**Software Requirements Specification (SRS)**

**Project**: BetterThanJira - Project Management and Skill Tracking Web Application  
**Version**: 1.0  
**Last Updated**: 10/10/2024

**1. Introduction**

**1.1 Purpose**

The purpose of this SRS document is to outline the functional and non-functional requirements for the BetterThanJira project. The application is intended to help academic teams manage tasks and projects while tracking individual skill development over time.

**1.2 Scope**

BetterThanJira is a web-based platform aimed at improving collaboration and productivity in academic group projects. It includes features for task management, milestone tracking, skill tracking, and gamification to enhance user engagement. The application is designed for students and teams, with an intuitive user interface that works across multiple devices.

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **UI**: User Interface
* **UX**: User Experience
* **SQL**: Structured Query Language
* **MySQL**: A relational database management system
* **Kanban**: A project management framework visualizing work stages
* **Responsive Design**: A design approach that ensures the app adapts to different screen sizes and devices

**1.4 References**

* Angular Documentation
* MySQL Documentation
* Project Management Tools (e.g., Trello, Jira)

**2. Overall Description**

**2.1 Product Perspective**

BetterThanJira is a standalone web application built with Angular (front-end), JavaScript (back-end), and MySQL (database). It will provide a user-friendly interface for task management, skill tracking, and project milestones in an academic setting. The system will be accessible via desktop and mobile devices.

**2.2 Product Features**

1. **Task Management**:
   * Users can create, assign, and manage tasks for individual and group projects.
   * Drag-and-drop functionality for task movement between different project phases (To-do, In Progress, Blocked, Completed).
2. **Skill Tracking**:
   * Automatic and manual skill tracking based on project tasks and milestones.
   * Graphs to visualize skill progression.
   * Team members can endorse skills of others.
3. **Project Milestones**:
   * Users can define and track project milestones.
   * Deadline reminders and in-app notifications.
4. **Kanban Board**:
   * Visual representation of project tasks and workflow.
5. **Burnout Monitoring**:
   * Tracks user workload and time spent on tasks to prevent burnout.
6. **Gamification**:
   * Badges, achievements, and leaderboards to motivate users.

**2.3 User Characteristics**

The users of this platform include:

* **Students and Academic Teams**: Primary users who will manage tasks, track skills, and collaborate on group projects.
* **Instructors/Clients**: Users with limited access to view project progress and provide feedback.

**3. Functional Requirements**

**3.1 User Roles and Permissions**

* **Admin/Project Manager**:
  + Can create projects, assign tasks, and manage users.
  + Has access to all system features, including user management and skill endorsement.
* **Team Members**:
  + Can participate in projects, update task progress, and track personal skills.
  + Can endorse skills of other team members.
* **Viewer/Client**:
  + Limited to viewing project progress, milestones, and reports.
  + Cannot modify tasks or skills.

**3.2 Task Management Features**

* Ability to create individual and group projects.
* Assign tasks with deadlines and priorities.
* Drag-and-drop functionality on the Kanban board.
* Mark tasks as "In Progress", "Blocked", or "Completed."

**3.3 Skill Tracking Features**

* Auto-update skills upon completing tasks or milestones.
* Allow users to manually input and edit their learned skills.
* Track skill progression with visual graphs.
* Enable team members to endorse each other's skills.

**3.4 Notifications**

* Email or in-app notifications for project deadlines, new tasks, and task updates.
* Option for users to set notification preferences.

**3.5 Gamification**

* Award badges and achievements for skill development and project completion.
* Display leaderboards for top performers based on skill acquisition and task completion rates.

**3.6 Security**

* Implement role-based access control (RBAC) to ensure only authorized users can access specific features.
* Secure data storage with user authentication and encryption.

**4. Non-Functional Requirements**

**4.1 Performance**

* The application should handle up to 1000 concurrent users without performance degradation.
* Response times for user interactions (e.g., task updates, skill tracking) should be less than 2 seconds.

**4.2 Usability**

* The system should provide an intuitive interface for both technical and non-technical users.
* The user interface must be responsive and adapt to various devices (desktops, tablets, and smartphones).

**4.3 Reliability**

* The application must have 99.9% uptime, ensuring it is accessible for students and teams at all times.
* Automatic backups should be performed daily to prevent data loss.

**4.4 Security**

* User data, including project and skill information, must be encrypted at rest and in transit.
* The system must require strong user authentication.

**4.5 Scalability**

* The system must be scalable to accommodate future growth in the number of users and projects.

**5. System Models**

**5.1 Use Case Diagram**

**(To be added)**

**5.2 Database Schema**

**(To be added)**

**6. Appendices**

Any additional information, diagrams, or references required for system development.