**Software Design Document**

**For**

**BlackLetters**

**Distributed Information System For The Management Of A Library**

**Names:**

Cucu Sorin Mihai Grupa CEN 3.1B [sorinmihai.c09@gmail.com](mailto:sorinmihai.c09@gmail.com)

Dicu Cristian Paul Grupa CEN 3.1B [dicu.cristian20@gmail.com](mailto:dicu.cristian20@gmail.com)

Florea Andrei Cristian Grupa CEN 3.1B [andreifcristi19@gmail.com](mailto:andreifcristi19@gmail.com)

Craioveanu Vlad Mihail Grupa CEN 3.1A [vlad.craioveanu@yahoo.com](mailto:vlad.craioveanu@yahoo.com)

**Lab Section:** Software Engineering

**Date:** (03/25/2020)

**TABLE OF CONTENTS**

**INTRODUCTION**...................................................................................................................2

1.1 Purpose ...............................................................................................................................2

1.2 Scope ..................................................................................................................................2

1.3 Overview........................................................................................................................... 2

1.4 Reference Material............................................................................................................ 2

1.5 Definitions and Acronyms................................................................................................ 2

2. SYSTEM OVERVIEW....................................................................................................... 2

3. SYSTEM ARCHITECTURE.............................................................................................. 2

3.1 Architectural Design.......................................................................................................... 2

3.2 Decomposition Description ...............................................................................................3

3.3 Design Rationale ...............................................................................................................3

4. DATA DESIGN ..................................................................................................................3

4.1 Data Description................................................................................................................ 3

4.2 Data Dictionary .................................................................................................................3

5. COMPONENT DESIGN ....................................................................................................3

6. HUMAN INTERFACE DESIGN .......................................................................................4

6.1 Overview of User Interface ...............................................................................................4

6.2 Screen Images ....................................................................................................................4

6.3 Screen Objects and Actions............................................................................................... 4

7. REQUIREMENTS MATRIX ..............................................................................................4

8. APPENDICES ......................................................................................................................4

**1. INTRODUCTION**

**1.1 Purpose**

The SDD document is intended for the code development team and, therefore, it must contain all the information required by a programmer to write code.

In this document the software requirments are translated into a representation of software components, interfaces, and data necessary for the implementation phase.

**1.2 Scope**

The scope of our application is to give users the possibility, through an easy to use interface, to reserve a book, monitor the borrowed books and purchase a monthly subscription to the library.

The product is designed to facilitate the access of library visitors to a number of services, without requiring them to go to the library. It also facilitates the process of managing books by the users or by the employees.

The final goal is to encourage and to make accessible to more people the pleasure of reading a hard cover book instead of a digital one.

**1.3 Overview**

This document will be organized in 8 chapters:

**1.3.1**  An introduction which includes the purpose and the scope of our application.

**1.3.2** A system overview which gives a general description of the functionality, context and design of our project.

**1.3.3** A system arhitecture which is concentrated in 3 paragraphs: an arhitectural design, decomposition description and a design rationale.

**1.3.4** A data design overview which includes informations about how the information domain of your system is transformed into data structures and how the major data or system entities are stored, processed and organized.

**1.3.5** A component design overview which includes a closer look at what each component does in a more systematic way. It provides a summary of your algorithm for each function in procedural description language (PDL) or pseudocode.

**1.3.6**  A human interface design which shows the functionality of the system from the user’s perspective. It explains how the user will be able to use our system to complete all the expected features. There are also in-app screen images provided with creen objects and actions.

**1.3.7**  A requirment matrix which provides a cross-­reference that traces components and data structures to the requirements in our SRS document.

**1.3.8**  Appendices section.

**1.4 Definitions and Acronyms**

**SRS:** A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

**HTTPS:** Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet.

**FTP:** The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

**2. SYSTEM OVERVIEW**

Upon accessing the application, the user will have the option to register himself, by creating a username and a password. Doing so, will result in creation of a personal account, where all his information is stored. This will also include the user in an existing User database, which is fully available to the administrator of the application.

