LIFE Food & Exercise External Developer Guide

Author: Vincent Lin
8/4/2020

LIFE (Learning Important Factual Equivalents) Food & Exercise is a **POC (proof of concept)** app that allows the user to select a specific **fast food entree/item** and **compare** it to up to **three physical activities**. A user can select items from several popular fast food franchises (such as Arbys, Burger King, Dairy Queen, In and Out, McDonalds, and Wendys). The physical activities number almost 300, and include everything from aerobics to whitewater sports, and includes everything in between (bowling, carpentry, golf, fencing, painting a wall, and tai chi to name a few). The program then **calculates the number of hours and minutes (and in some cases days) required to burn the equivalent calories**, and shows the data in a “Meme Card” that is meant for education and entertainment (sharing for fun).

The bulk of the proof of concept functions were implemented with two main functions and several nice-to-have functions omitted in order to complete the project in the time allotted.

These include:

1. Sharing the Meme Card through a social media platform by clipping or exporting the Meme Card
2. Providing some kind of user management with the saving of preferences or previously generated Meme Cards.
3. Implementation of the proof of concept as a Web App hosted on a website.
4. Scraping functions to pull food item images and/or calorie values directly from restaurant websites

USERS

The intended users of the project are generally for anyone, but because of the addition of a single “racy” activity, should be considered PG-13. Generally, any user who is interested in learning what the activity equivalent is for common food items for health or entertainment purposes might find this POC useful and/or fun.

PRIMARY TASK/PROBLEM

Most people don’t know what calories mean in practical terms. This POC helps provide a **meaningful equivalency in minutes, hours or days of a selected physical activity** of specific foods. As a practical measure, the information provided might help them make better choices when selecting their meals.

WORKFLOW
The design of the workflow changed early on in development as the user and database management aspects were deemed more complicated than necessary for this POC. As a consequence, the workflow was simplified to include the basic steps necessary to input the desired data, as well as the final output of the processed results.

The final work flow is as follows:

1. User starts the POC application.
2. **Main Screen:** User is greeted by a simple Splash screen with the application title and high level value prop. User can either navigate to the next step from the menu bar, or by clicking on "Start Now"
3. **1. Select Activity:** User is prompted to select up to three physical activities. S/he can scroll through a list of almost 300 activities, or type a string into a search field which will automatically filter the list. User can either: 1. Move on to the next task, 2. Select multiple activities (the last 3 of which are retained), or 3. Clear the list of activities selected.
4. **2. Set Weight:** User is prompted to enter weight, and given the option to select if the units are in pounds or kilograms. The default is set in preferences and is 150 pounds. User is able to move on to the next task.
5. **3. Select Food Item:** User is prompted to select a restaurant, and a single menu item that appears in a cascade menu based on the selection. This step has a built-in error control and detects whether the conditions necessary to display a result are met - that is a physical activity, weight, and food item must have been designated. An error message is displayed if one or all of the conditions are not met. Once those conditions are met the user is able to move on to the next task.
6. **4. Review Result Summary:** The User is shown the summary of the data entered, as well as the resulting equivalent values. The User is able to move to the next step.
7. **Show Meme Cards:** The User is shown a rendered Meme Card. The Meme Card consists of an image of the fast food item (most of which are on a white or transparent background). The image of the food item is composited onto a white background card. Several text phrases are rendered on top of the card to produce the Meme Card. The phrases are dependent on various thresholds.

   The minimum Meme Cards generated is one, and the maximum is 3, depending on how many physical activities were chosen for comparison. The current set of Meme Cards are automatically saved into the root directory of the program, but are overwritten on each run of the program.

