

Smart Bake

Nimra Sharnez & Khalid Aldawood

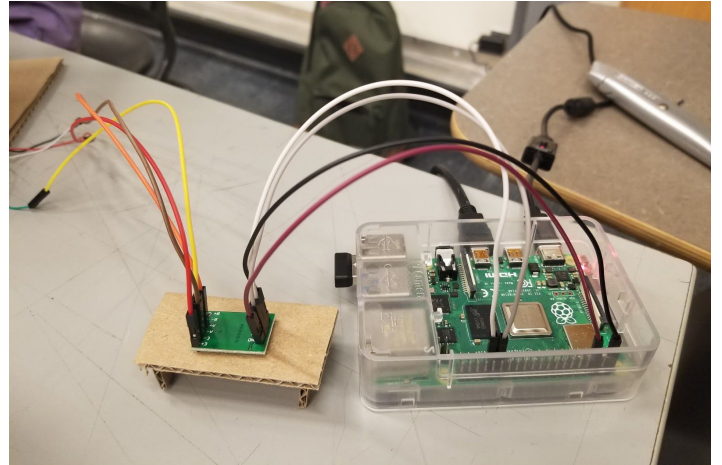
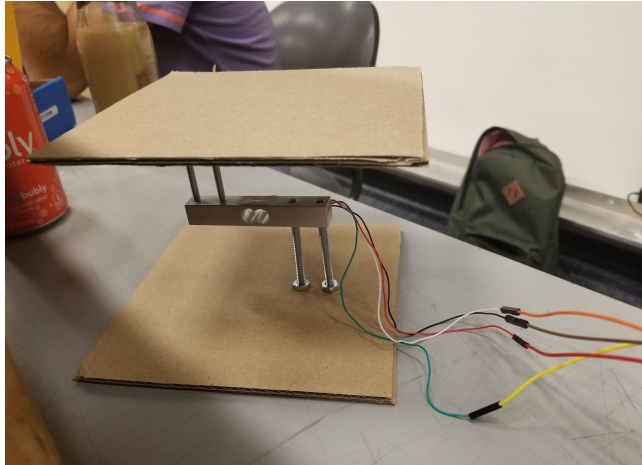
Final State of System Statement

Currently, our project has a functioning site with a recipe creation tab that turns inputted recipes into a SmartBake Software friendly format. In addition to a generated recipe.txt, users can view their inputted recipes on a personalized page in our Recipe Book. This can be found on our “Explore” tab.

In addition to the website, with SmartBake application, users can then access their uploaded recipes (as well as other recipes in the Recipe Book) and load the data into the SmartBake Software! With the fully functioning DIY scale built, the application will notify the user once the scale has measured the satisfying amounts (in grams) of each ingredient without the user ever needing to zero out the scale. This allows for the user to speed up their mixing process and use the same bowl for all their baking needs!

