Semester Final

Md. shahidul Islam
ID:IT-21024

Assignment-1:

i) Product Backlog for user stories:

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

Tasks:

a) Backend:

- Implement authentication
- Develop login API.
- Encrypt passwords and implement validation.

b) Frontend:

- Design login UI.
- Implement input validation.

c) Testing and Deployment:

- Write unit tests for login functionality.

User story 2: As a user, I want to search for products by category to finds items easily.

Tasks:

a. Backend

b. Frontend

c. Testing and Deployment

ii) Prioritization in Sprint Planning:

a. Value to the customer

b. Technical feasibility

c. Sprint plan Decision

iii) Tracking with a scrum board:

A scram board will track task progress with three columns:

To Do (Planned)	In Progress	Done (completed)
Develog login API.	Design login form UI	Implement password encryption
Implement seanch API	Implement search filters	Redirect usen after login
Optimize seanch performance	Integnate API with UI	Validate login finctionality

ID: IT-21024

Assignment-2 :

Agile is the best choice for this project due to its high adaptability, iterative approach and cost effective evolution. It allows frequent adjustments based on client feedback while managing risks through incremental development and continuous testing.

Comparision of methodologies for high-risk, envolving project:

1. Spiral model.
   - Risk management
   - Adaptability: Allows changes at each phase but required detailed planning.
   - Best for: High risk project.
2. Agile methodology:
   - Risk management: Reduce risk through continuous feedback.
   - Adaptibility: Highly flexible

- Best for: Project with envolving requirements and close client collaboration.

3. Extreme Programming (XP):
   - Risk management: Addresses risk through rapid itenations.
   - Adaptability: Extremly flexible but requires high client involvement.
   - Best for: Small to medium teams with rapidly changing requirements.

Assignment-3:

Comparison of Development Methodologies:

1. Waterfall:
   - Best for: Fixed requirments and strict deadlines
   - Predictability: High, as all phases are planned upfront.
   - customer collaboration: Minimal feedback

- Risk management: Limited flexibility, high risk if changes arise late.

2. Agile:
   - Best for: Evolving requirments and continuous feedback.
   - Predictability: Low, but adaptable.
   - Customer collaboration: High
   - Risk management: Lower risk due to inenemental changes.

3. Extreme Programming (XP):
   - Best for: Rapidly changing requirements.
   - Predictability: Low but ensures continuous improvement
   - customer collaboration: Very high with direct engagement.
   - Risk management: Reduces risk through test-driven development.

ID: IT-21024

4. Spiral model:

- ✓ Best for: High risk projects requiring iterative risk assessment.
- ✓ Predictability: Moderate, with flexibility for adjustments.
- ✓ Customer collaboration: Periodic feedback at each iteration.
- ✓ Risk management: Strong focus on risk analysis before each development phase.

Best methodoly for each project:

Project A → waterfall

Project B → Agile or xp

Assignment-4:

Principle of software engineering ethics:

1: Public interest: Prioritize user safety, privacy and societal well-being.

2. client & Employer Responsibility: Act in the best interest of stakeholders while maintaining honesty and integrity.
3. Product Quality: Ensure reliability, security and efficiency in software.
4. Professional competence: Continuously improve skills and apply best practices.
5. Fairness and honesty
6. confidentiality.
7. Respect for law and Ethics.

Professional Responsibility issue:

- ✓ security and privacy violation
- ✓ Bias and Discrimination

Role of ACM/IEEE code of Ethics

- ✓ Provides ethical guidelines
- ✓ Encourage transparency, fairness, and accountability.

ID: IT-21024

Assignment- 5:

Functional Requirements:

1. Flight booking: Users can search and book flights.
2. Payment processing: secure online payments via multiple methods.
3. User authentication: Login and account management for passengers.
4. Flight status and notification: Real-time updates on flight schedules.
5. Booking cancellation and refund

Non-functional Requirments:

1. Performance: System should handle 1000+ concurrent users.
2. security: Data encryption and multi-factor authentication.
3. Scalability: supports growing users and expanding features.

4. Usability: Intuition UI fast response time.

5. Maintability: Modular design for easy updates and bug fixes.

Assignment- 6:

V-model of Testing in a plan- Driven software process:

The v-model (verification and validation model) is a sequential software development process where testing activities run panallel to development activities.

Illustration of v-model:

Requirements → Acceptance Testing

Design → system Testing

Architechture → Integration testing

Coding → Unit testing

Development flows downwards (left side of v) while

connesponding testing activities occurs upwards (right side of V).

phases and connesponding testing activities:

1. Requirment analysis → Acceptance testing
2. System design → System testing
3. Architecture design → Integration testing
4. Module design → Unit testing
5. Coding

Relationship Between Development and Testing:

- Each development phase has a connesponding testing phase.
- Testing starts early, reducing defects in later stages.
- Ensure quality and minimizes project risks

ID: IT-21024

Assignment-7:

Prototype development process: Prototype Development Process involves creating a preliminary version of the software to understand requirments and gather user feedback.

