

Software Requirements Specification

for

TODO: Task Management System

CS 471

Draft 0.3.0

Prepared by:
Ethan Tullar
John Chiment
Blake Eggemeyer

November 11, 2011

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions	2
1.4	References	2
1.5	Revision tracking:	3
2	Overall description	3
2.1	Product functions	3
2.2	User Characteristics	3
2.3	Assumptions and Dependencies	3
3	Specific requirements	3
3.1	Functional requirements	3
3.1.1	Displayed data for tasks	3
3.1.2	Displayed data for appointments	4
3.1.3	Active List	4
3.1.4	Inactive List	5
3.1.5	Modification of fields	5
3.1.6	Interactions	5
3.1.7	Behavior without interaction	5
3.2	Non-functional requirements	6

1 Introduction

1.1 Purpose

This Software Requirements Specification for TODO is for use by the customer and development team. This document is a formal listing of the functional and non-functional requirements of the TODO system.

1.2 Scope

The TODO Task Management System will assist in the prioritization and alteration of a list of tasks which are to be completed by a given due date, the remaining estimated time on the project given a total time estimate, and the time spent on each task. This system will not keep track of work done on a listed task unless the user instructs TODO that work is being done.

1.3 Definitions

1. **Appointment:**
2. **Immediately:** Immediately refers to actions that will begin as soon as the user has given the input for the action to occur. This applies to the reordering of the list when new input is given. The action will take a measurable, non-zero amount of time.
3. **Should:** Requirements with this marker are desired, but not crucial, and will be a part of the final deliverable contingent on time and progress.
4. **Task:**
5. **TBD:** Acronym for To Be Determined. This is used in this document to signify that the information necessary for a part of this document is "To Be Determined".
6. **TODO:**
7. **User:** The person, or persons, who operate or interact directly with the product.
8. **Will:** Requirements with this marker are guaranteed to be in the final delivered product.

1.4 References

Written with the IEEE Recommended Practice for Software Requirements Specifications as a reference and guide. The Tsunami SWR and RPC Donor SWR were referenced to find appropriate wording for some sections.

1.5 Revision tracking:

0.1	Nov 2	Document constructed.
0.2	Nov 7	Edits made based on recommendations in first draft.
0.2.1	Nov 8	Begin incorporating Ethans work. Will upgrade to 0.3 when consensus is reached on merging.
0.2.2	Nov 9	Merge draft sent to team mates
0.2.3	Nov 11	Polishing
0.3.0	Nov 11	Merge completed, considered to be suitable for final draft

2 Overall description

2.1 Product functions

TODO is a new self-contained product, produced to aid the client in their consulting, with no planned extensions or dependencies.

1. A task manager that helps the user prioritize their work flow.
2. An appointment calendar which reminds the user of upcoming appointments and meetings.
3. A storage system for completed tasks
4. A time manager that tracks time spent on a task.

2.2 User Characteristics

TODO is intended to have a narrow user base with access to training material and documentation.

2.3 Assumptions and Dependencies

1. **Time information:** The software relies on time information from the computer it is running on. Inaccurate clock data will be reflected in the software output.
2. **Language:** The interface for the user is in English.
3. **Platform:** The platform has not been specified.

3 Specific requirements

3.1 Functional requirements

3.1.1 Displayed data for tasks

Tasks will have data associated with them described below.

3.1.1.1 Name: User defined label

3.1.1.2 Description: User defined field

3.1.1.3 Project: User defined field

3.1.1.4 Due date: User defined field

3.1.1.5 Time estimate: User defined field

3.1.1.6 Elapsed time: computed based on 'working on top item in list' function.

3.1.1.7 Priority: user controlled relative ranking.

3.1.1.8 Prerequisites: a task that must be completed or inactive before work may be done of this task.

3.1.2 Displayed data for appointments

Appointments will have data associated with them described below.

3.1.2.1 Name: User defined field

3.1.2.2 Description: User defined field

3.1.2.3 Project: User defined field

3.1.2.4 Date and time: User defined field

3.1.2.5 Planned duration: User defined field

3.1.2.6 Actual duration: computed based on 'working on top item in list' functionality.

3.1.2.7 Priority: calculated to change the appointment position in the list.

3.1.3 Active List

The active list is a set of the tasks and appointments that are not yet been marked as completed, or inactive.

3.1.4 Inactive List

The inactive list is a set of tasks and appointments that have been marked as completed, inactive, or ignored.

3.1.5 Modification of fields

3.1.5.1 Modify any user created field: User will be able to modify the contents of any field the user created in a task, project, or appointment. Any modifications made will be reflected in the list immediately after the alteration has been made.

3.1.5.2 Make top priority: The user will be able to make a task the top priority and see the list order change immediately after the alteration has been made. This action will be done through a short-cut rather than the normal task modification method.

3.1.6 Interactions

Listed below are methods for the user to interact with the tasks.

3.1.6.1 Working on top item in list: When indicated by the user, the 'time worked' field of the task will accumulate. The 'estimated time to completion' will correspondingly be reduced.

3.1.6.2 Mark inactive: When indicated by the user the task will be removed from view in the active list.

3.1.6.3 Completed: When indicated by the user the task will be removed from view in the active list.

3.1.6.4 Done for now: When indicated by the user 'time worked' will stop accumulating.

3.1.6.5 Changing list order: The user will be able to change the order of the tasks in the active list, but the software will not allow a task to be worked on before its prerequisites have been completed.

3.1.7 Behavior without interaction

Listed below are actions that will take place without interaction from the user.

3.1.7.1 Appointments: Appointments will move up in the to do list as the appointment time get nearer. The appointment will be at the top of the list when the system time and date matches the reminder time and date.

3.2 Non-functional requirements

3.2.0.1 Update frequency: The active and inactive lists will update in a manner sufficient to capture any changes to it.

3.2.0.2 Number of entries in a list: The active and inactive lists will not be limited in their number of tasks or appointments in a manner which adversely affects the user.

3.2.0.3 Inactive list viewing: The inactive list will make data available to the user in a manner which will allow them to determine how much time was spend working on separate tasks and appointments.

3.2.0.4 Storage: Tasks and appointment will be saved regardless of their completion status.