



DESCRIPTION OF COURSEWORK

Course Code	CST207
Course Name	Design and Analysis of Algorithms
Lecturer	Najla'a Ateeq Mohammed Draib
Academic Session	2024/09
Assessment Title	Group Project – (20%)

A. Introduction/ Situation/ Background Information

The purpose of this project is to develop a **Library Management System** that automates library operations such as managing book inventories, handling user transactions, and maintaining records. The system includes separate control panels for administrators and library users, ensuring efficiency, accessibility, and data integrity through robust error handling. Additionally, the project serves as a practical exercise to test students' understanding and implementation of various algorithms, including **Merge Sort**, **Binary Search**, **Quick Sort**, and others. It evaluates how effectively students can apply these algorithms in real-world scenarios, highlighting their problem-solving and programming skills.

B. Course Learning Outcomes (CLO) covered

At the end of this assessment, students are able to:

CLO 3 Demonstrate problem-solving skills via concepts, techniques, and proficiencies in algorithm analysis which leads to potential applications in the self-development stage e.g., further studies.

C. University Policy on Academic Misconduct

1. Academic misconduct is a serious offense in Xiamen University Malaysia. It can be defined as any of the following:
 - i. **Plagiarism** is submitting or presenting someone else's work, words, ideas, data or information as your own intentionally or unintentionally. This includes incorporating published and unpublished material, whether in manuscript, printed or electronic form into your work without acknowledging the source (the person and the work).

- ii. **Collusion** is two or more people collaborating on a piece of work (in part or whole) which is intended to be wholly individual and passed it off as own individual work.
 - iii. **Cheating** is an act of dishonesty or fraud in order to gain an unfair advantage in an assessment. This includes using or attempting to use, or assisting another to use materials that are prohibited or inappropriate, commissioning work from a third party, falsifying data, or breaching any examination rules.
2. All assessments submitted must be the student's own work, without any materials generated by AI tools, including direct copying and pasting of text or paraphrasing. Any form of academic misconduct, including using prohibited materials or inappropriate assistance, is a serious offense and will result in a zero mark for the entire assessment or part of it. If there is more than one guilty party, such as in case of collusion, all parties involved will receive the same penalty.

D. Instruction to Students

1. **Task Completion:**

Complete the tasks listed in **Section F** below.

2. **Group Assessment:**

This assessment is to be completed in groups. By **November 30, 2024**, complete the provided form through the below link to designate your group representative and list all group members. <https://forms.office.com/r/kf3XVBYNsx?origin=lprLink>

3. **Submission Guidelines:**

Only **the group representative** should submit the following four files to Moodle:

- **Source Code:** Submit the **.c** or **.cpp** file.
- **Project Report:** Submit both **.pdf** and **.docx** formats, including:
 - i. A cover page.
 - ii. A table outlining each group member's contributions.
 - iii. Well-commented source code.
 - v. The marking rubric.
- **Turnitin report:** Upload a Turnitin report to ensure the plagiarism rate is below 20%.

- **Oral Presentation:** Submit a video file in **.mp4** format, limited to **7** minutes.
4. **Submission Format:**

- Upload a soft copy of the files to the Moodle system.

5. **Deadline:**

- Submit all files by **Monday, December 23, 2024 (Week 15, 6:00 PM)**. Late submissions will incur a penalty.

6. **Additional Notes:**

- Each group should submit only once via the group representative's Moodle account. Name your files and folders using the format:

YourGroupName_CST207_GroupProject_202409.pdf.

7. **Turnitin Report**

- Submit a softcopy of your report content to Turnitin for similarity checking. The **Turnitin similarity** should be **below 20%**.

1. Turnitin Class ID: 46218849

2. Enrolment Key: CST202409

E. Evaluation Breakdown

No.	Component Title	Percentage (%)
1.	Project	60
2.	Report and Individual work	40
	TOTAL	100

F. Task(s)

You are a software developer working for a software development company. One of your assigned tasks is to create a library management system (LMS) for a university library. The library serves students, faculty, and staff, offering a collection of books, journals, and other academic resources.

