



Software Requirements Specification

for

The Todo List

Version 2.0

Prepared by

Group Name: Ratpack

Kevin Zavadlov

11569849

Kevin.zavadlov@wsu.edu

Riley Barnes

11566375

riley.g.barnes@wsu.edu

Date:

12/12/2020

| | |
|--|-----|
| Revisions..... | iii |
| 1 | |
| Introduction..... | 1 |
| 1.1 Document Purpose..... | 1 |
| 1.2 Product Scope..... | 1 |
| 1.3 Intended Audience and Document Overview..... | 1 |
| 1.4 Definitions, Acronyms and Abbreviations..... | 1 |
| 1.5 Document Conventions..... | 1 |
| 1.6 References and Acknowledgments..... | 2 |
| 2 Overall Description..... | 3 |
| 2.1 Product Perspective..... | 3 |

| | | |
|-------|---------------------------|---|
| 2.2 | Product | |
| | Functionality..... | |
| .. | 3 | |
| 2.3 | Users and | |
| | Characteristics..... | 3 |
| 2.4 | Operating | |
| | Environment..... | |
| 3 | | |
| 2.5 | Design and Implementation | |
| | Constraints..... | 4 |
| 2.6 | User | |
| | Documentation..... | |
| | 4 | |
| 2.7 | Assumptions and | |
| | Dependencies..... | 4 |
| 3 | Specific | |
| | Requirements..... | |
| | 5 | |
| 3.1 | External Interface | |
| | Requirements..... | 5 |
| 3.2 | Functional | |
| | Requirements..... | |
| 6 | | |
| 3.3 | Behavior | |
| | Requirements..... | |
| | 6 | |
| 4 | Other Non-functional | |
| | Requirements..... | 7 |
| 4.1 | Performance | |
| | Requirements..... | 7 |
| 4.2 | Safety and Security | |
| | Requirements..... | 7 |

| | | |
|--------------------|------------------|---|
| 4.3 | Software Quality | |
| Attributes..... | | 7 |
| 5 | Other | |
| Requirements..... | | |
| | | 8 |
| Appendix A – Data | | |
| Dictionary..... | | 9 |
| Appendix B - Group | | |
| Log..... | | |
| 10 | | |

| Revisions | | | |
|-------------|--------------------------------|---|----------------|
| Version | Primary Author(s) | Description of Version | Date Completed |
| Version 1.0 | Kevin Zavadlov Riley Barnes | Initial version of document before any code was written. | 11/05/2020 |
| Version 2.0 | Kevin Zavadlov Riley Barnes | Final revision of document after the program was written. | 12/12/2020 |

1 Introduction

Our project is a to-do list where users can add, complete, edit, and delete tasks. There is a point and streak system. This section describes what the whole document's purpose is and gives an overview of the project.

1.1 Document Purpose

The product that this SRS is describing is the "Todo List" Version 2.0. The SRS will describe the total functionality of the "Todo List" program, its requirements, any constraints, who will use it, and its expected use.

1.2 Product Scope

The software described is the Todo List program. The main feature of the program will be adding and removing tasks. The user will be able to modify the tasks on the TodoList such as setting the date (differentiating between due soon and overdue tasks), name, and priority of the task. Also, the user will be able to add subtasks one level deep. The other benefit of the software is the point system. The user will gain points when they complete tasks and gain more points if they are on a streak. Setting due dates (differentiating between due soon and overdue tasks), subtasks, and task priority. Also, the program will save data to local PC storage.

1.3 Intended Audience and Document Overview

This document is meant for the person grading the project and the basic user that will use the product's functionality to benefit themselves. This SRS is split into 4 major sections: introduction, overall description, specific requirements, and nonfunctional requirements. The introduction gives a basic overview of this document and the product. The overall description will go into the total functionality and constraints of the product and its environment requirements. The specific requirements section will explain the interfaces the product will use to interact with the client and other systems. The specific requirements section will also explain all functions of the product and the expected interactions between users and the product. The nonfunctional requirements section will explain requirements not directly related to how the product should function such as performance, safety and security, and software quality attributes.

1.4 Definitions, Acronyms and Abbreviations

Any use of "the software", "the program", or "the product" refers to the Todo List V2.0.

HTML = Hypertext markup language

SRS = software requirement specification.

PC = personal Computer

1.5 Document Conventions

This document uses Arial font size 11 with single spacing and 1" margins for body text.

Sections use heading 1 and subsections use heading 2.

1.6 References and Acknowledgments

V2.0 of the SRS has no references or acknowledgments.

