**Software Design & Architecture**

**Assignment No. 1**



**Session: 2021**

**Submitted by:**

**Wali Muhammad 2021-SE-39**

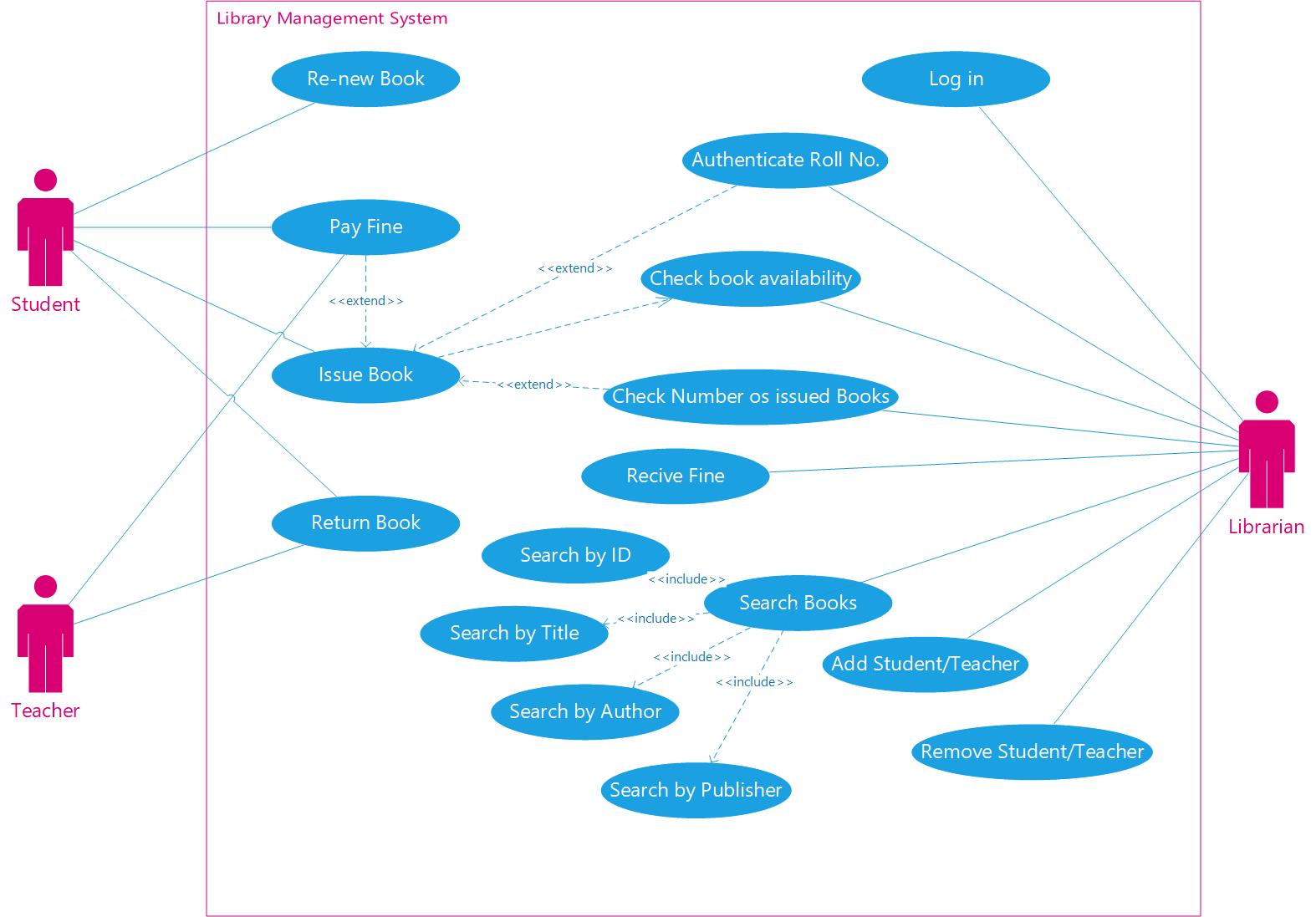
**Submitted to:**

**Sir Zeeshan Ramzan**

Department of Computer Science, New Campus

**University of Engineering and Technology Lahore, Pakistan**

**UML Use case Diagram**

****

**Use Case scenarios details**

|  |  |
| --- | --- |
| Name | Issue Book |
| **Primary Actor** | Student |
| **Goal** | To get book issued from library |
| **Trigger** | Student goes to librarian to issue book |
| **Pre-conditions** | * Valid student card * Librarian authenticates roll no. * Fine paid * Book should be available * Number of issued books less than 5 |
| **Post-conditions** | * Book marked unavailable * Student book issue count increased * 30 day countdown for return/renewal |
| **Main success scenario** | * Student shows student card to librarian * Librarian authenticates student using their roll no. * Librarian checks book availability * Librarian checks number of issued books by student * Student gets the desired book |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Security**   * Every student can only issue books using their own ID card only * Student cannot issue more than 5 books and for more than 30 days |
| **Exceptions** | * Student has no ID card * Student has not paid fine * Desired book is unavailable * Student has already issued 5 books * Roll No. cannot be authenticated by Librarian |

