ECE 437/CS 481

PA05

Project Report

Our code has various threads. We split our code into various functions, so we have a proper code modularity. We have various threads like the Main and the car threads for all the passengers to get into their rides.

In the main thread we make the CARNUM and the MAXPERCAR as global variables, so we can change them throughout the course of the code. We also keep of the total number of arrivals, rejections and waiting lists and initialized them to the 600 which is the total number of minutes from 9 AM to 7 PM. To keep a quick check, we initialized the waiting list for all of them as -1 so that way we can keep a check that the waiting list is empty. Then we loop over the totalmaxpeople which is 800 and increment the waiting time of the people who are waiting. We also made sure that there can’t me more than 800 people and we reject them. To keep a clear check, we have generated their time stamps and the total\_arrivals function takes care of this. The thread also takes care and calculates if we could add more people into the waiting list or no add adds them accordingly to either the rejections or the waiting lists.

We store all of this information in the ride\_status.txt file and will be using this for our plots.

Since the prompt doesn’t specify what happens to the people remaining in the waiting list after the park closes at 7, we decided to add these people to the rejected list, but their waiting times is not included in the average waiting time. Me and my partner had few differences in approaching this problem, but we agreed to this because they waited for their ride, but they never had a chance to ride so adding it to the average waiting time would not make sense to us.

The car threads make sure to check how many people to send in a car. This done by looking at their waiting times, a person can only get in if he has a longer waiting time than the others who were waiting and this is calculated as soon as they get in. After everyone is in, we decided to create a thread for the car with a struct so we can easily pass the information to the car threads (we took a few example codes from the slides to implement this). So, the car\_info struct holds the information about the waiting time and the number of people who are available to get into the car. We did this because the car also had a few specifications on the minimum and maximum people to get in. If the waiting list is full then then we fill -1’s from the right to show that these spots are now available. We do the same process until we finished looping through all the cars. As stated in the prompt all cars leave at the beginning of the minute and return at the end of the minute.

We now know that we after all the 600 minutes have been iterated to all the cars loaded for each iteration, now we can take all this data and put them in the csv files as stated in the document. We use this output data.csv files which contains the data for plotting and table generation. The main thread calls the output file and sends all this data to the main.py file to parse it. Python main.py should give the plots.

Here are our outputs after running our python file. We run it for various parameters.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A picture containing writing implement, pencil

Description automatically generated

A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated