Amanda Ryman

Task Management Software

Requirements Document

Table of Contents

1 Introduction

This document will outline the purpose, background, functionality and timeline of development for the Task Manager project.

1.1 Purpose and Scope

This document aims to provide a detailed description of the requirements for the Task Manager software, outlining the needs of the users as well as any other restrictions on the program. This document also serves as a proposal to the client so that project direction can be affirmed and tailored to the client's needs.

Task Manager is intended for use by the general public as a simple organizational tool. The native app will be available as a free download with possibility for development as a cross-platform mobile web app.

Users will manually add items to their "to do" list with the availability to flag their status and mark them as complete. Beyond the initial download, no further connectivity or input from other programs is necessary, although the addition of online accounts may be supported in future iterations of the project. For example, in future developments, a user will be able to sign into their Task Manager account from a public computer and be able to access their information using the web version of the software.

1.2 Target Audience

This document is provided as a reference document for the current and future development team, and for the client commissioning the development of this program.

2 Product Overview

This section will provide an overview of the complete system of the Task Manager program, including system context, users, and system constraints.

2.1 Users and stakeholders

The user profile for this program is extremely simple. There is only one kind of user, who serves also as an administrator with the power to create and delete tasks. The program will be able to support multiple users who each have separate private accounts and projects.

2.2 Use Cases

This section will outline the various use cases for the software.

2.2.1 Add a Task

The user adds a task to their "To Do" list.

2.2.2 Flag a Task
The user may flag a task as ‘to do’, ‘in progress’ or ‘completed’.

2.2.3 Delete a Task
Once the user no longer needs a task’s data, they may completely delete it.

3 Functional Requirements
This section provides a summary of all of the functional requirements of the Task Manager software -- that is, what the software specifically ought to do.

3.1 User Account Management
The software must be able to create and manage multiple user accounts.

3.1.1 User Account Creation
The software must be able to create and save new user accounts.

3.1.2 User Account Login
The software must be able to verify correct user credentials while rejecting incorrect credentials.

3.2 Task Management
The software must be able to accurately create and manage tasks for each user.

3.2.1 Task Creation
A user is able to use the software to create a new task. By default, new tasks are flagged as ‘to-do’. Tasks are created with fields such as ‘Description’, ‘Details’ and ‘Due Date’.

3.2.2 Task Editing
A user is able to edit all fields of the tasks they create.

3.2.3 Task Flagging
A user is able to change to tag of any task to ‘To-do’, ‘In Progress’ or ‘Completed’.

3.2.4 Task Deletion
A user is able to permanently delete a task.

4 Non-Functional Requirements

This section outlines the non-functional requirements of the Task manager -- that is, what qualities the software ought to have.

4.1 Session Integrity

The user and client expect a certain level of reliability and responsiveness.

4.1.1 Session Permanence

Work done by a user is recorded into a database at the end of their session to be accessed by them in their next session.

4.1.2 Session Interruption

The program should be able to handle unexpected aborts without losing user data.

4.1.3 Exception Handling

To avoid unintended program crashes, the software will anticipate and handle any potential exceptions (to be extensively tested in the testing phase).

4.2 Usability

For Task Manager to be viable software, it must accomplish its tasks in a user-friendly and clearly represented manner.

4.2.1 Simplicity

Task Manager is a simple program. The interface should be clean and efficient.

4.2.2 Self-Explanation

Users should be able to use Task Manager without any previous or external information. All usage should be clear and self-explanatory with all necessary prompts to guide the first time user through usage.

4.2.3 Task Viewing

Whether ‘To-do’, ‘In Progress’ or ‘Completed’, each task will be visible in its respective pane or in the ‘View All’ pane.

4.2.5 User Errors

The software ought to predict and make provisions for user errors. For example, if a required field is not filled in, prompt the user to fill it in. Create exception handling for buffer overflows.

5 Milestones and Deliverables

This section provides an overview of the work continuum and future efforts. Following this requirements document, a design document and test plan will be released, followed by the final deliverable, report and post-delivery maintenance.

5.1 Design Document

The design document will consist of a detailed description of the system and subsystem, complete with detailed diagrams, and will outline the full shape and form of the Task Manager.

5.2 Test Plan

A test plan will be executed to ensure that the Task Manager meets all requirements on both levels of individual units and the whole system. The test plan will detail what a passing case is as opposed to an exceptional one.

5.3 Final Deliverables

After testing, a final user-ready version of the Task Manager will be released.

Ryman