CIS 657 LAB3 report

A Train Simulation system

Name: Jie Ren

SUID: 21037-2590

netID:jren21

**CIS657 Fall 2019**

**Assignment Disclosure Form**

Assignment #:

Name: Jie Ren

1. Did you consult with anyone other than instructor or TA/grader on parts of this assignment?

If Yes, please give the details.

NO!

2. Did you consult an outside source such as an Internet forum or a book on parts of this assignment?

If Yes, please give the details.

No!

I assert that, to the best of my knowledge, the information on this sheet is true.

Signature:\_\_\_Jie Ren\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date : 09/28/2019

PART1 : Time I spent on this lab

* How much time did you spend to do:
  + Analyze the problem, determine specifications, and create your design

**About 1 day of design. Including thinking about the details and discuss with others.**

* + Implement the design
    - write the program

**About 1.5 days of implementation**

* + Test/debug the program
    - Find/fix errors
    - Create test cases, and try them out

**About another 1.5 days of debugging.**

**(all the days mentioned above is about 8-10hours full day, so the workload of this lab is still a little bit too heavy).**

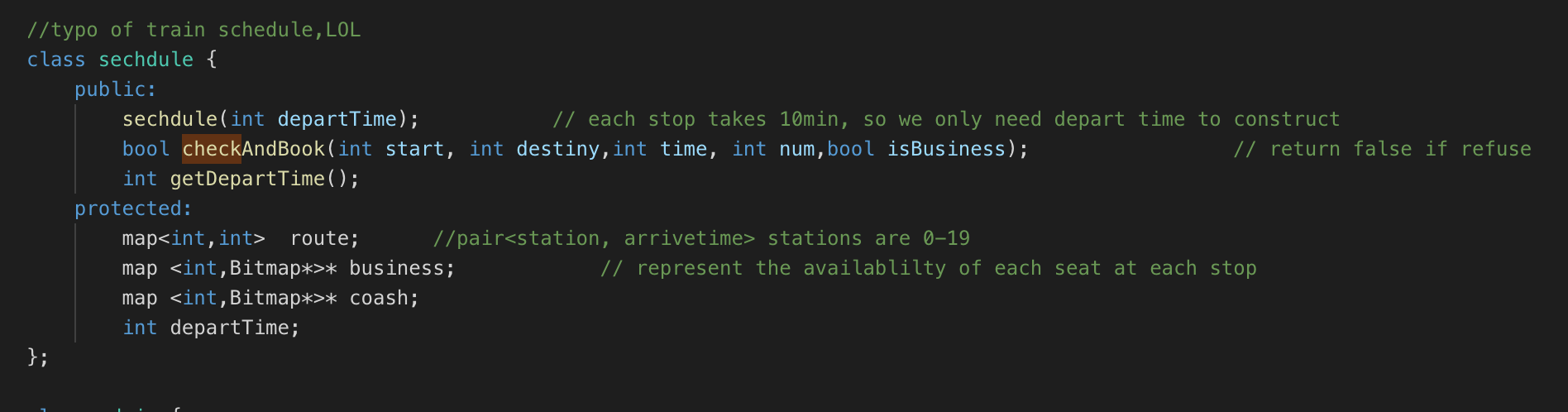
**PART2 : brief introduction of my design**

I create 2 classes for the entire simulation.

1. class sechdule{}

This holds all the information of the train sechdule. Including the departing time and arriving time of the train as well as the seats available on that train.

