NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

MINI PROJECT

CE1003/CZ1003:
Introduction to Computational Thinking

Real-time Canteen Information System Report

SCHOOL OF COMPUTER SCIENCE and ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Written by:
Hoo Kah Jun U1922508A
Suhana Gupta U1923230B
Ho Jia Jia Stefanie U1630857F

Overview

Chapter 1: Introduction
Introduction to the program we use, and the purpose of the program

Chapter 2 : Algorithm Design
The brief description of the important user functions, and a top level flow chart of our program

Chapter 3 : Program testing
Error handling test cases of our program by the user

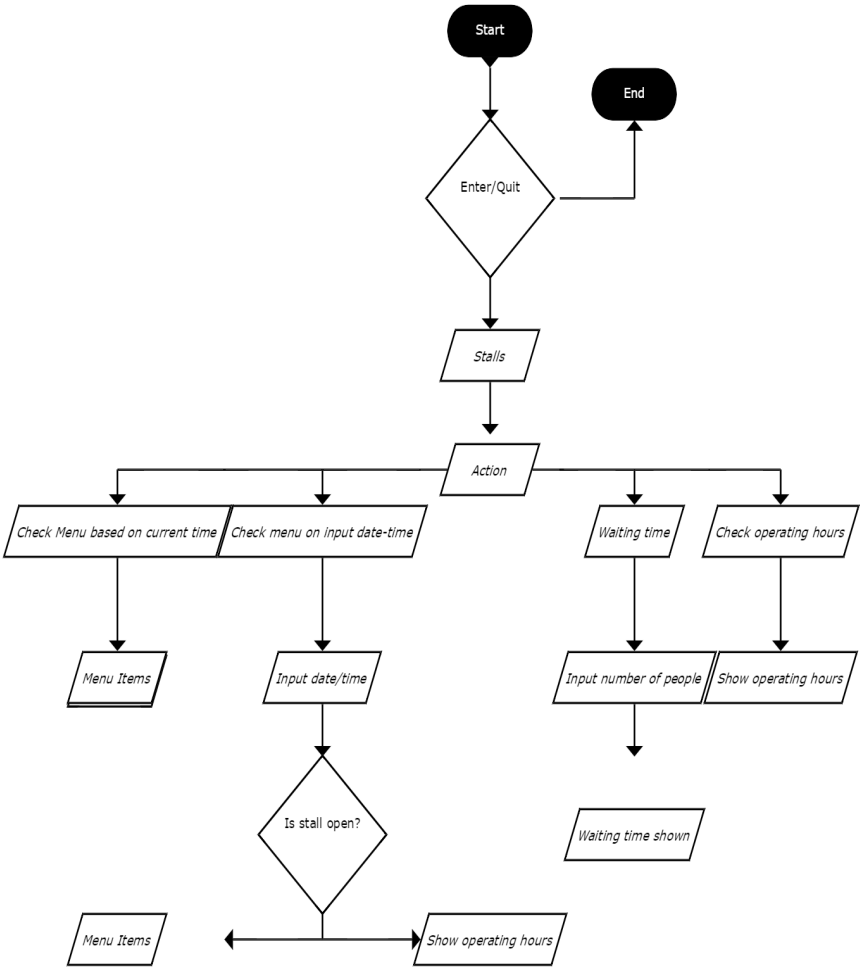
Chapter 4 : Reflection
The difficulties encountered and the way to conquer in developing the program. Furthermore, we look into improvements we can make.

1 Introduction

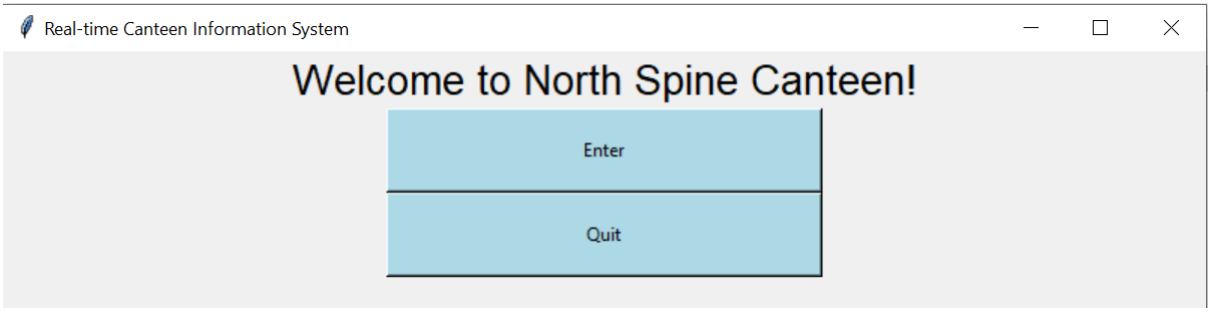
This is a software designed for NTU students and staff allowing the user to browse the opening stalls in NTU North Spine Canteen and their current available menu at any date and time. We are able to display real-time NTU North Spine Canteen information using this program. We also require the use of Python 3.0 IDLE to run the program on computer as we are using Python 3.0 in conjunction with GUI Program Tkinter.

2 Algorithm Design

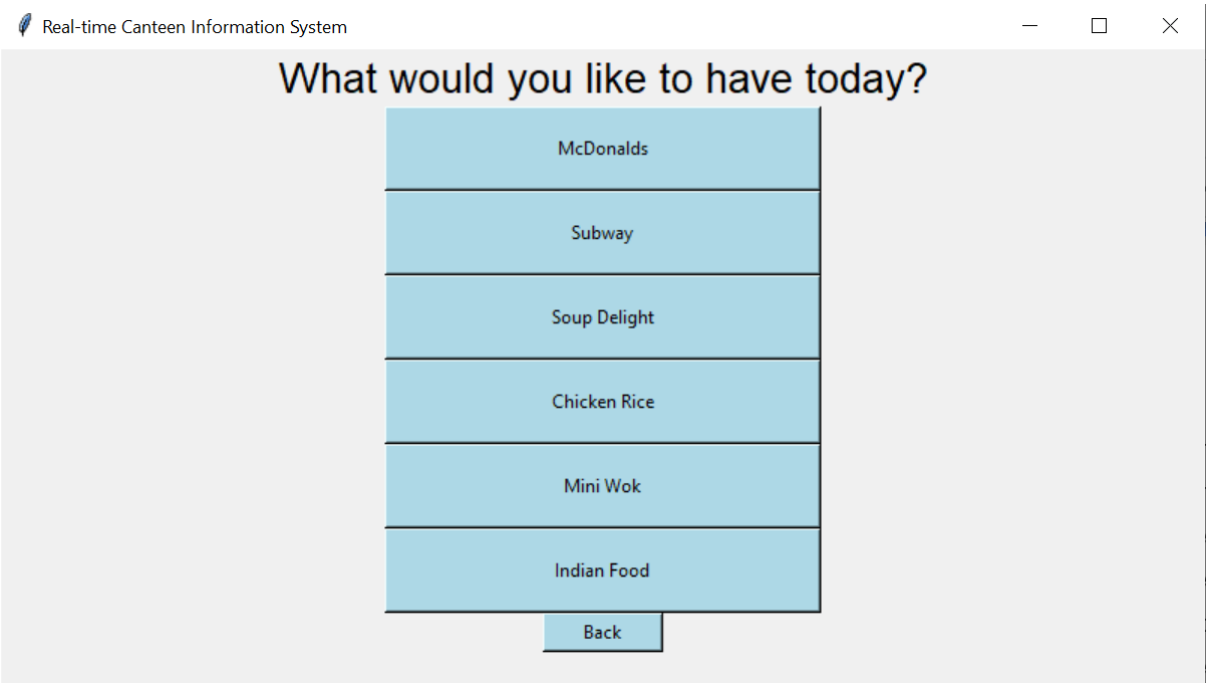
Firstly, we import several modules such as tkinter, datetime, pickle, system. We created functions for each program feature such as StallMenu, OperatingHours, WaitingTime etc, and compiled it at the end of the program so as to create the Real-time Canteen Menu. Therefore, our program has appropriate modularity due to the creation of different, separate functions necessary for the whole program to operate smoothly. Furthermore, it would be much easier to debug any possible errors if we defined functions as we would only need to adjust the code in the functions instead of adjusting all the affected lines in the code. With that, we would only need to call the corrected function to run the program. Other reasons for choosing to make use of user-defined functions are to improve clarity of the code, prevent excessive repetitions and prevent our code from being unnecessarily long. Besides functions, we also utilised serialization via the Pickle module (pickle.dump and pickle.load) for efficient file management.