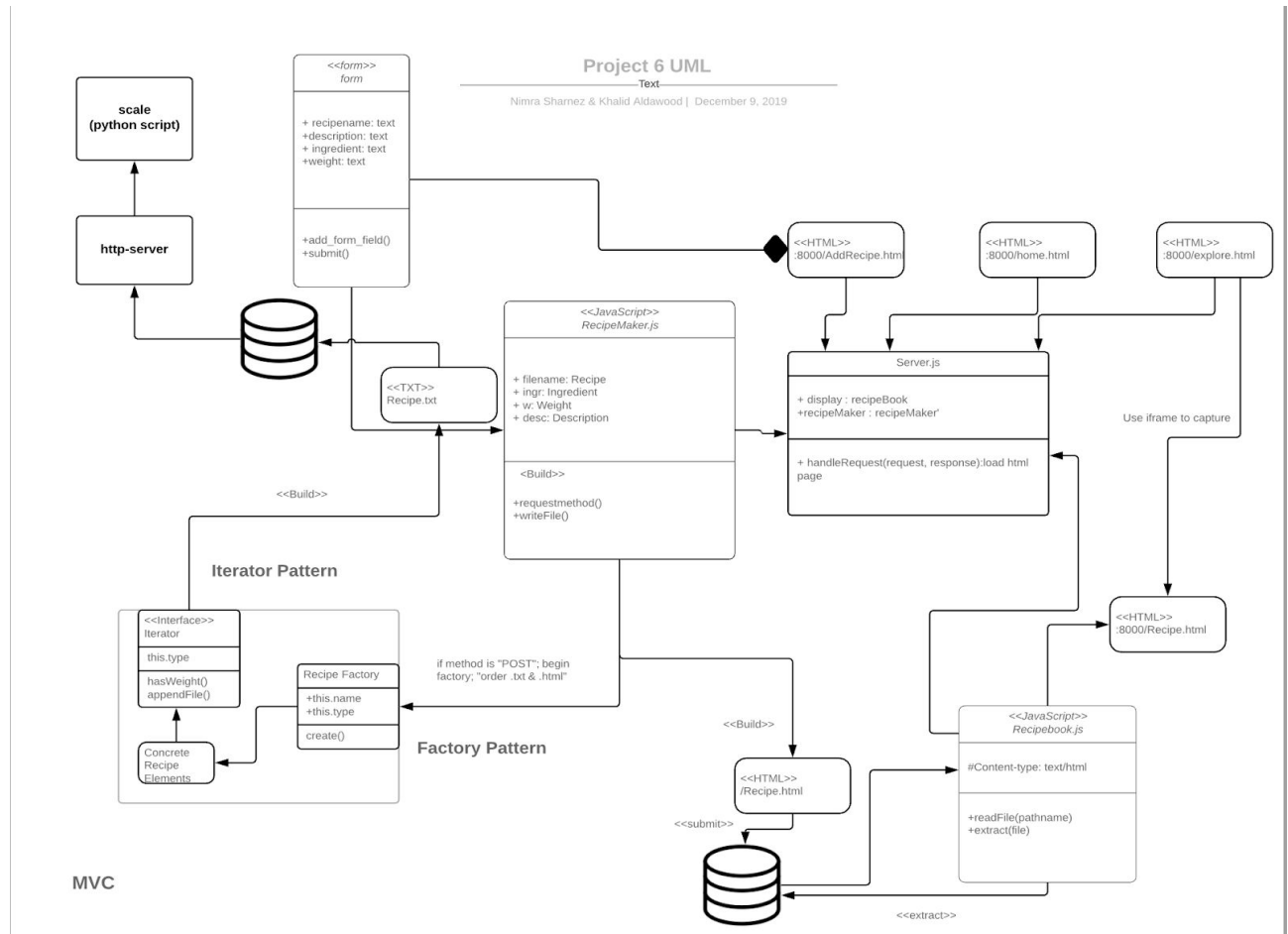
1. [View the website in use here!](#)
2. [View the application & the scale in use here!](#)

Features Implemented	Features To Add
<ul style="list-style-type: none">• User Recipe Creation• Viewable SmartScale Recipes• SmartBake Application<ul style="list-style-type: none">◦ Load in desired recipes and hook up to the scale◦ Notify users every step of the way in the mixing process• SmartBake Scale<ul style="list-style-type: none">◦ Scale that is hooked up to the application	<ul style="list-style-type: none">• Modifiable/(decoratable) Recipes & Serving Sizes• A progress bar showing percent of each ingredient

