

# Software Requirements Specification

Hunter Dansin

## Contents

<b>1. Introduction</b>	<b>2</b>
1.1 Purpose . . . . .	2
1.2 Document Conventions . . . . .	2
1.3 Project Scope . . . . .	2
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective . . . . .	2
2.2 User Class and Characteristics . . . . .	3
2.3 Operating Environment . . . . .	3
<b>3. System Features</b>	<b>3</b>
3.1 Description and Priority . . . . .	3
3.2 Stimulus Response Sequences . . . . .	3
<b>4. External Interface Requirements</b>	<b>4</b>
4.1 User Interface . . . . .	4
4.2 Hardware Interface . . . . .	4
4.3 Software Interface . . . . .	4
4.4 Communication Interface . . . . .	4
<b>5. Other Requirements</b>	<b>4</b>
5.1 Performance Requirements . . . . .	4
5.2 Security Requirements . . . . .	5
5.3 Software Quality Attributes . . . . .	5

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to build a todo tracker that provides an easy to use, secure, and performant system for managing personal and/or professional tasks.

## 1.2 Document Conventions

This document uses the following conventions:

Abbreviation	Term
DB	Database
UI	User Interface
OS	Operating System

## 1.3 Project Scope

The purpose of the todo tracker is to provide professionals and everyday consumers with an easy to use application for managing complicated tasks. The initial prototype will be available as a web app.

# 2. Overall Description

## 2.1 Product Perspective

A good todo system tracks the following:

- The associated task (input by user).
- Information relevant to the task (input by user, optional):
  - A more detailed description of the task.
  - Relevant information/items needed or helpful for completing the task (e.g. shopping list, contact).
  - A location where the task will be performed.
  - A date the task needs to be completed by.
- Task metadata (recorded by application):
  - Date when the task was created.
  - Date when the task was completed.

The goal is not to force the user to follow a specific todo formula, but to provide a flexible tool that will fit their workflow and habits. However, the app may provide resources for

users who are looking to improve their methods.

## **2.2 User Class and Characteristics**

Users should be able to login to their account and manage their task list. Specifically they should be able to:

- Login to their account.
- Create a new task.
  - Add description.
  - Add location.
  - Add due date.
- View tasks.
  - Mark tasks as done.
  - Delete tasks.
  - Edit tasks.

## **2.3 Operating Environment**

The operating environment is as listed:

- Storage via MongoDB.
- Authentication with Auth0.
- Platform: NodeJS, Javascript, Express.
- OS agnostic (any modern web browser).

# **3. System Features**

## **3.1 Description and Priority**

The todo tracker maintains a dynamic list of a user's tasks. It is a high priority because loss of access to the user's task list result in loss of productivity.

## **3.2 Stimulus Response Sequences**

- Create a new task.
- Display updated list of uncompleted tasks.
- Mark task complete.

## 4. External Interface Requirements

### 4.1 User Interface

- Front-end: EJS/HTML.
- Back-end: Express, MongoDB/mongoose, Auth0.

### 4.2 Hardware Interface

- Windows, Linux, Mac.
- A browser that supports HTML & Javascript.

### 4.3 Software Interface

Software used	Description
OS	The application will be designed to run in any modern browser, including mobile. This will allow it to support as many users as possible.
DB	To save users and their lists, I have chosen MongoDB for it's stability and performance.
EJS	I chose EJS because it is close to HTML5, and adds useful functionality.
SASS	I chose SASS because it makes theming with CSS easier.

### 4.4 Communication Interface

This application will support all modern browsers. It uses simple HTML5 forms for submitting and updating tasks.

## 5. Other Requirements

### 5.1 Performance Requirements

The application should be designed to be light, and deliver the user the information they need as fast as possible. The database will be designed with performance in mind.

## 5.2 Security Requirements

The application will use Auth0 to keep users' task lists secure. It will also encourage the creation of strong passwords and best practices for data security.

## 5.3 Software Quality Attributes

- **Availability:** The system should be available 24/7 so that users do not lose access to their task list.
- **Scalability:** The application should be designed in such a way that new features can be implemented in the future.
- **Usability:** The application should satisfy as many user needs as possible.