PROJECT DOCUMENTS

Meeting scheduling

Android: https://gitlab.cecs.anu.edu.au/u6889341/comp2100_6442-group-project.git
Server: https://gitlab.cecs.anu.edu.au/u6889341/6442-project-server-code.git

MAY 29, 2020

COMP 2100_6442

Git Commit History Link

https://gitlab.cecs.anu.edu.au/u6889341/comp2100_6442-group-project/commits/master

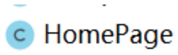
Wiki Link

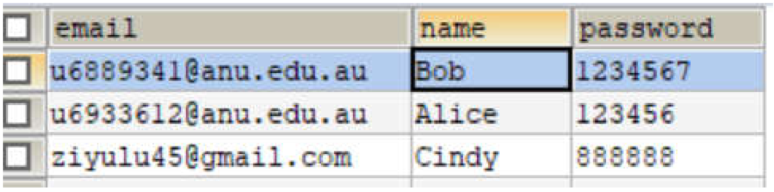
https://gitlab.cecs.anu.edu.au/u6889341/comp2100_6442-group-project/wikis/Meeting-Scheduling

Team Structure and Roles

1. Team members:
    a) LU Ziyu (u6889341)
    b) Erin Xiong (u6933612)
2. Roles:
    a) LU Ziyu:
        i. Build a Data Tier by using mySQL database
        ii. Build an Application Tier by using tomcat server and Java EE
        iii. Build some Presentation Tier logic between different UI pages and the connection to the network
    b) Erin Xiong
        i. Build the Presentation Tier with all the user interfaces (UI).
        ii. Note down the meeting content and write meeting minutes.
        iii. Responsible for client testing to find potential bugs.

App Summary

1. Introduction
   - App question chosen: Meeting Scheduling
   - App name: Meeting Scheduling
   - Main activity is  HomePage
2. Note before use
   a) Please make sure your phone has connected to the network
   b) Please use virtual keyboard on phone to type **do not use computer key board.**
   c) Before the quick guide, there are two users in the database (Alice and Bob). Cindy is the user used to finish the quick guide. We encourage you to create your own user to try it and it is interesting.



□	email	name	password
□	u6889341@anu.edu.au	Bob	1234567
□	u6933612@anu.edu.au	Alice	123456
□	ziyulu45@gmail.com	Cindy	888888

3. Features:
   a) Default features
      i. Adding a meeting event
         1. A meeting event can be added (See 4. quick guide → d))
      ii. Timeslot preference submission from a (registered/unregistered) user
         1. Time slot can be set (See 4. quick guide → f))
      iii. Storing the scheduling results on a proper server

1. All information and logic calculations are on the server. We bought a cloud hosting from Tencent Cloud company. Tomcat is the server and mySQL is the data database.
2. When a user creates a group or a meeting. All users in this group will receive this meeting information.

iv. Visualizing scheduling results

1. Scheduling results can be viewed (See 4. quick guide → g))

v. Setting deadline for scheduling

1. Deadline can be set when the user create the meetings, this will influence the modification of the meeting scheduling (See 4. quick guide → d),e))

b) Advanced features

i. Scheduling multiple meetings at once

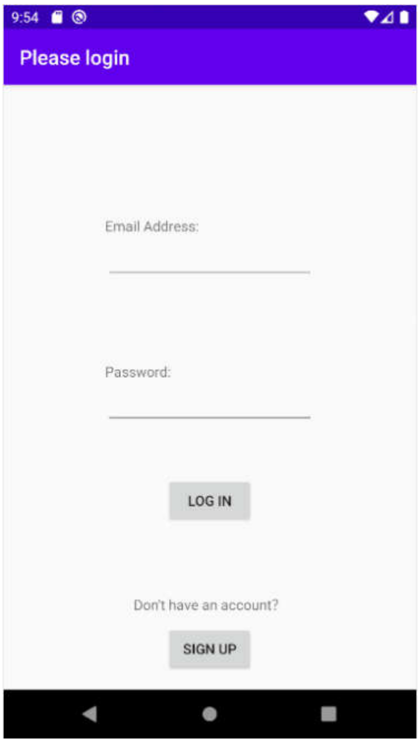
1. You can click one key to schedule all the meetings (See 4. quick guide → g))

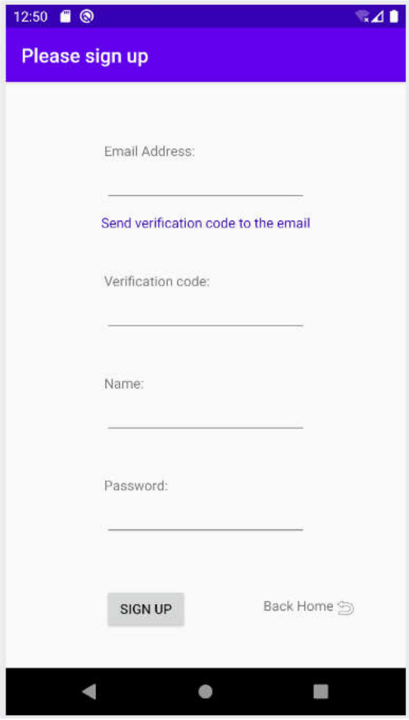
c) Other advanced features

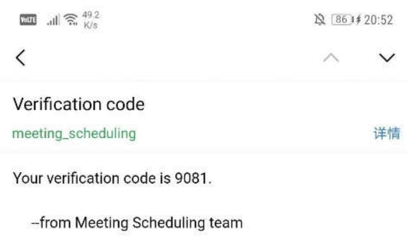
i. People can receive an email of verification code when signing up. (See 4. quick guide → a))

