RecycleSort

By Abbigail Nicholson, Kara Wallace, and Maura Winstanley

Final State of System

Implemented:

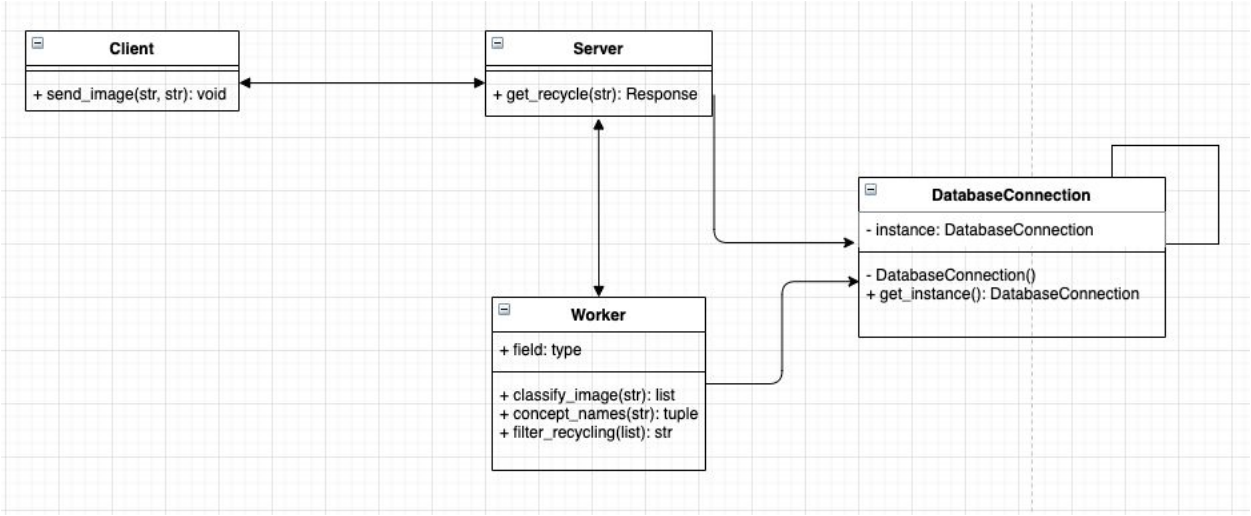
- Returning the recyclability of objects in an image
- A web app user interface

Not Implemented:

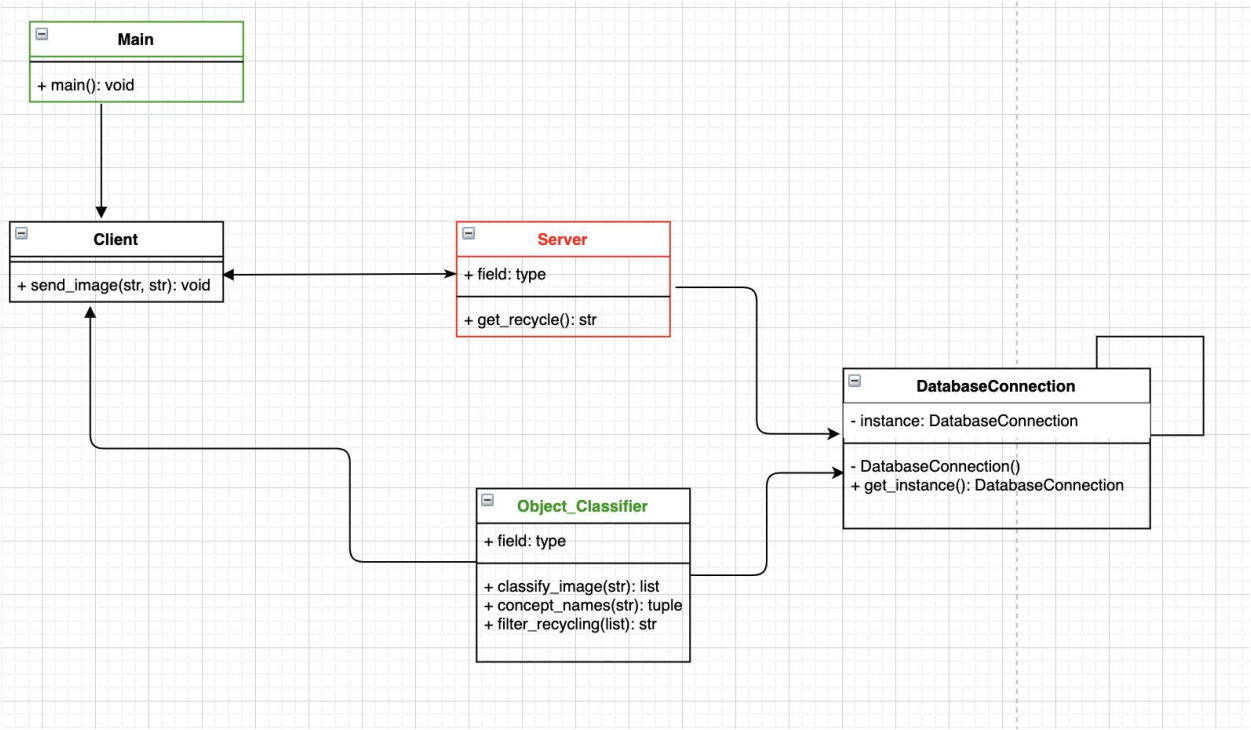
- A photo repository containing images with the recyclability of the objects. This was not implemented because we encountered some problems and a lot of bugs which left not enough time to implement this feature.
- Storing the images in a Cloud storage bucket. This was not implemented because it was also a lower priority that is not completely necessary. We ran out of time due to other challenges with the main functionality

