# CS-242/243

Project - #9

# Smart Warning System

Code Testing Document

**Course Instructor -**

Prof. Samit Bhattacharya

**Team - #23**

Jatin Goyal - 160101036

Namit Kumar - 160101046

Nitin Kedia - 160101048

# 1. Introduction

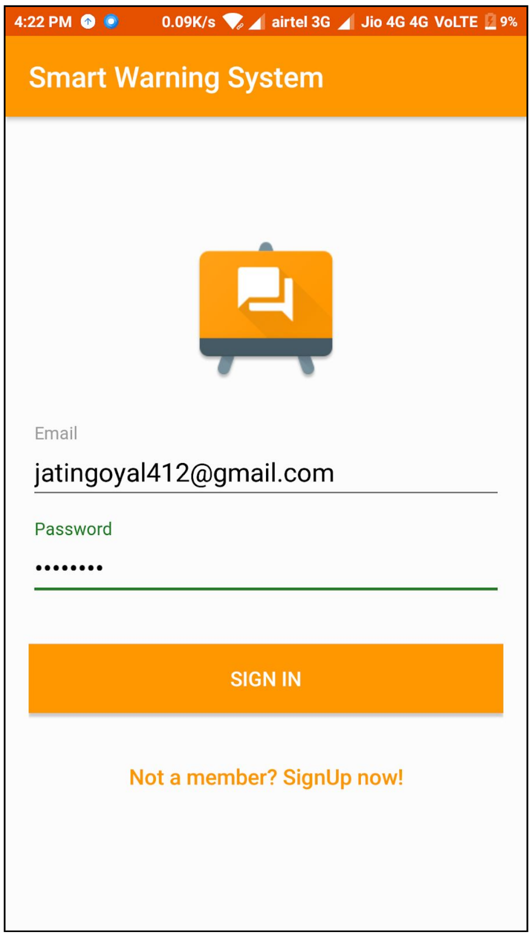
The purpose of this report is to document the Black Box and the White Box testing of our app, Smart Warning System. Unit testing has been performed only after the corresponding module was coded and successfully reviewed.
As a summary, this document is a means to equip the reader with the bugs, errors and the shortcomings of the app.

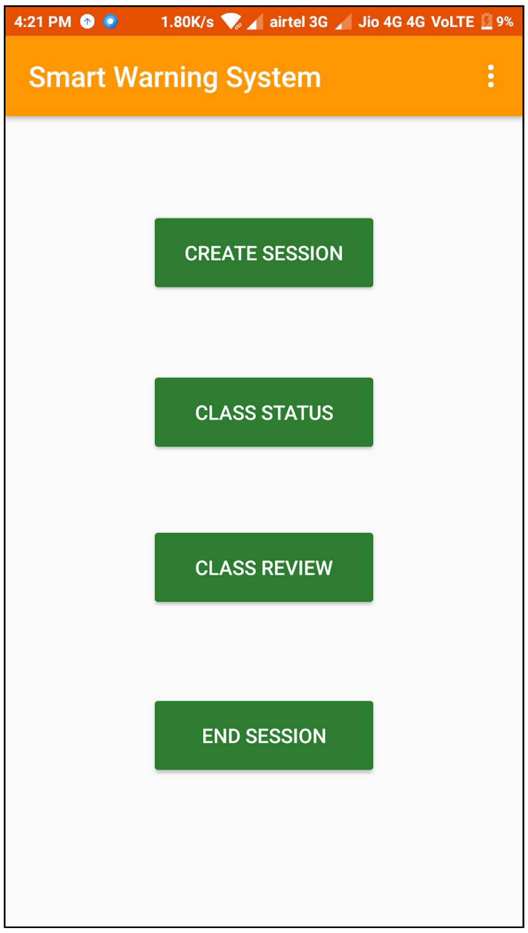
# 2. Black Box Testing

## 2.1 Login Module

Equivalence Classes:

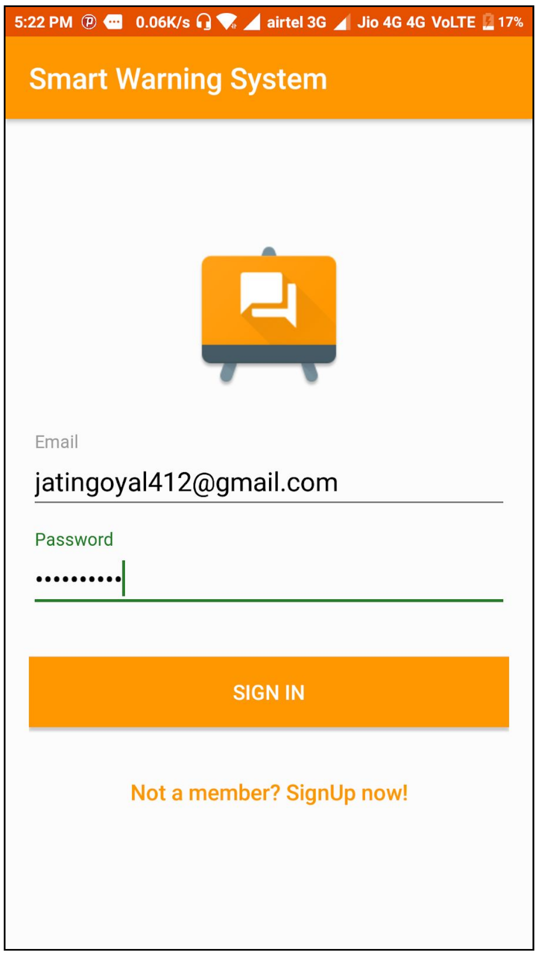
1. Both email and password are entered correctly.

