**Step by Step Process**

**3/23/2021**

Began planning process

Estimated number of .jsp classes

Created Step by Step Document

**3/24/2021**

Repository created

Used Spring Initilizr to create project and downloaded zip file

Imported maven project to Eclipse

Added Register and about page

**3/25/2021**

Added Login page

Added Success page, added Update page

Added some css to existing pages

**3/26/2021**

Created Task entity and made a Repository and Service for it

Updated the User entity and linked the id from it to the Task

Added Dashboard, Tasks, and added functionality to view, add and delete tasks

**3/29/2021**

Added CSS changes, made website more aesthetically pleasing with bootstrap.

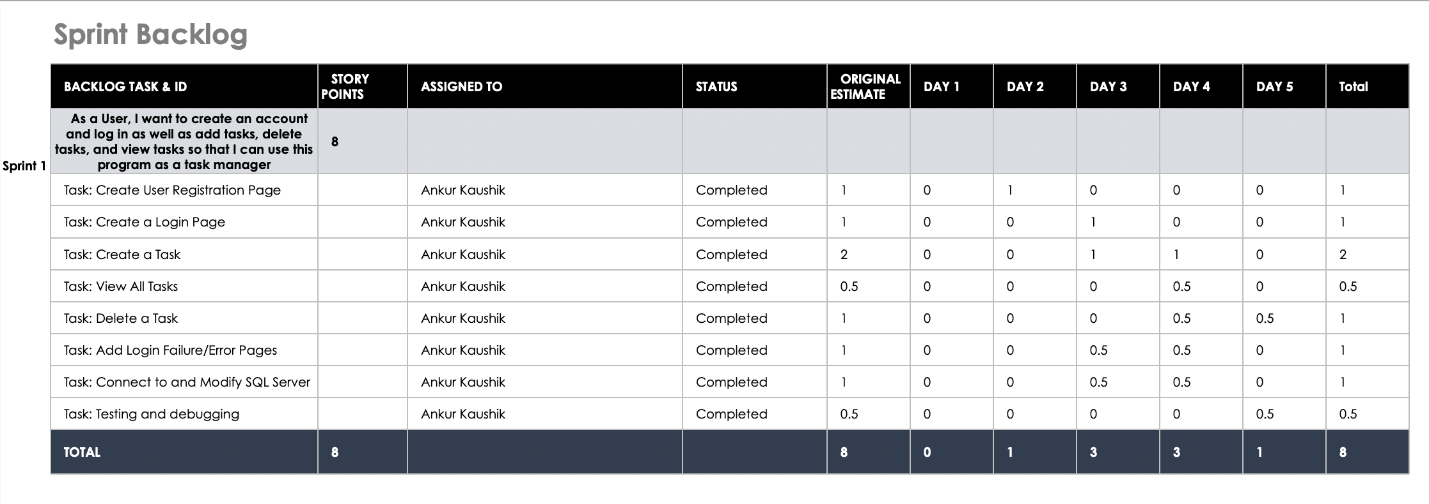
Finished writeup

Finished acquiring screenshots

Finished sprint backlog

Submit project

**Sprint Backlog**

****

**Writeup**

First I went to Spring Initilzr and created a maven project. I then downloaded the zip file and imported it into eclipse. I added the needed dependencies to the pom.xml file for the project. I then edited the Application Properties file to connect to my database, adding the username and password. I created a file for the homepage and links to reach the registration page and the login page. Once the user registers, their username and password are added to a SQL database table. They are then prompted to visit the login page so they can access the rest of the application. Upon successful login, the user is taken to a dashboard where the can choose to view all tasks, add a task, or delete a task. Adding a task creates a new Task object in a SQL table with the data provided by the user. The view and delete page shows all the tasks for the current user. (Note: the user can only see their own created tasks, not those of everyone). The user can choose to either delete a task in the table by providing its id in a form on the page or return to the dashboard. If the user returns to the homepage, their session expires and they must login again to access the dashboard and other pages. If the user inputs invalid login information they are taken to an error page and must login again.

I created a controller class which handles mapping for the home page, index page, update page, update2 page, register page, dashboard page, error page, success page, tasks page, and the no user page. I then added a user class, initialized it with the necessary components (id, name, email, password), and added getters and setters for these elements. I added a Task entity class with the required fields and created getters and setters for those as well. I created UserRepository and TaskRepository, classes that extend CrudRepository and allows me to make changes to the database. Next, I added UserService and TaskService that used UserRepository and TaskRepository respectively to find elements from the database. Last, I made css changes to center the text and add a background. I used bootstrap to give everything an aesthetically pleasing look and then uploaded everything to my github which can be found here: https://github.com/AnkurKaushik/TaskManager.