Food Butler

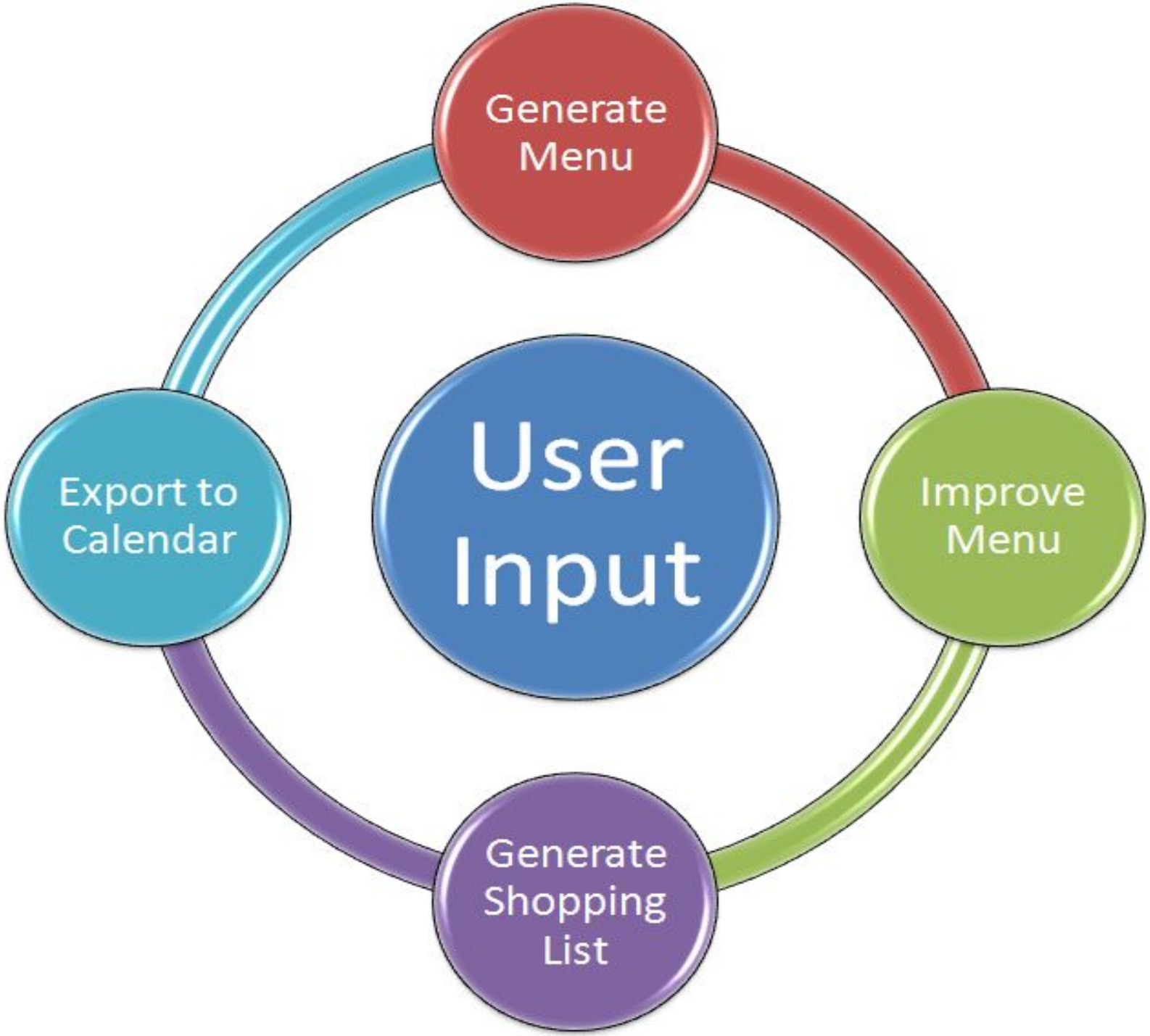
A Weekly Food Menu Planner
Just for **YOU**

Team: Gary Song, Lisa Li, Tony Yan, Xinyi Ge

From Last Time…

Daily Struggles

- Don’t know what to cook for today’s meals
- Don’t know what to buy in the supermarket
- Don’t know how to deal with ingredients left in the fridge
- Don’t know how to address nutritional needs for fitness or dietary restrictions