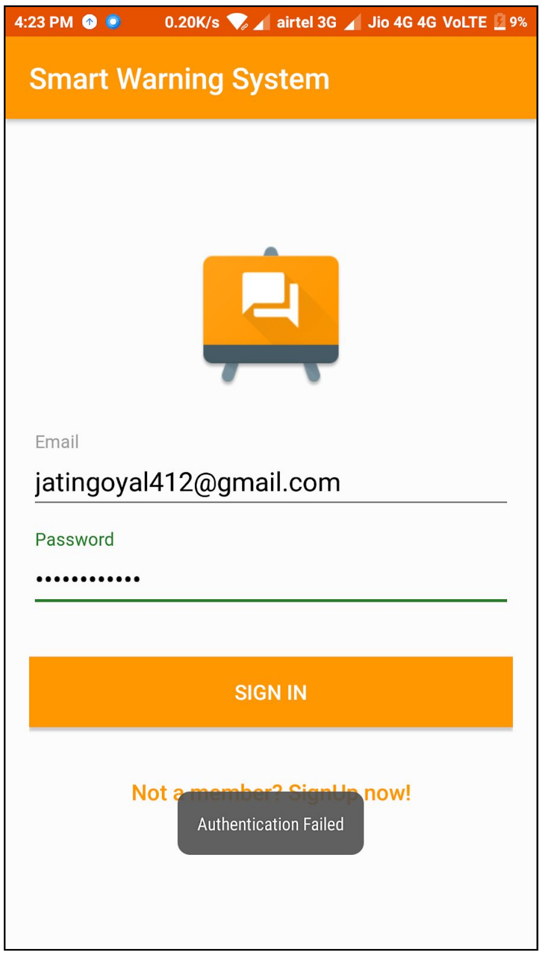




**Input**: Email- "jatingoyal412@gmail.com", Password- "abcdefgh"
**Expected Output**: Successful Login
**Observed Output**: Successful Login

2. At least one out of email and password is entered incorrectly.

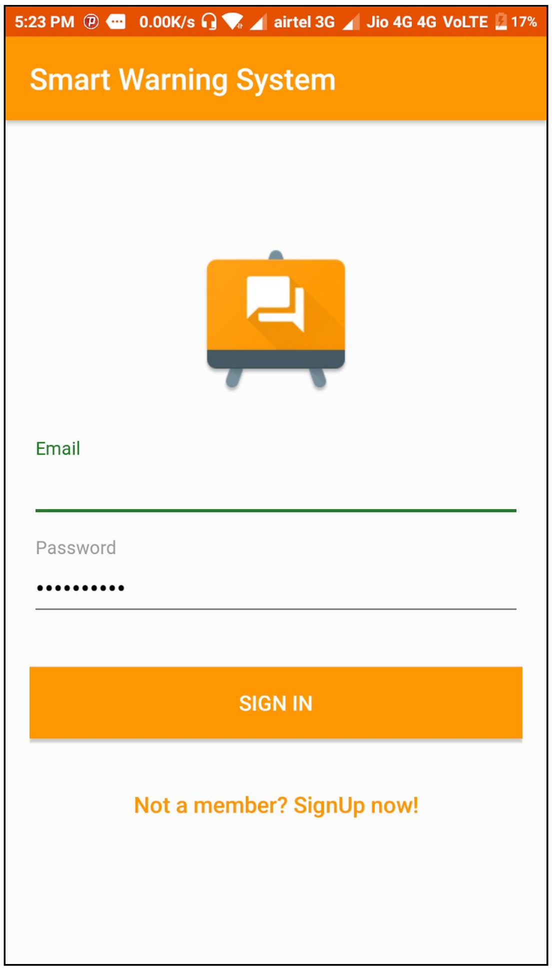


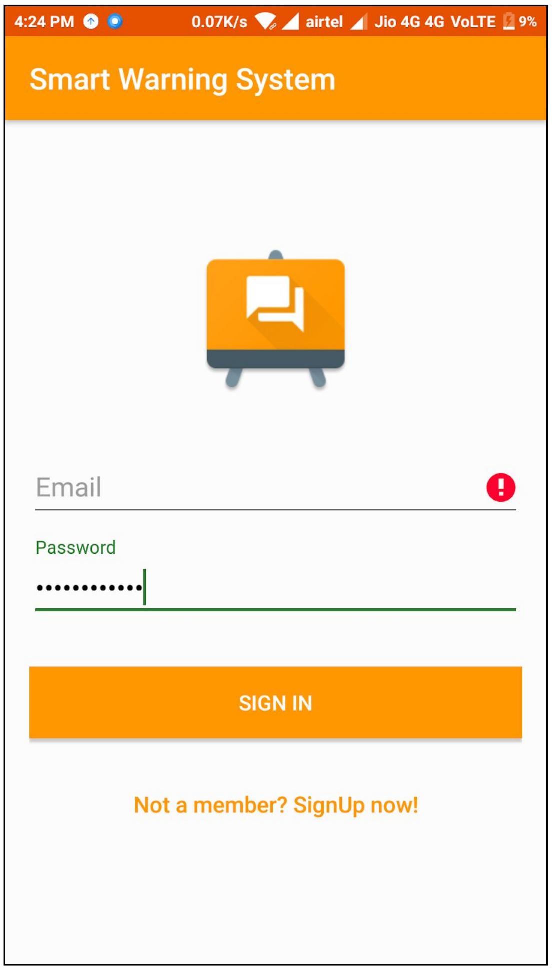


Authentication Failed

**Input**: Email- “jatingoyal412@gmail.com”, Password- “abcxyzsd”
**Expected Output**: Unsuccessful Login
**Observed Output**: Unsuccessful Login

3. At least one out of email and password field has been left blank.


5:23 PM 0.00K/s airtel 3G Jio 4G 4G VoLTE 17%
Smart Warning System
Email
Password
SIGN IN
Not a member? SignUp now!

