****

**NANYANG TECHNOLOGICAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**CZ1003: Introduction to Computational Thinking**

**Mini-Project**

**Prepared By:**

**Poh Kai Kiat**

**Parthan Muralidharan**

**Palaniselvam Shyam Sundar**

**TABLE OF CONTENTS**

**1.Introduction**

**2.Algorithm**

* Flowchart
* Descriptions of functions created

**3.Error Handling**

**4.Reflection**

**5.Contributions**

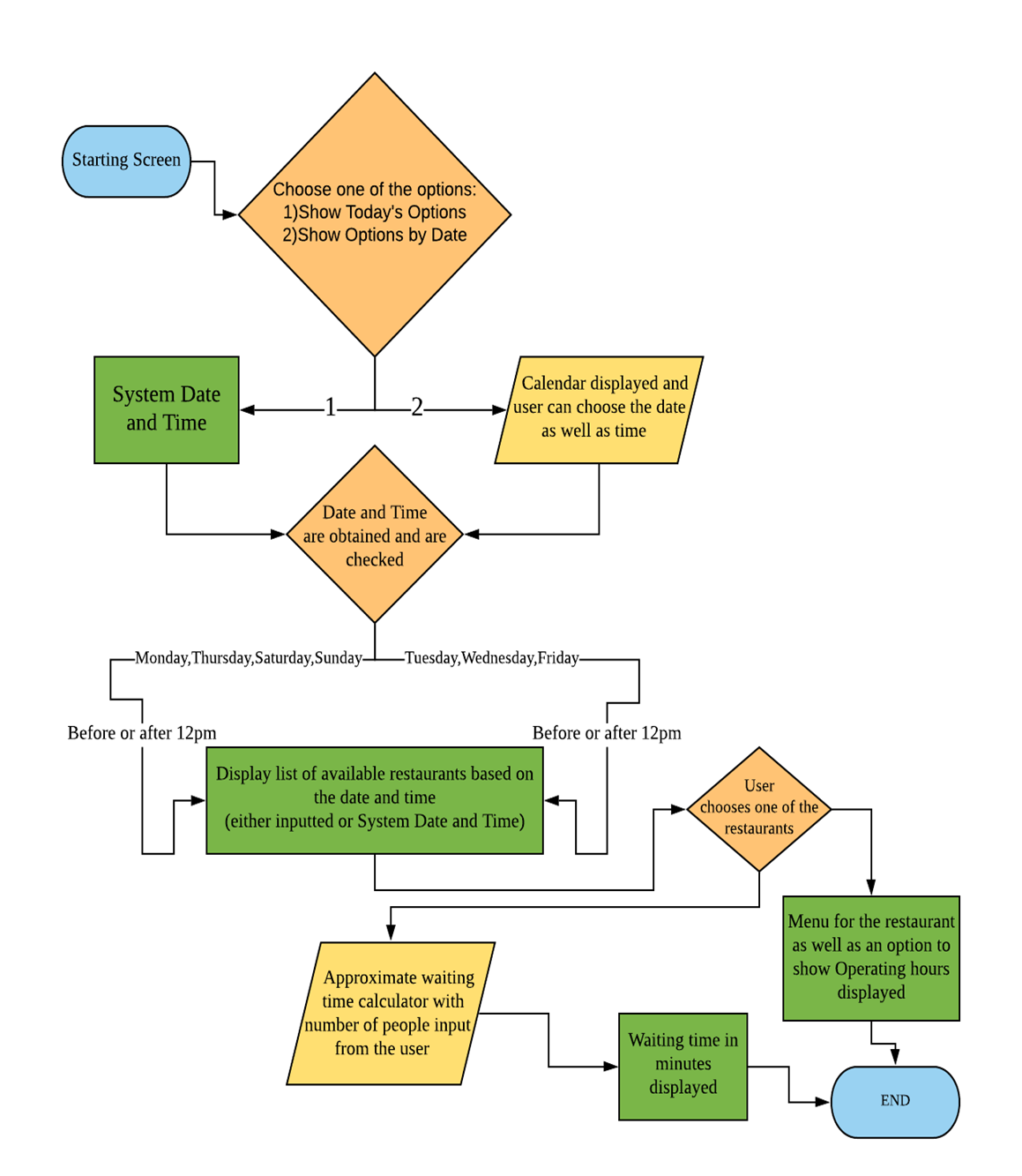
**Introduction:**

The program implements a Real-time NTU Canteen System. The framework used in this program is tkinter.

Included in the working of the program:

* A python file named ‘data.py’ which is a dictionary for storing operating hours.
* Menu folder with text files of the different restaurant menus.
* Images folder which consists of the background images for the various pages.

