Software Requirements Specification

For

Cloud Task Managing Application

Version 1.0

Charlie Marshall

Shelley Murphy

Wesley Rupert

September 2013

Revision History

# 1. Introduction

## 1.1 Purpose

This SRS describes the functional requirements for release 1.0 of the Cloud Task Managing System. The project team will use this to implement the system and verify the desired functionality is in place. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

## 1.2 Project Scope and Product Features

The Cloud Task Managing System will allow users to manage lists of tasks that they can access on all their devices through the cloud. A more in depth description of the system can be found in the *Cloud Task Management System Vision Scope Document* [1]. Said document describes the features that are scheduled for implementation in this release.

## 1.3 References

1. Marshall, Charlie and Murphy, Shelley and Rupert, Wesley. *Vision and Scope for Cloud Task Managing Application.*

# 2. Overall Description

## 2.1 Product Perspective

The Cloud Task Management System is a new system that enables students and other members of the Case Western Reserve University community to create and manage shared, cross-platform task lists and items. This section illustrates the system interfaces and user relationships for release 1.0. The system is expected to evolve over several releases, ultimately connecting with existing social networks and handling externally-received system and task notifications.

## 2.2 User Classes and Characteristics

Visitor: a user of the product that is not logged in. There can be an arbitrary number of visitors, as there is no server data allocated to them until they log in or create an account and become a User.

User: a logged in user of the product. The number of Users is unknown and dependent on adoption of the product, but is expected to be a few thousand total, with up to 400 Users online at peak times.

## 2.3 Operating Environment

OE-1: The Cloud Task Management System shall operate in the following Web browsers: Internet Explorer 9 and up, Google Chrome 29 and up, Mozilla Firefox 24 and up, Apple Safari 6,  iOS Safari 3.2.2, Android browser 2.3.3, Windows Phone browser 10.0.

OE-2: The Cloud Task Management System shall operate on a server running the current versions of Fedora and Apache WebServer.

OE-3: The Cloud Task Management System shall permit users access from any part of the Internet.

## 2.4 Design and Implementation Constraints

CO-1: The system’s design, code, and maintenance documentation shall be provided in a consistent manner.

CO-2: The system shall use the provided mySQL database engine provided in current Django releases.

CO-3: All HTML code shall conform to the HTML5 standard

CO-4: All scripts shall be written in Python Django

## 2.5 User Documentation

UD-1: The website will give users basic direction to create lists, order lists, add tasks, check off tasks, delete tasks, and delete lists.

UD-2: When users access the site as a non-registered user, they will be directed to either create an account, sign in, or click to acknowledge they understand their data will only be accessible on the web browser they are currently using.

UD-3: Account creation will direct users to fill in necessary fields required to form an account in the database.

## 2.6 Assumptions and Dependencies

AS-1: The user has an internet connection

AS-2: The user has a reasonably current internet browser   
AS-3: If the user doesn’t have an account, they must have cookies

DE-1: GSM and SMTP support for notification stretch g

# 3. System Features

## 3.1 Application keeps track of users

### 3.1.1 Description and Priority

A user account is necessary to deliver a cross-platform experience. Without an account, the user cannot access any part of the product. This is High priority.

### 3.1.2 Stimulus/Response Sequences

Stimulus: A user visits the product not logged in.

Response: The system acknowledges the user and prompts to create an account or log in.

Stimulus: A user creates a new account.

Response: The system creates and saves a new user in the database. The system then logs the user into this account.

Stimulus: A user logs into an existing account.

Response: The system loads the user’s account and information and delivers it to the user.

Stimulus: A user views his home page.

Response: The system delivers the user’s lists and upcoming due tasks.

### 3.1.3 Functional Requirements

|  |  |
| --- | --- |
| **Function** | **Description** |
| User.CreateNew | The System shall allow a Visitor to create a new account to use on the System. |
| User.LogIn | The System shall allow a Visitor to identify themselves as a User and gain access to their information and task lists. |
| User.View | The System shall retrieve the User’s information and display it to the User. |
| User.Edit | The System shall ask for changes to the User’s information and update it. |
| User.LogOff | The System shall log the User off from the device. |

## 3.2 Application keeps track of user’s tasks in a given task list

### 3.2.1 Description and Priority

A user may see all task lists they have access to given that they are signed into their account. A user may check a task off as completed, as well as add new tasks. This is High priority.

### 3.2.2 Stimulus/Response Sequences

Stimulus: A user marks a task as completed.

Response: System acknowledges the user and indicates the task is complete by putting a checkmark beside it.

Stimulus: A user marks a previously indicated complete task as still in progress

Response: The system removes the check mark from beside the task.

Stimulus: A user adds a new task to a task list.

Response: The system will query the user as to the specifics of the task.

Stimulus: A user saves a task to a task list.

