Criterion C: Development

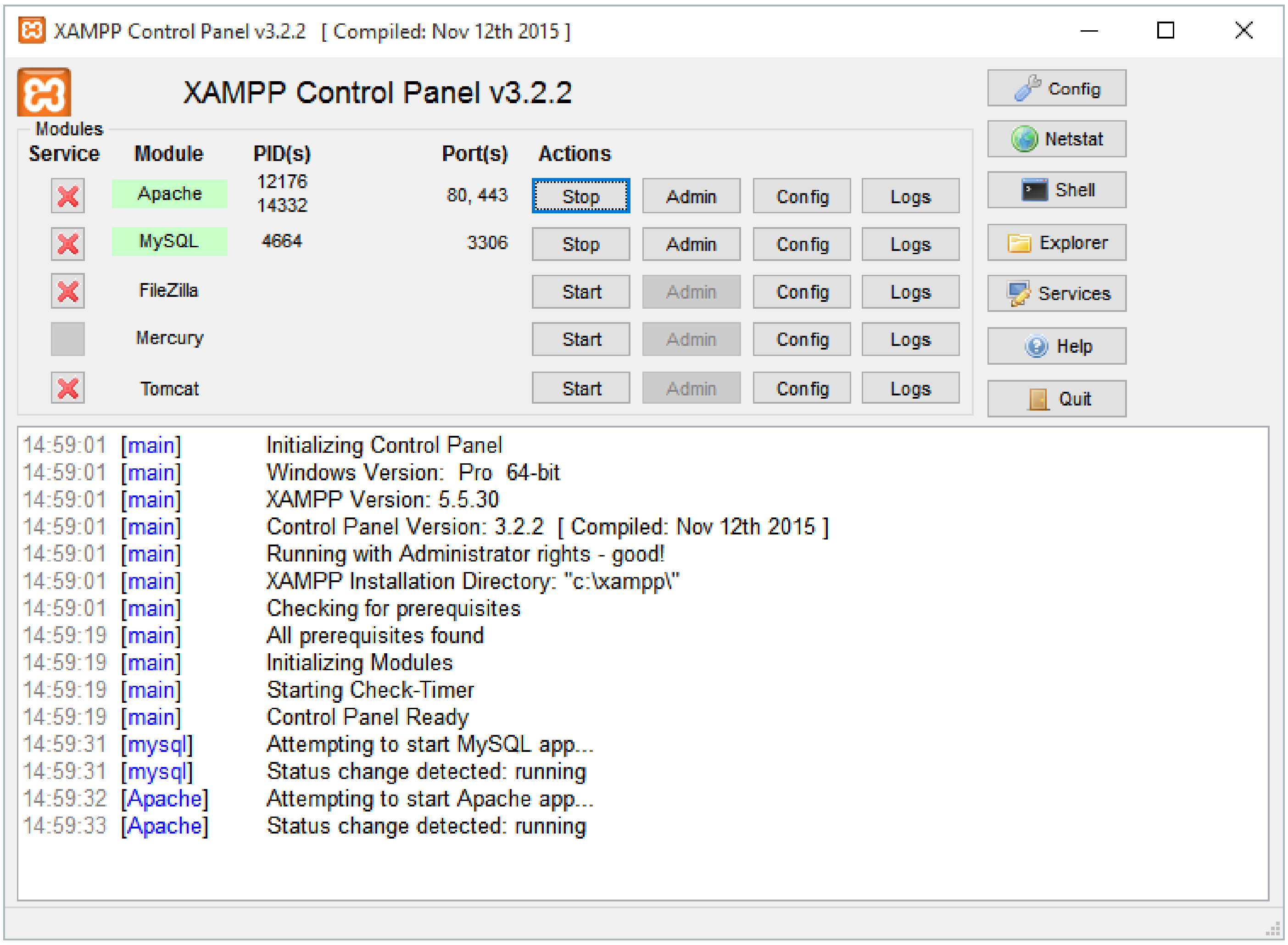
Overview

The solution uses a MySQL database to store the required data on the meetings and users whiles using Java and queries to manipulate the database and check for clashes and finally using JavaFX to create forms which serve as the user interface.

