

My Bucket List

A Python odyssey

Maggie Pentcheva 19th August 2021

The Vision

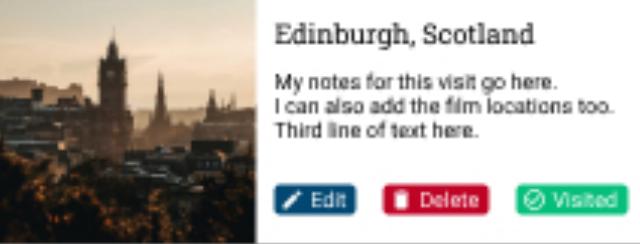
- A Travel Bucket List item that allows the user to also input film locations to travel to.
- Wireframes designed on Figma

My Bucket List

My List Add new location Inspire me

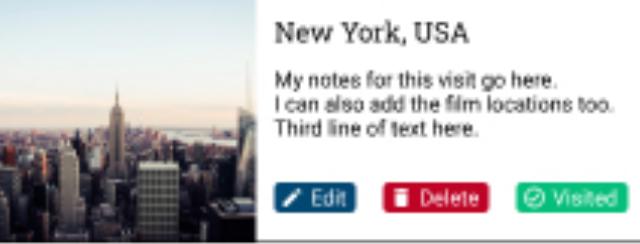
My List

Here's some text to describe how the bucket list works. It's super easy.



Edinburgh, Scotland
My notes for this visit go here.
I can also add the film locations too.
Third line of text here.

Edit Delete Visited



New York, USA
My notes for this visit go here.
I can also add the film locations too.
Third line of text here.

Edit Delete Visited



Wellington, New Zealand
My notes for this visit go here.
I can also add the film locations too.
Third line of text here.

Edit Delete Visited

My Bucket List

My List Add new location Inspire me

Add new

Here's some text to describe how the bucket list works. It's super easy.