Key stages of Prototyping:

1. Requirment gathering
2. Quick design
3. Prototype development
4. User evaluation
5. Refinement based on feedback

Benefits of prototyping:

- User feedback: Helps refine requirments early and improve usability.
- Risk Reduction: Identifies potential issues early.
- Iterative Development: Enables continuous improvment and adaptation to user

The prototyping model enhancing clarity, reduces risks, and ensures a better final product throughout the development process.

Assignment- 8:

Process Improvement cycle:

The process improvement cycle in software engineering involves continuously evaluating and improving software development processes to increase efficiency and quality.

key stages of the process Improvement cycle:

1. Process measurements: collect data on current processes.

2. Process Analysis: Analyze the collected data to identify inefficiencies.

3. Process improvement

4. Process implementation

5. Process control.

Commonly used process metrics:

1. Defect density
2. code churn
3. Cycle time
4. velocity

These metrics help monitor and refine the software process, enhancing efficiency, quality and overall performance.

P.t.o.

ID: IT-21024

#Assignment-09:

SEI Capability Maturity Model (CMM):

The SEI capability maturity model (CMM) is a framework development by the software engineering Institute to helps organizations improve their software development processes. It defines five maturity levels that indicate the maturity of an organization's processes and provide a path for improvement.

Five levels of CMM:

1. Level 1: Initial (Ad hoc)
   - Description: Process are unpredictable, poorly controlled and reactive.
2. Level-2: Managed

   Increased project predictability and better management practices lead to improve

efficiency and reduced project failure.

3. Level-3: Defined
With standard processes in place, teams can follow best practices.

4. Level-4: Quantitavely managed
Performance is measured and managed using data, enabling more accurate prediction.

5. Level-5: Optimizing
The organization continuously refines its processes, leading to ongoing improvement in quality, cost reduction and innovation ensuring long-term success and competitiveness.

Impact on software Development and organizational performance:
The cmm provides organizations with a structured approaches to software process improvement. By Progressing through each level organization

can systematically enhance theme development practice, leading to higher quality software, improved project management and better overall performance.

Assignment - 10:

Core Principles of Agile software Development:

1. Customer collaboration
2. Responding to change
3. Working software
4. Individuals and Interactions
5. Sustainable Development
6. Continuous improvement

Application in different Environments:

✓ Startups: Rapid Prototyping and quick iteration.

ID: IT-21024

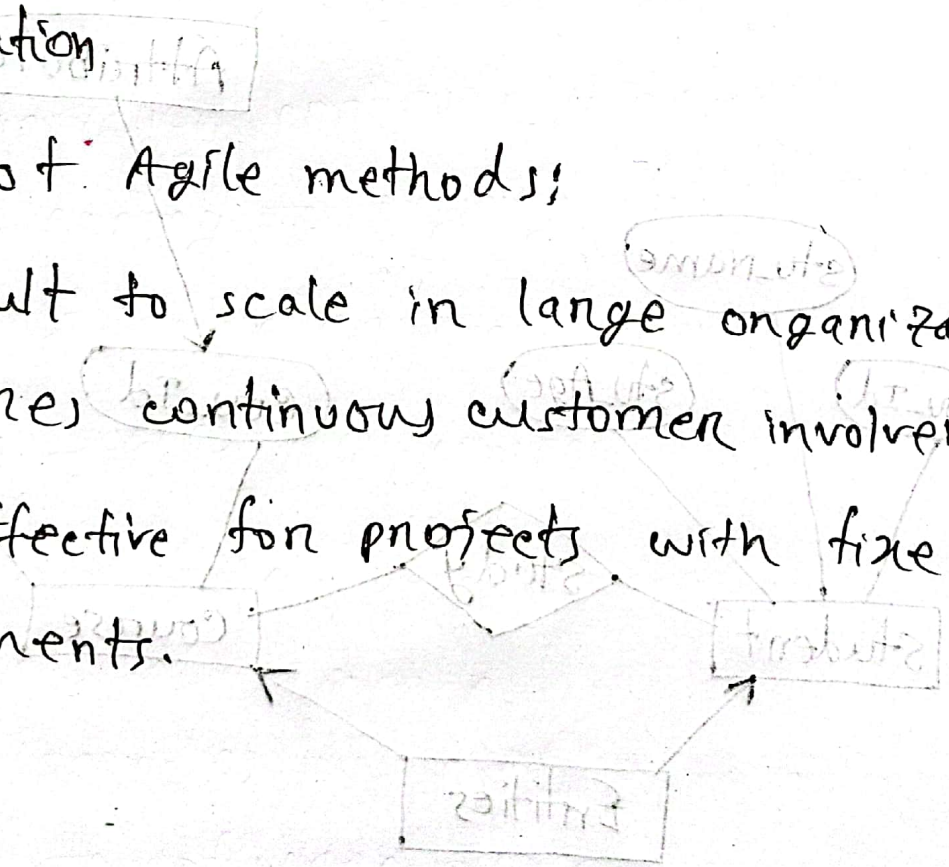
- Large enterprises: Scaled Agile (SAFe) to align multiple teams.
- Regulated Industries: Hybrid model balancing Agile flexibility with compliance.