**Algorithm design:**



**FLOWCHART**

**USER-DEFINED FUNCTIONS**

1. **main()**

* This function helps create a canvas with a non-resizable window ‘win1’.

1. **landing\_page(window name)**

* This is the starting page of the program when it is run.
* Gives the user the option to choose one out of the two choices: 1)Show Today’s options 2)Show Options by Date

1. **showOptionsToday(window name)**

* It shows the available restaurants for the day.
* It prints the date and time from the **Date\_Time(window name)** function and obtains the day of the week.
* Using this information about the day of the week, it checks this string value against a list of items in two separate lists ‘List\_someDays1’ and List\_someDays2’.
* Upon detecting a match, the corresponding available restaurants will be displayed in the form of buttons.
* The value associated with the button is sent into the function Today\_Chosen using **lambda** upon clicking it.

1. **calculate(window name)**

* It helps to calculate the waiting time based on the value of the input from the user.
* It also throws an error screen if the user inputs an invalid character.

1. **show\_operating\_hrs(id,window name)**

* This function creates an Operating Hours button in every restaurant menu page.
* It has another function inside it called **operating\_hrs\_window** which creates a window, upon clicking the Operating Hours button, and shows the corresponding restaurant’s operating hours.
* This information is obtained from another python file named ‘data.py’. It contains the information in the form of a dictionary.

1. **showOptionsOtherDays(window name)**

* This page allows user to choose the day and time and brings them to **show\_list\_avalble\_store** page.
* It has an internal function called **getDate()** which validates the user input as well as converts the time to 24hrs format. The time will then be used to check with a dictionary imported from ‘data.py’ and check if there are any stores available during the particular time.
* When iterating through the dictionary we append the key(id) of stores that are available into a list called ids,this list serve as a reference for the **show\_list\_avalble\_store** page .
* An error window will pop up when the user inputs an invalid time or no stores are available.

1. **show\_list\_avalble\_store(final\_time,ids,window name)**

* This page will show the stores that are available in a particular time frame.
* This is done by iterating through ids and created a button. The button links to the menu of the store via **function 8.**

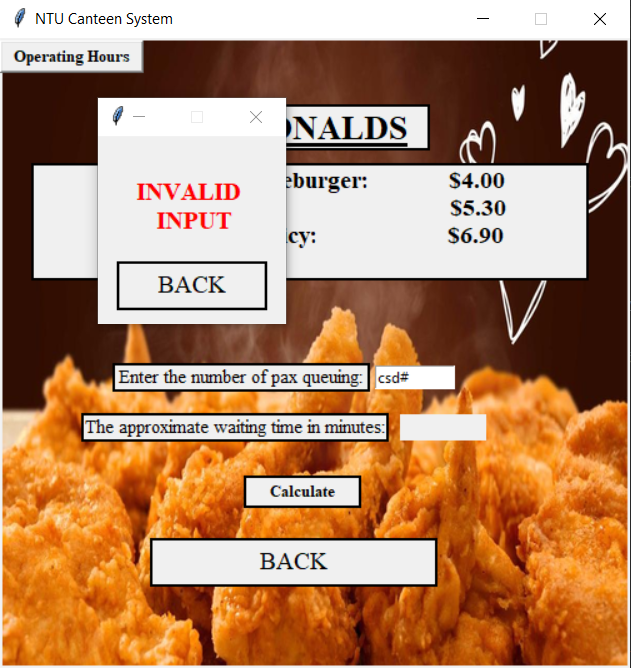
1. **Today\_Chosen(window name,id,ids,time)**

* This page shows the menu of the store.
* Using the id of store that we are supposed to show, we open a text file which contains the corresponding menu.
* As our menu is separated into lunch and breakfast, using time, we can determine which menu to show on the page.
* This function further invokes **calculate(window name)** and **show\_operating\_hrs(id,window name)** , which allows to estimate waiting time as well as view the operating hours of the store

**Error Handling Test Cases**

1. Checking for invalid entry during approximate waiting time calculation.

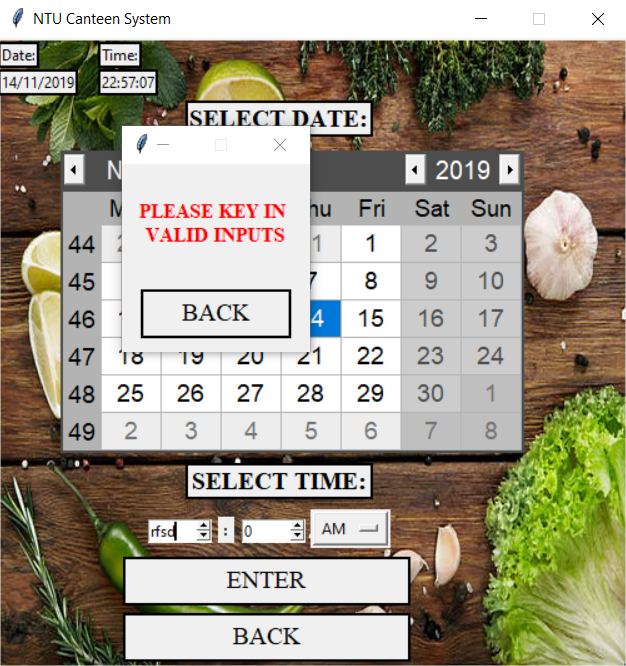
If the user enters anything other than a number, an error screen will pop up alerting the user of the invalid input. Pressing the ‘BACK’ button will allow the user to key in values again.



