MY CAL APPLICATION

Project Final Report

Presented To: Prof. Zaher Dawy

Done By:
Rashad Saab
Ousama Kanawati

Spring 2015

EECE 350 : COMPUTER NETWORKS

Table of Contents

List of Figures

Code Structure and Organization

In order to use MyCal, the user should create an account (Sign up). He should provide a an unused Username, valid Email address, birthday date, mobile phone number, gender and a password. Now the user can log in.

We added two classes Account and Event for readability. Also we had three classes for the GUI Design LogInGUI , MyCalGUI , and NewEventGUI. We added TCPClient_CNX_Manager class as a client application class, Server class as the server application class, and database class as the database class.

Client Class:

The code is build based on an organized way where there is the main function called “Connect to Server” where every button in the GUI sends a specific “request” to this function that manages sending this request to the server and waits for the corresponding reply. This reply will be finalized by the local client application GUI classes and easily presented on an interactive interface.

Server Class:

The server manages the connection functions, allows multi thread handling and connects to the database tables. The server works on replying to clients requests and calling the specific database function.

The server code works mainly using two basic steps. The first step is splitting the message received at ";" that allow us to figuring out the client request and provide us with the username and password. The second step is using a case structure that calls the specific function in the server, this function handles the connection with the client and calls the database function to execute the client request. The returned object will be send to the client by the server function.

Requests Implemented:

SignUp: Add the client info to the database with an unused username.

LogIn: Authenticate the username and the password of the client. Client is online.

NewEvent: Create a new event with the specific info set by the client .

UpdateAccount: Update the client account info (email, mobile number).

GetEvents: Fetch the events that the client in going to from the database.

GetInvitedEvents: Fetch the events that the client in invited to from the database.

AcceptEvent: The client accepts a specific event.

RejectEvent: The client rejects a specific event.

SignOut: Client is offline.

LoadOnlineFriends: The server sends an ArrayList of the client online friends.

LoadFriends: The server sends an ArrayList of the client friends.

LoadUsernames: The server sends the usernames of all the clients.

AddFriend: The client sends an friend invitation to a specific username.

UpdateEvent: The client updates the info of an event.(Date,time,rate,comment,title,location)

GetFriendRequests: The client fetches the friend requests that the client received .

AcceptFriendRequest: The client accepts a specific friend request.

RejectFriendRequest: The client rejects a specific friend request.

Database Class:

In our project the database is divided into four tables. “Status”, “Accounts”, "Invitations", and “Friends” where these four tables are used by the database functions to manage the server requests and the client requests as well.

Account	**Status**	**Friends**	**Invitations**
AccountID	StatusID	FriendsID	InvitationID
Username	Username	Username	Username
Gender	Online	friendname	Rejected
Email		RequestAccepted	InvitationTo
Birthday		Request Rejected	Accepted
Displaypic			EventComment
Mobilephone			EventTitle
newFrndRequests			EventLocation
newMsgs			EventTime
nwEventInvites			EventDate
nbrOfFriends			EventRate

Application Functionality

The application we created is called "MyCal", it is a social calendar network application used to organize events and schedule meetings. The application implementation is divided in to two modules server/client. The sending/receiving of data is over TCP multi threaded connection.

The client application involves an elegant calendar GUI interface were the client is able to connect to the server , login with a username/password if he already registered ,and sign up to enter his personal information if he isn't registered yet. Upon signing up a welcoming email will be sent to the client's email. When the client logs in a home screen will be displayed were he can add friends from users list , create events and invite friends to this event, and delete/reschedule events. Also, the client can view events he have been invited to and accept/reject these invitations. Upon creating an event the client can comment and rate his event and he can view the comments and ratings of each event. Client can accept a friend request or reject it. He can login to twitter in order to share any event as tweet.

The server application keeps all the information about the accounts and events in a database were the user can edit his personal settings. The server is supported by multithreading in order to deal with multiple clients at the same time.

