|  |  |
| --- | --- |
| **Item** | **Answer** |
| **Section** | G1 |
| **Name(s)** | Adeline Chin Wen Jie (Adeline)  Gokarn Mallika Nitin (Mallika) |
| **Brief description of project** | Our project is a restaurant simulator. The owner of Gluttons Bay Restaurant (fictional) is interested in looking at the difference between his company and the restaurant Subway, in terms of how much time it takes to service a certain number of customers.  Subway has a very linear process and attends to one customer fully at a time before moving onto the next customer. For sake of comparison, we have split the role of the waiter into waiter and chef. The process at Subway is as follows:  Gluttons Bay on the other hand is a Vegan 5 course restaurant that consists of Soup, Salad, Appetiser, Main Course and Dessert. Gluttons Bay provides seating as well, thereby requiring multiple customers to be serviced at a time. The process at Gluttons Bay is as follows:  In order to compare Subway and Gluttons Bay, a few assumptions have been made, they are as follows:   1. The food preparation overall, takes the same amount of time for chefs of both Gluttons Bay and Subway. The division of time on each part of preparation is as shown below:      1. The time taken by each customer on average overall, to decide on their food and thereafter let the waiter place the order is the same between both Gluttons Bay and Subway. The division of time on each part of ordering is as shown below:      1. The time taken by each waiter on average overall, to serve customers their food is the same between both Gluttons Bay and Subway. The division of time on each part of ordering is as shown below:      1. The time taken by each customer on average overall, to eat the food served has been *assumed* to be the same between both Gluttons Bay and Subway. *Additionally, we are looking at a high class Subway that provides seating and functions like a normal restaurant apart from the fact that they profess and follow self-service.* The division of time on each part of ordering is as shown below:      1. We are assuming that both Subway and Gluttons Bay have only one efficiently working waiter or chef at a time, that is to say, that to maximise efficiency and keep the time taken on preparation, serving etc. constant, the employees are employed in shifts that result in only 1 waiter and 1 chef being on duty at any given point in time. 2. As Gluttons Bay’s previous data shows that we cater to 15 customers on average per day, we will be using that as the upper-bound for our comparison.   One caveat of this comparison is that, on comparing the Single-threaded version and Multi-threaded version, there is no way to take into account the effect of the print statements and the time taken in switching between threads for multi-threaded. However, this is acceptable to the owner, as it can represent and account for the time taken by the waiter and chef to slack while walking and working continuously. |
| **Justification for multi-threading** | As shown in the process above, Glutons Bay caters to multiple customers at a time and has the technology and ability to keep track of multiple orders thereby letting a single worker concentrate on doing the job dictated as the technology tells him to do it. This means that the waiter can wait on a customer and immediately after placing the order either wait on the next customer or serve the next order to the respective customer. To correctly represent the process, we would therefore require concurrency.  In subway on the other hand, a single waiter, can only remember and prepare a single order for a single customer at a given point in time. Therefore only allowing the process to run in a sequential and linear manner. |
| **Transactional integrity** | Were there any potential race conditions; if so, how did you resolve them? Or are there still unresolved race conditions that you are aware of? |
| **Performance** | Is there anything noteworthy you did that improved performance? (include stats if any) |
| **Evidence of exploration** | Is there anything noteworthy that you did that required additional research (of topics not covered in class)? |
| **Innovation** | Why is your project innovative? |
| **Adherence to coding conventions & good practices** | Is there anything noteworthy that you want to mention? E.g. usage of a well-known multi-threading “best-practice”, or OO design pattern etc. |
| **References/Acknowledgement** | We used third party code from github user heidtJJ (Jared Heidt) as our starting point and built on it  The code to his project is in the folder titled reference.  We have also used the StopWatch |