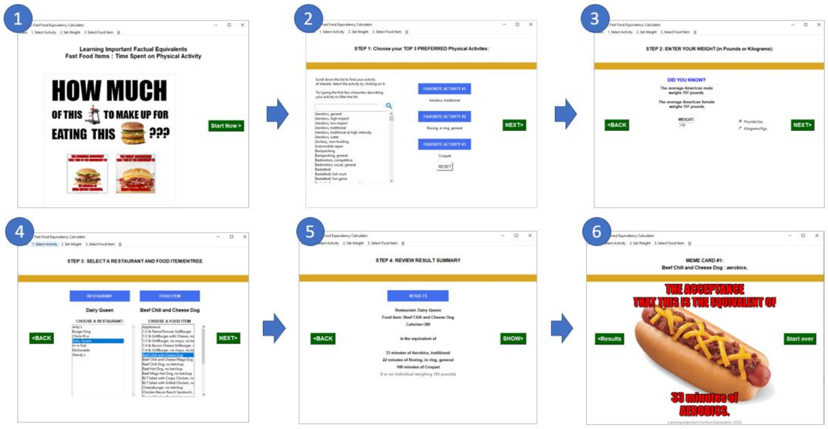
PROGRAM STRUCTURE AND INTERACTION LOOP

main.py initiates the primary loop
|
tkinter main.py calls tkinter directly from Python
|
app_def.py main.py calls app_def.py - main application functions
|
preferences.py app_def.py calls preferences - program defaults and settings
|
lookup.py app_def.py calls lookup.py - time string conversion and meme card render
|
|
account_class.py app_def.py calls account_class.py - data class storing primary variables

Primary Program flow is executed through a sequence of functions with app_def.py

	PRIMARY STEP	FUNCTION
1.	Main Screen	switch_to_main
2.	Select activity	switch_to_setexpref
3.	Specify weight	switch_to_weight
4.	Select restaurant/food item	switch_to_findfood
5.	View results	switch_to_result
6.	View Meme Cards	show_memes

The program also contains a number of sub-functions designed to handle key-press events, data import, error handling, and other utility steps. More information about each function is available in the docstrings of each function.



1. Learning Important Factual Equivalents
Fast Food Items : Time Spent on Physical Activity

2. STEP 1: Choose your TOP 3 PREFERRED Physical Activites:
Scroll down the list to find your activity of interest. Select the activity by clicking on it.
Try typing the first few characters describing your activity to filter the list.
Aerobics, general
Aerobics, high impact
Aerobics, low impact
Aerobics, traditional
Aerobics, traditional at high intensity
Aerobics, water
Archery, non-hunting
Automobile repair
Backpacking
Backpacking, general
Badminton, competitive
Badminton, social, general
Basketball
Basketball, full court
Basketball, fun game
FAVORITE ACTIVITY #1: Aerobics, traditional
FAVORITE ACTIVITY #2: Boxing, in ring, general
FAVORITE ACTIVITY #3: Croquet
RESET

3. STEP 2: ENTER YOUR WEIGHT (in Pounds or Kilograms):
DID YOU KNOW?
The average American male weighs 197 pounds.
The average American female weighs 157 pounds.
WEIGHT: 150
● Pounds/Lbs.
○ Kilograms/Kgs.

4. STEP 3: SELECT A RESTAURANT AND FOOD ITEM/ENTREE
RESTAURANT: Dairy Queen
FOOD ITEM: Beef Chili and Cheese Dog
CHOOSE A RESTAURANT:
Arby's
Burger King
Chick-fil-a
Dairy Queen
In-n-Out
McDonalds
Wendy's
CHOOSE A FOOD ITEM:
Applesauce
1/2 lb FlameThrower GrillBurger
1/2 lb GrillBurger with Cheese, no
1/2 lb GrillBurger, no mayo, no ke
1/4 lb Bacon Cheese Grillburger,
1/4 lb GrillBurger, no mayo, no ke
Beef Chili and Cheese Dog
Beef Chili and Cheese Mega Dog
Beef Chili Dog, no ketchup
Beef Hot Dog, no ketchup
Beef Mega Hot Dog, no ketchup
BLT Salad with Crispy Chicken, no
BLT Salad with Grilled Chicken, no
Cheeseburger, no ketchup
Chicken Bacon Ranch Sandwich,

5. STEP 4: REVIEW RESULT SUMMARY
Restaurant: Dairy Queen
Food Item: Beef Chili and Cheese Dog
Calories=380
is the equivalent of
33 minutes of Aerobics, traditional
22 minutes of Boxing, in ring, general
108 minutes of Croquet
(For an individual weighing 185 pounds)

6. MEME CARD #1:
Beef Chili and Cheese Dog : aerobics.
Learning Important Factual Equivalents, 2020

DATA

Data used within this program were culled from various sources. The primary data tables are stored as CSVs and are imported into the program as DictReader objects, and further converted to nested dictionaries or a dictionary with tuples as the returned value. Modification to the data can be done directly to the CSV, but must maintain the column structure, and must be comma delimited. Conversion metrics and calorie values in the burn rate table and food calorie table respectively must be a positive, real number. There is no error handling for non numerical values.