Your goal is to develop software to automate the library's operations. The system will include three main control panels:

- **Main Control Panel:** Provides navigation and user authentication.

- **Administrator Control Panel:** Allows managing library resources, including adding, modifying, searching, and deleting books.
- **Library User Control Panel:** Enables users to search for books, borrow them, and view their transaction history.

Each book record will include a unique ID, title, author, category, and availability status. The system will generate unique transaction receipts for all borrow and return actions.

1. System Panel

- Implement a secure login screen to allow only authorized personnel to access the system.
- Develop a main screen that serves as a control panel. From here, authorized users can log in as an admin, library user, or exit the system. (Credentials: Username - admin/user, Password - 123456)
- Allow users to select one of three actions from the main screen: log in as admin, log in as library user, or exit.

2. Admin Control Panel

When logged in as an admin:

- **Authentication:** Validate credentials (Username - admin, Password - 123456).
- **Functionalities:**
 - Add new books with details (ID, title, author, category, availability).
 - Search for a book by ID using the Binary Search algorithm.
 - Edit book details (modify title, author, category, availability).
 - Delete books from the system.
 - View an unsorted list of all books.
 - Sort and display books by:
 - ID (using Merge Sort)
 - Title (using Bubble Sort)
 - View transaction receipts sorted by receipt number (using Selection Sort).
 - Search transaction receipts by user ID (using Binary Search).

- Return to the main control panel.

3. Library User Panel

When logged in as a user (student, faculty, or staff):

- **Functionalities:**

- Search for books by title or part of the title (using Binary Search).
- Borrow books by selecting their ID and specifying the quantity.
- View transaction history filtered by user ID (using Linear Search).
- Return to the main control panel.

- **Book Borrowing Process:**

- Allow users to borrow books if available.
- Generate a transaction receipt, including a unique random receipt number, the customer's name, and details of the borrowed books, sorted by book ID (using Quick Sort).
- Display a return due date and prompt users to return books on time.
- After each transaction, ask if the user wants to borrow more books or exit to the main panel.

4. Error Handling

To ensure the system runs smoothly and provides a good user experience, it is essential to incorporate proper error handling. You are required to handle different types of errors within the system, including:

- **Invalid Input Errors:**

- Ensure that the user enters valid data for all fields (e.g., Product ID, Book ID, Quantity). For example, if the user enters a non-numeric value for the quantity of books to borrow or a negative number for the product price, the system should prompt an error message and ask the user to re-enter the data.
- If a user enters an invalid login username or password, the system should display an appropriate error message and allow the user to attempt to log in again a

specified number of times before exiting or locking the system for security purposes.

- **Book/Inventory Errors:**

- When borrowing a book, check if the requested book is available in the library's inventory. If not, display an error message informing the user that the book is currently unavailable.
- Handle the case where the user tries to borrow more copies of a book than are available. The system should ask the user to enter a valid quantity or suggest the available number of copies.

- **File Handling Errors:**

- Implement error handling for file operations such as reading and writing the transaction history or saving changes to the book catalogue. If the system encounters an issue (e.g., file not found, unable to write to the file), it should display an error message and allow the user to retry or log the error for system admins.

- **Transaction Errors:**

- If there is an issue with generating receipts or updating records after a borrow or return action (e.g., a database writes failure), the system should provide a clear error message and roll back any changes made during the transaction to prevent corruption of the library's records.

- **Error Handling Requirements:**

- For each type of error mentioned, implement appropriate checks in the system to handle invalid inputs, file errors, transaction failures, and other edge cases.
- Provide meaningful error messages to the user, guiding them on how to fix the error.
- Ensure that the system does not crash or produce unexpected results due to user or system errors. It should fail gracefully and inform the user when necessary.

5. Report Requirements

Prepare a comprehensive report with the following components:

1. Video Presentation (max 7 minutes, 100 MB):
 - **System Demo:** Showcase key features, including Admin and User panels.
 - **Algorithm Explanation:** Briefly explain implemented algorithms and their purpose.
 - **Error Handling:** Describe techniques and show examples during the demo.
 - **Challenges:** Discuss any difficulties faced and how they were resolved.
2. Searching Algorithm Comparison: Compare the implemented search algorithms based on performance metrics, such as time complexity (e.g., $O(n)$, $O(\log n)$), and discuss challenges faced during implementation.
3. Your opinion on the easiest search algorithm to implement, with justification.
4. Sorting Algorithm Comparison: Compare the sorting algorithms used in your project, discussing their performance and any difficulties encountered during implementation.
5. Discussion on the easiest sorting algorithm to implement, with reasons.
6. Support your analysis with online resources, citing references.
7. Brief explanation of algorithms used, accompanied by system screenshots.
8. Include your algorithm code in the appendix (code screenshots are acceptable).
9. A section detailing your error handling approach and challenges faced. Your report should include a section explaining how you handled errors in your system and a discussion of the challenges you faced while implementing error handling.
10. Limit the report to 10 pages (excluding appendix and references).
11. Outline each group member's contribution to system development and report writing.
12. Submit one copy of your work through the group representative's Moodle account. Ensure that all required files, including the report, video, and code, are uploaded as per the submission guidelines provided on Moodle.

APPENDIX 1

MARKING RUBRICS

Component Title	Project					Percentage (%)	
Criteria	Score and Descriptors					Weight (%)	Marks
	Excellent (5)	Good (4)	Average (3)	Need Improvement (2)	Poor (1)		
Quality of implementation	Implementation is flawless and adheres to all project requirements.	Minor issues in code, but functionality is mostly intact.	Several issues, but the core functionality is present.	Multiple errors, incomplete functionality.	The project is incomplete or poorly implemented	20	
The Implementation of the Search Algorithm	Search algorithm is highly efficient, working as expected in all cases.	Search algorithm works with minor inefficiencies.	Search algorithm works, but with noticeable inefficiencies.	Search algorithm does not work as intended.	The search algorithm is not implemented or fails.	15	
The Implementation of the Sorting Algorithm	Sorting algorithm is efficient and works as intended.	Sorting algorithm works with some inefficiencies	Sorting algorithm works, but has significant inefficiencies.	Sorting algorithm fails or is poorly implemented.	Sorting algorithm is not implemented or is unusable.	15	
Video	Video is well-paced, clear, and demonstrates all functionalities effectively.	Video demonstrates the project with minor issues in clarity or pacing.	Video is adequate but has some unclear or poorly paced sections.	Video is unclear, poorly paced, or lacks important details.	No video or an incomplete video is submitted.	10	
TOTAL						60	

Note to students: Please include the marking rubric when submitting your coursework.

Component Title	Report and Individual work					Percentage (%)	
Criteria	Score and Descriptors					Weight (%)	Marks
	Excellent (5)	Good (4)	Average (3)	Need Improvement (2)	Poor (1)		
Quality and the Report Format	The report is well-organized, clearly written, and follows the required format with no errors.	The report is organized and follows the format with few minor issues.	The report is readable, but lacks clarity or has formatting issues.	The report lacks organization or has significant formatting issues.	The report is poorly structured and difficult to follow.	10	
The Justification for Choosing the Search Algorithm and the Comparison Between Them	Clear, well-reasoned justification with thorough comparison of multiple search algorithms.	Good justification with comparison of at least two algorithms.	Justification is provided but lacks depth or clarity in comparison	Justification is unclear or lacks a solid comparison of algorithms.	No clear justification or comparison of search algorithms.	10	
The Justification for Choosing the Sort Algorithms and the Comparison Between Them	Well-argued justification with detailed comparison of sorting algorithms	Clear justification and comparison, but lacks some depth	Justification is present but does not fully explain the choice or comparison.	Justification is weak, lacks clarity, or does not compare sorting algorithms effectively.	No justification or comparison of sorting algorithms.	10	
Each Student's Contribution to the Project	Clear and detailed description of each student's contribution with well-documented roles.	Descriptions of contributions are clear but could provide more detail.	Contributions are mentioned, but lack detail or clarity.	Contributions are vague or incomplete.	No clear description of contributions or lacks details.	10	
						40	

Note to students: Please include the marking rubric when submitting your coursework.