1st YEAR COMPUTER SCIENCE SOLVED EXERCISE

UNIT #01 — Introduction to software development

Prepared by: Mahmood Khan | MS Cyber Security
Codymentor Software House Sahiwal | 03186879182

Codymentor Software House

Multiple Choice Questions

1. Primary purpose of the Software Development Life Cycle (SDLC) is to:

a) design websites

b) deliver high-quality software within time and cost estimates

c) manage database systems

d) create hardware components

2. A type of requirement specifying system performance:

a) Functional Requirements

b) non-Functional Requirements

c) Technical Requirements

d) Operational Requirements

3. Role of a framework in the context of SDLC is to:

a) write code from scratch

b) provide a structured foundation with predefined components and architectures

c) manage hardware

d) perform manual testing

4. Software development model involving short cycles or sprints:

a) Waterfall Model

b) Agile Methodology

c) Lean Software Development

d) Scrum

5. Crucial aspect of comprehensive project planning:

a) Understand the project scope and tasks

b) Deciding the project' color scheme

c) Hiring a large development team

d) Ignoring the potential risks

Codymentor Software House

6. Factor that does not influence cost estimation of a software project:

a) Scope of the project
b) Technology stack

c) Number of meetings held

d) Operational costs

7.The purpose of Use Case Diagrams is to:

a) document the system's architecture

b) identify and document the system's functional requirements

c) illustrate the database schema

d) define the system's user interface design

d) define the system's user interface design

MCQ's Answer Key:

Q. No	1	2	3	4	5	6	7
Answer	**b**	**b**	**b**	**b**	**a**	**c**	**b**

Codymentor Software House

Short Questions

1. Differentiate between functional and non-functional requirements.

answer: Functional Requirement

Functional requirements describe the specific behaviors or functions of a system. These requirements outline what the system should do and include tasks, services, and functionalities that the system must perform.

- They define the interactions between the system and its users or other systems.

Example:

login, registration, or generating reports.

Non-functional Requirement:

Non-functional requirements define the quality attributes, performance criteria, and constraints of the system. These requirements specify how the system performs a function rather than what the system should do.

Example:

security, speed, reliability, and usability

2. Explain why the testing phase is important in the Software Development Life Cycle (SDLC), and provide two reasons for its significance.

Answer:

Testing is important because it identifies and fixes errors before the software is delivered to users.

Reasons:

1. It improves software quality and reliability.
2. It helps ensure the software meets user requirements.

3. Illustrate the concept of continuous integration in Agile Methodology and discuss its importance in software development.

Answer:

Continuous Integration (CI) means developers frequently integrate their code into a central-repository, where automated tests check the changes.

Codymentor Software House

Importance: It helps

- detect errors early
- reduces integration problems
- improves software quality

4. Evaluate the main steps involved in risk assessment and management, and assess their importance in a software project.

Answer:

Risk assessment and management are crucial aspects of any software project. They involve identifying potential risks that could impact the project's success, analysing the likelihood and impact of these risks, and developing strategies to manage them.

Steps in Risk Assessment and Management:

1. Identify Risks: List all potential risks that could affect the project. These could be technical risks, such as technology changes; operational risks, like resource shortages; or external risks, such as market fluctuations.

2. Analyze Risks: Evaluate the likelihood of each risk occurring and its potential impact on the project.

3. Develop Mitigation Strategies: For each significant risk, develop a plan to

reduce its likelihood or minimize its impact.

4. Monitor and Review: Continuously monitor the project for new risks and review existing risks to adjust strategies as necessary.

Importance: These steps help reduce project failures, delays, and unexpected costs.

5. Explain the purpose of a Use Case Diagram in software development.

Answer:

A Use Case is a description of a set of interactions between a user (actor) and a system to achieve a specific goal.

Purpose

The purpose of a Use Case is to identify the functionalities that the system must support to meet the user's needs and describe the complete workflow from the user's perspective.

