# CZ1003/CE1003

## Introduction to Computational Thinking

## Telegram Bot Project Report

## Team name: UberRun

Date of submission: 31st March 2017

## Members:

Tay Su Wen (U1621173K)

Srushti Sunder (U1622844H)

Bak Zhee Shuen (U1622696K)

Cheryl Lim Yu Jun (U1540348C)

Saldanha Karen Rachel (U1622767A)

**Problem Statement**

One of the marvels of “wearable” technology, from smartwatches to fitness trackers, lies in its potential to motivate users to make choices that are beneficial to their health. Fitness trackers in particular, such as the Fitbit and the Nike+fuelband, have been gaining ground among athletes and non-athletes alike. Fitness trackers appeal to people of all levels of physical activity through the two important facets of activity: measurement and motivation. (Beckham, 2012)

By obtaining an objective measure of activity through fitness trackers, people are equipped to judge their level of physical activity by gauging how many calories they burn through exercise. Users experience an increase in motivation to become and remain active as they can monitor their progress and exercise accordingly by being aware of their level of fitness through the visualisation of data.

An issue our team has identified however, is that fitness trackers or smartwatches are not calibrated to calculate the distance and propose a running route based on a user-specified amount of calories to burn. Fitness trackers and smartwatches do not include functions that provide exercise data prior to a workout. The function of our Telegram bot, UberRun, is to create a route for users based on the amount of calories they wish to burn by running/walking.

People intending to run/walk to burn a certain amount of calories often lack the following information:

1. The required distance to run/walk
2. The route to take to complete the required distance based on their starting location

A person may be deterred from exercising as the likelihood that they may abandon their efforts to engage in physical activity (in the form of running/walking) is higher if they do not know what goals to set due to the lack of the aforementioned information. By visualising data such as calories and distance, users can set measurable goals beforehand. They will hence be more motivated to keep up their level of fitness and make progress.

**Proposed Solution** Data and Online Source

Upon starting the Telegram bot, the user is asked to select running or walking and input the amount of calories they wish to burn, their weight in kilograms and the postal code of their starting location before the distance to be covered, route map and directions are sent to the user.

The computation of the distance to run/walk is done by the Telegram bot, while the route is mapped by a third party website ([www.routeloops.com](http://www.routeloops.com)). The directions computed by [www.routeloops.com](http://www.routeloops.com) will be extracted and sent to the user, followed by the image of the running route which will be obtained through a screenshot of the map on the [www.routeloops.com](http://www.routeloops.com) .

UberRun Program Flowchart and Discussion

Our team has created a Telegram bot, UberRun, which interacts with users who would like to base their run/walk on the amount of calories they want to burn. UberRun is programmed using Python. The functions of UberRun are:

1. Calculate a distance to run/walk based on user-input calories
2. Provide the user with a map image and directions of the most suitable route to fulfil the calculated running distance based on the postal code of their starting location

The flowchart of the program is as follows:

