Design and Analysis of Data Structures and Algorithms

Coursework Overview

This assignment assesses the following module learning outcomes:

- ✓ Analyze requirements and select appropriate solutions
- ✓ Design programmes that use appropriate data structures.
- ✓ Implement data structures and the algorithms that maintain them, allowing for secure processing of the data.

This is an individual assignment.

Working on this assignment will help you to develop the ability to relate organizational requirements into design choices for the storage and management of data. It will also help you practice your programming skills

Coursework Specification

An online subscription service company allows members (users) to borrow and use the following types of items for up to a week each time.

- Movies
- Music
- EBooks
- Licenced Software
- Games
- Courses

The conditions of usage are as follows:

- Each Item can be booked by up to 10 different users concurrently. The company might offer multiple instances of popular items, so more than 10 users can actually borrow /use the same title by accessing different items.
- Each user will be able to book up to 5 items at any one time

- Each user will be able to queue for up to 5 requested items at any one time –depending on how many items the user is already using at the time. The total number of items (borrowed and requested) cannot exceed five per user at any one time.
- Each item can be booked for use for a week and then it is automatically retrieved
- A user cannot place a request on an item that they are currently using.

Tasks:

1. Select appropriate data structure(s) to accommodate effective storage of data for the above subscription service.
2. Identify suitable algorithms that would allow the company to run the service as per the above conditions.
3. Produce Pseudo Code or any other form of software design for the software that you are going to implement.
4. Build the software to manage item bookings and user usage as per the above conditions.
    - ✓ The software must allow the user to borrow an item that is available
    - ✓ The user may place a request for an item that is currently unavailable
    - ✓ The software must automatically remove an item from a user's access upon the seven-day period expiring
    - ✓ The software must enforce all conditions of usage specified above.
    - ✓ The system should allow the user to create new users
    - ✓ The system should allow the user to enter data as manual input or to read them from a CSV / text file.
    - ✓ The system should be able to search for an item based on the title or the Item Number.
5. Justify your choice of data structures and algorithms utilized and evaluate the outcome of your work in the format of a short report of about 800 words.

NOTES: Please note the following key instructions

Sample data will be provided, but you may feel that you want to use your own or supplement the given data. This is acceptable as long as the structure of the data is kept as shown below.

Structure of data to be used

Item – Item No, Title, Number available

Member – Member ID, Member Name

Deliverables

One folder in zip format (only) must be uploaded via the relevant link on the Moodle page. The link will be available two weeks before the due date and will be communicated to students via an announcement.

You can reuse code and design that you have produced for one of the tasks here to fulfil the requirements of another task.

The folder must contain:

- ✓ The software modules that you have produced.
- ✓ A word / text file that provides clear instructions as to how to run the program – This is particularly important where more than one program files have been submitted.
- ✓ All the data files packaged in a way that the software will run and access them without any problems.
- ✓ A short video demonstrating your software running and fulfilling each of the tasks in the correct order.
- ✓ A word / PDF file with your justification of the design choices.
- ✓ A word /PDF file with your pseudocode

Marking Criteria

The following table gives details of the marking criteria for this coursework.

Marks will be awarded for clear rationale justifying design choices.

Clarity in the pseudo code submitted allowing to map the full logic of the solution implemented is expected.

Code must be well structured, appropriately commented, neat and efficient. Clear use of functions and reduced repetitions of blocks of code are expected.

The use of GUI or other user interface will not attract any specific marks, but simplicity and efficiency of its design will be considered when awarding for an overall efficient system developed.

NOTE – No hard coded data will be allowed. Hard coded data in the submitted work will result in the work marked at 0 (zero).

Rubric

Aspect of assessment and proportion	**No Work**	**Incomplete**	**Acceptable**	**Good**	**Very Good**	**Excellent**
	0%	**0%-50%**	**50%-60%**	**60%-70%**	**70%-80%**	**80-100%**
1 Structure & Algorithm (30%)		Data structure or algorithms is not appropriate and some of the important aspects ignored (eg: space and time complexity).	Data structure and algorithms are described but some aspects are not very clear.	Data structure and algorithms are well described but the proposed solution is not very appropriate.	Well, described but either the proposed data structure or the proposed algorithm is not the most appropriate solution.	The proposed Data structure and algorithm is well explained and can be considered as very appropriate and an excellent solution.
2 Pseudo code (10%)	No pseudo code	Code is not matching to the proposed data structure and algorithm.	In general code reflects the proposed data structure and algorithm but difficult to understand.	In general code reflects the proposed data structure and algorithm and easy to understand.	The code is well structured, the functionality is full and there is no difficulty in understanding. But there is so room for improvement	Exact reflection of proposed data structure and algorithm. Code is well written and has a professional look.
3 Functionality (30%)	Not functioning	Most of the functionalities are lacking.	Most functionalities are working but	Software is functioning and as per the	Software is functioning as per the requirement	Software is perfectly functioning as per the

			requirement not fully met.	requirement, but an alternative approach could have been better function	but not perfect.	requirement .
4 Java Code (20%)	No Code	There are some codes that demonstrates understanding of the problem but not functioning.	The code works but possibly without observing all constraints.	The code works but the efficiency is not the expected one.	The code is well structured, and the functionality is full.	The code is adorned with useful comments, is efficient and intelligent.
5 Evidence of testing (10%)	No Evidence of testing	Some testing done not but little evidence of testing.	Testing is well documented , but test data used are not very appropriate or testing not sufficient and some aspects of testing not considered.	Testing well documented but has 2 of the following shortcomings: -irrelevant test data - testing not sufficient - some aspects not tested	Either test data used is not relevant, not sufficient or some aspects not tested.	Tested all required functions with relevant and sufficient test data.