

# Project Part 3

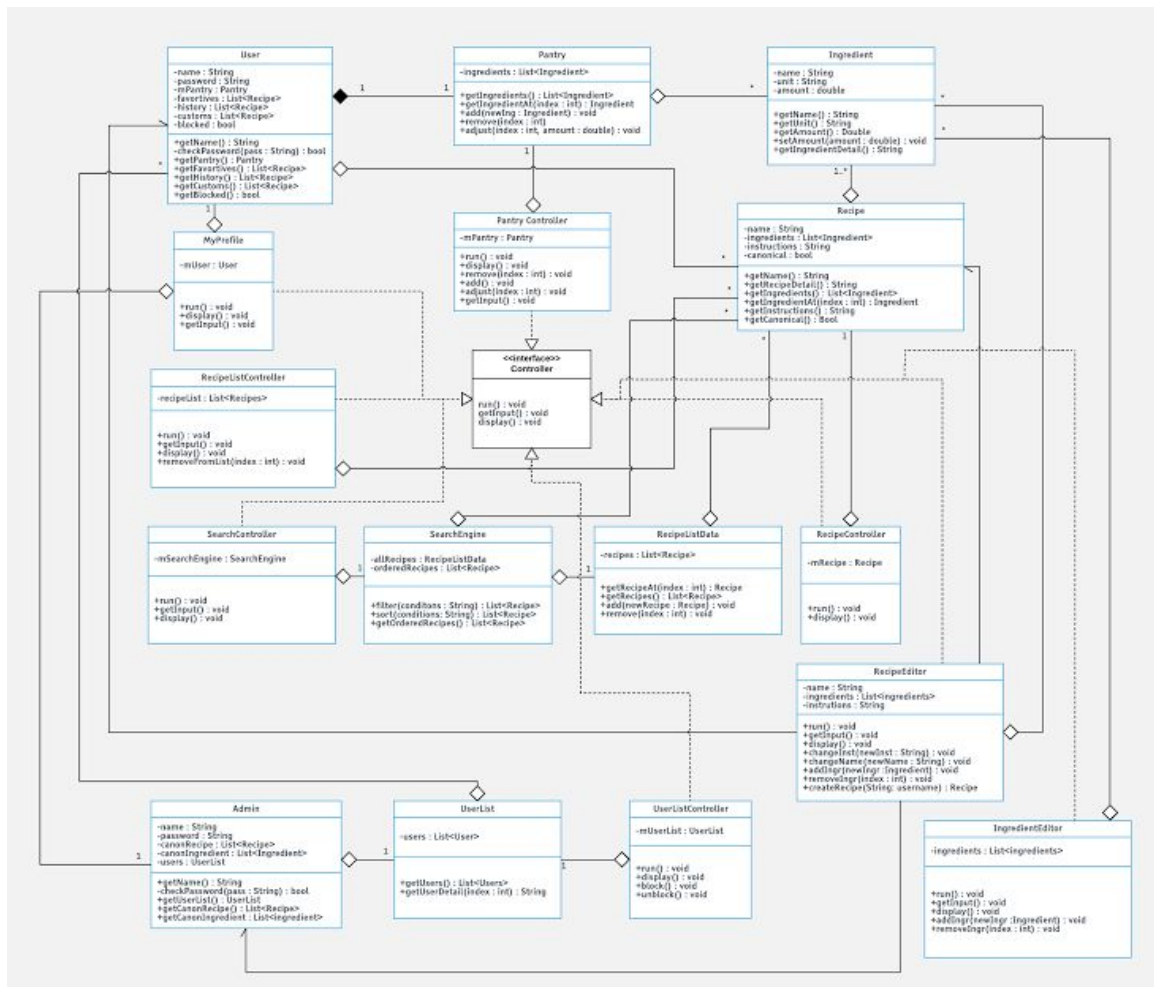
Team: Xiaolan Cai  
Chi Chen  
Andrew Guttman

