



Software Requirements Document

for

Consultant Todo List System

Version 1.1

Prepared by: *****

UAF CS471

Date

Table of Contents

1. Introduction.....3
 1.1 Preface3
 1.2 Purpose3
 1.3 Glossary.....3
 1.4 References.....3
2. General Description.....4
 2.1 User Requirements4
 2.2 Non-Functional Requirements.....5
3. System Requirements Specification.....5
 3.1 Functional Requirements.....5

1. Introduction

1.1 Preface

This software specification document is intended to be used by the software design team as an outline of requirements for the project. The document will also be used by the party commissioning the software development project. This requirements document may serve as the basis for a contract between the aforementioned parties.

1.2 Purpose

Our client is a consultant who must complete many disparate tasks during the course of a working day. The client must keep track of the time spent on each of these tasks for billing purposes. This new “todo” list system will combine the organization of tasks and the tracking of time usage into one comprehensive package. This system will be a custom design to fit the client’s unique needs and will be the primary tool that the consultant uses to manage their time during the working day.

1.3 Glossary

Should: implies that a requirement is desirable but that it is not required to be part of the final deliverable.

Must: implies that a requirement is desirable and that it is required to be part of the final deliverable.

Task: the basic element of the todo list. Each event to be scheduled should be a separate task.

Appointment: a sub type of task used for in the case of an event occurring at a specific time.

Todo List: reference to the projects collection of tasks ordered by priority and a short hand method of referring to the entire system under development.

Active: an active task is a task on which the user is currently working and it must be included on the list. FIX

Complete: A completed task is a task which has been marked “complete” by the user of the system and is no longer shown on the todo list but is archived for future reference.

Archive: a storage space for completed tasks.

Due Date: The exact time and date when a task must be finished (or in the case of an appointment task when the appointment begins).

1.4 References

Based on template information from *Software Engineering* (9th edition) by Sommerville

2. General Description

2.1 User Requirements

The following are the general user requirements for the project. The order of the list is not significant:

- 2.1.1. **The system must allow the user to create new tasks:** When a new task is created the system must obtain all the relevant information for that task from the user.
- 2.1.2. **The system must allow the user to manually edit any existing task:** The system must also make any changes that become necessary based on the result of the manual edit. An example of this is that changing the priority of one item will necessitate updating the priority of other items.
- 2.1.3. **The system must display all active tasks in a prioritized list:** The core of the todo list system is the interface where the user can view all active tasks ordered by their priority. The priority of the task will indicate the task's position on the list with the first item on the list (item number one) being the top priority. The higher the priority of the task the closer to the top of the list the item will be.
- 2.1.4. **The system must track and log the time that each task is being worked on by the user:** The time from when a user indicates that they are starting a task to the time that a user indicates that they have finished their work session must be tracked and then deducted from the time left until the task is complete.
- 2.1.5. **The system must change the priority of tasks marked as appointments automatically:** Some tasks, such as meetings, are time sensitive. This means that as a certain time approaches their priority should gradually become higher until the task is at the top of the list a user defined time before the task's due date. The system should be able to change the priority of a task marked as an appointment automatically.
- 2.1.6. **The system must allow the user change the status of a task from active to complete:** Once the user has finished a task or has determined that the task is no longer necessary the task must be removed from the active list. When a task is marked complete it is removed from the active list but is retained in an archive of tasks.
- 2.1.7. **The system must maintain an archive of completed tasks:** For the purpose of recording keeping, tasks must never be fully deleted but rather preserved in an archive even after they are marked as complete. Once a user marks a task as complete it will be removed from the list of active tasks but will still be accessible in the archive.

2.1.8. The system must allow the user to move any task to the top of the active list:
This functionality is independent of the manual task editing feature.

2.1.9. The system should support task prerequisites: If a task has a prerequisite the system should account for the prerequisite when displaying tasks and prevent the user from creating a state where the proper order of prerequisites is not maintained.

2.2 Non-Functional Requirements

At this time that client has not specified what platform this software should run on. There have also been no minimum hardware specifications outlined by the client. The client has not specified the type of interface (e.g. GUI or CLI) that the software should have.

3. System Requirements Specification

3.1 Functional Requirements

3.1.1. The system must allow the user to create new tasks:

- 3.1.1.1. The system must obtain new task priority
- 3.1.1.2. The system must obtain new task name
- 3.1.1.3. The system must obtain time to complete task
- 3.1.1.4. The system must obtain task due date
- 3.1.1.5. The system must obtain any task prerequisites
- 3.1.1.6. The system must obtain whether the task is an appointment
- 3.1.1.7. The system must obtain a description of the task
- 3.1.1.8. The system should obtain the name of the project associated with the task

3.1.2. The system must allow the user to manually edit any existing task:

- 3.1.2.1. The system must allow the user to edit any element of a task
- 3.1.2.2. The system must update itself based on any new information from the editing process which includes updating other tasks information if necessary

3.1.3. The system must display all active tasks in a prioritized list:

- 3.1.3.1. The system must display the task with the highest priority at the top of the list (item number one)
- 3.1.3.2. The system must display the rest of the items in the list by decreasing priority
- 3.1.3.3. The system should display basic task information for each task entry on the list

3.1.4. The system must track and log the time that each task is being worked on by the user:

- 3.1.4.1. The system must allow the user to indicate when they are starting and then stopping work on a task
- 3.1.4.2. The system must use the start and stop times indicated by the user to calculate the time worked on the project and then update the time left until the task is complete

3.1.5. The system must change the priority of tasks marked as appointments automatically:

- 3.1.5.1. If the task is designated as an appointment the system must automatically raise its priority as the due date nears
- 3.1.5.2. The task in question must be made top priority a user defined amount of time before its due date

3.1.6. The system must allow the user change the status of a task from active to complete:

- 3.1.6.1. The system must remove tasks marked as complete from the active list
- 3.1.6.2. The system must retain tasks marked as complete in an archive

3.1.7. The system must maintain an archive of completed tasks:

- 3.1.7.1. The system must archive all tasks marked as complete
- 3.1.7.2. The system should make completed tasks accessible for purposes like billing
- 3.1.7.3. The system should allow the user to make administrative changes to the archive

3.1.8. The system must allow the user to move any task to the top of the active list:

- 3.1.8.1. The system must update other tasks on the active list as necessary when this operation occurs
 - 3.1.8.2. This operation must be independent from the manual task editing system
- (3.1.2)

3.1.9. The system should support task prerequisites:

- 3.1.9.1. The system must inform the user that a task has a prerequisite task
- 3.1.9.2. The system must inform the user when changing a task's priority will move it in such a way that it is no longer in the proper order with its prerequisites