BookMarked

Ella Gage and Jules Geneser
https://github.com/jgeneser/CSCI-4448-5448-OOAD-project-6-gabgage-jgeneser

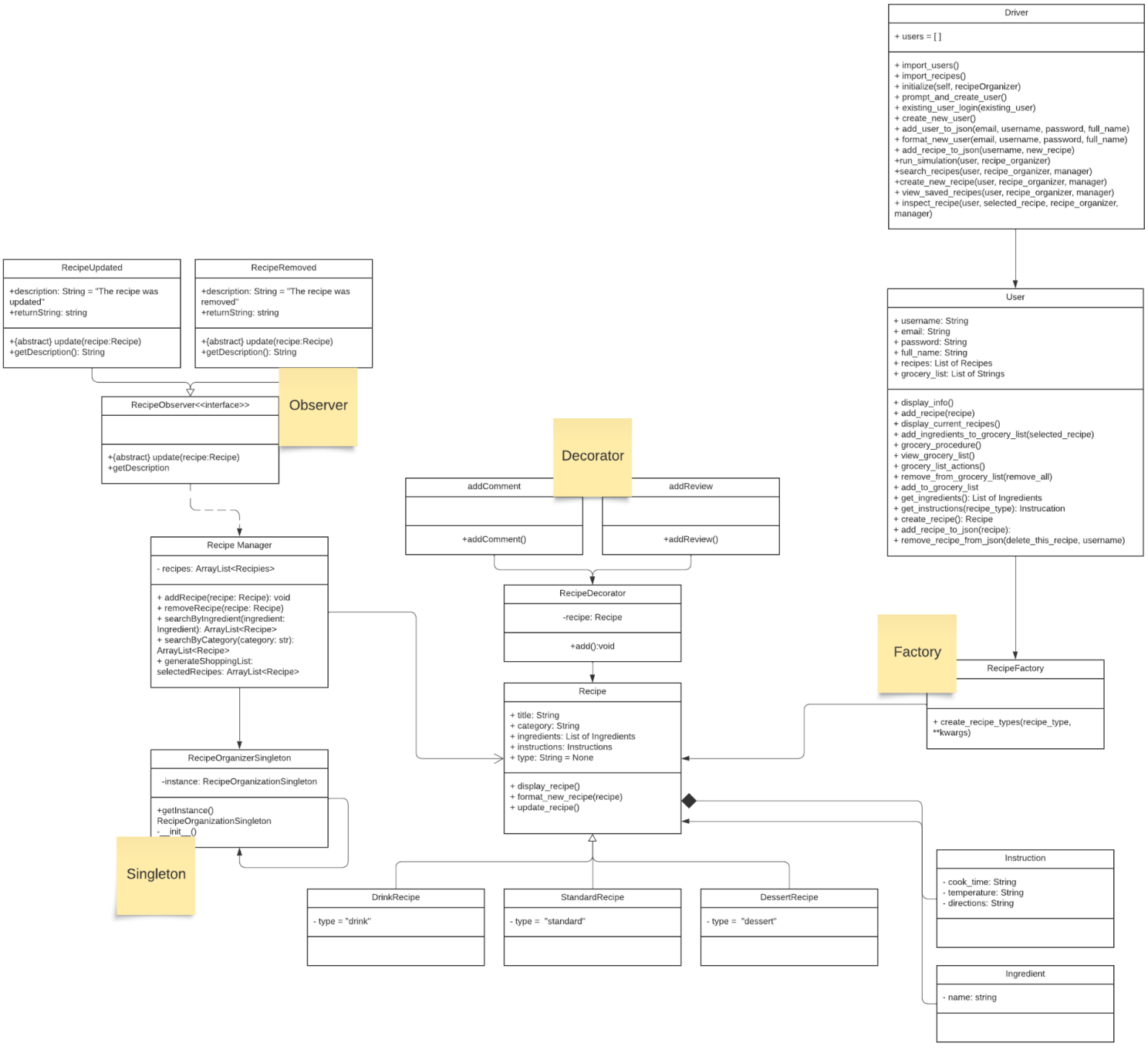
Final State of System Statement

The primary purpose of BookMarked remains the same as we originally intended. The final program's features reflect those of what we wanted to accomplish in project 5, those being able to save a recipe, view saved recipes, create recipes, add ingredients to a grocery list, and search for a specific recipe. The features that differ in our