```
#include "caleb_server.h"
#include "andrew_serverFuncs.h"


// int main(int argc, char const *argv[]) {
//   // This is is a test file for creating a user UI
//   readFromUser();

//   return 0;
// }
#define STRING_BUFFER_MAX 300//for tcp

int mainMenu(int socket){
    printf("\nMain menu called.\n");
    char stringBuffer[STRING_BUFFER_MAX];

    //send train string
    strcpy(stringBuffer,"\n      __       _____________\n     /  \\        ========   _________________\n
||         =      =  /          ]\n  ___==============      = /           ]\n  \\_[ Group I    ========= [
]\n
[\\=====================^==============\n___//_(_)_(_)_(_)___\\__/_____(_)_(_)_(_
)\n");//=========================================\n\n\n
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    //actual menu parts
    strcpy(stringBuffer,"Hello User! Welcome to the Group I train ticket reservation system!\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    strcpy(stringBuffer, "1. Make a reservation\n2. Inquiry about the ticket.\n3. Modify the
reservation.\n4. Cancel the reservation.\n5. Exit the program\n\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    //receive response via tcp
    strcpy(stringBuffer,"needint"); //code that customer will read and no to scanf for int
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    int intInput; //int buffer to hold client main menu input
    recv(socket,&intInput,sizeof(int),0);
    if (intInput == 5) {
        return 5;
    }
    else if (intInput == 4) {
        return 4;
    }
    else if (intInput == 3) {
        return 3;
    }
    else if (intInput == 2) {
```

```
        return 2;
    }
    else if (intInput == 1) {
        return 1;
    }
    else {
        strcpy(stringBuffer,"isn't a valid input, please try again!\n");
        send(socket,stringBuffer, sizeof(stringBuffer), 0);
        return 0;
    }
    return 0;
}

customerInfo reservationMenu(int socket){
    printf("reservationMenu() called\n"); // for debugging
    char stringBuffer[STRING_BUFFER_MAX];
    customerInfo nextCustomersInfo;

    // // Get full name
    char firstname[20];
    char lastname[40];

    strcpy(stringBuffer,"Please enter your First name\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    //receive response via tcp
    strcpy(stringBuffer,"needstring"); //code that customer will read and no to scanf for int
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    recv(socket,&stringBuffer,sizeof(stringBuffer),0);

    strcpy(nextCustomersInfo.fullName, stringBuffer);

    strcpy(stringBuffer,"Please enter your Last name\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    //receive response via tcp
    strcpy(stringBuffer,"needstring"); //code that customer will read and no to scanf for int
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    recv(socket,&stringBuffer,sizeof(stringBuffer),0);

    strncat(nextCustomersInfo.fullName, " ", 2);
    strncat(nextCustomersInfo.fullName, stringBuffer, sizeof(nextCustomersInfo.fullName));

    strcpy(stringBuffer,"Please enter your Date of Birth. (MM/DD/YYYY Format)\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);
```

```
//receive response via tcp
strcpy(stringBuffer,"needstring"); //code that customer will read and no to scanf for int
send(socket,stringBuffer,sizeof(stringBuffer),0);
recv(socket,&stringBuffer,sizeof(stringBuffer),0);

strcpy(nextCustomersInfo.dateOfBirth, stringBuffer);



strcpy(stringBuffer,"Please enter your Gender\n");
send(socket,stringBuffer,sizeof(stringBuffer),0);

//receive response via tcp
strcpy(stringBuffer,"needstring"); //code that customer will read and no to scanf for int
send(socket,stringBuffer,sizeof(stringBuffer),0);
recv(socket,&stringBuffer,sizeof(stringBuffer),0);

strcpy(nextCustomersInfo.gender, stringBuffer);



strcpy(stringBuffer,"Please enter your Government ID number\n");
send(socket,stringBuffer,sizeof(stringBuffer),0);

//receive response via tcp
strcpy(stringBuffer,"needstring"); //code that customer will read and no to scanf for int
send(socket,stringBuffer,sizeof(stringBuffer),0);
recv(socket,&stringBuffer,sizeof(stringBuffer),0);

strcpy(nextCustomersInfo.governmentID, stringBuffer);



strcpy(stringBuffer,"Are you reserving for:\n1.Today\n2.Tomorrow\n");
send(socket,stringBuffer,sizeof(stringBuffer),0);

//receive response via tcp
strcpy(stringBuffer,"needint"); //code that customer will read and no to scanf for int
send(socket,stringBuffer,sizeof(stringBuffer),0);
recv(socket,&nextCustomersInfo.dayOfTravel,sizeof(int),0);


strcpy(stringBuffer,"How many people are in your party?\n");
send(socket,stringBuffer,sizeof(stringBuffer),0);

//receive response via tcp
```

```
    strcpy(stringBuffer,"needint"); //code that customer will read and no to scanf for int
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    recv(socket,&nextCustomersInfo.numberOfTravelers,sizeof(int),0);

    printCustomerFromStruct(nextCustomersInfo);

    /*strcpy(stringBuffer,"end"); //end code to be sent to client, client will then know to call its
own exit function
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    sleep(4);
    exit(0);*/

    return nextCustomersInfo;
}

bool confirmReservationMenu(int socket){
    printf("confirmReservationMenu() called\n"); //for debugging
//return false if they do not confirm, could say reservation not confirmed or something
    char stringBuffer[STRING_BUFFER_MAX];

