**Project Participants:**

Charles Fehringer

**Title:**

 Irrigation Pivot Monitoring system

**Executive Summary:**

This will be a backend that allows customers to connect remotely to their pivots. It will allow them to check the direction their pivots are facing as well as alert them if something breaks down. They will be able to start/stop the pivots, as well as change their direction and turn the water on/off.

There will be two different clients connecting to the database. The pivots themselves will connect and report position as well as any problems. The customer will connect and to view the pivot status as well as send any control changes.

**Initial Features:**

* Entities: Customer, Pivot, Contacts, Customer\_Pivots
* A Customer can perform the following operations:
  + Read:
    - get customer info
  + Update:
    - put customer info
* A pivot can perform the following operations:
  + Create:
    - post new pivots
  + Read:
    - get pivot data
  + Update:
    - put new pivot data
  + Delete:
    - delete pivots
* A contact can perform the following operations:
  + Create:
    - post new contact data
  + Read:
    - get current contact data
  + Update:
    - update existing contact data
  + Delete:
    - delete contact data
* A customer\_pivot can perform the following operations:
  + Create:
    - post a new customer pivot relationship
  + Read:
    - get current relationships
  + Delete:
    - delete a relationship
* Extra API endpoints
  + Generate Key – used to generate new keys for manually entered entities

**Stretch Goals (to be completed if time allows, or after graduation):**

* Add security
  + Prevent customers from viewing pivots that are not theirs - Authentication
  + SSL/TLS encryption so no one can steal passwords
  + Learn how to safely store passwords in the database – Hashing?
* Add zones
  + These will allow automated features such as the ability to automatically turn the water off when walking over a certain area
  + Auto Stops - Automatically stop when the pivot reaches a certain position

Questions for approval meeting

* Should I make a separate controller/service layer/Dao for every entity. Where should they merge?
* Alert system with rest. Does it have to use polling, or is there a way to have callbacks?
* Information (stuff I can also research on my own):
  + AWS & Other hosting options
  + Security