Generate Menu
Improve Menu
Generate Shopping List
Export to Calendar
User Input

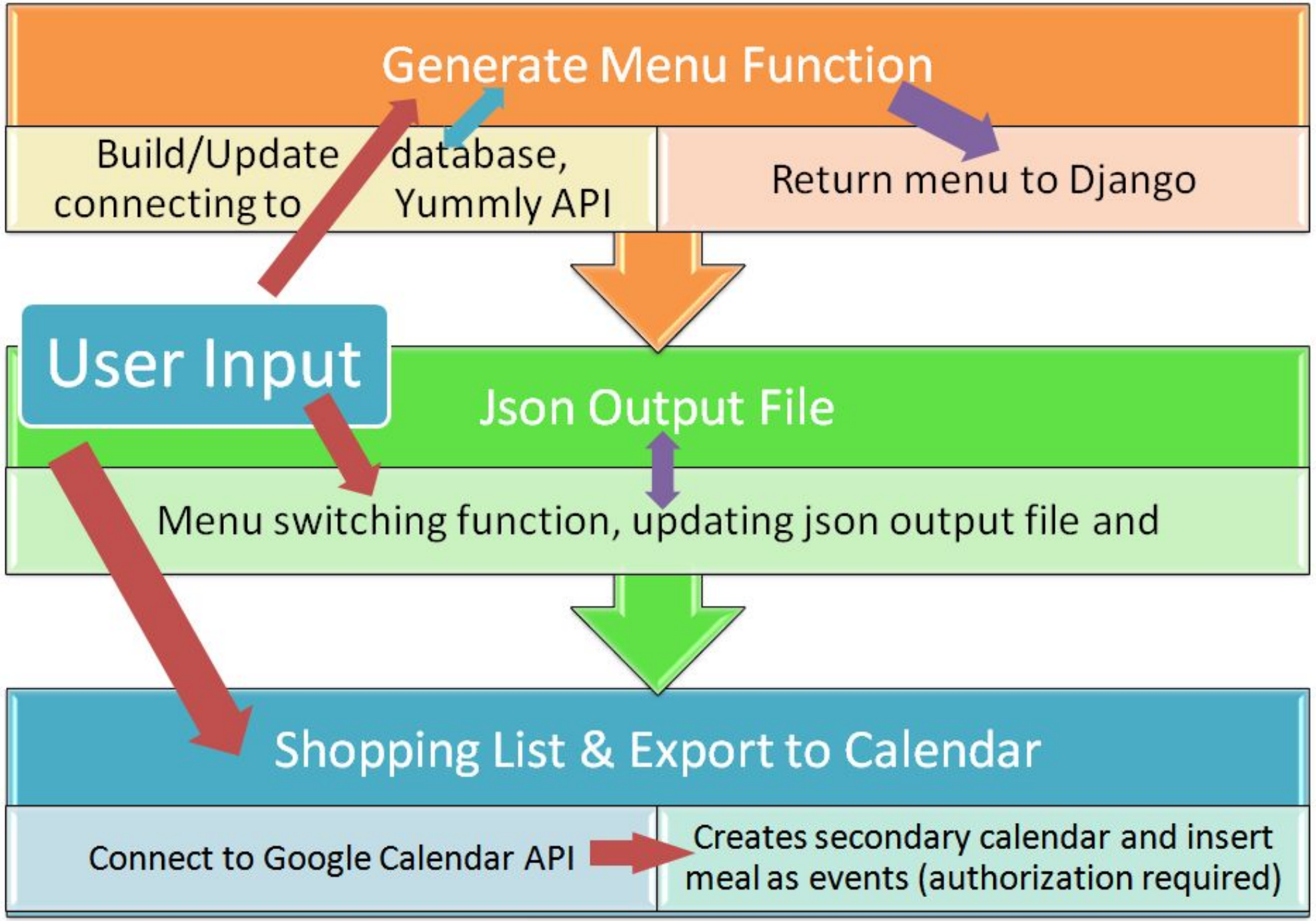
Demo

Original Goal	Status	Remark
Generate a 3-meal-per-day, 7-day food menu	Checked	Non-repetitive and tasty!
Provide alternative meal plan	Checked	Change and switch specific meals
Consider personal preferences, dietary restrictions and existing ingredients	Checked	Ingredients to include/exclude
Generate a shopping list	Checked	View shopping list for specified days

Original Goal	Status	Remark
Compute price and satisfy budget constraint	Omitted	Restriction of data
Show cooking instructions	Checked	Click images to redirect to instruction page
Display an interactive Cooking Timer	Omitted	Data inaccuracy, easily substitutable
Plan afternoon tea and midnight snacks as well	Omitted	Healthier diet
Synchronize to Google Calendar	Checked	Easy upload, easy delete

Additional features we included:

- Major ingredients do not repeat within three days if enough recipes qualified
- Cooking time variable could be set for breakfast and main meal, separately


Generate Menu Function
Build/Update database, connecting to Yummly API
Return menu to Django
User Input
Json Output File
Menu switching function, updating json output file and
Shopping List & Export to Calendar
Connect to Google Calendar API
Creates secondary calendar and insert meal as events (authorization required)

User Input

Django form → Dictionary /List → Call own functions → Post results on Django webpage