Country

City

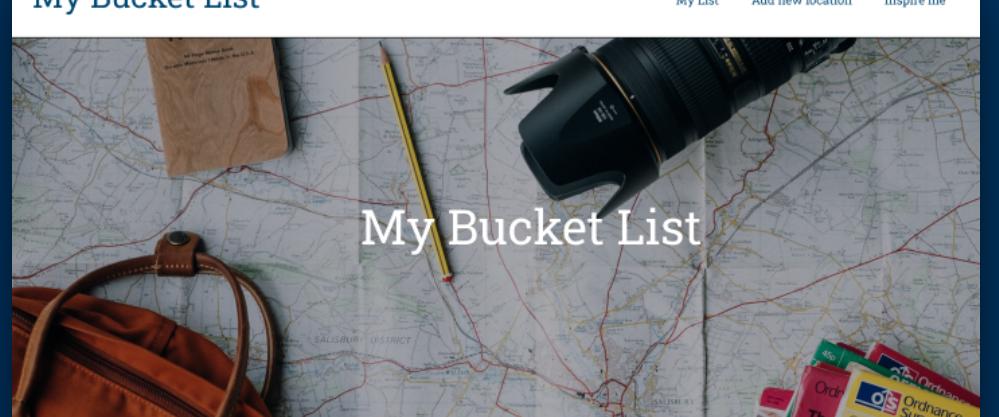
Film location