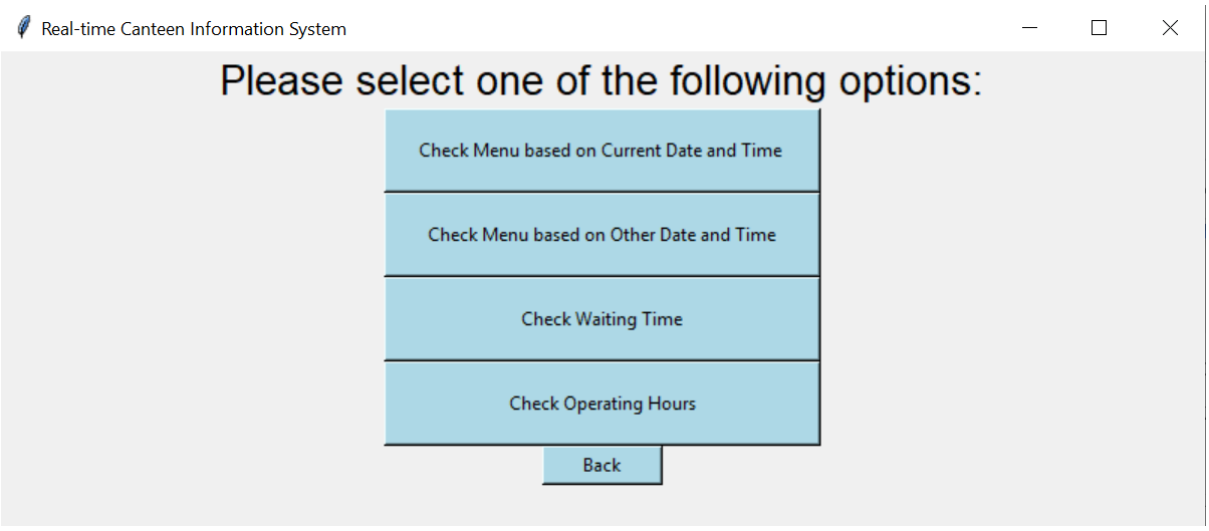
Moving forward, we will provide a brief run-through of how our program works. The first page of the program will begin with a welcome message “Welcome to North Spine Canteen!” along with 2 buttons which are Enter and Quit. If the user wishes to proceed with the viewing of the canteen’s stalls and menu, he will click on the Enter button. Otherwise, he can press Quit and exit the program.



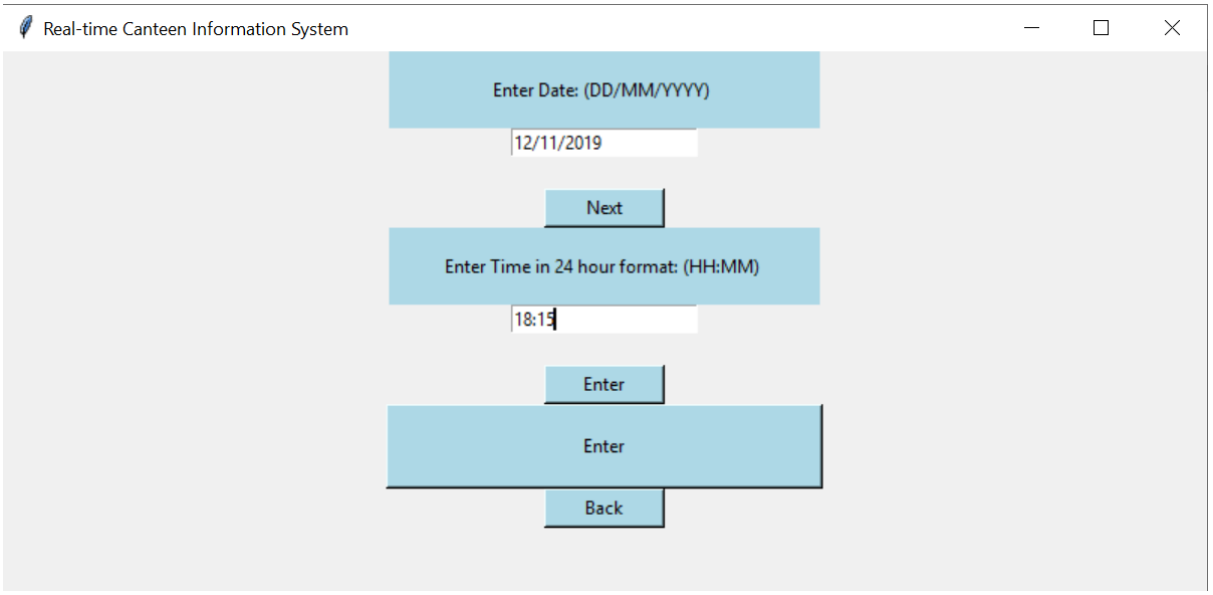
After clicking the Enter button, the next page will ask “What would you like to have?” followed by a list of food stalls available - McDonalds, Subway, Soup Delight, Chicken Rice, Mini Wok and Indian food. There is also a back button in case the user changes his mind and does not want to view at stalls and their menus.



