Task List

Ryan L. Dockstader Project Proposal

# Project Description

The Task List will be a web application that will allow a user to setup task groups, and task sub groups, and then add tasks to those groups. The user will be able to add tasks, mark them as complete, mark them as no longer needed, or remove them. The audience of this web app is anyone trying to balance multiple things in life, is often completing tasks surrounding those things on a computer and needs something to track their progress through those tasks. An example of this is someone who is balancing work, school, and family. This person could create three primary task groups: Work, School, and Family. They could then add a sub group for each class in school, and then add the tasks they need to complete that week on each class. They could then do the same things with their work, and family tasks.

# Design Overview

The user will be able to log in to the app and see each of their primary task group. They can then either add a task, or a new group to each of the task groups. They can also add as many sub groups as they would like. To accomplish this, I will need three tables. A user table, a task table and a taskGroup table. These tables will be linked together so each user can only see their task groups and tasks. The front end of the app will be simple and user friendly. There will need be three screens: the login page, the task groups page, and a page that will display tasks based on which group (or sub group) you select. Other partial views (such as headers and footers) will likely be used as well.

# Database

The app will be username and password based. The username will not be required to be an email, but will be required to be unique. The password will need to be stored and encrypted. The user table will the root table, with the other two linking back to the user.

## tUser

* UserID: string, required, PK
* username: string, required
* password: string, required

## tTaskGroup

* TaskGroupID: string, required, PK
* groupname: string, required
* ParentGroupID: tTaskGroup, FK ttaskgroup
* userID: tUser, FK tUser, required
* Priority: int

## ttask

* TaskID: string, required, PK
* taskname: string, required
* taskdesc: string
* TaskGroupID: tTaskGroup, FK ttaskgroup, required
* userID: tUser, FK tUser, required
* addDate: date, required
* completedDate: date

# Database Retrieval

The task groups will be retrieved and displayed on the apps home page. These will be listed in task groups and sub groups. Each one will also be a link that will open a new page. That new page will list each task with the ability to complete the task, mark it as canceled, or delete it. The tasks groups will be ordered by priority, and then alphabetically within each of their respective groups. The tasks will display the oldest task first, and the most recently added task last. It will only show the tasks that are currently not completed, or canceled.

# Database update

The user will be able to add new task groups, and new tasks. They will also be able to update and delete the tasks and tasks groups. When a task group is deleted, all the sub groups, and all tasks tied to any of the task groups will need to be deleted. Once the username is set it will not be able to reset. I will need to ensure that any time an update or delete takes place, that it is the owner of the task or task group that is doing the update or delete.