# *QUESTION NO:1*

**Describe Waterfall Model and list the stages of Waterfall Model for software development and list three of its advantages and disadvatages?**

# *ANSWER:*

The Waterfall Model is a classical model used in System Develop,emt Life Cycle to create a system with a linear and sequential approach. It is also referred to as a linear sequential life cycle model.

## STAGES:

It has following stages:

1.Communication

2.Planning

3.Modeling

4.Construction

5.Deployment

- **COMMUNICATION**:

Requirements Gathering

- **PLANNING**:

Estimating/Scheduling/Tracking

- **MODELING**:

Analysis & Design

- **CONSTRUCTION**:

Coding/Implementation/Test

.**DEPLOYMENT**:

Delivery/Support/Feedback

## *ADVANTAGES:*

1.Upfront documentation and planning stages allow for large or shifting teams to remain informed and move towards a common goal.

2.Forces structured,disciplined organization.

3.Is simple to understand,follow and arrange tasks.

## *DISADVANTAGES:*

1.Design is not adaptive,often when a flaw is found,the entire process needs to start over.

2.Delays testing until the end of the development life cycle.

3.Does not consider error correction.

# *QUESTION NO 2:*

**List the stages of the software development lifecycle(SDLC).Describe each stage in one phrase each?**

# *ANSWER:*

- Planning & Requirement Analysis phase

. Requirement Definition Phase

. Designing Phase

. Development Phase

. Test Phase

. Deployment & Maintenance Phase

## ***Stage 1*:Planning and Requirement:**

.Requirement analysis is the most important and fundamental stade SDLC.

## ***Stage 2*:Defining Requirement:**

.Once the requirement analysis is done the next step is to clear define and

document the product requirements to get them approved from the customer

## *Stage 3*:System Design:

**.**Based on the requirements in SRS desired features and operation in detail are specified and documentated in a DDS.

## *Stage 4*:Building or Developing Product:

**.**In the stage of SDLC the actual development starts and the prduct is built.

## *Stage 5*: Testing the product:

**.** The stage refers to the testing of the product where products defects are reported,tracked,fixed and retested,until

the product reaches the quality standards defined inn the SRS.

## *Stage 6:* **Deployment**:

**.** Once the product is tested and ready to be deployed it is released formally in the appropriate market(i.e.where the software is put into the production and runs actual business).

**Maintenance:**

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

# *QUESTION NO 3:*

# *ANSWER :*

## *User level requirements:*

A Libray Management System is a software built to handle the primary housekeeping functions of a library. Libraries rely on library management system to manage asset collections as well as relationshipps with their members. Library management systems help libraries keep track of the books and their checkouts,as well as member' subscriptions and profiles.

Library management system also involve maintaining the database for entering new books and recording books that have

been borrowed with their repective due dates.

## *System level requirements:*

We will focus on the following set of requirements while designing the Library Management System:

**1.**Any library member should be able to search books by their title, author, subject category as well by the publication date.

**2.**Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.

**3.**There could be more than one copy of a book, and library members should be able to check-out and reserve any copy. We will call each copy of a book, a book item.

**4.**The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.

**5.**There should be a maximum limit (5) on how many books a member can check-out.

**6.**There should be a maximum limit (10) on how many days a member can keep a book.

**7.**The system should be able to collect fines for books returned after the due date.

**8.**Members should be able to reserve books that are not currently available.

**9.**The system should be able to send notifications whenever the reserved books become available, as well as when the book is not returned within the due date.