4. Quick guide

a) Login page and sign up page, if you are the first time to login, please sign up. You will receive an email with the random verification code to help you to sign up.







Your verification code is 9081.

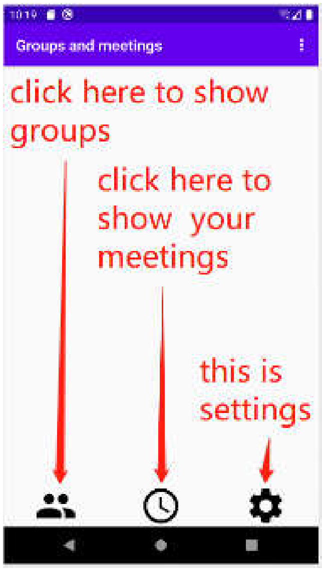
--from Meeting Scheduling team

b) Main page

i. The button on the bottom are showing my groups, showing my meetings and settings

ii. One key schedule is after you have several meetings and have set your time slot, the meetings can be classified as available and un available

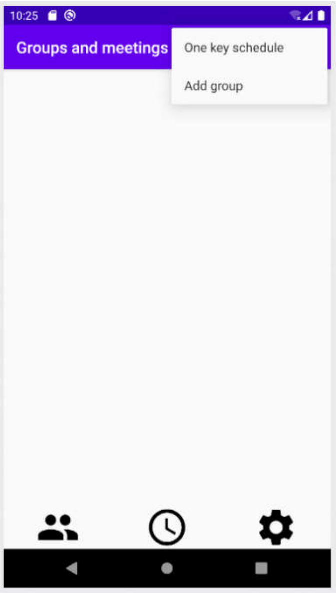
iii. Because you do not have groups and meetings now, you can create a group firstly.



click here to show groups

click here to show your meetings

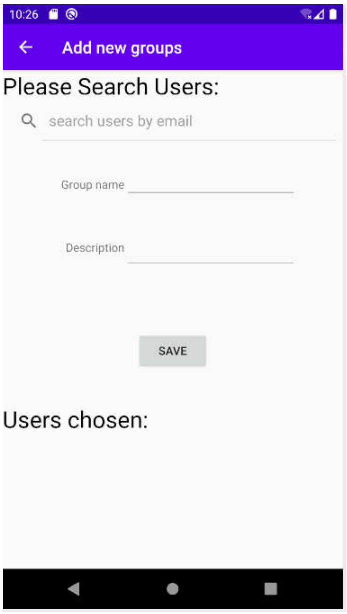
this is settings

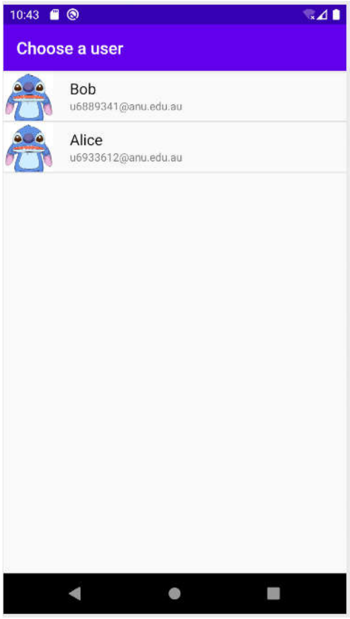


c) Create groups (All users in this group will find this group in their Meeting Scheduling accounts)

i. You need to search users by characters within the email to add group members, you are added into this group by default. You can choose group name and descriptions as you like

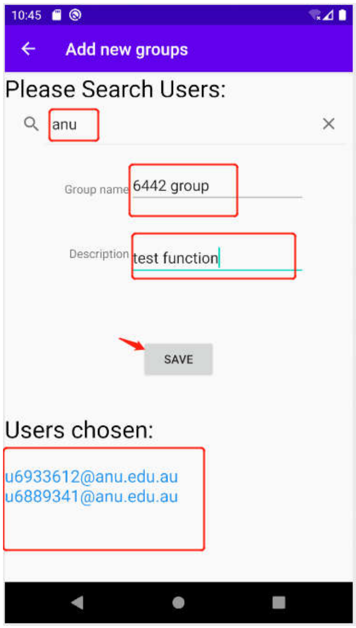
ii. G27 is the group id generated by system automatically. That can help us extent functions such as searching groups.