1. Generate Menu

Send request to Yummly API for list of receipes

Send request to Yummly API for informatio n about individual receipe

Update database

Generate menu using algorithm

Output in a Json file

2. User Improves Menu

Django form

Update Json file

Reflect new changes in menu on website

3. Generate Shopping List

Django form

Information from Json file

Generate shopping list on separate webpage

4. Export to Calendar

Django form user inputs

Connect to Google Calendar API

Creates secondary calendar (requires authorization)

Insert meal as events (required authorization)

Lamb Salad

Cooking Time:

Calories: **130.0**

Lamb And Goat Cheese Stuffed Eggplant

Cooking Time:

Calories: **620.0**

Lamb and Asparagus Salad with Hummus Dressing

Cooking Time:

Calories: **210.0**

Slow Cooker Greek Gyros

Cooking Time:

Calories: **150.0**

Spiced Lamb with Corn and Chickpea Salad

Cooking Time:

Calories: **290.0**

Spiced Lamb Burgers with Herbed Yogurt

Cooking Time:

Calories: **390.0**

Greek Lamb Kabobs

Cooking Time:

Calories: **280.0**

Moussaka

Cooking Time:

Calories: **340.0**

Spiced Lamb Ragu Shells

Cooking Time:

Calories: **230.0**

Kheema With Peas

Cooking Time:

Calories: **450.0**

Lamb Jalfrezi

Cooking Time:

Calories: **190.0**

Tagliatelle with Quick Lamb Sugo

Cooking Time:

Calories: **240.0**

Osso Bucco Lamb Curry

Cooking Time:

Calories: **360.0**

Braised Lamb Shanks with Lemon Confit

Cooking Time:

Calories: **510.0**

Lamb Exohiko (Lamb, Spinach and Cheese Stuffed Phyllo Parcels)

Cooking Time:

Calories: **270.0**

Slimming World's lamb tagine

Cooking Time:

Calories: **240.0**

Baked Lamb and Apricot Casserole

Cooking Time:

Calories: **540.0**

Arayes

Cooking Time:

Calories: **470.0**

Slow-Cooker Braised Lamb Shanks

Cooking Time:

Calories: **540.0**

Minty Lamb Koftas

Cooking Time:

Calories: **190.0**

Doner Kebab

Cooking Time:

Calories: **280.0**

Challenges We Met

- Data quality (calories, cooking time, ingredient lines, etc)
- Shopping list consolidation -- string containing fraction numbers
- Synchronize to calendar across months (datetime)
- “Major ingredients” definition

- Python2
- Using various APIs
- Passing calls to API vs. generating our own database (data storage)
- Optimal parameters in algorithm

¼ C strawberries, chopped	1
1 tablespoon brown sugar	1
1 TB taco seasoning	1
3 tbsp butter, melted and cooled	1
1/2 TB olive oil	1
1/2 cup quinoa	1
1 small onion	1
1 7oz can diced green chili	1
1 tsp vanilla extract	1
1/2 TB cumin	1
1 boneless chicken breast	1
2 eggs, lightly beaten	1
⅔ C strawberries, finely chopped	1
1 14 oz can fire roasted tomatoes plain is fine too	1

it)

Honey (optional for serving)	1
200 ml plain yogurt	1
1 medium yellow onion, sliced into strips	1
1 lb ground beef – organic and grass-fed preferably	1
5 huge red Dates	1
4 sub rolls, sliced in half (but not all the way through)	1
1 serving pistachios (around 50 pistachios)	1
Salt and pepper, to taste	1
2 cups shredded provolone cheese	1
1 Tablespoon butter	1
1 tsp salt or onion salt	1

Some Parameters in algorithm

MAX_TRIAL_BEFORE_GOING_TO_ALT = 5

MAX_TRIAL_BEFORE_REPEATING_INGREDIENT = 10

MAX_TRIAL_BEFORE_IGNORE_CALORIES = 20

BREAKFAST_CALORIES_WEIGHT = 0.5

LUNCH_CALORIES_WEIGHT = 0.6

DINNER_CALORIES_WEIGHT = 0.6

New Technologies Summary

- Python2 Programming Language
- Running shell script (to call python2 program) in a Python3 program
- Interacting with website using APIs (Yummly, Google Calendar), some with authorizations
  - Caching
  - Datetime library
- Django, html, jquery (toggle)

These are interesting…

```python
# Helper python program to help initiate
# the syncal.py program

from subprocess import call
import json

def sync(time_dict):
    '''
    helper function to call python2 syncal.py
    '''

    with open("time_dict.json", "w") as f:
        f.write(json.dumps(time_dict))

    call("python2 syncal.py", shell=True)
```

```python
def generate_available_recipes(args_from_ui):
    '''
    Generate available recipe lists, calling build_db.py
    '''

    with open("temp_dict.json", "w") as f:
        f.write(json.dumps(args_from_ui))

    call("python2 build_db.py temp_dict.json", shell=True)

    with open("recipe_lists.json") as f:
        recipe_lists = json.load(f)

    return recipe_lists
```

Your Stomach Thanks You!