

# Report On Term Project

Name: Ahnaf Faisal

Section: A

Id: 1505005

- Advisor: Mohammad Saifur Rahman

Project Name:

## Meeting Scheduler

## How to Run my Project:

All my java source codes are under “meetingscheduler” package. In this package server main function is in the “MeetingChedulerServer.java” file and Client main function is in the “MeetingScheduler.java” file. So to build my project we will have to use this command, “javac meetingscheduler/\*.java”. After that the class files will be built, then we will have to run the server using “java meetingscheduler.MeetingChedulerServer” and run the Client using “java meetingscheduler.MeetingScheduler” command. The network IP address is set “localhost”. So to run the client on different machine, we have to change the address in the codes accordingly.

## Google Drive link:

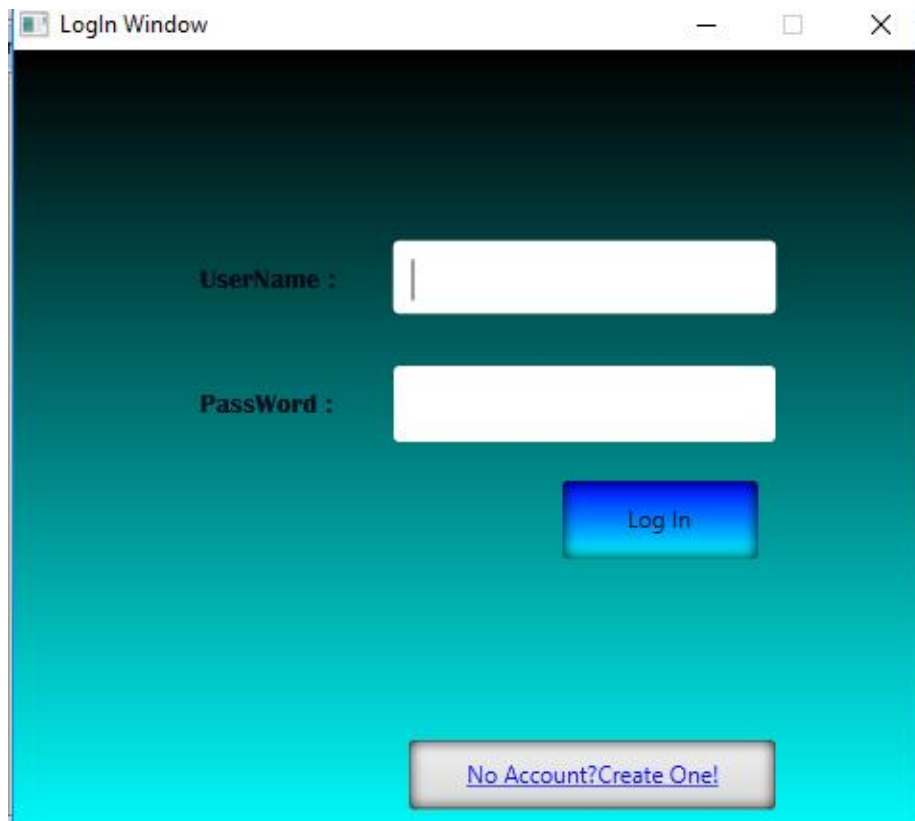
<https://drive.google.com/drive/folders/0Bz-wm-W5mJiBMlU1bEtKdHZkZlE?usp=sharing>

## Summary:

This “Meeting Scheduler” application provides a server-client based system. The information of the clients will be stored by the server. Any client can send another client signed with the application can send a request to hold a meeting. This request will be forwarded to another client whom the first client requested. If the client either accepts or declines the request, the first client will get the notification. If the client accepts, then a meeting will be scheduled. Both the clients are allowed to cancel the meeting. If anyone cancels the meeting, other one will get the notification. There is also an alarm system to remind the users about their meetings. Meetings which have ended will be stored in history.

## Snapshots and Descriptions:

When a user opens the application, there will be a log in window. If the user provides correct username and password, then the user will be logged in. Otherwise it will show “Invalid Login”.



A screenshot of a 'Login Window' with a dark teal background. It features two white input fields for 'UserName' and 'PassWord'. Below the password field is a blue 'Log In' button. At the bottom, there is a button with the text 'No Account? Create One!' in blue, underlined font.