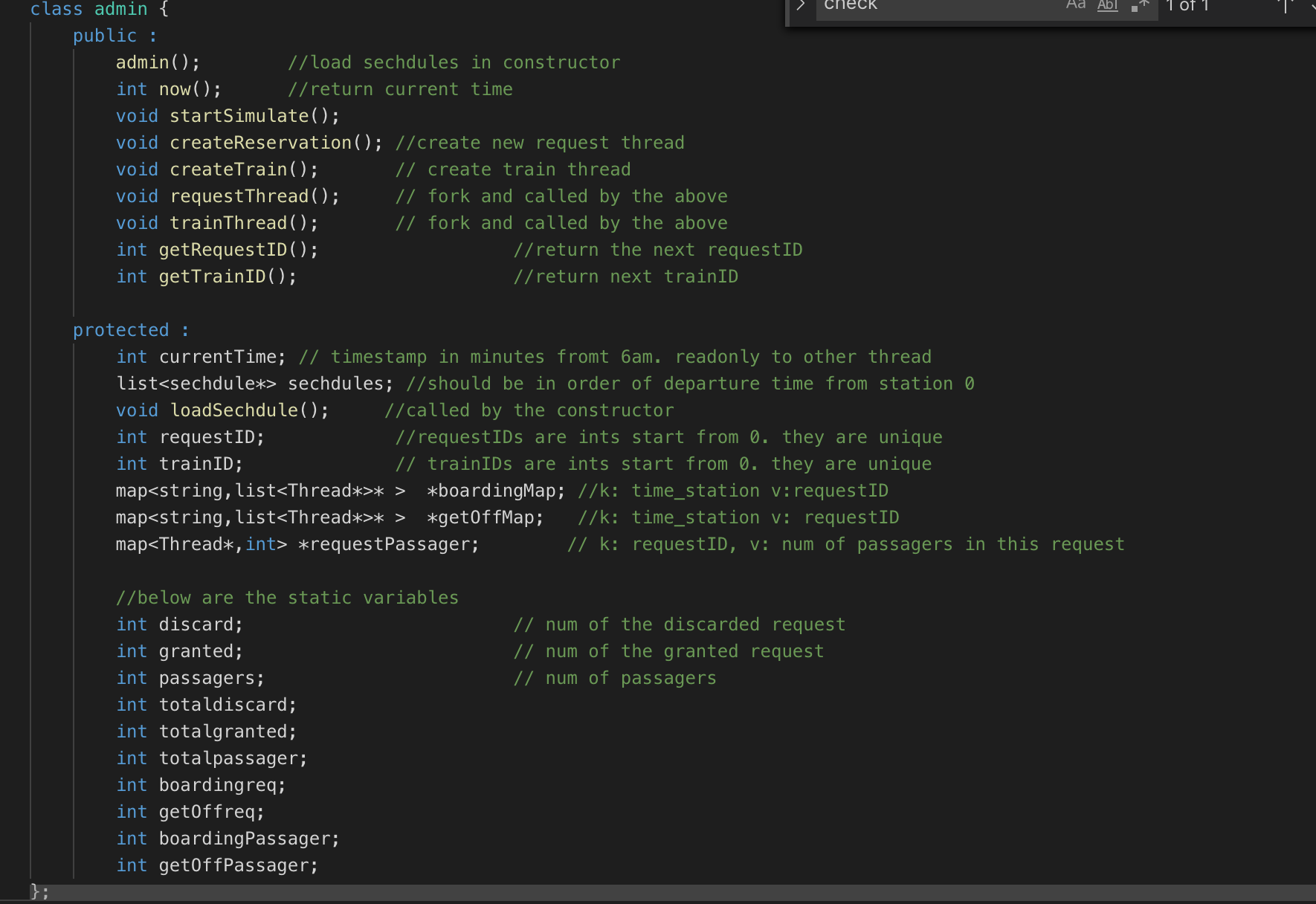
**(Yes, in my design. I do not have a train class. Instead, I stroe the information of each train as sechdules in the admin center. The admin center class has a list of sechdules)**

****

1. the admin center class{}

This is the class that serves as the admin center of a train system. It holds all the sechdules of trains, including the seats reservation information of each train. It also has member functions that simulate the action of create reservation request and running a train(get passagers on and off).

What ‘s more , on top of that. It has a startSimulation() function which simulate the time flow and schedule the threads in the entire process.