Final Class Diagram and Comparison Statement

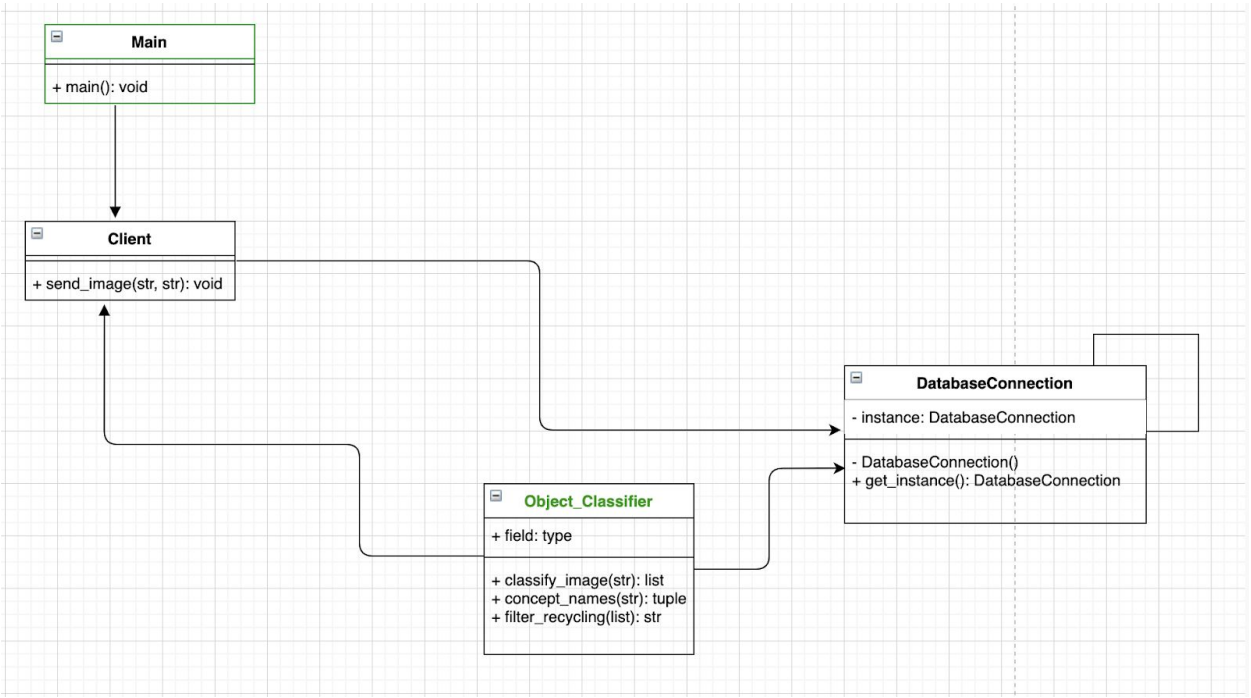
Project 4 UML:



Project 5 UML:



Final Result



From the progression you can see that what we had called the server class was not implemented. This is because it didn't really end up needing to be there because the object classifier acts as the server. We were not able to implement the photo repository where that class (which would have been named something else) would have been necessary.

We were still able to implement some patterns:

- Client-server: The client handles the user input which gets sent to the object classifier where it interacts with the clarifai api and connects to the database.
- Singleton: to instantiate a database connection

Third-Party code vs. Original code Statement

- Clarifai Documentation: https://docs.clarifai.com/ used to learn how to use the API to classify objects in images
-

Statement on the OOAD process for your overall Semester Project

- One big element that started as a problem was designing a pattern based solution. We came up with our idea relatively quickly but we struggled to create and design based on the pattern approach. However, we were still able to implement patterns we just did not necessarily plan patterns and then details, we did that simultaneously.
- Another element that was key was the server class, or lack thereof. It was key in our design at the beginning to have this be a part of our project but in the process of implementing our ideas we realized the issue was merely that we made things too complex. We were so focused on implementing things in an object oriented way and through the OOAD process we failed to realize it just wasn't necessary to accomplish our goal.
- A smaller issue in the design process but bigger in the analysis and testing process and key to ensuring implementation was an issue with the api flask processing and debugging in different environments. It was resolved but it was time consuming and affected the team's ability to complete every part of our original design.