Database

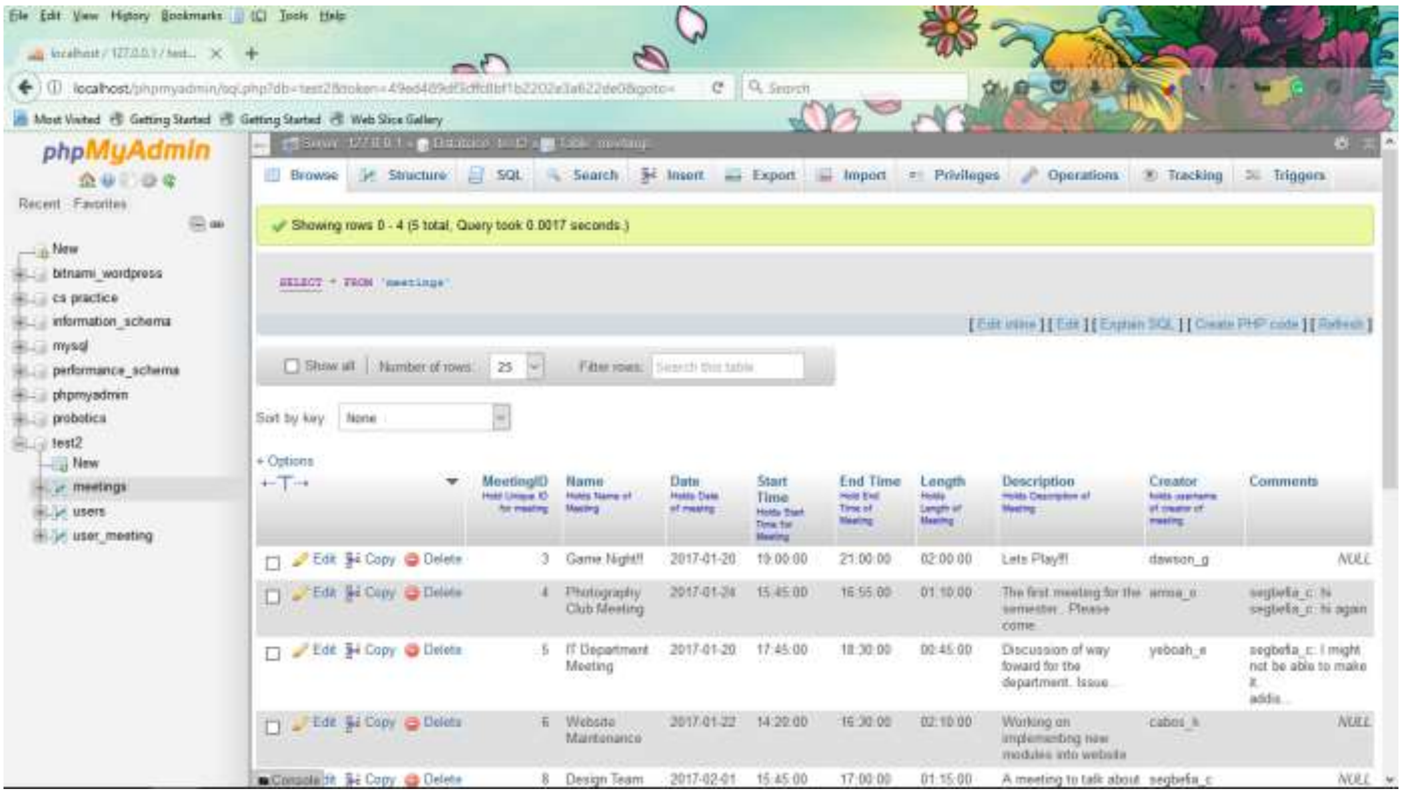
Database Tools Used

The tool 'phpMyAdmin' was used to easily view data in the database and run queries while testing. The database and 'phpMyAdmin' were run on the apache distribution 'XAMPP' (XAMPP). Screenshots are shown below.


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
✗	Apache	12176 14332	80, 443	Stop	Admin	Config	Logs
✗	MySQL	4664	3306	Stop	Admin	Config	Logs
✗	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
✗	Tomcat			Start	Admin	Config	Logs

```
14:59:01  [main]      Initializing Control Panel
14:59:01  [main]      Windows Version:  Pro  64-bit
14:59:01  [main]      XAMPP Version: 5.5.30
14:59:01  [main]      Control Panel Version: 3.2.2  [ Compiled: Nov 12th 2015 ]
14:59:01  [main]      Running with Administrator rights - good!
14:59:01  [main]      XAMPP Installation Directory: "c:\xampp\"
14:59:01  [main]      Checking for prerequisites
14:59:19  [main]      All prerequisites found
14:59:19  [main]      Initializing Modules
14:59:19  [main]      Starting Check-Timer
14:59:19  [main]      Control Panel Ready
14:59:31  [mysql]     Attempting to start MySQL app...
14:59:31  [mysql]     Status change detected: running
14:59:32  [Apache]    Attempting to start Apache app...
14:59:33  [Apache]    Status change detected: running
```



Showing rows 0 - 4 (5 total, Query took 0.0017 seconds.)

