Requirements

Software Requirements Specification (SRS)

The purpose of this SRS document is to provide a comprehensive outline of the functional and non-functional requirements for the Project Diary application. This application is designed to assist users in efficiently managing projects by tracking project activities, logging time, recording challenges, and analysing productivity. The SRS will serve as the primary guideline to support the development of this application and ensure it fulfils the specified requirements.

1. Functional Requirements

1.1. User Management

- Role-based Access: Users are assigned roles (e.g., Project Manager, Developer) to enable relevant access controls.
- User Authentication: Users must log in with unique credentials (username and password) to access their projects.
- Session Management: User sessions should be securely maintained until the user chooses to log out.

1.2. Project Management

- Create Project: Users can initiate a new project with information such as the project name, description, start date, and deadline.
- Edit Project: Users can modify details for an existing project.
- Delete Project: Users can delete a project once it’s completed or no longer relevant.
- Iteration Management: Each project can be divided into multiple iterations, allowing users to structure project phases and track progress incrementally.

1.3. Activity Tracking

- Create, Edit, and Delete Activities: Users can create, update, or delete activities within a project. Activities are organised hierarchically within categories (e.g., "Design > UI Design").
- Categorise Activities: Activity categories can be configured to facilitate efficient tracking and organisation.
- Assign Activities: Activities can be assigned to team members, ensuring accountability and clear task ownership.

1.4. Time Tracking

- Automatic Time Logging: Users can start a timer for automatic time logging on a specific activity.
- Manual Time Entry: Users can also manually log time if necessary.
- Pause/Resume Tracking: The timer can be paused and resumed to accommodate interruptions or breaks.
- Timer History: Users can view a history of their time logs for transparency and review.

1.5. Notes and Reflection

- Record Objectives, Challenges, and Solutions: Users can document personal objectives, challenges faced, and solutions applied for each assigned activity. This supports personal and project-wide reflection.
- Activity Evolution Matrix: Users can update the status of challenges (e.g., “pending,” “in progress,” “resolved”) and propose solutions to share with the team.

1.6. Data Analysis and Visualization

- Time Allocation Analysis: The application should support users in analysing the distribution of their time across different activities, categories, or iterations.
- Generate Visual Reports: Users can view reports with graphs (e.g., bar charts, pie charts) illustrating time spent per activity, category, or iteration.
- Custom Reports: Users can customise reports based on criteria like activity type or team member to derive specific insights.

1.7. Data Export

- Export Activity Data: Users can export logged activity data into various formats (e.g., CSV, PDF, Excel) for external reporting or documentation.
- Selective Export: Users can select particular activities or date ranges for export to create more tailored reports.

1.8. Team Collaboration (Additional)

- Activity Comments: Team members can leave comments on specific activities to enhance communication and support.
- Team Notifications: Notifications are sent when comments are added or solutions are proposed for an activity.
- Dashboard Customization: Users can personalise their dashboard by adding or removing widgets for quick access to critical metrics.

2. Non-Functional Requirements

2.1. Performance

- Response Time: The system should respond within 2 seconds for loading project and activity details.
- Efficient Time Tracking: Time tracking should occur in real-time, with minimal latency to ensure accurate time logs.

2.2. Usability

- User Interface: The application must have an intuitive UI, enabling users to easily navigate primary functions such as creating projects, logging activities, and viewing reports.
- Error Messages and Tooltips: Helpful tooltips and error messages should guide users in completing tasks without requiring extensive training.

2.3. Reliability

- Data Consistency: The system must reliably save all user-entered data, including time logs, activity details, and notes, to prevent data loss.
- Concurrency Handling: Multiple users should be able to interact with the system concurrently without performance issues.

2.4. Security

- Data Protection: User data (e.g., project details, time logs) should be securely stored with encryption to prevent unauthorised access.
- Authentication and Authorization: Secure protocols should be employed for user login and role-based access to ensure that users only access data they are permitted to view.

2.5. Compatibility

- Platform Compatibility: The application should be compatible with the specified JavaFX and MySQL versions, ensuring functionality on common operating systems (Windows, macOS, Linux).
- Browser Compatibility: When accessed via a browser, the UI must work effectively across major browsers.

2.6. Data Backup

- Automatic Backup: The application should support regular automatic data backups to secure user information.
- Data Recovery: Users should be able to recover deleted or lost data in case of accidental deletions.

3. Constraints

The following constraints apply to the development and deployment of the Project Diary application:

3.1. Technology Constraints:

- The project must use Java as the primary programming language and JavaFX for the user interface, based on lab software availability.
- MySQL will serve as the database system, although it is not available in the lab environment.

3.2. Resource Constraints:

- Development will need to be coordinated with access limitations to required tools, including MySQL setup in external environments.

3.3. Project Constraints:

- Core features as listed must be implemented as mandatory components of the application, with additional features only being included if they enhance functionality without affecting project scope or timeline.

3.4. Time Constraints:

- All functional and non-functional requirements must be completed by the project deadline, including initial setup, core functionality, and testing.

3.5. Performance Constraints:

- Given the scope, the system’s performance must maintain high responsiveness to support real-time time tracking without delays.