Design Document

For

Cloud Task Managing Application

Version 1.0

Charlie Marshall

Shelley Murphy

Wesley Rupert

September 2013

Contents

[1. Introduction 1](#_Toc369550134)

[1.1. Purpose 1](#_Toc369550135)

[1.2. Scope 1](#_Toc369550136)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc369550137)

[2. Development 1](#_Toc369550138)

[3. Architectural Goals and Constraints 1](#_Toc369550139)

[3.1. Technical Platform 1](#_Toc369550140)

[3.2. Security 1](#_Toc369550141)

[3.3. Persistence 1](#_Toc369550142)

[3.4. Reliability 1](#_Toc369550143)

[3.5. Performance 2](#_Toc369550144)

[4. Work Flow 2](#_Toc369550145)

[4.1. Account Registration 2](#_Toc369550146)

[4.2. Login 2](#_Toc369550147)

[4.3. List Creation 2](#_Toc369550148)

[4.4. List Item Creation 2](#_Toc369550149)

[4.5. List Item Deletion 2](#_Toc369550150)

[4.6. List Sorting 3](#_Toc369550151)

[4.7. List Share 3](#_Toc369550152)

[4.8. List Delete 3](#_Toc369550153)

[5. Data Flow 4](#_Toc369550154)

[5.1. Data Model 4](#_Toc369550155)

[5.2. Create User 4](#_Toc369550156)

[5.3. Create List 4](#_Toc369550157)

[5.4. Update List 4](#_Toc369550158)

[5.5. Save Items 5](#_Toc369550159)

[5.6. Delete List 5](#_Toc369550160)

# Introduction

## 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.

## Scope

The scope of this design document is to lay out the architecture of the Cloud Task Managing System defined by us.

## Definitions, Acronyms, and Abbreviations

ACM: Association for Computing Machinery, an organization we have a chapter of on this campus

# Development

This product will be developed using PyCharm 3 as a standalone web-based application.

# Architectural Goals and Constraints

## Technical Platform

The Cloud Task Managing System will be deployed on an ACM server, the specific details of which are still being gathered.

## Security

The system will allow users to create profiles with personal information. This will be encrypted and stored securely. Other basic security measures are not allowing access to an account unless provided with the correct user credentials and not allowing users with the wrong credentials the power to edit or delete task lists.

## Persistence

Data will be persisted through the use of a relational database.

## Reliability

The site will be up and running constantly, in case anyone ever wants to check on their tasks in the middle of the night

## Performance

All Web pages generated by the system shall be fully downloadable in no more than 10 seconds

# Work Flow

## Account Registration

Upon clicking a link titled “New User”, users will be directed to a new page with blank fields to enter a username, password, and name and a button titled “Register”. When the form is filled out, they will click “Register”. If the entered username is already taken or one of the fields is blank, they will be prompted to use another one and click “Register” again. If the information is good when clicked, they will be logged into the newly created account.

## Login

If users access the website and are not logged in, they will see two empty fields on the homepage titled “Username” and “Password” with a button titled “Login”. The user will enter their username and password into the respective fields and click “Login”. If the information does not match a username and password pair in the database, the page will notify the user to enter different information. If the username and password match, the user will be logged into the account by that username.

## List Creation

A user who is logged in will be able to see a button that says “Create New List”. When pressed, a window will pop up that will have the fields titled “Name” and “Category”, as well as a button labeled “Create” and a button labeled “Cancel”.. When the “Create” button is pushed, if all fields have been filled out, a new list will be added to the user’s task lists. If they have not been filled out, a prompt will pop up asking the user to fill out the fields before proceeding. If at any time the user presses the cancel button, the window will close.

## List Item Creation

When a user selects a given list, they will see each of the items currently in the list and button that says “Create Task”. If the user were to click the button, they would have a window pop up titled “New Task”. There will be fields for “Name”, “Due Date”, and “Description”, as well as a “Create” button and a “Cancel” button.

## List Item Deletion

When a user selects a given list, they will see each of the items currently in the list. Each task will display its name, its due date, and have a trash can icon on the far right of the task. Clicking this will bring up a prompt asking if you really want to delete the task, with a “Yes” button and a “No” button. Clicking the “Yes” button will remove the task from the list and the database. Clicking “No” will return to the previous screen.

## List Sorting

When a user changes the sorting property, they will see the list’s items in a different order, either by due date, number added, category, or alphabetical. Clicking the sort button will cycle through these sorting methods.

## List Share

When a user selects a given list, they will see each of the tasks currently in the list. Under the “Create Task” button will be a button that says “Share List”. When clicked by a user, a window will pop up with two columns. The left will be titled “Add User” and the right will be titled “Permission Level”. The user will be able to search for another user and have their username added to the first column. The user will then be able to choose one of three permission types (read-only, read-write, or owner) for the right column. There will be two buttons at the bottom, one that says “Ok” and one that says “Cancel”. If the user does not enter any users, the window will close. If they entered users but not permission levels, the users will be added to the list with “read-write” permissions.

## List Delete

When a user selects a given list, they will see each of the items currently in the list. Under the button for “Share List” will be a button that says “Delete List”. If a user clicks this button, a window will pop up. It will warn the user that Deletion is permanent with a message “WARNING: If you click yes, your list will be permanently deleted! Delete this list?” There will also be a button labeled “OK” and a button labeled “Cancel”.

# Data Flow

## Data Model



## Create User

Once a User has created an account, the server will create a new User entry in the database and add that User’s instance to the cookie for that browser. A user can clear this cookie by logging out, and retrieve it again by logging in.

## Create List

When the user creates a list, the server will create a new TaskList entry in the database and add the list’s ID to the user’s collection of owned lists and the User’s ID to the TaskList’s collection of owners. It will be populated with default metadata to start.

## Update List

When the user changes a list’s metadata and saves it, the server will update the corresponding TaskList entry in the database, adding and removing the list’s ID from the affected users’ (if any) owned/read-write/read-only collections and updating the TaskList’s collection of owners.

## Save Items

When the user saves a list’s items, the server will update the corresponding TaskList entry in the database.

## Delete List

When a user deletes a list, the server will remove the list’s ID from all connected Users and delete the corresponding TaskList from the database.