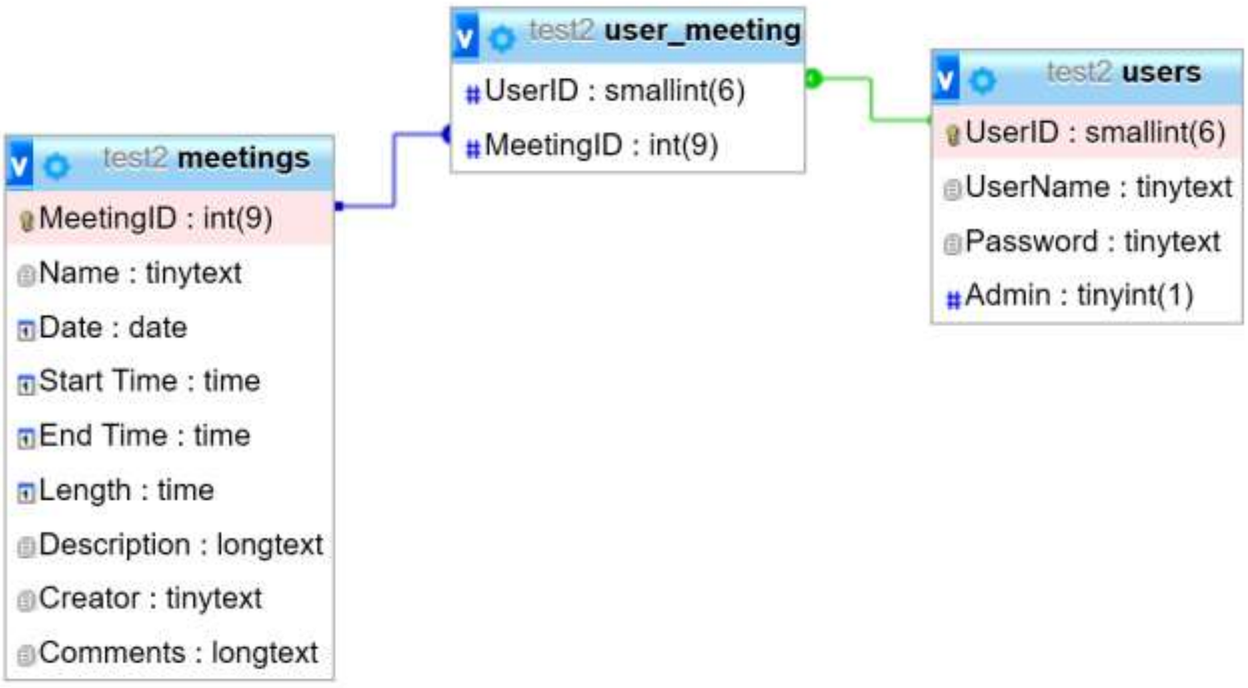
```
SELECT * FROM `meetings`
```

MeetingID Hold Unique ID for meeting	Name Holds Name of Meeting	Date Holds Date of meeting	Start Time Holds Start Time for Meeting	End Time Hold End Time of Meeting	Length Holds Length of Meeting	Description Holds Description of Meeting	Creator holds username of creator of meeting	Comments
3	Game Night!!	2017-01-20	19:00:00	21:00:00	02:00:00	Lets Play!!	dawson_g	NULL
4	Photography Club Meeting	2017-01-24	15:45:00	16:55:00	01:10:00	The first meeting for the semester... Please come	amoa_o	segbefia_c: hi segbefia_c: hi again
5	IT Department Meeting	2017-01-20	17:45:00	18:30:00	00:45:00	Discussion of way foward for the department. Issue...	yeboah_e	segbefia_c: I might not be able to make it. addis...
6	Website Maintenance	2017-01-22	14:20:00	16:30:00	02:10:00	Working on implementing new modules into website	cabos_h	NULL
8	Design Team	2017-02-01	15:45:00	17:00:00	01:15:00	A meeting to talk about...	segbefia_c	NULL

Tables

The database is made up of three tables.

The student table to store user login credentials, the meeting table to store meeting data and the user-meeting table to link each user to a meeting and each meeting to its members. All three tables are shown below.