Add notes

Submit

My Bucket List

My List Add new location Inspire me

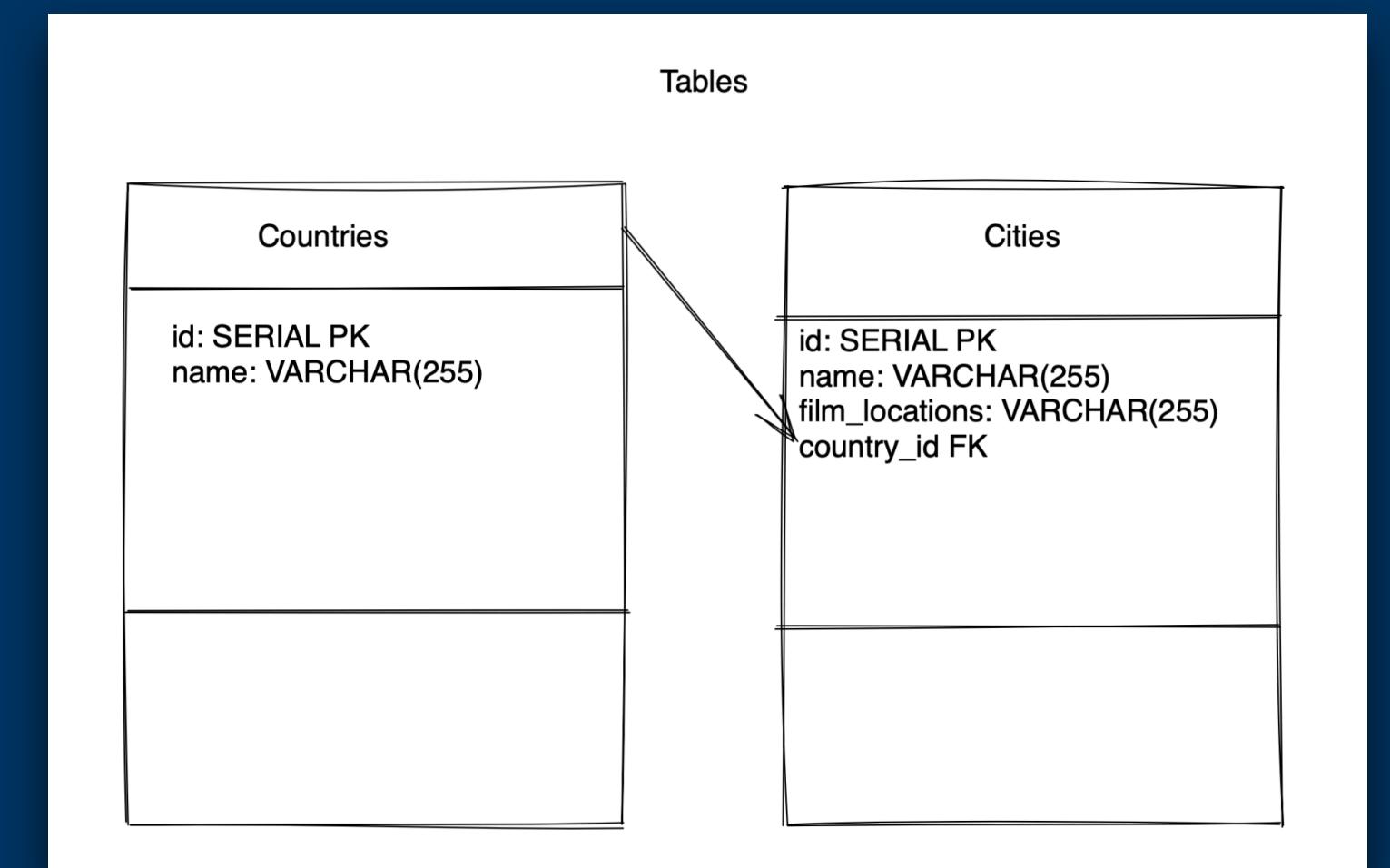
