Software Engineering
Course Code: ICT-3209
Name : Afsana Aktermim
ID :- IT-21004
Topic: Assignment (CT Equivalent)- Final Preparation for the exam.

Question :1.

In a scrum-based software development project, the product owner has defined the following user stories for an e-commerce application:

- As a user, I want to log in securely so that I can access my account.
- As a user, I want to search for products by category to find items easily.

(1) Create a product backlog for these user stories by breaking them into tasks.

CamScanner

(II) Describe how the development team can prioritize these user stories during a sprint planning meeting, considering value to the customer and technical feasibility.

(III) Illustrate how these task will be tracked using a scrum board. Use proper terms like "To DO", "In Progress" and "Done". [6]

Answer (1)

1. Product Backlog Breakdown:

User Story 1: "As a user, I want to log in securely so that I can access my account."

- Task:-

1. Design the login page UI.
2. Implement username and password input field
3. Set up password encryption and authentication.
4. Integration login API with the backend

5. Test login functionality for security and usability.

User story 2:

"As a user, I want to search for products by category to find items easily."

Tasks:

1. Create UI for product categories and filters.

2. Develop backend logic for category-based filtering.

3. Implement search functionality in the frontend.

4. Test search results for accuracy and speed.

CamScanner

Answer - II

Prioritizing User stories in sprint Planning :-

(I) value to the customer

→ Secure login is critical as it enables users to access their, accounts, making it the highest priority.

→ Searching by category enhances user experience but is less critical than login.

(II) Technical Feasibility:-

→ login functionality involves complex encryption but is straightforward to implement.

→ Search functionality may require database optimization, which can take longer.

So, Prioritize the login user story first, then the search functionality in the subsequent sprint.

CS CamScanner

Answer (11)

3. Tracking task Using a Scrum Board :-

A scrum board will track the progress of task with columns. To Do, In progress and Done.

To Do	In Progress	Done
Design the login Page UI	Username and password input	Test login functionality.
Create UI for category filters.	Backend API integration	

- Task move 'To Do' to 'In progress' as work begin and to 'Done' when completed.
- These ensures visibility and helps the team stay organized and focused.

CamScanner

Question-2: development team

A software is about to start a project for a new innovative product. The project has some high-risk components due to it's novelty, and their uncertainty regarding the client future needs. The client is open to iterative changes, but the team must ensure that the software evolves in a manageable, cost-effective way.

- considering the high risks and the evolving nature of the clients needs, discuss how the spiral, Agile, and Extreme methodologies address risk management and adaptability. Which methodology would be the most suitable for a project with significant risk and evolving requirements, and why?

CamScanner

Answer

1. Spiral Model :-

(I) Risk Management :- The spiral model is explicitly designed for high-risk project. It emphasizes identifying and addressing risk early in phase of the spiral.

(II) Adaptability :- This model allows iterative development, enabling the client to refine requirements as the project progress.

(III) Suitability : Ideal for projects where risk is a major concern, but it can be resource intensive and may require highly skilled professionals for effective risk management

2. Agile methodology :-

(I) Risk management :- Agile minimizing risk by delivering small, functional increaments of the product at the end of each sprint. Continuous feedback and testing ensure issues are detected and resolved early.

(II) Adaptability : Agile excels in handling evolving requirements through iterative delivery and frequent collaboration with the client.

(III) Suitability :- Best suited for project where client needs may change rapidly, as it prioritizes flexibility and customer collaboration

CamScanner

3. Extreme Programming (XP):

(I) Risk Management:- Xp addresses risk through practices like Test-Driven Development (TDD), continuous integration, and pair programming. These practices help reduce errors and improve code quality.

(II) Adaptability:- Xp emphasize frequent releases and constant customer involvement, making it adaptable to changing requirements.

(III) Suitability:- Suitable for projects requiring frequent client interaction and where maintaining code quality is critical, though it might struggle with very complex, high-risk projects.

CamScanner

The spiral model is the most suitable methodology for this project. Cause-

The project involves high risks due to it's nature, and the Spiral Model is specially designed to manage such risk through iterative risk analysis. Additionally, it's iterative approach allows for flexibility in adapting to ~~iv~~ evolving client requirements, making it a balanced choice for both risk management and adaptability.

CS CamScanner

Question-3:- A company is working on two different Projects. Project A ~~will~~ has well-defined requirements and a strict deadline, Whik project B has evolving requirements with an uncertain timeline and continuous customer feedback. Both project involve high stakes, and the team must decide which development methodology to use.

- Compare and constrast the waterfall, Agile, Extreme and spiral development methods. Based on the characteristics of both projects. Which methodology would best suit each? Support your answer with a detailed analysis of how each methodology would address the specific needs of the projects, considering factors such as predictability, customer collaboration, and risk management.

CamScanner

Solution

Here's a simple comparisn and recommendation for project A and project B based on the given methodologies:-

1. Waterfall Development Model

→ Linear and sequential process, each phase (requirement, design, development, testing, deployment) is completed fully before moving to the next.

→ High predictibility, clear deadlines and well-suited for projects with well defined requirements.

→ Not flexible to changing requirements, feedback is limited to the end stages.

→ Project with clear requirements and a strict timeline.

CamScanner

For project A :- Waterfall works well because the requirements are fixed and the deadline is strict. The sequential approach ensures the team stays on track and delivers on time.

2. Agile Development Model :-

→ Iterative and flexible; focus on delivering small, functional parts (sprints) while continuously incorporating customer feedback.

→ High adaptability, encourage collaboration, and customer involvement throughout the process.

weakness

→ less suited for fixed deadlines or projects requiring detailed ~~out~~ upfront planning.

CamScanner

Best for
→ Project with evolving requirements and a need for frequent feedback

For project B:- Agile is ideal as the requirements are evolving, and customer feedback is continuous. It ~~adapt~~ allows the team to adapt quickly to changes and deliver value incrementally.

3. Extreme Programming (XP):-

→ A type of Agile with an emphasis on technical excellence, frequent realease, and close customer collaboration.

