|  |  |
| --- | --- |
|  | **CS 313: Intermediate Computer Programming**  **Project Work 2 (Wednesday, 17th April 2018)** |

**PART A – PROJECT DESCRIPTION**

1. **Project Details**

***Project Title:*** Library Management System

***Team Name:*** Gbet-Nams-Library-System

***Team Members:*** Nana Ama Atombo-Sackey and Jean-Sebastien Dovnon

***Description:*** The project is a library system that aims at allowing users (librarian, student, and lecturer) to perform some functions such as borrowing, viewing, updating and deleting a book from a system. However, is the librarian is the only user that has the access to update and delete a book from the system. The student can only read or view and borrow from the system.

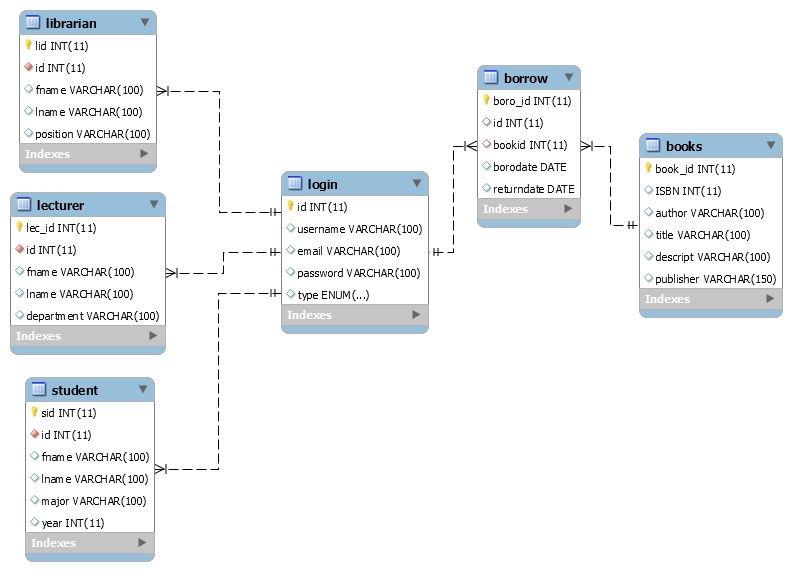
Just as Ashesi has a library system known as OPAC, which allows students to view only books that have been borrowed or issued out, and that of the librarian been updating a user’s books that have been borrowed, Gbet-Nams-Library-System also works similarly, however, the challenges associated with OPAC is that a user cannot borrow a book from the system until the librarian checks in the books for the user. Our system helps users to borrow without verification from the librarian.

