ENGR 212 Programming Practice
Mini Project 2
March 15, 2016

One of the challenges that SEHIR students face in their daily life is to decide on what to order when they go to Cafe Crown (CC). CC provides a large selection of delicious entrees, and choosing one among many may often be a big hassle. In this mini project, you are going to develop a recommendation engine that will help CC guests decide what to order. Details regarding the requirements are as follows:

1. Your program will have a graphical user interface (GUI) which will look like as shown in Figure 1. Details about how it should work are provided below.

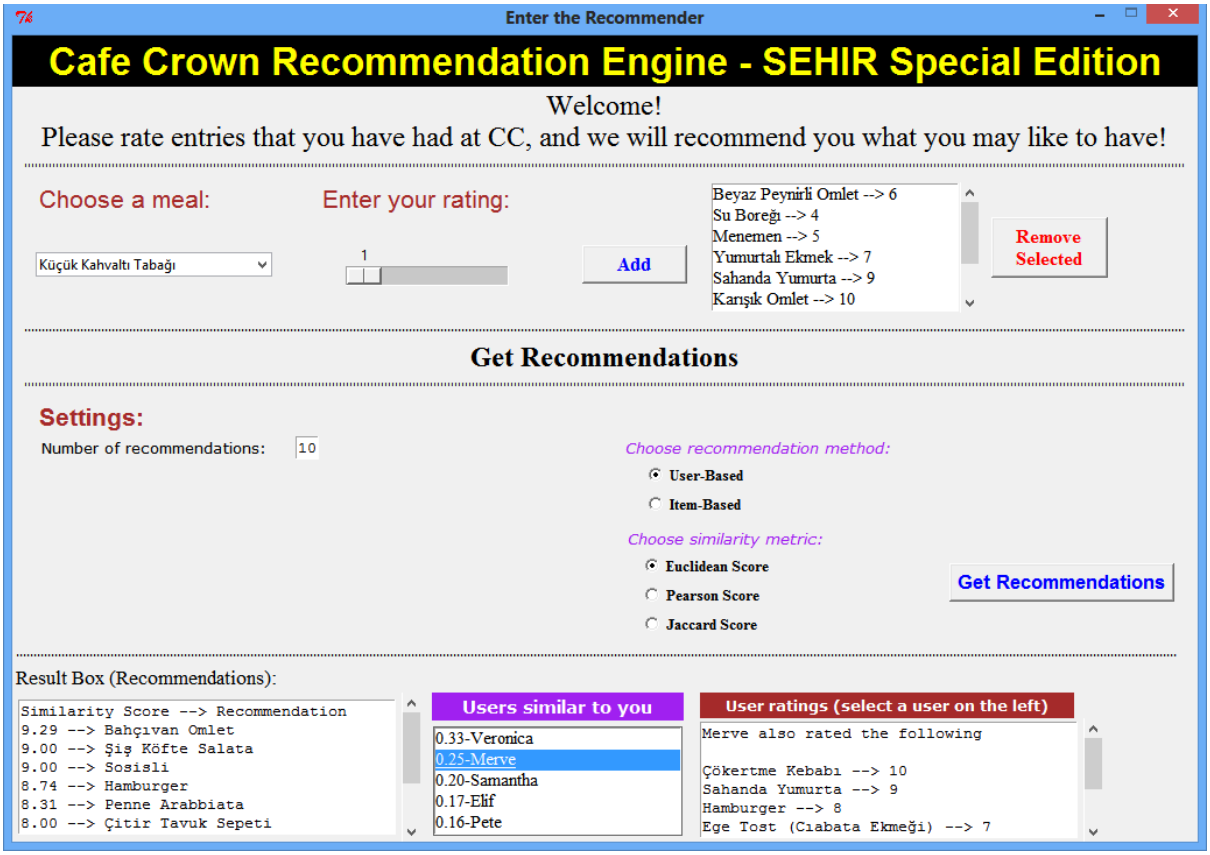

Figure 1

- At the top part, the user will enter her ratings for entrees she had before at CC. The available entrees at CC will be provided to your software as an Excel file (Menu.xlsx), and you will populate the corresponding combo box with entrees by reading from this file. The user will select a meal from here, then a select a rating for it through a slider widget, and add it to a listbox that keeps track of user ratings. The "remove selected" button will allow the user to remove any rating from this box. The user's ratings should be stored in a database file called "ownratings.db". For the first run, if this file is not available, it should be created by your program. For the subsequent runs, the ratings listbox should be automatically populated with entries from 'ownratings.db' database when your program is starting. Any new rating additions or removals done on the GUI should be also reflected into the database file right away as well. This way, the user will not lose her existing ratings when the application is closed and started again.
  Hint: While reading entries from Menu.xlsx file, in order to handle special Turkish characters, use encode('utf-8') (e.g., `sheet.cell(row_index,col_index).value.encode('utf-8')`)

