# Task 1 Individual Project: **TaskMaster**

You are the **manager** of a team of engineers. You have a collection of Tasks that need to be completed, and you want to come up with a plan for assigning tasks to engineers. You envision an application that will allow the **manager** to create a list of engineers, adding and removing from the engineer list as necessary. There is a separate list of tasks, and the manager can add and remove tasks from the task list as necessary. Each task that is created has an estimated time to complete in minutes.
There needs to be a visualization of the engineers and the tasks that they have been assigned. When tasks are completed, they are moved into a *completed task list*.

## Assigning Tasks

There is a visual representation of the existing tasks that have yet to be assigned. The manager can assign a task to one of the engineers – once a task has been assigned to an engineer, it cannot be unassigned nor can it be removed.

## Complete Task

An engineer can complete a task that they had been assigned. To do so, they **inform the manager** the actual number of minutes they worked on the task (by email, text, or a handing the manager a piece of paper). **Then the Manager enters in the actual completion time for the task and removes it** from the list of tasks that are associated with that engineer; **the task is then** placed into the *completed task list*.

## Use Cases

1. Add Engineer
2. Remove Engineer
3. Add Task
4. Remove Task
5. Assign Task
6. Complete Task

## Entity Classes

For the HW1.Analysis task (due September 19th) each student will complete the class diagram for entity classes

## StoryBoards

Mock-up some sample GUI images to visualize the experience from the point of view of the manager, showing all the different visual elements that are needed for this assignment. This must be constructed in HTML (with CSS as you see fit). This provides a concrete example for the final deliverable.
The storyboard (and application) is only to be used by the Manager; they are the only actor.

## Code

For the final deliverable, students will submit their project together with test cases that will ensure 80% code coverage of the application.

## End Notes

1. There is no need for persistent information! Start from an empty state and make it possible to add engineers, tasks and complete tasks
2. There are two deliverables (Analysis document, September 19th) and (Final Code, September 30th)

**Deleted:** predicted

**Deleted:** must enter in

**Deleted: The task is then removed**

**Deleted:** and

**Formatted:** Font: Bold

3. First Version released 9-2-2025 at 5PM
4. Clarify that once a task has been assigned to an engineer, it cannot be removed
5. It must be possible to see the estimated time for all tasks, as well as the actual time to complete for a task, once it has been completed by an engineer
6. The application should show (a) the total number of estimated minutes for all unassigned tasks; (b) the total number of estimated minutes **by engineer** for all assigned (but not completed) tasks; (c) the total number of actual minutes for all completed tasks
7. Clarifies that in this simple assignment there is only ONE ACTOR – the Manager. To keep things simple, just assume the manager is the one who declares a task complete (with the actual number of minutes) and then the task is moved into the completed tasks list.