Bob
u6889341@anu.edu.au

Alice
u6933612@anu.edu.au

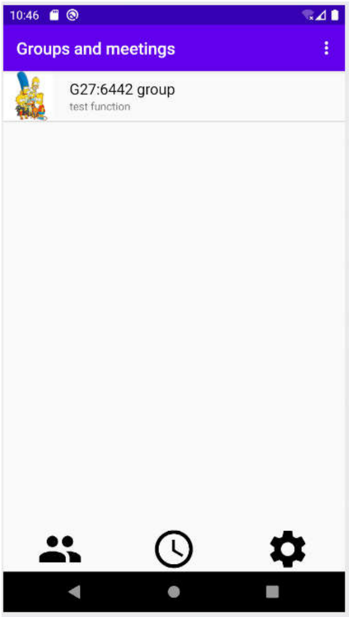


anu

Group name 6442 group

Description test function

u6933612@anu.edu.au
u6889341@anu.edu.au



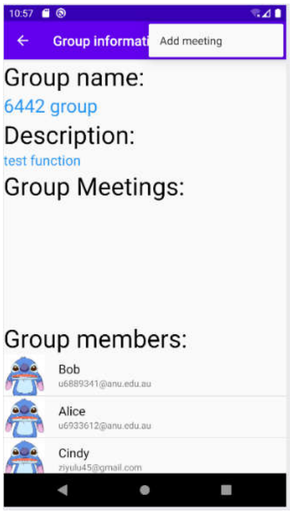
G27:6442 group
test function

d) Choose a group, see the information in detail and add a meeting (All users in this group will reveice a new meeting information.)

i. You can set all the meeting informations.

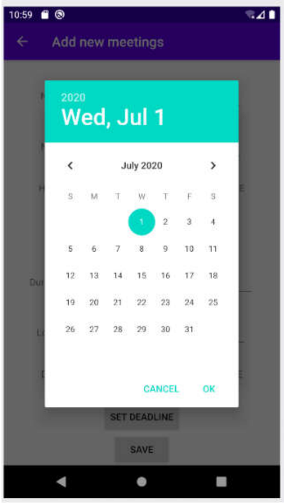
ii. You can choose a deadline. Normally, the meeting information can be modified. If current time is later than the deadline, you will not change the meeting information.

iii. We created 3 meetings to show functions


Group informati | Add meeting

Group name:
6442 group
Description:
test function
Group Meetings:

Group members:
Bob u6889341@anu.edu.au
Alice u6933612@anu.edu.au
Cindy ziyulu45@gmail.com


Add new meetings

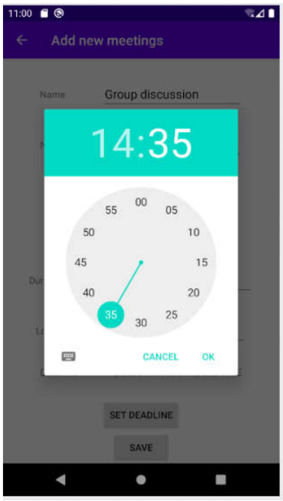
2020
Wed, Jul 1

July 2020

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

CANCEL OK

SET DEADLINE
SAVE

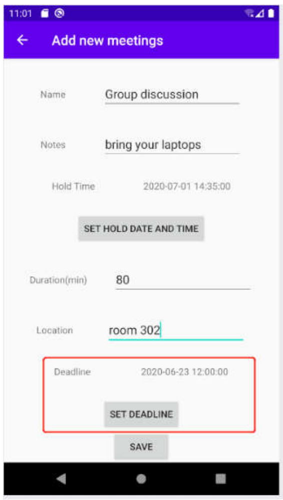

Add new meetings

Name Group discussion

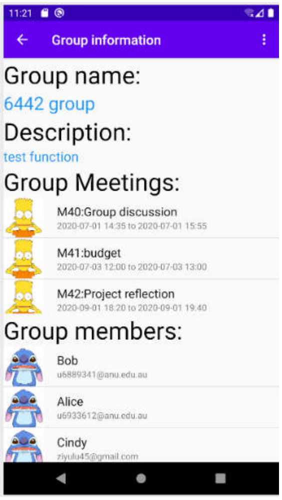
14:35

CANCEL OK

SET DEADLINE
SAVE


Add new meetings

Name Group discussion
Notes bring your laptops
Hold Time 2020-07-01 14:35:00
SET HOLD DATE AND TIME
Duration(min) 80
Location room 302
Deadline 2020-06-23 12:00:00
SET DEADLINE
SAVE