→ Great for rapidly changing requirements and delivering high quality code.

→ Requires a highly skilled team and close customer interaction, which can be resource-intensive.

CS CamScanner

→ Dynamic projects with a strong focus on technical quality.

For Project B:- Xp could work if the team is technically skilled and the project demand continuous delivery and customer feedback.

4. Spiral Development Model:-

→ combines elements of Waterfall and iterative models, focusing on risk assessment and iterative development with prototypes.

→ Excellent for high risk projects, allows for gradual refinement and continuous risk evaluation.

CamScanner

→ Can be complex and expensive, not ideal for simple or low-risk projects.

→ projects with high risk or uncertain requirements.

For project B:- Spiral could work if the stakes are exceptionally high, as it manages risks well.

Recommendation

Project A :- Use waterfall because of its structured approach, predictability, and alignment with fixed requirements and deadlines.

Project B :- Use agile for its flexibility, continuous customer collaboration, and adaptability to evolving requirements.

CamScanner

Q:-4 Software Engineering Ethics:-

Software Engineering ethics are rules or guidelines to help software developers make good decisions while creating technology. These principles focus on doing what is right, safe, and beneficial for society. Here are the key points:

1. Public Good:- Always prioritize the safety and well-being of people who will use or be affected by the software.

2. Honesty:- Be truthful about the capability, limitations, and risks of the software.

3. Quality Work:- Strive to create software that works well, is reliable, and secure.

4. Respect Privacy:- Protect user data and ensure privacy is not violated.

CamScanner

5. Fairness :- Treat everyone involved (clients, Colleagues and users) fairly and with respect.

Professional Responsibility Issues

Software Engineer have responsibility to :

1. Deliver work that is free of error or harmful effect.

2. Avoid shortcut which may lead to security risk.

3. Speak up when they see unethical practices, like biased algorithm or ~~un~~ misuse of data.

ACM/IEE Code of Ethics :-

The ACM /IEE code of provides clear guidelines for making ethical choise in software development. It emphasizes:

1. Continuous learning :- Keeping skills up-to-date to deliver quality work.

2. Accountability: Taking responsibility for mistaks and fixing them.

How it help:

This codes guide engineers to make fair and moral decesion, even in tough situation, by reminding them of their duties to society, clients, and colleagues.

Answer- 5

Functional Requirements:-

1. Flight Booking: User can search, select, and book flights. Contributes to usability by Providing core functionality for user.

2. Payment Processing: Securely process various payment methods. Ensures security and smoth transaction experience.

3. Flight Cancellation:- Allow users to cancel booking and receive refunds. Improves user experience by offering flexibility.

CamScanner

4. User Authentication:- Ensure users log in securely before accessing personal data. Enhances system security and data protection.

5. Notifications: Send email or SMS updates for flight schedules or delays. Improves performance by keeping users informed proactively.

Non-functional Requirements:-

1. Performance: The system must handle 1000 concurrent users without delays. Ensures the system performs well during peak usage.

2. Scalability: Support growth as the number of user increases. Facilitates system maintenance and long time usability.

3. Reliability:- Achieve 99.9% uptime to ensure availability. Increases trust and satisfaction among users.

CamScanner

4. Data security:- Encrypt sensitive user data, such as payment details. Protects user information, improving system security.

5. Usability:- Provide an intuitive interface for easy nevigation. Enhances easy experience, making the system accessible to all users.

Question:6

Illustrate and explain the V-model of testing phases in a plan-driven software process, detailing the relationship between development activities and corresponding testing activities.

→ The V model of testing in a plan-Driven Software Process:-

The V-model (varification and validation model) is an extension of waterfall Model, where testing activities are explicitly integrated into each stage of the software development

CamScanner

CamScanner

[II] Relationship between ~~Depto~~ Development and testing activities:

Each development activity corresponds to a specific testing activity:

1. Requirement analysis ↔ Acceptance testing

→ Ensure that final product meet user's perspective.

2. System Design ↔ System testing

→ Ensures the system architechture and interface work as intended.

3. Detailed Design ↔ Integration Testing

→ Ensures thate integration between modules is correct.

4. Implementation ↔ Unit testing.

→ Varifies that individual code modules perform as expected.

CamScanner

Key Stages in prototype Development:

1. Requirement gathering: Identify the basic requirements for the prototype, focusing on essential features.

2. Quick Design: Create a simple, initial design of the system interface and basic workflows.

3. Prototype Development: Build a working model based on the quick design, focusing on the usability and core functionality.

4. User evaluation: Present the prototype to users to gather feedback on its functionality design, and performance.

5. Refinement: Use user feedback to improve and update the prototype. This process may repeat multiple times.

CamScanner

6. Finalization: Once the prototype meets user needs, it helps define the final software requirements for full development.

Benifits of using the prototyping model:-

1. User feedback: Users can ~~st~~ interact with the prototype early, providing valuable insights to ensure the software meets their expectations.

2. Risk Reduction: Identifies potential problems and unclear requirements early, reducing the chances of errors in the final product.

3. Iterative Development: Allows continuous improvement through multiple iterations, ensuring the final system in user-friendly and functional.

CS CamScanner

Question-8:- Explain the process improvement cycle in software Engineering and describe its key stages. Name and explain some commonly used process metrics, highlighting how they help in monitoring and improving software processes.

→ The process improvement cycle in software engineering focuses on making software development processes better. It has three key stages:

1. Measure: collect data about the current process to see how it performs.

2. Analyze: Look for problems or areas that need improvement.

3. Improve: Make changes to fix issues and improve efficiency.

CS CamScanner

Commonly used Process Matrics:

1. Defect Density: Measures the number of defects per unit of code. Helps identify areas needing quality improvement.

2. Cycle time:- The time it takes to complete a process. Helps track speed and efficiency.

3. Customer Satisfaction: Feedback from users about the product. Ensures the process delivers what customers want.

These matrics help monitor performance and guide better decisions to improve processes.

CS CamScanner

Question 9: Explain the software Engineering Institute capability Maturity model (SEI CMM) and its five levels of capability and maturity. Analyze how each level contributes to improving the software development process and organizational process.

