CSCI 217: Advanced Computer Programming
Project Ideas

Idea #1: Library System

Write a program that simulates a library system such that the librarian can add/remove books that are categorized within 5 types

- Biology
- Maths
- History
- chemistry
- politics

Also the system allows borrowing and returning back the books in addition to listing the books status.

The user should be able to apply one of the following scenarios:

1- Adding a book
2- Removing a book
3- Borrowing a book
4- Returning a book
5- View the library status
6- exit

Every book has the following characteristics:

Unique id (generated by the system)
Name
Category
Borrowed / not borrowed
Borrowing period
Borrowing date

When the system runs the user should enter the capacity of the library in which how many books the library can contain.

Adding a book:
When the user requests adding a book in the system, you will need to check whether the library can accept the book or not i.e. the library cannot accept more books if there is not enough space, then the user enters the data about the book to be inserted.

Removing a book:
When the user requests removing a book from the system, you will need to check whether this book is not currently borrowed, then allows the user to enter the id of the book to remove it, if the book is currently borrowed, the user should know when he will be able to remove it.

Borrowing a book:
When the user requests borrowing a book, you will need to check whether this book is not currently borrowed then allows the user to enter the id of the book to borrow it and the borrowing period of this book, if the book is currently borrowed, the user should know when he will be able to borrow it.

Returning a book:
When the user requests returning a book, you will need to check whether this book is currently borrowed then allows the user to enter the id of the book to return it back.

View the library status:
When the user requests viewing the status of the library, he should be able to see all the books in the library, the status of each book (borrowed or not), for the borrowed books show the borrowing period and borrowing date, the category of the book, number of books for each category and total number of borrowed books in the system.

Exit
When the user requests leaving the system, the program should terminate.

Idea #2: Seat reservation in a theatre

Write a program that enables the user to reserve seats in a theatre. The theatre has 3 classes of reservations from the first to the third, with the seats distributed as follows:

1. First class: Has 20 seats (4 rows x 5 column), and the ticket costs 50 L.E.
2. Second class: Has 40 seats (8 rows x 5 columns), and the ticket costs 15 L.E.
3. Third class: Has 50 seats (10 rows x 5 columns), and the ticket costs 10 L.E.

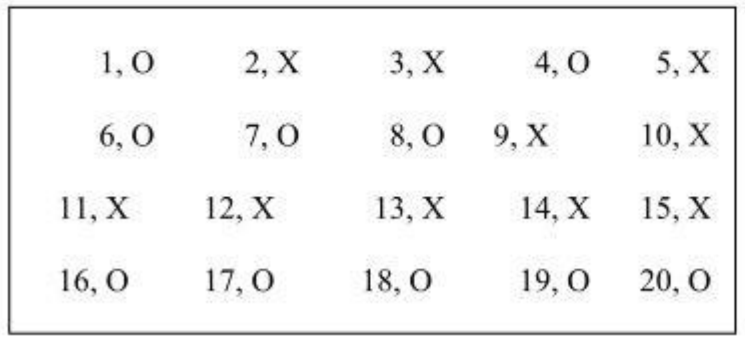
The user should be able to apply one of the following scenarios:

1- Ticket reservation
2- Ticket Cancellation
3- Reset the theatre reservations
4- exit

Ticket reservation:

This process is done through the following sequence:

1. The user chooses the class he wants to make a reservation in.
2. A screen displaying the current status of that class seats is shown, with every seat having a unique number (starting from 1), and the empty seats are displayed having an O mark, and the reserved seats are displayed having an X mark as follows.



```
    1, O      2, X      3, X      4, O     5, X

    6, O      7, O      8, O   9, X       10, X

 11, X     12, X      13, X     14, X     15, X

 16, O     17, O     18, O      19, O     20, O
```

Figure 1

3. The user enters the IDs of the seats he want to reserve.
4. If the user enters an ID for a seat that is already reserved, an error message should appear to him.

5. After the user finishes entering the IDs of the seats he wants to reserve, a sum of the money he should pay must appear on the screen.

Ticket cancellation:

This process is done through the following sequence:

1. The user enters the class and the ID of the seat he wants to undo the reservation for.
2. If that seat was already empty (this means that he used a wrong ID number), then an error message should appear to him.
3. If the seat was actually reserved, then a message displaying the sum of the money he should receive back should be displayed.

Reset the theatre reservations

Clear all the reservations and setting the theatre seats to be empty

Exit

When the user requests leaving the system, the program should terminate.

Idea #3: Pharmacy ordering project

Write a menu driven program that enables the user to call and order drugs from a Pharmacy where the drugs are categorized to be:

1- cosmetics
2- prescription drugs
3- other

When the program runs it ask the user to choose one of the following:

1. Add drug
2. Remove drug
3. Place an order
4. Get the total sales for one day
5. Exit

For the first time the system runs the user should enter the capacity of the pharmacy in which how many drugs the pharmacy can hold. And loop to display the same menu until 5 is selected.

Each of these services is described below:

Add drug

Add drug name, id, price, category, available quantity, the drug is added only if the pharmacy is still having place for adding a new drug.

Remove drug

When the user requests removing drug, allow the user to enter the id of the drug to remove it.

Place an order

When a user calls, asks for the availability of a certain drug the program search for the drug by its ID. Get unit price, print its price and ask for the prescription in case that the drug needs so, for the cosmetics, they are sold at price of 1.2 of its original price.

Get the total sales for one day

For every order, the program stores the quantity and total price so that at the end of each day, it can calculate the total sale.

Exit

When the user requests leaving the system, the program should terminate

Another Ideas with abstract description:

1. Attendance Management System:

This project is an application that manages the attendance of any student in school or college, also employees in an organization. It has an admin feature that holds the access to do any kind of changes like update, delete, and add user records to the list. Various users of the application where users can access are admin, student, faculty.

2. Banking Application:

In this modernized world, where time is money, everyone has got the habit of doing their tasks online. Within a click, a task is done. You get this application to make transactions just by sitting in your comfort zone. Every operation like money transfer and balance inquiry can be done in seconds.

3. ATM Simulation System:

This contains two sections – admin and user sections.

The admin mode is responsible for controlling the entire system like adding and deleting accounts and updating the records of the user.

The user-mode takes care of the deposit, withdrawal, and checking of the account balance. The whole process of this system is automated, from PIN (Personal Identification Number) validation to the transaction.

The card details will be secured enough by encrypting the details in the database and will only be accessible to the authorized user. The UI of the application contains a profile of the user, accounts added to it, and an option to withdraw, deposit and update details of the account.

4. School Management System:

This application's objective is to help the school management system in managing the data easily. The manual system could be a complicated one when it comes to keeping the records so, there comes the role of this project. It holds personal records of students, teachers, and staff. This system contains modules for different roles be it admin, student, staff, and teacher. Here, the admin has to be responsible for maintaining the records like adding users, updating the details of the user, and deleting the user's profile.

Notes:

- Snapshots for the simulation and results should be included.
- Error checking: the user inputs should be considered.
- Use Java programming languages to develop **GUI** Application.
- Project documentation must be provided.
- Failure to include all project source files will result in lost points.
- You will be subject to resolving / handling issues that will possibly appear during programming.
- Minimum number of Students per team is 2, maximum is 4.
- Any other project ideas are welcome and subject to doctor approval.
- **Due date is 20/5/2023 at 11:59 pm**
- The discussion of the project will be held in the lab at (21/5/2023 and 25/5/2023)