final product is the user is now able to delete, edit, and add comments or reviews to recipes. The main difference between project 5 and project 7 lies in the implementation of BookMarked. While we initially intended BookMarked to be a web service, it became more apparent to us that the scope of our plan in project 5 was too big for our project. To remedy this we changed directions so that BookMarked is now a terminal based program.

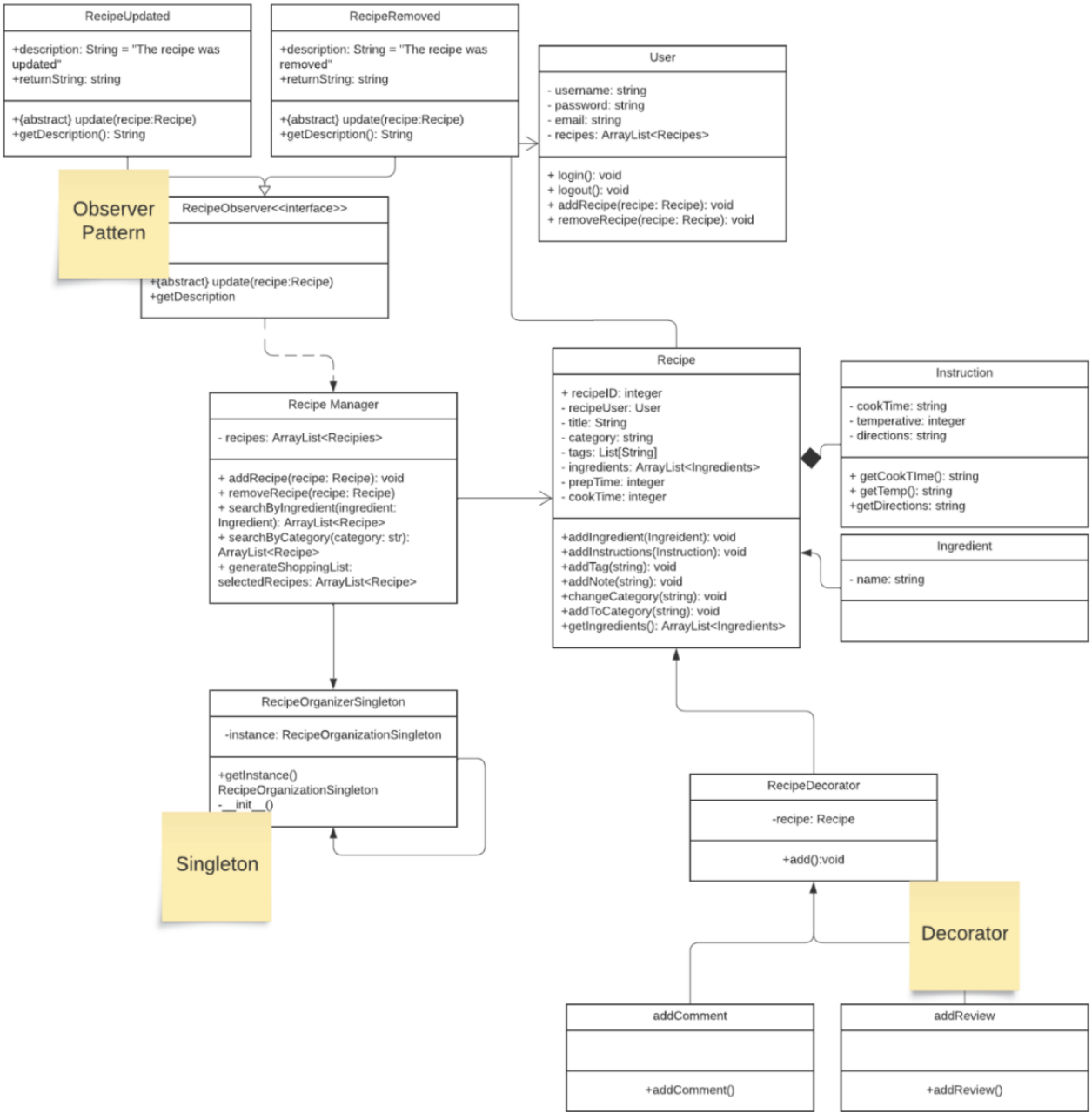
Our final project consists of the following parts:

- A UI framework for what we initially envisioned in project 5
- And a fully developed terminal based program to store cooking recipes

Final Class Diagram



Original UML Class Diagram



Comparison Statement

The primary difference between our final UML and our original UML lies in the implementation of the design patterns. We kept most of the patterns the same throughout our implementation and only slightly tweaked them to better fit our program's needs. We did end up adding in the factory design method into our code and UML to better help us differentiate between different recipe types. Overall, the UML didn't change drastically throughout our design process.

Another one of the large differences between our previous UML and the one that was developed in the final product was the usage of a Driver class. Initially we had all of the interaction in classes like the User and Recipe. In the final stages it was a necessity to have the Driver class, it really helped us with the user interaction and to get all of the cross over interactions into one place.

On the smaller side of things, we incorporated more methods than we initially thought the most notable were the ones having to do with the json interactions. We also add methods for searching (as this was an additional feature that we implemented that wasn't in the original plan).

Third-Party code vs. Original code Statement

For the creation of BookMarked, we used a Django framework which provided us with some base code for the UI side of the implementation. We also used and altered some public resources like that of codepen, and geeksforgeeks to implement the website's pages and styling, as well as Django documentation to help set up the framework. We also used sources from Tutorial points to help with pattern implementation and refreshers and Medium posts on how to work with json files (import, editing, deleting, etc).

Codepen:
https://codepen.io/paulomrcunha/pen/kdBqjw
https://codepen.io/colorlib/pen/rxddKy
https://codepen.io/nmekinci/pen/RwWYdEr

Tutorials Point:
https://www.tutorialspoint.com/design_pattern/factory_pattern.html

Medium:
https://medium.com/@KaranDahiya2000/modify-json-fields-using-python-1b2d88d16908

GeeksforGeeks:
https://www.w3schools.com/w3css/4/w3.css

Django:
https://docs.djangoproject.com/en/4.2/intro/tutorial03/#raising-a-404-error

Statement on the OOAD process for your overall Semester Project

Goal

Having a goal in mind was by far one of the design process elements that helped us the most during this project. Thanks to project 5 we had a plan of what the app was going to look and function like, which gave us a good foundation for starting to code the project. Without this plan I feel like the project would have felt much more daunting.

Plan

Like mentioned before, project 5 started us out with a plan for the project. With a UML diagram set in place before we even started coding, we always had a reference for how different aspects of the project were going to interact with each other, which was incredibly helpful. While we did end up changing the UML in the end, it acted as a really good reference point for what we still needed to accomplish.

Design

The designing of the project was both a benefit and an issue at the same time. Being able to create our own app allowed us to be creative with how we wanted different functionalities but also led us to one of our biggest drawbacks. With the scope of our project being slightly too big for the timeframe we were allotted combined with using new frameworks and languages, we weren't able to fully implement what we had wanted to do in project 5. Another drawback was the learning curve that we had when it came to Django. Although Jules had experience with this in her internship, it was a much more surface level interaction than we had initially thought.