The figure above shows the structure of class admin();



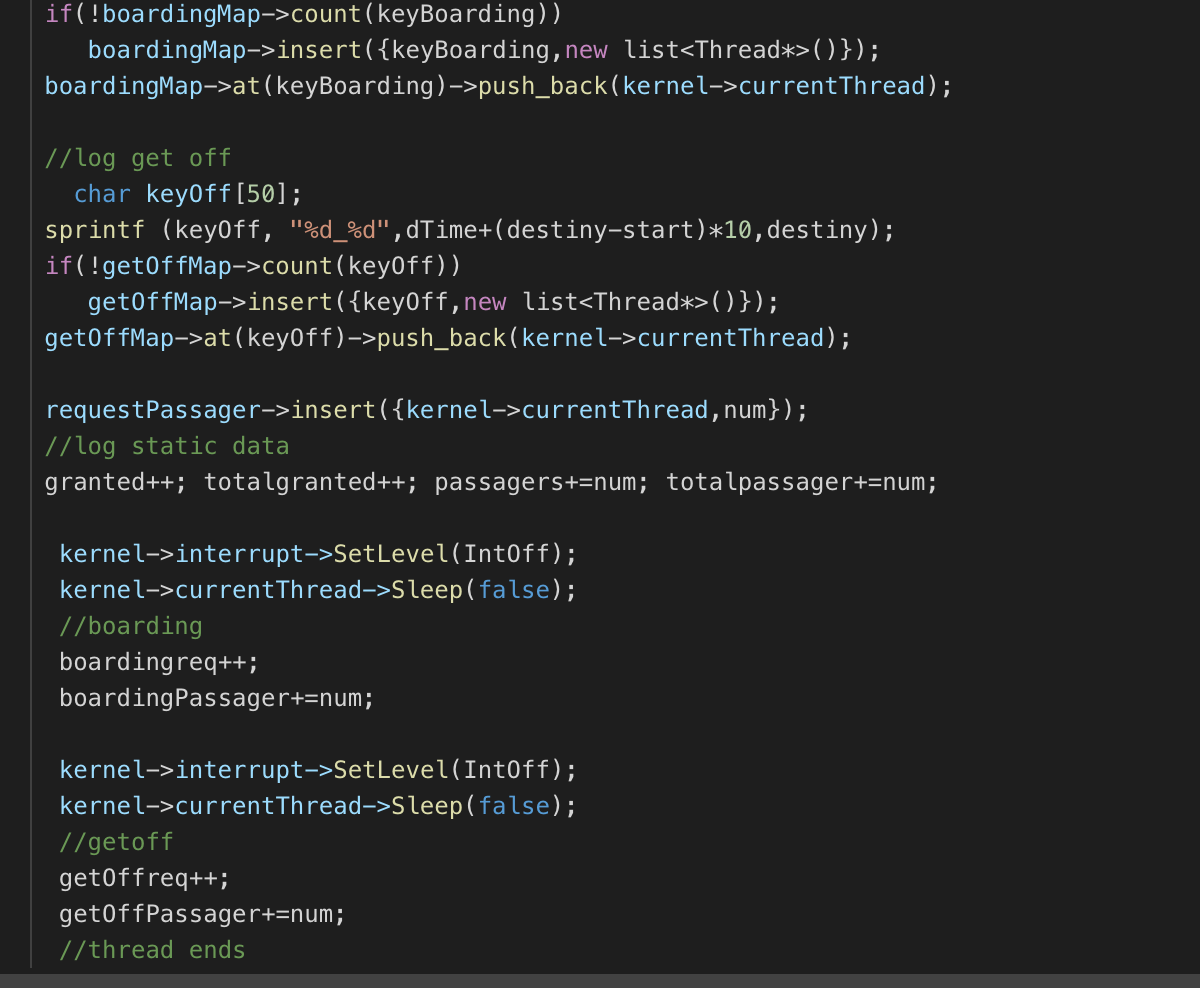
This figure shows the main entrance of the simulation.

In the createReservation() function, it generate 5 request(with 5 forked threads) every 10 min.