The user will access the application through the user interface. From there, he will have the ability to navigate through the available books, which are stored in a database. Upon deciding on a book, he will have the option to borrow it, which leads to a message sent to the library, stating that the user will come for the book in the following 48 hours. He will also have the option to reserve it, in the situation that the book is either unavailable or he can’t be at the library in the given borrowing time frame. Regardless of the selected option, the action will be registered as a transaction and will be stored in a database which is directly linked to the User database. At this moment in time, the User database, as well as the Book database will be updated. The user will also have the option to purchase a subscription, which provides some advantages. There is an option for book donation, which sends the user to a form that he has to complete in order to be able to donate a book.

**3. SYSTEM ARCHITECTURE**

**3.1 Architectural Design**

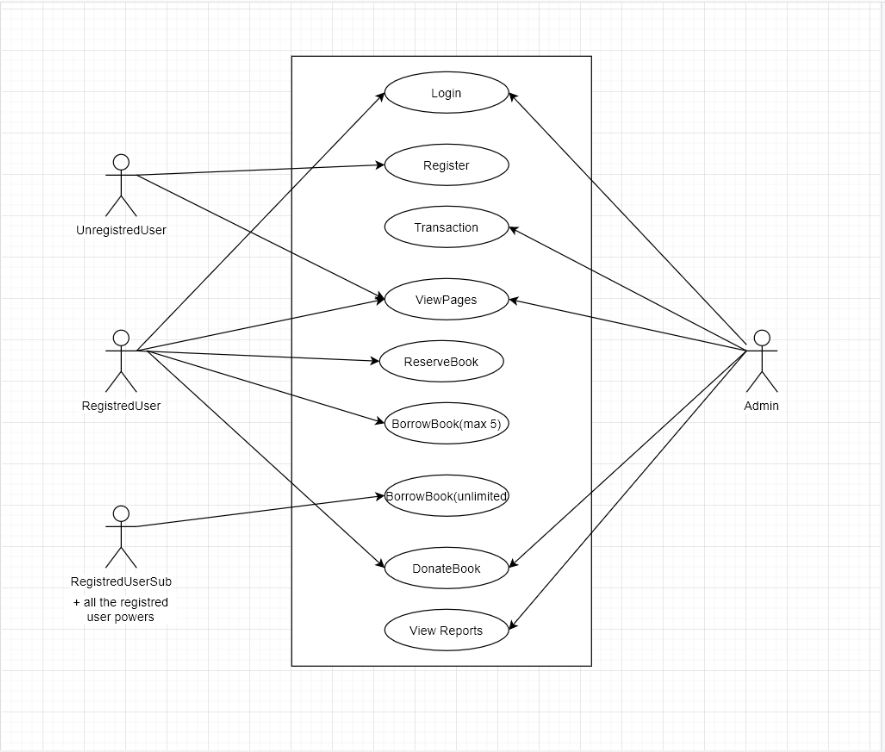
The main functional requirements for the application are:

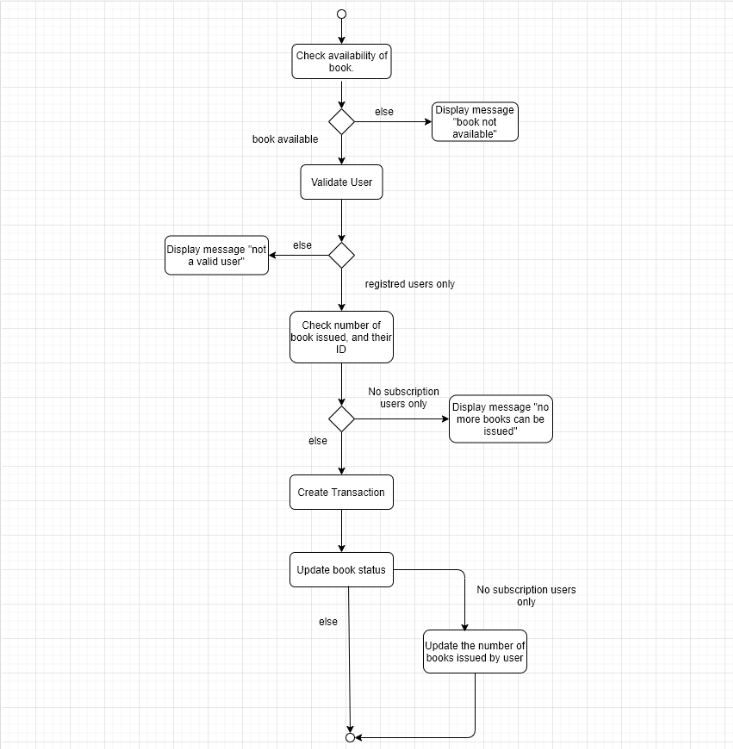
**3.1.1** Register and login, the unregistered users are only able to navigate the website without the posibility of using any of the funtionalities.

**3.1.2** Search a book. After you are logged in you can search a book by date, writer or genre.

**3.1.3** Donate / borrow a book. You are able to donate and to rezerve and borrow how many books you want if you pay a mounthly subscription otherwise you are allowed a maximum of 5 books.

**3.1.4** The admin system which enables him to manage the users and the database, to view reports and to make transactions.





**3.2 Decomposition Description**

The appliction is made of 6 classes: Book, Admin, Transaction, RegisterUser, UnregisteredUser and RegisterUserSubscription.

**3.2.1** The book class has the following subordonates: author, name, type, status, price.

The functions used for this class are: getBook(); displayBookDetail(); updateStatus();

**3.2.2** The Admin class has the following subordonates: firstName, lastName, password, email.

The functions used for this class are: verifyMember(); addBook(); validateTransaction();

**3.2.3** The Transaction class has the following subordonates: transid, memberld, bookld, dateOfAquire, dateOfReturn.

The functions used for this class are: createTransaction(); deleteTransaction(); retrieveTransaction();

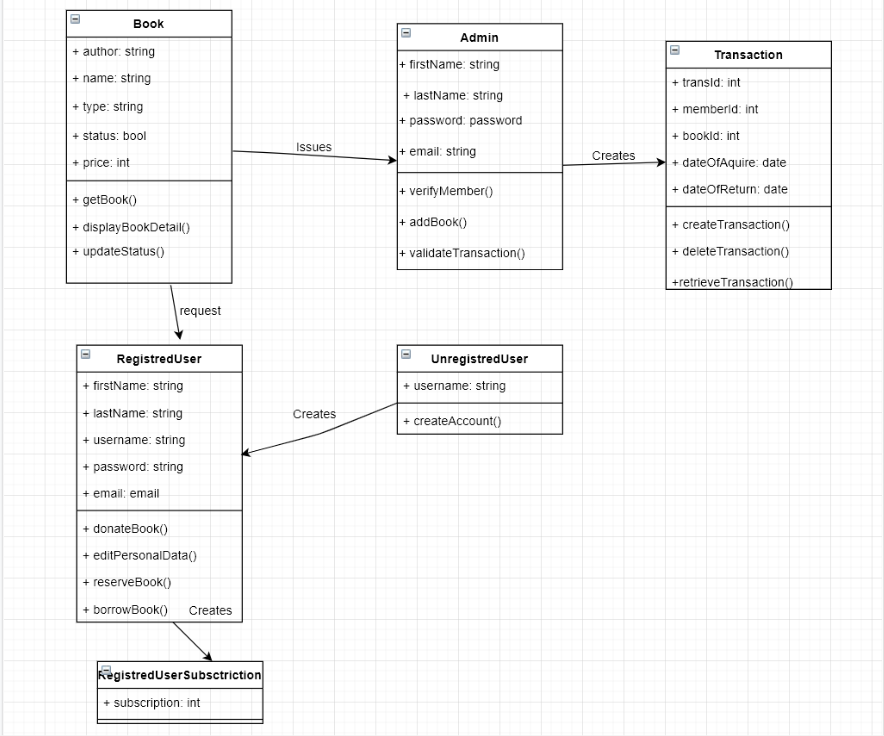
**3.2.4** The RegisteredUser class has the following subordonates: firstName, lastName, username, password, email.