Bonus Tasks

- **Real-time log** of the server activities.
- Ability to set a **profile picture** for your account.
- Exchanging username and password between server and client is done using **encrypted messages**.
- Upon signing up, we implemented a function that checks the **password reliability** and force the client to include a reliable password (8 alphanumeric char with special characters)
- We added a **background** to the GUI with the ability to choose between several options.
- We implemented a **delete account** function.
- Organization and creativity of the GUI were we placed **inspirational quotes** for the client to help him organize his life along with his time.
- We implemented upon signing up a function that sends to the client a **welcoming email** describing for him our application and how to use it.
- We implemented a function that send the client a **reminder email** for his upcoming events in the next day.
- We implemented the functionality of signing in to twitter through our application and **sharing event on twitter** as tweets.

Role of each group member

Our group has a good team work spirit that allowed us to reach this point in the project at this time. Currently, Rashad was responsible for the GUI part. He implemented the client function call in each GUI button and developed the connection between the client and the server from the client side. Also, he implemented twitter sharing feature, reminder email, and GUI extra features. Moreover, Ousama developed the database functions and designed the connection between the client and the server in addition to that he implemented it from the server part. Also, he was responsible for the reports.

Each member contributed by 50% to the full project development.

How to run the code

1. Add the jar files as external jars to your library,
2. Run the server class,
3. Run LogInGUI class.

Snapshots

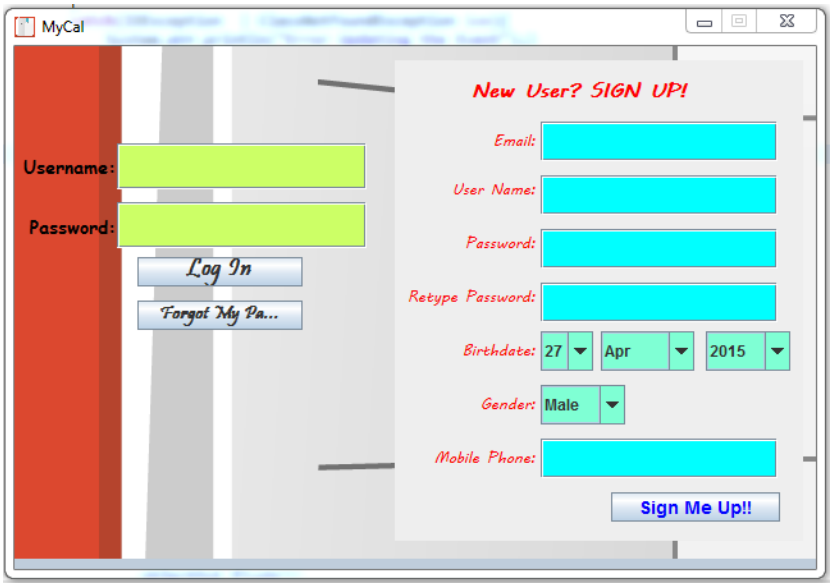

