Software Requirements Specification (SRS) Document. This is for online Task Management System. Made especially for public.

Software Requirements Specification

for

Task Management System

Version 1.4 approved

Prepared by Ali Ahasan

Business Automation Limited

26-09-2024

Software Requirements Specification (SRS) Document. This is for online Task Management System.

Contents

1. Introduction

This is the introduction section.

1.1 Purpose and Goals

The purpose of this Software Requirements Specification also known as SRS document which is to provide a detailed description. Consist of the functional and nonfunctional requirements for the development that is of an online Task Management System. Specifically for a small team. This document serves as an allusion. For all stakeholders involved in the project. Including developers, testers and users. Making it to ensure that the final product meets the required specifications. As well as the user expectations.

The primary goal of this project is to design a system that is user friendly. That facilitates effective task management, collaboration as well as productivity for small teams. This online Task Management System will allow team members which is to create, assign and track tasks in instant. Ensuring that all team members are informed of ongoing and upcoming work. The system will feature user roles, task categories, deadlines, notifications and reporting. This is just all designed to optimize workflow also improve efficiency.

Main objectives consist of:

- Task Creation and Assignment**:** Users have the ability to create tasks. As well as assign them to team members who hold particular roles.
- Task Categorization**:** Categorizing tasks helps to enhance clarity in workflow organization.
- Deadlines and Progress Tracking**:** Deadlines will be set for each task and users can keep track of task progress with status updates.
- User Roles and Permissions**:** Various levels of access and control over tasks will be available for different user roles. Including Admin, Manager, and Team Member.
- Collaboration Tools**:** The platform will offer communication including alert functions. This is just to improve teamwork within the group.
- Ease of Use**:** The system will be created with simplicity as the primary focus to guarantee. That it is easy for all team members to use, regardless of their technical skills.

This document will guarantee. That the requirements are explicit, thorough along with approved by all involved parties. This is to avoiding scope creep and serving as a foundation for system validation.

1.2 Scope

The Task Management System online will enable small team members to effectively coordinate and work together on tasks. Main components of the system which will consist of:

- ❖ User Account Management: Users have the ability to sign up, sign in that includes oversee their profiles. The system will provide support for various roles. Including team leader and team member, each having their own specific permissions.
- ❖ Task Creation and Assignment: Users have the ability to create tasks, assign priorities, and delegate them to themselves or teammates. Responsibilities will consist of detailing, sorting, and setting due dates.
- ❖ Task Tracking and Updates: The system will offer a dashboard. Making it to allow users to monitor the status of their tasks. As well as the tasks of their team members. Modifications that can be done to tasks. Such as updating status, appending comments and changing deadlines.
- ❖ Notifications and Alerts: Notifying team members immediately of task deadlines, assignments and changes. This is through automatic alerts. Just to keep everyone informed in instant.
- ❖ Reporting and Analytics: Generating reports using task completion rates, member activity and other metrics. This is to assist in enhancing team productivity. That is known as Reporting along with Analytics.
- ❖ Security: Guaranteeing data integrity including security. Which is through suitable authentication also authorization procedures.
- ❖ Integration: Possible upcoming collaborations with tools. Like calendars, email clients and project management software. This is just to improve functionality.

The system will be created to be user friendly. In order to motivate all team members to use the tool consistently. The main aim is to simplify the task management process. In order to boost productivity as well as teamwork among team members. The web-based Task Management System will guarantee accessibility on various devices. This is with the help of internet connection.

1.4 Overview

This document is divided into various main sections:

- ❖ Overall Description: This part offers a deep explanation of the overall aspects. That influence the product and its needs. It includes the viewpoint of the product, the requirements of the user including its limitations.
- ❖ System Features: This part explains the particular demands linked to each system feature. It consists of functional requirements, user interaction, and data management.
- ❖ External Interface Requirements: Details how the system connects with other software, interfaces for user interaction, and hardware interfaces.
- ❖ Non-functional Requirements: Nonfunctional Requirements refer to the characteristics. Like performance, security and reliability that are essential. For the system that has effectiveness. Though they do not directly perform specific functions.

