Project Analysis

Katherine Galvin, Ryan Buckmaster and Maureen O’Malley

University of Maryland Global Campus

CMSC 495 6380 Current Trends and Projects in Computer Science

Group 4

Revision 1

7 June 2022

**Revision History Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Date** | **Description** | **Name** |
| 1 | 5/27/2022 | Created the Document with basic details/requirements; added input data and sources; added output data and destinations | Maureen O’Malley |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Input Data and Sources**

The user will generate input data from the GUI which will be passed to the database. The following input data can be generated from the user:

* **Registration**: The user will send username and password information from the GUI to the database to be added to the database.
  + **Username:** The username entered must be valid in number of characters and must not already be in the database.
  + **Password:** The password entered must be valid in number and types of characters.
* **Login**: The user will send username and password information from the GUI to the database to be compared to the database for an exact match (password is hashed in the database).
  + **Username:** The username entered must match a username in the database exactly.
  + **Password:** The password entered must match the password for the matched username in the database exactly.
* **Check out a book:** The user is able to specify book ID which will be matched to an ID in the database. The specified ID must be available to be checked out.
* **Renew books:** The user is able to specify book ID which will be matched to an ID in the database. The specified ID must match a book the user already has checked out.

**Output Data and Destinations**

When the user generates input data, output data is generated across the Glassfish server and causes information to be displayed to the user.

* **Registration**: The user will see through the GUI if their registration was successful or not by entering their desired username and password. The entered username is checked against the database to confirm it does not exist yet.
  + **Username:** The username entered must be valid in number of characters and must not already be in the database.
    - **Invalid username entered:** This is checked through front-end validation. An error dialog message box is generated indicating why the username is invalid.
    - **Username already in the database:** The username that was sent to the database matches a username which already exists, so the database sends back an indication of such. The front-end checks for this indication and displays an error dialog message box explaining why the username is invalid.
  + **Password:** The password entered must be valid in number and types of characters. This is checked through front-end validation. An error dialog message box is generated indicating why the password is invalid.
* **Login**: The user will see through the GUI if their login was successful by entering their username and password to be matched to the database exactly.
  + **Username:** The username entered must already match a username in the database.
    - **Username matches a username in the database:** The username that was sent to the database matches a username which already exists, so the database sends back an indication of such. The front-end checks for this indication and continues to check the database for an exact password match.
    - **Username does not match a username in the database:** The username that was sent to the database does not match a username which already exists, so the database sends back an indication of such. The front-end checks for this indication and generates an error dialog message box to explain why the login was unsuccessful.
  + **Password:** The password entered must match the password for the matched username in the database exactly.
    - **Password entered matches the password for the matched username exactly:** The password that was sent to the database matches the password of the previously matched username exactly, so the database sends back an indication of such. The front-end checks for this indication and allows a successful login.
    - **Password entered does not match the password for the matched username exactly:** The password that was sent to the database does not match the password of the previously matched username exactly, so the database sends back an indication of such. The front-end checks for this indication and generates an error dialog message box explaining why the login was unsuccessful. Additionally, the count of unsuccessful logins is incremented by 1. Too many unsuccessful logins will close the program.
* **Check out a book:** The user is able to specify book ID which will be matched to an ID in the database. The specified ID must be available to be checked out.
  + **The input book ID exists:** The book ID is sent to the database to be checked if it exists. The database sends back an indication that the book ID exists. The front-end checks for this indication and continues with the check-out process.
    - **The input book ID is able to be checked out:** The book ID is found in the database to see if it is able to be checked out. The database sends back an indication that it is able to be checked out. The front-end checks for this indication and generates a dialog message box to inform the user they have checked out the book.
    - **The input book ID is not able to be checked out:** The book ID is found in the database to see if it is able to be checked out. The database sends back an indication that it is not able to be checked out. The front-end checks for this indication and generates a dialog message box to inform the user they have not checked out the book because it is not available.
  + **The input book ID does not exist:** The book ID is sent to the database to be checked if it exists. The database sends back an indication that the book ID does not exist. The front-end checks for this indication and does not continue with the check-out process.
* **Renew books:** The user is able to specify book ID which will be matched to an ID in the database. The specified ID must match a book the user already has checked out.
  + **The input book ID exists:** The book ID is sent to the database to be checked if it exists. The database sends back an indication that the book ID exists. The front-end checks for this indication and continues with the renewal process.
    - **The book is already checked out by the user:** The book ID is found in the database to see if it is already checked out by the user. It is, so the database sends back an indication of such. The front-end checks for this indication and generates a dialog message box indicating that the book was renewed.
    - **The book is not already checked out by the user:** The book ID is found in the database to see if it is already checked out by the user. It is not, so the database sends back an indication of such. The front-end checks for this indication and generates an error dialog box indicating that the book cannot be renewed because it is not checked out already.
  + **The input book ID does not exist:** The book ID is sent to the database to be checked if it exists. The database sends back an indication that the book ID does not exist. The front-end checks for this indication and does not continue with the renewal process.

**Data Processing Step**

**Context Diagram: Data between the system and the user**

**Subsystem Diagram: All Data Listed**

**Subsystem Descriptions**

**Mapping from Subsystems to Requirements**

**Possible Enhancements**

**Risk and Risk Mitigation**

**Rubrics**

