Name	**Mohsin Waheed**
Reg No.	**4335/BSSE/F-21/SEC-B**
Assignment	**introduction to Software Engineering**
Submitted to	**Sir Shakir Rasheed Khattak**

Q No. 01 Waterfall model:

The Waterfall Model was the first Process Model to be introduced. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap. This model is divided into different phases and the output of one phase is used as the input of the next phase. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.

This type of software development model is basically used for the project which is small and there are no uncertain requirements. It is termed as waterfall due to its downward linear fashion of execution.

Advantages:

1. Upfront documentation and planning stages allow for large or shifting teams to remain informed and move towards a common goal.
2. Reinforces good coding habits to define before design and then code.
3. Clearly defines milestones and deadlines.

Disadvantages:

1. Design is not adaptive; often when a flaw is found, the entire process needs to start Over.
2. Reduces efficiency by not allowing processes to overlap.
3. Not ideal for complex, high risk, ongoing or object-oriented projects.

Q No. 2: Stages of software development lifecycle:

Stage 1: Planning and Requirement analysis :

The most important phase in which senior members of team takes inputs from the user and understand the problem of user.

Stage 2: Defining requirements :

Once the requirement analysis is done the next step is to clearly define and
document the product requirements and get them approved from the customer.

Stage 3: System Design:

In this phase, the software model is designed on the basis of requirements on DDS(Design document specification). For Example, Screen layouts, Business rules, Process diagrams and other documentation.

Stage 4: Building or Developing the Product :

Here actual development starts in which the code is generated as per DDS.

Stage 5: Testing the Product :

Here the testing starts which searches for defects. In case of any fault software is sent back to previous development stages and then re-tested.

Stage 6: Deployment :
Once the product is tested and ready to be deployed it is released formally in
the appropriate market.

Maintenance:
What happens during the rest of software's life: changes corrections, additions
and more.

Q No.03 Write One user level and several system level requirements:

User Requirement:

I want a software that could manage our library system and could keep tracking all the activity and status of library.

System Requirements:

System requirements are divided into two parts i.e functional and non functional.

Functional Requirements:

- Allow the librarian to add and remove new members.
- Allow the user to search for books based on ISMN, title, author, etc., and find their location in the library.
- Users can request, reserve, or renew a book.
- Librarian can add and manage the books.
- The system should notify the user and librarian about the overdue books.
- The system calculates the fine for overdue books on their return.

User - The user can log in, view the catalog, search for books, checkout, reserve, renew and return a book.

Librarian - The librarian registers new users, adds and maintains the books, collects fines for overdue books, and issues books to users who need them.

System - The system is the library management system itself. It keeps track of the borrowed books and sends notifications to the user and librarian about the overdue books.

Non Functional Requirements:

Usability:
Software should be easy to use and should not be complicated for a user. User should be able to access all the permitted operations easily.

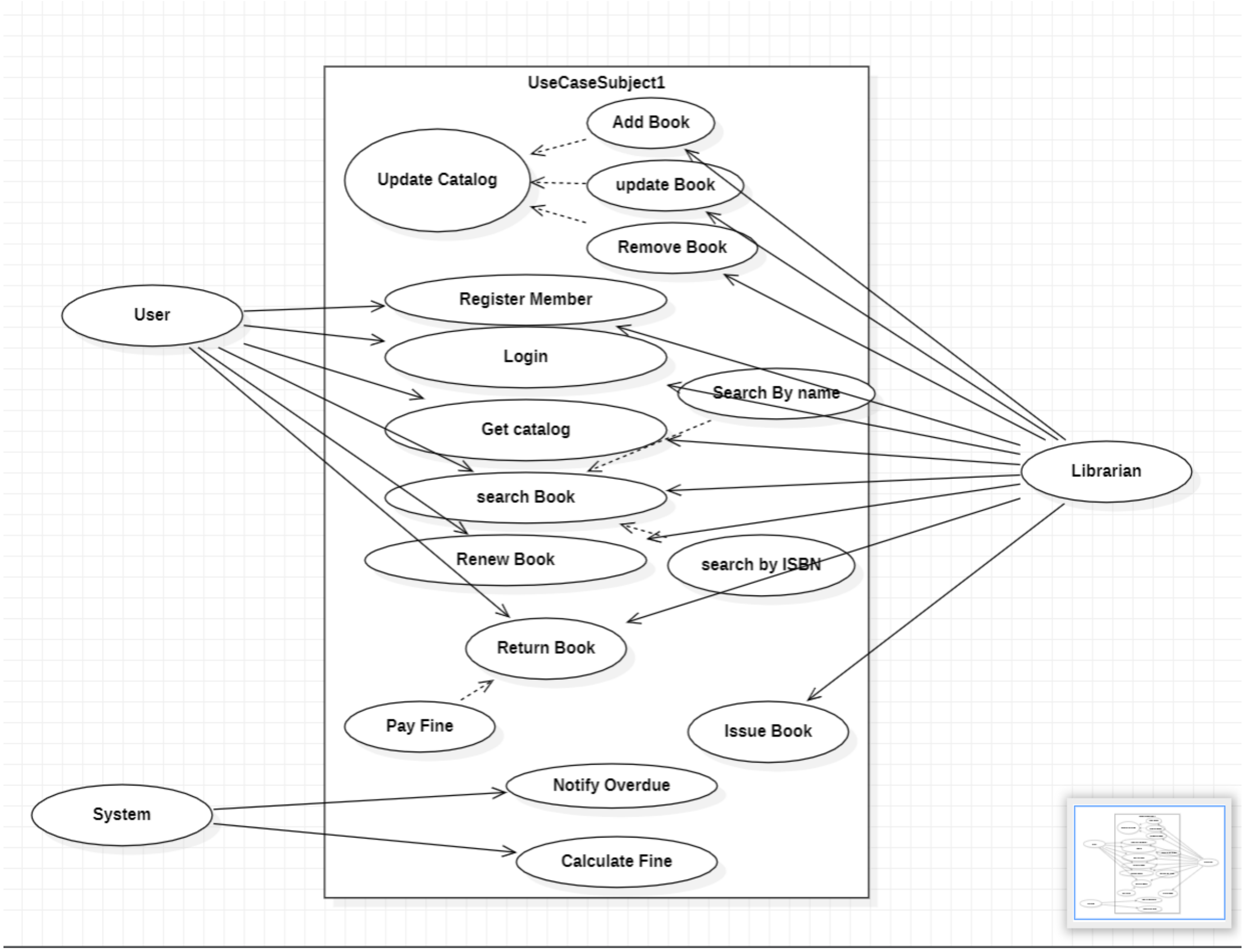
Accuracy:
Accuracy is very Important for the library management system. The data stored about the books and the fines calculated should be correct, consistent, and reliable.

