ASSIGNMENT-3.1

Given:
SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.

Software Development Life Cycle (SDLC)

1. Requirements

- **Importance**:
    - Gathering and documenting customer needs and expectations.
    - Defining project scope, objectives, and constraints.
- **Interconnection**:
    - Forms the foundation for all subsequent phases.
    - Ensures alignment with stakeholder expectations.

2. Design

- **Importance**:
    - Creating the architecture and detailed design of the system.
    - Specifying hardware and system requirements.
- **Interconnection**:
    - Translates requirements into a blueprint for development.
    - Guides the implementation phase.

3. Implementation

- **Importance**:
    - Writing the code and building the system.
    - Transforming design documents into a working product.
- **Interconnection**:
    - Follows the design specifications.
    - Results in a functional system ready for testing.

4. Testing

- **Importance**:
    - Verifying and validating the system against requirements.
    - Ensuring the system is free of defects and works as intended.
- **Interconnection**:
    - Checks the implementation phase output.
    - Identifies issues to be fixed before deployment.

5. Deployment

- **Importance**:
  - Releasing the system to production.
  - Making the system available for end-users.
- **Interconnection**:
  - Follows successful testing.
  - Involves final checks and user training.

Infographic Visualization:

1. **Title**: Software Development Life Cycle (SDLC)
2. **Visual Flow**: Use arrows to show the flow from one phase to the next, indicating the cyclical nature of SDLC.
3. **Icons**:
   - Requirements: Clipboard with checklist.
   - Design: Blueprint or architectural drawing.
   - Implementation: Code or computer screen.
   - Testing: Test tube or magnifying glass.
   - Deployment: Rocket or cloud.

ASSIGNMENT-3.2

Given: Requirements Gathering - Conduct a 30-minute mock interview to gather requirements for a fictional app that helps organize community events. Summarize the requirements and how you would document and trace them in a one-page brief.

Summary of Requirements:

1. Event Management:

(i)Create, edit, and manage event details (dates, locations, categories).

(ii)Event promotion features (social media sharing, invitations).

2. Attendee Management:

(i) Track RSVPs and manage attendee lists.

(ii) Communication with attendees via notifications/messages.

3. User Experience:

(i) Intuitive interface with search and filtering options.

(ii) Support for different user roles and permissions.

4. Additional Functionality:

(i) Integration with mapping services.

(ii) Feedback and rating system for events.

Documentation and Tracing:

1)Requirements Document:

- Create a concise document summarizing each requirement.
- Include stakeholder names, descriptions, and acceptance criteria.
- Specify priority (must-have, nice-to-have).

2)Traceability Matrix:

- Create a matrix linking requirements to design, development, and testing phases.
- Ensure each requirement is addressed in the project plan.
- Track changes and updates throughout the project lifecycle.

ASSIGNMENT-3.3

Given: Agile Principles Application - Write a two-paragraph reflection on how the Agile values of individuals and interactions, working solutions, and customer collaboration apply to the development of the community event app.

Agile Principles Application in the Development of the Community Event App:

Individuals and Interactions:

- Foster effective communication and collaboration among team members.
- Emphasize continuous interaction to share insights and feedback.
- Quickly adapt to changes and resolve issues collaboratively.
- Enhance team cohesion and shared understanding.

Working Solutions:

- Prioritize building a functional, user-friendly application.
- Deliver working software continuously for real-time feedback.
- Ensure the app evolves in alignment with user expectations.
- Focus on immediate usability over extensive documentation.

Customer Collaboration:

- Maintain a close relationship with stakeholders (event organizers, community members).
- Conduct regular feedback sessions and collaborative planning.
- Ensure the app addresses real-world challenges and enhances user experience.
- Build trust and satisfaction among customers for long-term success.

ASSIGNMENT-3.4

Given: Scrum Framework Overview - Prepare a one-page cheat sheet on the Scrum framework that includes roles, responsibilities, artifacts, and ceremonies. Provide a brief example of a Sprint task list for the earlier mentioned app project.

Scrum Framework Cheat Sheet:

Roles and Responsibilities

1. Product Owner

- Responsibilities:
  - Define and prioritize the product backlog.
  - Communicate vision and goals to the Scrum team.
  - Ensure the team delivers value to the business.
  - Collaborate with stakeholders and gather feedback.

2. Scrum Master

- Responsibilities:
  - Facilitate Scrum ceremonies.
  - Remove impediments and obstacles for the team.
  - Ensure adherence to Scrum practices and principles.
  - Coach the team for continuous improvement.

3. Development Team

- Responsibilities:
  - Deliver potentially shippable product increments each Sprint.
  - Self-organize and determine the best way to accomplish work.
  - Collaborate and communicate effectively within the team.
  - Participate in Scrum ceremonies and provide input.

Artifacts

1. Product Backlog

- An ordered list of all desired work on the project.
- Managed and prioritized by the Product Owner.

2. Sprint Backlog

- A selection of Product Backlog items chosen for the current Sprint.
- Includes a plan for delivering the product increment.

3. Increment

- The sum of all Product Backlog items completed during a Sprint.
- Must be in a usable condition and meet the Definition of Done.

Ceremonies

1. Sprint Planning

- **Purpose**: Define what can be delivered in the Sprint and how to achieve it.
- **Participants**: Scrum Master, Product Owner, Development Team.
- **Outcome**: Sprint Goal and Sprint Backlog.

2. Daily Scrum

- **Purpose**: Synchronize activities and create a plan for the next 24 hours.
- **Participants**: Development Team.
- **Outcome**: Updated plan to achieve the Sprint Goal.

3. Sprint Review

- **Purpose**: Inspect the increment and adapt the Product Backlog if needed.
- **Participants**: Scrum Team, stakeholders.
- **Outcome**: Feedback on the increment and updated Product Backlog.

4. Sprint Retrospective

- **Purpose**: Reflect on the past Sprint to improve processes and teamwork.
- **Participants**: Scrum Team.
- **Outcome**: Actionable improvement plan.

Example Sprint Task List for Community Event App:

1. Develop user registration feature

2. Implement event creation functionality

3. Design event listing page UI

4. Integrate mapping service for event locations

5. Set up push notification system for event reminders

6. Test attendee RSVP functionality

7. Optimize app performance for faster loading times

8. Create user documentation and help section

9. Conduct user acceptance testing with stakeholders

10. Prepare for sprint review and demonstration to stakeholders.