|  |  |
| --- | --- |
| Name | Re-new Book |
| **Primary Actor** | Student |
| **Goal** | Renewal of book issuance |
| **Trigger** | Student goes to librarian to renew the issuance of their book |
| **Pre-conditions** | * Valid student card * Librarian authenticates roll no. * Fine paid * Number of issued books less than 5 |
| **Post-conditions** | * Student book issue count increased * 30-day countdown for return/renewal is reset |
| **Main success scenario** | * Student shows student card to librarian * Librarian authenticates student using their roll no. * Librarian checks number of issued books by student * Librarian renews the book issuance for the student |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Security**   * Every student can only renew books using their own ID card only * Student cannot issue more than 5 books and for more than 30 days |
| **Exceptions** | * Student has no ID card * Student has not paid fine * Student has already issued 5 books * Roll No. cannot be authenticated by Librarian |

|  |  |
| --- | --- |
| Name | Return Book |
| **Primary Actor** | Student |
| **Goal** | To return the book to the library |
| **Trigger** | Student goes to librarian to return issued book |
| **Pre-conditions** | * Valid student card * Librarian authenticates roll no. * Student returns book before 30 days of issuance |
| **Post-conditions** | * Book marked available * Student book issue count decreased |
| **Main success scenario** | * Student shows student card to librarian * Librarian authenticates student using their roll no. * Librarian checks how long since book has been issued * Book is returned to the library |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Security**   * Student must return issued books within 30 days |
| **Exceptions** | * Student has no ID card * Student returns book after 30 days * Roll No. cannot be authenticated by Librarian |

|  |  |
| --- | --- |
| Name | Log in |
| **Primary Actor** | Librarian |
| **Goal** | To be logged into the library management system |
| **Trigger** | Librarian clicks on the login feature of the Library Management System |
| **Pre-conditions** | * Account already created |
| **Post-conditions** | * Account logged in, allowing librarian to perform the various tasks |
| **Main success scenario** | * Librarian enters information into the fields * Correct information is entered * Librarian is logged in |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Security**   * Only librarian can login to the system and perform the functionality |
| **Exceptions** | * Librarian’s account is not created |

|  |  |
| --- | --- |
| Name | Authenticate Roll No. |
| **Primary Actor** | Librarian |
| **Goal** | To authenticate a student’s roll no. to let them issue or renew a book |
| **Trigger** | Librarian clicks the authenticate roll no. after a student shows their card |
| **Pre-conditions** | * Librarian must be logged in * Student must show ID card |
| **Post-conditions** | * Student roll no. is authenticated * Student can issue/renew book |
| **Main success scenario** | * Student shows student card to librarian * System checks for Roll no. in the database * Roll No. is authenticated |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Security**   * Student is only allowed to issue or renew books after their Roll No. is authenticated |
| **Exceptions** | * Librarian is not logged in * Student does not have ID card |

|  |  |
| --- | --- |
| Name | Check book availability |
| **Primary Actor** | Librarian |
| **Goal** | To get to know whether or not the desired book is available or not |
| **Trigger** | Librarian checks book availability to issue it for a student or teacher |
| **Pre-conditions** | * Librarian must be logged in * Book must exist in the system |
| **Post-conditions** | * Book availability status is known * Details of student/teacher who has issued the book are shown |
| **Main success scenario** | * Student asks for a book * Librarian searches for book information * Librarian checks the availability status of the book * Librarian gets the information |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily check the availability of a book |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a book that does not exist in the system |

|  |  |
| --- | --- |
| Name | Check no. of issued books |
| **Primary Actor** | Librarian |
| **Goal** | To get to know how many books have been issued by a particular student |
| **Trigger** | Librarian checks student’s record to know how many books have been issued by them |
| **Pre-conditions** | * Librarian must be logged in * Student information must be present in the system |
| **Post-conditions** | * Number of books issued by a student is known by the librarian * Librarian can know if more books can be issued to the student |
| **Main success scenario** | * Librarian gets student Roll no. * Librarian searches for student’s information * Librarian finds number of issued books |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily check how many books have been issued by any student   **Security**   * Students cannot issue more than 5 books |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a student that does not exist in the system |