The SEI Capability Maturity model (CMM) is a framwork that helps organizations improve their software development process. It has five levels of maturity, each building on the previous one:

1. Initial: Process are unorganized and unpredictable. Success depends on individual effort.

2. Repeatable: Basic project management practices are in place. Processes can be repeated for similar projects.

3. Defined: Process are standardized and documented across the organization

4. Managed: Process are measured and controlled using data.

5 Optimizing: Continuous process improvement is achieved by analyzing and refining Processes.

Contribution to Improvement:

- Initial to Repeatable: Adds consistency to projects.
- Repeatable to Defined: Ensures better coordinatio and communication.
- Define Managed:- Introduces data-driven decision-making.
- Managed to Optimizing: Encourages innovatio and efficiency.

This progression leads to higher quality, reduced risks, and better performance.

CamScanner

Question 10:- Describe the core principle of Agile Software Development methods. Analyze how these principles are applied in different software development environments, and asses the benifit and challenges of using Agile methods in various project types and organizational setting.

Core principle of Agile Software Development:

1. Customer Collaboration: Agile emphasizes close Collaboration with customer to gather continuous feedback and ensure the product meets their needs.

2. Individuals and Interactions:- Agile prioritize effective teamwork and communication over rigid processes or ~~&~~ tools.

3. Working Software: Delivering functional software in small, increamental releas is more important than comprehensive documentation.

CamScanner

4. Responding to change: Agile embraces flexibility, allowing teams to adapt to evolving requirements and priorities.

Application in Different Environments:

1. Small team : Agile thrives in small teams where communication is straight forward, and decision can be make quickly.

2. Large organizations : In large organization scaling agile practice ensures that multiple team collaborate while maintaining flexibility.

3. Product Development: Agile supports iterative development in industries like Software as a service (SaaS), where updates and feedback loops are continuous.

CS CamScanner

Benifits:

- Faster delivery of functional product.
- Enhanced Adaptability to changing requirements
- Increased customer satisfaction
- Improve team collaboration.

Challenges:

- Difficulties scaling agile in large organizations or highly regulated industries.
- Resistance to change from traditional process.
- Dependency on team experience and communication skill.
- Potential misalignment between agile and long-term planning in some project type.

CamScanner

Question-11:-

Draw the release cycle of Extreme programming (XP) and explain the influential programming practices.

Answer:-

The extreme programming (XP) release cycle focuses on frequent, small release, ensuring that the software is always in a deployable state and customer feedback is continuously integrated.

Here's how the realease cycle typically works

Extreme programming (XP) Release cycle:

1. Planning: Initial and iteration planning define features and deliverables.

2. Development Iteration: Code is developed in short iterations (1-2 weeks).

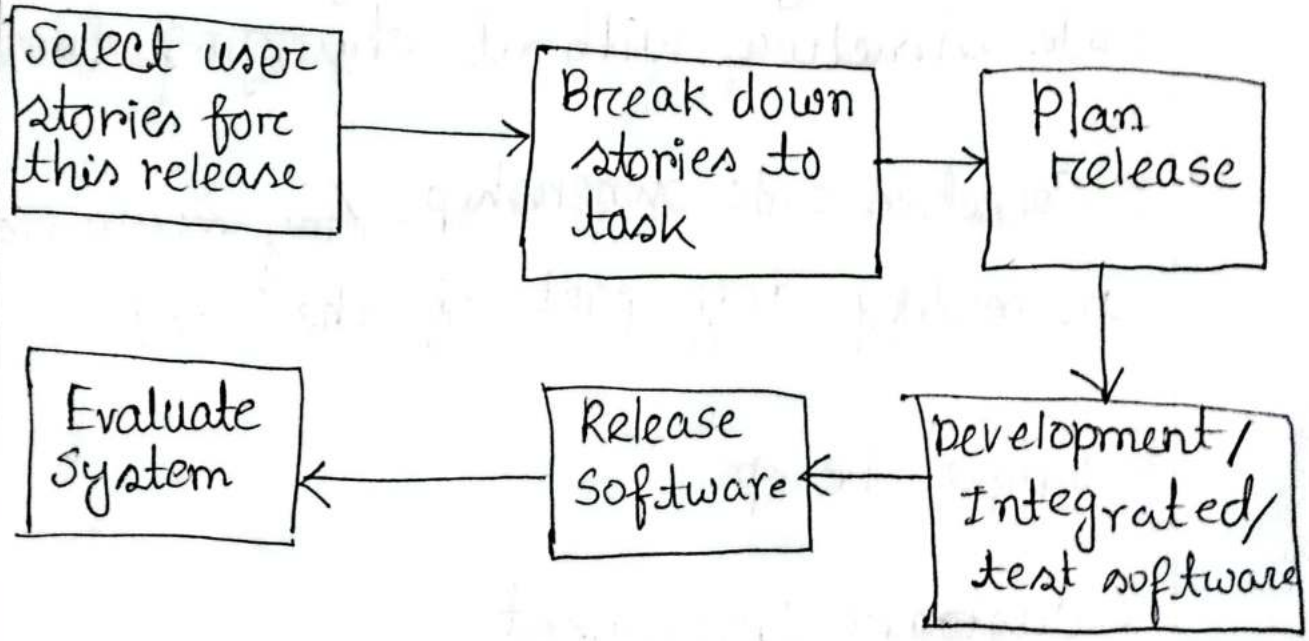
CamScanner

3. Continuous testing : Automated tests ensure functionality and quality.

4. customer feedback : Customer reviews each iteration and provides input.

5. Release : A small, working release is delivered after each iteration.

6. Repeat : The process repeats in cycles refining the software with each iteration.



CamScanner

Influential XP Programming Practices:

1. Pair Programming: Two developers collaborate on the same code to improve quality.

2. Test-Driven Development (TDD): Write tests before code to ensure correctness.

3. Continuous Integration (CI): Integrate code frequently with automated testing.

4. Refactoring: Continuously improve the code structure without changing functionality

5. Collective code ownership: Any team member can modify any part of the cod.

6. Simple Design

7. Customer Involvement.

CamScanner

Question-12

A local library wants to create a digital system to manage its operations. The system will teac books, Members and borrowing activities. Each book has attributes like title, author, IBBN and genre. Members has attributes such as name, membership ID and contact details. when a member borrows a book, the system records the borrowing date, return due date and return status. The library also want to maintain a catalog of overdue books and their ~~rep~~ respective fines.

