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 Phase:

- **Importance:**
  - Sets the foundation for the entire project.
  - Ensures alignment between stakeholders' expectations and the software solution.

- **Interconnect:** Provides a foundation for the design phase by outlining system functionalities and constraints.

2. Design Phase:

- **Importance:**
  - Translates requirements into a technical blueprint.
  - Guides developers during implementation.
- **Interconnect:** Guides the implementation phase by specifying system structure and components.

3. Implementation Phase:

- **Importance:**
  - Converts design into executable software.
  - Quality coding practices are crucial.

- **Interconnect:** Utilizes the design phase's output to build the software system.

4. Testing Phase:

- **Importance:**
  - Identifies defects and ensures functionality.
  - Improves software quality.

- **Interconnect:** Identifies defects and issues for rectification before deployment, ensuring software reliability.

5. Deployment Phase:

- **Importance:**
  - Ensures smooth transition from development to production.
  - Monitors performance and user feedback.
- **Interconnect:** Deploys the tested and approved software, completing the SDLC cycle.

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:**
   - Create, edit, and manage event details (dates, locations, categories).
   - Event promotion features (social media sharing, invitations).
2. **Attendee Management:**
   - Track RSVPs and manage attendee lists.
   - Communication with attendees via notifications/messages.
3. **User Experience:**
   - Intuitive interface with search and filtering options.
   - Support for different user roles and permissions.
4. **Additional Functionality:**
   - Integration with mapping services.
   - Feedback and rating system for events.

Documentation and Tracing:

Requirements Document:

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

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 concepts are essential to the development of the community event app since they direct our strategy and promote teamwork. The Agile principle of prioritizing "individuals and interactions over processes and tools" highlights the significance of proficient communication and collaborative efforts. For our project, this entails promoting candid communication among team members to facilitate the sharing of concepts, criticism, and understanding. We prioritize in-person contacts and frequent meetings to make sure that all viewpoints are heard and considered, which improves decision-making and problem-solving. This method also helps team members develop a feeling of accountability and ownership, which promotes a cooperative and encouraging atmosphere where everyone is inspired to give it their all.

In addition, The Agile principles of "customer collaboration over contract negotiation" and "working solutions over comprehensive documentation" are also essential to the creation of the community event app. We place more emphasis on providing real, workable solutions that meet our users' needs than we do on meticulous specifications and strict schedules. We can collect feedback early and often with this iterative method, which helps us to modify and improve the app depending on user feedback and real-world usage. Our app emphasizes delivering working software frequently. Rather than drowning in extensive documentation, we focus on tangible outcomes. Event organizers can create, manage, and track events directly within the app. Regular releases ensure that features are usable, tested, and aligned with user needs. By valuing functional software, we maintain momentum, gather feedback, and iterate based on real-world usage.

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:

- **Product Owner (PO):**
    - Represents stakeholders.
    - Defines product vision, backlog, and priorities.
    - Ensures value delivery.
- **Scrum Master:**
    - Facilitates Scrum events.
    - Removes impediments.
    - Coaches the team.
- **Development Team:**
    - Self-organizing.
    - Delivers increments.
    - Collaborates on tasks.

Artifacts:

- **Product Backlog:**
    - Prioritized list of features.
    - Owned by the PO.
    - Evolves throughout the project.
- **Sprint Backlog:**
    - Subset of Product Backlog.
    - Contains tasks for the Sprint.
    - Owned by the Development Team.
- **Increment:**
    - Deliverable product increment.
    - Created during each Sprint.
    - Must be potentially shippable.

Ceremonies:

- **Sprint Planning:**
    - PO and team plan the Sprint.
    - Define Sprint goal and select backlog items.
    - Outcome: Sprint Backlog.
- **Daily Stand-up (Daily Scrum):**
    - 15-minute daily meeting.
    - Team shares progress, plans, and blockers.

  - Focus on collaboration.
- **Sprint Review:**
  - PO presents Increment to stakeholders.
  - Feedback and adjustments.
  - Inspect and adapt.
- **Sprint Retrospective:**
  - Team reflects on the Sprint.
  - Identify improvements.
  - Continuous learning.

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.