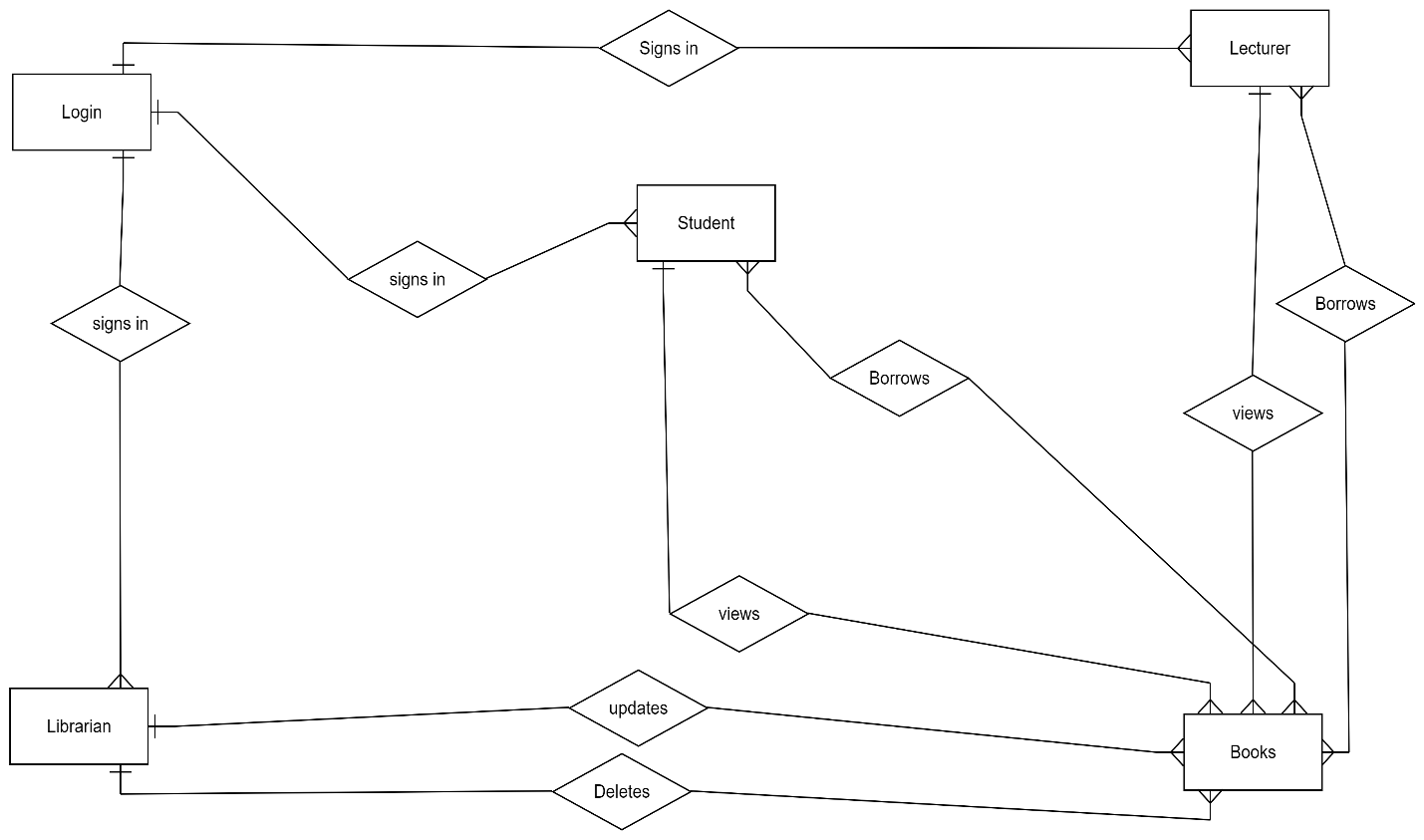
1. **Software Engineering Cycle**
2. ***Requirement Specifications***

The system is supposed to allow Lecturers and students to view books in order to borrow books from the system. In addition, if any of users (lecturer and student) is not registered on the system, the user should be able to register and have access to the books in the system. On the other hand, a librarian should be able to update books to the system for borrowing and remove books from the system.

1. ***Analysis and Design***

Below is representation of how database of the system and the ER representation of the system. Basically, a lecturer, a librarian, and a student can login in into the system. Based on the user that logins, a user is able to perform the specific functionalities that it can perform as specified ib the requirements. The various entities in the system are the Student, Lecturer, Books and Librarian with their various attributes.





1. ***Implementation***

This system was built using the Java language and SQL (database). The SQL aspect of the program was used to create tables namely (lecturer, student, librarian, login, books, and borrows). These tables had various attributes with their respective foreign keys and primary keys. The purpose of the database was to store data in a persistent and efficient way for it to be used and accessed all the time.

Due to our proficiency in the Java language, Java was used for both the front end and the logical aspect of the system. The views or front end of the system was done using Java FX which works like other IDEs, but faster and intuitive in making GUI because of its support for markup in FXML and CSS. In addition, Scene builder was used in conjunction with the Java FX to design the interface. It was chosen because it gives more professional and advanced features as compared to the swing.

However, this project was implemented using MVC pattern (Model-View-Controller). MVC was used because it was to help the clarity of our design, to ensure modularity that is, changes made to an aspect of the program does not affect the other and also to have multiple views.

The Model provided access connection to the database. It also had various functionalities such as inserting books, registering a student, registering a lecturer, registering a librarian, updating a book, deleting a book, borrowing a book and a logging in and a returning book.

For the controller, the use of the Java FX made it possible for each controller to have a view. This was done because the Controller controls the flow of data based on the inputs from the user in the view which uses the functionalities in the model.

1. ***Verification and Validation***

The application is a very robust application because, with the use of try and catch or exception handling to each of the functionalities in the model and inputs from the view, a user is always made to enter the right input before he or she can continue to use the application.

1. ***Evolution (Future Work)***

With regards to this project, we would like to have features such as a user being able to download books be it in the form a softcopy (pdf’s etc). In addition, more functionalities such as making users aware of when to return the books borrowed, fining students who do not return the books when due and many others could be added to make the system more diverse.