Benefits of Agile methods:

- Faster delivery and early feedback
- Higher adaptability to changing requirments.
- Increased collaboration and team motivation.

Challenges of Agile methods:

- Difficult to scale in large organizations.
- Requires continuous customer involvement.
- Less effective for projects with fixed requirments.



ID: IT-21024

Assignment-12

Entity Relationship diagram: ER diagram is a diagram that represents relationships among entities in a database. It is commonly known as an ER diagram. An ER Diagram in DBMS plays a crucial role in designing the database.

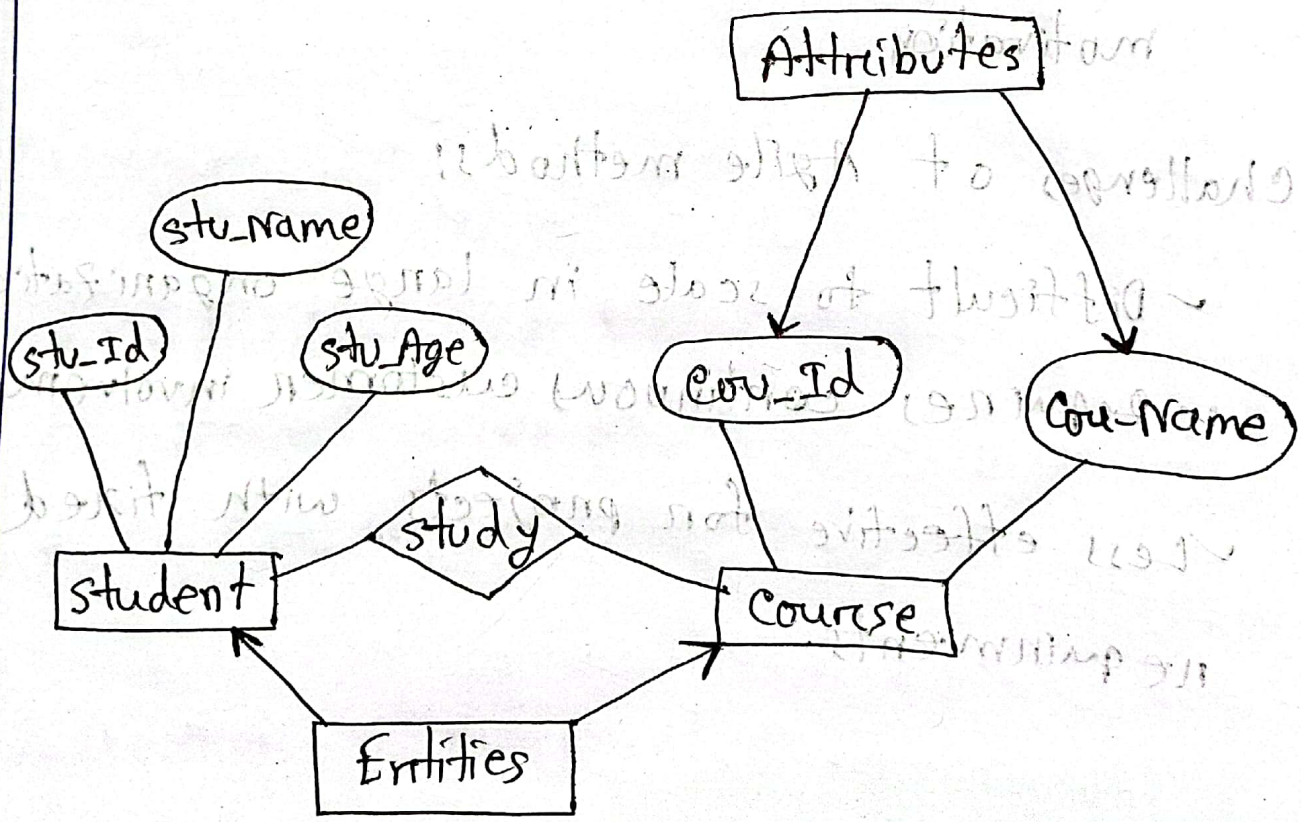


figure: ER Diagram

ID: IT-21024

The following diagram showcase two entities- student and course and their relationship. The relationship described between student and course is many-to-many as a course can be opted by several students and a student can opt for more than one course. student entity possesses attributes - stu_Id, stu_name and stu_Age. The course entity has attributes such as Cou_ID and Cou_name.

Why use ER Diagram in DBMS?

ER diagram helps you conceptualize the database and lets you know which fields need to be embedded for a particular entity.