- In the next section, the user will configure settings for the recommendation engine (e.g., choose the similarity method, etc.), and click on "Get recommendations button".
- The bottom part will show the recommendations. On the left, you will list the recommendations with predicted ratings. At the middle and right hand side, the content will be populated based on the recommendation method chosen by the user as specified next:
  - For user-based recommendation: The middle box will list the set of similar users with their similarity values. When a user is selected here, the box on the right will show the ratings of the selected person.
  - For item-based recommendation: The middle box will list original ratings of the user (the same content listed at the topmost rating box). When an entree is selected here, the box on the right will show the similar items to the selected item with their similarity values.

  The label texts in the middle and right region in each of the above cases should be updated appropriately. You will be provided a database file called "cc_ratings.db" that will list ratings of CC entrees by a number of people. You will use it to come up with the recommendations for the current user.

2. After getting a set of recommendations, the user should be able to modify any part in the topmost and middle regions of the interface, and then click on the "Get Recommendations" button again to get a new set of recommendations based on the latest settings and ratings. You **should not** use the place geometry manager in any part of the project.

For this week, focus on GUI and user-based recommendation that we covered in the class. We will cover item-based recommendation next week, which is also required for this project.

Can you provide any further pointers that may be helpful? :

- You may use **xlrd** module to read Excel files. The following tutorial have example usages of this module. The code piece on page 9 of this tutorial should be sufficient for you.
  - http://www.simplistix.co.uk/presentations/python-excel.pdf
- For meal selection, you may use the ComboBox widget of Tkinter. The following example code piece may help:
  - http://stackoverflow.com/questions/17757451/simple-ttk-combobox-demo
- In several parts, you will need to provide a vertical scrollbar to accommodate entries that does not fit to the widget. The following link shows an example for how to add a vertical scrollbar.
  - http://www.java2s.com/Tutorial/Python/0360__Tkinker/ListBoxwithscrollbar.htm
- Slider widget example: http://www.python-course.eu/tkinter_sliders.php
- For the configuration part, you may use the RadioButton widget of Tkinter. The following example code pieces may help:
  - http://effbot.org/tkinterbook/radiobutton.htm
  - http://www.python-course.eu/tkinter_radiobuttons.php

Warnings:

- **Do not** talk to your classmates on project topics when you are implementing your projects. **Do not** show or email your code to others. If you need help, talk to your TAs or myself, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve "**should**", "**should not**", "**do not**", and other underlined/bold font statements.

- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.
- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code. You may be interviewed about any part of your code.

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students (doing it individually is recommended). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file (Do **not** include recommendations.py that you import). Name it with your and your partner's first and last names (see below for naming).
  - If your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do **not** use any Turkish characters in file name).
  - If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on March 29, 2016**.

  Late Submission Policy:

  - -10%: Submissions between 17:01 – 18:00 on the due date
  - -20%: Submissions between 18:01 – midnight (00:00) on the due date
  - -30%: Submissions which are 24 hour late.
  - -50%: Submissions which are 48 hours late.
  - Submission more than 48 hours late will not be accepted.

Grading Criteria? :

- Does it run? (Submissions that do not run will get some partial credit which will not exceed 30% of the overall project grade).
- Does it implement all the features according to the specifications and produce correct results?
- Code organization (Meaningful names, sufficient and appropriate comments, proper organization into functions and classes, clean and understandable, etc.)?
- Interview evaluation (your grade from interview will be between 0 and 1, and it will be used as a coefficient to compute your final grade. For instance, if your initial grade was 80 before the interview, and your interview grade is 0.5, then your final grade will be 80*0.5 = 40). Not showing up for the interview appointment will **result in** grade 0.

Have further questions? :

- Please contact your TAs (Jareth or Dogukan are focusing on projects. You may want to talk to them first, but you may talk to Ali and Bekir as well) if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**