The functions used for this class are: donateBook(); editPersonalData(); rezerveBook(); borrowBook();

**3.2.5** The UnregisteredUser class has the following subordonates: username.

The functions used for this class are: createAccount();

**3.2.6** The RegisteredUserSubscription class has the following subordonates: subscription.



**3.3 Design Rationale**

We chose a modular approach to our application architectural design in order to optimize the response time and the required computational work. The class creation logic was to focus on the users types and start from there to add more and more functionalities. We emphasize on creating the implementation this way because there are definitive differences between the users and we want to keep the design simple and easy to understand and implent but also fully functional.

**4. DATA DESIGN**

**4.1** **Data Description**

There are 3 databases that are being used to store and manage the information:

**4.1.1** User

**4.1.2** Book

**4.1.3** Transaction

The Transaction and User databases are linked together, the transactions data being available for each user individually

The Book and User databases get linked through the Transaction database. When a user borrows a book for example, this is registered as a transaction and the information is processed and registered in the User’s books, in the User’s transactions and in the Book’s status.

**4.2** **Data Dictionary**

System entities:

**4.2.1** Administrator

**4.2.2** Book

**4.2.3** RegisteredUser

**4.2.4** RegisteredUserSubscription

**4.2.5** Transaction

**4.2.6** UnregisteredUser

**4.2.7** User

**5. COMPONENT DESIGN**

**Admin:**

**firstName:** First name of the account for the admin.

**lastName:** Last name of the account for the admin.

**password:** The password of the account for the admin.

**email:** The email address of the account for the admin.

**verifyMember():** A method the admin uses to verify a member.

**addBook():** A method the admin uses to add books for the site.

**validateTransaction():** A method the admin uses to validate a transaction that a registred user does.

**UnregistredUser:**

**username:** Name of the account for the user.

**createAccount():** The ability for the unregistred user to create an account.

**RegistredUsers**:

**firstName**: First name of the account for the user.

**lastName:** Last name of the account for the user.

**username:** Name of the account for the user.

**email:** Email for the account for the user.

**donateBook():** A method the users are able to donate a book.

**editPersonalData():** A method the users use to edit his own personal data.

**reserveBook():** A method the users use to reserve a book to borrow.

**borrowBook():** A method the users use to borrow books.

**Subscription users only:**

All the powers the registred users + :

subscriction: The number of months the user has on his subscription.

Transaction:

transID: The number of the transaction (in order to differentiate the transactions between them).

memberID: The number of the member (in order to differentiate the members between them).

bookID: The number of the book (in order to differentiate the books between them).

dateOfAquire: The date on which the book has been borrowed.

dateOfReturn: The date on which the book needs to be returned.

createTransaction(): The creation of the method.

deleteTransaction(): After the transaction is done, and the book returned the transaction will be deleted.

retrieveTransaction(): For the users and admin too see the transaction (the users can only see his own transaction).

**Book:**

author: The name of the author of the book.

name: The name of the book.

type: The genre of the book.

status: The status of the book (available or not).

price: The price of the book.

getBook(): A simple get method to get all the details about the book.

displayBookDetails(): A simple method to show those details of the book.

updateStatus(): A method to update the status of the book (available or not).