1.Meetings

Column	Type	Comments
MeetingID	int(9)	Hold Unique ID for meeting
Name	tinytext	Name of Meeting
Date	date	Date of meeting
Start Time	time	Start Time for Meeting
End Time	time	Hold End Time of Meeting
Length	time	Length of Meeting
Description	longtext	Description of Meeting
Creator	tinytext	username of creator of meeting

2. Users

Column	Type	Comments
UserID	smallint(6)	unique user ID
UserName	tinytext	the unique name of a User in the form Lastname_firstName
Password	tinytext	Users Password
Admin	tinyint(1)	whether user is Administrator or not

3. User_Meetings

Column	Type	Links to
UserID	smallint(6)	-> users.UserID
MeetingID	mediumint(9)	-> meetings.MeetingID

Relationships Created

To remove redundant data and allow for users to be queried by their meeting and meetings to be queried by their users two relationships were made. These allow 'JOIN' statements to be used to query all the tables at once.

1. A relationship between 'Users' and 'User_Meetings' using UserID as a foreign key. It is a one to many relationship as one UserID can have many MeetingID's
2. A relationship between 'Meetings' and 'User_Meetings' using the MeetingID as a foreign key. It is a one to many relationship as one MeetingID can have many UserID's

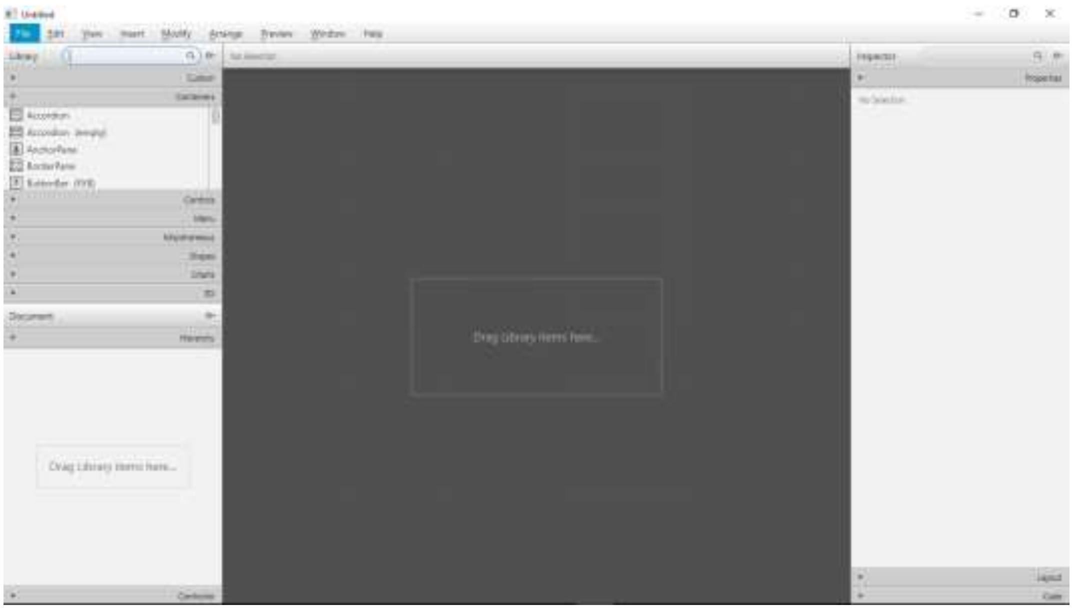
Queries

Multiple queries are performed on the database and some are listed below with the syntax used.

Query function	Query Syntax
Query to get user password based on given username	Select \`UserName\`, \`Password\` FROM users WHERE \`UserName\` = ‘givenUsername’
Query to insert new meeting into meetings table	"INSERT INTO \`meetings\`(\`Name\`, \`Date\`, \`Start Time\`, \`End Time\`, \`Length\`, \`Description\`, Creator\`) VALUES (a, b, c, d, e)
Query to get all relevant records in meetings table based on given username	SELECT * FROM meetings m JOIN user_meeting um ON m.MeetingID = um.MeetingID JOIN users u ON um.UserID = u.UserID WHERE UserName = 'givenUsername'
Query to get all users with a meeting for given meeting ID	SELECT * FROM users u JOIN user_meeting um ON u.UserID = um.UserID JOIN meetings m ON m.MeetingID = um.MeetingID WHERE m.MeetingID = 12
Query to get all meetings for a user	SELECT * FROM meetings m JOIN user_meeting um ON m.MeetingID = um.MeetingID JOIN users u ON um.UserID = u.UserID WHERE UserName = ‘givenUsername’ ORDER BY \`Date\` DESC

Java and JavaFX

The solution is separated into five modules each containing java classes and/or forms, these are: View, Edit, Create, Login and main. All forms are created using JavaFX with the drag and drop JavaFX form creator, ‘scene builder’ by gluon. JavaFX is a software platform that comes included with Java 8 and was created to make desktop applications. The scene builder is shown below (Scene Builder).



