

Introduction to Software Engineering

Assignment

Name: Raja Shahbaz Qaiser
Reg no: BSSE F21A 4262
Class: F21 A

Q1. What is Waterfall model and list the stages of waterfall model along with advantages and disadvantages?

Answer:

Waterfall model:

The Waterfall Model is a classical model used in System Development Life Cycle (SDLC) to create a system with a linear and sequential approach. It is also referred to as a linear-sequential life cycle model. In a waterfall model, each phase must be completed fully before the next phase can begin. 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.

Waterfall model Stages:

1. Requirements:

This stage handles the defining and planning of the project such as Potential requirements, deadlines and guidelines without mentioning specific processes.

2. Analysis:

The system specifications are analyzed to generate product models and business logic that will guide production.

3. Design:

A design specification document is created to outline technical design requirements such as programming language, hardware, data sources, architecture and services.

4. Coding/Implementation:

The source code is developed using the models, logic and requirements designated in the prior stages. Typically, the system is designed in smaller components, or units, before being implemented together.

5 Testing:

This is when quality assurance, unit, system and beta tests take place to report issues that may need to be resolved.

6 Deployment:

The product or application is deemed fully functional and is deployed to a live environment.

7 Maintenance:

Corrective, adaptive and perfective maintenance is carried out indefinitely to improve, update and enhance the final product.

Advantages of the waterfall model

1. Forces structured, disciplined organization.
2. Is simple to understand, follow and arrange tasks.
3. Clearly defines milestones and deadlines.

Disadvantages of the waterfall model

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.

Q2 List the stages of SDLC.

Answer:

Planning and Requirement Analysis:

Requirement analysis is performed by the senior members of the team with inputs from the customer, the market surveys and domain experts in the industry.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage.

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.

System Design:

Based on the requirements in SRS desired features and operation in detail are specified and documented in a DDS (Design Document Specification).

Building or Developing the Product:

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage.

Testing the Product:

This 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 in the SRS.

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 production and runs actual business)

Maintenance:

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

Q3.....?

User level requirements:

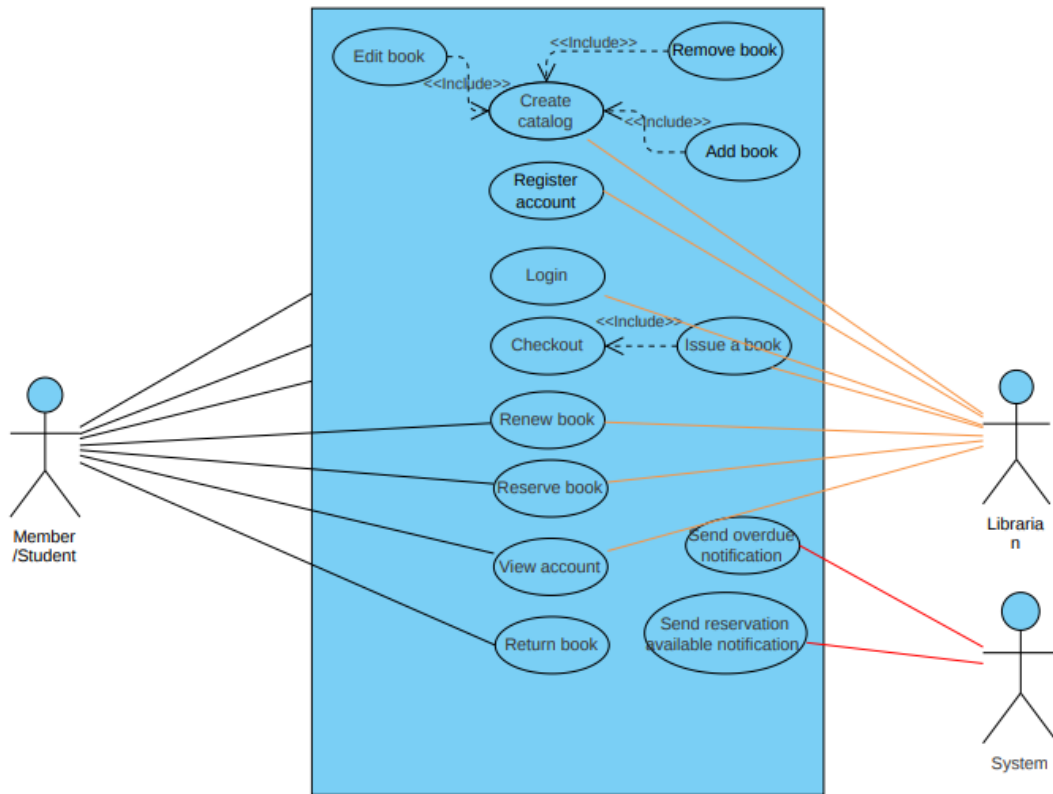
1. The system shall manage all inventory and assign every book a unique number