Figure 1: Login/Signup Screen

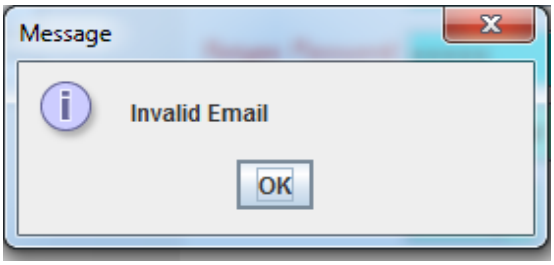

Figure 2: Password Strength pop-up

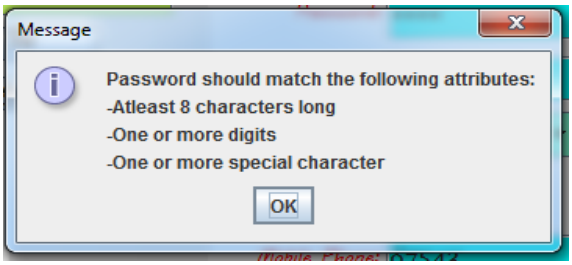

Figure 3: Invalid Email pop-up

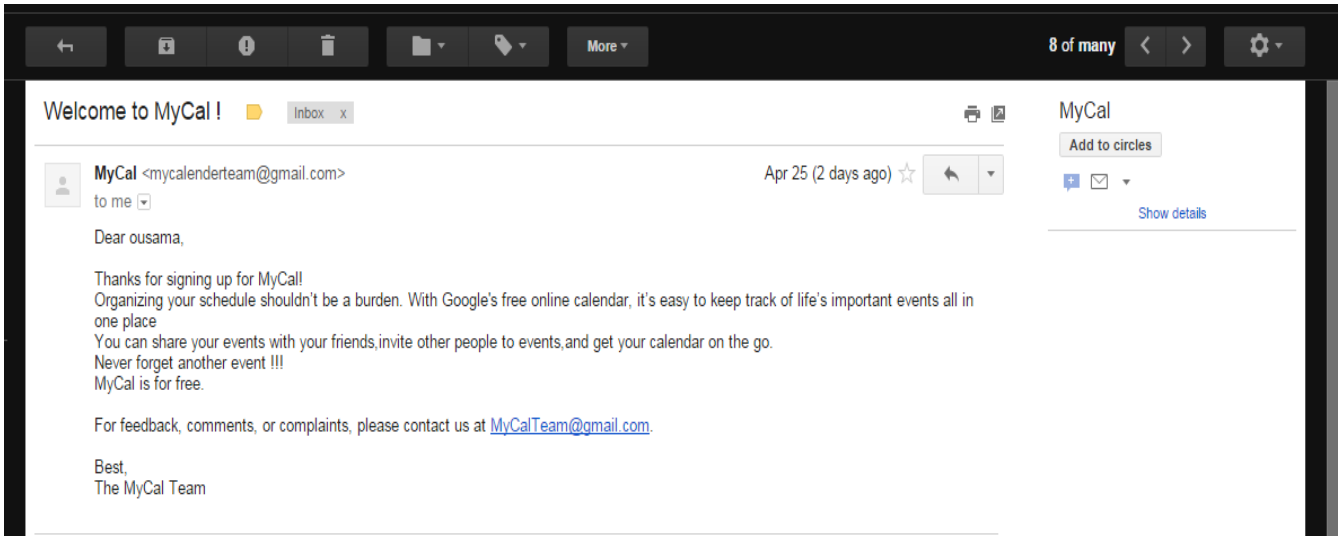

Figure 4: Welcome Email

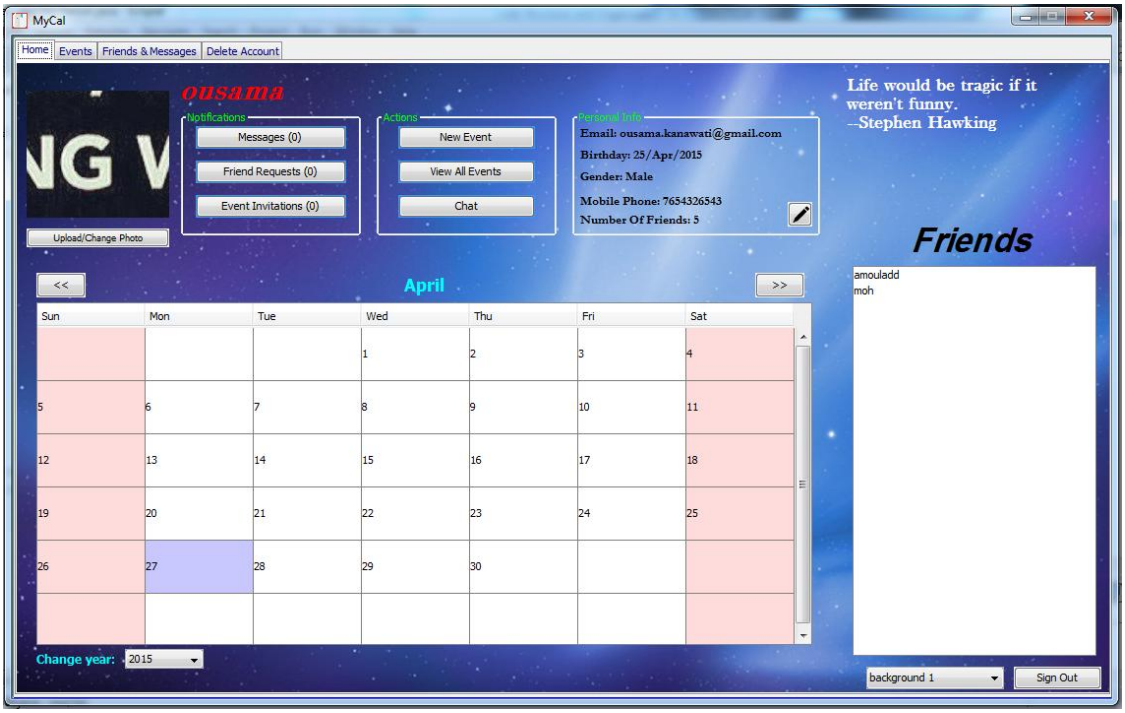

Figure 5:Home Tab

Figure 6: Creating a New Event

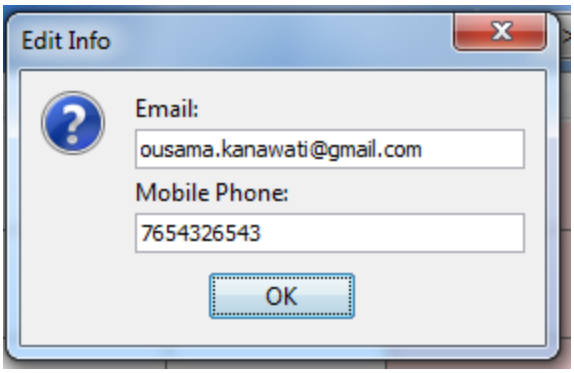

Figure 7 Editing Profile Info

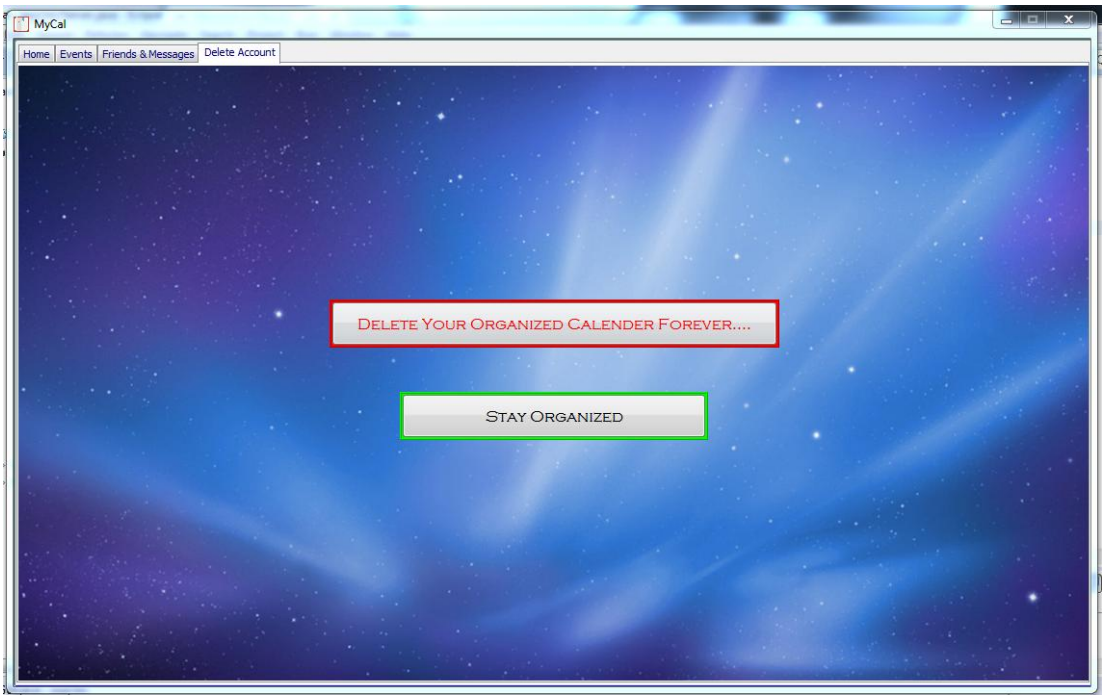

Figure 8: Delete Account Tab

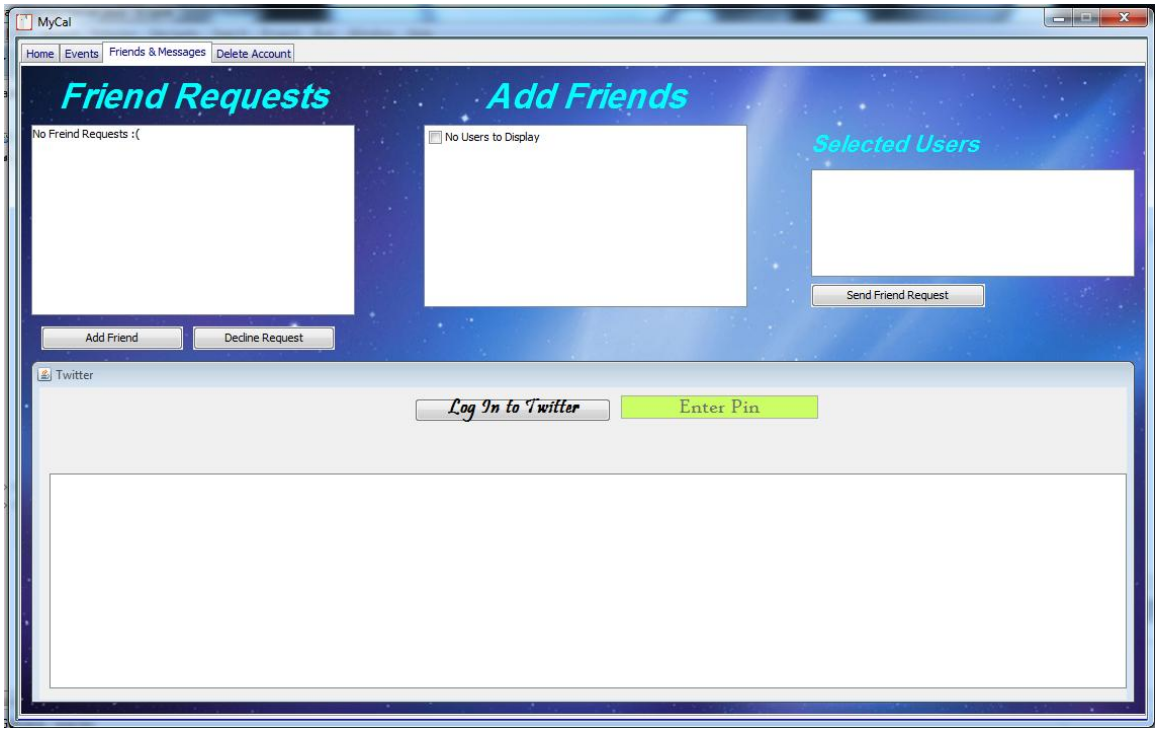

Figure 9: Friends and Messages Tab

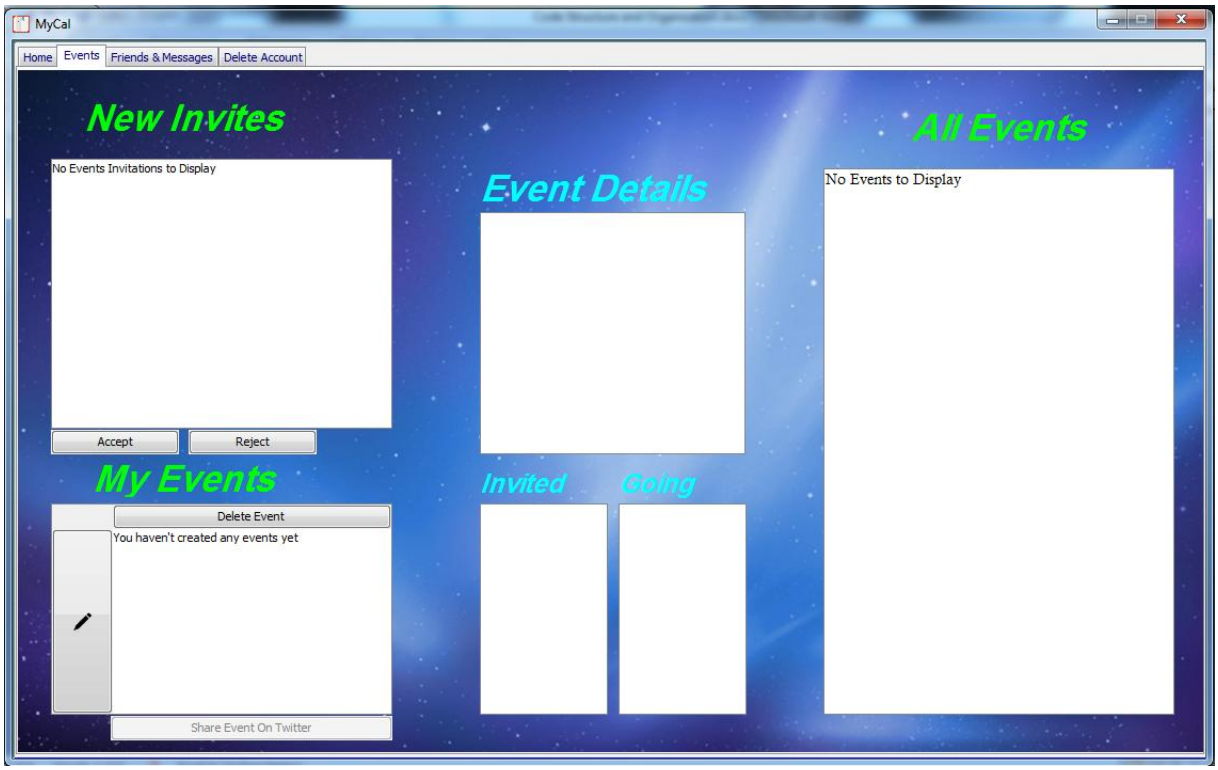

Figure 10: Events Tab

Design Decisions

- Client-Server request messages.
- Storing events in the database as invitations with indicators to host and invited people, and accepted or rejected
- Allowed the user to change only his email, phone number and profile picture.

Reference:

StackOverow, Programming Q&A: http://www.stackoverow.com

SQL Tutorial: http://www.w3schools.com/sql

GUI design: http://java.sun.com/docs/books/tutorial/uiswing/index.html

JAVA docs: http://java.sun.com/j2se/1.5.0/docs/api/

http://www.dreamincode.net/forums/topic/25042-creating-a-calendar-viewer-application/

http://quartz-scheduler.org/

http://twitter4j.org/en/index.html