Codymentor Software House

6. Compare and contrast a Sequence Diagram with an Activity Diagram, highlighting the key differences.

Answer:

Sequence Diagram	Activity Diagram
1. Shows the order of interactions/messages between objects over time.	1.Shows the flow of activities or processes in a system.
2. Focuses on object interactions.	2.Focuses on workflow/process flow.
3. Shows how objects communicate with each other.	3.Shows how a process moves from one activity to another.

7. Describe the Factory Pattern and explain how it differs from directly creating objects, with an example.

Answer:

The Factory Design Pattern is like having a special workshop that knows how to create different products, but you don't need to worry about the details of how those products are made. Instead, you just tell the factory what you need, and it gives you the finished product.

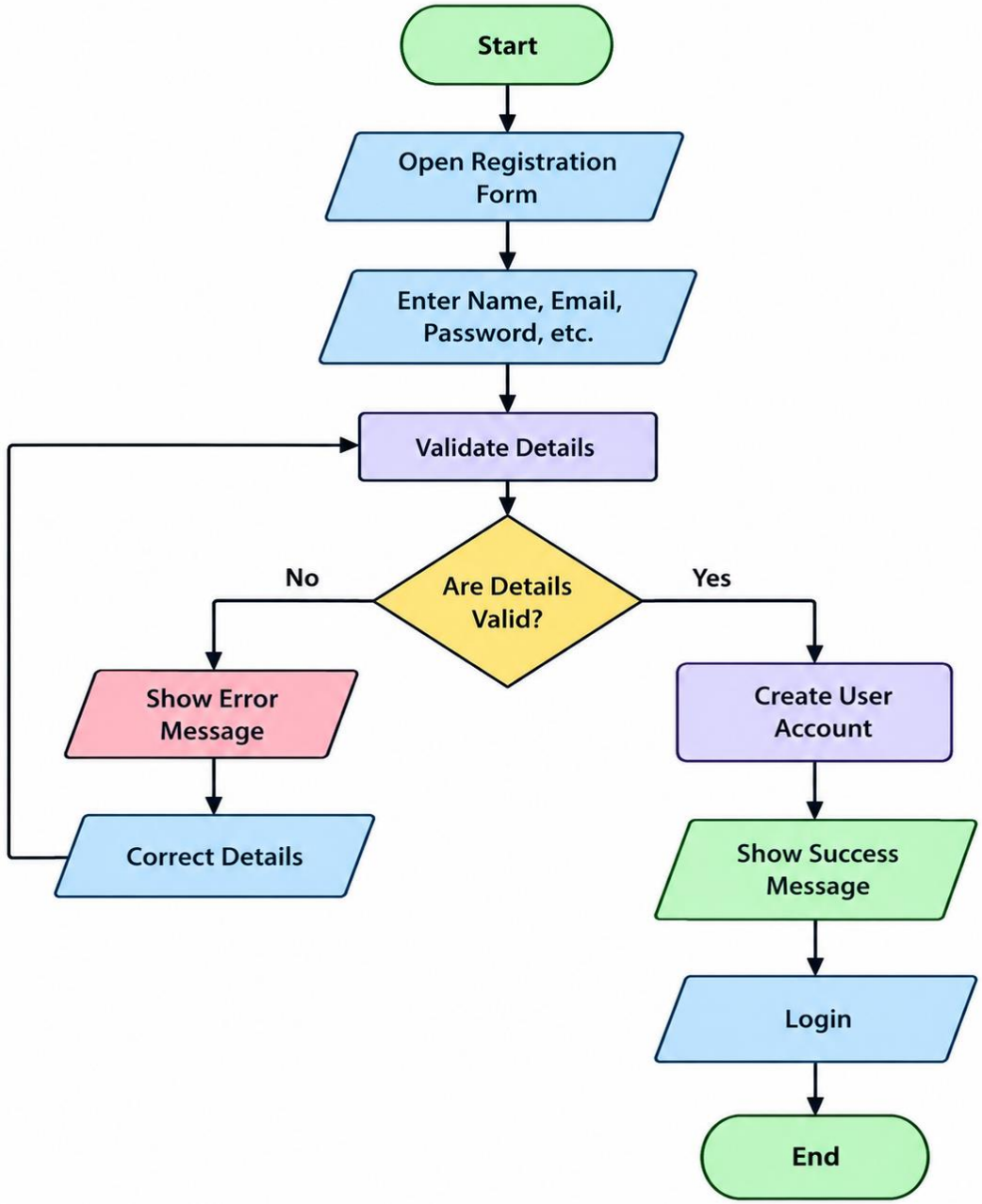
Example: Instead of writing new Car () or new Bike () directly, a factory can create the required vehicle based on the user's choice.