Group information

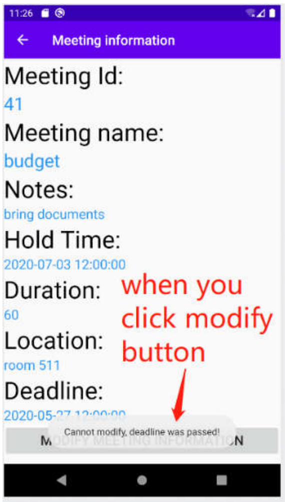
Group name:
6442 group
Description:
test function
Group Meetings:
M40:Group discussion 2020-07-01 14:35 to 2020-07-01 15:55
M41:budget 2020-07-03 12:00 to 2020-07-03 13:00
M42:Project reflection 2020-09-01 18:20 to 2020-09-01 19:40
Group members:
Bob u6889341@anu.edu.au
Alice u6933612@anu.edu.au
Cindy ziyulu45@gmail.com

e) Find detailed meeting information, and modify

i. When you click the clock icon, the meeting will be shown in the home page

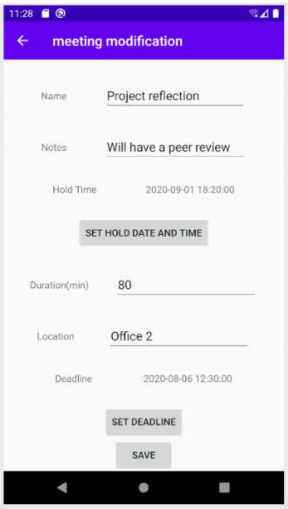
ii. You can choose one meeting to see meeting info in detail, if the deadline has not been passed, you can modify it. (Deadline of M41 has been passed, thus it cannot be modified.)

iii. The second picture is to modify M42: change meeting location as "office 10"

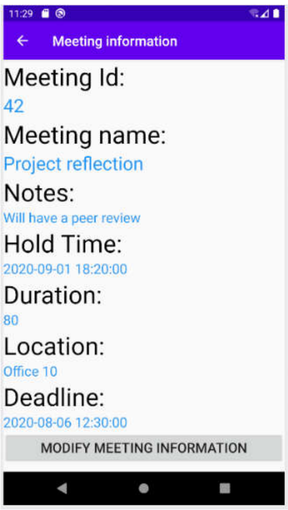

Meeting information

Meeting Id:
41
Meeting name:
budget
Notes:
bring documents
Hold Time:
2020-07-03 12:00:00
Duration:
60
Location:
room 511
Deadline:
2020-05-27 12:00:00
Cannot modify, deadline was passed!
MODIFY MEETING INFORMATION

when you click modify button


meeting modification

Name Project reflection
Notes Will have a peer review
Hold Time 2020-09-01 18:20:00
SET HOLD DATE AND TIME
Duration(min) 80
Location Office 2
Deadline 2020-08-06 12:30:00
SET DEADLINE
SAVE

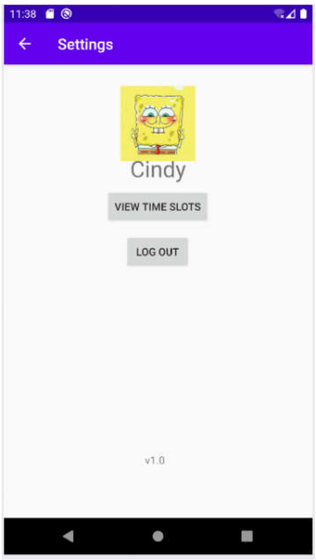

Meeting information

Meeting Id:
42
Meeting name:
Project reflection
Notes:
Will have a peer review
Hold Time:
2020-09-01 18:20:00
Duration:
80
Location:
Office 10
Deadline:
2020-08-06 12:30:00
MODIFY MEETING INFORMATION

f) Settings, view time slots, set time slots

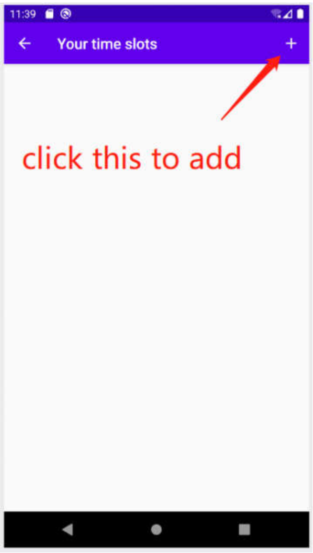
i. The time slots are the time period when you are free to attend meetings. After setting this, you can use one key schedule (will talk about later) to see which meeting you can attend and which meeting you cannot attend. If the end time is earlier than the start time, it cannot be saved.