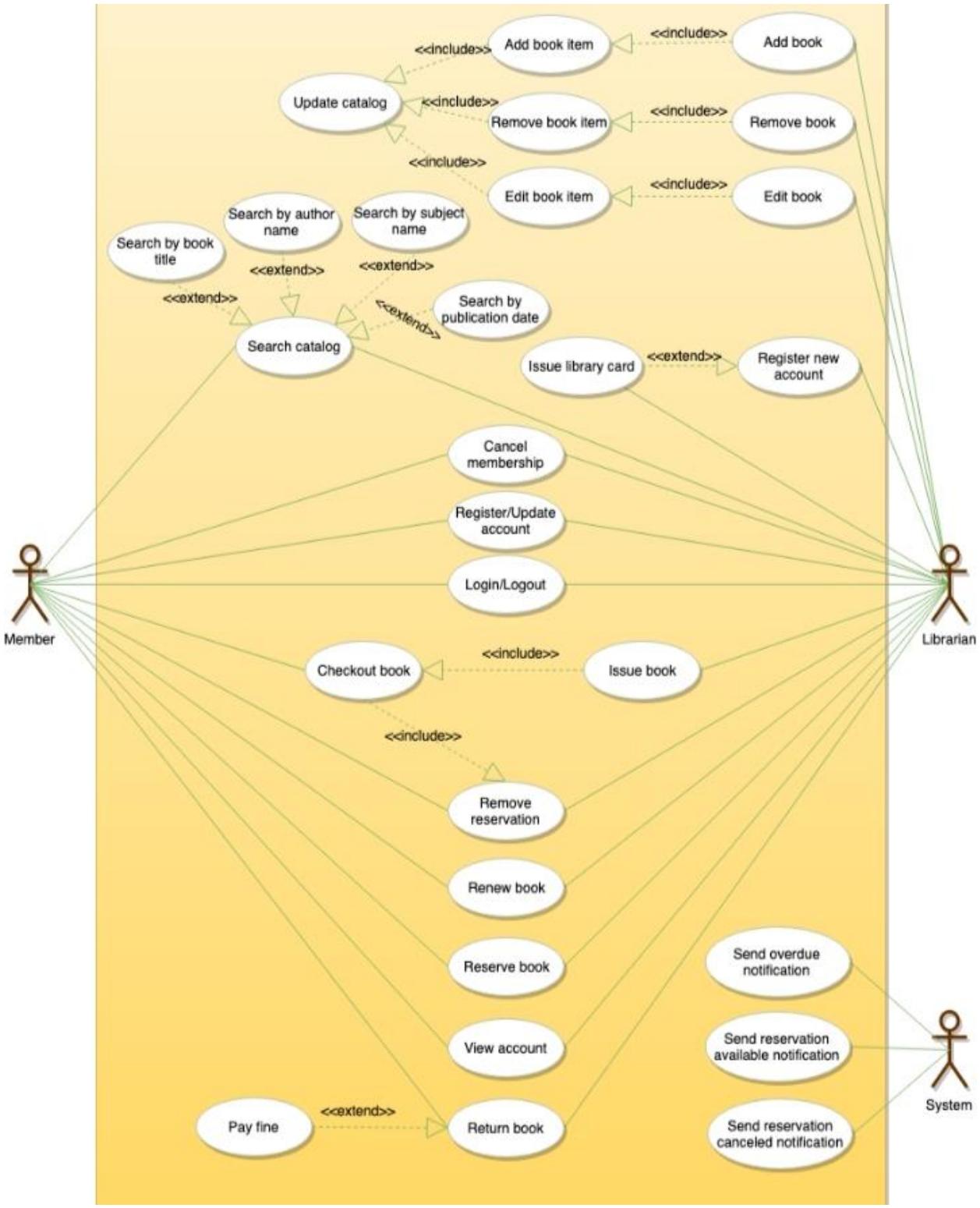
**10.**Each book and member card will have a unique barcode. The system will be able to read barcodes from books and members' library cards.

