

1. Introduction 1.1 Purpose The To-Do List web application aims to provide users with an intuitive and efficient platform for managing tasks and improving productivity.

1.2 Scope The application will include features such as task creation, editing, deletion, categorization, due dates, user authentication, and collaborative task management.

1.3 Stakeholders Stakeholders involved in the project include end-users, developers, project managers, and potential investors.

2. System Overview 2.1 System Architecture The To-Do List application will adopt a three-tier architecture, comprising a client-side interface, server-side logic, and a database for data storage. RESTful APIs will facilitate communication between the client and server.

2.2 Technologies Used Frontend: HTML5, CSS3, JavaScript (React.js for dynamic UI) Backend: Node.js with Express.js Database: MongoDB for flexible and scalable data storage Authentication: JSON Web Tokens (JWT) Version Control: Git for source code management 3. Functional Requirements 3.1 User Stories As a user, I can create a new task with a title, description, and due date. As a user, I can mark a task as completed or incomplete. As a user, I can categorize tasks into different lists or projects. As a user, I can set priority levels for tasks. As a user, I can collaborate on tasks with other users. 3.2 Use Cases Create Task User inputs task details and creates a new task. Update Task User modifies task details (title, description, due date, etc.). Delete Task User removes a task from the to-do list. Mark as Completed User marks a task as completed or incomplete. Categorize Task User assigns tasks to different lists or projects. 4. Non-Functional Requirements 4.1 Performance The system should have a response time of under 500 milliseconds for task updates. The application should handle up to 10,000 simultaneous users. 4.2 Security User authentication should be secure and use JWT for token-based authorization. Data should be encrypted during transmission and storage. 4.3 Usability The user interface should be intuitive, responsive, and accessible. The application should support multiple devices and browsers. 5. User Interface Design 5.1 Wireframes Include wireframes for key screens, such as the task list, task details, user authentication, and collaboration features.

5.2 Navigation Describe the navigation flow, including menus, buttons, and user interactions.

6. Database Design 6.1 Entity-Relationship Diagram (ERD) Provide an ERD illustrating the relationships between entities like users, tasks, categories, and priority levels.

6.2 Data Storage Specify the database schema and how data will be stored, retrieved, and updated.

7. System Components 7.1 Frontend Detail the frontend components, libraries, and frameworks.

7.2 Backend Specify the backend components, server architecture, and API endpoints.

7.3 Third-Party Integrations List and describe any third-party services or APIs to be integrated, such as authentication providers or notification services.

8. Testing Plan 8.1 Unit Testing Detail the strategy for unit testing individual components.

8.2 Integration Testing Explain how different components will be tested together.

8.3 User Acceptance Testing (UAT) Outline the plan for involving end-users in testing to validate the system.

9. Deployment Plan 9.1 Deployment Architecture Specify the deployment environment, including hosting services, server configurations, and scalability considerations.

9.2 Release Plan Provide a detailed timeline for releases, considering beta or pilot phases and any rollbacks in case of issues.

10. Maintenance and Support 10.1 Bug Tracking Describe the process for tracking and resolving bugs, including tools and workflows.

10.2 Updates and Enhancements Outline the procedures for rolling out updates, incorporating user feedback, and implementing new features.

11. Conclusion Summarize the key points, emphasizing the importance of delivering a secure, efficient, and user-friendly To-Do List web application.

This expanded DDS should serve as a comprehensive guide for developing and implementing a To-Do List web application. Adjustments can be made based on the specific needs and goals of your project.