**6. HUMAN INTERFACE DESIGN**

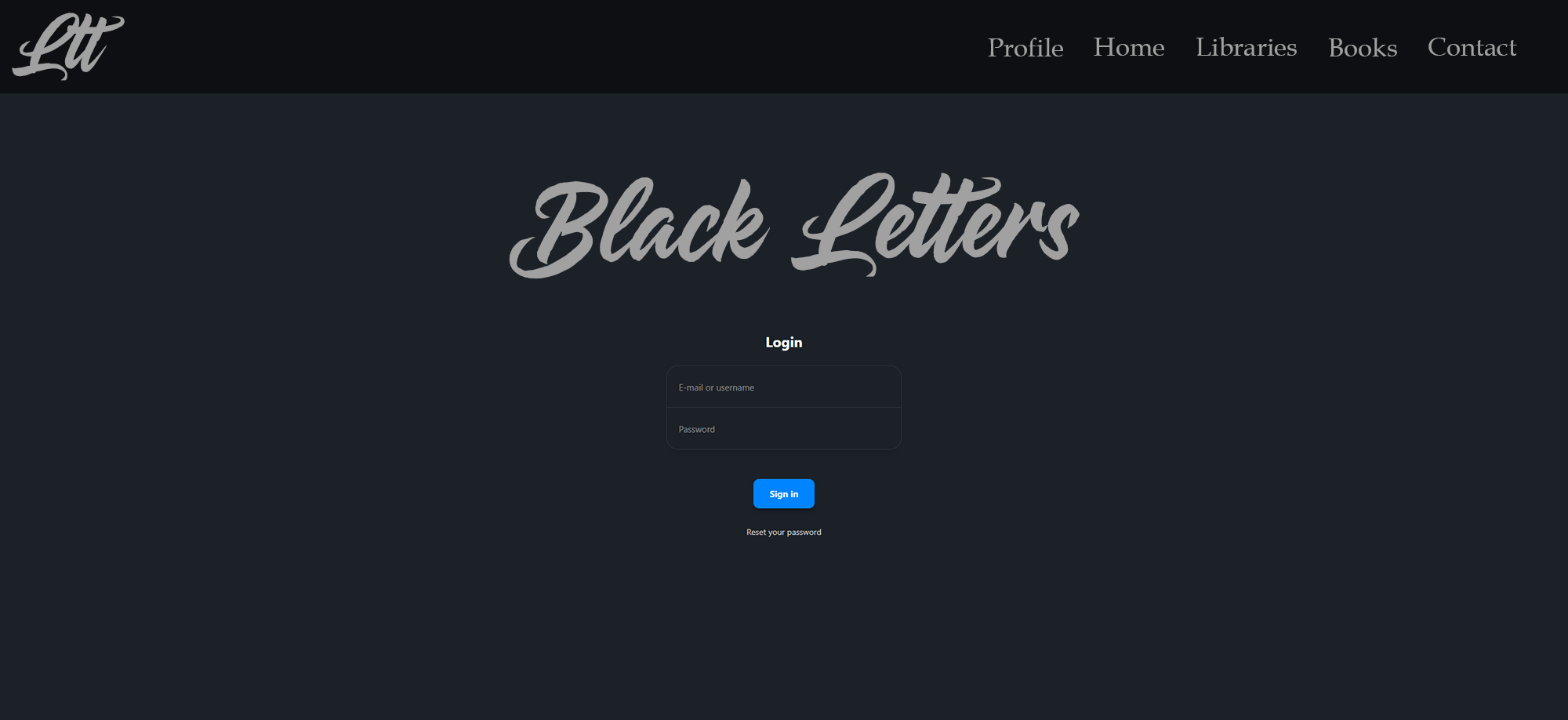
**6.1 Overview of User Interface**

