

SDLC (Phone Pe)

Project Training – SDLC

18/06/2024

Areeb Hammad N

Version: 1.0

Created: 18/06/2024

Last Updated: 18/06/2024

Status: DRAFT (The status would change to finalized post the BA, PM and dev team review and sign off)

Software Development Life Cycle (SDLC) Overview for Phone Pe

1. Requirements Gathering:

Importance:

- Identify user needs and business requirements.
- Define functional and non-functional requirements.
- Stakeholder meetings and documentation.

Interconnection:

- Sets the foundation for all subsequent phases.
- Ensures the development aligns with business objectives.

2. Design:

Importance:

- Architectural design and UI/UX planning.
- Database design and API specifications.
- Create wireframes, mockups, and prototypes.

Interconnection:

- Translates requirements into a blueprint.
- Guides developers and testers on system structure.

3. Implementation:

Importance:

- Actual coding and software development.
- Integration of various components and modules.
- Continuous integration and version control.

Interconnection:

- Brings design to life.
- Regular updates based on testing feedback.

4. Testing:

Importance:

- Verification and validation of functionality.
- Identify and fix bugs and performance issues.
- Various testing types: unit, integration, system, and acceptance.

Interconnection:

- Ensures the application meets quality standards.
- Provides feedback to developers for improvements.

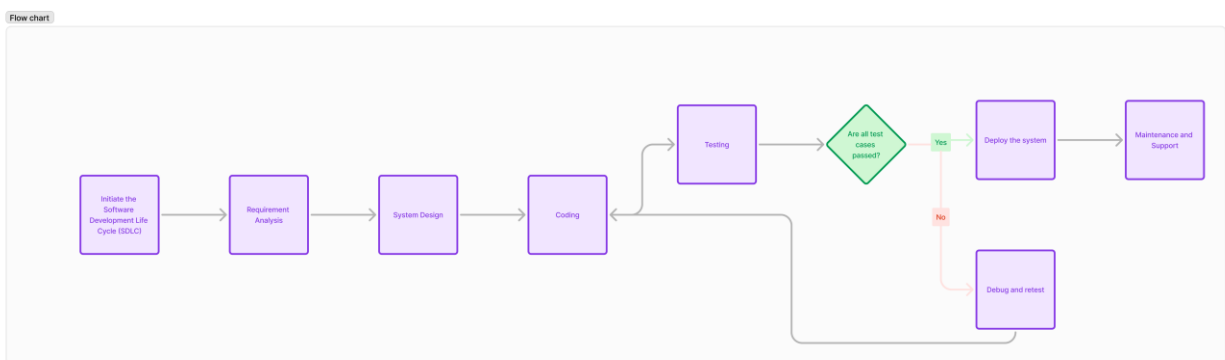
5. Deployment:

Importance:

- Release of the application to the production environment.
- Deployment planning and execution.
- Monitoring and support for initial issues.

Interconnection:

- Delivers the final product to users.
- Requires coordination with all teams for a smooth launch.



Mock Interview to Gather Requirements for a Community Event Organization App

Participants:

- **Interviewer:** App Development Team Member
- **Interviewee:** Community Event Organizer

Interview Agenda:

1. Introduction and Objective
2. Understanding the Community Events
3. Key Features and Functional Requirements
4. Non-Functional Requirements
5. User Roles and Permissions
6. Questions and Clarifications

Interview Transcript:

Interviewer: Thank you for joining us today. Could you start by telling us a bit about the community events you organize?

Interviewee: We organize various events like local festivals, charity runs, community clean-ups, and educational workshops.

Interviewer: That sounds great! What are the main challenges you face in organizing these events?

Interviewee: Coordination and communication are the biggest challenges. It's hard to keep track of volunteers, manage schedules, and ensure everyone is informed about updates.

Interviewer: I see. What specific features would you like to see in an app that helps organize these events?

Interviewee: We need features for event creation and scheduling, volunteer sign-up and management, communication tools, and a calendar view. It would also be helpful to have reminders and notifications, a way to share documents, and integration with social media.

Interviewer: Those are excellent points. Are there any non-functional requirements we should consider, such as performance or security?

Interviewee: The app should be user-friendly and work on both Android and iOS. Security is crucial since we'll handle personal information. It should also be scalable to handle multiple events simultaneously.

Interviewer: Understood. What different user roles do you foresee needing for this app?

Interviewee: There should be roles for event organizers, volunteers, and general participants. Organizers should have full access, volunteers need access to their specific tasks, and participants should see event details and sign-up options.

Interviewer: Perfect. Is there anything else you'd like to add or any questions you have?

Interviewee: Not now. Just that the app should be simple to use and reliable.

Interviewer: Thank you so much for your time and insights. This information will be very helpful for us.

Summary of Requirements:

Functional Requirements:

7. Event Creation and Management:

- Create, edit, and delete events.
- Set event schedules and locations.

8. Volunteer Management:

- Volunteer sign-up and task assignment.
- Track volunteer hours and contributions.

9. Communication Tools:

- Messaging system for organizers and volunteers.
- Broadcast notifications and reminders.

10. Calendar and Scheduling:

- Calendar view for events.
- Sync with personal calendars.

11. Document Sharing:

- Upload and share documents related to events.

12. Social Media Integration:

- Share events on social media platforms.

Non-Functional Requirements:

13. User-Friendly Interface:

- Intuitive navigation and design.

14. Multi-Platform Support:

- Available on Android and iOS.

15. Security:

- Protect user data with encryption.
- Secure login and authentication.

16. Scalability:

- Support multiple concurrent events.

17. Performance:

- Fast load times and smooth operation.

User Roles and Permissions:

18. Event Organizers:

- Full access to all features.

19. Volunteers:

- Access to assigned tasks and communication tools.

20. Participants:

- View event details and sign-up options.

Documentation and Traceability Plan:

1. Requirement Documentation:

- **Document:** "Community Event Organizer App Requirements Specification"
- **Sections:**
 - Introduction and Objectives
 - Functional Requirements
 - Non-Functional Requirements
 - User Roles and Permissions

2. Traceability Matrix:

- **Purpose:** Ensure each requirement is tracked through design, implementation, and testing.
- **Structure:**
 - **Requirement ID:** Unique identifier for each requirement.
 - **Description:** Summary of the requirement.
 - **Source:** Origin of the requirement (e.g., stakeholder interview).
 - **Design Reference:** Design document section that addresses the requirement.
 - **Implementation Reference:** Code module or component that implements the requirement.
 - **Test Cases:** Specific tests that verify the requirement.

Example Traceability Matrix:

Requirement ID	Description	Source	Design Reference	Implementation Reference	Test Cases
FR-01	Create, edit, and delete events	Stakeholder Interview	Section 3.1	Event Manager Module	TC-01, TC-02, TC-03
NFR-01	User-friendly interface	Stakeholder Interview	Section 4.1	UI/UX Design Document	TC-04, TC-05