- Other Requirements: Additional requirements that do not fall. This is under the above categories are outlined here.

Every part is focused on thoroughly. Defining the elements needed for the effective creation and implementation of the Task Management System. Guaranteeing that all stakeholder needs are satisfied. And in line with the overall business goals.

2. Software Development Life Cycle (SDLC) Steps

The SDLC is a systematic approach that directs software projects from start to finish including maintenance. The SDLC for the online Task Management System created for a small team consists of various important stages. All pf these are Planning, Requirements Analysis, Design, Implementation, Testing, Deployment and Maintenance. Here is a deep explanation of every step-in relation to this project.

2.1 Planning

Definition**:** The initial stage sets out the project's limitations, goals and practicality. It establishes clear goals and expectations. Making it serving as the basis for the entire development process.

Application to the Project**:**

- ❖ Project Scope Definition**:** Define the project scope by detailing the features and functions of the Task Management System. Including task creation, assignment, tracking, user roles, task categories, deadlines and collaboration tools.
- ❖ Feasibility Study**:** Have to conduct a feasibility study to evaluate technical, economic and operational feasibility. This is to verify that the project is achievable with the team that has resources, time limitations and technical skills.
- ❖ Resource Allocation**:** Resource Allocation involves identifying the necessary resources. To give examples it will be personnel (developers, designers, testers), technology (software, hardware) and budget.
- ❖ Risk Analysis**:** Risk analysis involves recognizing possible risks. Such as scope creep or technology restrictions. As well as creating plans to reduce or avoid them.
- ❖ Project Schedule**:** Develop a project timeline that includes milestones. Also, deadlines for every phase of the SDLC.
- ❖ Stakeholder Identification**:** Identifying stakeholders involves acknowledging all individuals or groups connected to or impacted by the project. That includes team members, project leaders and users.

2.2 Requirements Analysis

Definition**:** This stage includes collecting in depth data on the system requirements from stakeholders in order to guarantee the end product satisfies user requirements.

Application to the Project**:**

- ❖ Stakeholder Interviews and Surveys**:** Gather feedback from stakeholders through interviews and surveys to comprehend their requirements and anticipations.
- ❖ Functional Requirements Gathering**:** Record functionalities such as task allocation, progress reports, alerts including permission controls based on roles.

- Non-Functional Requirements**:** Nonfunctional Requirements involve specifying system performance criteria, security requirements, usability standards as well as scalability needs.
- Use Case Development**:** Generate use cases and user stories to demonstrate user interaction with the system.
- Requirements Documentation**:** Compile a document by outlining all gathered requirements. This is known as Software Requirements Specification also known as SRS for documentation purposes.
- Requirements Validation**:** Verify the accuracy and completeness of the SRS. Through reviewing it with stakeholders.

2.3 Design

Definition**:** The design stage converts requirements into a plan for building the software, specifying system structure, interface layouts also data models.

Application to the Project**:**

- System Architecture Design**:** System Architecture Design involves identifying the complete framework. Such as client-server setup, database layout as well as network needs.
- Database Schema Design**:** Design a relational database schema to store tasks, users, roles, categories including deadlines for effective data organization.
- User Interface (UI) Design**:** Design user interface also known as UI by producing wireframes and mockups concentrating on easy navigation and also user-friendly experience.
- Technology Stack Selection**:** Pick programming languages, frameworks. Like React for frontend, Node.js for backend. Also pick database systems. Like MySQL, MongoDB.
- Security Design**:** Strategize authentication methods, data encryption and secure communication protocols.
- API Design**:** API Design involves outlining the methods of communication among system components. That includes third party integrations.
- Design Documentation**:** Create comprehensive design papers and visuals. Like UML diagrams.

2.4 Implementation (Coding)

Definition**:** Developers turn design documents into code, create software components also merge them to produce a working system.

Application to the Project**:**

- Module Development: Module Development involves breaking down the system into different modules. Like user management, task management and notification system. After then delegating them to developers.
- Coding Standards: Set guidelines and optimal approaches for code quality upkeep.
- Version Control: Make use of tools such as Git for managing source code also working together.
- Continuous Integration: Continuous Integration involves setting up automated building as well as testing processes to identify problems at an early stage.
- Code Reviews: Peer reviews should be conducted for code. This is to guarantee that it meets quality standards.
- Integration: Integration involves merging modules. Just to ensure they function smoothly in unison.