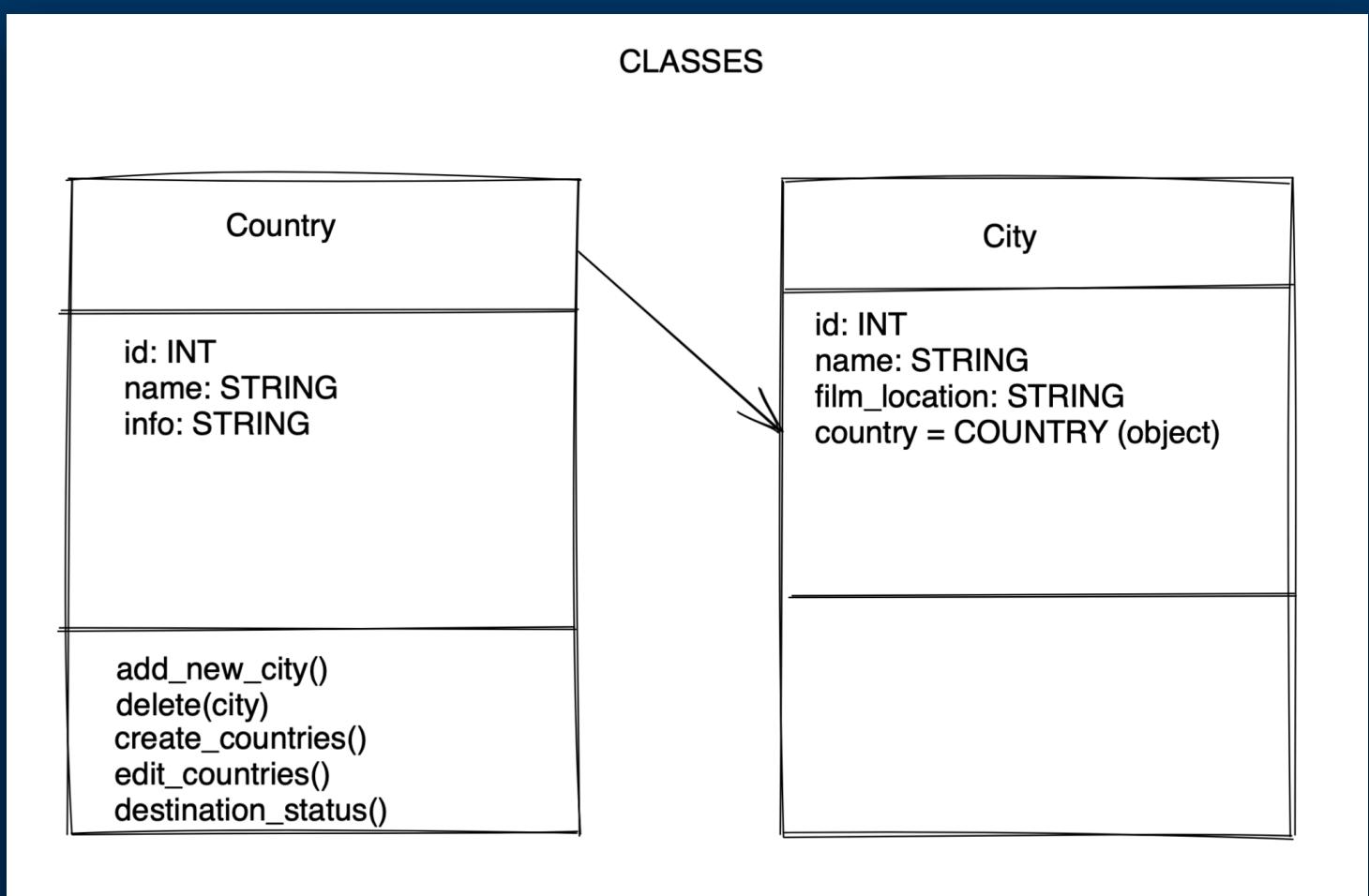
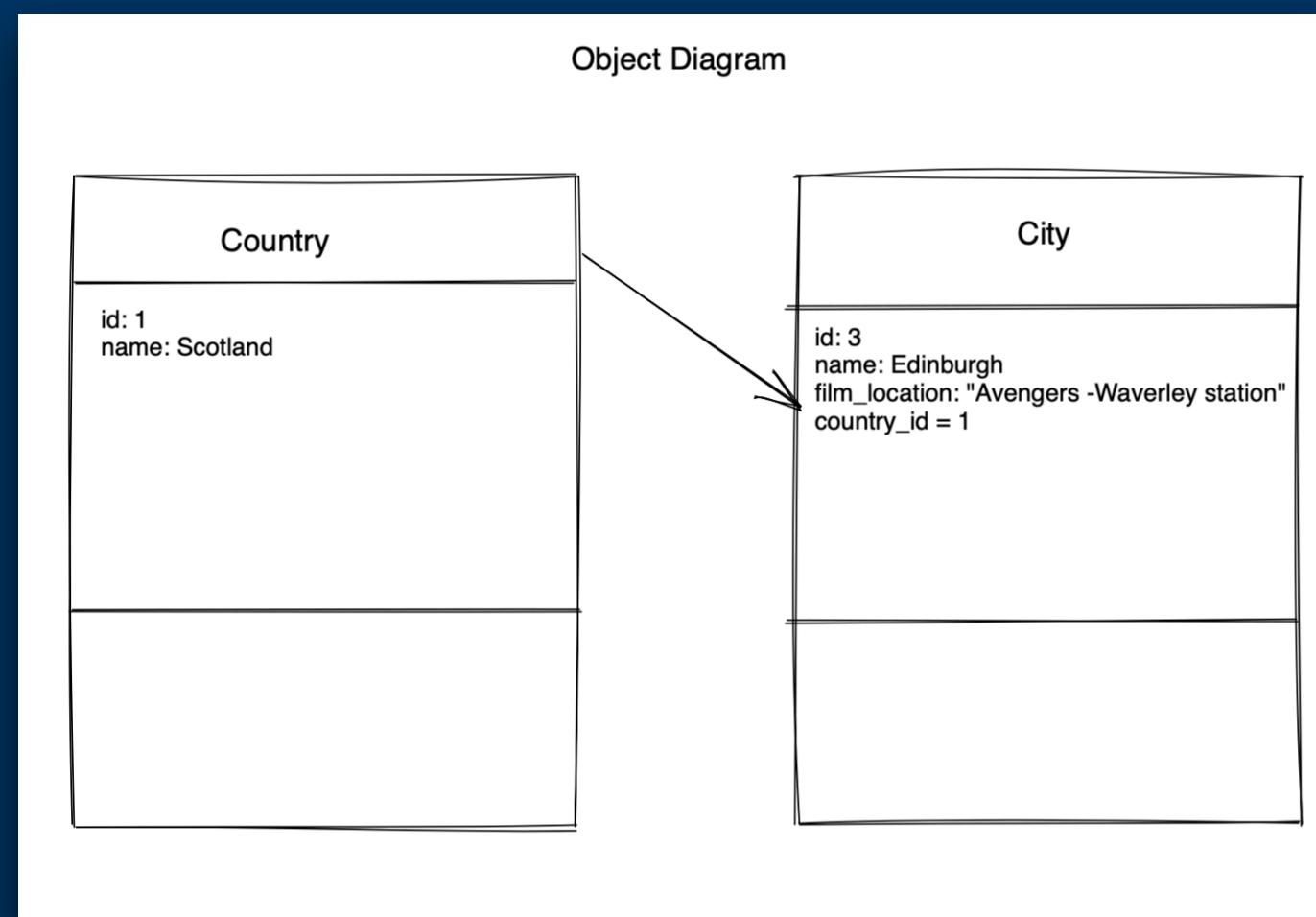
