# CS480 – Course project

### Summer 2021

### Database: Productivity Tracker

**Description:**

The aim of this application aims to be a tool for teams and individuals to go become more productive. Users are able to create a task list and keep track of their progress. Teams are able to create a group of tasks and assign who needs to work on what!

Our application allows teams and individual users to organize tasks and be able to track statistics. Our database will be composed of three main entities: Users, Tasks, and Teams. A user can be part of 0 or more teams, where 0 means the user intends personal use. The user will have their own unique id, user name, email, team position is any and a login password. A team will have a many-to-many relationship with users where each team has its own unique Team ID, Team Name. Individual users will be able to complete tasks on their individual task list or a team's task list. Tasks in general for all teams and users will be in one single table. Teams and individuals can organize tasks by giving each task a category attribute/category as well as a priority level. Tasks can be ongoing tasks or even completed tasks and this will be able to be updated by a user. A task will include a unique task ID, task name, author, description, creation date, due date, actual completion date, category, and a completed boolean state indicating if the task is still active. Users will be able to visualize data such as total tasks completed in a given week, and completion consistency, and other stats from the task table.

# Part 2 – CRUD (Create, read, update, and delete)

**List of strong entities:**

1. User
2. <strong entity 2>
3. <strong entity 3>

**List of weak entities:**

4. Task (Each task must have a user)
5. Team (Comprises of users)
6. <weak entity 3>

We will implement the following functionality using Java and SQL with necessary GUI interfaces.

1. Create/read/update/delete an **User** (all attributes except the user id). The user-id should be generated by the system automatically using MySQL autoincrement.
2. Create/read/update/delete a **Team** (all attributes except the team id). The team id should be generated by the system automatically using MySQL autoincrement.
3. Create/read/update/delete a **Task** (all attributes except the task id). The task id should be generated by the system automatically using MySQL autoincrement. None of the tasks can be deleted, to maintain the history of tasks. The tasks can be set to active or inactive using a boolean value.

# Part 3 – Queries

Based on the Demo, we will implement the following functionality using Java and SQL with necessary GUI interfaces.

**Trivial Queries:**

1. List all Users
2. List all Teams
3. List all Tasks

**Non-trivial Queries:**

1. List all Tasks for a particular Team and User.
2. Get the time spent by a particular user on completed tasks in a certain time period.
3. Get the Time spent by all users in a time on the team tasks.
4. Filter tasks based on priority for a given Team or User.
5. List tasks due on a given date.