**Program assessment rubrics**

Lab Group:  **DD2**

Names of members: LEONG JING WEN

LIM YU JIE

LISA SHANNON GOH

Include this page in your softcopy submission. You are required to fill in the first 4 entries in the table, indicating how you have made use of these CT concepts in your software design.

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment Criteria** | | **Description/Comment** | **Marks** |
| 1 | Use of Pattern recognition  (10) | We used pattern recognition in the designing of our GUI by creating different pages such that the user can easily navigate to other pages. This is because there are multiple instances where the same display has to be rendered, such as next buttons/back buttons, instead of using the same code twice, we created a page format.  Also, we used it to sort by price and review as we realised that the sorting mechanism was similar for both. It was applied in displaying of data, as we iterate through the lists and blit images as well as change the position of the images through a loop. |  |
| 2 | Use of Abstraction (10) | There is abstraction in data, which is the use of data structures and abstraction in algorithms, which is the use of functions. The use of composite data types such as dictionaries and lists to store and sort our data is an example of how we used abstraction. Functions we defined such as sort\_by\_price\_or\_review() and update\_info\_for\_changing\_food() etc. are also used to sort, calculate, update information and abstract data from our dictionary. When coming up with the distance functions, we are also abstracting information from Google API. |  |
| 3 | Use of decomposition (10) | We used decomposition in our update information function where we split function into 4 smaller subproblems, namely, changing price, changing ratings, adding food and removing food.  While trying to find out the cantees that match the stalls the user is interested in, we split it into 4 different scenarios, whether the person has a preference as to stalls and/or canteens, and tackle them individually.  For calculating distance, we broke it down into firstly, getting the user location, then calculating the distance to each of the 8 reference points, to get the average value for higher accuracy. |  |
| 4 | Algorithm Design (20) | We created functions that were called many times which shortens our code for better user readability.  We also included error-handling. For instance, under the update information function, when the user wishes to change the price or ratings of a food but chooses a canteen which does not have that food, our code will require the user to choose again, until he chooses a food that is in that canteen.  We also used different methods for sorting. For instance, we used bubble sort for the sorting of ratings and prices, and we used merge sort to sort the distances.  We have also used linear search to search for food under a specified budget or search canteens containing a particular food. |  |
|  |  |  |  |
| 5 | User Interface Design (10) |  |  |
| 6 | System Complexity (10) |  |  |
| 7 | Teamwork & Presentation (10) |  |  |
| 8 | Individual Oral Assessment (20) |  |  |
| Others (Optional) | | | |

Date of Assessment:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ By:\_\_\_\_\_\_\_\_\_\_\_\_\_\_