2 Overall Description

2.1 Product Perspective

This program is a new standalone program that does not extend the functionality of another program. The program will interact with the user through the browser. The program will save data to the users pc through HTML5 local storage.

2.2 Product Functionality

1. Add tasks
2. Remove tasks
3. Create subtasks
4. Complete tasks
5. Edit tasks
6. set task due dates
7. Mark tasks if they are overdue
8. Set a task's priority
9. Display the point total
10. Give points for completing tasks
11. Streaks for completing multiple tasks
12. Reset streak after completing an overdue task
13. Change the username
14. Change the font
15. Change the font color
16. Change the background color
17. Change the amount of tasks displayed on the homepage.
18. Cheat page to change the amount of points and streak

2.3 Users and Characteristics

The program is intended to run in a users browser. The program would start as a blank list each time a user opened it they would need to set their username and load their List from their local storage if they previously saved it. If the program was “deployed” users would be able to access the program online through a webpage.

Super User/Tester/Developer/Grader: knows the “cheat” password to unlock the cheat page that allows them to increase and decrease their points and streak.

Normal user: is only able to use the normal functionality of the product. Uses the product many times.

2.4 Operating Environment

The software should work in windows 10 version 19041 and above. The software should run on any modern processor. The software will run for sure in firefox and chrome..

2.5 Design and Implementation Constraints

Space on user's computer for local storage for tasks and user information, possible compatibility issues with operating systems.

The maximum number of points is $2^{53} - 1$ as that is the JS max integer value. The furthest date possible is not an impactful constraint as it is past the year 100,000.

2.6 User Documentation

There will be a brief tutorial page on the website that explains how to use the product.

2.7 Assumptions and Dependencies

The pc has enough space to save the data.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The software will have a home page, tutorial page, settings page, task page, and a hidden cheat page.

3.1.2 Hardware Interfaces

The product uses the semantic UI library. The program is interacted with through the mouse and keyboard.

3.1.3 Software Interfaces

The product will interact with the operating system through the browser it is run in.

3.1.4 Communications Interfaces

The data will be stored to the local pc through HTML5 local storage.

3.2 Functional Requirements

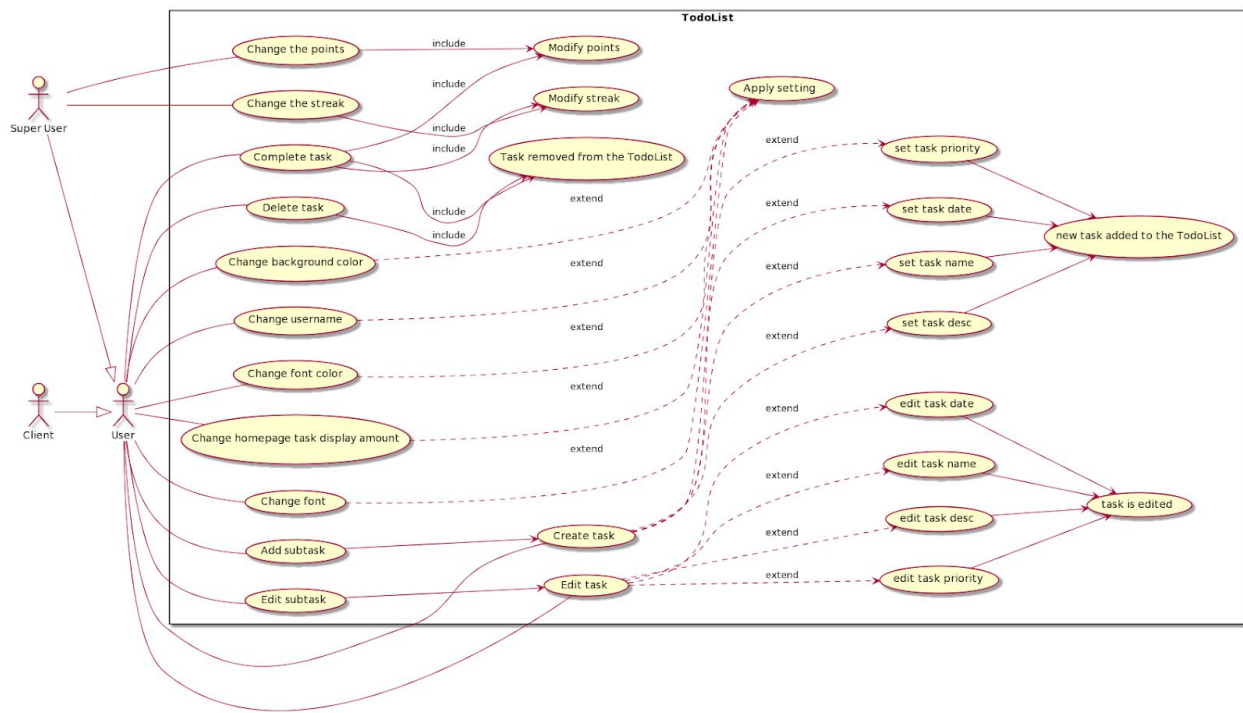
1. Actions to do to tasks
 - a. Add a task
 - i. Add a task with all task properties (name, date, priority) to the TodoList
 - b. Remove a task
 - i. Delete a task from the TodoList
 - c. Add a subtask
 - i. Add a new task to the TodoList that is a subtask to another task
2. Editing tasks
 - a. The user will be able to name and rename tasks
 - b. The user will be able to set and change the date of a task
 - c. The user will be able to set and change a tasks priority
 - d. The user will be able to set and change the description of a task
3. Point system
 - a. The system will display the current points the user has accrued
 - b. When the user completes a task they will gain points
 - c. When a task is overdue the user will lose their streak
 - d. If multiple tasks are completed in a row the user will gain a "streak" that gives more points per completed task.
4. Settings
 - a. The user will be able to set their username
 - b. The user will be able to select different fonts for text.
 - c. The user will be able to select different font colors.
 - d. The user will be able to select different background colors
 - e. The user will be able to select the amount of tasks displayed on the homepage