    strcpy(stringBuffer,ANSI_COLOR_GREEN "Confirm reservation? (yes/no)"
ANSI_COLOR_RESET "\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    //receive response via tcp
    strcpy(stringBuffer,"needstring"); //code that customer will read and no to scanf for int
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    recv(socket,&stringBuffer,sizeof(stringBuffer),0);

    if (strcmp(stringBuffer, "yes") == 0) {
      strcpy(stringBuffer,"Reservation Confirmed.\n"); // Letting user know that their reservation
was confirmed
      send(socket,stringBuffer,sizeof(stringBuffer),0);
      return true;
    }

    printf("-d input was not yes, it was %s\n", stringBuffer); // for debugging purposes

    strcpy(stringBuffer,"Reservation not Confirmed.\n"); // Letting user know that their
reservation was not confirmed
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    return false;
}

//will ask for ticket customer via tcp for ticket number, returns ticket number
```

```
int ticketInquiryMenu(int socket){
  printf("ticketInquiryMenu() called\n"); //for debugging
  int ticketNumber = requestInt("Please enter your ticket number for lookup.\n", socket);

  return ticketNumber;
}

//asks what fields customer want to modifiy, returns int based on choice
int modifyReservationMenu(int socket){
   printf("modifyReservation() called\n"); //for debugging

   int option = requestInt("\nWhich would you like to modify:\n1.Seat\n2.Travel Date\n3.Size of
party\n",socket);

   if (option == 1) {
     return 1;
   }
   else if (option == 2) {
     return 2;
   }
   else if (option == 3) {
     return 3;
   }
   else {
     sendMessageToClient("\nNothing changed!\n", socket);
   }

   return 0;
}

// //asks what fields customer want to modifiy, returns struct holding customers modified info
// //have to get ticket number to use to search summary files
// customerInfo modifyReservationMenu(int socket){
//    printf("modifyReservation() called\n"); //for debugging
//    customerInfo customersMods;  //struct that holds modfied info
//    int ticketNumber = requestInt("Please enter your Ticket Number:",socket);
//    customersMods.ticketNumber = ticketNumber;

//    sendMessageToClient("\nPulling up reservation now . . . \n", socket);

//    int option = requestInt("\nWhich would you like to modify:\n1.Seat\n2.Travel Date\n3.Size
of party\n",socket);

//    if (option == 1) {
//      sendMessageToClient("\nSeat Changed!\n", socket); // will need more info on seats
//
```

```
//     }
//     else if (option == 2) {
//       customersMods.dayOfTravel = requestInt("\nWhen would you prefer to
travel:\n1.Today\n2.Tomorrow\n",socket);
//     }
//     else if (option == 3) {
//       customersMods.numberOfTravelers = requestInt("\nHow many people are in the
party?\n",socket);
//     }
//     else {
//       sendMessageToClient("\nNothing changed!\n", socket);
//     }

//     return customersMods;
// }


//cancel confirmation sent over tcp if customer sends back yes then returns true, else false
bool confirmCancellationMenu(int socket){
    printf("confirmCancellationMenu() called\n"); //for debugging
    char stringBuffer[STRING_BUFFER_MAX];
    int cancel;

    strcpy(stringBuffer,ANSI_COLOR_RED "Confirm Cancellation?\n1.Yes\n2.No"
ANSI_COLOR_RESET "\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    //receive response via tcp
    strcpy(stringBuffer,"needint"); //code that customer will read and no to scanf for int
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    recv(socket,&cancel,sizeof(int),0);

    if (cancel == 1) {
      return true;
    }
    return false;
}

void sendMessageToClient(char *message, int socket){
  char stringBuffer[STRING_BUFFER_MAX];
  strcpy(stringBuffer,message);
  send(socket,stringBuffer,sizeof(stringBuffer),0);
}
int requestInt(char *message, int socket){
 int returnInt;
```

```
 char stringBuffer[STRING_BUFFER_MAX];
 strcpy(stringBuffer,message);
 send(socket,stringBuffer,sizeof(stringBuffer),0);

 //receive response via tcp
 strcpy(stringBuffer,"needint"); //code that customer will read and no to scanf for int
 send(socket,stringBuffer,sizeof(stringBuffer),0);
 recv(socket,&returnInt,sizeof(int),0);

 return returnInt;
}/*
void requestReadSemaphor(sem_t *read, sem_t *write){
  sem_wait(read);
  int readers = sem_getValue(read);
  if (readers == 4) {
    requestWriteSemaphor(write);
  }


}
void giveBackReadSemaphor(sem_t *mutex, sem_t *write){

}
void requestWriteSemaphor(sem_t *write){
  sem_wait(write);
}
void giveBackWrtieSemaphor(sem_t *write){
  sem_post(write);
}*/
void printCustomerFromStruct(customerInfo info) {
  printf("\n================================\n");
  printf("Customer name: %s\nCustomer DOB: %s\nCustomer Gender: %s\nGovernmentID
Number: %s\n",info.fullName,info.dateOfBirth,info.gender,info.governmentID);
  if (info.dayOfTravel == 1) {
    printf("Travelling today with a party of size %d",info.numberOfTravelers);
  }
  else {
    printf("Travelling tomorrow with a party of size %d\n",info.numberOfTravelers);
  }
  printf("\n================================\n");
}
void printTrain() {
    printf("\n\n\n\n\n\n\n\n\n\n      __        _____________\n     /  \\        ========
_____________\n      ||         =      =  /          ]\n  ___==============      = /           ]\n  \\_[
Group I    ========= [           ]\n
```

```
[\\=====================^==============\n___//_(_)_(_)_(_)___\\__/_____(_)_(_)_(_
)\n=========================================\n\n\n");
}
```