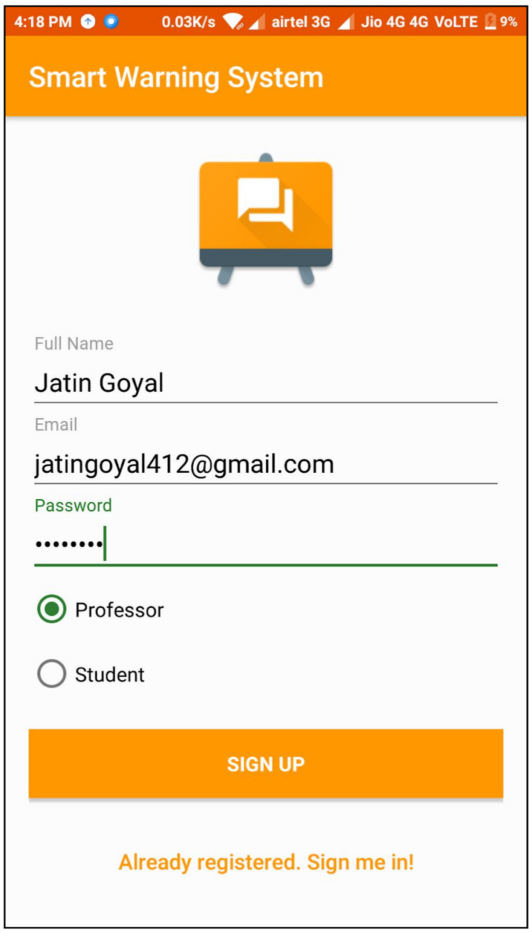

4:24 PM 0.07K/s airtel Jio 4G 4G VoLTE 9%
Smart Warning System
Email
Password
SIGN IN
Not a member? SignUp now!

**Input**: “{Empty}”, “abcxyzsd”
**Expected Output**: Red Error exclamation
**Observed Output**: Unsuccessful Login with error exclamation

## 2.2 Registration Module

Equivalence Classes:

1. Full Name, Email and Password has been entered in correct format.
   User type has also been selected.



4:18 PM 0.03K/s airtel 3G Jio 4G 4G VoLTE 9%

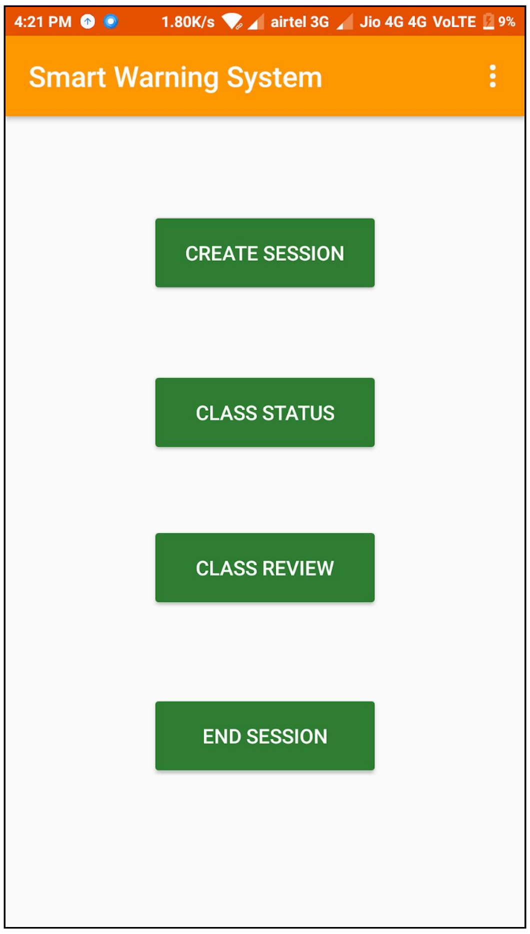
Smart Warning System

Full Name: Jatin Goyal

Email: jatingoyal412@gmail.com

Password: ••••••••

[✓ Professor] Student

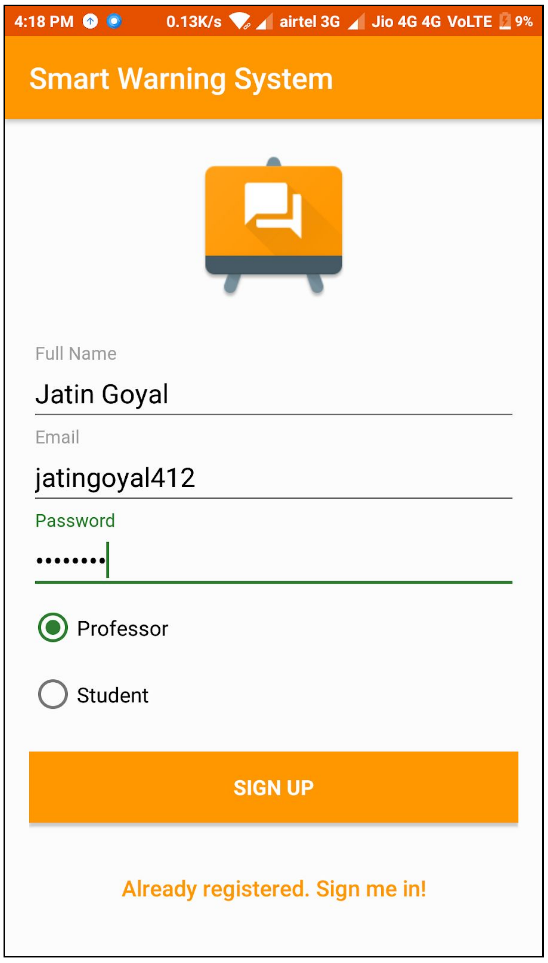


4:21 PM 1.80K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

**Input**: "Jatin Goyal", "jatingoyal412@gmail.com", "abcdefgh", Professor
**Expected Output**: Successful Registration
**Observed Output**: Successful Registration