ER Diagram gives a better understanding of the information to be stored in a database

IT-21024

Assignment: 11

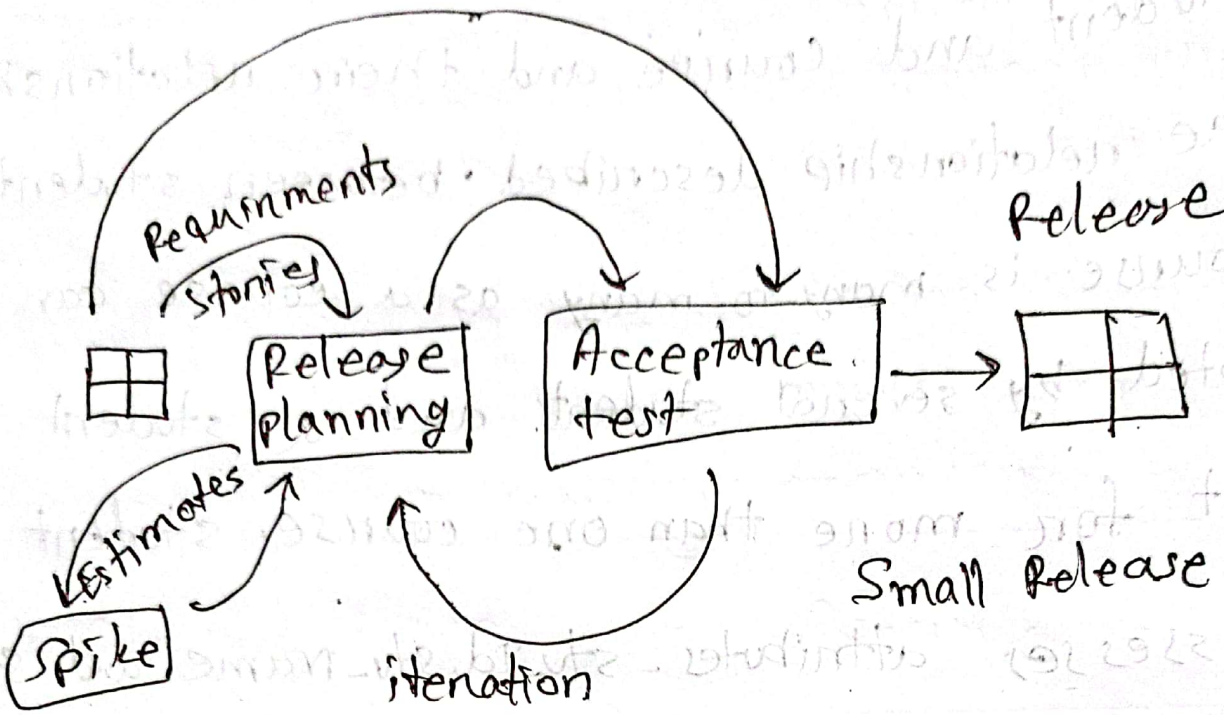


Figure: Release cycle of Extreme programming

The extreme programming (XP) release cycle follows an iterative approach with continuous feedback and improvements. The cycle consist of:

1. Exploration phase
2. Planning phase
3. Iteration phase

4. Release phase
5. maintenance phase
6. Next iteration begin

Assignment-13:

Software testing: Software testing is the proces of evaluating a software application to ensure that it meets the specified requirments and functions connectly. It helps identify defects, errors and gaps in the software before deployment.

Difference between validation and verification are given below:

IT-21024

Aspect	Verification	Validation
Definition	Ensures the software is being built correctly.	Ensures the right software is built as per user needs.
Focus	Checking documents design and code before execution	Checking the actual product after execution
Process type	Static	Dynamic
Objective	Confirms adherence to specifications.	Confirms fulfillment of customer requirements
Example	Reviewing SRS	Performing unit system

IT-2024

Assignment-14:

Layered Architechture Model:

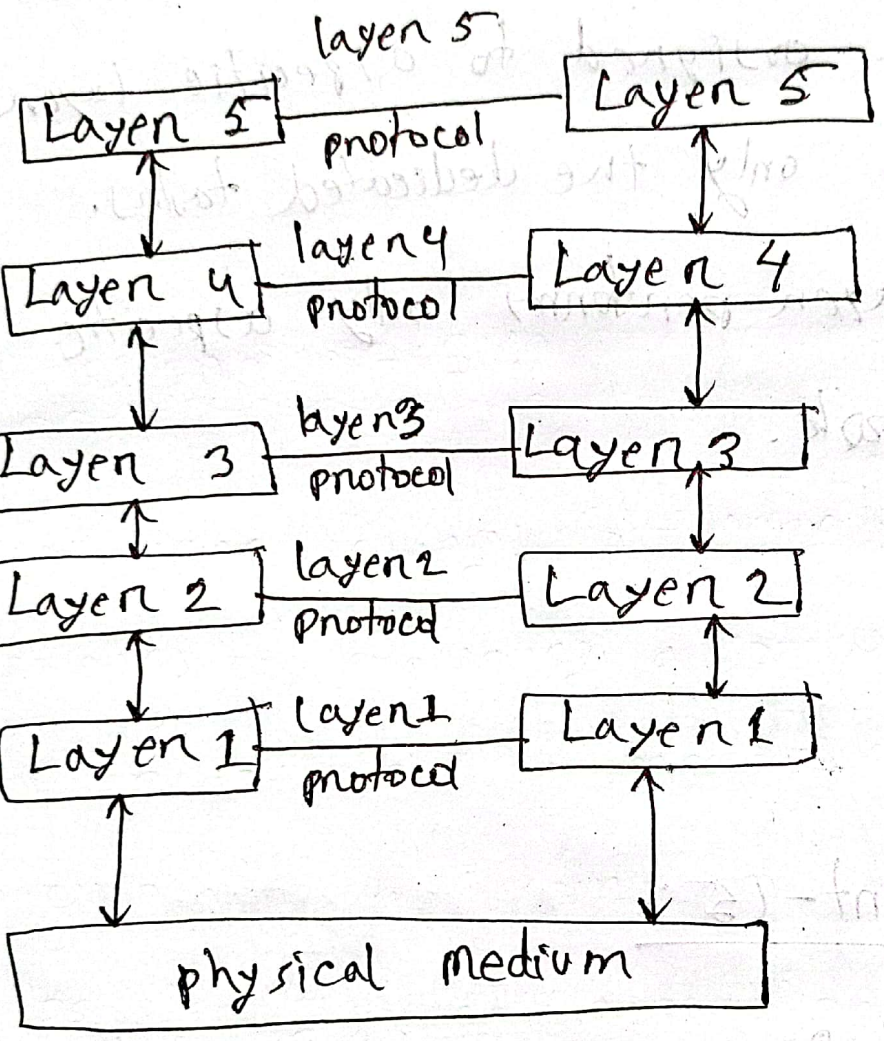
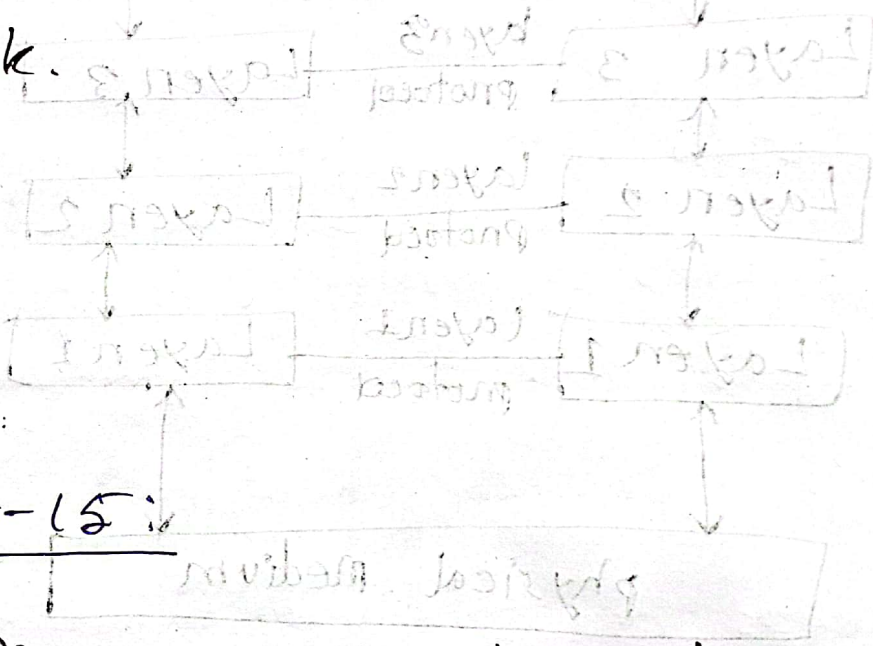


Fig: Layered Architechture model

Every network consists of a specific number of functions layer and tasks to perform.

Layered architechture in a computer network

is defined as a model where a whole network process is divided into various smaller sub-tasks. These divided sub tasks are then assigned to a specific layer to perform only the dedicated tasks. A single layer performs only a specific type of task.



Assignment-05:

Data Flow Diagram is the abbreviation for Data Flow Diagram. The flow of data in a system or process is represented by a Data Flow Diagram (DFD). It also gives insight into the inputs and outputs of

each entity and the process itself.

