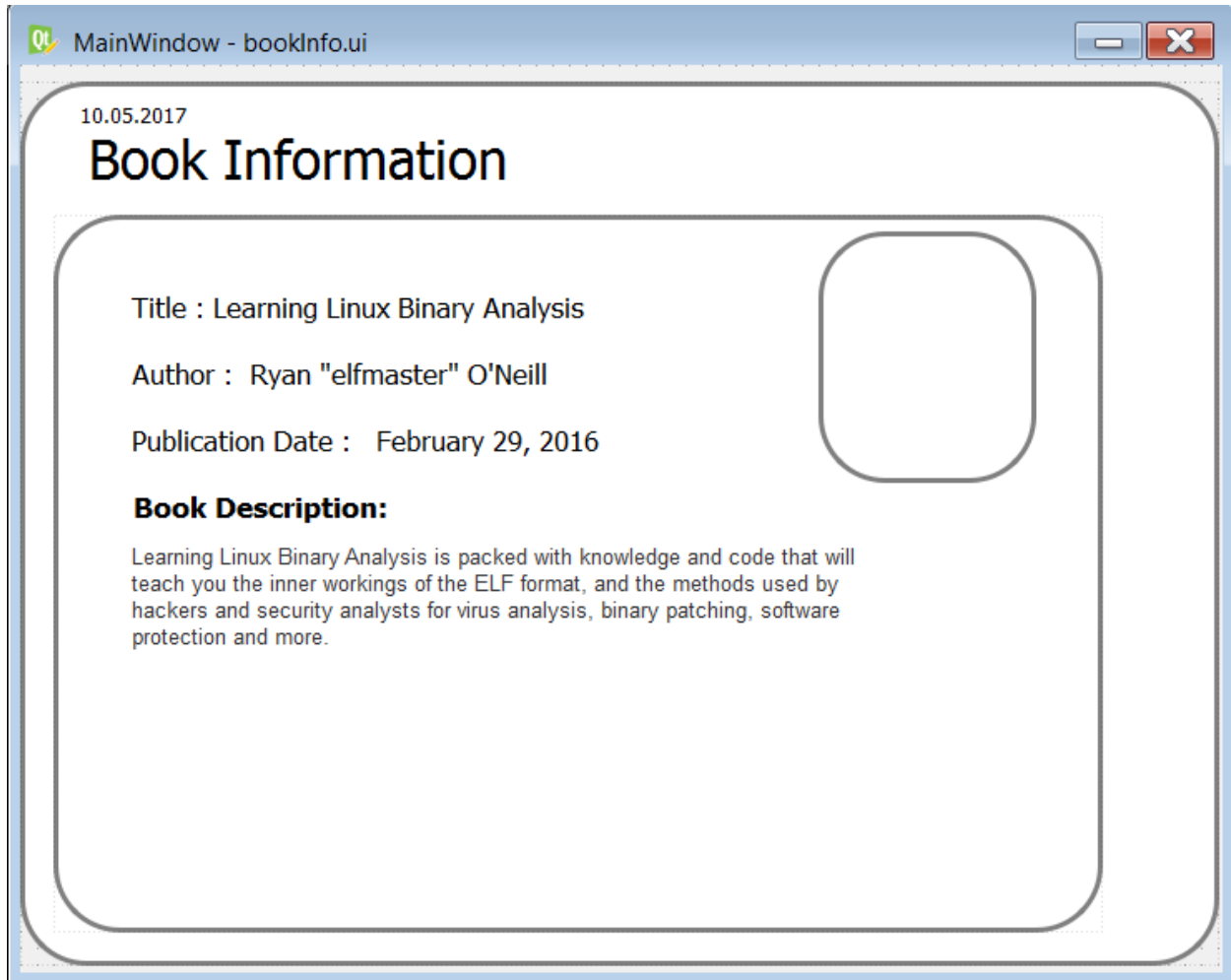
Book info for guest user

Related Use Cases :
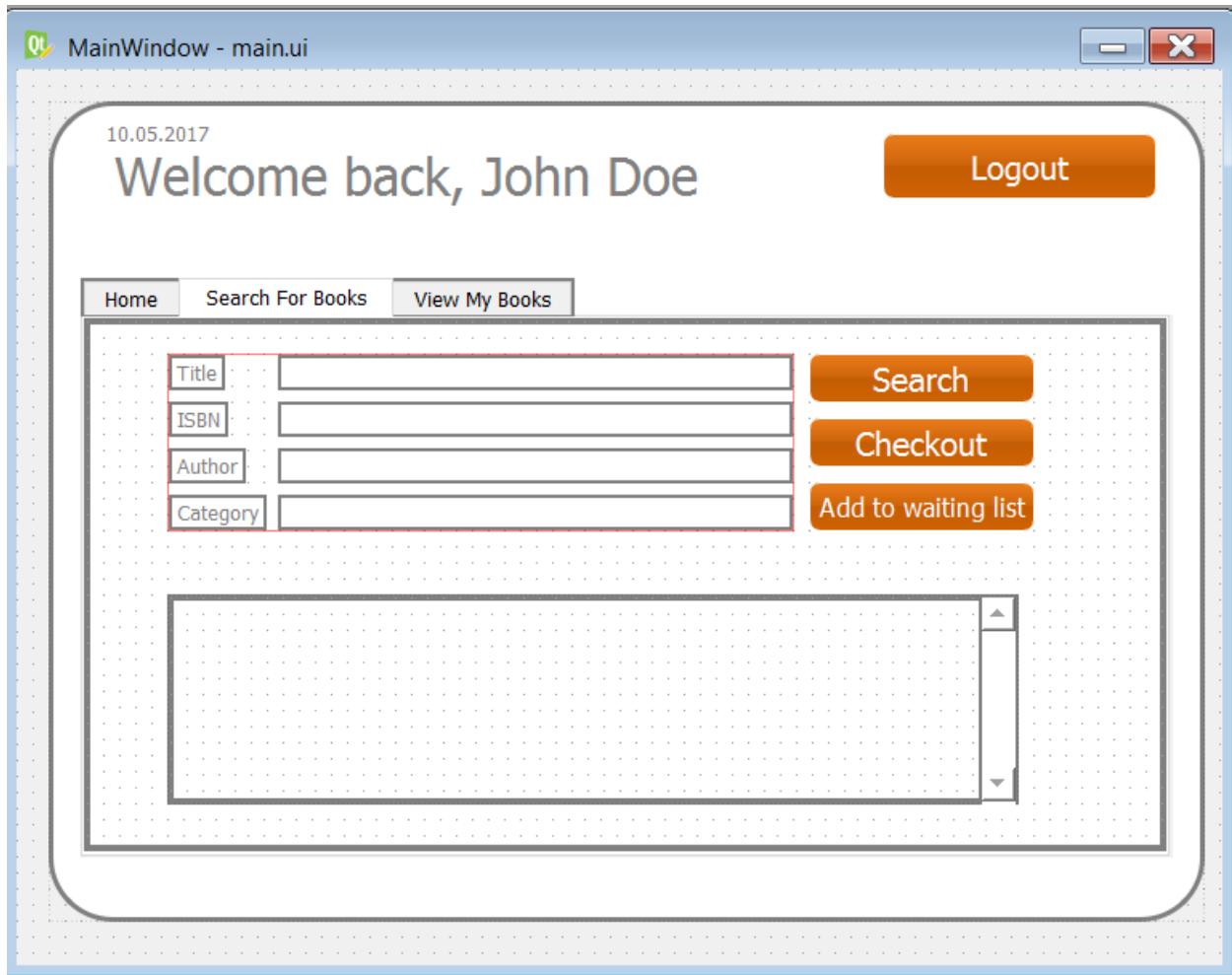
- Search for books

Search for books for member user

Related Use Cases :

- Search for books
- Add to waiting list
- Self-Checkout
- Logout

Search for books for guest user

Related Use Cases :

- Search for books

Manipulate Books for Librarian

Related Use Cases :

- Search for books
- Manipulate books

Manipulate User for Librarian

Related Use Cases :

- Search for books
- Manipulate users

APPENDIX C
ER DIAGRAM