Images used for the food items should be in PNG format. Although the Pillow renderer should resize the image to a size suited for the Meme Card, generally, it is suggested that the size not exceed 500x500.

Food calorie and image table
    http://fastfoodmacros.com/
Calorie exercise burn rate table
    https://exceltemplate.net/weight/calorie-tracker/
Food images
    Sourced from individual restaurant franchise websites
    Arby's, Burger King, Chick-Fil-A, Dairy Queen, McDonald's, Wendy's

Install/deployment/admin issues:

The program is written and tested under Python 3.7. Tkinter functions is part of the standard Python interface.

Run:

```
python -m tkinter
```

If Python and tkinter is correctly installed, a tk window will open displaying the current version number of tkinter installed.

Pillow (a PIL fork) version 7.0.0 is used extensively to render the Meme Cards. A working install of Pillow is required. For more information on installing Pillow on your Python install check here.

Known Issues:

The following issues are replicated in the readme.md file:

1. **MOST OF THE POTENTIAL ERRORS ARE TRAPPED AT STEP 3**: The majority of possible error conditions have been accounted for and a check is made at Step 3. You will be unable to see the Results or MemeCards without correctly 1. Selecting at least one activity, 2. Specifying a weight between 25 - 250 kgs or 50 and 500 lbs, and 3. Selecting a restaurant and food item.
2. **TEXT FORMATTING OF FOOD ITEM IN STEP 3**: There was insufficient time to fix the formatting of the text in Step 3. for the selected Food Items. In several cases, the selected food item string is too long, and is truncated at the front and the back. This has no effect on the ability of the program to show the correct equivalency values and to display the Meme Card.
3. **MEME CARD TEXT TRUNCATION**: There may be some iterations of the Meme Cards where the text strings are insufficiently formatted, and are also truncated in rendering.
4. **BACKGROUND CONTRAST**: There may be some backgrounds where the contrast between the text and the background clash badly. A possible future feature would be to allow the user to change the color of the text and shadow/outline.
5. **IMAGE CENTERING/PLACEMENT**: Because of the diverse sources of the food images, some of the items may not be correctly centered and may be too low or too high. In some cases the images may be too small or of insufficient resolution.
6. **DISCLAIMER**: The program does not include text that gives more accurate advice regarding the average number of calories burned in a daily wakeful activity (e.g. - walking, breathing, sitting still). The equivalent values given assume that any other net calories in or out are zero, and that the food item is in excess beyond the required calories an average individual needs to maintain homeostasis. In addition, the program does not specify that the burn rate is based on the average calories burned using a heart rate monitor during specific activities, assuming specific weights of individuals. Genetic variation as well as the exact amount of exertion in each activity may vary greatly. The description of the amount of exertion is also fairly subjective. As a result, the results given here are for directional use only, and not to be used as medical advice.

Future work and Ongoing Development

The specifications for this Proof of Concept was originally written as a webapp that could be hosted on a site for anyone to access. Due to time constraints and scope reduction those features were dropped.

A potential Roadmap for continued improvement of this project include:

1. Setting up the POC to select and save Meme Cards by specifying location and file name using a file explorer interface.
2. Connecting the application to a social media API to share Meme Cards and commentary text - such as through Instagram, Twitter, and/or Facebook.
3. Transforming the program into a web app that can be hosted on a web server using a Python web framework such as Django or Flask.
4. Adding functions that would allow users to add additional food items, caloric values, and images to the database.
7. Additional optimization of data structures and modules.

THANKS

Special thanks to Professor Chris Harding, Iowa State University for his guidance and troubleshooting on the project.

CONTACT INFORMATION

For questions, comments regarding this project, feel free to reach out to me:

Vincent Lin
vincent.joshua.lin@gmail.com