Response: The system will add the task to the list of tasks already written

### 3.2.3 Functional Requirements

|  |  |
| --- | --- |
| **Function** | **Description** |
| Task.Create | The system shall let a Patron create a new task given the user has read/write permissions. |
| Task.View | The system shall let a Patron view an already-created task. |
| Task.Complete | The system shall let a Patron mark a task as completed. |
| Task.Incomplete | The system shall let a Patron mark a task as not yet completed |
| Task.Delete | The system shall allow a Patron to delete a task they find is no longer relevant given the user has read/write permissions. |
| Task.Edit | The system shall allow a Patron to edit a task if they find that it needs to be changed given the user has read/write permissions. |

## 3.3 Application keeps track of all task lists

### 3.3.1 Description and Priority

The application will allow the user to easily see all of lists of tasks they currently have permission for upon logging into the system.

### 3.3.2 Stimulus/Response Sequences

Stimulus: A user creates a new task list.

Response: The system will query the user as to the specifics of the task list.

Stimulus: A user saves a new task list.

Response: The system will add the task list to the list of task lists.

### 3.3.3 Functional Requirements

|  |  |
| --- | --- |
| **Function** | **Description** |
| List.Create | The System shall create a new List and assign it to the Patron’s list collection. |
| List.View | The System shall show the Patron the requested List, provided the Patron has access to it. |
| List.Delete | The System shall remove the requested List, provided the Patron has administrative access to it. |
| List.ShareWith | The System shall assign the requested List to the identified Patron with the appropriate permissions. |
| List.Edit | The System shall request changes to the List’s properties and update them accordingly. |

# 4. External Interface Requirements

## 4.1 User Interfaces

UI-1: The Web pages shall permit complete navigation across all major modern browsers.

UI-2: The user will be directed to the homepage to start which will have all lists displayed or listed themselves

UI-3: When selecting a specific task list, there will be a page that gives the name of the task list, the names of all tasks to be completed, and buttons for adding/removing tasks from the list.

UI-4: All pages will give the user a link to their homepage, their owned lists, their shared lists, and their account page.

UI-5: When first arriving at the site, a user will be greeted with a login screen. The screen will ask for a username and a password.

## 4.2 Hardware Interfaces

None Defined

## 4.3 Software Interfaces

SI-1: List of Task Lists Interface

SI-1.1: The List interface will display all task lists the current user has access to.

SI-1.2: The List interface will allow a user to select any list for more detailed viewing.

SI-1.3: The List interface will allow a user to delete or add a new task list.

SI-2: The Task List

SI-2.1: The task list will display all tasks in the list, sorted by the current priority type.

SI-2.2: The task list will allow for the creation or deletion of tasks.

SI-2.3: The task list will allow for a task to be checked off or unchecked.

SI-3: The New Task Interface

SI-3.1: The new task interface will allow users to enter title, due date, and short description to each task.

SI-3.2: The new task interface will allow users to cancel the creation of a task.

## 4.4 Communications Interfaces

None defined

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

PE-1: The system shall accommodate 400 users during the peak usage time window of 10:00am to 3:00pm local time, with an estimated average session duration of 5 minutes.

PE-2: All Web pages generated by the system shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.

PE-3: Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query.

PE-4: The system shall display confirmation messages to users within 4 seconds after the user submits information to the system.

## 5.2 Safety Requirements

No safety requirements have been determined.

## 5.3 Security Requirements

A user’s data shall not be accessible by anyone unless the user is registered and signed in or the user is accessing the website on their personal machine.  A user may also grant other registered users to access specific lists that they own with read-only or read-write permissions.

SE-1:  All network transactions that involve financial information or personally identifiable information shall be encrypted per BR-33.

SE-2:  Users shall be required to log in to the Cloud Task Management System for all operations.

SE-3:  Users shall log in via an encrypted HTTPS session.

SE-6:  The system shall permit Users to view only their own tasks and task lists as well as those share with them, not those of other Users.

## 5.4 Software Quality Attributes

Availability-1: The Cloud Task Management System shall be available to users connected to the internet 99.9% of the time.

Robustness-1: If the connection between the user and the system is broken prior to a task or list creation being finished, the Cloud Task Management System shall enable the user to recover.

# Appendix A: Data Dictionary and Data Model

administrator = \*The administrator of a given list will be the creator of the list. They will be the only one able to delete the list, and they can share the list with anyone they choose. They can give others  either read/write privileges or read only privileges.\*

due = YYYYMMDDHHMM \*the due date and time of the task\*

list = name

+ priority type

+ tasks

name = \*null or 1-32 characters\*

number = \*int from 1 to n, n being the number of tasks in the list, with no repeated number in the list that indicates priority\*

owned lists = \*array of lists that were created by the user\*

password = \*4 to 16 characters\*

priority type = [due | number | alphabetical] \*the way the user would like the list to be sorted\*

read/write = \*This user can add new tasks to a task list, as well as being able to delete tasks that are not relevant. They may mark tasks as complete as well.\*

read only = \*This user can see the tasks in the task list and check them off (or uncheck them) but they may not add or delete tasks.\*

shared lists = \*array of lists that were shared with the user\*

task = text

+ priority

tasks = \*array of type task\*

user = username

+ password

+ name

+ owned lists

+ shared lists

username = \*4 to 16 characters\*

visitor = \*A user that has not logged in to the service.\*