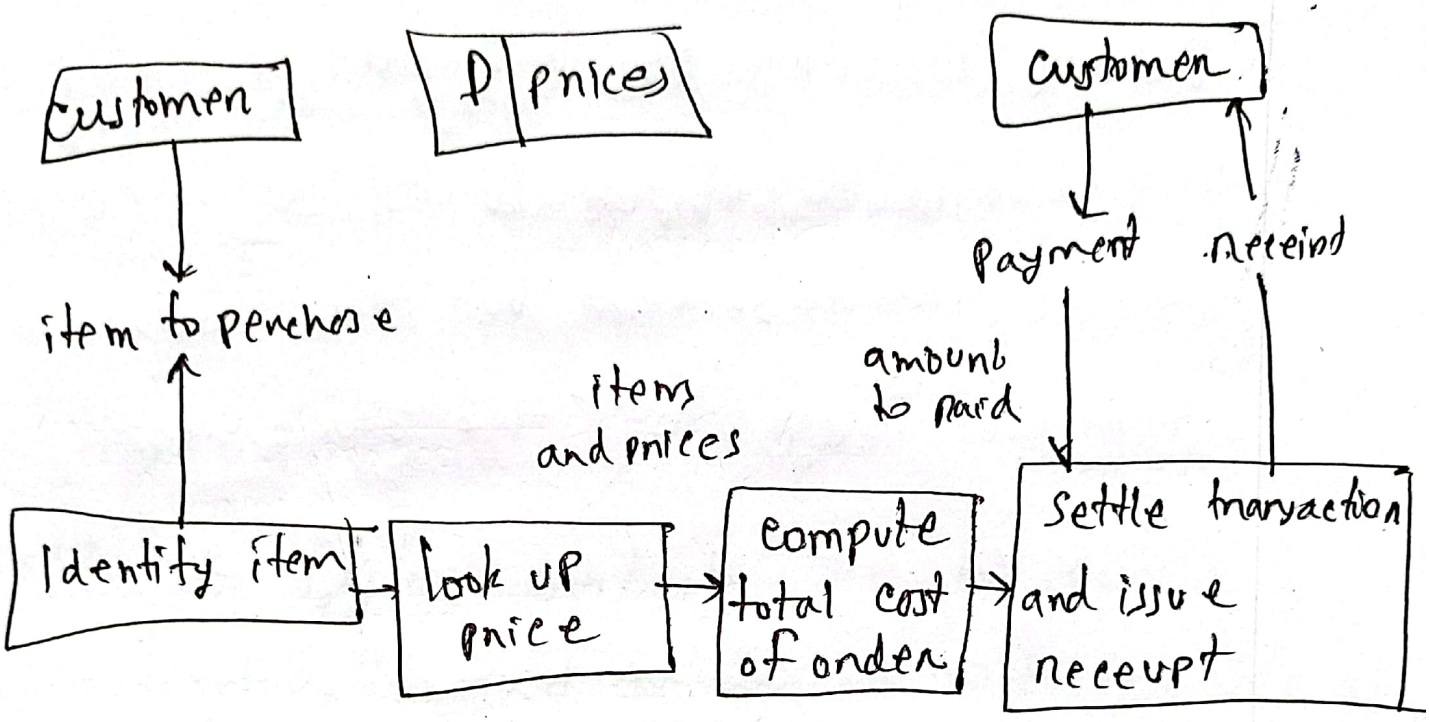


fig: Logical Data Flow Diagram (DFD)