Using the scenario of a digital library manageme system, design an Entity-Relationship Diagram (ERD) to represents the entities (e.g. books, membe borrowing activities) and their relationships. 'Ele explain the attributes of each entity and how they are interconnected.

CamScanner

Answer:

To design an entity relationship diagram (ERD) for the digital library management system, we will break down the entitities, their attributes and the relationship between them based on the requirements provided. Heres how it can be structured:

1. Entities and Attributes:

Book

Attributes:

Book (ID) Primary key: Unique identifier for each book

Title: Title of the book

Author: Author of the book.

ISBN: International Standard Book Number.

Genre: Genre of the book

Member

- Attributes
- Member ID (Primary key) : Unique Identifier for each member
- Name : Name of the member
- Contact details : Contact information (phone number, email, etc).
- Address : Physical address of the member.

Borrowing Activity

- Attributes:

BorrowingID (Primary key): Unique identifier for each borrowing activity.

- BookID (Foreign key): Refers to the borrowed book (links to the Book entity).
- MemberID (Foreign key): Refers to the member who
- borrowed the book (links to the member entity)
- Borrowing Date : The date the book was borrred.
- ReturnDue Date : The date the book is due for return.

CamScanner

• Return Status : Whether the book has been returned or not (could be "Returned" or Not Returned).

• Fine: The fine imposed if the book is returned late (can be NULL if returned on time).

Overdue Book

- Attributes:

• OverdueID (Primary key): Unique identified for each overdue book entry.

• BorrowingID (Foreign key): Refers to the borrowing activity of the overdue book.

• FineAmount : Fine imposed on the overdue book.

• OverdueDate : Date when the book become overdue.

CamScanner

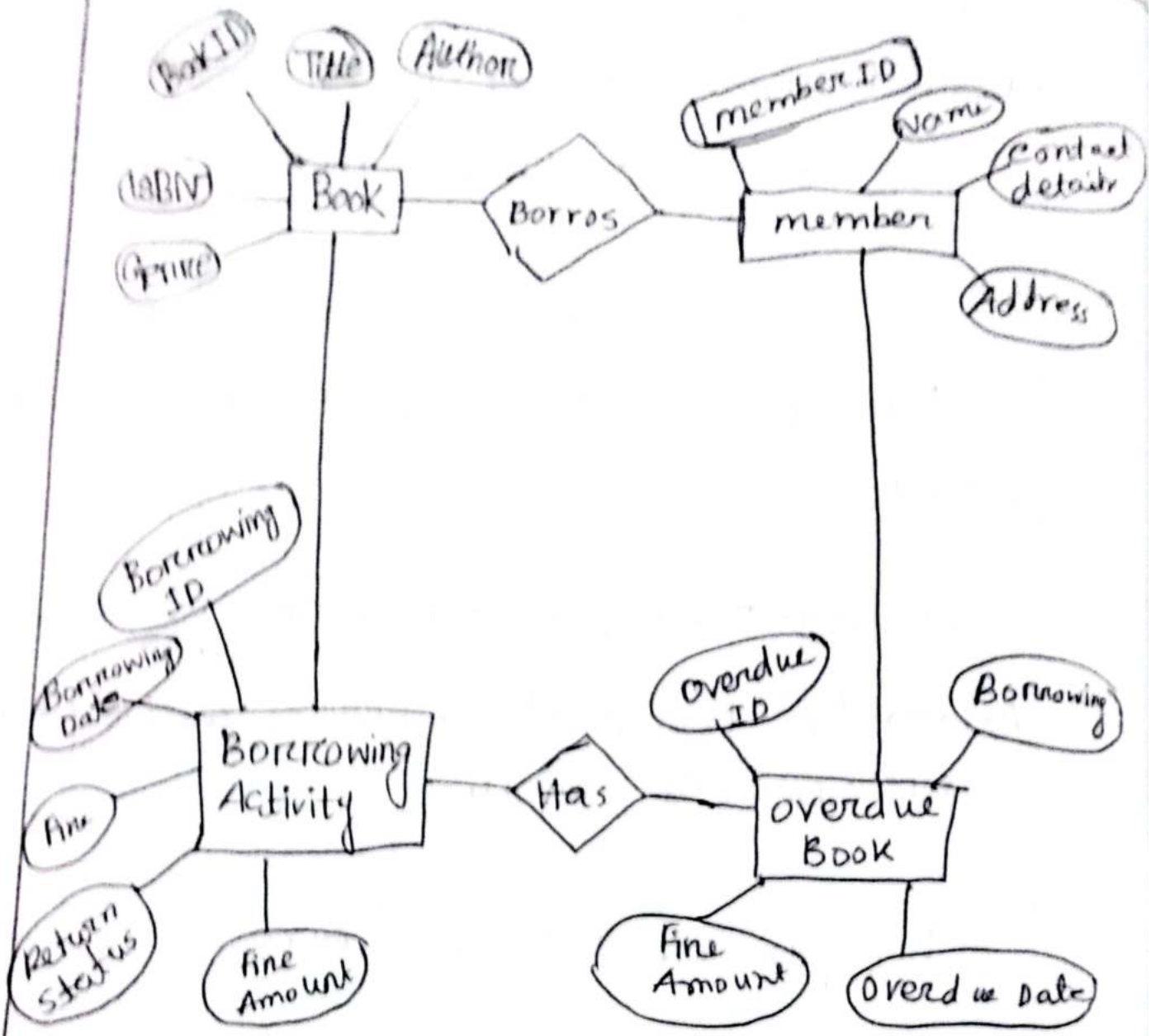
2. Relationships:

- Book Borrowing Activity (one-to-many)
- Relationship: A book can apear in many borrowing activities.