System level requirements:

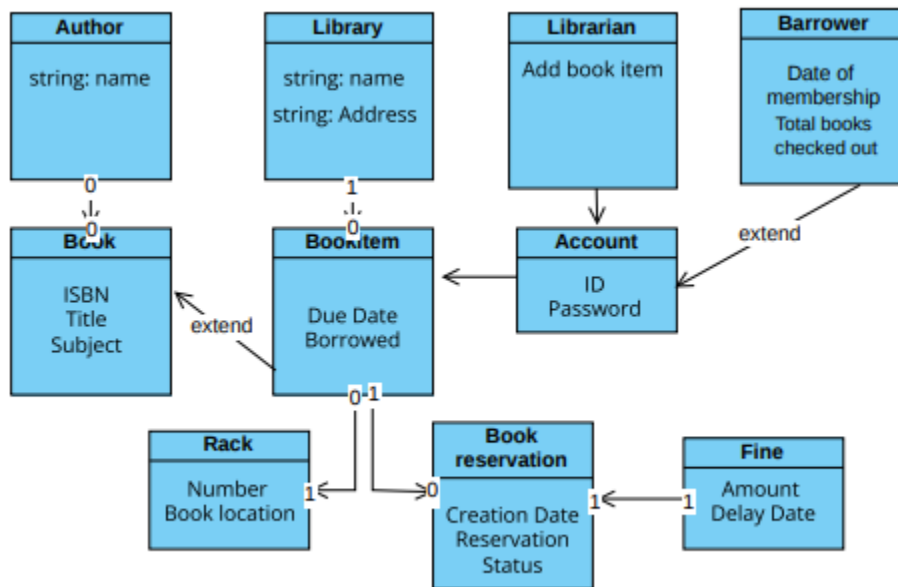
1. Each book will have a unique identification number (ISBN) and other details including a rack number which will help to physically locate the book.
2. There could be more than one copy of a book, and library members should be able to check-out and reserve any copy.
3. The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.
4. There should be a maximum limit on how many books a member can check-out.
5. There should be a maximum limit on how many days a member can keep a book.
6. The system should be able to collect fines for books returned after the due date.
7. Members should be able to reserve books that are not currently available.
8. 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.

Q4

Use case:



Domain model:



Q5 Write non-functional requirements for.

a) **Bike racing game:**

Platform:

The game must run in Windows XP and in Linux. Installing the game in a Windows XP and other environments and run simple tests to verify if the game properly works can test this requirement.

Required resources:

The game must be able to run with minimum of 1024 MB of RAM. The game must use less than one gig of hard disk space. Checking the total size of the folder in which the game was installed, for the hard disk space can test this requirement. For the RAM used, when playing the game, we can check the physical memory in the Windows Task Manager performance tab.

Response time

The average response time between click and reaction must be less than 0.5 seconds. The maximum response time between click and reaction must be two seconds.

b) Banking system:

For a bank management system, the most important non-functional requirements include security, performance, usability, and availability.

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.

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.

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.