**Author: Dionysios Tachtsidis**

**Title: Superhero Sightings**

**Functional description**

This program allows a user to interact with a database and perform CRUD activities on a database designed to represent and record superhero sightings.

The program uses a database to store the Superhero, Location, Organization and Sighting data.

The program allows the user to perform the actions through a front end page, that calls POST and GET requests through a controller. The controller then communicates with the appropriate DAO and performs the required action.

### Prioritization

The biggest priority for that project, is to design the database and translate the appropriate relationships.

For those purposes, at first, I created a database, inserted data through the workbench and worked on sql scripts that retrieve the data I want.

After that, we prioritize the database communication part.

In order to achieve that, I reused some code found in the materials, and configured it to the project’s needs.

Then, through the controller class, I set up paths for POST and GET requests that handle the user requests.

Lastly, we create a front-end landing page. Then the landing page for each element, and an edit page for that element.

**KNOWN PROBLEMS**

* **BEFORE ANYTHING HAPPENTS THROUGH THE FRONT END, THE FULL SQL SCRIPT MUST BE RUN. IT INSERTS HERO DATA MANUALLY.**
* **THAT HAPPENTS, BECAUSE THE USER MAY TRY TO ADD AN ORGANIZATION BEFORE THEY ADD A HERO, AND THAT WOULD BREAK THE PROGRAM.**
* **So far, the working parts are heroes, locations, organizations. I left out the sightings table, because it handles the relationship with both heroes and locations, and it will require further configuration to work as planned.**
* **Edit doesn’t work for the Organizations, and edit for the location latLong doesn’t go through**
* **DATABASE PASSWORD IS LEFT BLANK**.

**FILES INCLUDED**

1. Project file with my project.
2. SQL scripts for the DB, the testDB and a script with various select requests that I used within the actual project.
3. ERD diagram of the DB.
4. Three wireframes, highlighting the general concept of how the front end will look.