Member-Borrowing Activity (one-to-many).
Relationship: A member can have many borrowing records

Borrowing Activity: Overdue Book (one-to-one).
Relationship: Each overdue borrowing activity has a corresponding overdue.

4. Explanation of the relationship:


Book ID
Title
Author
ISBN
Genre
Book
Borros
member ID
Name
Contact details
member
Address
Borrowing ID
Borrowing Date
Borrowing Activity
Fine
Return status
Fine Amount
Has
overdue ID
Borrowing
overdue Book
Fine Amount
Overdue Date

Question 13:-

What is called Testing? Differentiate between validation and verification!

Answer:

Testing in software Engineering is the process of evaluating a software system or component to find defacts or errrors. It's a critical step in the software development lifecycle to ensure the quality, reliability, and functiality of the final product.

Difference between validation and verification:

Aspect	validation	varification
Definition	Ensures the product meets the user's needs and expectation	Ensures the product is built accounting to the specific requirments.
Focus	Focuses on the end product and its Functionality	Focuses on the processes and intermediate word products.

CamScanner

Type of Testing	varification is the static testing.	validation is dynamic testing.
Execution	It does not include the execution of the code.	It includes the execution of the code
Timing	It comes before validation.	It comes after validation.
Performance	verification finds about 50 to 60% of the defects.	validation finds about 20 to 30% of the defects.

Question-14

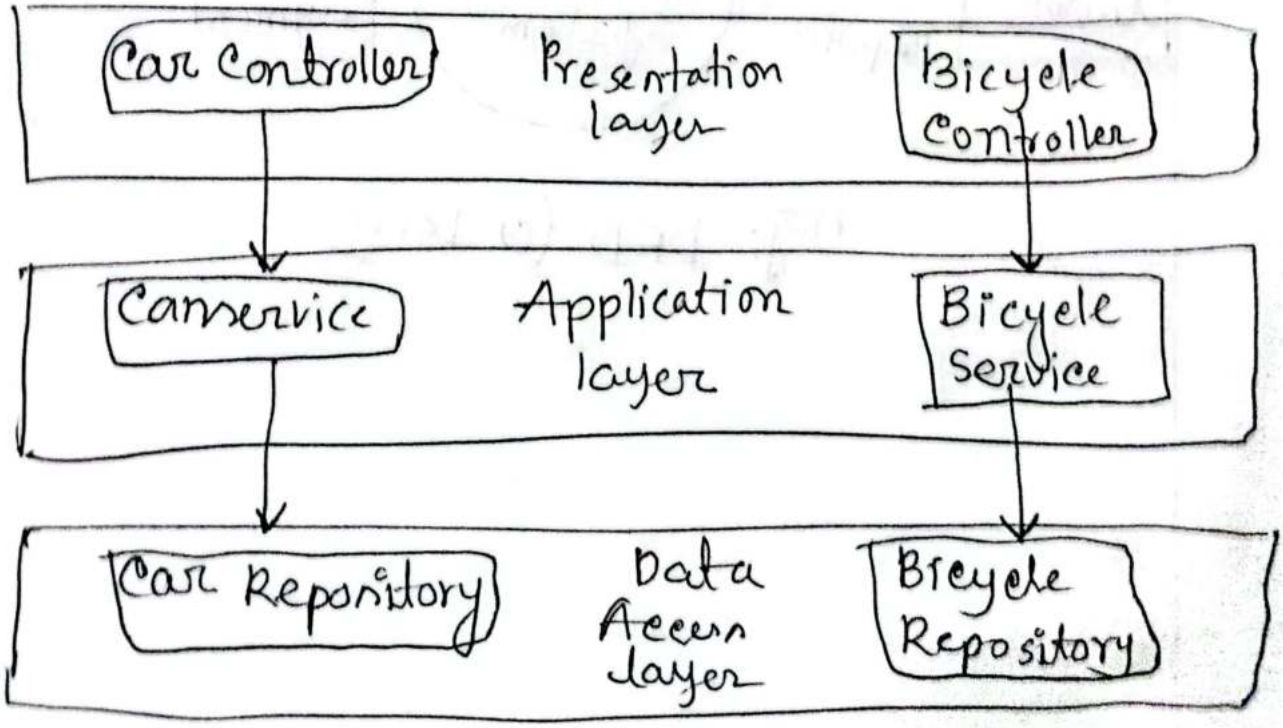
Design a layered architechture model for an online judge system, identifying the key layers (e.g. presentation, application, business logic, and data). Explain the responsibilities of each layer and analyze how this architechture ensures scalability, maintainability, and efficient performance.

CamScanner

Answer:

A Layered architechture for an online judge system would typically include three primary layers: presentation layer (UI), Application layer (Business logic), Data access logic layer (Database interaction) with each layer having well defined responsibilities and interaction only with layered directly above on below it, promoting modularity and maintainability.

Layered Architechture diagram:



Ans to the Q. N. 15

Data flow diagram hospital management System is used to create an overview of Hospital management without going in too much detail.