2.5 Testing

Definition: Testing confirms the proper functionality of the software as well as ensures it complies with specified requirements. Also pinpointing any defects that require fixing.

Application to the Project:

- Unit Testing: Unit Testing involves checking the functionality of individual components.
- Integration Testing: Integration testing guarantees that modules work together as expected.
- System Testing: System Testing involves verifying the entire system according to the Software Requirements Specification also known as SRS.
- User Acceptance Testing (UAT): Enable real users to evaluate the system in a practical setting.
- Performance Testing: Performance Testing involves evaluating how a system responds and remains stable in different situations.
- Security Testing: Security testing involves discovering vulnerabilities also making sure that data remains protected.
- Bug Tracking and Resolution: Utilize tools like JIRA for bug tracking and resolving issues.

2.6 Deployment

Definition: The release phase includes deploying the software to the production environment. This is made for end users to access.

Application to the Project:

- Deployment Planning**:** Create a plan for implementing the system with little to no interference.
- Environment Setup**:** Setting up the environment involves preparing production servers and infrastructure to meet system requirements.
- Data Migration**:** Move any current data to the new system. That is if needed.
- Release Management**:** Release Management involves carrying out the deployment plan. Which may be done in phases such as a beta release.
- User Training**:** Offer training sessions or resources to assist users in adjusting to the new system.
- Documentation**:** Provide user manuals, installation guides as well as support documentation.

2.7 Maintenance

Definition**:** Maintenance includes continuous assistance and software upgrades. This is done after deployment to resolve issues and improve functionality.

Application to the Project**:**

- Bug Fixes**:** Always resolve any problems that occur during real world usage.
- System Updates**:** Always Install updates on the system for enhancements or security fixes.
- Feature Enhancements**:** Integrate new functions in response to user input or evolving needs.
- Performance Monitoring**:** All of these continuous monitoring of system performance as well as user satisfaction is essential.
- Technical Support**:** Offer help to individuals. Who are facing issues.
- Documentation Updates**:** Update documentation. This is to reflect system changes.

2.8 Review and Evaluation

Definition**:** This phase assesses the project that has outcomes against its objectives. This is to identify successes and areas for improvement.

Application to the Project**:**

- Project Evaluation**:** Always assess the project that has effectiveness based on time, cost, and quality.
- User Feedback Collection**:** Collect user feedback to assess satisfaction as well as the effectiveness of the system.

- Process Improvement**:** Evaluate successes and also failures to enhance upcoming projects.
- Lessons Learned Documentation**:** Document insights also recommendations for future use.

The development team guarantees that the Task Management System is well planned, tailored to user requirements, rigorously tested and sustained for lasting usability. This is done by carefully adhering to every SDLC phase. This organized method reduces dangers and improves the chances of project achievements. Offering a strong instrument for team cooperation and task administration.

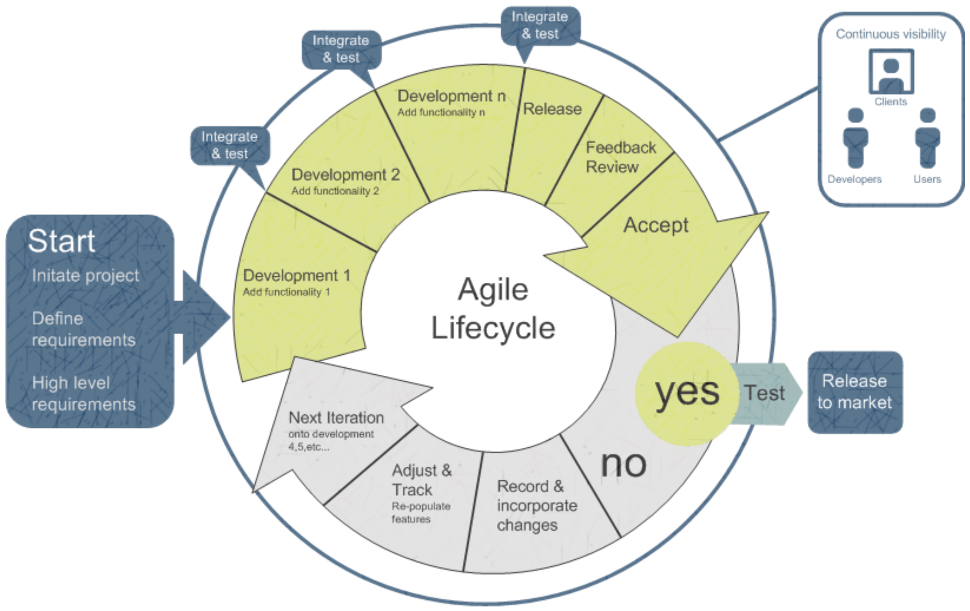
3. Development Model Selection and Application