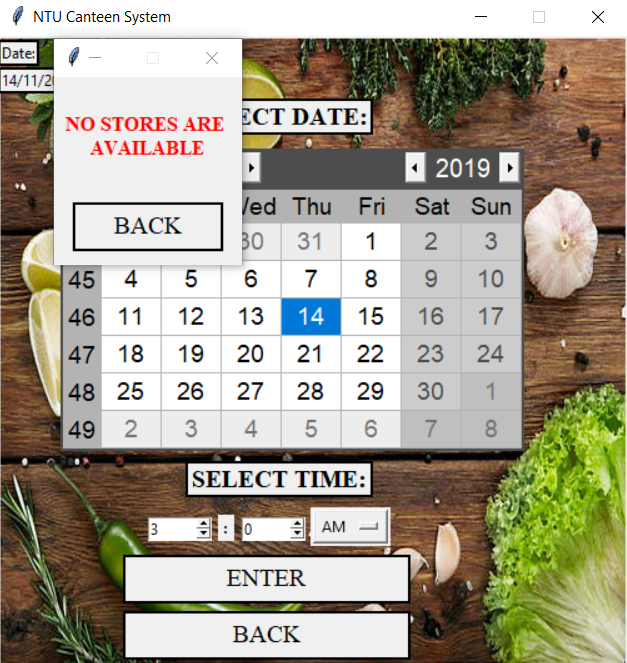
1. Similarly, integer numbers can only be entered. Number of people input cannot be in fractions or decimal. Pressing the ‘BACK’ button will allow the user to key in values again.



1. If the user enters an invalid input when asked to choose a time, an error screen will be displayed. By pressing the ‘BACK’ button the user is allowed to choose another date and time.



1. If the user chooses a time that is out of the operating hours of the entire Canteen, then a window will open that will inform the user that no stores are available. By pressing the ‘BACK’ button, the user is allowed to choose another date and time.



**LEARNING OUTCOMES**

Difficulties encountered

* The first difficulty we faced was creating the GUI without prior knowledge or experience. We only got started after going through several tutorials and documentations.
* Another problem we faced was relating to getting user input of date and time. Initially we created a drop-down list that takes day, month and year as inputs. However, this method was more error prone and therefore we decided to implement a calendar using tkcalendar.
* The initial design of the program was not too difficult. However, we realized there was a lot of redundant code. Figuring out the logic; making the code concise by using proper functions instead of repeatedly copy pasting the same code, was challenging.

Knowledge Gained

**Kai Kiat**

* This project provided a good platform for us to improve the knowledge taught in the course, such as, try and except, dictionaries, for loops, using of lists, etc.
* Another great takeaway from this course was the extensive use of the datetime module which was something not taught in this course. Throughout this project, we had to manipulate time to different format, such as 1230AM to 0030, etc.

**Parthan Muralidharan**

* An opportunity to practice all the basics that was learnt during ICT.
* I specifically learnt how to call pictures and files that have been stored in your own computer using Python. Not just calling documents but integrate them with the Python program that runs.

**Palaniselvam Shyam Sundar**

* I feel like this is an effective way to learn a new programming language. Knowing that we have to work on this project with little to no prior knowledge lead to a lot of self-directed learning. Working in teams to complete this project was also greatly beneficial because we learnt from each other.
* I learnt Python GUI through several video tutorials. Implementing so many different aspects of Python in the program, although challenging, was actually very interesting.
* File Handling through Python, implementation of calendar and using datetime module were my key takeaways from this project.

Improvement to project and course

* Perhaps the grouping for the mini-project could be announced earlier so that we can have sufficient time to learn the frameworks required. Also, it will be great if the lab assistant could give us suggestions on how to improve on our code after every lab.
* We could have improved the way we collaborated as it was hard to keep track of changes when more than 1 person is working on the code. We could have use tools such as Google Collab which allow everyone to work on a script together.
* Working between Windows and Mac was a hassle. Code that compiled perfectly on Windows would work on Mac, only after certain adjustments were made to the code. Maybe we could have had more regular meetings and worked on one system to better deal with this issue.