2. Email has not been suffixed with “@abc.xyz”.(“abc” and “xyz” are strings)


4:18 PM 0.13K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

Full Name

Jatin Goyal

Email

jatingoyal412

Password

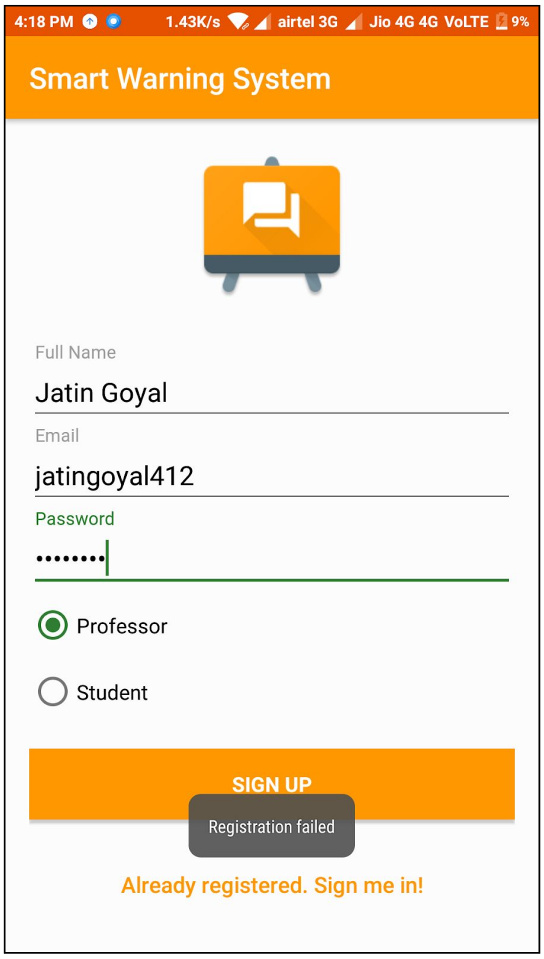
••••••••

[✓ Professor]

Student

SIGN UP

Already registered. Sign me in!


4:18 PM 1.43K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

Full Name

Jatin Goyal

Email

jatingoyal412

Password

••••••••

[✓ Professor]

Student

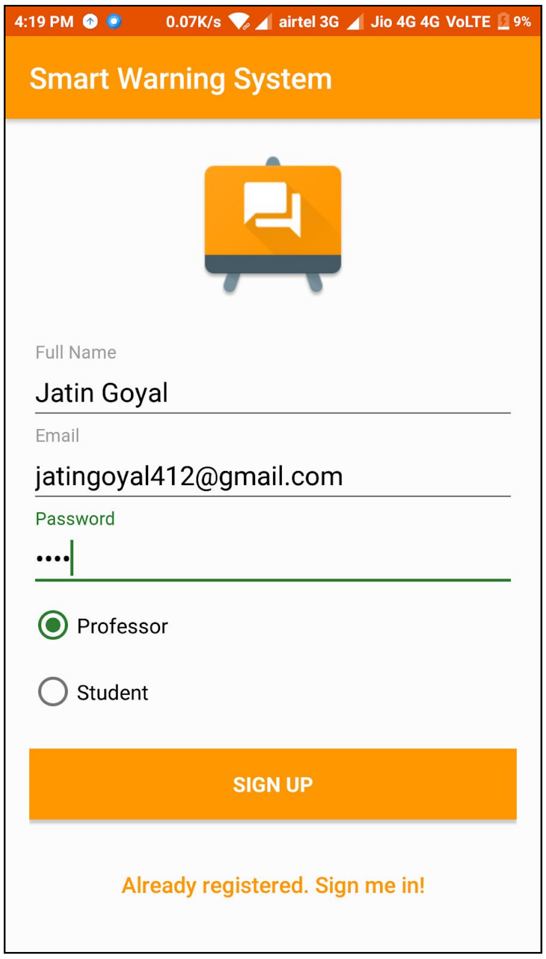
SIGN UP

Registration failed

Already registered. Sign me in!

**Input**: “Jatin Goyal”, “jatingoyal412”, “abcdefgh”
**Expected Output**: Unsuccessful Registration
**Observed Output**: Unsuccessful Registration

3. Password entered has length less than 6 characters.



4:19 PM 0.07K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

Full Name
Jatin Goyal

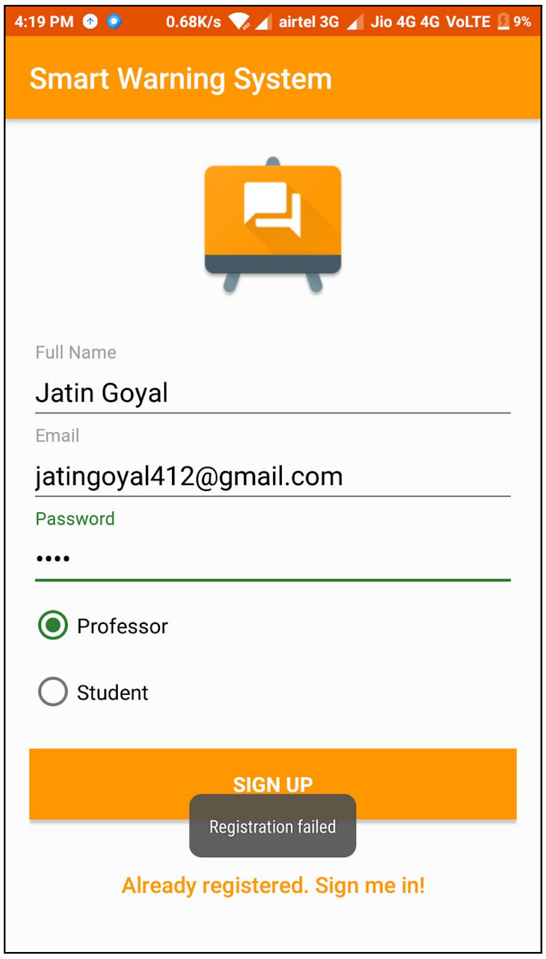
Email
jatingoyal412@gmail.com

Password
••••

[✓ Professor]
Student

SIGN UP

Already registered. Sign me in!