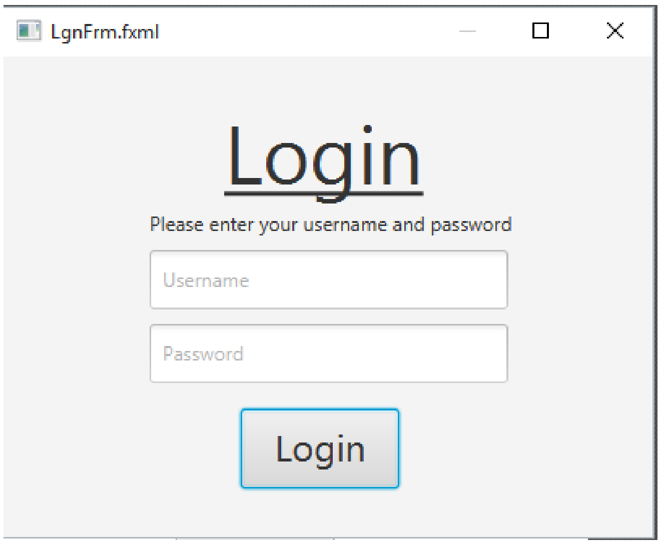
Main Module

Database Manager

This is a java class that contains methods to connect to the database and run queries and return results depending on the query.

Login

Login Form



Login Class

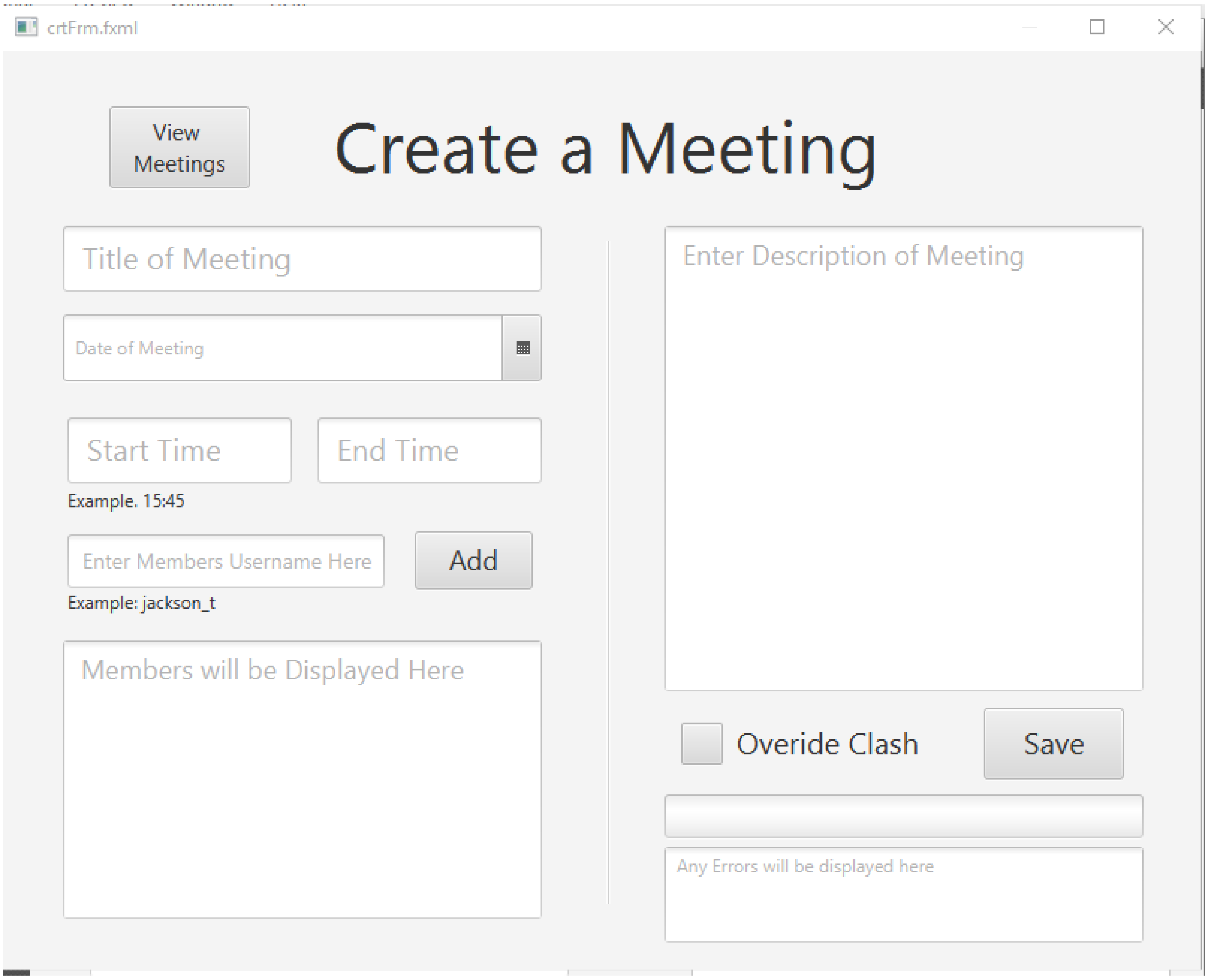
```
String Username = inputString("Please enter username");
String Password = inputString("Please enter password");
if (Username is in student login table) {
    if (password is equal to password in student login table){
        output("Welcome");
        open [[View Form]] ;
    }else {
        output("password is incorrect");
    }
}else {
    output("username is incorrect");
}
```