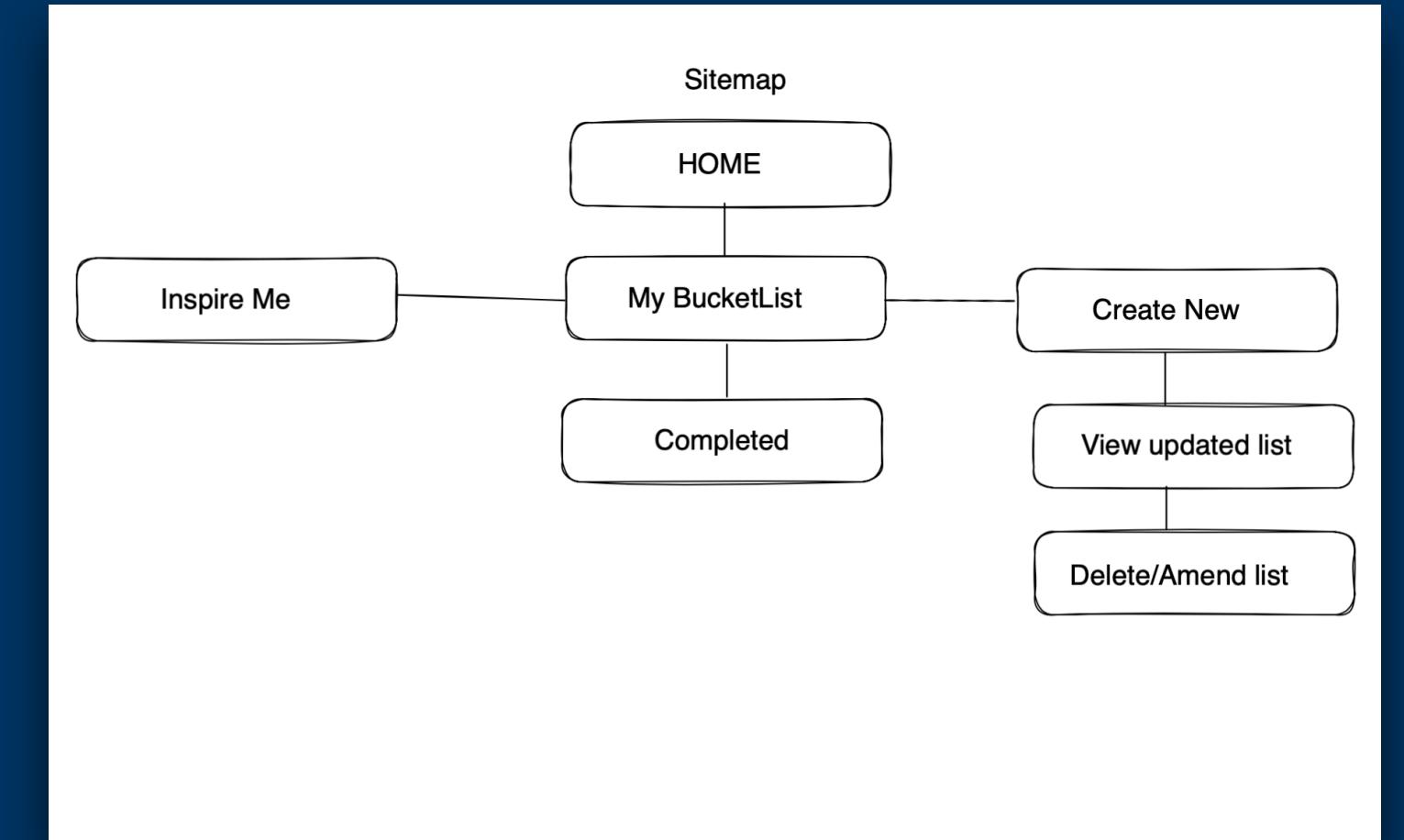