ii. Let's added two time slots

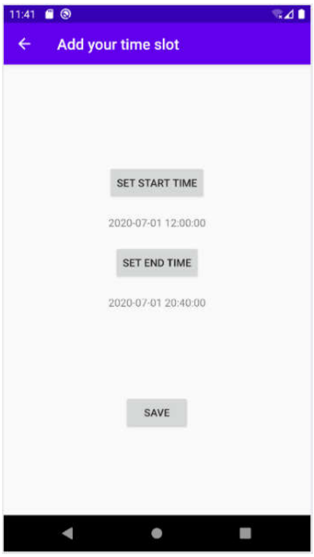

Settings

Cindy
VIEW TIME SLOTS
LOG OUT

v1.0

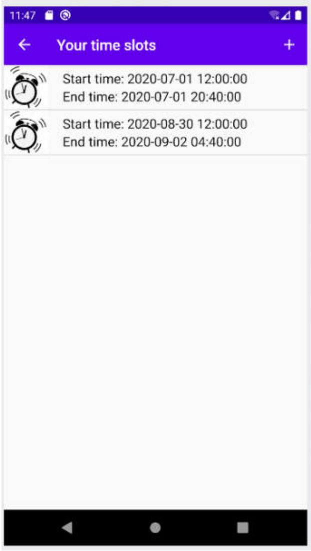

Your time slots +

click this to add


Add your time slot

SET START TIME
2020-07-01 12:00:00
SET END TIME
2020-07-01 20:40:00

SAVE

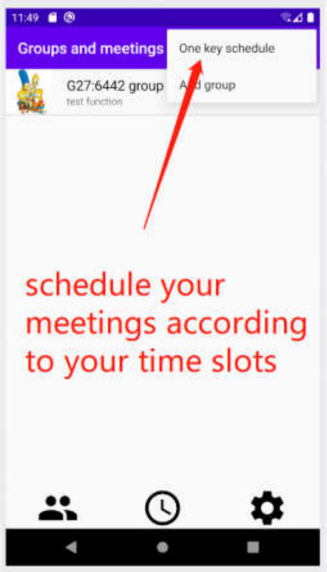

Your time slots +

Start time: 2020-07-01 12:00:00
End time: 2020-07-01 20:40:00

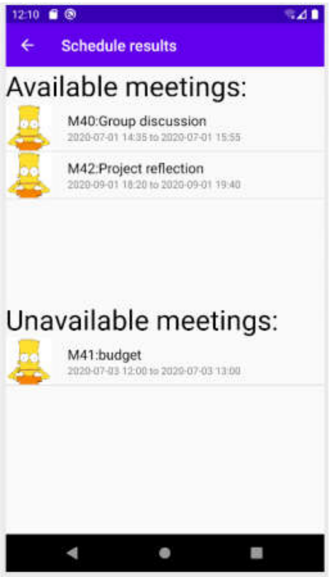
Start time: 2020-08-30 12:00:00
End time: 2020-09-02 04:40:00

g) One key schedule

i. You meetings will be classified by your time slots when you click “one key schedule”. This can help you to make a better plan, and you will not to spend time in ckecking your timetable frequently anymore!



schedule your meetings according to your time slots



Available meetings:

M40:Group discussion
2020-07-01 14:35 to 2020-07-01 15:55

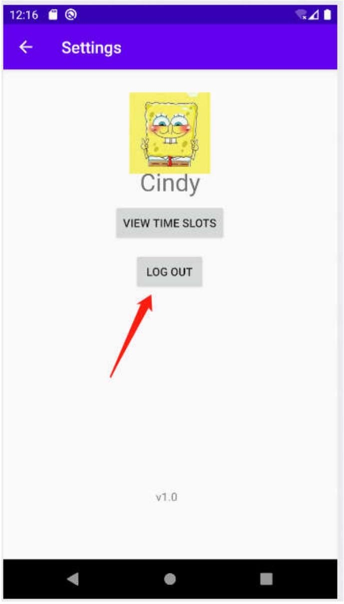
M42:Project reflection
2020-09-01 18:20 to 2020-09-01 19:40

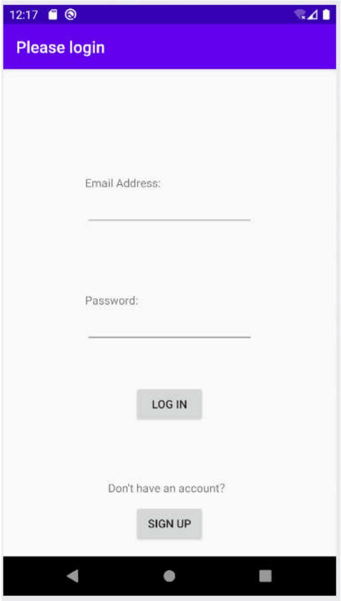
Unavailable meetings:

M41:budget
2020-07-03 12:00 to 2020-07-03 13:00

h) Log out

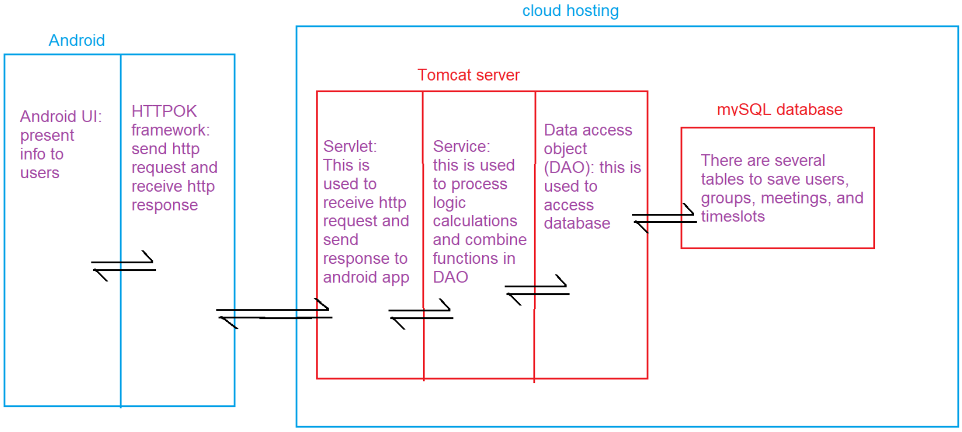
i. From the settings, you can log out, and you can log in again with any account.