This is the Development Model Selection and Application part.

3.1 Selection of the Development Model

The Agile methodology is suggested for the creation of the Online Task Management System. This choice is determined by many important elements. That correspond with the characteristics and requirements of the project:

- Flexibility in Requirements**:** Agile methodology permits adjustments in system requirements. As per user feedback and iterative testing. Crucial for dynamic applications like task management systems with evolving user needs.
- Incremental Development**:** Incremental Development involves developing the system in smaller increments also known as sprints. Making it to allow the team to focus on key functionalities such as task creation and assignment in initial sprints. And enhancing or including extra features like notifications and reporting in subsequent sprints.
- Continuous Improvement and Collaboration**:** Agile encourages consistent improvement and teamwork through frequent collaboration among diverse teams. Which is beneficial for a task management system aimed at boosting team cooperation.



3.2 Application of Agile Methodology

The Agile methodology can be divided into several key phases for this project. With keeping each phase in line with Agile principles:

i. Planning and Analysis

- Product Backlog Creation: Creating a product backlog involves gathering requirements from stakeholders. Initially and forming a list of desired features and functionalities. Which is organized by priority.
- Sprint Planning: Sprint Planning involves choosing important items. From the product backlog for delivery in the first sprint.

ii. Design

- Initial System Design: Begin by developing basic design diagrams and user interface mock ups. This is to illustrate the system architecture also how different components interact.
- Iterative Design Refinement: Iterative Design Refinement involves refining the system design. This is based on feedback from ongoing sprints. As well as testing results in an iterative process.

iii. Implementation

- Sprint Execution: Sprint Execution involves developing, testing and integrating functionalities chosen for each individual sprint. A usual sprint duration is 2 to 4 weeks.
- Daily Stand-ups: Hold daily meetings to go over progress, obstacles and next steps. Also holding daily meetings ensuring ongoing alignment and adaptation.

iv. Testing

- Continuous Testing: Maintain continuous testing throughout the sprints. By incorporating unit testing, integration testing also with user acceptance testing.
- Feedback Loop: Use feedback obtained during testing. To improve the product backlog and make adjustments to upcoming sprint plans.

v. Deployment

- Incremental Deployment: Incremental Deployment involves rolling out the system gradually. This allows users to start using it with essential functions. Also getting regular updates as more features are added.
- Continuous Integration and Deployment (CI/CD): Utilize CI/CD practices to automate deployments and streamline integration of new features.

vi. Review and Maintenance

- Sprint Reviews: Hold a review with stakeholders. This is to showcase new features and receive input. This is done after every sprint.
- Retrospectives and Continuous Improvement: Conduct a retrospective meeting following each sprint. Just to review successes, areas for improvement including ways to optimize workflow.

3.3 Advantages for the Project

Employing Agile for this task management system project provides numerous benefits:

- ❖ It guarantees that the final product. That meets the actual needs of users through incorporating iterative client feedback and testing.
- ❖ It improves team efficiency and productivity. By conducting regular evaluations as well as making necessary changes.
- ❖ Decreases risk with the help of creating in small stages and providing regular evaluation and adjustments as required.

The Task Management System that has development process will be adaptable including incremental also strongly connected to user needs by following Agile principles. Which is vital for producing a successful and user-friendly product.