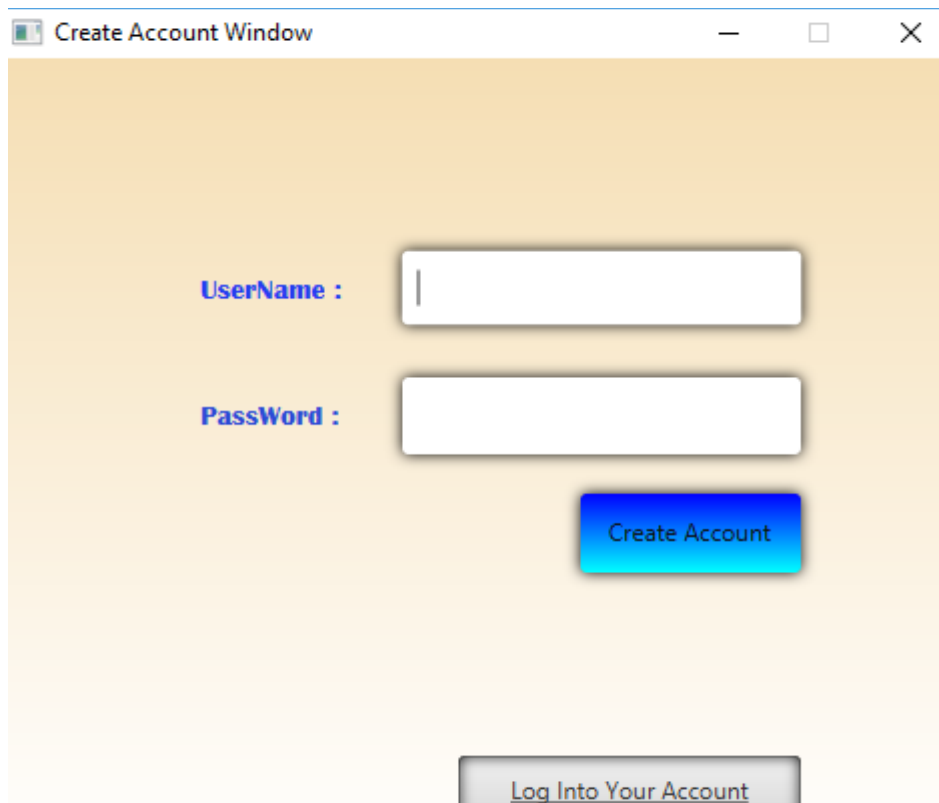
UserName :

PassWord :

Log In

[No Account? Create One!](#)

If the client does not have an account, he can go to “No Account? Create One!” and get signed with the application.



A screenshot of a 'Create Account Window' with a light beige background. It features two white input fields for 'UserName' and 'PassWord'. Below the password field is a blue 'Create Account' button. At the bottom, there is a button with the text 'Log Into Your Account' in blue, underlined font.

UserName :

PassWord :

Create Account

[Log Into Your Account](#)

If the login is successful, then the user will go to the main window where there will be calendar, a digital clock and other options. The calendar current date has a yellow border around itself and if there is a meeting in any date, that date is highlighted by yellow orange color. By clicking on previous and next the user can change the month the calendar is showing. By clicking on the log out button he will be logged out of the application and if he clicks on the close button it will automatically close the application. The main window's snapshot is given below.



If the user clicks on the “New Event” button then he will get the following window where he can send his request.

The image shows a web form for creating a new event. At the top, there are three buttons: "Send Request", "Clear", and "Back". Below these is a section titled "Details" in blue. The form includes several input fields: "Meeting Name", "Location", "Person Name", "Meeting Date" (with a calendar icon), "Start" (a dropdown menu), "End" (a dropdown menu), and "Meeting Description" (a large text area). There is also a checkbox labeled "All Day". The form has a light blue background with a pink footer.

If the user clicks on the “Users” of the main window, he will visit this window.

<i>Refresh</i>	<i>Back</i>
<i>UserName</i>	<i>Number of Meetings</i>
<i>Ahnaf</i>	<i>8</i>
<i>Saifur</i>	<i>4</i>
<i>Badhon</i>	<i>0</i>
<i>Pro</i>	<i>6</i>
<i>Hridoy</i>	<i>0</i>
<i>End of List</i>	

After the user sends any request he can check them in the sent window, see the details and even cancel them if he needs.

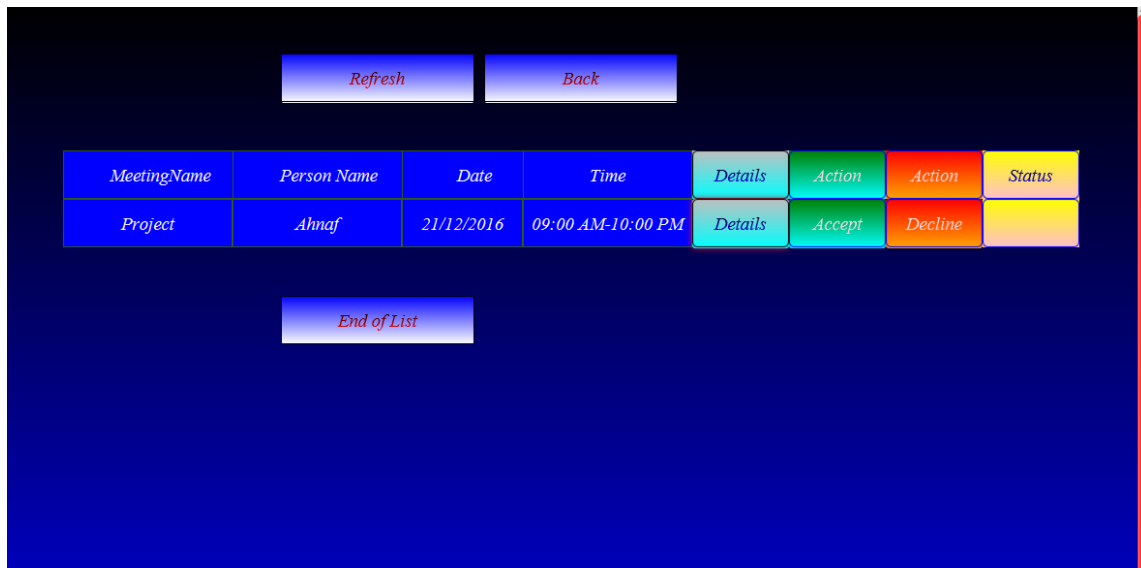
Refresh

Back

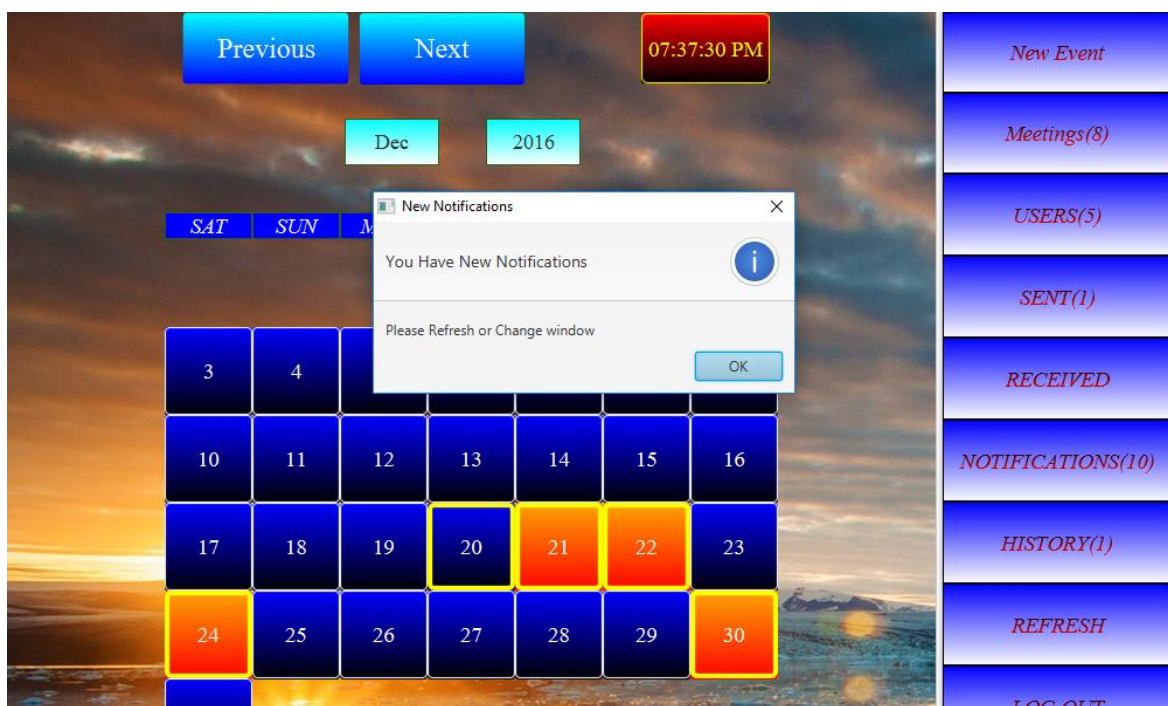
MeetingName	Requested	Date	Time	Details	Action
Project	Badhon	21/12/2016	09:00 AM-10:00 PM	Details	Cancel

End of List

When the user sends a request to other user, the other user can check them in the “Received” window and accept or decline as he wishes.