4:19 PM 0.68K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

Full Name
Jatin Goyal

Email
jatingoyal412@gmail.com

Password
••••

[✓ Professor]
Student

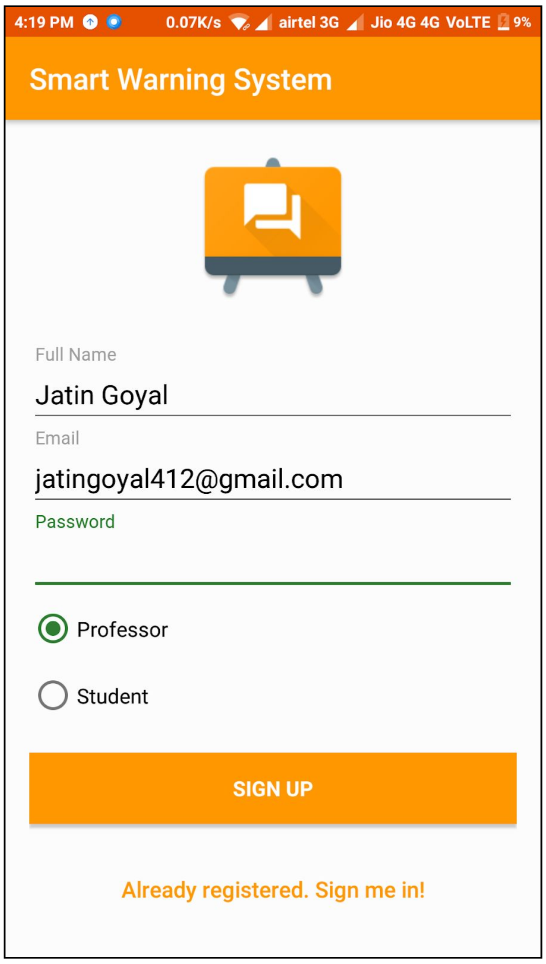
SIGN UP

Registration failed

Already registered. Sign me in!

**Input**: “Jatin Goyal”, “jatingoyal412@gmail.com”, “abcd”
**Expected Output**: Unsuccessful Registration
**Observed Output**: Unsuccessful Registration

4. At least one field out of Email, Password and Full name has been left blank.


4:19 PM 0.07K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

Full Name

Jatin Goyal

Email

jatingoyal412@gmail.com

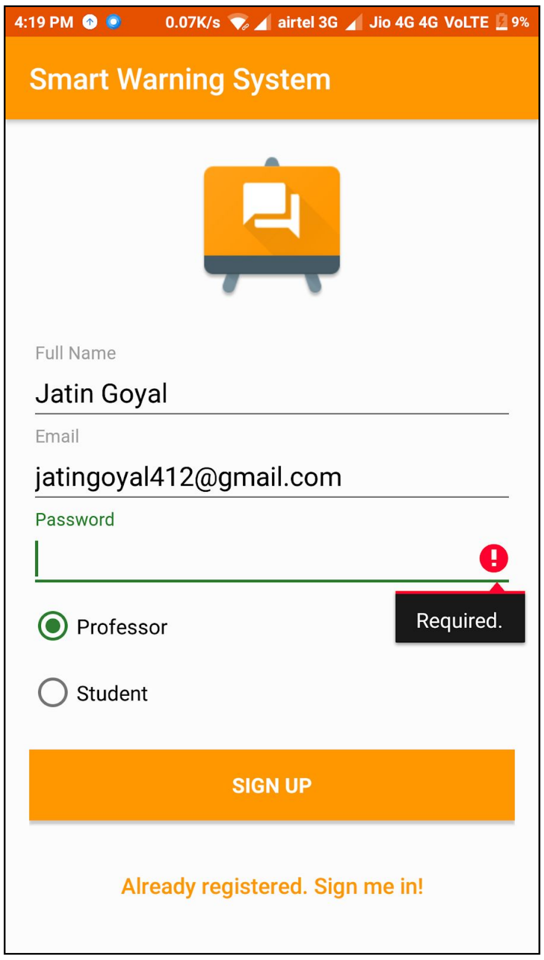
Password

Professor

Student

SIGN UP

Already registered. Sign me in!


4:19 PM 0.07K/s airtel 3G Jio 4G 4G VoLTE 9%

Smart Warning System

Full Name

Jatin Goyal

Email

jatingoyal412@gmail.com

Password

Required.

Professor

Student

SIGN UP

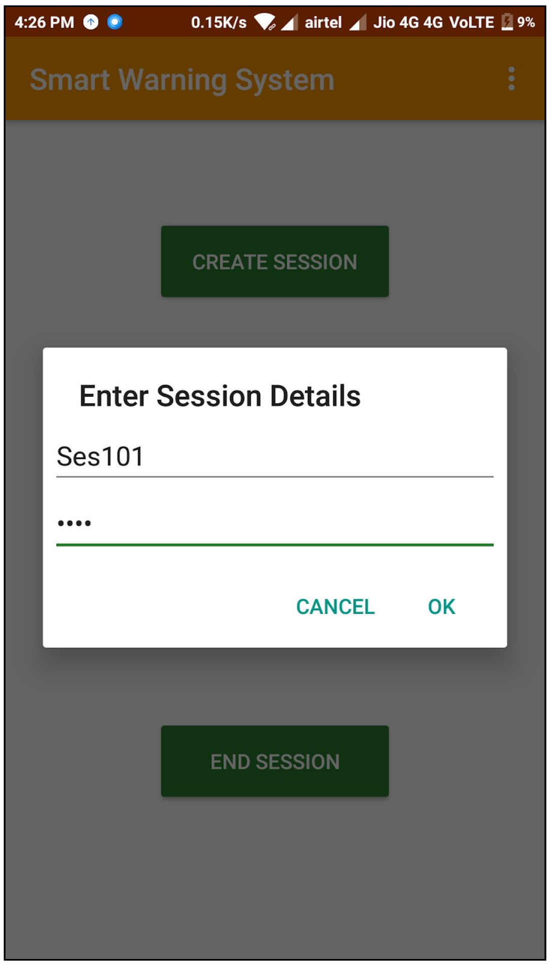
Already registered. Sign me in!

