Library Management System

Jayden Robertson

1958082@dundeeandangus.ac.uk

1. Overview

This documentation covers the implementation phase of a Library Management System, developed to manage library stock. The system allows library staff to manage various types of items, including books, journals and audio/video materials.

2. System Requirements

This section covers the Functional requirements and non functional requirements in order for the Library Management System to function correctly.

2.1 Functional Requirements

- Display all items and their types in the console
- Add, delete, edit and search for items
- Calculate and display total cost of all items
- Calculate and display insurance cost (50% of total cost, capped at £400)
- Input validation and error messages

2.2 Non-Functional Requirements

- Object oriented design
- Data validation
- Proper documentation

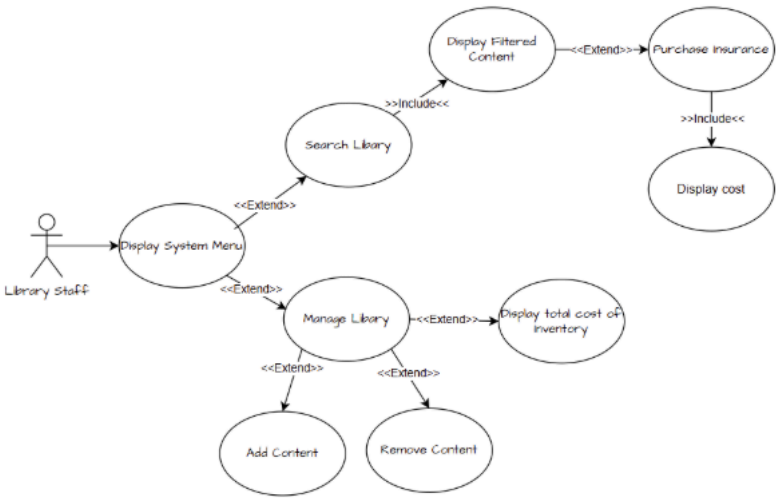
---

3. Diagrams

Before development, diagrams were created to provide a blueprint for the systems architecture and functionality. These diagrams were visual guides throughout the implementation phase.

3.1 Use Case Diagram

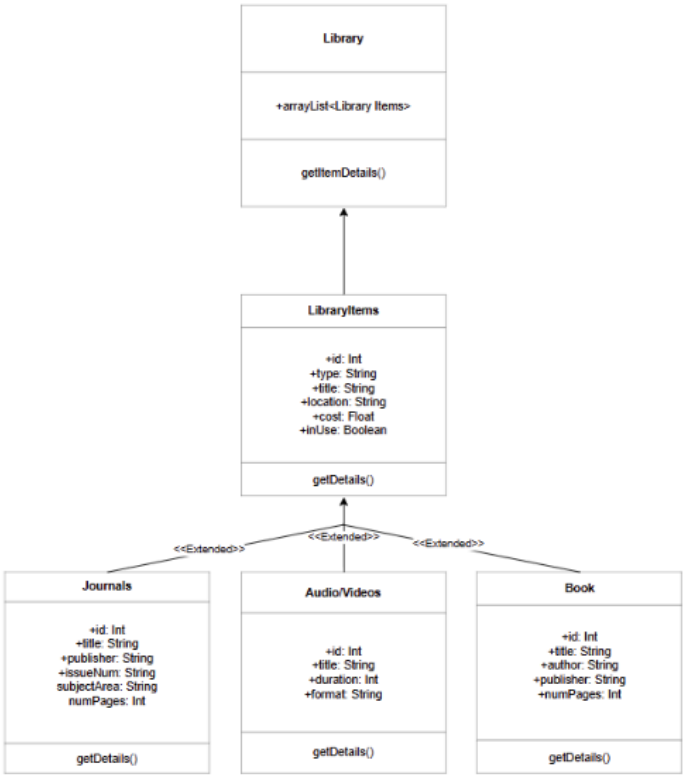
Use Case Diagrams represent the interactions between users as actors and the system. It showcases the functionalities available to the library staff who are the main users of the system.



Use case diagram

3.2 Static Model

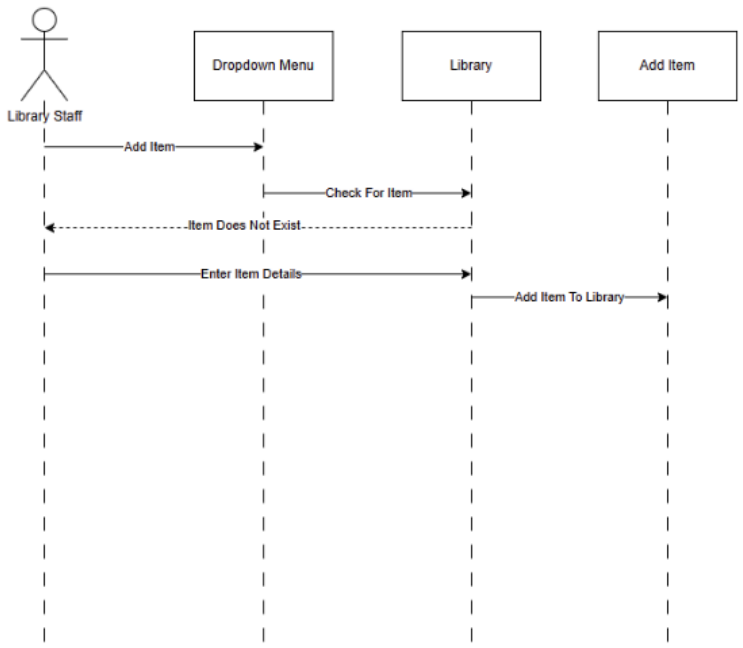
The structure of the Library Management System is represented through a Class Diagram. Class Diagrams show the systems architecture and relationships between entities.