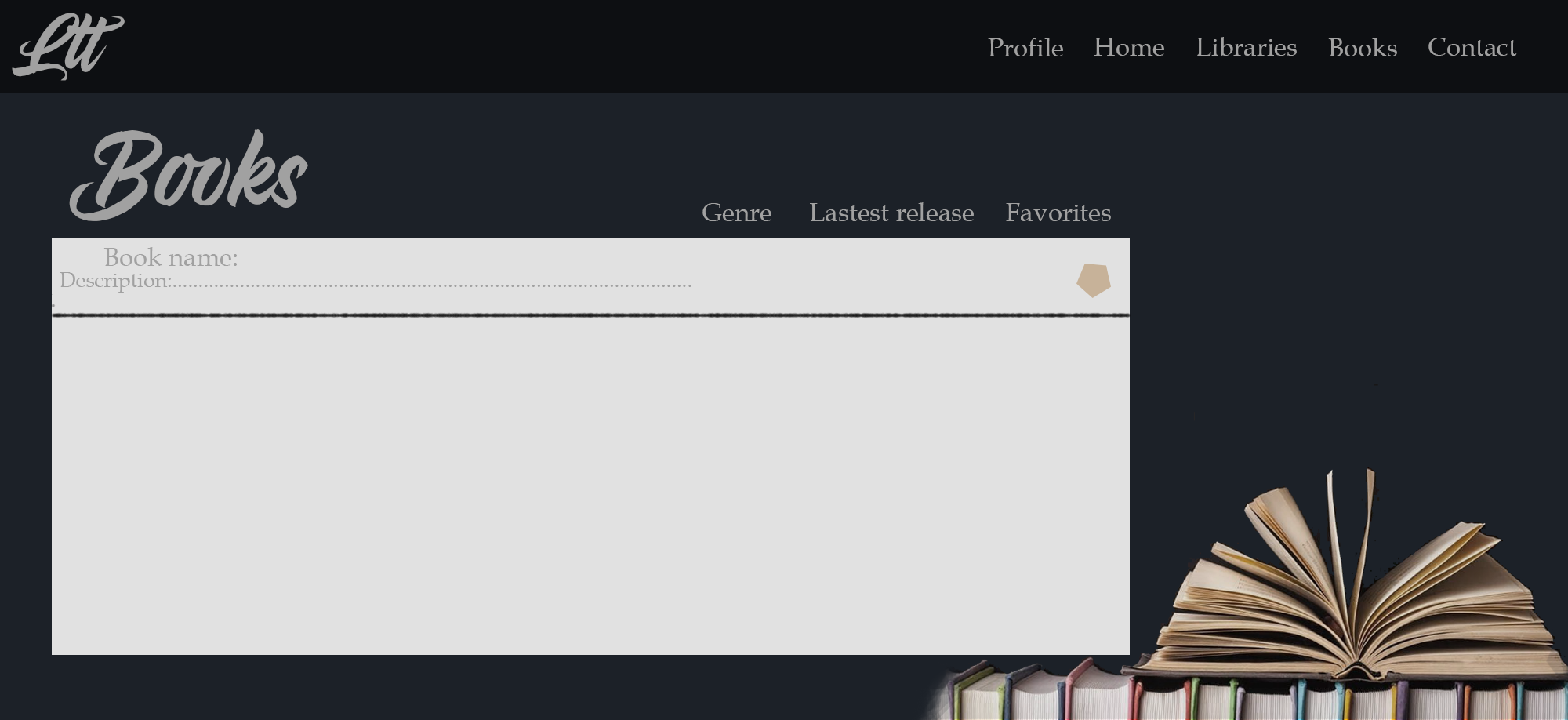
Describe the functionality of the system from the user’s perspective. Explain how the user

