**CMSC 12200 Final Project Proposal -- Food Butler**

**Project name:** A Weekly Food Menu Planner

**Project team:** Lisa Li, Xinyi Ge, Jiacheng Yan, Guchuan Song

**Goal:** This application helps students/working professionals plan the food menu for a week, or any other preferred periods, which could cater to customized needs indicated optionally. Examples include dietary restrictions, fitness schedule (calories), ingredients that the user already has and would like to use, dinning style and taste preferences, and budget constraints. This application also helps generate the list of ingredients that is needed for the planned week’s menu and the associated prices for budgeting purpose. To further help the users in the cooking process, we will include the recipes for each food course planned and possibly a timer as well. To optimize user experience, we may further address dishes beyond the traditional three meals which could possibly include morning breaks, afternoon tea, and midnight snacks/fourth meal, all under user’s choice. If time permits, this project will include additional features such as recommending liquor and music along with each meal.

**Potential sources of data:**

* Allrecipes: <http://allrecipes.com/>
* Recipe: [www. recipe.com/](http://www.recipe.com/)
* United States Department of Agriculture, Agricultural Research Service, National Nutrient Database for Standard Reference Release 28: <http://ndb.nal.usda.gov/ndb/foods>
* Peapod: <http://www.peapod.com/> (estimated cost of meals)
* List of cuisines: <https://en.wikipedia.org/wiki/List_of_cuisines>
* A food classification and description system developed by European Food Safety Authority (efsa): <http://www.efsa.europa.eu/en/data/data-standardisation>
* Two good recipe sources: <http://www.bbc.co.uk/food/cuisines> <http://cafeworld.wikia.com/wiki/List_of_Cuisines>

**List of work required (including but not limited to):**

1. Collect data
2. Process data into suitable database, including relevant tables for data storage
3. Complete basic functions that returns the menu
4. Complete functions that take in added parameters and APIs
5. Add in price estimator
6. Add in timer

**Planned working schedule**

Deadlines:

* Written Proposal (Due: Jan 26th at 5pm)
* (1) scraping websites and storing data into databases (end of 7th week)
* (2) algorithms and functions for putting together a menu (end of 8th week)
* (3) Django website structure ie. search engine, output/menu page, timer on recipe pages (end of 8th week)
* (4) putting results on Django site (end of 9th week)
* Completed Software (March 15th at 5pm)
* Final Project Presentations (10th week)

General Outline:

* Work on scraping data ( #1) and experimenting Django (#3): week 5, 6, 7
* Work on #2 with sample data and start building Django webpage (#3): week 6, 7, 8
* #4: week 9
* clean-up and minor adjustment to webpage: week 10
* clean up codes: post-presentation

**Possible new data structure/programming technology:**

1. interactive timer during cooking process
2. image (taken from source) that redirects you to the recip page if you click on it
3. Django designs
4. Capability of synchronizing into iCalendar or Google Calendar
5. The selecting regime used for generating/optimizing food menu, innovative regression method possible

========

Wednesday Presentation

1. the idea was derived from problems we met in everyday life:
2. Don’t know what to buy in the supermarket.
3. Want to purchase food ingredients efficiently and economically.
4. Don’t know how to do with some ingredients left in the fridge
5. Tired of thinking what to eat.
6. Addressing nutrition needs for fitness or disease

**Goal (above)**

**Potential Source (above)**

**Planned Schedule (above)**

**New tech (above)**

Inputs:

* Calories: 2 integers, lower and upper limits (range per day)
* things to consider but probably not going to use
  + (Protein: 2 integers, lower and upper limits)
  + (Sugar: 2 integers, lower and upper limits)
  + (Fat: 2 integers, lower and upper limits)
* Ingredients we have: drop down box
  + Choice of protein: beef, chicken, pork, lamb, egg, shrimp, fish, crab, lobster
  + Choice of greens:
  + Choice of Carb: rice, bread, pasta,
  + Others: milk, cheese, peanut,
* Ingredients to avoid
  + Choice of protein: beef, chicken, pork, lamb, egg, shrimp, fish, crab, lobster
  + Choice of greens:
  + Choice of Carb: rice, bread, pasta,
  + Included dietary restrictions: Dairy, Egg, Gluten, Peanut, Seafood, Sesame, Soy, Sulfite, Tree Nut, Wheat, Vegetarian, Vegan (may need a dictionary that maps each dietary restriction to a list of avoided ingredients)
* Price for the whole week: an integer representing upper limit
* Time to prepare and cook each meal: 2 integers representing maximum minutes for breakfast and lunch/dinner
* Number of people designed for: an integer (we must have more than one course per meal if more than 1 people)

sample\_input = {“calories”: [50, 500],

“ingredients\_have”: {“greens”: [“onion”, “tomato”], “protein”: [“lamb”]},

“ingredients\_avoid”: {“dietary\_res”: “vegetarian”},

“price”: 200,

“time”: [20, 60]

}

Output for backend:

An ordered list of meals (three per day), each meal represented by a tuple of three elements, one of which represents the default choice and the other two represent the alternative choices.

Outputs:

A calendar that has each meal listed on. Clicking on each meal will redirect the user to the corresponding website containing the recipe. Client can change each meal to alternative choice. Click a button to generate shopping list after finalizing the menu.

## All variables should be able to applied both generally and to a specific meal?

Algorithm:

* requirement
  + No repeated recipe in a whole week
  + Ceteris paribus, meal with fewer kinds of ingredients should be chosen
* divided into Breakfast (B) and Main Meal (MM) then filter by cooking time
* Filter B and MM by ingredients to avoid, , then divide recipes (preferable ingredients should be put into another list called prefered meals, and the rest called common meals)
* Filter by calories. Any meal with more than ½ of maximum calories per day should be abandoned. Any meal with less than ¼ of minimum calories per day should also be abandoned.
* Divide the maximum weekly price by 6 to get the maximum per day price. remove any dishes that have a cost of more than ¾ of maximum price per day
* By now, we have four lists: preferred breakfast meal, common breakfast meal, preferred MM, common MM
* From the preferred lists, using random number generator, get a breakfast and two MM for the first day. Check if the price and calories limits are met by this one-day arrangement. If not, generate another one using the same method until it meets requirement. Then delete the selected courses from the eligible list
* Generate 7 such one-day arrangement. If recipes from preferred list are not enough, select from the common lists. Put together to get the final calendar.

=======

base: calendar (3 meals), capable of choosing ingredients, shopping list