**National University of Sciences and Technology (NUST)**



**Object Oriented Programming**

**Assignment 2**

**Project Proposal**

**BESE 12-B**

**Lecturer**

**Dr. Ahsan Saadat**

**Group Members**

|  |  |
| --- | --- |
| **Name** | **CMS ID** |
| Muhammad Bilal | 389994 |
| Muhammad Ashhub Ali | 380078 |
| Abdul Arham | 374696 |

**Task – 1**

**Provide a 1 or 2 page proposal of your selected project, explaining the overall aims and achievable targets. You should concisely explain the real-world problem you are trying to solve. Please mention any platforms, languages in addition to Java that you plan to use.**

**Project Name: Library Management System**

**Project Details:**

The aim is to build a complete system that simulates the workings and logistics of a generic library. Our program will have a plethora of dimensions which include, but are not limited to:

1. An Employee Management System
2. A Library Membership System
3. Library Stock and Book Collection System

To start, we intend to make our project such that it intricately mimics the daily occurrences inside a library, using samples and templates. Our goal is to make it workable, such that it can be implemented and used by any public or private library in the region, e.g., NUST’s Main Library.

**Basic Implementations & Features:**

**Databases:**

In today’s world, almost everything is connected digitally. So it would be very effective for libraries to use a management system to keep a record of all sorts of information like books, members, and staff. A key component of our project would be to use databases for real-time storage of all this data. Using MySQL, the data will be stored in the form of tables and blocks in a database and accessed through the JDBC (Java Database Connectivity) Library.

In general, user information such as Name, ID, Books Issued, Current Status, fines/ charges will be stored. Whereas, Employee Databases will include employee types (Librarian, Assistant, Technician, Janitors, etc.), their attendance, performance, and rank. For books, the number of copies, damaged items (if any), current availability, new arrivals, etc., will be saved in these databases.

**Graphical User Interface**

We intend to use and implement Java Swing Library for the outlook of our System. The interface created will allow the users (both the staff and the members) to interact freely with the application. We will set up navigation bars to access each page/ window. These pages or windows include but are not limited to Homepage, Admin Login, Library Stock, User Login, and Comments/ FAQs Section. To make it more appealing, we will add simple animations such as those enabled on hovering, etc.

**Admin Controls**

A core component of our application will be Serial Controls for the Administration (Librarian, in general). These controls will enable the Admin to cancel, renew, provide new membership, or fine a customer. The Librarian will have access to all the Library Stock (inflow as well as outflow).

**Dynamic Search**

This feature will allow the users to search for a book by entering either the Author Name, Book Title, Genre, or ISBN Number.

**Request a Book**

If a certain book is not available in the Library Stock, the user can request for new books, which will be conveyed to the Management to order in the future. The priority of each book will be proportional to its demand by the users.

**Book Suggestion**

The application will suggest books to each user depending on the Genre they frequently issue, or books read by other users with similar tastes.

**Upload Profile Picture**

The Library Management System will also enable the user to upload their profile picture to their account. The picture will then be displayed along with other account details.

**Task – 2**

**Provide group member details along with task distribution, i.e. consider your group as a team undertaking this project for timely delivery to a client. You should clearly allocate primary roles to each group member.**

|  |  |
| --- | --- |
| **Group Member Name** | **Role / Tasks Allocated** |
| Abdul Arham | * File Handling * Implementation of Administration Controls * Design of Database * Integration of Database * Frontend-Backend Integration |
| Muhammad Ashhub Ali | * Design of Graphical Interface * Dynamic Search implementation * Employee Management System |
| Muhammad Bilal | * Backend refinement * Exception handling * Library Stock and Management |

**Task – 3**

**In this section, you need to elaborate your project idea by mapping the below mentioned OOP concepts to your project. Please mention briefly how a particular concept will be applied in your project. This mapping may change in next few weeks as you build and expand your project, but at this stage it should be elaborate enough to quantitatively monitor your project progress on weekly basis.**

|  |  |
| --- | --- |
| **Topic** | **Application in your project** |
| Flow Control | Specific blocks of the program will be executed as preferred by the user, e.g., the Librarian may want to add new books or remove damages books. |
| Composite Data Types | Creation and Implementation of Administration Class |
| Setters & Getters, Default and No-Argument Constructors | In all created Classes, subclasses and interfaces |
| Method (Function) Overloading | In Admin Superclass, Employee Class, |
| Static Class Members, this Reference | In the superclasses at the top of each hierarchy e.g., in Administration class, in Employee class and in Member class |
| Arrays and strings / functions | As datafields in most of the classes |
| Inheritance In Java | Hierarchal inheritance will be implemented explicitly in all parts of the code |
| Super classes and sub classes | Hierarchal inheritance will be implemented explicitly in all parts of the code |
| Method Overriding, Constructors in subclasses | Method overriding shall be used in all relevant subclasses, especially to invoke toString() method |
| Polymorphism | Will be used while creating instances of Student subclass, storing it in Member superclass reference array |
| Abstract Classes & Methods | Will be implicitly used in interfaces, and explicitly wherever required. |
| Final Methods and Classes | Will be used to invoke methods that display information about the library e.g., a Greeting Message, Library rates and fee etc. |
| Interfaces and Abstract Classes | General interfaces for Administration, Employee, Student superclasses will be created |
| Exception Handling | In all exception-prone parts of the code |
| Graphical User Interfaces | Swing used to create a GUI to represent a Graphical Interface that the student/employee interacts with |
| File Manipulation | Used to save data and credentials of each and every user that interacts with the system, connected to a local database |