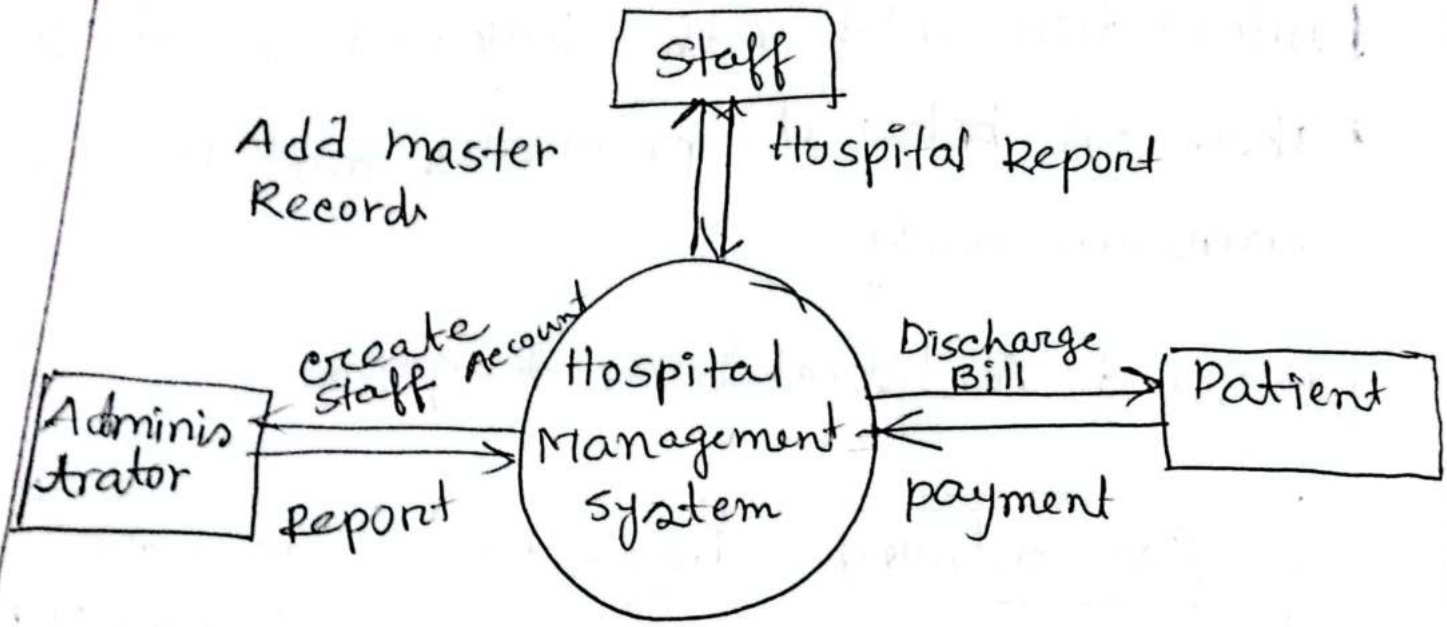
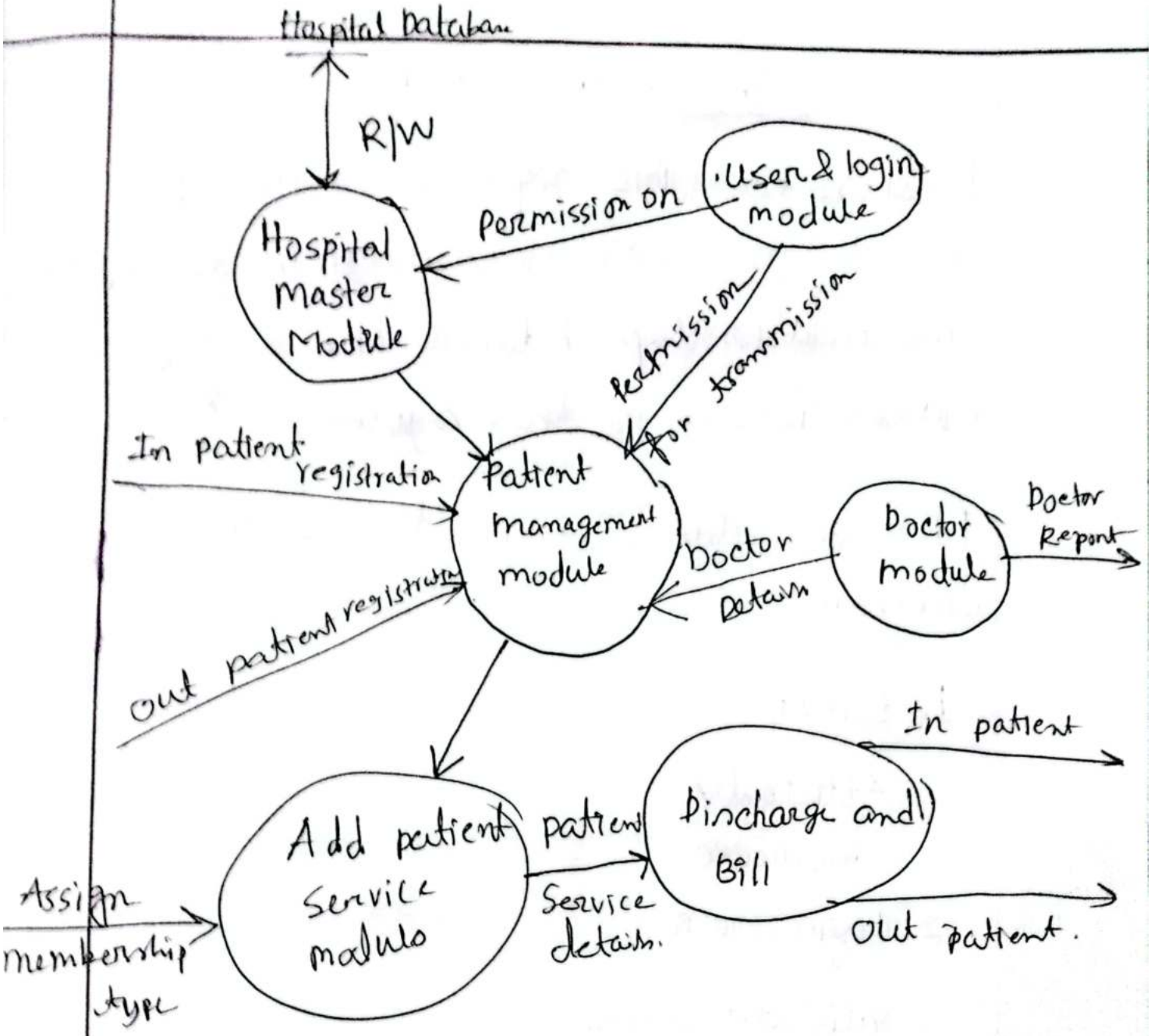


Fig: DFD (0 Level)