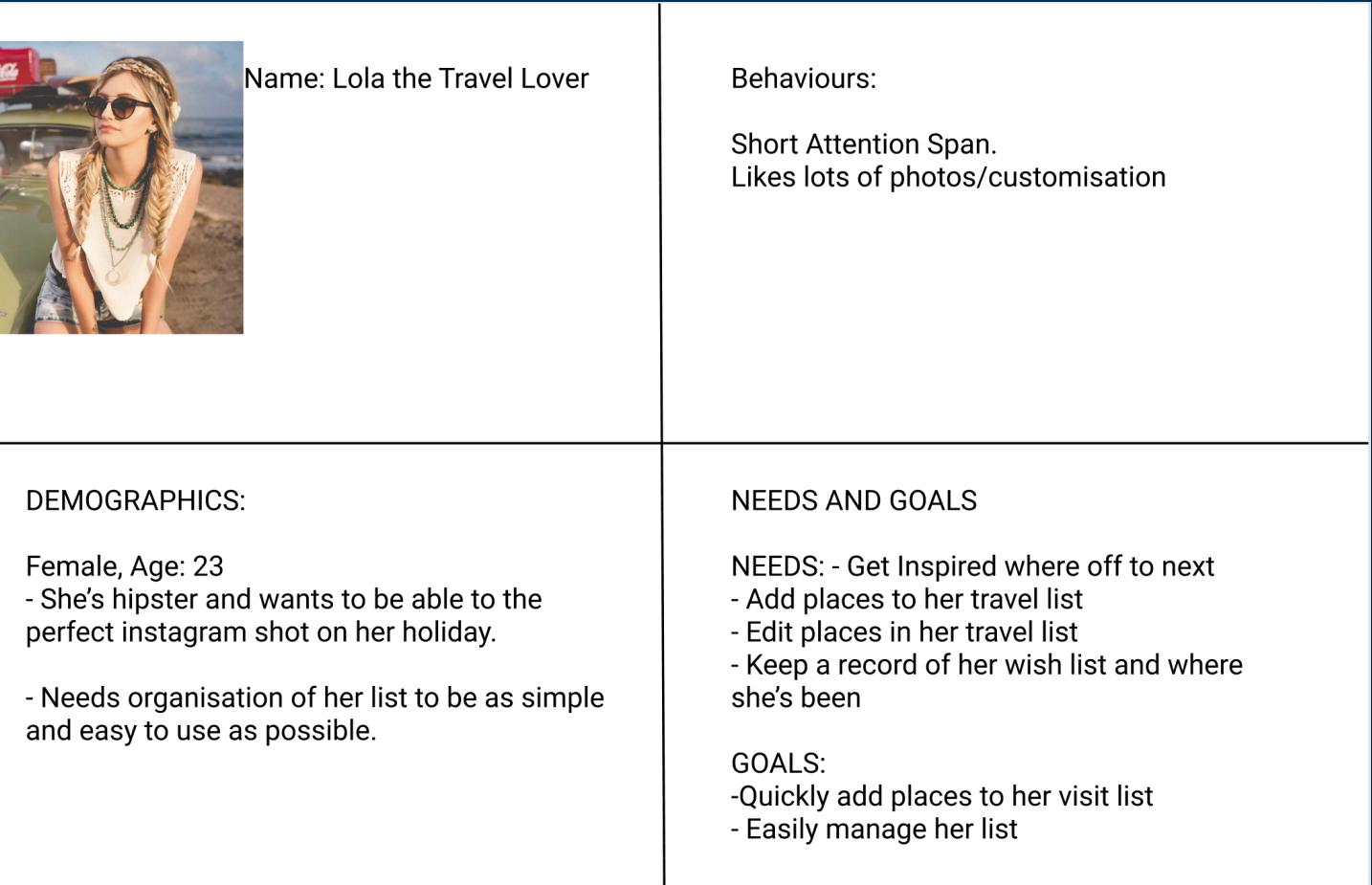
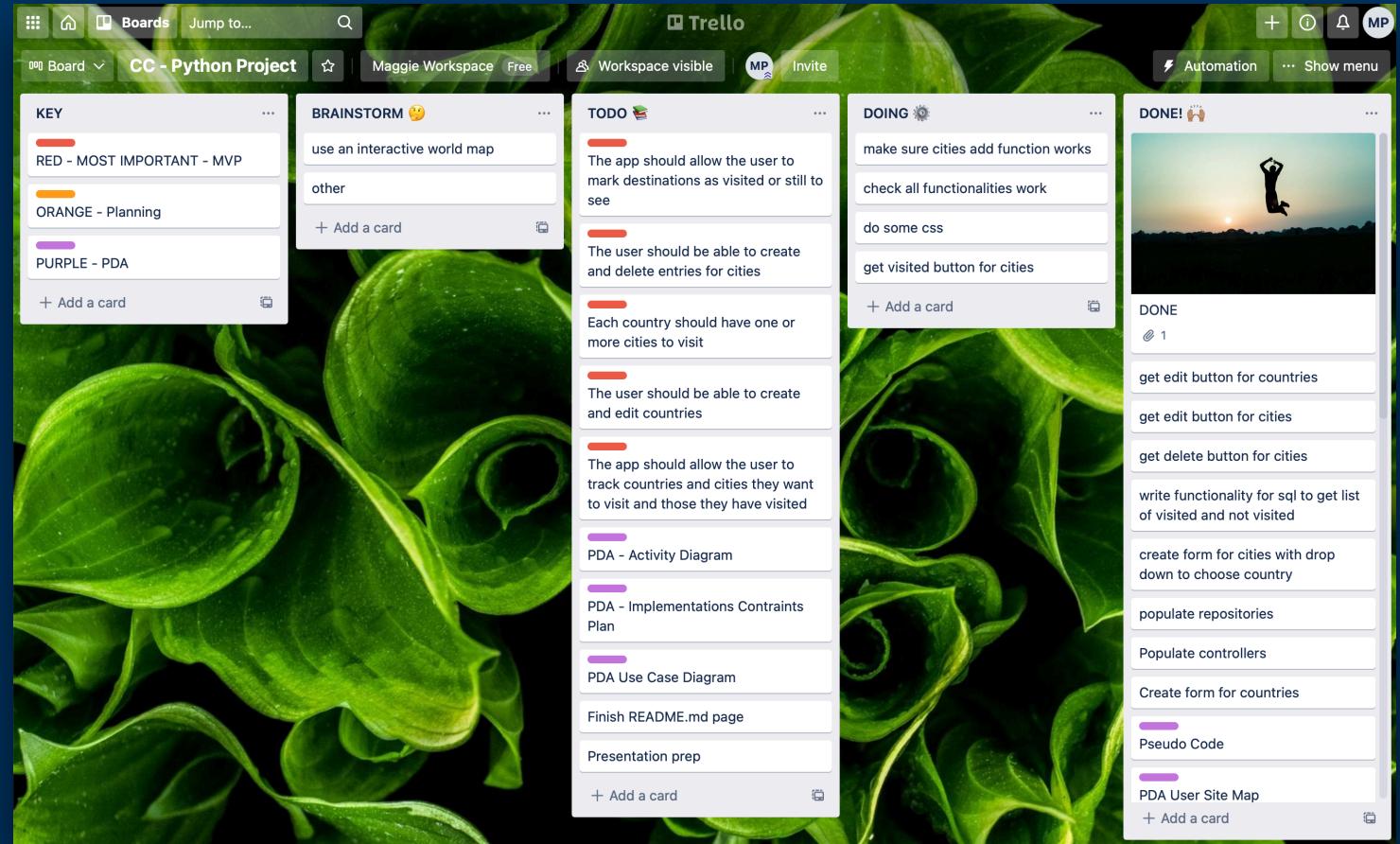