Create

This is the module which allows users to enter information about a database and check if there is a clash. The module also checks the syntax of inputted fields to ensure data is consistent.

CrtFrm

It will allow users to input the details of the meeting and contains prompts for the type and format of data to be inputted.



Meeting Clash Check Pseudocode

```
If (meeting is in database) {
    output "Meeting is already in database"}
}else {

For (all meetings in the database for that entered date) {

if (! ((dbStart.before(mtnStart) && dbEnd.before(mtnStart)) || (dbStart.after(mtnEnd)
&& dbEnd.after(mtnEnd)))){

for (i=0;i<membersArray.length;i++;){
If (membersArray[i] is in database meeting) {
Output ("Member" +memberArray[i]+ "has a meeting during that time ";)
}}}}}
```

Error Checks

A variety of checks were used to validate the data being entered by a user about the meeting into the form. Some checks were enforced using java try…catch methods. If any check failed an error was outputted and these checks are listed below.

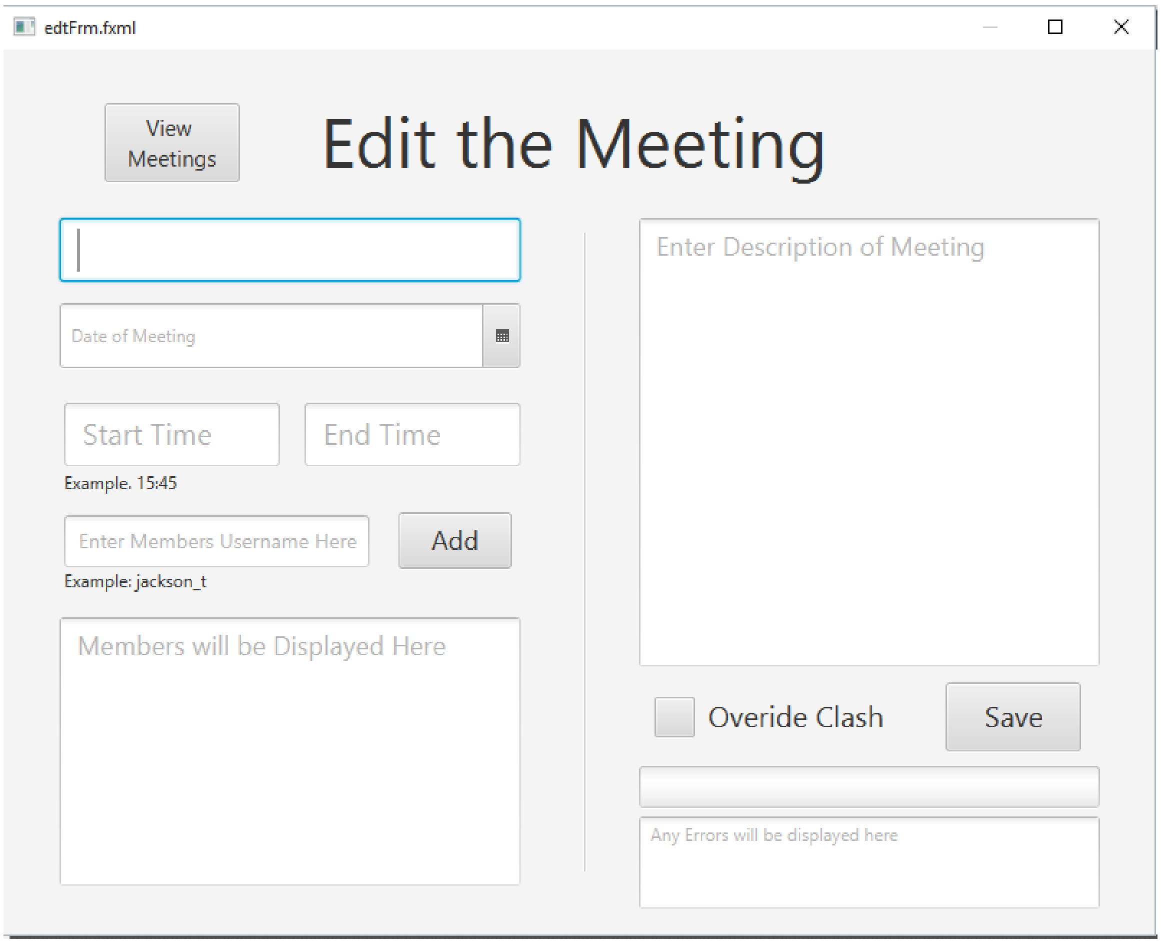
1. Data is entered into every field
2. Start and end times are in the format HH:mm
3. Start time before end time
4. Date has not passed
5. Member entered is in database
6. Member has not already been entered