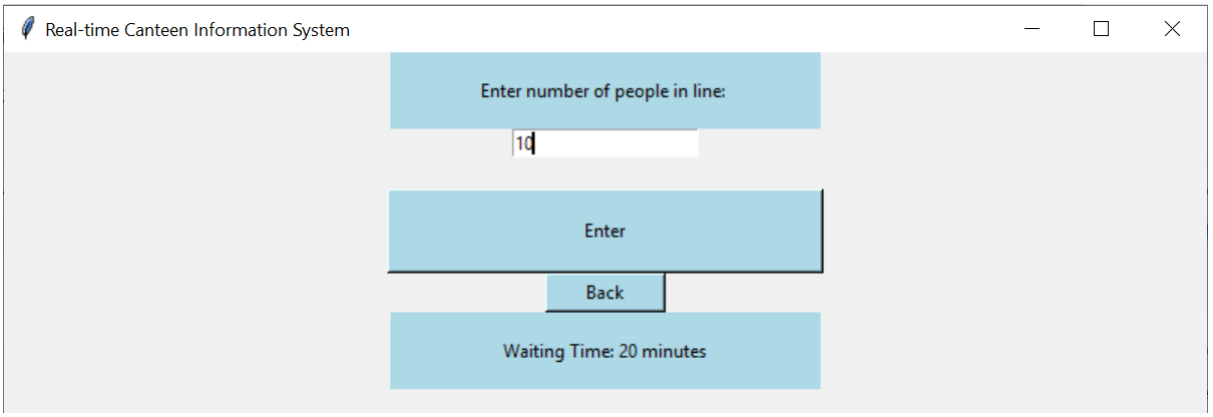
Once a user selects one of the six available stalls, he will be brought to the next page where he will be asked to select what kind of information he would like to check for the particular stall. The options are listed as follows: (a) Check menu based on current date and time (b) Check menu based on other date and time (c) Check waiting time (d) Check operating hours.



We will now use McDonald's as an example to provide a straightforward example. The user can choose to check McDonald's menu at current date and time using the program's clock which we imported with datetime. Alternatively, he can choose to input his desired date and time, and the program will show the food items available at the specified date and time (be it current or user input).



The user can also get an estimate of the waiting time required should he choose to dine at McDonalds. The user simply has to input the number of people in the queue for McDonalds into the text-box provided and click Enter to get an estimate of the waiting time required. For example, if the user entered that there were 5 people in line for McDonalds, the program would later show that the estimated waiting time is about 10 mins with the assumption that one person would require about 2 mins.

