Project Overview

In this project, you are asked to implement an aircraft parking reservation system. Each airline representative will be able to search and reserve the available aircraft gates for a defined period of time (time slot) depending on the available number of gates and their corresponding reservations. Also, each airline representative should be able to track and log his activity while interacting with the server.

Austin-Bergstrom International Airport, Texas, US

General Requirements

In the aircraft parking reservation system, you should use a client/server system where the client component is used to manage the reservations and the server centralizes the corresponding data.
The clients should be able to login and register their accounts. Once the User (airliner representative) is logged in, he/she should be able to search for aircraft parking slots and
make reservations. The server should be able to track and save all the transactions each and every user has made and produce a report when requested.
The total number of gates available at an airport terminal shall be a parameter that can be set by the system at the server side (for example you may select 10 gates initially).
Each parking time slot is 2 hours. Thus, you can divide the 24 hours per day into 12 parking time slots.

Client Features:

A client can:

1. view the aircraft parking gates available
    - a gate is available if at least one parking time slot is not reserved at this gate
2. view the list of parking time slots available in each gate
3. search a particular parking time slot. If the parking time slot is available, display the list of gates that are available during the requested time slot
4. reserve an aircraft parking time slot
5. view the history of his interactions with the server

Server Features:

The server should:

1. provide users with an interface where they can create accounts and log in to them. Try to implement features that you may have encountered on actual servers when creating accounts, like password verification, human verification (not a robot), etc.
2. manage the accounts that are stored in a MySQL database.
    You should create the database in MySQL and its tables to store the users data and the tracking information that the library server generates
3. be multithreaded in order to serve multiple clients concurrently
4. inform the user that the gate is available or occupied during a time slot
5. generate an activity report for user activities within a specified time interval