Hospital Database
R/W
Hospital Master Module
Permission on
User & login module
Permission for transmission
In patient registration
Patient management module
Doctor Details
Doctor module
Doctor Report
out patient registration
Add patient service module
patient
Service details.
Discharge and Bill
In patient
Out patient.
Assign membership type

Ans. to the Q. N. 1c

Based on the UML sequence diagram & we can design a UML class diagram to represent the relationship between the key Classes involved in this system:

Identified classes from the sequence diagram:

1. Student :

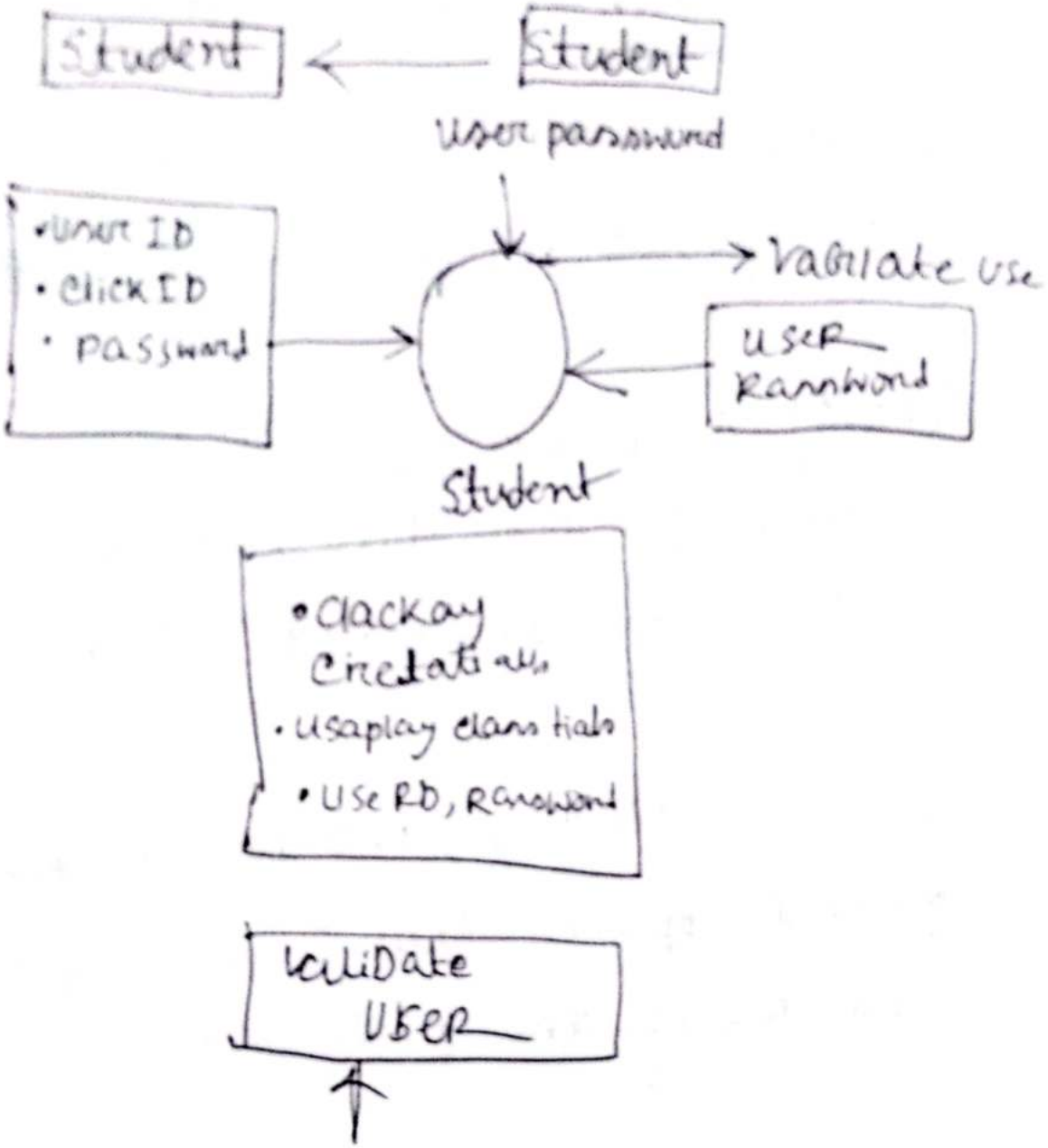
   Attributes
   Methodes

2. Login Screen

3. Validate Screen

4. Database

CamScanner


Student
Student
User password
• User ID
• Click ID
• Password
Validate use
User Password
Student
• Clackay Credentials
• Usaplay class tials
• User ID, Password
Validate User

Answer to the Q. N. 17

Quality Assurance (QA) vs Quality control (QC) Explanation, Differences, and Implements:

Quality Assurance (QA) is the proactive, proce focused approach that ensures quality is built into the development process.

Quality control (QC) is a reactive, product-focused approach that involves inspecting and testing the final product.

Difference between Quality Assurance and Quality control is given below:

CamScanner

Aspects	Quality Assurance	Quality control
Focus	Process oriented	Product-oriented
Objectives	Prevent defects	Defect and correct Defect.
Approach	Proactive	Reactive
Responsibility	Everyone in the process	Dedicated testing team

Impediments to QA and QC :

for QA :

(I) Lack of standardization

(II) Poor management support

(III) Resistance to chang

(IV) Limited Resources

(V) Time constaints.

CamScanner

For QC :

1. Late defect detection
2. Inadequate testing
3. Time pressure
4. Dependence on QA.
5. Limited automation.

Ans to the Q. N. 18

Role of quality assurance (QA) in software Development Life cycle (SDLC):-

The goal of quality Assurance (QA) is not just to finds bugs enty but to ensure that quality into built Into the software development process from the beggining.

QA Role at Each phase of SDLC:

SDLC phase	Role of Quality Assurance (QA)
1. Requirement Analysis	- Ensure clarity, Completeness, and Feasibility of requirements. - Conduct requirement reviews to prevent ambiguity. - Define acceptance criteria and quality standards early.
2. Planning	- Identify risks and mitigation strategies. - Define QA objective, scope and test strategy. - Plan resources, timelines, and test ev environments.
3. design	Review system architecture and design for quality aspects like security, scalability and Perfor - mance. - Identify potential failure points early.