It makes object creation more flexible and easier to manage.

Codymentor Software House

Long Questions

1. **Design a flowchart for a user registration process in a software application. Outline its key steps.**
   Answer:
   flowchart for a user registration process in a software application:



Key steps flowchart for a user registration process in a software application:

1. **Start**
2. **Open Registration Form**
3. **Enter User Details** (Name, Email, Password, etc.)
4. **Validate Details**
5. **Are Details Valid?**
6. **No** → Show Error Message → Correct Details → Validate Again
7. **Yes** → Create User Account
8. **Show Success Message**
9. **Login**
10. **End**

2. Imagine you are managing a project to develop a simple mobile application. Describe how you would use the Agile Methodology to handle this project.

Answer:

Using Agile Methodology for a Mobile Application Project

Agile Methodology is a flexible and adaptive approach that focuses on developing and delivering small, functional parts of software quickly.

How I Would Use Agile

I would divide the mobile application project into short development cycles called sprints or iterations.

- Sprint 1: Develop user registration and login.
- Sprint 2: Develop the user profile.
- Sprint 3: Develop the main application features.
- Sprint 4: Testing, review, and improvements.

Activities in Each Sprint

At the end of each sprint, the developed features would be:

1. Tested to ensure they work correctly.
2. Reviewed by the team and stakeholders.
3. Deployed when ready.

Codymentor Software House

4. Improved based on feedback and changing requirements.

The team could also use Agile practices such as:

- Continuous Integration: Regularly merge code changes to detect and fix issues early.
- Test-Driven Development: Write tests before writing the code.
- Pair Programming: Two developers work together, with one writing code and the other reviewing it.

Benefits

- High Flexibility: Changes in requirements can be handled even after development has started.
- Improved Customer Satisfaction: Customers receive regular updates and can provide frequent feedback.