5. Cheat page

- The cheat page is only revealed when a user inputs the correct password
- The cheat page allows the user to modify their points
- The cheat page allows the user to modify their streak

3.3 Behavior Requirements

3.3.1 Use Case View



4 Other Non-functional Requirements

4.1 Performance Requirements

Program should be well optimized provided the user internet connection is adequate. Modifying any information such as editing tasks, adding tasks, or editing settings will at most

take seconds and worst case less than a minute. This performance is guaranteed up to 100 total tasks and subtasks.

4.2 Safety and Security Requirements

- Potential loss of data stored in local save if it is tampered with.
- Basic level of security as most items are local

4.3 Software Quality Attributes

4.3.1 Adaptability

The software can be used by many different users on the same pc/browser because each would have their own unique username and would have different saves.

4.3.2 Availability

The software will be available to anyone who has the link to the website and whose pc meets the operating environment requirements.

4.3.3 Correctness

The software shall not modify user input if the input is in the correct format.

4.3.4 Flexibility

The program state will be stored as a JSON file so other programs could use said JSON file

4.3.5 Interoperability

The software could be configured to store to a database instead of local storage by modifying the save and load functions

4.3.6 Maintainability

The software will be coded in such a way that similar functions are grouped. This will make it easier to modify and debug.

4.3.7 Portability

The software will be able to run in modern browsers.

4.3.8 Reliability

The software will create the same task when given the same input.

4.3.9 Reusability

The software will be able to save its current state to the local pc and retrieve this state when asked.

4.3.10 Robustness

The software will be able to handle edge cases without a crash of the program.

4.3.11 Testability

There will be a super user account that is able to change variables and to create predetermined tasks to test multiple program functions.

4.3.12 Usability

There will be a tutorial page to show a user how to use the program. All tasks will be able to be viewed, changed, or deleted.

Appendix A – Data Dictionary

| | |
|-----------------------|---|
| Encryption | Transforms the data so it is unreadable without a key. |
| pc | Personal computer refers to either a desktop or laptop. |
| html5 | Hypertext markup language the language that determines how web pages are formatted. |
| Local storage | The storage space on the pc you're currently using. |
| menu | Home page of the program which has every possible page like settings and page. |
| settings | A page that allows a user to change variables of a program i.e. username, homepage color, window size, graphics setting, etc. |
| libraries | A pack of programs developed by someone else. |
| srs | Software requirements specification |
| Operating environment | The set of programs that allow your program to run including OS, browser, plugins, etc. |

| | |
|------------|---|
| developer | Person who wrote tested and published a program. |
| Super user | An administrator user account that has access to the program and uses but has more permissions than a regular user. |
| user | One who uses the program. |

Appendix B - Group Log

10/13/20: Setup github with initial files. (Kevin)

10/18/20: Initial writing of sections 1.1 . (Kevin)

10/24/20: worked on 1.2 and 1.3. (Kevin)

10/27/20: finished all sections except 4.3. (Kevin, Riley)

11/05/20: finished srs for first submission (Kevin, Riley)

12/12/2020: edited srs for final submission (Kevin, Riley)