# *QUESTION NO 3:*

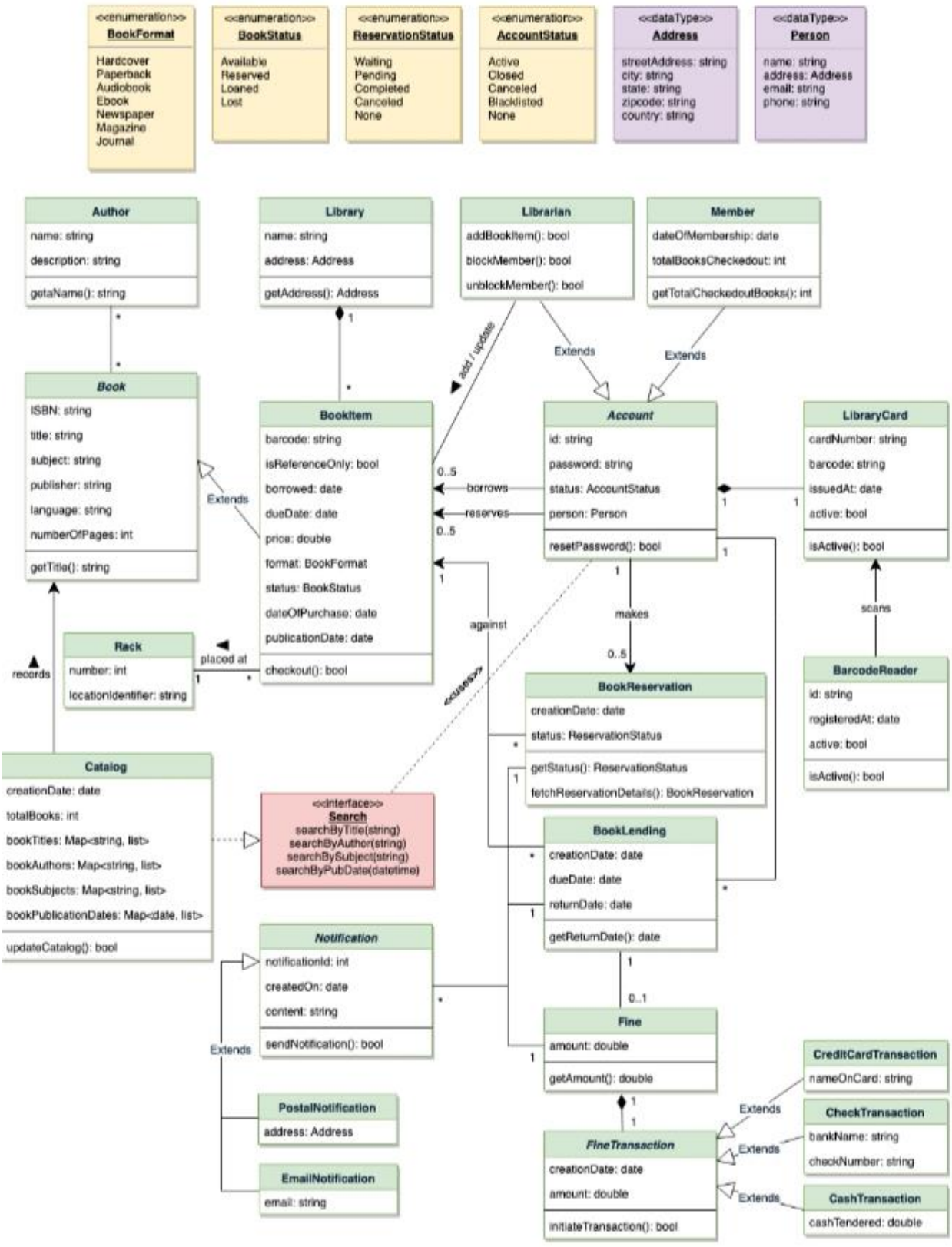
**Draw a Use Case Diagram and Domain Model for the problem mentioned in Q3?**

