Final Project Criteria

**Project Requirements:**

•         Database design which contains at least 3 entities and 3 tables

•         Contains all CRUD operations (Create, Read, Update & Delete)

•         **Each entity** should have**CRUD operations** with one entity having all 4 CRUD operations (Create, Read, Update & Delete).

•         Contains at least 1 one-to-many relationship

•         Contains at least 1 many-to-many relationship with one or more CRUD operations on this relationship

**Project Participants:**

Bryan Crawley

**Title:**

Parks

**Executive Summary:**

This is where you will describe what your project is intended to do. Remember, this project should be completed in 2 weeks time. If you can’t explain it in 1-2 paragraphs, you may need to scale it back so you can complete your initial features on time.

*This project will be a small sample of Portland, Or. Parks and the features they have. It will be connected through an entity with a singular purpose of the connection, called Parks-Features-Link. This simple project will demonstrate the required relationships but is easily able to grow for an expanded portfolio piece.*

**Initial Features:**

Create a bulleted list of planned features you plan to have completed in your project by the deadline. This should include a list of API endpoints for each feature. If you are working in a group, please note which team member will be assigned to each feature.

Example of list of Features/Endpoints for a Library API:

* Entities: Users, Administration, Books, Genre, Checkout, BookReviews, etc.
* A User can perform the following operations:
  + Login, and use system
  + Browse all Books (GET on Books)
  + Browse Books by genre (GET on Books with genre specified)
  + View all details about a specific Book (GET on Books by primary key)
  + Leave review on a Book (POST in BookReviews)
  + Read reviews on a Book (GET on BookReviews)
  + Checkout X amount of books at a time (POST in Checkout), set dueDate for 2 weeks from today (e.g. Checkout Date) and (PUT Change Status of Book to Unavailable)
  + Return a Book (PUT Change Status of Book to Available, PUT Checkout to RETURNED status, POST Fee on Users, by primary key, if the book is past due, etc.)
* Entities: Parks, Features, Parks\_Features\_Link, Events
* Browse all Parks (GET on Parks)
* Add a Park (POST in Parks)
* Update the size or features of a park (PUT in Parks by ID)
* Delete a Park (DELETE on Parks by ID)
* Add a Feature to the list of possible features (POST on Features)
* Add a Feature to a Park (POST on Park\_Feature\_Link)
* See all possible Events (GET on Events)
* Add an Event to the list (POST on Events)
* Delete an Event (DELETE on Events)

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

Create a bulleted list of planned features you plan to have once you have completed your initial features. These should be features that may require more research in how to implement or features that would take longer than the allotted time frame.

* More parks
* More features
* Delete features
* Search Parks by available Feature (GET on Park\_Feature\_Link)
* Events that can be linked to parks or features
  + Search by events, giving USER a list of what parks said Event is at
* Multi-faceted or stages of events
  + Bi-Weekly tournaments
* New entity with more public spaces such as local lakes or libraries
* Many more exception handling