Edit

This module queries data on a meeting based on a given meetingID and places it in a form for editing. The edited data is used to update the meeting record using the meetingID.

EdtFrm

It will allow users to edit the details of the meeting and contains performs the same checks as the create form.

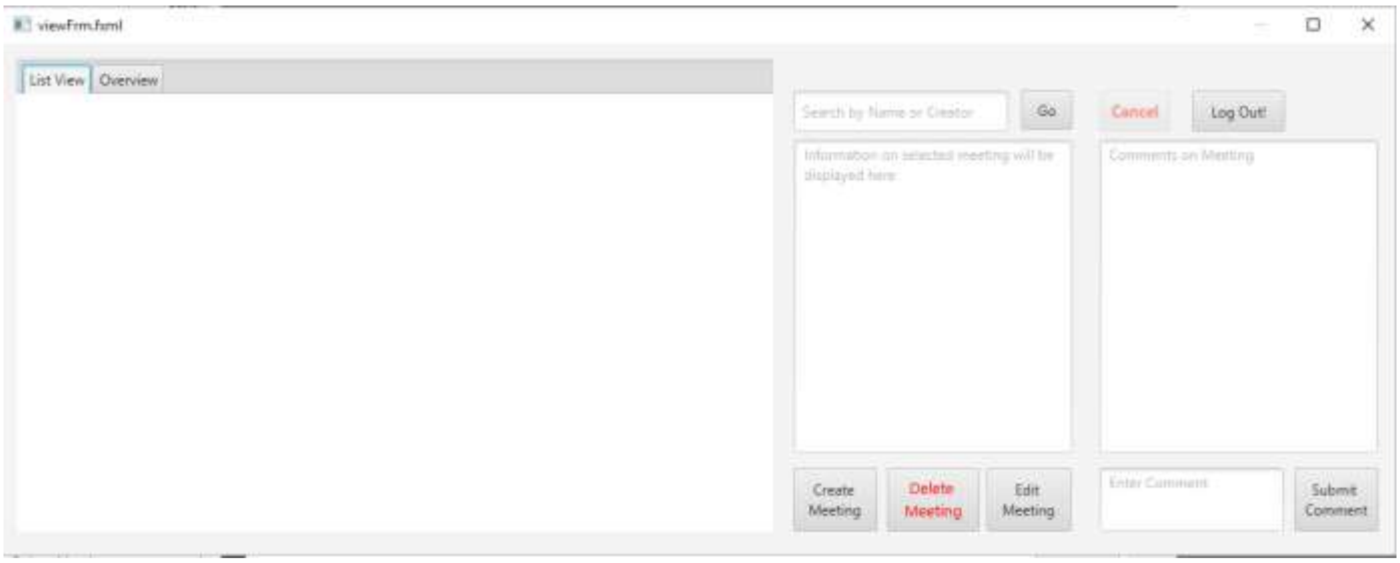


View

This module retrieves data on all relevant meetings based on userName and displays it on a form with two tabs.