Answer to the question No:- (19)

Rapid Application Development (RAD) Model in Software Engineering

Introduction :- The Rapid Application (RAD) model is an iterative and adaptive Software development methodology that focuses on quick prototyping and fast feedback loops instead of extensive planning and documentation. It was introduced in the 1990s by James Martin as an alternative to traditional models like the waterfall model.

Key Phases of the RAD Model :-

i. Business Modeling
ii. Data Modeling
iii. Process Modeling
iv. Application Generation & Testing

Principle of the RAD Model :-

i. Prototyping over Planning
ii. Active User Involvement
iii. Iterative and Incremental Development
iv. Component - Based Development

CamScanner

v. Quick Delivery

Advantages of the RAD Model :-

i. Faster Delivery
ii. Higher User Satisfaction
iii. Flexibility
iv. Reduced Risk
v. Reusuable components

How the RAD Model Ensures Faster Delivery while Maintaining Quality and User Satisfaction :-

i. Speed Through Prototyping
ii. Continuous User feedback
iii. Iterative Testing Ensures quality
iv. Focus on Reusable components

The RAD model accelerates software development without compromising quality by emphasizing prototyping, user feedback and ~~require fast it~~ reusability, it is ideal for projects that require fast iterations and evolving requirements ensuring that the final product is not only delivered quickly but also meets high-quality standards and user satisfaction

Answer to the question No: 20

White box testing ensures that all possible branches of a program - are tested by covering decision Points such as if, else, and loops, Below, we create a test cases table for the given Java code and than imple-ment a Junit test class in Java.

Test cases Table

Decision statement	x Input	Y Input	Expected output
If (y= 0)	5	0	"y is zero"
else if (X= =0)	0	3	"x is zero"
for loop does not run	0	2	"" (No output)
For loop Prints numbers divisible by y	4	2	"2", "4"
For loop prints numbers divisible by y	4	3	"3"
For loop with negative y	5	-2	"2" "4"
Negative x, loop does not execute	-3	2	"" (No output)

Junit Test class in Java

```
import static org.Junit; Junit.Jupiter.opi.Assertions
import org.Junit.Jupiter.opi.Test;
import Java.util.ArrayList;
import Java.util.List;

class DecisionTest {
    Private List<String> output = new ArrayList<>();
    private void Println (String message){
    .output add (message);

  }
Private void process (intx) int y){
    if (y==0){
      Println ("y is zero")
  } else if (x==0){
    Println ("x is zero");
  } else {
    for (inti=1;=1; i<=x; i++){
      if (i%y==0){
  } } } Println (String.valueof (i));
```

```
@Test
void testYIsZero(){
    output.clean();
    process(5,0);
    assertEquals(List.of("y is zero"), output
}

@Test
void testLoopDoesNotRun(){
    output.clean();
    process(0,2);
    assertTrue(output.isEmpty());
}

@Test
void testEdgeCaseXNegative(){
    output.clean();
    process(-3,2);
    assertTrue(output.isEmpty());
}
}
```

Explanation of the Junit Test class:

i. Simulating

ii. Implementing the original logic

iii. Junit Test methods

In conclusion, This Junit test class effectively tests all possible branches in the given Java program. white box testing principles are applied to ensure complete code coverage and data coverage. This implementation is structured similarly to the Python version, ensuring easy verification of results.

CamScanner

Question 21:-

Black Box Unit testing is earlier and more precise than Black Box System Testing - It can find errors very easily, even before the entire SW first version is finished. Now, consider the Production codes that need function testing. Suppose you have Junit 4 APS in your IDE and you are said to develop test codes for these production codes showing the application of Exception, Setup Function and Timeout Rule How do you solve it?

Answer to the question No: 21

Junit 4 Black Box Unit Testing Approach:

Black Box Unit Testing detects early by testing functions independently without Knowing internal logic.

To test exception handling, setup function, and time out rule, we develop Junit 4 test cases for the following production code

Production code (calculator.java)

```
Public class Calculator ?
Private int memory;
```

```
~~Poli~~ Public Calculator ()r this, memory = 0; }
Public int divide (int a , int b) r
if (b = - 0) thow new Arithmetic Execption ("cannot divide by zero")
return a/b;
}
Public double sqrt (double number) r
if (number < 0) throw new IllegalArgument Excaption
("Negative number");
try { Thred. sleep (500); } catch (Interrupted Excaption. e) r }
}
Public Void store InMemory (int value) r
this, memory = value;
}
Public int getMemory () r
return this, memory;
}
}
```

In Junit q. we can apply:-

i. Exception Handling

ii. Setup Function

iii. Timeout Rule

Junit & Test class For Function Testing:-

```
import static org.Junit.Assert.*;
import org.Junit.Before;
import org.Junit.Rule;
import org.Junit.Test;
import org.Junit.rules.Timeout;

Public class calculator Test {
    Private calculator calculator;
    Public Timeout Global Timeout = Timeout.
                                    millis(1000);

@Before
  Public void setup() {
      calculator = new calculator();
      Calculator.storeInmemory(10);
  }

@Test (expected = Arithmetic Exception.Class)
  Public void test DivideBy zero() {
      calculator.divide(5,0);
```

```
@Test
    Public void testsqrt computationTime()
    {
      double result = calculator.sqrt(16);
      assert Equals (4.0) result, 0.01);
    }

@Test
  public void testMemory Function ()
    assert Equals (10, calculator.getMomory();
  }
}
```

Explanation of the Junit 4 test class:-

i. Setup Function

ii. Exception Handling Test

iii. Timeout Rule

iv. Function Execution Test

v. setup verification

In conclusion,

This Junit 4 test class effectively applies exception testing, setup functions, and timeouts. The black box approach ensures function correctness ~~wh~~ without accessing internal logic. These techniques help detect errors early before full system testing. This meth

ensures early error detetion and efficient Function validation

CamScanner