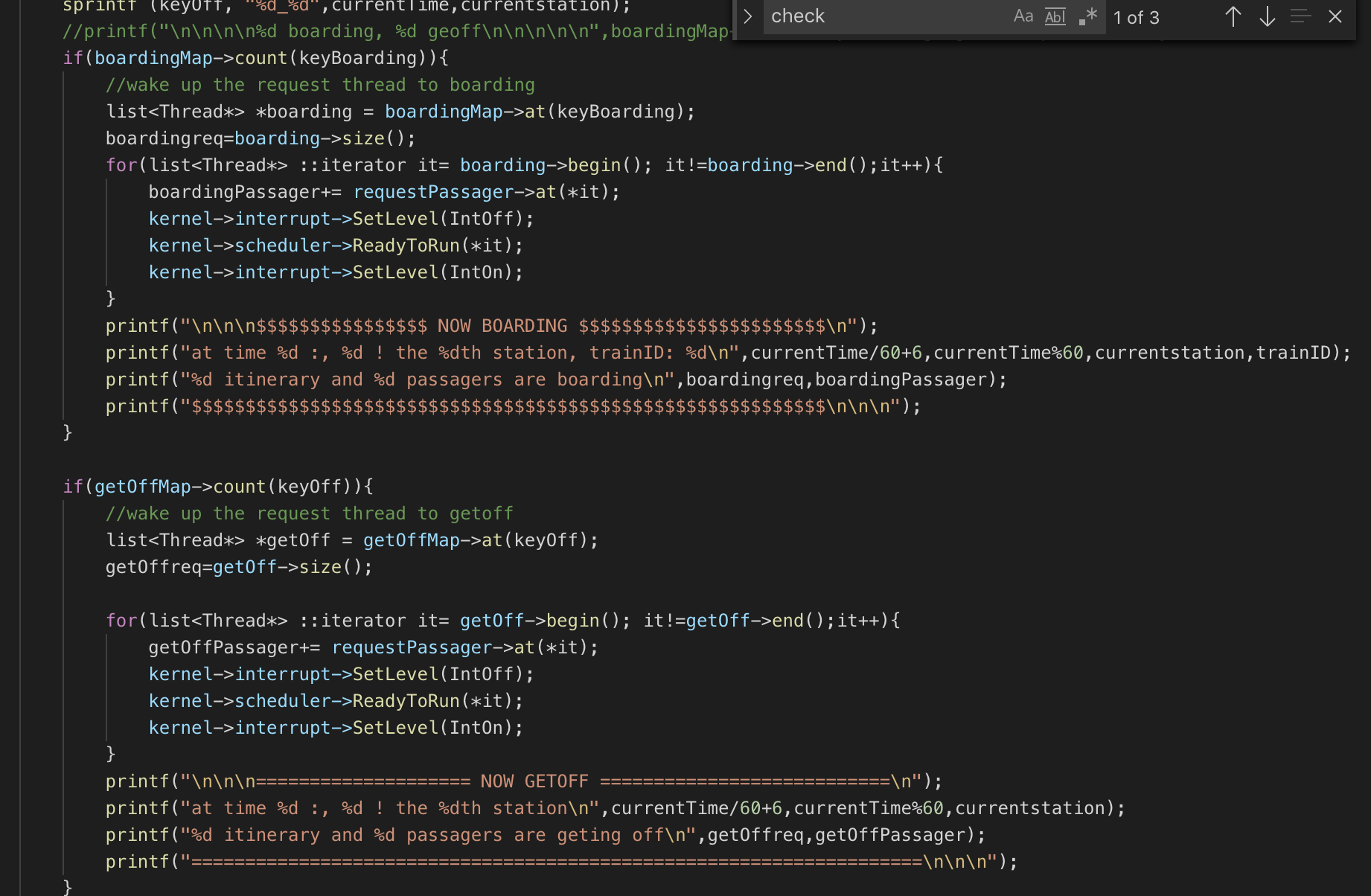
If the request is denied, the thread dies at instantly.

If the request is granted, it will sleep until the train thread(created by the createTrain(),according to the schedules loaded from the files) to wake it up when the passagers shold boarding or get of the train. After the passengers in the request get off the train and finish their trip. The request thread is distoried.

The logic that print out the static data is also embeded in the admin center.



In the request thread. It store the information of when and where to aboard and getoff to a map. And then sleep until the train thread wake it up.



Wake up the request thread in train threads.

**PART3 : build instruction**

**Cd to the build.linux folder.**

**$ Cd code/build.linux**

**$ make depend**

**$ make**

**$ nachos -K**