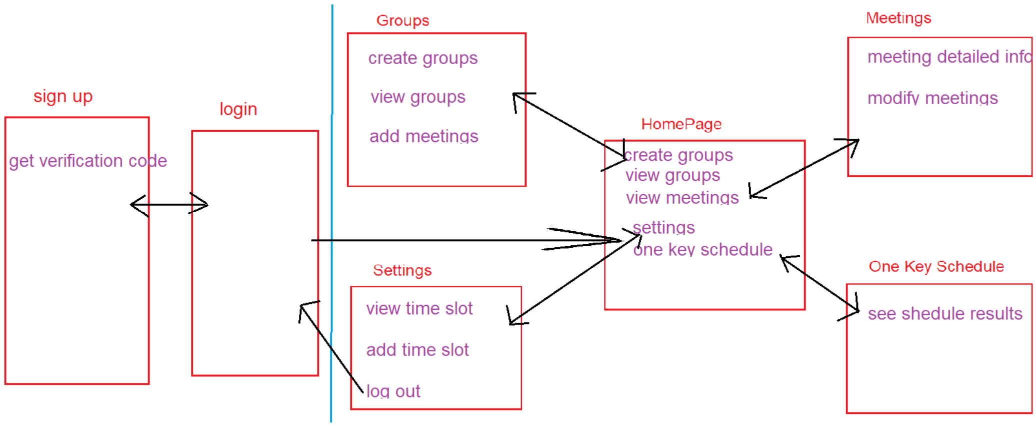
Design Summary Page

1. Architecture of app:
   a) There are 3 part with the app.
      i. Database is used to save data.
      ii. Tomcat server is used to manipulate database and do some logic calculation and finally send data to android app.
      iii. android only pay attention to present information to user and connect to network.
   b) Project is well modularized. Easy to extend. You can achieve any functions by change the particular part of code.

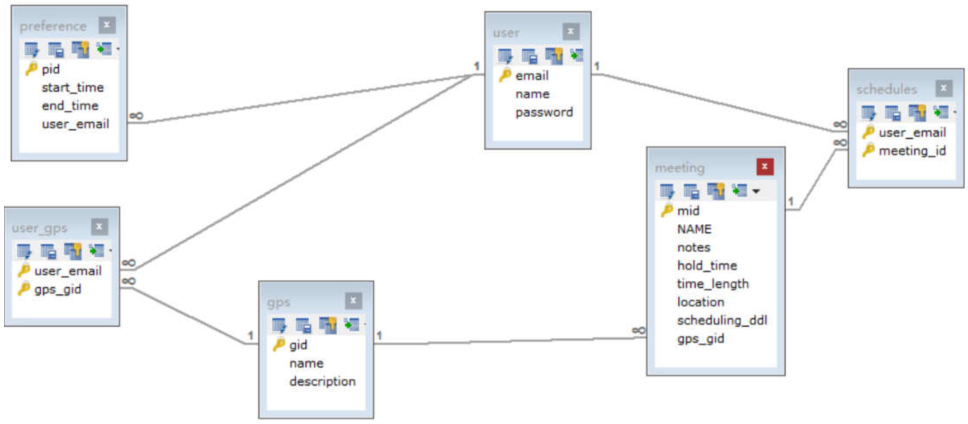


cloud hosting
Android
Android UI: present info to users
HTTPOK framework: send http request and receive http response
Tomcat server
Servlet: This is used to receive http request and send response to android app
Service: this is used to process logic calculations and combine functions in DAO
Data access object (DAO): this is used to access database
mySQL database
There are several tables to save users, groups, meetings, and timeslots

2. Android
   a) The following are some parts. Each part includes one or more activities. The arrows show the path among them.

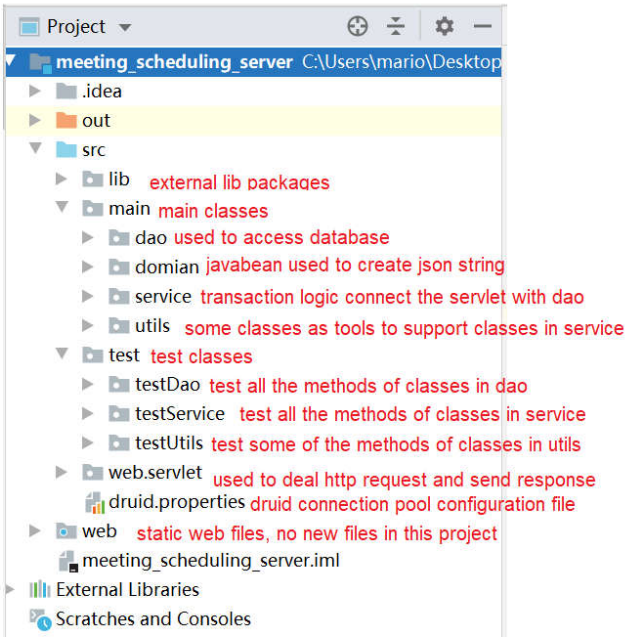


3. mySQL database
   a) There are 6 tables where there are 4 main tables and 2 intermediate table to make it easier to handle.