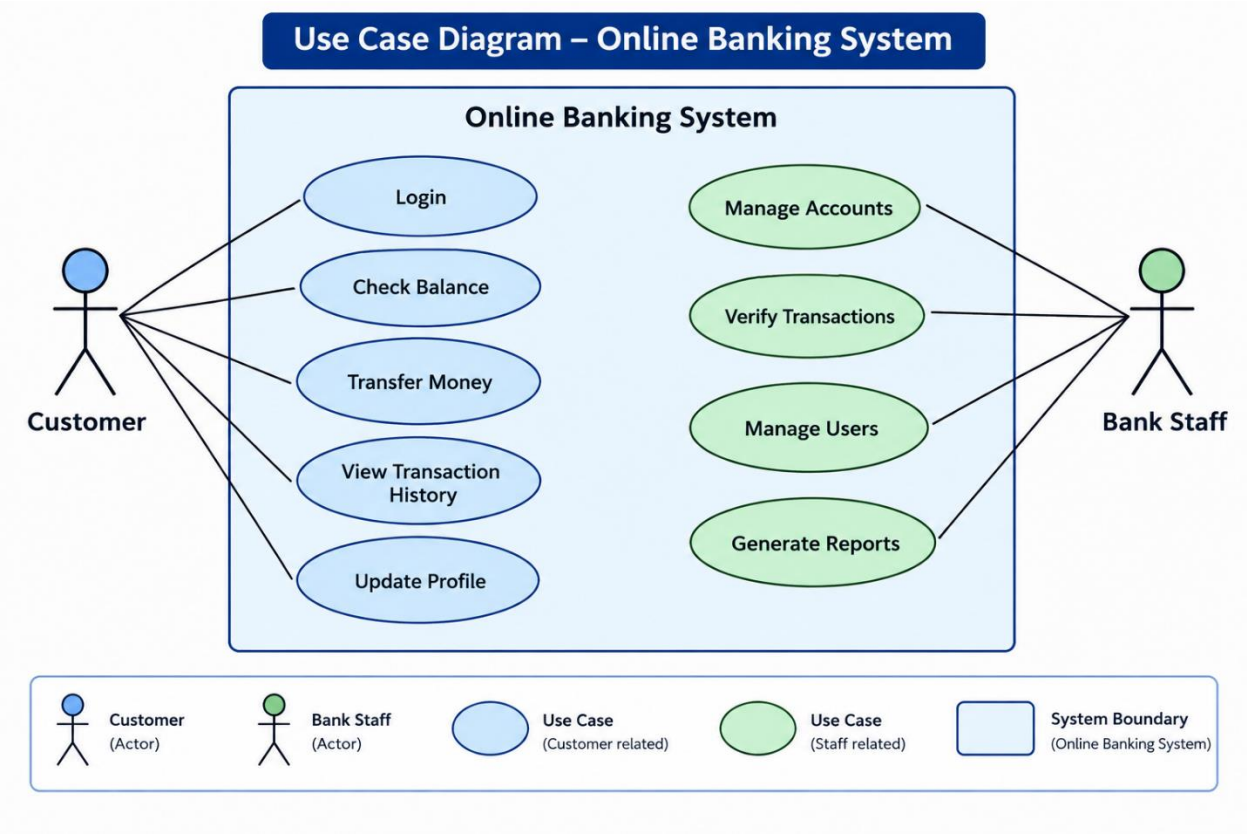
Conclusion

Thus, Agile would help me deliver functional software incrementally, gather feedback early, adapt to changes, and continuously improve the mobile application.

Codymentor Software House

3.Consider an online banking system. Create a Use Case Diagram to show the interactions between customers, bank staff, and the system.

Use Case Diagram:



Actors: Customer and Bank Staff.

Customer Use Cases: Login, check balance, transfer money.

Bank Staff Use Cases: Manage accounts, verify transactions, manage users.

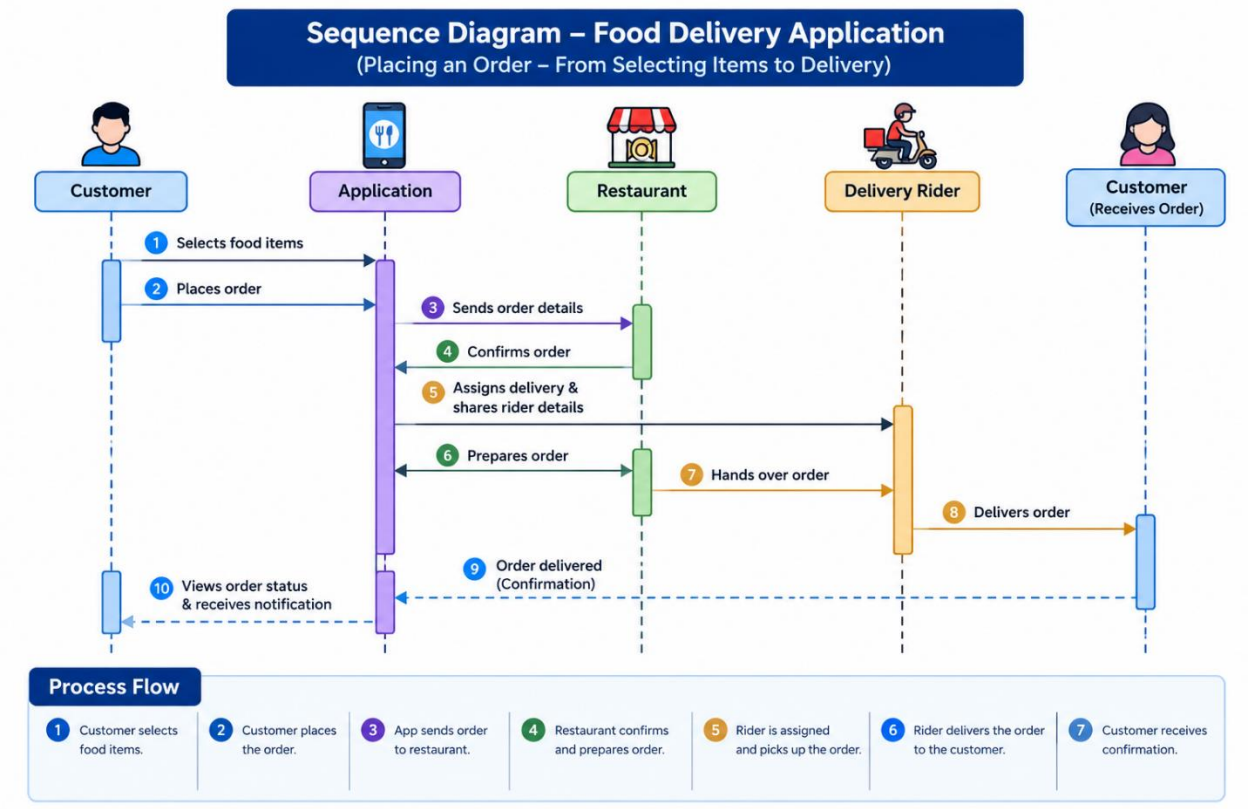
The diagram shows how different actors **interact with the online banking system**.