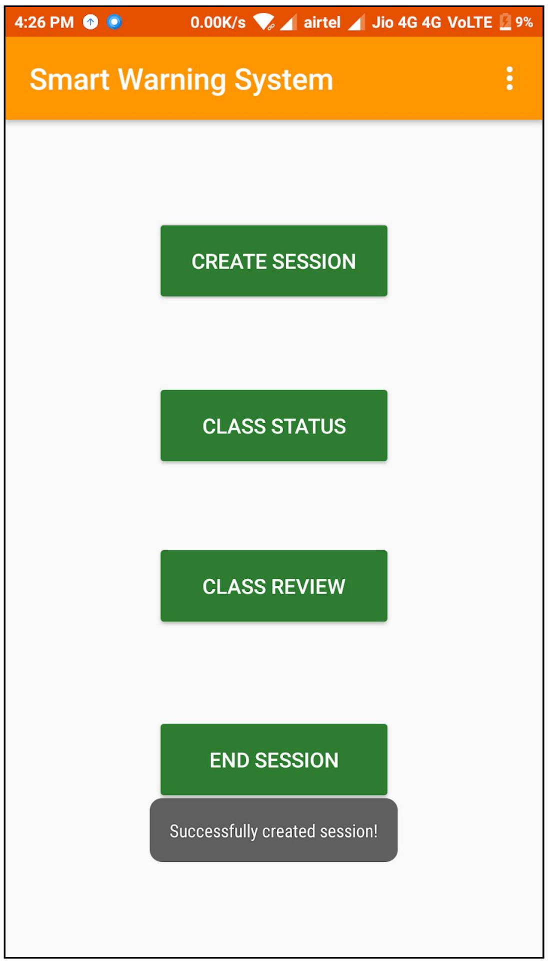
**Input**: “Jatin Goyal”, “jatingoyal412@gmail.com”, “{Empty}”
**Expected Output**: Unsuccessful Registration with “Required” alert
**Observed Output**: Same as Expected.

## 2.3 Create Session Module

Equivalence Classes:

1. There is no current session active and Session Name and Password are entered.

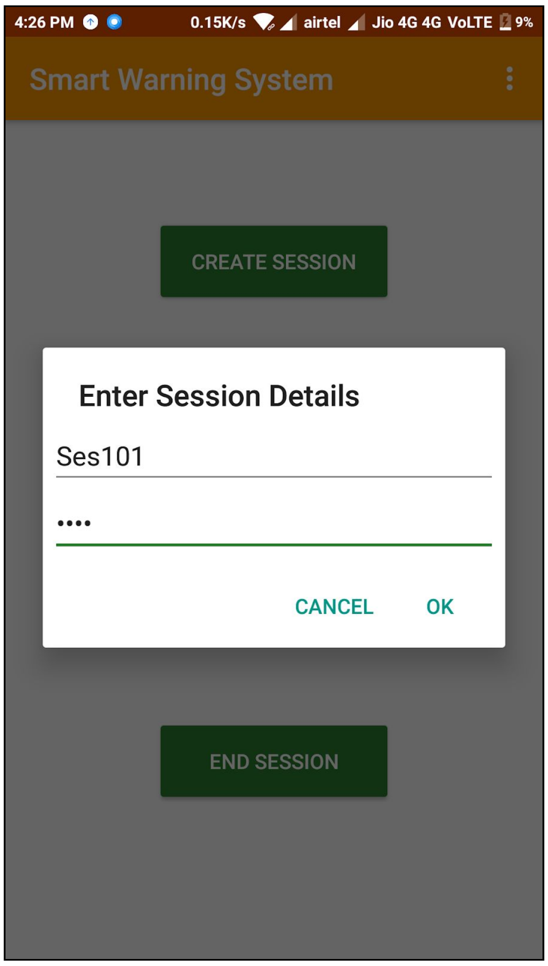




Successfully created session!

**Input**: “Ses101”, “abcd”
**Expected Output**: Successful Session creation
**Observed Output**: As expected.

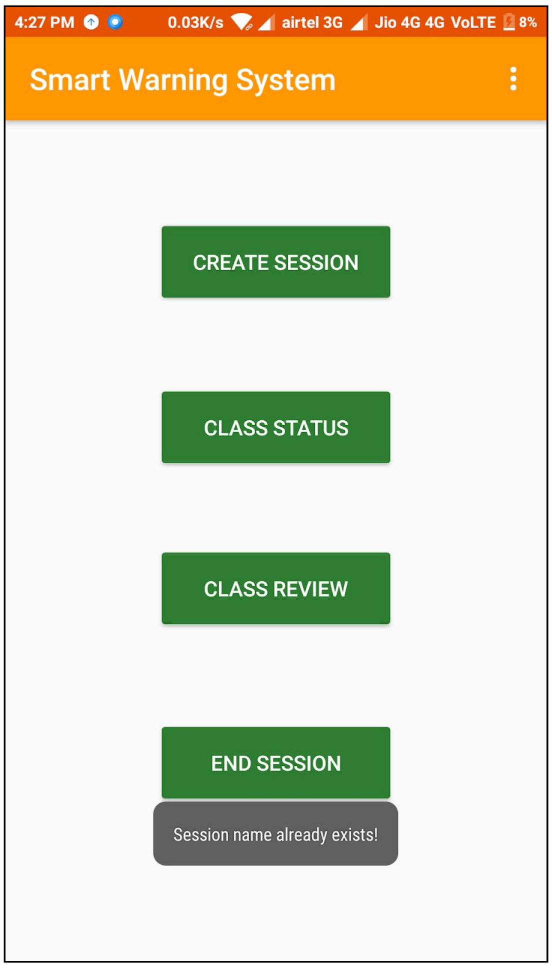
2. There is no current session active and the Session Name already exists.



Enter Session Details

Ses101

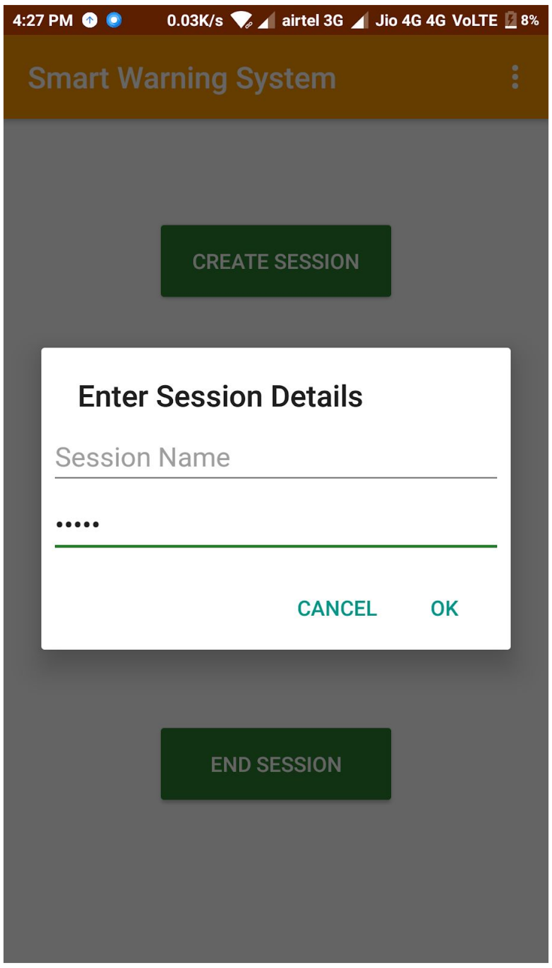
••••

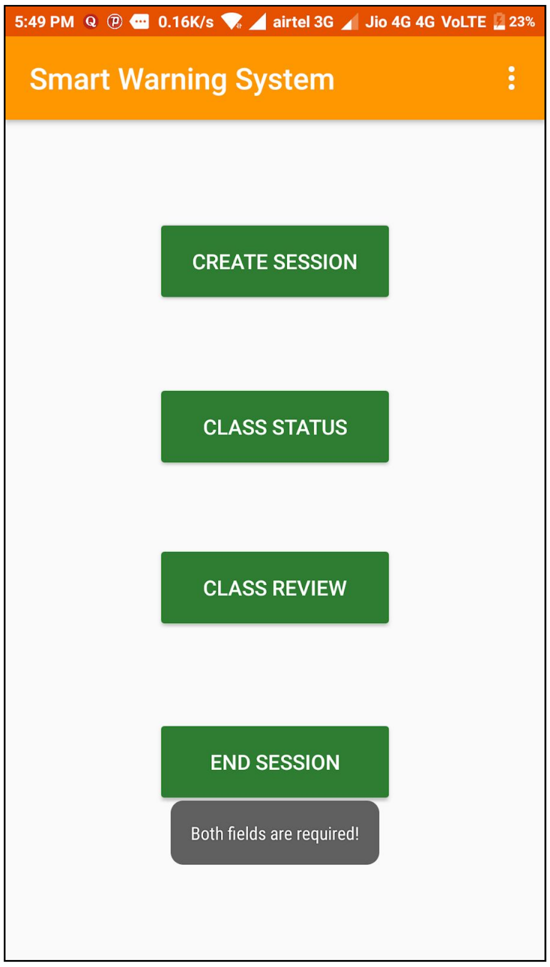


Session name already exists!

