

# **TODO: Task Management System**

## **SOFTWARE REQUIREMENTS SPECIFICATION**

*Author:*  
Blake Eggemeyer

November 7, 2011

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Scope . . . . .	2
1.3	Definitions . . . . .	2
1.4	References . . . . .	2
<b>2</b>	<b>Overall description</b>	<b>3</b>
2.1	Product perspective . . . . .	3
2.2	User Characteristics . . . . .	3
2.3	Assumptions and Dependencies . . . . .	3
<b>3</b>	<b>Specific requirements</b>	<b>3</b>
3.1	External interface requirements . . . . .	3
3.1.1	User interface . . . . .	3
3.2	Functional requirements . . . . .	3
3.2.1	Stored data for tasks . . . . .	3
3.2.2	Stored data for appointments . . . . .	4
3.2.3	Modification of fields . . . . .	4
3.2.4	Interaction . . . . .	4
3.2.5	Behavior without interaction . . . . .	5
3.3	Non-functional requirements . . . . .	5

# 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. **user:** The person, or persons, who operate or interact directly with the product.
2. **will:** Requirements with this marker are guaranteed to be in the final delivered product.
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. **immediately:** Immediately refers to actions that will begin as soon as the user has given the input for the action to occur. This - computed based on 'working on top item in list' function. This applies to the reordering of the list when new input is given. The action will take a measurable, non-zero amount of time.
5. **active list:** The active list is a set of tasks and appointments that have not yet been marked as completed, or inactive.
6. **inactive list:** The inactive list is a set of tasks and appointments that have been marked as completed, or inactive.

## 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.

## 2 Overall description

### 2.1 Product perspective

TODO is a new self contained product with no planned extensions or dependencies.

### 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 External interface requirements

#### 3.1.1 User interface

The user interface has not been specified.

### 3.2 Functional requirements

#### 3.2.1 Stored data for tasks

Tasks will have data associated with them described below.

1. **Name:**
2. **Description:**
3. **Project:**
4. **Due date:**
5. **Time estimate:**
6. **Elapsed time:** computed based on 'working on top item in list' function.
7. **Priority:** number between 1 and 5 inclusive.

8. **Prerequisites:** a task that must be completed or inactive before work may be done of this task.
9. **Color:** color used to mark tasks in a manner to be determined in the design phase.

### 3.2.2 Stored data for appointments

Appointments will have data associated with them described below.

1. **Name:**
2. **Description:**
3. **Project:**
4. **Date and time:**
5. **Planned duration:**
6. **Actual duration:** computed based on 'working on top item in list' function.
7. **Priority:** calculated.
8. **Color:** color used to mark tasks in a manner to be determined in the design phase.

### 3.2.3 Modification of fields

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.
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.

### 3.2.4 Interaction

Below are listed interactions that will allow the user to interact with the tasks.

1. **Working on top item in list:** When activated the 'time worked' will begin accumulating. The 'estimated time to completion' will begin to reduce.
2. **Mark inactive:** When activated the task will be removed from view in the active list.
3. **Completed:** When activated the task will be removed from view in the active list.

4. **Done for now:** When activated 'time worked' will stop accumulating.
5. **Changing list order:** The software will not allow a task to be worked on before its prerequisites have been completed.

### 3.2.5 Behavior without interaction

Below are listed actions that will take place without interaction from the user.

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.

## 3.3 Non-functional requirements

in a manner sufficient to capture any changes to it

**Version tracking:**

- 0.1 Nov 2 Document constructed.
- 0.2 Nov 7 Edits made based on recommendations in first draft.