Availability
The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

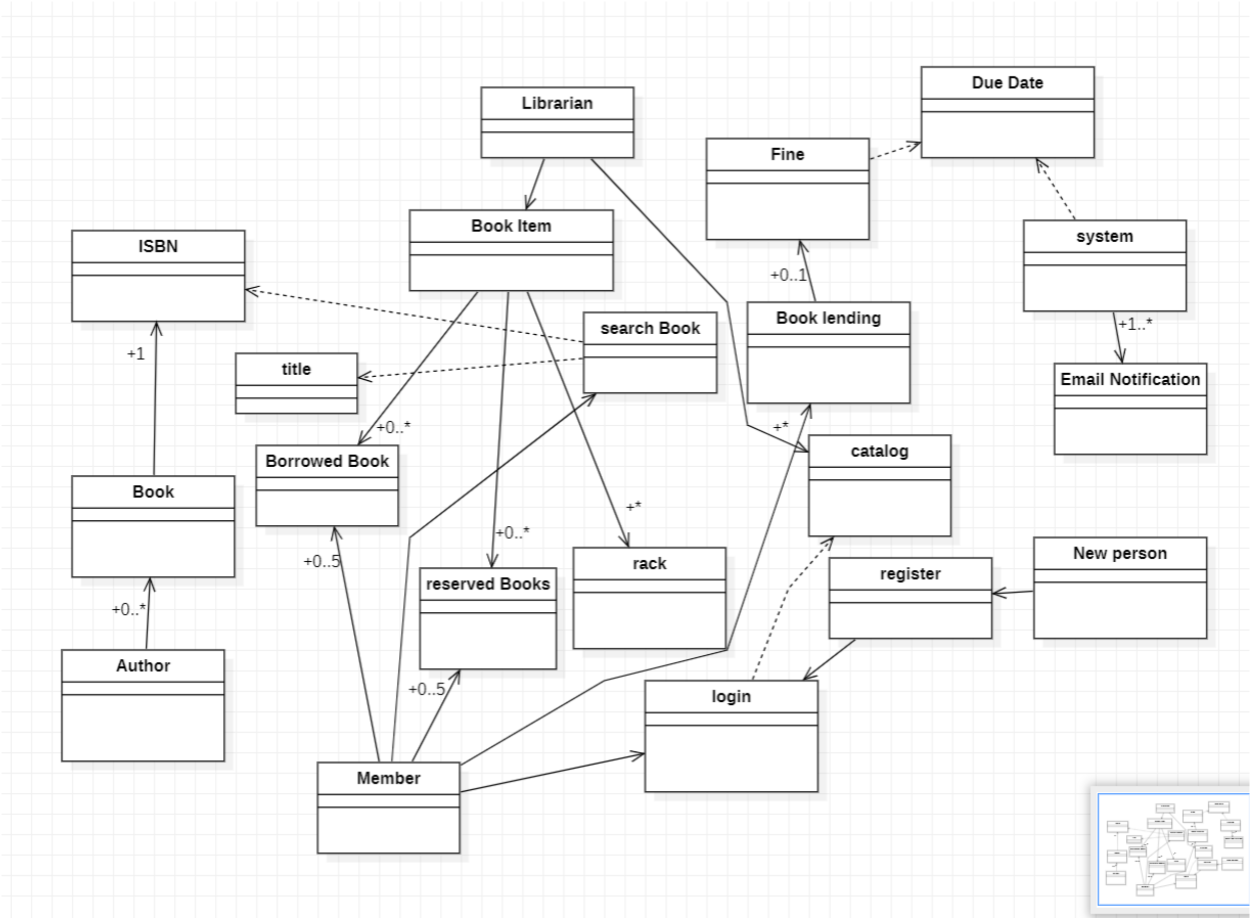
Maintainability
The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

Q No. 04 use case diagram:

Use Case of Library Management system:



(b) Domain Model



Q No. 05: Non functional requirements:

Bike Racing:

- The game must not crash.
- Minimum frame rate should be 30 frames per second.

Online banking system:

- Increase Customer Satisfaction: Internet banking system must allow customers to access banking services 24 hours a day, 365 days a year with minimum downtime period for backup and maintenance.
- Expand Product Offerings: The new services allow bank to capture a larger percentage of their customers' asset base. The internet banking system should provide facilities for bank to offer new services and products onto its homepage.