CS 4323
Project Progress Report

Members:
Parker Hague
Rasheed Abid
Thomas Okonkwo

Submitted on: 04/19/2021

Project Assumptions

1. The number of seats in each train is 30.
2. There will be a total of n=5 servers in the system.
3. The name of the customer is one word and their government ID is unique. This government ID is the confirmation to the purchase/booking of the ticket.

Extra steps we considered for the project

1. If the number of seats given by the customer is greater than the number of seats remaining, the system will ask the customer again until he inputs a valid number.
2. The reservation will generate an unique receipt id, but the modifying/cancelation of the ticket will also generate unique receipt ids, which will be traceable back to the root with the trailing number.
3. The users can select desired seat numbers from available seats. During selection, the menu will display the train and available seats at that moment.

Work Distribution

Parker:

- TCP connection between client and server
- Creating N number of servers via creating new processes
- Creating threads and threadpool

Rasheed:

- Server section code
- Menu and selection options
- User inputs and queries on the server side.

Thomas:

- Available seats for the train reservation
- Reserve the seat
- Inquire seats: returns the customer details and seats

Work Completed So Far

Parker:

- TCP connection between client and server
- Creating N number of servers via creating new processes
- Creating threads to handle connections

Rasheed:

- Menu implementation
- Single User interaction with the server and the files
- Printing unique id receipts and sending them to files as required.

Thomas:

- Demo: availableFunction
- Demo: reserveSeats
- Demo: inquireTicket

What's Left to Do

Parker:

- Work out the bugs in my threadpool queue to get the threadpool working

Rasheed:

- Threadpool and semaphore implementation in the server for multiple users in a single server

Thomas:

- availableFunction
- reserveSeats
- inquireTicket

Limitations and Remedies

Parker:

- Right now I can spawn a new thread to handle each connection. I need to finish the implementation of the threadpool so we don't have to create a new thread for each new client connection.

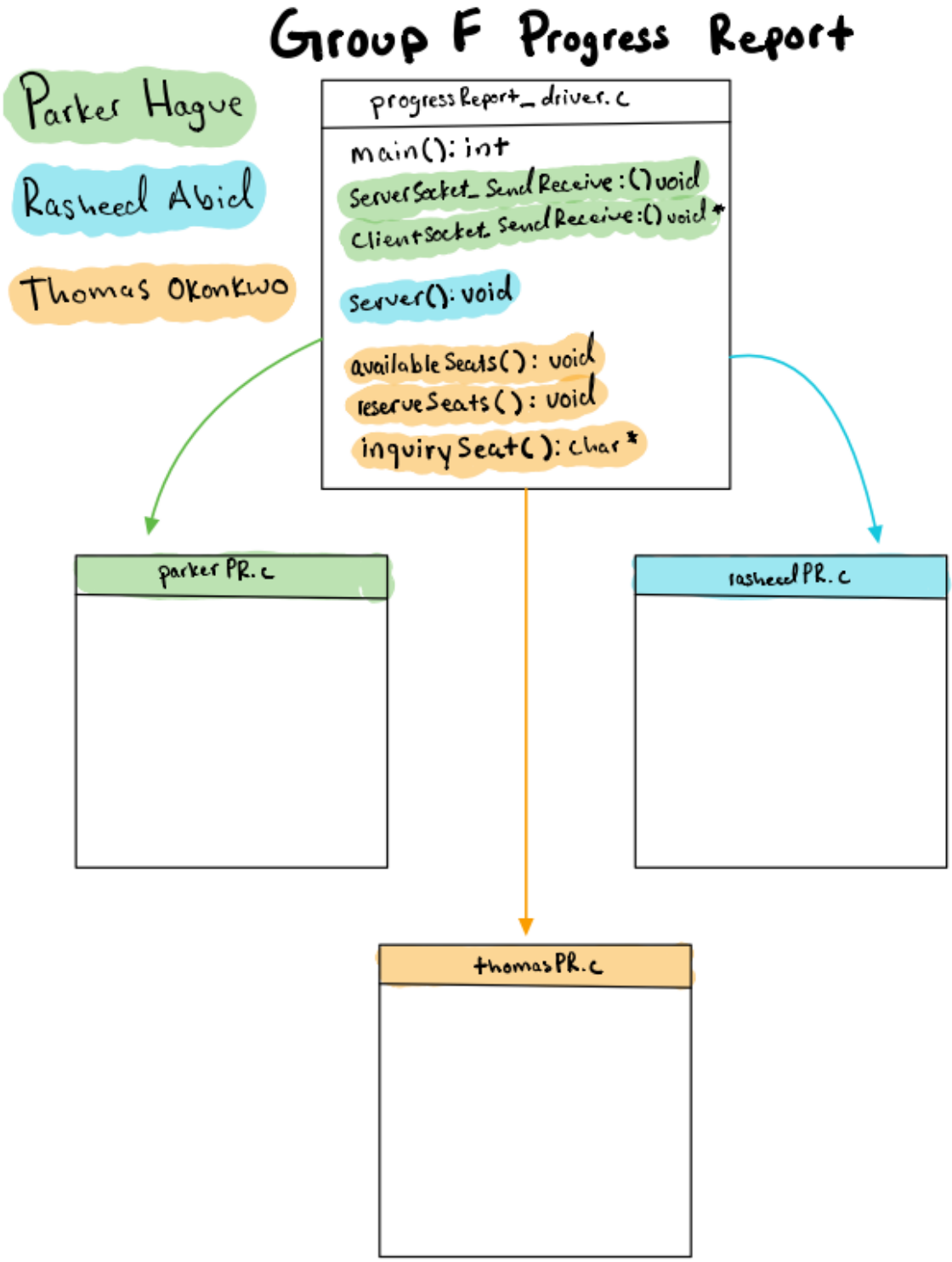
Rasheed:

- Synchronizing the system between multiple users. We will take a stepped approach towards this problem. Implement a semaphore on two processes and slowly step up as we solve each problem.

Thomas:

- Retrieval of information from the file - unable to return accurate number of seats when more than one customer comes to reserve a seat
- File handling problem

UML Diagram


Group F Progress Report
Parker Hague
Rasheed Abid
Thomas Okonkwo
progressReport_driver.c
main(): int
ServerSocket_SendReceive:()void
ClientSocket_SendReceive:()void*
server(): void
availableSeats(): void
reserveSeats(): void
inquirySeat(): char*
parkerPR.c
rasheedPR.c
thomasPR.c