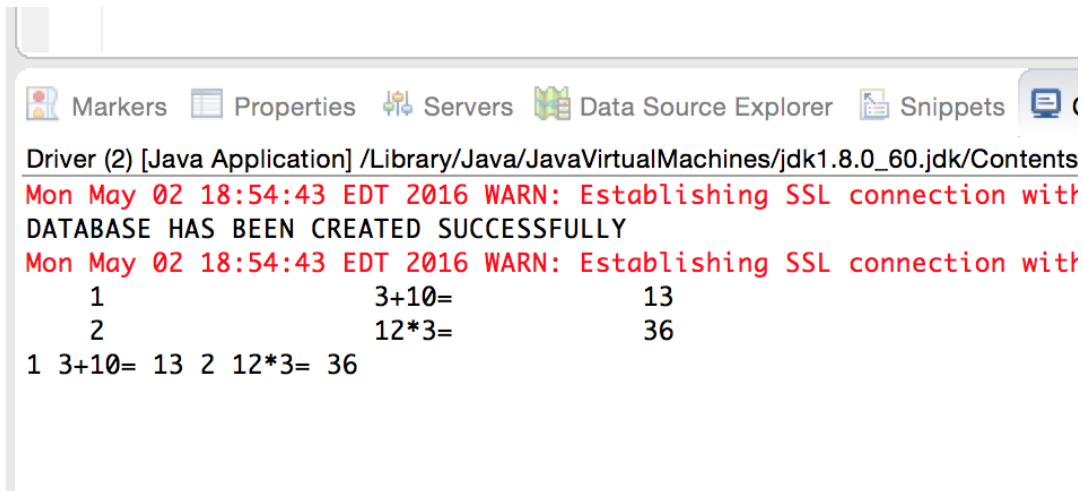


Test Document

In this document, we will talk about the test case which includes all the possible situations may happen for our application followed by our output result.

At the beginning of our project, we have to open the MySQL server. Then run our driver class which will automatically connect to the Mysql and write in the data we set in advance. Our driver class also includes read data from the database and then use the OutputStream in SocketServer class to interact with our application. We output the data in MySQL to show our database do work following is the screenshot.



The screenshot shows an IDE console window with the following content:

```
Driver (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents
Mon May 02 18:54:43 EDT 2016 WARN: Establishing SSL connection with
DATABASE HAS BEEN CREATED SUCCESSFULLY
Mon May 02 18:54:43 EDT 2016 WARN: Establishing SSL connection with
1      3+10=      13
2      12*3=      36
1 3+10= 13 2 12*3= 36
```

Figure 1 – Display what was added to the remote MySQL database.

Then we flush the math question and its answer. After open the server side, we open our application and set the same port with the server side. Our project start working from the MainActivity and it will load anything of the previously added alarms from the SQLite database. If no alarms were previously set, it will leave it blank.

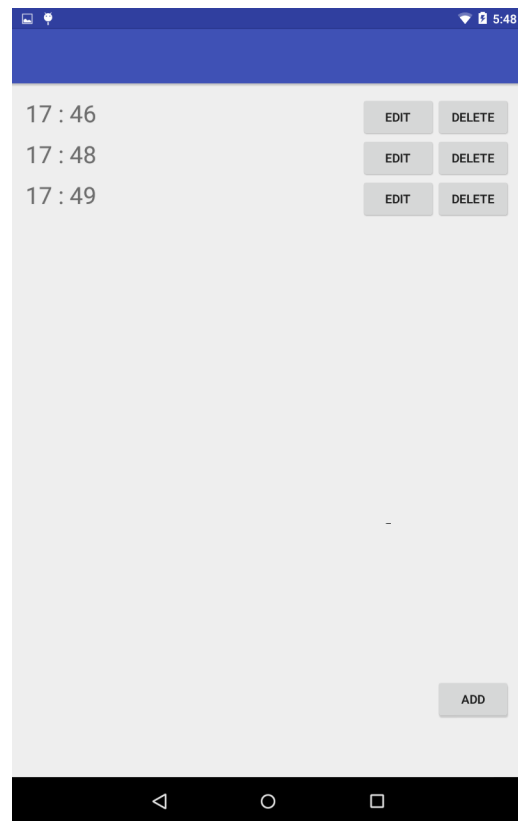
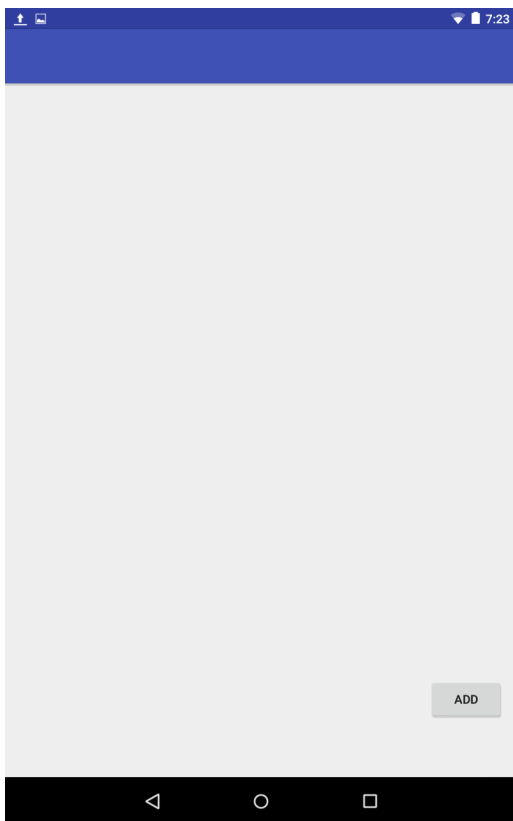


Figure 2 – If no alarm was previously set. Figure 3 – Main page loading previously set alarm.

Then user press the add button to add a new alarm, our application will jump to the setting page and dynamically load all locations.

User can set time and choose tune for the alarm. User can also set the tag for alarm.

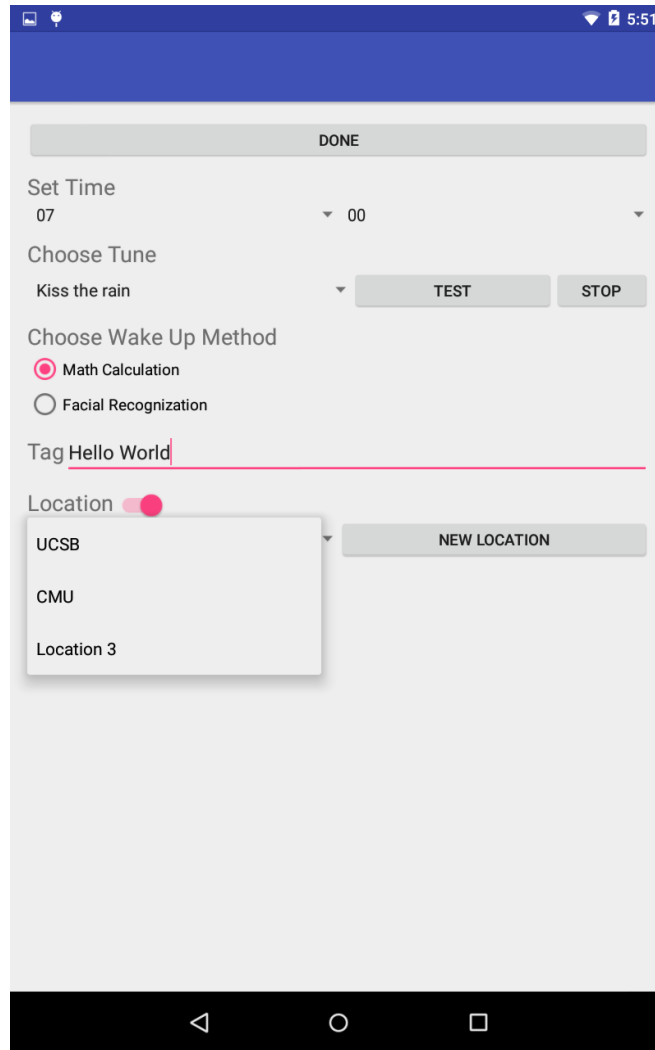


Figure 4 – An example Setting page.

User can also choose location which is one of our main features. When user choose the location feature and press the new location button, it jumps to the location page as following screenshot.

In here, the user can input the name of the location and press the “SEARCH” button to find the location on the Google Maps. If nothing is entered in the Location field, nothing will happen.

To save the location, the user can press the “SAVE” button. If nothing is entered in the Tag field, it will automatically generate one “Location #” for that particular location. You’ll be able to see the result in the Setting page.

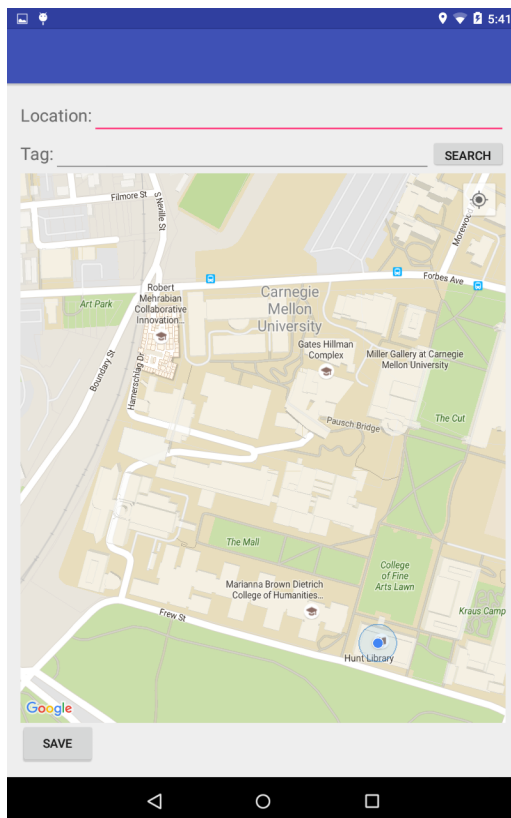


Figure 5 – Search with nothing entered.

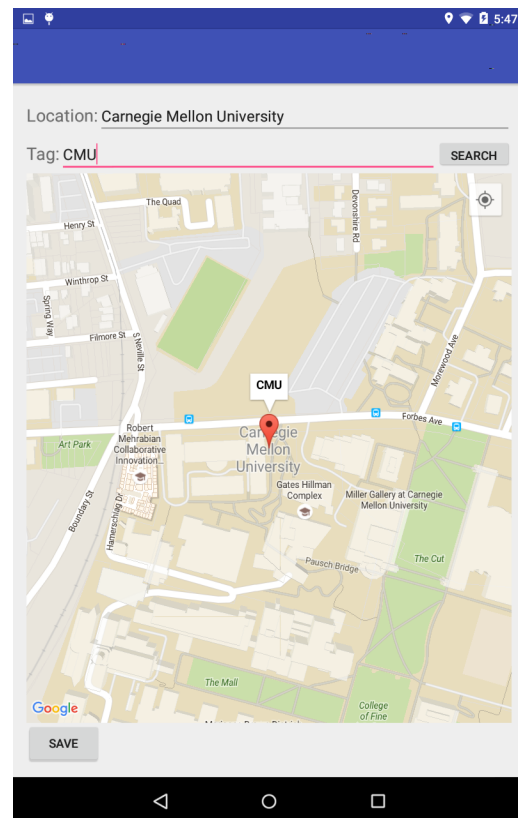


Figure 6 – Search with location.

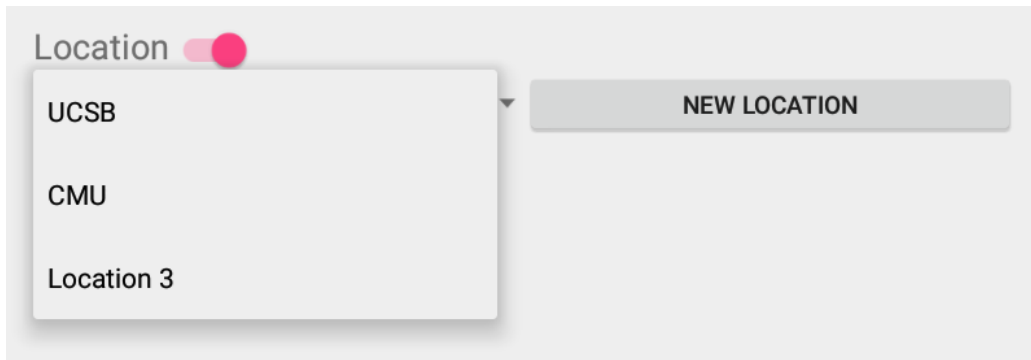


Figure 7 – Auto-generated “Location #”.

When it’s time for the alarm to sound and the location service is enabled for that particular alarm, it will detect user location first and compare it to the location specified for the alarm. An example is shown here with the alarm location set to University of California, Santa Barbara (UCSB) and us being on the Carnegie Mellon University campus. When the alarm is supposed to sound, it will see that the device is nowhere near UCSB as shown in the Log below.

```
.....
i/jingl.cmu.edu.drwaker D/nearLocation: Build GoogleApiClient
i/jingl.cmu.edu.drwaker I/GoogleApiClient: onConnected()
i/jingl.cmu.edu.drwaker E/CompareLocation: alarmLocation: (34.4139629, -119.848947)
i/jingl.cmu.edu.drwaker E/CompareLocation: mLastLocation: (40.4411056, -79.9437969)
i/jingl.cmu.edu.drwaker E/CompareLocation: Distance: 3563698.0m
i/jingl.cmu.edu.drwaker E/CompareLocation: Disable alarm due to location service
i/ClearcutLoggerApiImpl: disconnect managed GoogleApiClient
```

Figure 8 – Log showing the disabling alarm due to location service.

If the alarm does sound, it will jump to the math question or face detection feature. For math question, if the user enters the wrong answer, it will notify the user and let the user answer again. The alarm will continue to sound until the user gets the correct answer. After user input the right answer, it will turn off the alarm and jump back to the main page.

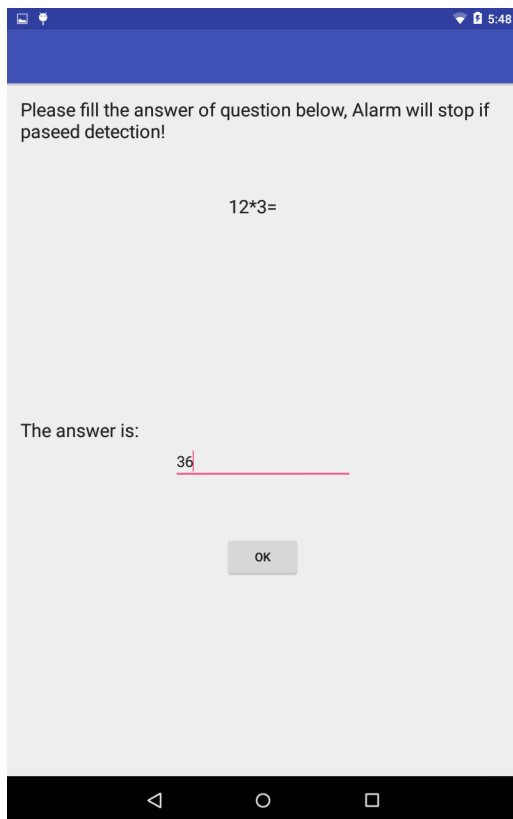


Figure 9 – Screen of entering the answer.

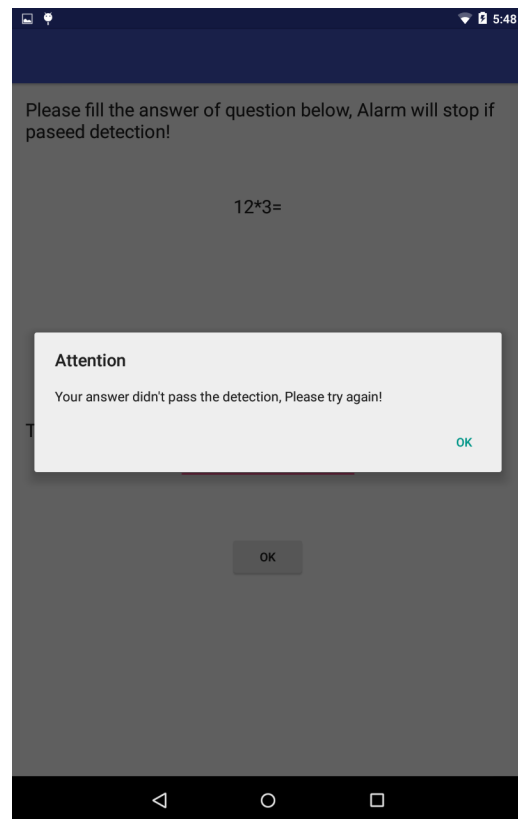


Figure 10 – The user entered the wrong answer.

If the user chose the feature of face detection, it will jump to face detection page. Similarly, the alarm will not turn off until it detects a face. If no face was detected, it will alert the user and allow the user another chance of turning off the alarm.



Figure 11 – No face in view.

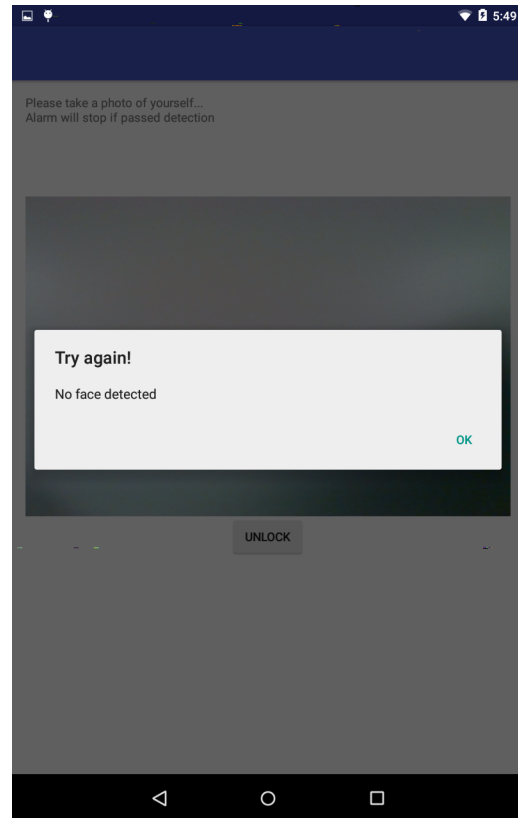


Figure 12 – No face was detected.

If the application detects that there is a face in front of the camera, it will turn off the alarm and go back to main page.

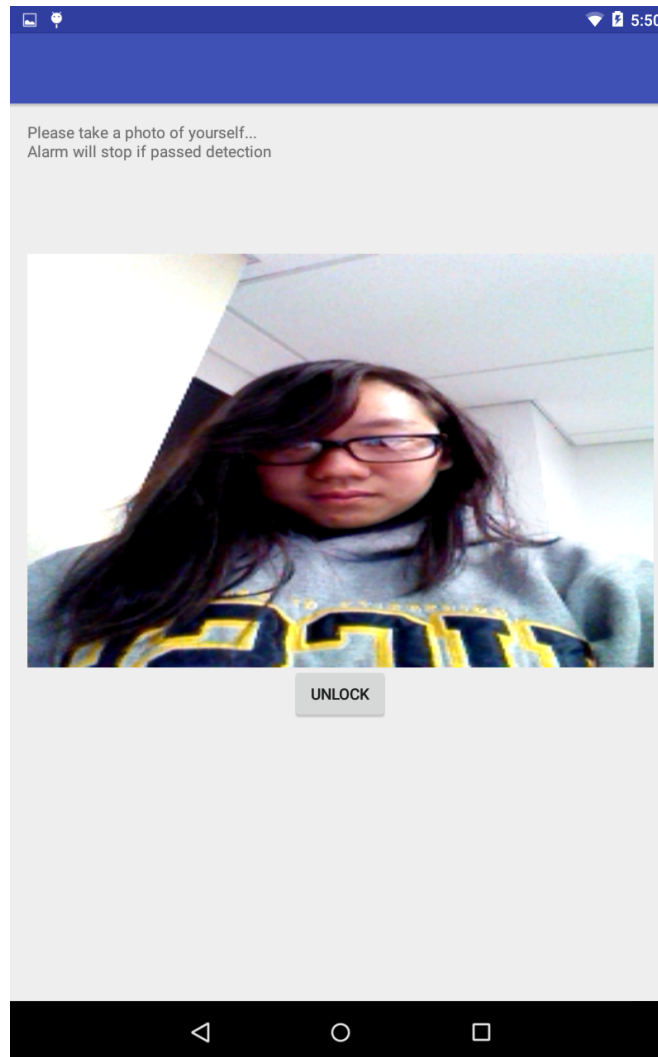


Figure 13 – A picture with a face is captured.