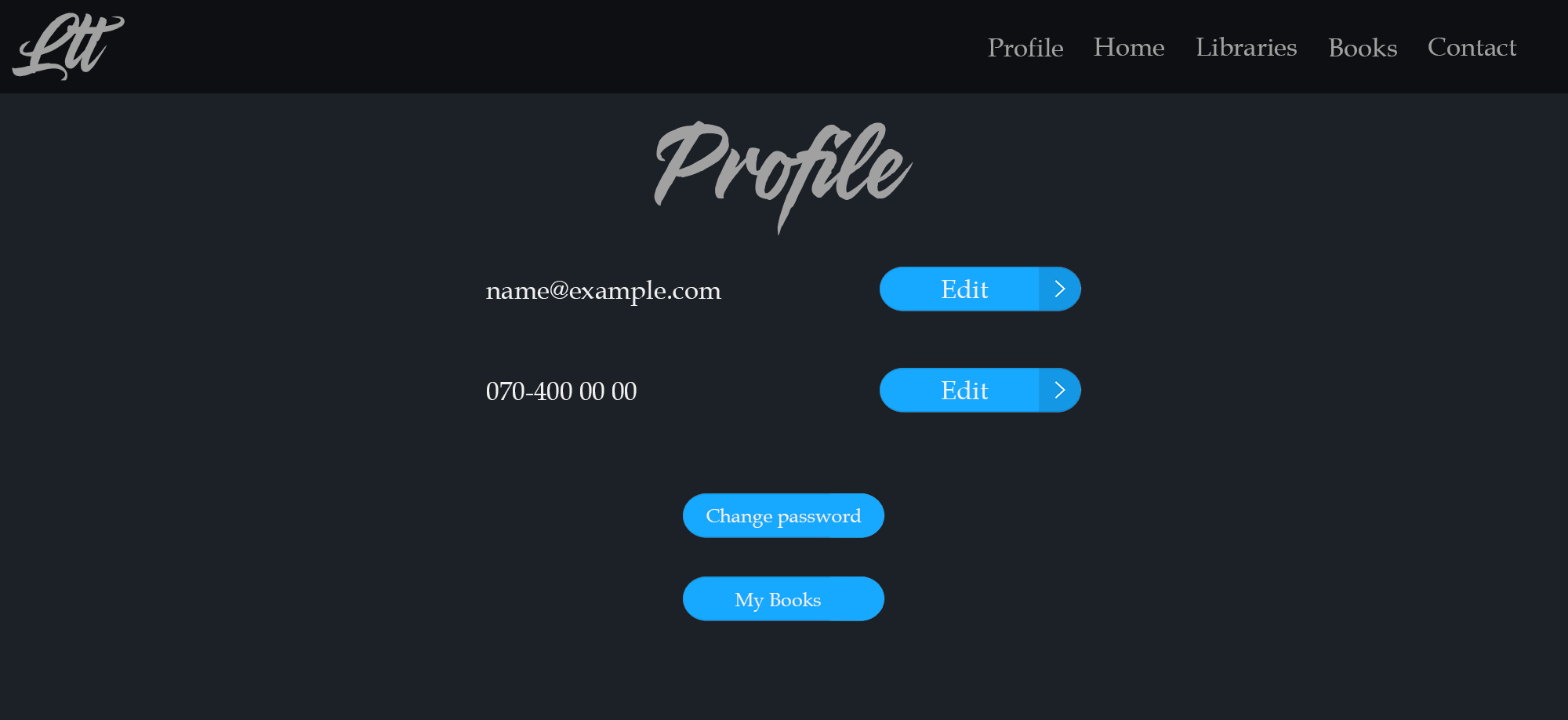
will be able to use your system to complete all the expected features and the feedback