Static model diagram

3.3 Sequence Diagrams

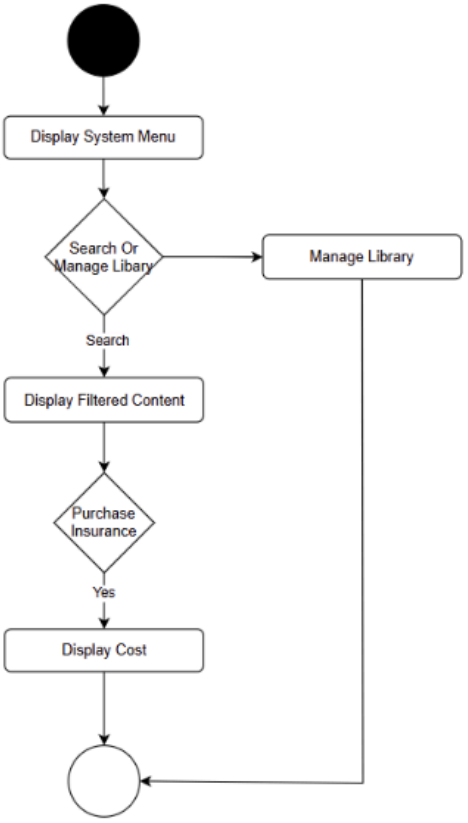
This Sequence Diagrams illustrates the interaction between the systems components and library staff when they add a new book to the library.



Sequence diagram for adding item

3.4 Activity Diagram

Activity Diagrams show the workflow for displaying the cost of insurance for an item.



Activity diagram

4. Test Case

The following test cases were carried out to validate the functionality of the Library Management System. Each tests includes the user input, expected system behaviour, actual results and pass/fail status. Failed tests include remediation and verification of fixes.

User Input	Expected Result	Actual Result	Pass/Fail	Comments
Enters "hello" in main menu	Displays “Invalid input. Please enter a number.”	Correctly displayed error message	Pass	Input validation working as intended
Enters “1” in main menu to add item	Prompts: “What type of item do you want to add? 1.Book 2.Journal 3. Audio/Video:”	Correct prompt appeared	Pass	Menu navigation functional
Enters "Book" for item type	Displays: "Invalid choice. Please try again.”	Proper error message shown	Pass	Numeric input enforcement working
Enters "4" for item type	Displays: "Invalid choice. Please try again.”	Proper error message shown	Pass	Numeric input enforcement working
Enters “7” in main menu to calculate the total cost	Displays: “Total cost of all items in the library: £10.00”	Correct prompt appeared	Pass	Menu navigation functional
Enters “9” in the main menu to exit the program	Displays: “Exiting the program.”	Proper message show	Pass	Menu navigation functional
Enters “2” in the main menu to delete an item	Displays: “Enter the ID of the item to delete:”	Correct prompt appeared	Pass	Menu navigation functional
Enters “Harry Potter” in the delete menu	Displays: “Invalid ID. Must be a number.” asks the user to enter again	Proper error message show. Prompts the user to enter again	Pass	Numeric input enforcement working
Enters “2” in the delete menu	Displays: “Item deleted successfully”	Proper message show	Pass	Deleting item working
Presses Enter with blank input in main menu	Displays: “Invalid choice. Must be a number.”	Proper error message show	Pass	Numeric input enforcement working

User Input	Expected Result	Actual Result	Pass/Fail	Comments
Searches for non-existent title "XYZ123” in search for item by title	Displays: “No items found with that title”.	Proper error message show	Pass	Search working as expected
Enters invalid type "Magazine” in search for item by type	Displays: “No items found with that type”.	Proper error message show	Pass	Type validation working
Calculate insurance cost for a book that costs “£50.00”	Insurance shows "£25.00”	Correct 50% calculation	Pass	Book insurance math correct
Adds book with cost "1000.00”	Insurance capped at "£400.00”	Proper capped applied	Pass	Insurance cap enforced
Enters "true" for availability in add a journal	Correctly stores Boolean value	Proper data type handling	Pass	Boolean conversion working
Enters "123" for page count in add a book	Accepts valid integer	Positive number handling	Pass	Numeric validation
Enters "TRUE" for availability when adding a book	Converts to proper Boolean	Case sensitive handling	Pass	User-friendly input

5. Evaluation

The Library Management System meets all the functional requirements with input validation, clear menu navigation and correct calculations. However if I were to continue working on this project I would implement persistent storage such as saving items to an XML file or a local database. This would prevent data loss between sessions. Overall the project was successful.

Jayden Robertson 1958082@dundeeandangus.ac.uk