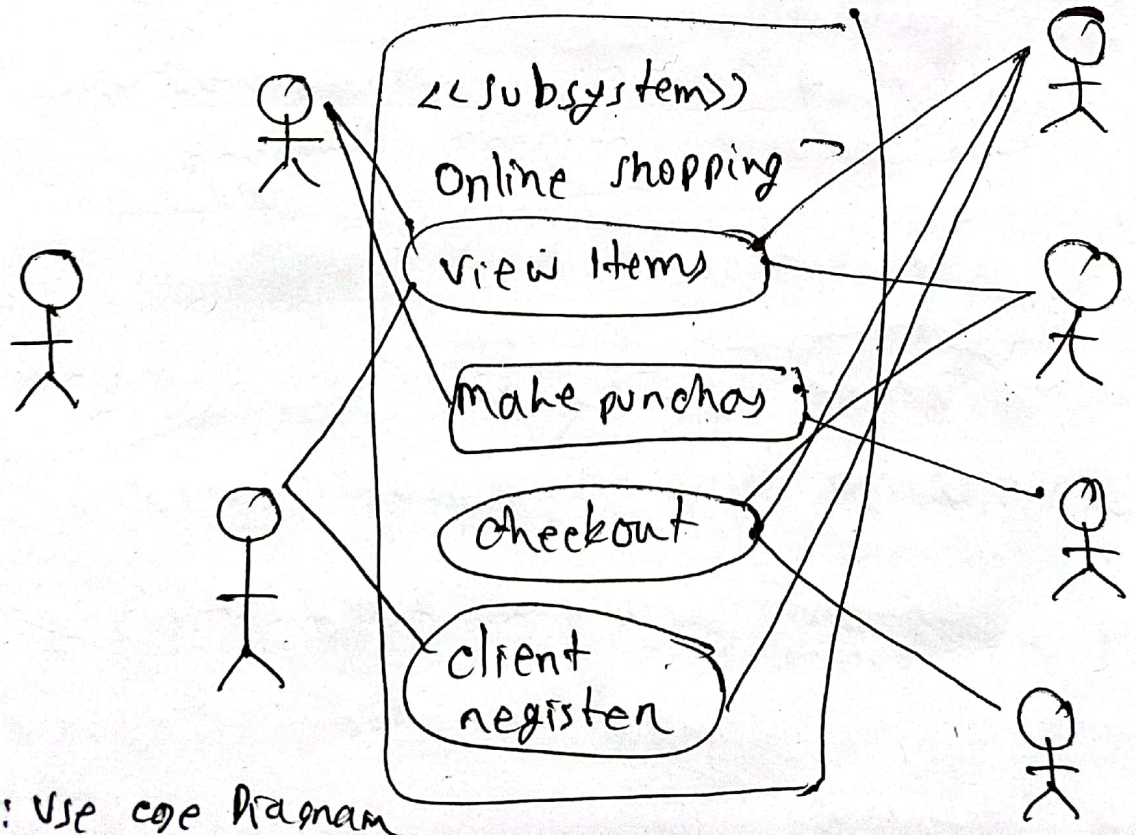
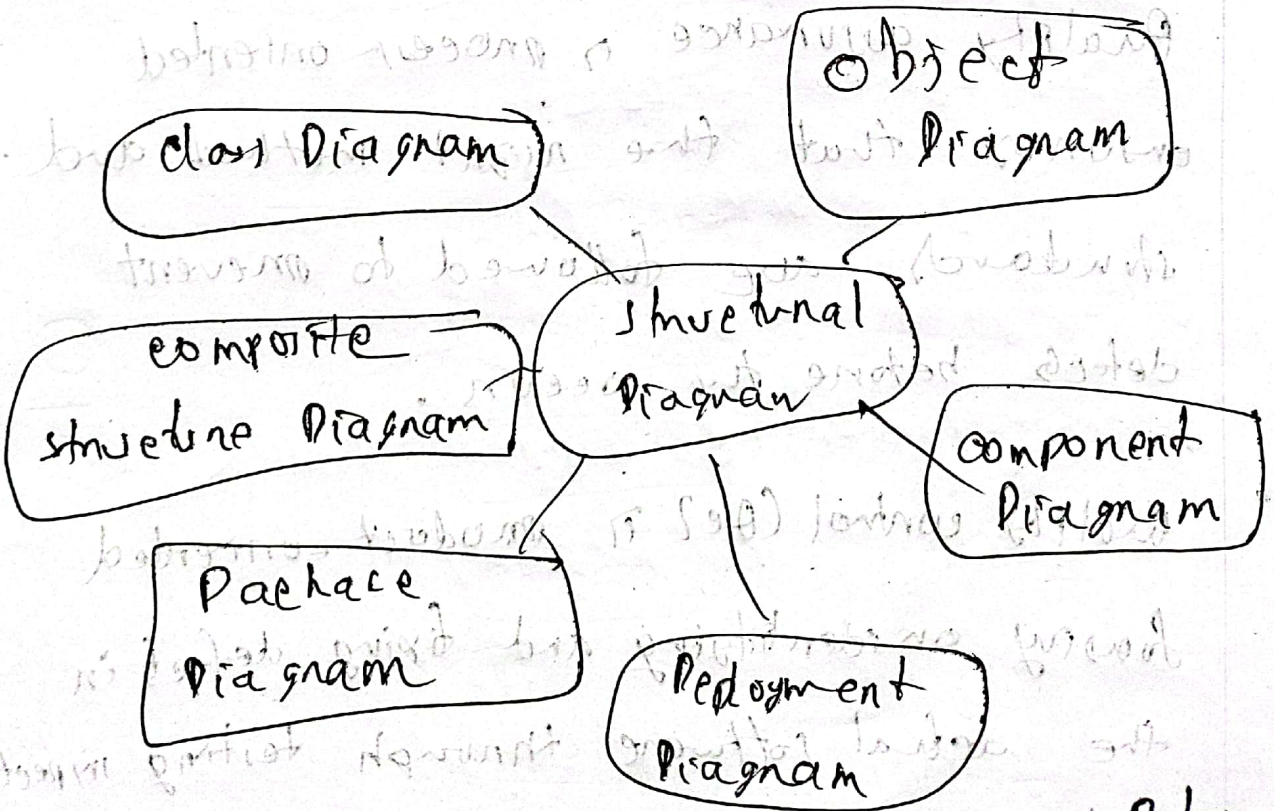


fig: Use case Diagram

IT-21029

Assignment-16:

Unified modeling Language (UML) is a standardized visual modeling language that is a versatile, flexible and user-friendly method for visualizing systems design. Software system artifacts can be specified visualized built and documented with the use of UML.



P.t.o.

IT-21024

Assignment- 17:

Quality Assurance (QA) and Quality control (QC) are essential components of software quality management but they focus on different aspects of the development lifecycle.

Explanatory Description:

Quality assurance is process oriented ensures that the right methods and standards are followed to prevent defects before they occur.

Quality control (QC) is product-oriented focuses on identifying and fixing defects in the actual software through testing inspections and reviews.

IT-21024

Difference between QA and QC:

Quality Assurance	Quality control
Processes and standards	Product testing and Defect Detection
Prevent defects	Identify and fix defects
Proactive	Reactive
Everyone in the team	Dedicated testers and QC teams

Assignment-18:

Role of Quality Assurance (QA) in Each phase of the SDLC:

Quality assurance (QA) is not just about finding bugs - It ensures the entire software development

process follows best practices to prevent defects and improve quality. QA plays a proactive role in every phase of the Software Development Life Cycle (SDLC).

1. Requirment analysis phase:
   - Validate clarity, completeness
   - Identify inconsistencies or missing requirments early.
2. Design phase
3. Development phase
4. Testing phase
5. Deployment phase
6. Maintenance & support phase

IT-22024

Assignment-19:

Rapid Application Development (RAD) Model in software Engineering.