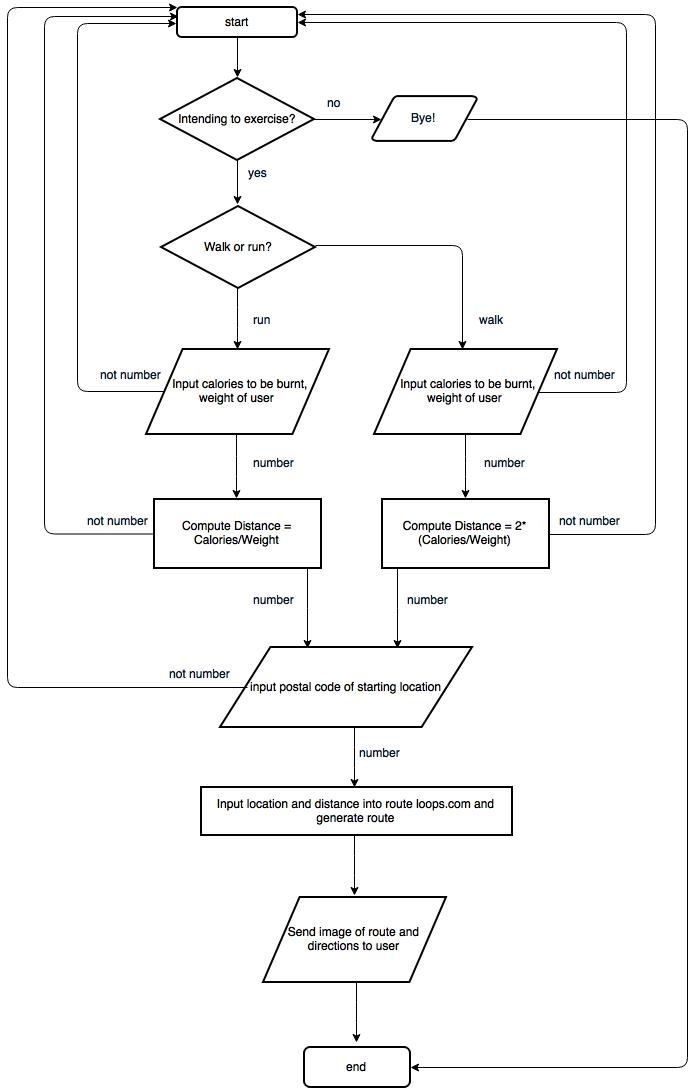
****

Fig. 1: UberRun program flowchart

UberRun Program Flowchart and Discussion (continued)

The UberRun Bot is initialised */start* is keyed in.

The user is asked questions (“Intending to exercise?” & “Walk or Run?”) and their response input is sent through a custom keyboard. The custom keyboard was created through the *BOT* *API ReplyKeyboardMarkup* and .*sendMessage* method. The user chat\_id is extracted so messages can be delivered to them through the bot using the telepot *module* and *.glance* method.

The user is then required to key in an integer when asked for the calories to burn and their weight. If a non-number is entered, the bot informs the user that the input cannot be processed and prompts the user to enter a valid input for the question.

The distance to run/walk will be calculated using the formula defined in custom function *distancecalc*. The input data from the user, regarding their option to run/walk to calories, weight and their postal address, is extracted via the custom function: *communicator*

Through the python module, *selenium,* the UberRun python code opens the Firefox web browser on the server and directs it to routeloops.com where the starting location (postal code) and distance calculated are entered to generate the route.

There is a possibility that the actual distance of the route generated by [www.routeloops.com](http://www.routeloops.com) may differ from the distance calculated by the bot. Thus, the team has included a function in the bot, *test\_route\_generation,* to check whether the distance generated falls within an acceptable range, defined by a custom function in the bot, *adjustment\_float.*

Once an suitable route is generated, the directions are extracted from the web source, using *selenium* module. The directions are sent as a message through the Telegram bot to the user. In order to aid the user in navigating the route, in addition to the directions, a screenshot of the route as displayed on [www.routeloops.com](http://www.routeloops.com) is extracted via the pillow module and sent to the user.

UberRun in Telegram

|  |  |
| --- | --- |
| IMG_9003.PNG  IMG_9007.PNG  Fig. 2.1   1. Initialisation of UberRun bot. Question sent with custom response keyboard to user.   User is able to choose to run/walk/decline exercise. | IMG_9003.PNG  Fig. 2.2   1. The UberRun bot requests user input for number calories to be burnt, weight in kg and postal code of starting location. |
| IMG_9003.PNG  Fig. 2.3   1. The distance to burn the user-specified amount of calories is calculated using *distancecalc* function and is sent to the user as a text.   Generic formula used for ‘run’  Computed distance = Calories/weight  Generic formula used for ‘walk’  Computed distance = 2\*(Calories/weight) | IMG_9004.PNG    Fig. 2.4   1. Sample of directions of running/walking route based on inputs in steps 2 & 3. It is sent as a text message to user.   https://lh4.googleusercontent.com/JRndwJnKTQMx9tOCTFvB3oDYTcx_P-1dJ-GTvwimiILbkfnBsng2_zlWrNeUM4GXS46qBxWftaV8qe2OaHJXfp-Ulx-6zGWaukjUs8lGMj29SA5BsZ0fEdlo4udVMLNLwEtEdShR  Fig.2.5  5. The UberRun Bot sends the user an image of the route to be taken followed by a final greeting. The image can be enlarged upon tapping on it and saved to the users device. |