Title: Ingredient Tracker

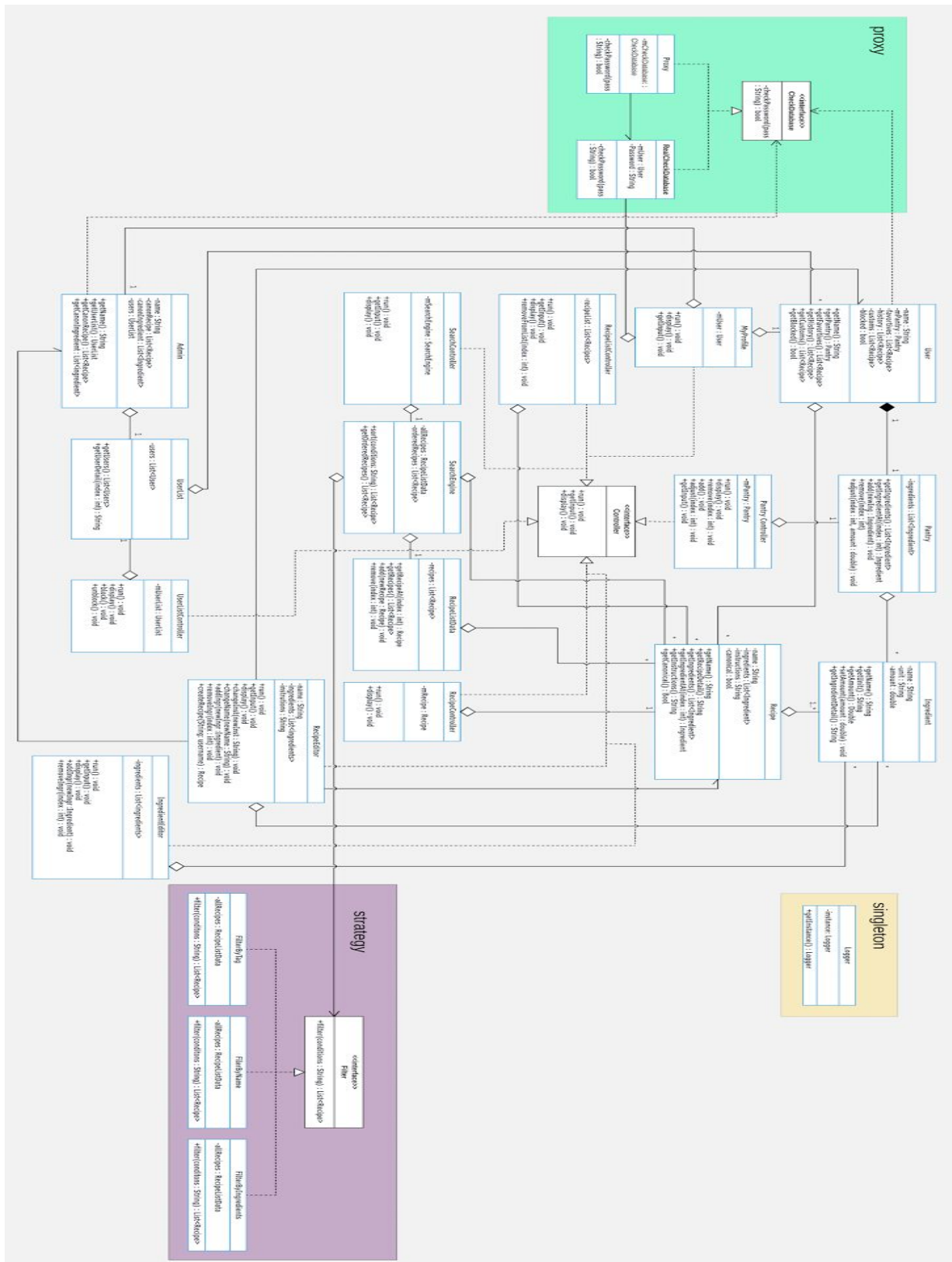
## Description:

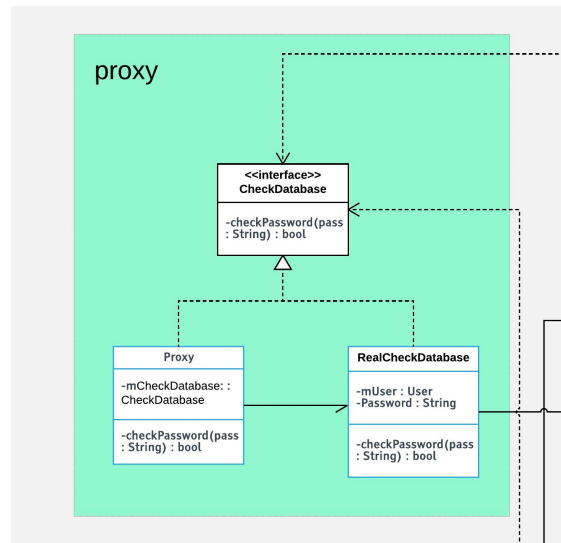
A web application that allows users to keep a digital pantry of ingredients, create and share recipes and search/filter/sort through recipes based on available ingredients and user metrics.

## Class Diagram Before:

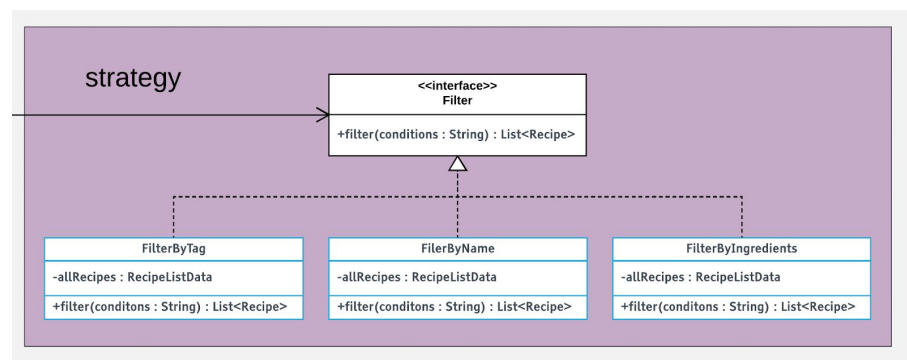


### Class Diagram After:





First, as you can see, we apply the Proxy pattern. We removed the `checkPassword()` method and password attribute from `User` and `Admin`, and implemented this function with a Proxy pattern and password stored in a database that gets its own class. The interface, “Proxy” class and “`RealCheckDatabase`” class are used to implement the Proxy pattern. The reason we think it might be better is that when checking the password, we access to the database instead of carrying around the password with each user, and this is exactly when we should use Proxy pattern to protect our system. The clients of the Proxy are “`User`” and “`Admin`”.



Another part we changed is applying the Strategy pattern to “`SearchEngine`”. We removed the `filter()` from “`SearchEngine`”, and then implement the different types of filtering that might occur with a Strategy pattern. Using a interface “`Filter`” and three classes for three ways implementing filter to realize the Strategy pattern. The reason we think it might be better is that we have three ways to filter the recipe, and Strategy pattern can help us here to classify three way for us.

## singleton

Logger
-instance: Logger
+getInstance() : Logger

In the end, we think we should have a logger to log all the action in our program, and if we have more than one user, the logs might not be clear do to interweaving. We want that one log should map to one and only one action. For this function, Singleton pattern will be the best pattern to apply. Using a class name "Logger" and a method in this class to implement Singleton pattern. We do not represent this on the Class diagram since it would get very messy, but Logger is used by every class that implements the "Controller" interface.