Ammar Rafiqui (218931410)

MD Ahnaf Hyder (218207159)

Rigzin Shawootsang (217242124)

Kevin Dao (218118190)

Yash Dani

EECS 3311

Alvine Boaye Belle

2024-02-14

Task1: Use Case Diagram (25 points)

Your first task is to identify the main actors in the system. Please also indicate the sources of your use cases (i.e., from which Reqs). In your report, you are expected to clarify how each Reqs can be achieved with the use case you designed. Use a UML tool of your choice to draw a use case diagram based on the identified actors and use cases.

Actors: Client, User, Student, Faculty Members, Non-Faculty Staff, Visitors, Library Management System Team, Publishers, Library Management System, and Library Database.

Req1: Any client should be able to register as a user of the system with a unique/valid email and strong password (i.e., a combination of uppercase letters, lowercase letters, numbers, and symbols). The system currently allows four types of clients to be registered, i.e., students, faculty members, non-faculty staff, and visitors, while it's open for new types in the future. If a client registers as a student, a faculty member or a non faculty staff, her/his registration requires further validation from the management teams.

This request is achieved through the client first going through the register as user use cases in which it checks the email and password either denying or giving them a successful registration. If they are not a visitor, they go through the use cases to become one of the four types of clients, which further requires them to go through the validation process of the use case connected back to the library management team actor.

Req2: Using the system, any registered client can rent a physical item (i.e., books, magazines, CDs), open an online book, or subscribe to an online university-provided newsletter (e.g. NY Times), etc. Each physical item has 20 copies in the library. Penalty will be applied if a book is overdue (i.e., 0.5$ a day). A user can borrow up to 10 physical items and can keep an item for at most 1 month. All physical items borrowed from the library (books, magazines, CDs) count toward the total of 10 items. A user will lose his borrowing privileges if he has more than 3 items overdue. Books that are 15 days overdue will be considered lost.

This request is achieved through the client having become a user giving them the option to through the use case rent a physical item, open an online book, or subscribe to an online university-provided newsletter. It then leads to the use cases for the penalty for overdue books and loss of browsing privileges if three items are overdue, should books be overdue for more than 15 days, it will be labelled lost.

Req3: After login, the system should show a list of hardcover books that a user is currently renting and the due date for returning the books. It should also prompt warnings about any book that is not returned yet and it is approaching (less than 24 hours until the due date) or past the due date.

This request is achieved through the client first logging in as the registered user use case which then connects to the view current rented books use case and the use case in which they are past the due date or the use case for the due date is approaching within 24 hours.

Req4: The system should allow a user to subscribe and read a paid-for newsletter via its interface, such as the NY Times. This can be done by opening a frame within the system where the NY Times website can be loaded. A user can decide at any time to cancel a newsletter subscription.

This request is achieved through the user going through the subscribe to pay for a newsletter use case which is connected to the view payment options use case which manages payment details use case connects to allowing them to purchase the newspaper and is connected by the cancel subscription use case to cancel their subscription whenever they want.

Req5: A user can search for a book using the application. For a book a user is searching, the app should also show recommendations of similar other books (based on the text similarity of book titles).

This request is achieved when the user goes through the search for a book use case which includes the show for similar recommendations of other book use cases.

Req6: If a user is a faculty, the app can keep track of the courses the user is teaching and the textbooks the user has previously used. The app then offers notifications to the user when a new edition of the textbook is available. If a textbook is not available, the app should notify the library management team of this, so that they could consult with the user to procure the book.

This request is achieved if the user is a faculty member who goes through the track of current teaching courses and previously used book use cases which is connected back to the library management system. This use case extends from the notification for the new textbook edition use case which extends from notifying the library management team if there is a new edition use case that includes consulting with the user to procure a new book use case.

Req7: Each item has a unique identification number and other details including its location in the library and whether the item can be purchased, which will help with the navigation for clients. Managers of the system can add, enable (can be rented), or disable (cannot be rented) an item.

This request is achieved by the library management system as it goes through the add item use case. It includes the manage item details use case which includes whether the item can be rented or not use case.

Req8: If a user is a student, the textbooks of a given course the student is taking, the app should make virtual copies of the textbooks available to the account of the user for the duration of the course. After that, the app should remove the virtual copies from the student account.

This request is checked when the library management system creates a virtual copy of a textbook for a student to use, it is then removed from the account once the course is over.

Req9: A user can request for a new book. A request can be of two types, i.e., textbooks for course teaching and self-improvement, etc. Depending on the type, the request will need to be prioritized by the app and the user should be notified of the priority accordingly. Often, textbooks for course teaching will be given higher priority.

This request is met by users who can request books, and what type it is, depending on that type the library management system will inform the user that based on their type it may not be prioritized.

Req10: The system could also offer discounted purchases of items via its special agreements with publishers, whose books/DVD are not normally freely available via the usual library management system. For this, the system needs to provide payment options like debit, credit, mobile wallet, etc.

This request is achieved through the special agreements with publishers use case which extends to the buy discounted items use case that is connected to the library management team and the user also includes the view payment options use case which extends to the manage payment details use case.

Req11: System data are stored in the database, we will use Csv/Excel files to simulate this process.

This request is achieved through the library database which goes through the manage and store system data use case.