|  |  |
| --- | --- |
| Name | Receive fine |
| **Primary Actor** | Librarian |
| **Goal** | To receive the fine payment to allow a student to issue or renew books |
| **Trigger** | Student pays the fine |
| **Pre-conditions** | * Fine must exist in the system * Student must pay the fine |
| **Post-conditions** | * Librarian receives the fine payment * Student is allowed to issue or renew books |
| **Main success scenario** | * Librarian issues fine to student * Student pays the fine * Librarian receives fine |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily check which student has been fined   **Security**   * Fined students cannot issue or renew books |
| **Exceptions** | * Librarian is not logged in * Student has not paid the fine |

|  |  |
| --- | --- |
| Name | Add student |
| **Primary Actor** | Librarian |
| **Goal** | To add a new student into the library’s database to allow them to issue or renew books |
| **Trigger** | Student asks Librarian to be added into the library management system |
| **Pre-conditions** | * Librarian must be logged in * Librarian must have student information * Student should be a new one |
| **Post-conditions** | * Student’s information is added to the database * Student can issue or renew books |
| **Main success scenario** | * Librarian checks if student already exists in database * Librarian asks the student the required information * Librarian adds the student into the record |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily add student into the database (rather than manual paper records) |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a student that already exists in the system * Librarian does not have sufficient information |

|  |  |
| --- | --- |
| Name | Add teacher |
| **Primary Actor** | Librarian |
| **Goal** | To add a new teacher into the library’s database to allow them to issue or renew books |
| **Trigger** | Teacher asks Librarian to be added into the library management system |
| **Pre-conditions** | * Librarian must be logged in * Librarian must have teacher’s information * Teacher must not be already present in the database |
| **Post-conditions** | * Teacher’s information is added to the database * Teacher can issue books |
| **Main success scenario** | * Librarian checks if teacher already exists in database * Librarian asks the teacher the required information * Librarian adds the teacher into the record |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily add teacher into the database (rather than manual paper records) |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a teacher that already exists in the system * Librarian does not have sufficient information |

|  |  |
| --- | --- |
| Name | Remove student |
| **Primary Actor** | Librarian |
| **Goal** | To remove a student from the library’s database to no longer allow them to issue or renew books |
| **Trigger** | * Librarian gets a reason to remove a student from the records |
| **Post-conditions** | * Student’s information is removed from the database * Student can no longer issue or renew books |
| **Main success scenario** | * Librarian checks if student already exists in database * Librarian searches for the roll no. * Librarian removes the record of the student |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily remove student from the database (rather than manual paper records) |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a student that does not exist in the system * Librarian does not have sufficient information |

|  |  |
| --- | --- |
| Name | Remove teacher |
| **Primary Actor** | Librarian |
| **Goal** | To remove a teacher from the library’s database to no longer allow them to issue or renew books |
| **Trigger** | * Librarian gets a reason to remove a teacher from the records |
| **Post-conditions** | * Teacher’s information is removed from the database * Teacher can no longer issue or renew books |
| **Main success scenario** | * Librarian checks if teacher already exists in database * Librarian searches for the ID * Librarian removes the record of the teacher |
| **Alternate flow** | No Alternate flow |
| **Qualities** | **Usability**   * Librarian can easily remove teacher from the database (rather than manual paper records) |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a teacher that does not exist in the system * Librarian does not have sufficient information |

|  |  |
| --- | --- |
| Name | Search book |
| **Primary Actor** | Librarian |
| **Goal** | To find a book in the database |
| **Trigger** | * Librarian gets needs to search the book for a teacher, student or for some other reason |
| **Post-conditions** | * Librarian finds the desired book |
| **Main success scenario** | * Librarian checks if book exists in database * Librarian searches for the ID * Librarian finds the desired book |
| **Alternate flow** | **Search by title**   * Librarian searches for the book’s title in the database * Desired book is found   **Search by author**   * Librarian searches for the book’s author in the database * Desired book is found   **Search by publisher**   * Librarian searches for the book’s publisher in the database * Desired book is found |
| **Qualities** | **Usability**   * Librarian can easily find and manage books’ records (rather than paper-based records) |
| **Exceptions** | * Librarian is not logged in * Librarian checks for a book that does not exist in the system * Librarian does not have sufficient information |

**THE END**