**Limitations**

1. The code only runs on devices containing the internet browser Mozilla Firefox as a browser. Therefore, for devices without Firefox, the bot will not be able to run the user’s input.
2. When the code runs on a Mac OS, directions can be generated and sent to the user but the screenshot of the map on the Firefox browser is obstructed by the Python IDLE window which is not minimised unlike when it is run on a Windows OS. A possible solution to code a way to minimise IDLE window or to bring Firefox browser in front of it so as to take a screenshot has yet to be found by our team.
3. Slopes: The Bot converts calories into the distance to be covered assuming that it is being done on a perfectly flat plain. More calories are burned when going uphill, therefore the user may be burning more calories than intended. If the calories burned in going uphill are taken into account, the route could potentially be shorter.
4. Different ending point: The bot only maps routes which loop back to the starting position of the user. Those running or walking to exercise mostly return to their starting position and therefore there is no general need to include a function of the bot to map a route with a different ending point.
5. The number of calories burned varies greatly according to one’s weight and the time taken to complete the run/walk. While the bot takes into account the weight of the user, faster runners/walkers burn more calories per unit distance than slower ones. Therefore, the actual number of calories burned per unit distance may not be represented as we used a generic formula.
6. If a user inputs a negative value, the distance to run/walk will be negative as well and the routeloops.com will not output a route as it cannot process negative distance. The bot can inform the user to use only positive values when user input is requested.
7. The bot does not link to map applications such as Google Maps that can use the device’s Global Positioning System to track the location of the user. The user’s current location will not be displayed while they are en route. The user will have to manually orientate their device and/or be familiar with the road names so that they can follow the route.
8. The user will need to know how many calories they want to burn instead of just naming the food item they consumed if they want to burn the equivalent amount of calories. Extra effort is required on the user’s part to find out the number of calories contained by the food they consumed.
9. The screenshot of the map displayed on [www.routeloops.com](http://www.routeloops.com) may not be updated and some routes may be blocked by temporary obstructions such as construction work. This would thus pose a problem for the user as they will not be able to complete the route seamlessly and they will have to make the effort to discern alternative routes to take.
10. The site [www.routeloops.com](http://www.routeloops.com) may be down due to website maintenance and therefore the bot will not be able to map a route in the first place. Hence, bot may not be able to perform its function completely.

**Future scope**

Other applications for the bot in the future is to include functions that compute calories burned through other workout options such as biking. Also, Google Maps could be used to map the routes instead in order to provide the user with more accurate route directions from the same website and provide the user with their location in real time so that navigating through the route is easier. In addition to the existing functions of the UberRun bot, instead of inputting calorie values, names of food items could be keyed in as input for users who want to burn off the equivalent calorie content from food they consumed.

**Conclusion**

In conclusion, the Telegram Bot UberRun supplements fitness tracking technology as it provides an alternative method of discerning a suitable workout for the users who are focused on burning a certain amount of calories. Since monitoring calorie intake and output aids in motivating those wanting to reach a fitness level goal, this bot is helpful in determining how much a user should exercise by running/walking. It also helps the user navigate the suggested route in order to fulfil the number of calories they would like to burn.

While there are limitations to the function of the bot with regards to accuracy and flexibility in choosing a route, it serves to provide users with an reasonable approximation of the amount of calories they burn through the workout. This will nevertheless aid them in keeping track of the data of their walk or run and fuel their motivation to maintain an active and healthy lifestyle.

# References

Beckham, J. (2012, 04 19). *Fitness trackers use psychology to motivate couch potatoes*. (WIRED) Retrieved from https://www.wired.com/2012/04/fitness-tracker-psychology/