Codymentor

4. You are developing a food delivery application. Create a Sequence Diagram to show the process of placing an order, from the customer selecting items to the delivery of the order.

Answer:

Sequence Diagram:



Process:

1. Customer selects food items.
2. Customer places the order.
3. Application sends the order to the restaurant.
4. Restaurant confirms the order.
5. A rider is assigned for delivery.
6. Rider collects the order.
7. Order is delivered to the customer.

5. Discuss the importance of Software Development Tools in the software development process.

Software development tools are programs or applications that assist in various stages of software creation. They are used to **write, edit, test, debug, and manage code**, ensuring that software functions correctly and efficiently.

a) Role of Language Editors, Translators, and Debuggers

1. Language Editors

Language editors, also known as **code editors**, are tools that help developers **write and edit code in different programming languages**.

They provide an environment where programmers can create and modify source code easily.

Role:

- Help developers write source code.
- Allow developers to edit and modify existing code.
- Make code development easier and more organized.

2. Translators

Translators are tools that **convert code written in one programming language into another language that the computer can understand**.

They convert **high-level programming languages**, such as Python, into **machine language (binary code)** that computers can execute.

There are two main types of translators:

- **Interpreters:** Translate and execute code **line-by-line**.
- **Compilers:** Translate the **entire code at once**.

Role:

- Convert high-level code into a form that the computer can execute.
- Make programming languages understandable to the computer.
- Help developers run their programs correctly.

3. Debuggers

Debuggers are tools that help developers **find and fix errors (bugs) in their code**.

Codymentor Software House

The purpose of a debugger is to allow developers to **test their code and identify where errors occur**.

Role:

- Help identify errors in programs.
- Help developers find where an error occurs.
- Assist in fixing bugs.
- Improve the correctness and reliability of software.

b) Examples and Contribution to Efficiency and Accuracy

Tool	Example	Contribution
Language Editor	Notepad++, VS Code	Helps developers write and edit code easily and efficiently.
Translator – Interpreter	Python Interpreter	Translates code line-by-line so it can be executed by the computer.
Translator – Compiler	GCC for C/C++	Translates the entire program at once into a form the computer can execute.
Debugger	GDB, Visual Studio Debugger	Helps developers find and fix errors, improving the accuracy and reliability of software.

Conclusion

Language editors, translators, and debuggers are important software development tools because they support different stages of software creation. **Editors help write and modify code, translators convert code into a form the computer can understand, and debuggers help find and fix errors.** Together, these tools improve the **efficiency, accuracy, and correctness** of software development.

Codymentor Software House