Lab 2 Calendar - Tanner Juby - Design Document

Version: 3/6/17

This document is a design of the Lab 2 - Calendar program for CSCI 4761 course in the spring semester of 2017 taught Dr. Ra. This project was written by Tanner Juby

1. Introduction

This document will go through the structure of the program and how to utilize the program.

2. Architecture Design

There are two program files: *client.cpp* and *server.cpp.*

The client.cpp file is simple and just communicates strings to the server.cpp file.

The server.cpp file is the more advanced one that talks to the database and executes functions based on what the user sends in. The server utilizes a number of functions outside of the main function to help it with all the work:

- bool isLoggedIn(char* id)
- char* login(char* email, char* password)
- bool logout(char* id)
- char* createAccount(char* email, char* password, char* firstName, char* lastName, char* phone)
- char* deleteAccount(char* id, char* password)
- char* editAccount(char* id, char* currentPassword, char* email, char* newPassword, char* firstName, char* lastName, char* phone)
- bool isValidDate(char* date)
- bool isValidTime(char* time)
- int compareDates(char* tempDate1, char* tempDate2)
- int compareTimes(char* tempTime1, char* tempTime2)
- char* createEvent(char* key, char* name, char* date, char* startTime, char* endTime)
- char* deleteEvent(char* id, char* key, char* password)
- char* editEvent(char* id, char* key, char* password, char* name, char* date, char* startTime, char* endTime)
- char* getEventByTime(char* key, char* password, char* date, char* time)
- char* getEventsByTimeRange(char* key, char* password, char* date, char* startTime, char* endTime)

3. Database Design

There are three files for the database: *User.csv, Appointment.csv*, and *Sessions.csv*

User.csv handles all of the records for user accounts.

Appointment.csv handles all of the records for appointments in the calendar

Sessions.csv holds the current users that are logged in.

Tables schemas

User.csv

ID,EMAIL,PASSWORD,FIRST NAME,LAST NAME,PHONE

Appointment.csv

ID,USER ID,NAME,DATE,START TIME,END TIME

Sessions.csv

ID

4. Graphical User Interface

- CommandLine

```
How to communicate with the server:
ACTION                 MESSAGE TO SEND
------                 ---------------
Login                  login <email> <password>
Logout                 logout <key>
Add Account            add_acount <email> <password> <first_name> <last_name> <phone>
Delete Account         delete_account <key> <password>
Edit Account           edit_account <key> <current password> <email> <new password> <first_name> <last_name> <phone>
Add Event              add_event <key> <name> <date (MM-DD-YYYY)> <start_time (HH:SS)> <end_time (HH:SS)>
Delete Event           delete_event <key> <password> <event_id>
Edit Event             edit_event <key> <password> <event_id> <name> <date (MM-DD-YYYY)> <start_time (HH:SS)> <end_time
(HH:SS)>
Event By Time          event_by_time <key> <password> <date (MM-DD-YYYY)> <time (HH:SS)>
Events By Time Range   events_by_time_range <key> <password> <start date (MM-DD-YYYY)> <start time (HH:SS)> <end time
(HH:SS)>


Instructions:
        1) Copy the MESSAGE TO SEND string into message field
        2) Replace all fields surrounded by '< >' with the value you want for that field
                (EXAMPLE: "login <email> <password> would be" ---> "login myEmail@sample.com myPassword1234")
        3) Press 'enter' and wait for response.
 *****
NOTE: Any action you wish to do with your account (besides login and create) you need your account KEY. This will be provided when
you login or create an account
*****
Input:
```

5. Class Diagram and Classes

No classes. Just a server and a client

6. Design process

Using the server.c and client.c files Dr. Ra provided, I added in my functionality. The template he provided simply took a string from the server, and sent it back. Instead, I took the string and separated it out by spaces. The first part of the string being an action, and the rest depending on what the action is. Depending on the action the client send, I make different calls to perform said action. Once complete, I send back the results from the action.

7. References

Dr. Ra’s client.c and server.c example code he provided.
stackoverflow.com
cplusplus.com