When a request is accepted or declined or sent the other user will receive a pop up message like this.



By going into the “Notifications” window he can check which request is accepted or which one is declined or which meeting is cancelled. The “Notifications” window is given below.



In the “Meetings” window he can check which meetings are going to held and check the details and even cancel them.



If the user clicks “Details” button , corresponding meeting’s details will be shown and the user can also adjust the reminder time.



Back

Event Reminder: 01.00 Hour

Meeting Name : dusksd

Person Name : Saifur

Location : sudfdf

Date : Dec 21, 2016

Start Time : 12:00 PM

End Time : 05:30 PM

Meeting Description : sdiolds sdiodfio sduiifddf

05.00 Min

10.00 Min

15.00 Min

20.00 Min

30.00 Min

45.00 Min

01.00 Hour

01.50 Hour

02.00 Hour

02.50 Hour

03.50 Hour

04.00 Hour

04.50 Hour

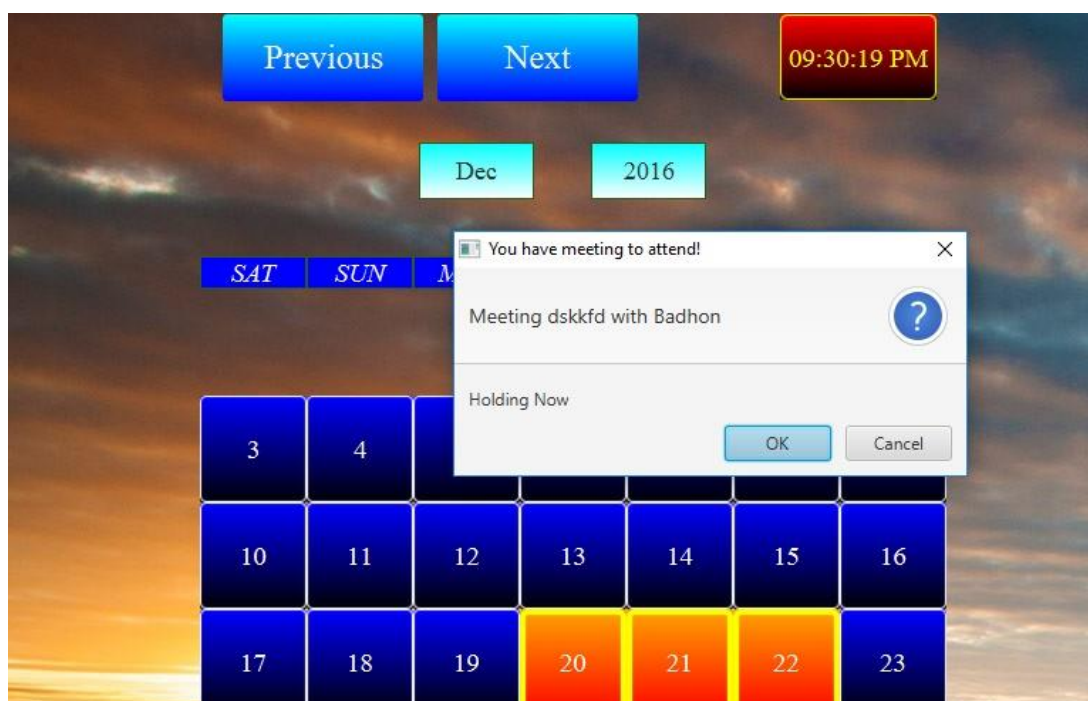
05.00 Hour

06.00 Hour

07.00 Hour

08.00 Hour

And finally there comes the alarm system.



### Tools I have used:

I have used Netbeans IDE and JDK 1.8 for my project development. I also used FXML files and scene builder for some windows. But in windows which I needed to work dynamically, I did not use scene builder. Rather hard coded it. I also used CSS stylesheet for a colorful interface.

### Lacking:

Although I tried heart and soul to build the project, there is a lacking. There is no option to hold a meeting with multiple persons at the same time. To do so in my application, the user has to send his desired persons requests individually. So for being used as professional software, this feature has to be added. But I have worked hard for this project and I am satisfied with what I have done. But there is no limit to improvements. So I will have to keep working harder. But at the end of the day I have learnt many things about the Java FX, Java Calendar, Networking, Threading and time management system. And I also hope this learning will be beneficial in my later life.