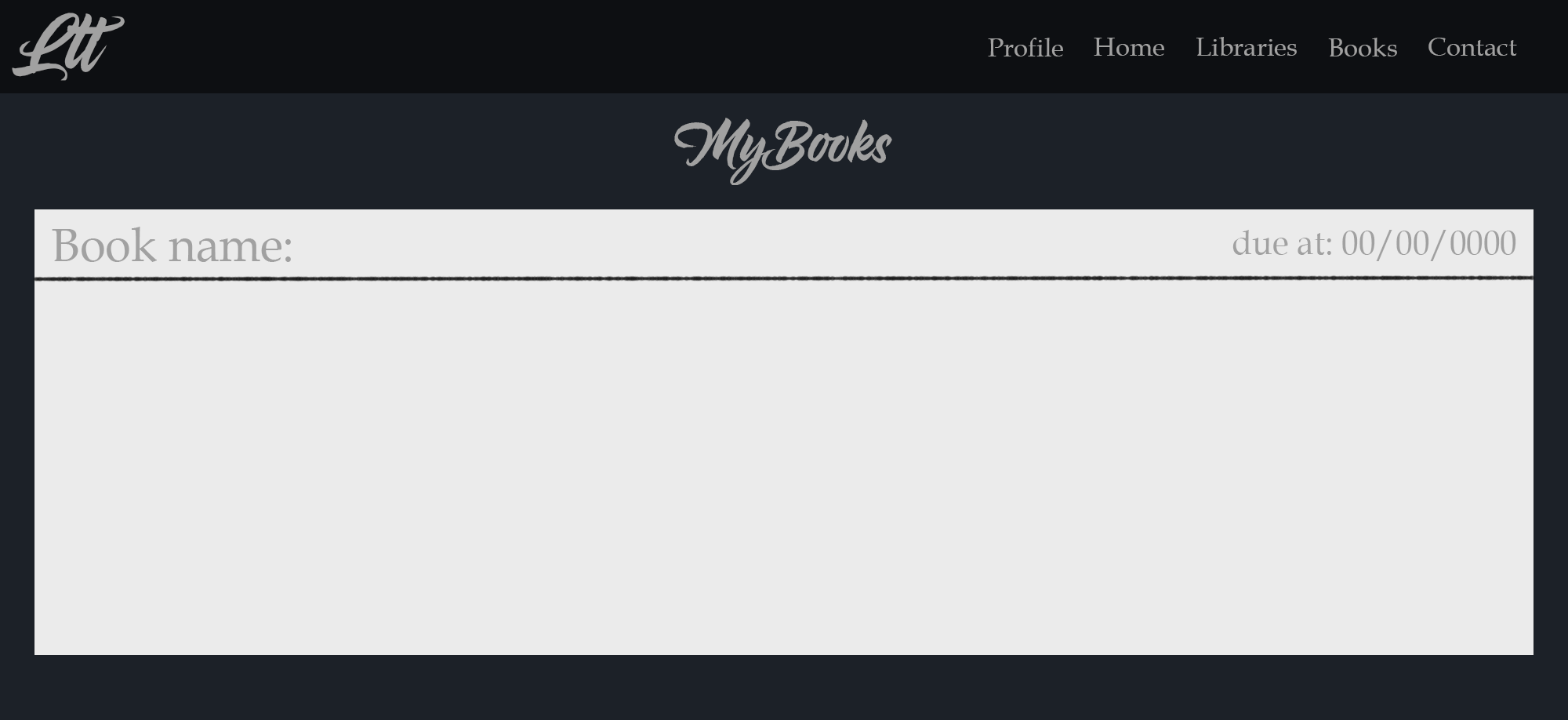
information that will be displayed for the user.

**6.2 Screen Images**









**6.3 Screen Objects and Actions**

Pic. 1 is the Login page which the user can acces.

Pic. 2 is the page for the books accesible from the store.

Pic. 3 is the profile of the registred users.

Pic. 4 is the page for the book that you borrowed, and the date at which the books need to be returned.

**7. REQUIREMENTS MATRIX**

Provide a crossreference

that traces components and data structures to the requirements in your

SRS document.

Use a tabular format to show which system components satisfy each of the functional

requirements from the SRS. Refer to the functional requirements by the numbers/codes that you

gave them in the SRS.

**8. APPENDICES**

This section is optional.

Appendices may be included, either directly or by reference, to provide supporting details that could

aid in the understanding of the Software Design Document.