**NAME : SHAYAN UMAR**

**REG NUMBER : 4316-FBAS-BSSE-F21-B**

**SUBJECT : INTRODUCTION TO SOFTWARE ENGINEERING**

**INSTRUCTOR : SIR SHAKIR RASHID KHAN KHATTAK**

**QUESTION NO:01**

**Describe waterfall model and list the stages of waterfall model for software development and list three of its advantages and disadvantages?**

**ANSWER**

**WATERFALL MODEL**

1. The term was first introduced in a paper published in 1970 by Dr. Winston W. Royce and continues to be used in applications of industrial design.
2. 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.
3. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion.
4. In a waterfall model, each phase must be completed fully before the next phase can begin
5. This model is divided into different phases and the output of one phase is used as the input of the next phase.
6. 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.
7. This type of software development model is basically used for the project which is small and there are no uncertain requirements.
8. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap.

**STAGES**

■ It has the following phases:

**1. Communication - Requirements Gathering**

**2. Planning – Estimating/Scheduling/Tracking**

**3. Modeling - Analysis & Design**

**4. Construction - Coding/Implementation/Test**

**5. Deployment – Delivery/Support/Feedback**

**ADVANTAGES:**

1.Clearly defines milestones and deadlines.

2. Forces structured , disciplined organization.

3. Easy to understand, follow and arrange tasks.

4. Facilitates departmentalization and managerial control based on schedule or deadlines.

**DISADVANTAGES:**

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

2. Does not handle request for changes,

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

4. Does not consider error correction.

**QUESTION NO: 02**

**LIST THE STAGES OF SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC) DESCRIBE EACH STAGE IN PHRASE**

**ANSWER**

**Stages of SDLC:**

1. **Planning and Requirement Analysis.**
2. **Defining Requirements.**
3. **System Design.**
4. **Building or Developing the Product.**
5. **Testing the Product.**
6. **Deployment and Maintenance.**

**Stage 1: Planning and Requirement Analysis : –**

**Requirement analysis** is the most important and fundamental stage in SDLC. – It 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.

**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 – This is done through ‘SRS’ – Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

**Stage 3: System Design: –**

Based on the requirements in SRS desired features and operation in detail are specified and documented in a DDS(Design Document Specification) – Including Screen layouts, Business rules, Process diagrams and other documentation

**Stage 4:** **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.

**Stage 5: 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.

**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 production and runs actual business)

**Maintenance: –** What happens during the rest of software’s life: changes corrections, additions and more

**Question No.03**

**USER LEVEL REQUIREMENT**

**i.**The end-user must be enrolled in university.

**ii.**The system should take the ISBN number as an input from the end-user.

**SYSTEM LEVEL REQUIREMENTS**

**i.**Each book must have a unique identification number.

**ii.**There should be more than one copy of a book.

**iii.**The system should be able to retrieve information like the availability of the book and due date.

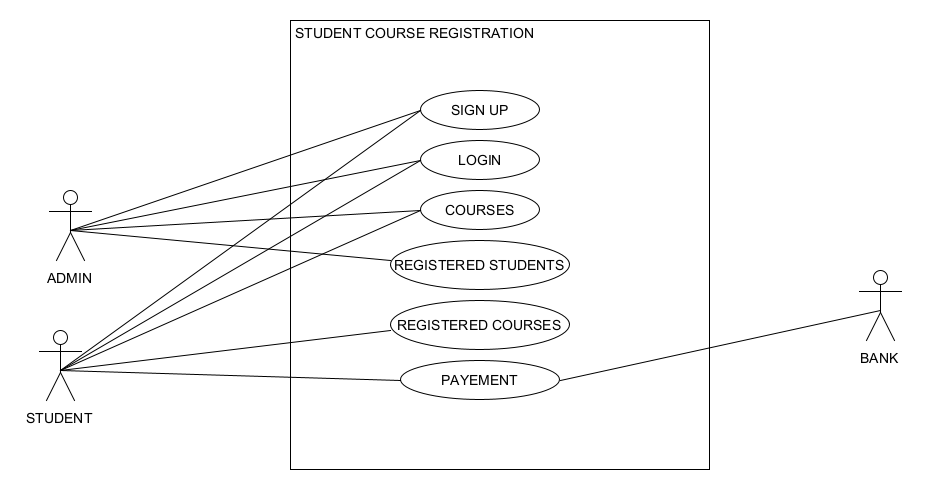
**iv.** The system should be able to keep record of the book borrower like his registration number, department name etc.

**v.**A book can be loaned only for two weeks at a time.

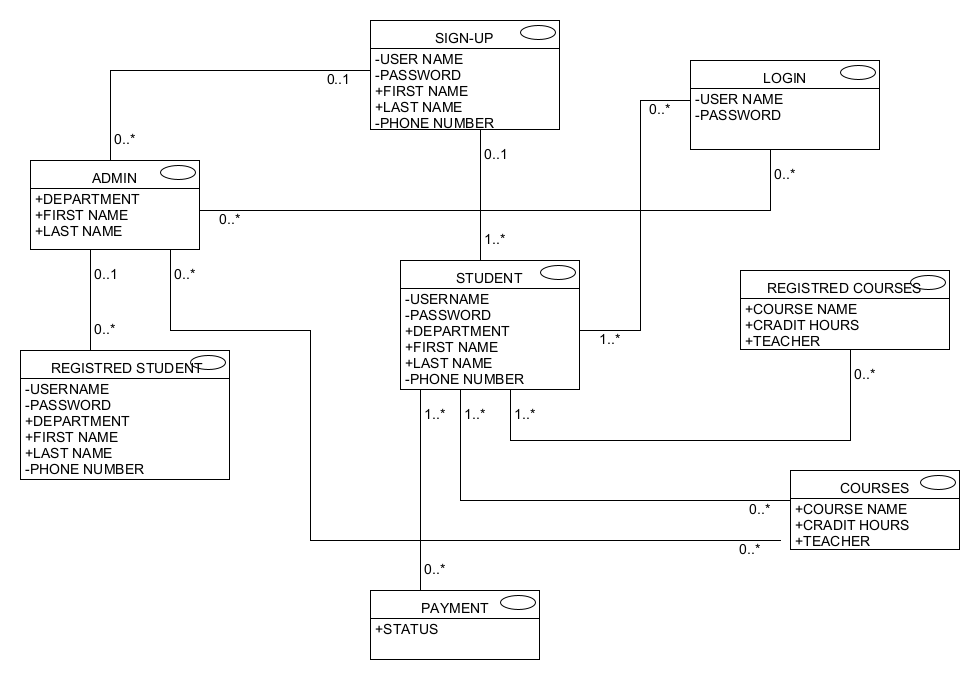
**vi.**The system should be able to search a book.

QUESTION NO:04

USE-CASES



DOMAIN MODEL

  
Q5: Write the non- functional requirements of the following projects?

1.Online Banking System:

* **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.
* **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.
* **Reliability** : Reliability reflects the capacity of the software to maintain its performance over the time. It implies how well the system performs in peak hours.
* **These are the non-functional requirements that the online banking should consist of.**

2. Bike-Racing Game:

* 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.
* The game must load within seconds.
* It should give multiple camera views to the user, i.e (first person, second person, top-down and isometric views) to make the game more realistic and interesting.