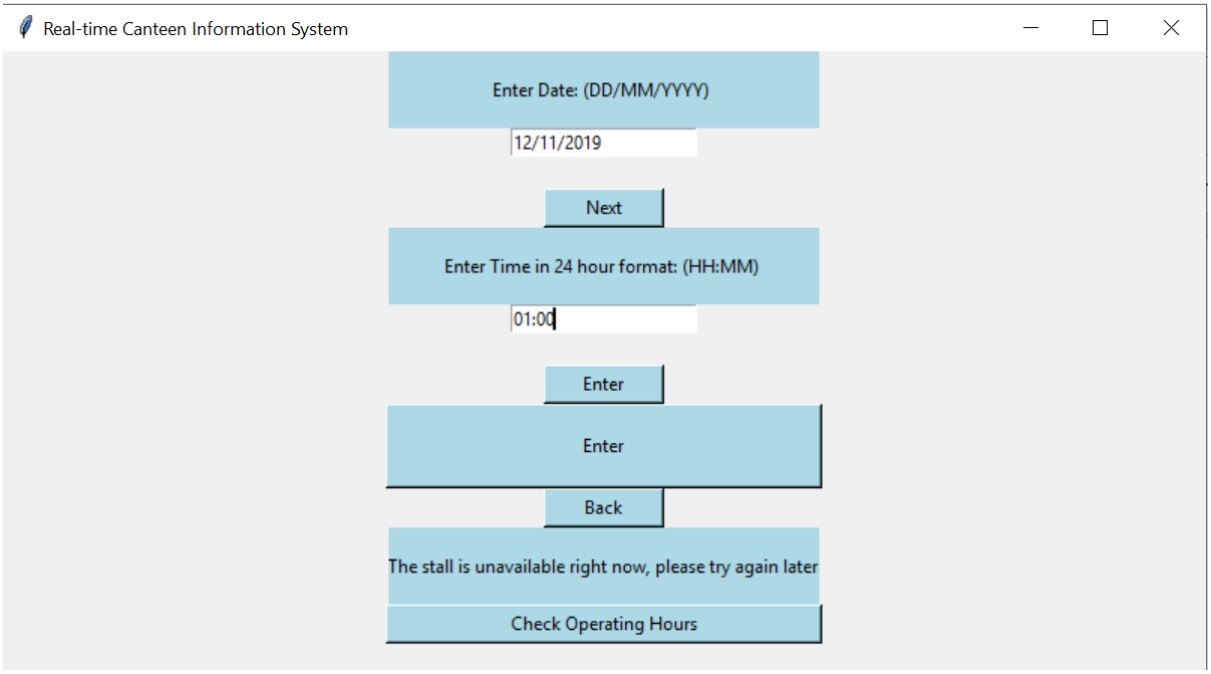


Furthermore, the user can also check McDonald's operating hours to take note of the opening hours and closing hours as they might differ on weekdays and weekends.

Should the user change his mind and wishes to look at the menu of a different stall, he can simply press the back button which will bring him back to the starting page where he can easily choose another stall and repeat the process. With regards to our algorithm design, we believe that it is simple enough for anyone to operate and that it provides comprehensive information regarding the stalls in North Spine Canteen.

3 Program Testing

Our program has a few instances where the user is required to input data. After choosing a particular stall, the user is then given a choice to view the menu based on current date and time or other date and time. If he chooses to look at the menu based on other date and time, he will be required to input the desired date and time. Therefore, to ensure that the program runs smoothly, we need to consider situations where the user does not input data in the correct format. Our algorithm requires the user to input date and time in a specific format where date is DD/MM/YYY and time is HH:MM. In the situation where the user does not follow said format and enters a bunch of random letters, the program will not let the user move forward until the right input is detected. Another case we took into consideration is when the user inputs the correct format for date and time but is unable to view the menu as the stall would be closed at that time. There would be a message saying "The stall is unavailable right now, please try again later!". Furthermore, this will trigger the appearance of another button that allows the user to check for the operating hours of the particular stall.



4 Reflection

One of the difficulties we faced in the process of writing the code was that we had little idea on how to allow the user to input his own date and time. We had to ensure that the input date and time by the user are linked to the system so that they recognise the exact day and display the menu items accordingly. Our solution is to divide the problem in sub-problems by using various functions.

Despite the many obstacles we faced during the course of this project, we still persevered and never gave up trying to find solutions to our issues or ways to improve our program. This hands-on project helped enrich our understanding of Python as we learnt how to apply the theoretical knowledge we learnt in lectures. Furthermore, we had to learn more about Tkinter as we opted for a GUI version of the program to make it more user-friendly with a better design, which was very challenging.

Furthermore, we felt like we could have incorporated pictures into our program to make it look more interesting and vibrant. The lack of pictures in our program does not attract the user and is not suitable for advertising the canteen and might not bring an increase in sales, which is probably the objective of using the program.