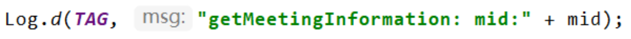
Testing Summary

1. Introduction
    a) Repositories
        i. Android: https://gitlab.cecs.anu.edu.au/u6889341/comp2100_6442-group-project.git
        ii. Server (run on tomcat, modified by Java language): https://gitlab.cecs.anu.edu.au/u6889341/6442-project-server-code.git
    b) According to our architecture, the code with different functions is applied different ways of test. Generally speaking, all the codes are used and tested in some ways.
2. Code on server
    a) Notes:
        i. Packages in lib should be imported
        ii. **Please do not run test methods in testDao and testService, because those test methods will change the current database contents, and may influence user's information in app**
        iii. You can run test methods in testUtils those will not influence database information.
    b) Methods:
        i. To test dao and service, we use junit4 and print out the result, and check our database (we use SQLyog to help us show information in database)
        ii. Classes in domain are just java bean classes, which are generated by keyboard shortcuts. They do not need to be tested.



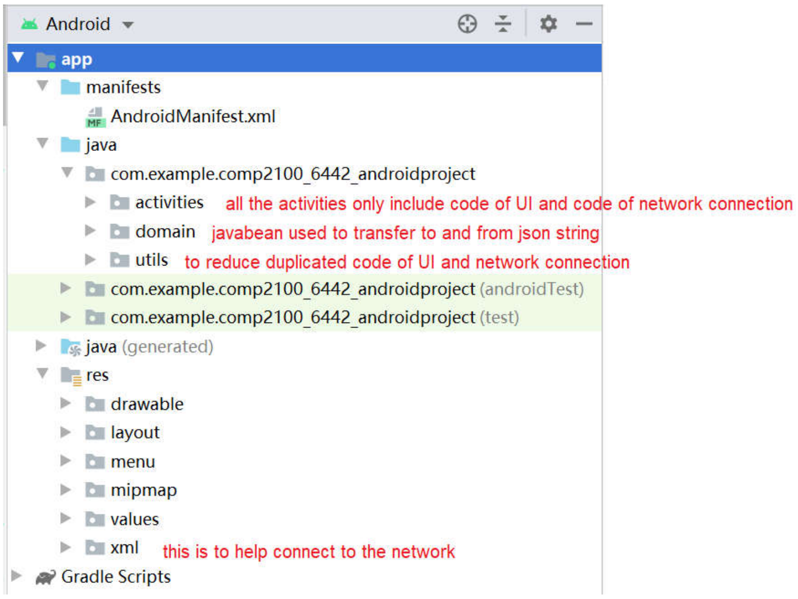
```
Project
▼ meeting_scheduling_server  C:\Users\mario\Desktop
  ▶ .idea
  ▶ out
  ▼ src
    ▶ lib  external lib packages
    ▼ main  main classes
      ▶ dao  used to access database
      ▶ domian  javabean used to create json string
      ▶ service  transaction logic connect the servlet with dao
      ▶ utils  some classes as tools to support classes in service
    ▼ test  test classes
      ▶ testDao  test all the methods of classes in dao
      ▶ testService  test all the methods of classes in service
      ▶ testUtils  test some of the methods of classes in utils
    ▶ web.servlet  used to deal http request and send response
      druid.properties  druid connection pool configuration file
  ▶ web  static web files, no new files in this project
    meeting_scheduling_server.iml
▶ External Libraries
  Scratches and Consoles
```

        iii. Utils are also test by junit 4, but with assert () syntax. Because these are simple java methods.
        iv. Classes in servlet are tested on browser. We input http address, and see the response on web page.
3. Code on android
    a) Notes:
        i. The code are only about UI and network connection, there no logic calculations, we cannot apply junit4 to test the code.
    b) Methods:
        i. Activities are tested by adding



```
Log.d(TAG, msg: "getMeetingInformation: mid:" + mid);
```

some Log.d(TAG, msg) in the codes, and we inspect the logcat to test those codes. As well as that, we run the project to see the reaction on UI



```
Android
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.example.comp2100_6442_androidproject
      ▶ activities  all the activities only include code of UI and code of network connection
      ▶ domain  javabean used to transfer to and from json string
      ▶ utils  to reduce duplicated code of UI and network connection
    ▶ com.example.comp2100_6442_androidproject (androidTest)
    ▶ com.example.comp2100_6442_androidproject (test)
  ▶ java (generated)
  ▼ res
    ▶ drawable
    ▶ layout
    ▶ menu
    ▶ mipmap
    ▶ values
    ▶ xml  this is to help connect to the network
▶ Gradle Scripts
```

        ii. Classes in domain are just java bean classes, which are generated by keyboard shortcuts. They do not need to be tested.
        iii. Utils are just to help reduce duplicated code of UI and network connection and are tested together with activities.

