Project Participant: Dan Gaffey
Title: My Wine Cellar

Executive Summary:
My REST API provides a simple way to take your physical wine inventory and place it into a virtual wine cellar that you can access from anywhere. It will utilize HTTP methods that will tell the API what action to take.

Initial Features:
Users will be able to:

GET: Retrieve wines in their inventory
POST: Add a new wine they've recently purchased to their inventory
PUT: Update a wine currently in inventory
DELETE: Remove a wine from their inventory

Entities:

- **wine** – represents the different wines in inventory
  - Attributes: wine_id, wine_name, vintage, varietal, color, appellation, quantity, price, tasting_note
- **cellar** – represents individual wine cellars or storage locations
  - Attributes: cellar_id, cellar_name, location, capacity, owner_id
- **owner** – represents users of the system
  - Attributes: owner_id, username, password,
- **cellar_wine** – represents the relationship between wines and cellars, each record in this table indicates a specific wine stored in a particular cellar

Relationships:

One-to-Many between owner and cellar
Each owner can manage multiple cellars, but each cellar is managed by only one owner. This relationship is represented by the **owner_id** foreign key in the **cellar** table, referencing the **owner_id** primary key in the **owner** table.

Many-to-Many between cellar and wine
Each cellar can contain many wines, but each individual wine is only stored in one cellar. This relationship is represented by the **cellar_id** foreign key in the **cellar_wine** table, referencing the **cellar_id** primary key in the **cellar** table.

List of API endpoints:

/wines

- GET – retrieve a list of all wines
- POST – add a new wine to the database

/wines/{wine_id}

- GET – retrieve details for a specific wine by its ID
- PUT – update details for a specific wine
- DELETE – delete a wine from inventory

/cellars

- GET – retrieve a list of all cellars
- POST – Add a new cellar to the database

/cellars/{cellar_id}

- GET – retrieve details for a specific cellar by its ID
- PUT – update details for a specific cellar
- DELETE – delete a cellar from the database

/owners

- GET – retrieve a list of all owners
- POST – Add a new owner to the database

/owners/{owner_id}

- GET – retrieve details for a specific owner by their ID
- PUT – update details for a specific owner
- DELETE – delete an owner from the database

/cellar_wines

- GET – retrieve a list of all cellar-wine relationships
- POST – Add a new cellar-wine relationship

/cellar_wines/{cellar_id}/{wine_id}

- GET – retrieve details of the cellar-wine relationship for a specific cellar and wine
- PUT – update details of the cellar-wine relationship
- DELETE – delete a cellar-wine relationship