Final Project Design Report

Tam Nguyen and Azul Aung

**Design Strategy**

Classes used:

* Restaurant: manages drivers and orders.
* Driver: an object representing a driver with all the relevant information and functions
* Order: an object representing an order with all the relevant information and functions
* Time: an object that stores the hour and minute of a specific time

STL container classes used:

* Deque (used in the Restaurant class): used for the cooking queue and the delivery queue. We wanted to use the queue at first but a deque allowed us to iterate through the list.
* Vector (used in the Restaurant class): contains a list of pointers to all the drivers in the system (logged in or not).

Commands in main are processed with a while loop and if-else statements. After that, the main file will call the corresponding functions which are also defined in main.cpp. These functions create the necessary Driver, Order, and Time objects and are mostly used the functions in the Restaurant class to navigate. Correspondingly, Time class handles the time input by converting the time string to appropriate time. In Restaurant class, its method call functions in the Driver, Order, and Time classes when needed.

The sample commands are easy and clear for user. We created functions in main.cpp with all lower case command to prevent confusion for user.

**Workload Division**

* Tam: functions in Restaurant and Order class with all implementation and documentation for these.
* Azul: functions in Driver and Time class with all implementation and documentation for these.
* Test file and report were done by both.

**Team Member responsibilities**

The responsibilities of the members were to be responsive with each other and to be clear when we were communicating with each other. We also had to finish the work assigned on time, so that the project would be able to proceed without problems. Testing was done both separately since we worked on our all classes before compiling together. We were also to assist each other in our tasks, seeking feedback from each other and to cooperate well.

* **Schedule of meeting:** Sunday 1 – 3pm and during lab hours.
* **Work sharing and organization:** <https://github.com/nmt64/PizzaDelivery>

**Team Process reflections**

The project was finished without any large problems or obstacles. Although, at the beginning of the project, it was a bit hard to cooperate together as the mode of communication wasn’t clear, as Azul wasn’t aware that he received an email. Implementation of the classes was done without any particular issues. Then Tam wrote the interface file, and the project was finished on the coding part. We went over the project together to make sure everything was working fine, and that there were no unexpected results.

In retrospect, something we wished to have worked on more was the detailed design for each class. Since we implemented our classes individually, there are many data members that need to redefining when we finally comply together. Our workflow could definitely have been improved with more communication, and the workload division could have been more equally as a result.

In the end, this project gave us a chance to expose to teamwork environment as a developer. Even when there were some disagreements, we learned to harmonize our ideas and everything turned out quite nicely in the end.