The Rapid Application Development (RAD) model is an iterative adaptive software development methodology focused on quick prototyping, user feedback and rapid delivery while maintaining quality. It is ideal for projects requiring fast development and continuous user involvement.

key phase of RAD model:

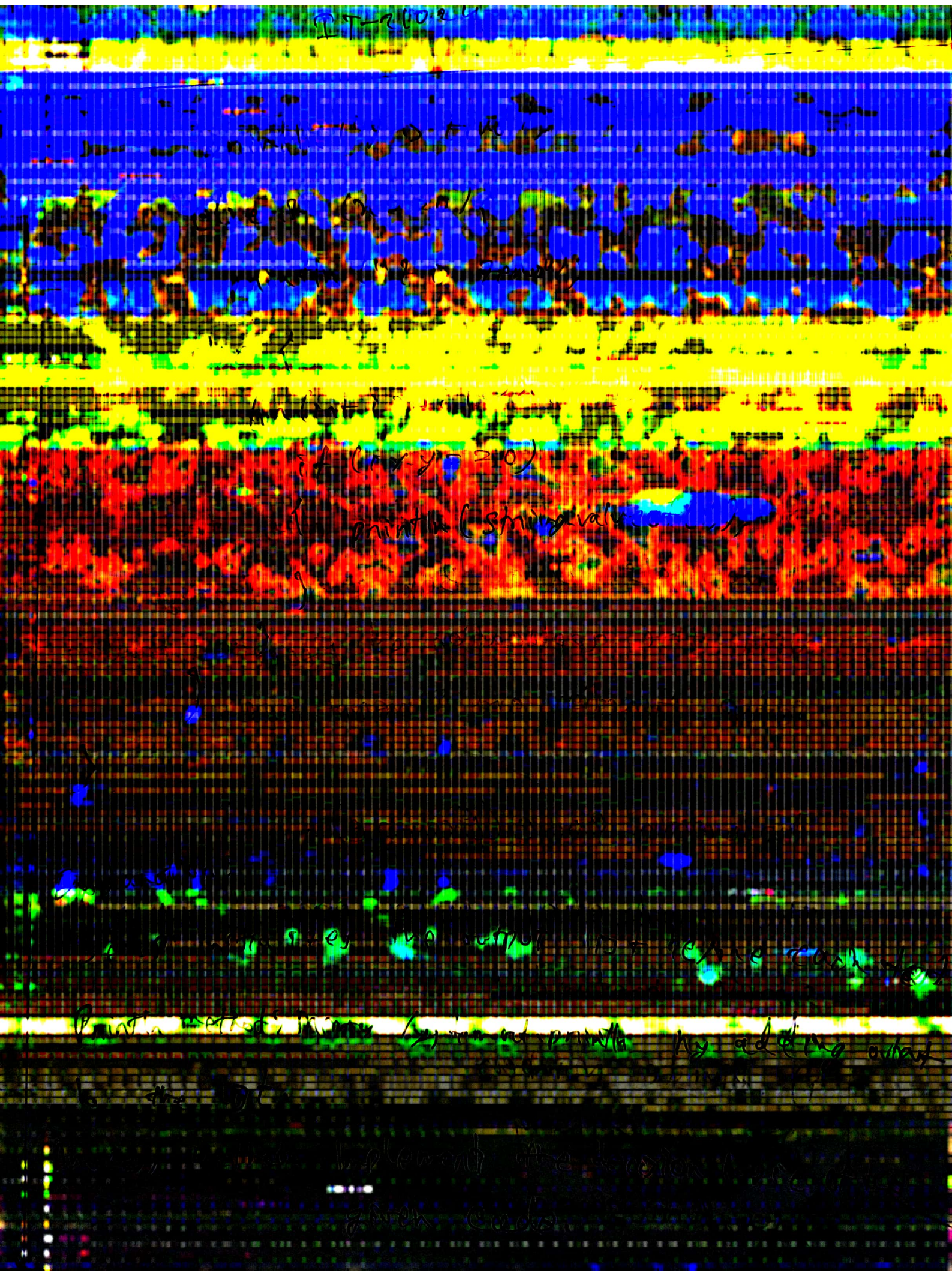
1. Business modeling
2. Data modeling
3. Process modeling
4. Application Gen
5. Testing and

Assignment-20:

Here is the equivalent JUnit test class in Java, following the same structure as the python example:

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import java.util.ArrayList;
import java.util.List;

class CalculatorTest {
    private List<String> output;
    @BeforeEach
    void setup() {
        output.add(message);
    }
    void process(int x, int y) {
        if (y == 0) {
```



IT-21024

```
public class ProductionCode {
    Public int divideNumber(int a, int b) {
        if (b==0)
        {
            throw new ArithmeticException("can not");
        }
    }
    public String concatenateString(String str1, String str2)
    {   return str1+str2;
    }
    public void longRunningProcess() throws InterruptedException
    {   Thread.sleep(1000);
    }
}
```

JUnit 4 Test Class:

- setup function
- Exception Handling
- Timeout Rule

IT-2 2024

Explanation:

- Before → Initialize
- Test (ExpectedArithmeticException.class)
- Test (timeout = 1000)