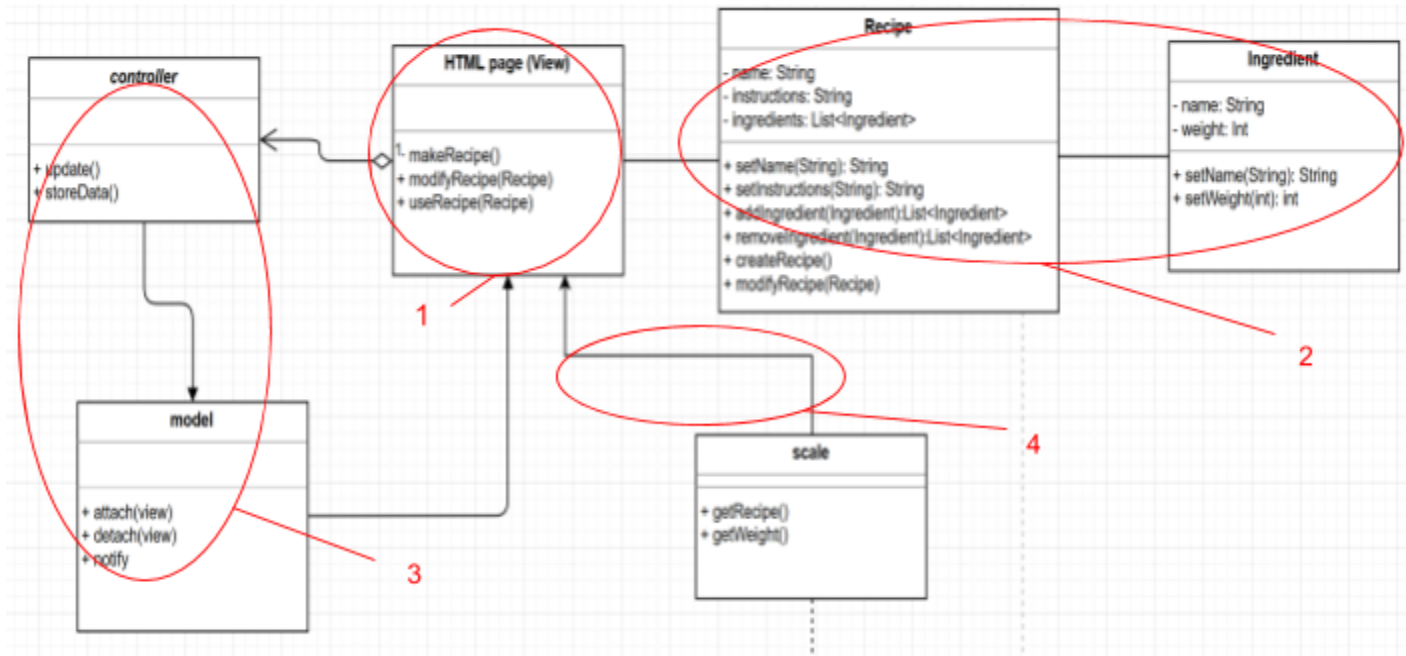


Final Class Diagram & Comparison Statement

[Final UML class Diagram](#)



Previous Class Diagram



1. Originally, we had not fully understood the complexities that would come along with the main view/ html page. We now have added a deeper level of complexities and changed our UML to show that our multiple html files are linked to the function holding .js file. This .js file is now what contains the classes, attributes and methods that get it's data from an html page's form (based on user inputted data). Once the user submits the form, the factory method then kicks in and creates concrete recipes. This is how we generate our .txt files for the SmartBake software to interpret.
2. Recipe was previously thought of as a class, but with further consideration, we have understood the user inputted info would better be served as objects. These objects would then help to generate the necessary files needed for the software.
3. The general idea did not change that much, more complexity and break down of classes happened, a factory and iterator were added and rather than having one server we have two, one for the website and one for the scale to fetch. Modifying the recipe was removed.
4. Previously, we were unsure how the scale would be connecting to the .txt files the website would be generating, so the direct connection was primarily made between the two. Understanding more after implementation, we now have

understood that it is through the server that the scale is able to use the method to getWeight(). From the file generated from the website and sent to the server, the scale can use the software to take the data necessary for measuring.

Third Party Code vs. Original Code

Third Party Code/ Example References	Original Code References
<ul style="list-style-type: none">• Creating html inside .js• Building Scale• Factory Pattern JavaScript (video)• HTML in Class Diagrams• CSS styling• Forms in html• RecipeStore.js• Hx711py (minus SmartScale.py)• tkinter	<ul style="list-style-type: none">• RecipeMaker.js• Server.js• RecipeBook.js• Addrecipe.html• Explore.html• Home.html• SmartScale.py

Statement on the OOAD process for your overall Semester Project

List three key design process elements or issues (positive or negative) that your team experienced in analysis and design of the OO semester project

1. Using a server makes getting data more difficult since we have to follow the internet protocols, and we have to constantly waiting for requests so server could thought of as a mediator .
2. Encapsulation made the server file readable and maintainable.
3. Factory pattern is suitable for making recipes.