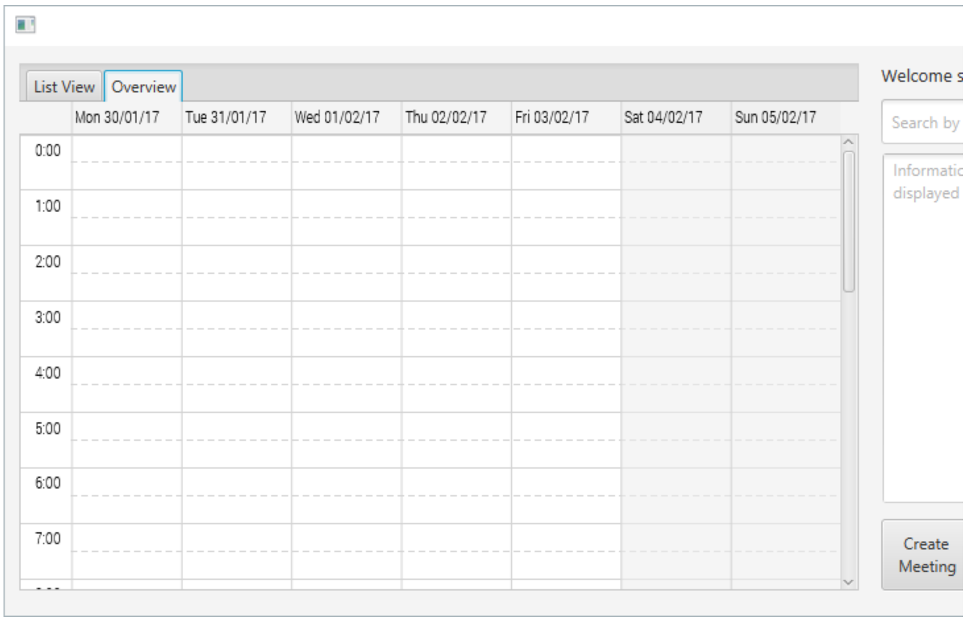
viewFrm

It presents the data on all meetings the user has for the day in one of two different ways the user can choose between using tabs. The user is also able to search for a specific meeting by meeting name.

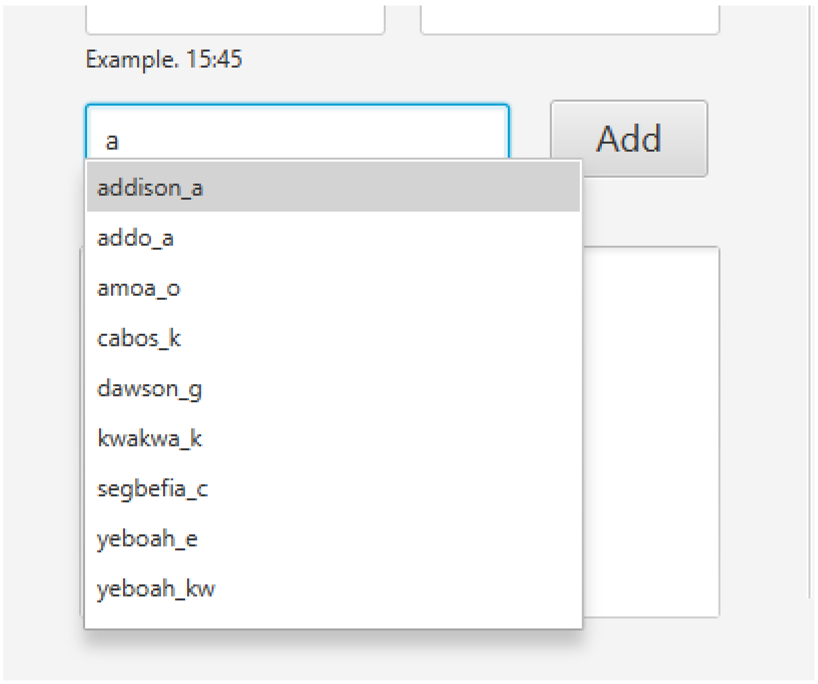


External Libraries Used

1. The month overview was made using the ‘Agenda’ Class from the JFXtras Library and is shown below (Jfxtras).



2. The ‘autocomplete’ class from the ControlsFX Library was used to allow the text field used to enter members to autocomplete using all members’ names from the database and is shown below (ControlsFX).



Word Count: 925

Bibliography

- XAMPP. Computer software. XAMPP. Vers. 5.5.30. Apache Friends, n.d. Web. <https://www.apachefriends.org/index.html>.
- Scene Builder. Computer software. Scene Builder - Gluon. Vers. 8.3.0. Cluon, 16 Dec. 2016. Web. 3 Feb. 2017. <http://gluonhq.com/products/scene-builder/>.
- Jfxtras-agenda. N.p.: Jfxtras, n.d. .java. http://jfxtras.org/
- ControlsFX. N.p.: Fxexperience, n.d. .java. http://fxexperience.com/controlsfx/