Meeting Minutes

Date 14. April.2020

1. Question: Which question should we choose?
   a) Discussion:
      i. Voting
         1. Dynamic tables should be produced
         2. UI is simple
         3. Identification should be produced
         4. Data storage
         5. Statistical graph
         6. Time count down
      ii. Meeting Scheduling
         1. Simple Ui
         2. Video, audio could be added.
         3. Timer
         4. Advertisement could be added
         5. Identification
      iii. Geo-tagging
         1. Too hard because maps are used.
         2. More image assets may be produced.
      iv. Treasure Hunting Game
         1. Too many images should be produced which is not about programming
   b) Decision:
      i. Choose the Meeting scheduling because there are less UI. We can focus on coding.
2. Question: what is the requirement of this project
   a) Discussion:
      i. Files should be produced
      ii. Codes should be managed by git
3. Question: what is the architecture?
   a) Monet has some experience on web developing. Therefore, we can also use the architecture of web developing, but the front-end is android app now.
4. Question: what is the delegation?
   a) Monet pays more attention on the back-end code
   b) Shawn focusses on front-end code
   c) Finally, Monet combine them together

Date 20. April.2020

5. Question: What is the design framework?
    a) The architecture of the app consists of 3 part: database, Tomcat server, android.
    b) Database is used to save data
    c) Tomcat server is used to manipulate and send data to the app
    d) Android achieve the user interface
6. Question: How can we make the database work properly?
    a) Erin Randomly create a few user accounts.
    b) Monet Use the exist user accounts to test if database successfully save the user information.
7. Question: How should the sign in/up page looks like?
    a) Create two new activity of sign in and sign up page with default layout in android.

Date 5. May.2020

1. Question: How will the home page looks like?
    a) The home page will contain 3 main function plate.
        i. Groups plate
        ii. Schedule plate.
        iii. Settings plate
2. Question: Which activity will be the main activity?
    a) We set HomePage.activity as the main activity.
3. Question: What method will be used to display the home page?
    a) Initially Erin created three fragments in HomePage, but then realize this method could be hard to connect with other functions, then Monet chose to use OnclickLisener to replace it.

Date 17. May.2020

1. Question: What functions the group and meeting page should achieve?
    a) Present the group information list and meeting information list
    b) Show details of each group and meeting
    c) Add a new group and add a new meeting
    d) A function of one-key schedule
    e) Modify meeting information
2. Question: What functions the setting page should achieve?
    a) Display the user name
    b) Show time slots of user
    c) Add new time slots
    d) Log out of the current account
3. Question: What functions do we need in the action bar

a) The button that send user back to the previous page.

b) The button of adding

c) The button of navigation menu

d) The label name of each page

Date 28. May.2020

1. Question: How can we find and solve the bugs?
   a) Erin is responsible to act as a user and test the programs with different functions and operation to find the edge cases.
   b) Monet is responsible for check the logcat and solve all the bugs.
2. Question: How should we organize the admin files?
   a) We divide the file into two parts and finish it separately.
   b) Us pdf format to display the hole admin files
   c) Try to show our great team work and communication
3. Question: How could we improve the code quality?
   a) Add the code comments to explain the key method and steps.
   b) Check if any code could be simplified or achieved in more efficient way

Statement of Originality

Declaration

I declare that everything I have submitted for this project it is entirely my own work, with the. following exceptions:

Code:

- Comment: The method of “setStartDate” and “setEndDate”
  Source: Textbook
  Reference: Android development details, Lei LI & Guohui Wang, 2018.7.

Image:

- Comment: he image used to show each group portrait.
  Source: http://www.duoziwang.com/head/meiwei/901080.html
- Comment**:** the image used to show each meeting portrait.
  Source**:** http://www.qqzhi.com/touxiang/499287/
- Comment**:** the image used to show each user portrait.
  Source**:** https://www.17qq.com/qqtouxiang/2277220.html

Eternal Libraries:

- From Gradle：
  implementation 'com.squareup.okhttp3:okhttp:4.5.0'
  implementation 'com.google.code.gson:gson:2.8.6'
- Server jar Package：
  druid-1.1.2.jar
  activation.jar
  mail.jar
  jackson-annotations-2.2.3.jar
  jackson-core-2.2.3.jar
  jackson-dababind-2.2.3.jar
  mysql-connector-java-8.0.20.jar
  protobuf-java-3.6.1.jar
  commons-logging-1.2.jar
  spring-beans-5.0.0.RELEASE.jar
  spring-core-5.0.0.RELEASE.jar
  spring-jdbc-5.0.0.RELEASE.jar
  spring-tx-5.0.0.RELEASE.jar

Name: Erin Xiong
Uid: u6933612
Name: LU Ziyu
Uid: u6889341