About

Here's some text to describe how the bucket list works. It's super easy.

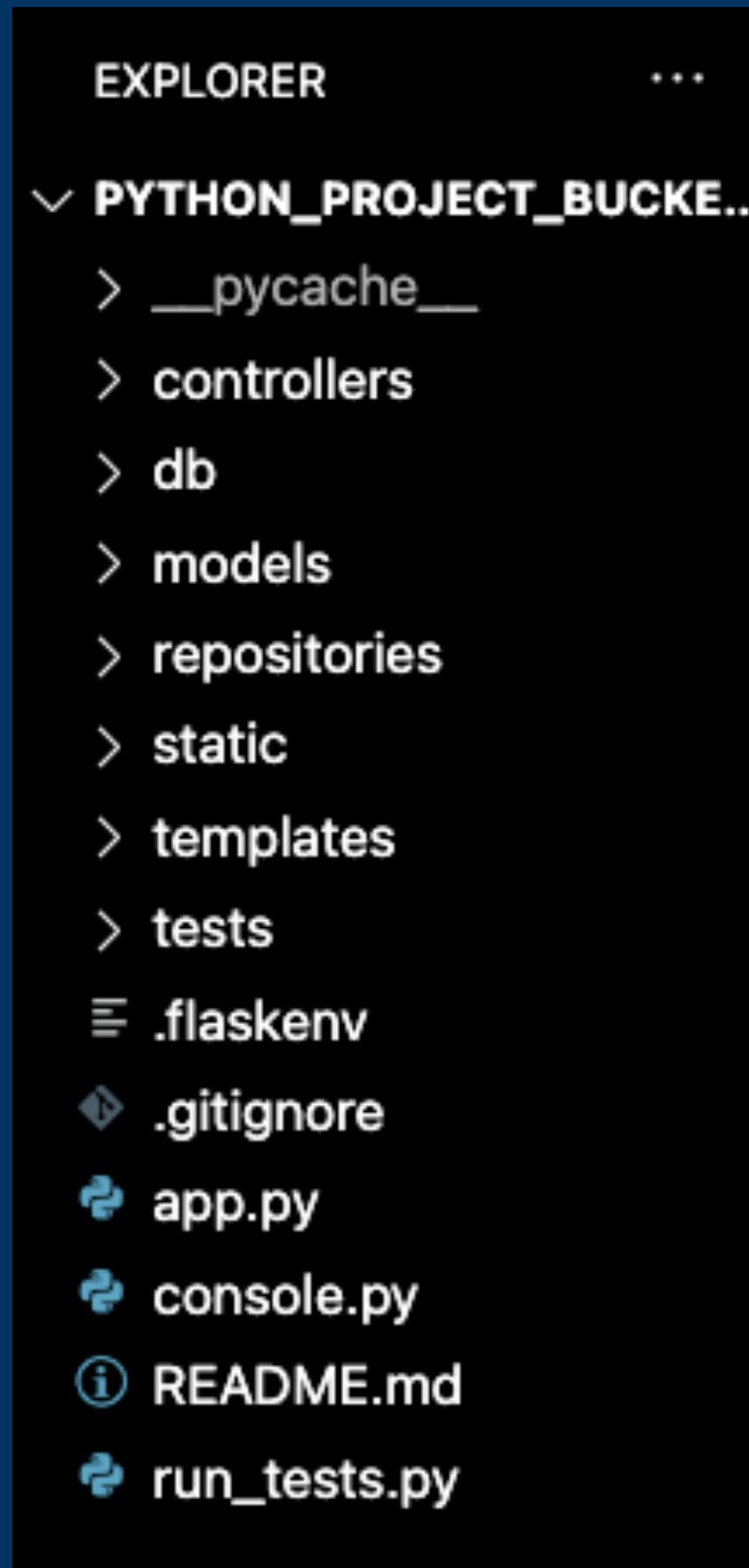
Add new location

The Planning



localhost:5000/DEMO

Good Code



The screenshot shows the VS Code editor with the file `city_controller.py` open. The code implements a Flask Blueprint for managing cities. It includes routes for listing cities, viewing a specific city, creating a new city, and editing an existing city. The code uses dependency injection for repositories and controllers.

```
city_controller.py — Python_Project_Bucketlist
-----
controllers > city_controller.py > delete_city
-----
1 from controllers.country_controller import countries
2 from flask import Flask, render_template, request, redirect
3 from flask import Blueprint
4 from models.city import City
5 import repositories.city_repository as city_repo
6 import repositories.country_repository as country_repo
7
8 cities_blueprint = Blueprint("cities", __name__)
9
10 @cities_blueprint.route("/cities")
11 def cities():
12     cities = city_repo.select_all()
13     return render_template("cities/index.html", cities=cities)
14
15 @cities_blueprint.route("/cities/<id>")
16 def show(id):
17     city = city_repo.select(id)
18     country = country_repo.countries(city)
19     return render_template("cities/show.html", city=city, country=country)
20
21 @cities_blueprint.route("/cities/new", methods=['GET'])
22 def new_entry():
23     countries = country_repo.select_all()
24     cities = city_repo.select_all()
25     return render_template("cities/new.html", countries=countries, cities=cities)
26
27
28 @cities_blueprint.route("/cities/new", methods=['POST'])
29 def create_entry():
30     country_id = request.form['country_id']
31     city = request.form['city']
32     film_locations = request.form['film_locations']
33     visited = request.form['visited']
34     country = country_repo.select(country_id)
35     new_entry = City(city, film_locations, country, visited)
36     city_repo.save(new_entry)
37     return redirect("/cities")
38
39 # GET/ EDIT 'cities/<id>/edit'
40 @cities_blueprint.route("/cities/<id>/edit", methods=['GET'])
41 def edit_cities(id):
42     city = city_repo.select(id)
43     countries = country_repo.select_all()
44     return render_template('cities/edit.html', city=city, countries=countries)
45
46
```

Main Learning Points

- Seeing how MCV interacts
- Making a good plan but being able to adapt on the go
- Leaving lots of time for debugging



ANY QUESTIONS

DO YOU HAVE?

makeameme.org