# *ANSWER:*

## ***USE CASE:***




# *DOMAIN MODEL:*


<<enumeration>>
BookFormat
Hardcover
Paperback
Audiobook
Ebook
Newspaper
Magazine
Journal
<<enumeration>>
BookStatus
Available
Reserved
Loaned
Lost
<<enumeration>>
ReservationStatus
Waiting
Pending
Completed
Canceled
None
<<enumeration>>
AccountStatus
Active
Closed
Canceled
Blacklisted
None
<<dataType>>
Address
streetAddress: string
city: string
state: string
zipcode: string
country: string
<<dataType>>
Person
name: string
address: Address
email: string
phone: string
Author
name: string
description: string
getaName(): string
Library
name: string
address: Address
getAddress(): Address
Librarian
addBookItem(): bool
blockMember(): bool
unblockMember(): bool
Member
dateOfMembership: date
totalBooksCheckedout: int
getTotalCheckedoutBooks(): int
add / update
Extends
Extends
Book
ISBN: string
title: string
subject: string
publisher: string
language: string
numberOfPages: int
getTitle(): string
BookItem
barcode: string
isReferenceOnly: bool
borrowed: date
dueDate: date
price: double
format: BookFormat
status: BookStatus
dateOfPurchase: date
publicationDate: date
checkout(): bool
Account
id: string
password: string
status: AccountStatus
person: Person
resetPassword(): bool
LibraryCard
cardNumber: string
barcode: string
issuedAt: date
active: bool
isActive(): bool
0..5
borrows
reserves
0..5
Extends
records
Rack
number: int
locationIdentifier: string
placed at
against
makes
0..5
scans
BarcodeReader
id: string
registeredAt: date
active: bool
isActive(): bool
<<uses>>
BookReservation
creationDate: date
status: ReservationStatus
getStatus(): ReservationStatus
fetchReservationDetails(): BookReservation
Catalog
creationDate: date
totalBooks: int
bookTitles: Map<string, list>
bookAuthors: Map<string, list>
bookSubjects: Map<string, list>
bookPublicationDates: Map<date, list>
updateCatalog(): bool
<<interface>>
Search
searchByTitle(string)
searchByAuthor(string)
searchBySubject(string)
searchByPubDate(datetime)
BookLending
creationDate: date
dueDate: date
returnDate: date
getReturnDate(): date
Notification
notificationId: int
createdOn: date
content: string
sendNotification(): bool
Extends
Fine
amount: double
getAmount(): double
0..1
PostalNotification
address: Address
EmailNotification
email: string
CreditCardTransaction
nameOnCard: string
CheckTransaction
bankName: string
checkNumber: string
CashTransaction
cashTendered: double
Extends
Extends
Extends
FineTransaction
creationDate: date
amount: double
initiateTransaction(): bool

# *QUESTION NO 5:*

**Write the Non-functional requirements for the following projects?**

**1.An online banking system**

**2.Bike racing system**

# *ANSWER:*

## *An online banking system:*

### *Availability:*

The system must be available during bank working hours. The mobile banking and ATM must be available round-the-clock with minimal maintenance times,

reaching 99.999% availability time per year.

***Performance:***

The bank management system is a multi-client system that must reach response time targets for each of the clients during simultaneous calls and must be able to run a target number of transactions per second without failure. The system must effectively utilize the hardware and energy resources to minimize operational costs.

***Usability:***

The system must provide different graphical interfaces for customers, tellers, and admins. All system interfaces

must be user-friendly and simple to learn, including helping hints and messages and intuitive workflow, especially in a client interface: the client must be able to fast learn and use the interface without prior knowledge of banking terminology or rules.

The interfaces must automatically adjust to devices with different screen sizes, and allow to change typeface size and color scheme to improve readability.

***Security:***

Bank management systems are notorious for being subject to malicious attacks, so security is the major requirement for the

system. Unauthorized access to the data is not permissible. The data must be backed up daily and stored in a secured location, at a distance from different facilities of the system.

Online transactions and stored digital files must be encrypted according to 128-bit or 256-bit AES encryption standards. The system also must employ firewall software as a defense against network attacks.

From the client-side, the system must provide an automatic log-out after an inactivity period, accept only secure

passwords that have sufficient length and non-alphabetic characters, and block login attempts after several unsuccessful trials.

## *BIKE RACING GAME:*

Randomly made up, high level, each functional requirement would need to be more logically categories and broken down much to be worked upon.

.It will be 3D open-world.

.You will be able to customise your character.

.You will be able to crawl, crouch walk, walk, run and jump.

.You will be able to drive vehicles.

## *Non-functional requirements:*

.The user must experience a strong story.

.The game must be fun.

.The game must be atmospheric.

.The game must not crash.

.The game must be accessible for all user segments.