**Input**: “Ses101”, ““abcd”
**Expected Output**: Already exist Alert message
**Observed Output**: As expected.

3. There is no current session active and at least one out of Session Name and Password field is left blank.

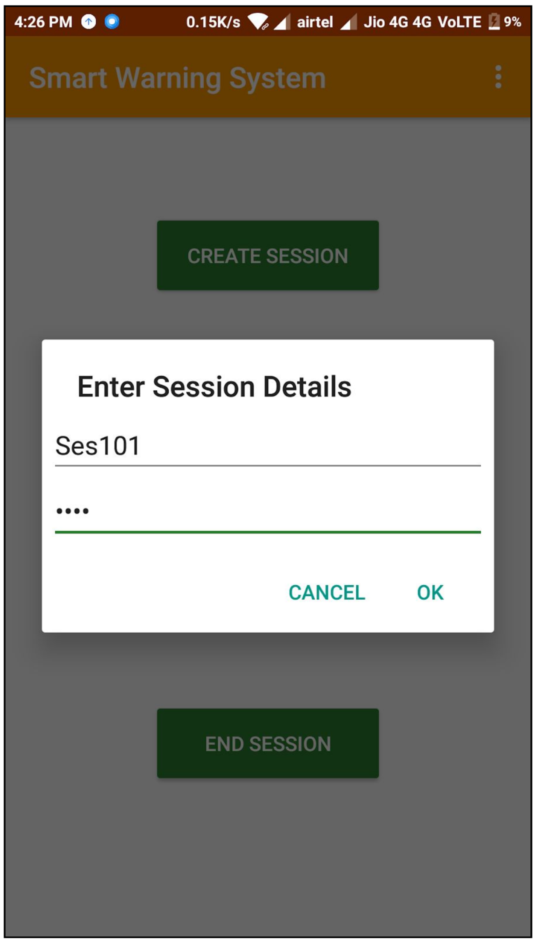


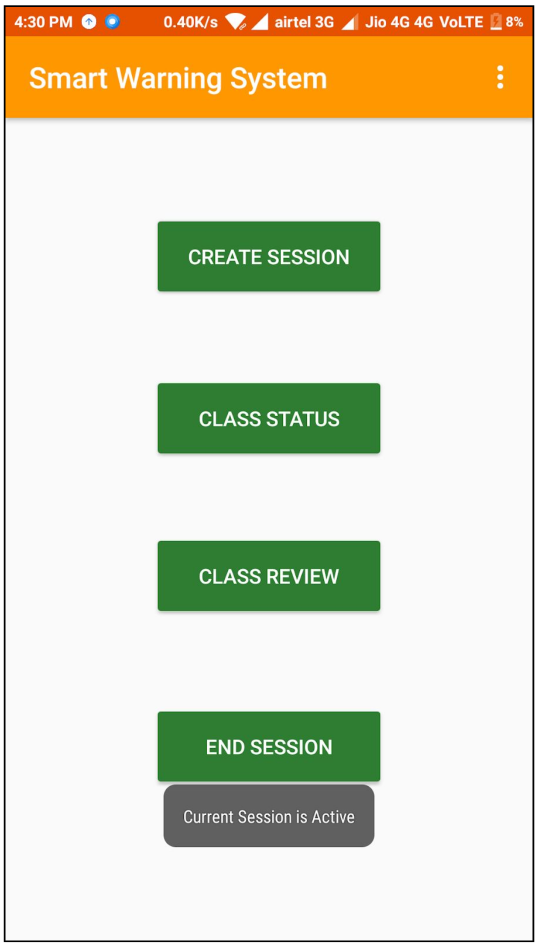


Both fields are required!

**Input**: “{Empty}”, “abcd”
**Expected Output**: Message asking for both fields
**Observed Output**: As Expected.

4. There is already a session active.





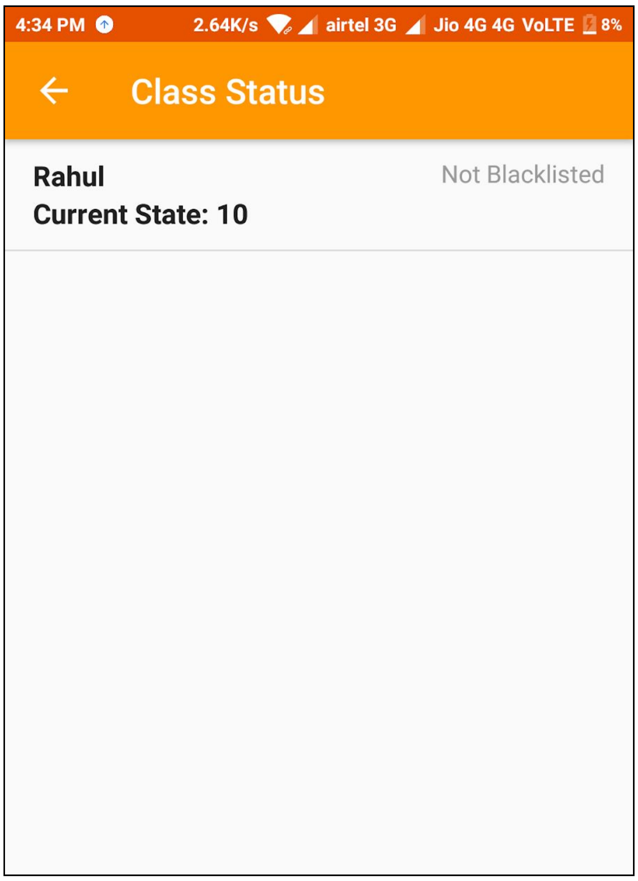
Current Session is Active

**Input**: “Ses101”, “abcd”
**Expected Output**: Message saying that a Current session is Active
**Observed Output**: As Expected.

## 2.4 Class Status Module

Equivalence Classes:

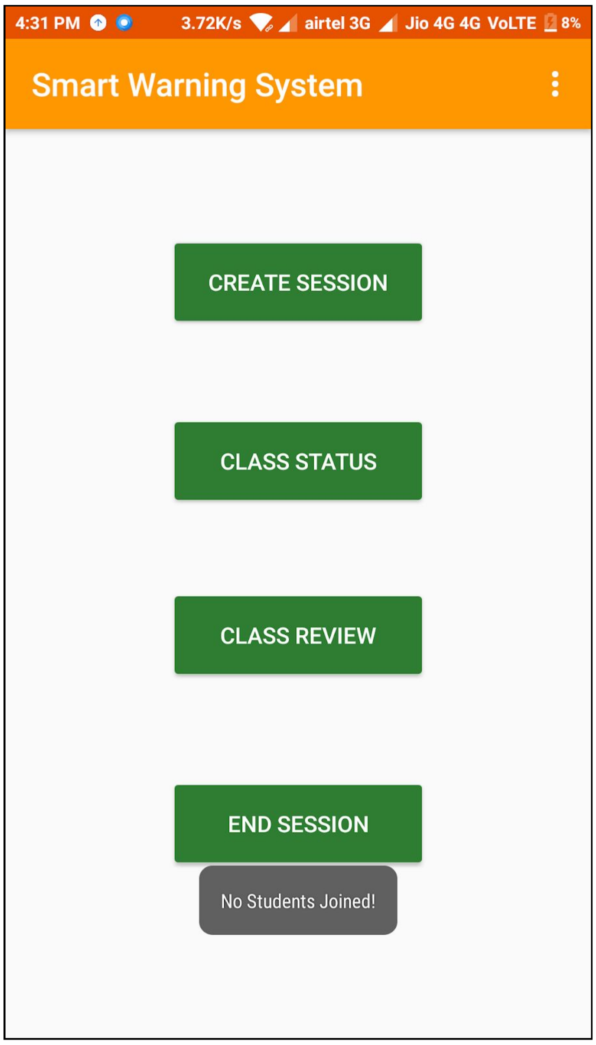
1. A session is running and at least one student has joined.



Rahul
Current State: 10
Not Blacklisted

**Input**: Tap on class status button
**Expected Output**: Class status window opens
**Observed Output**: As Expected

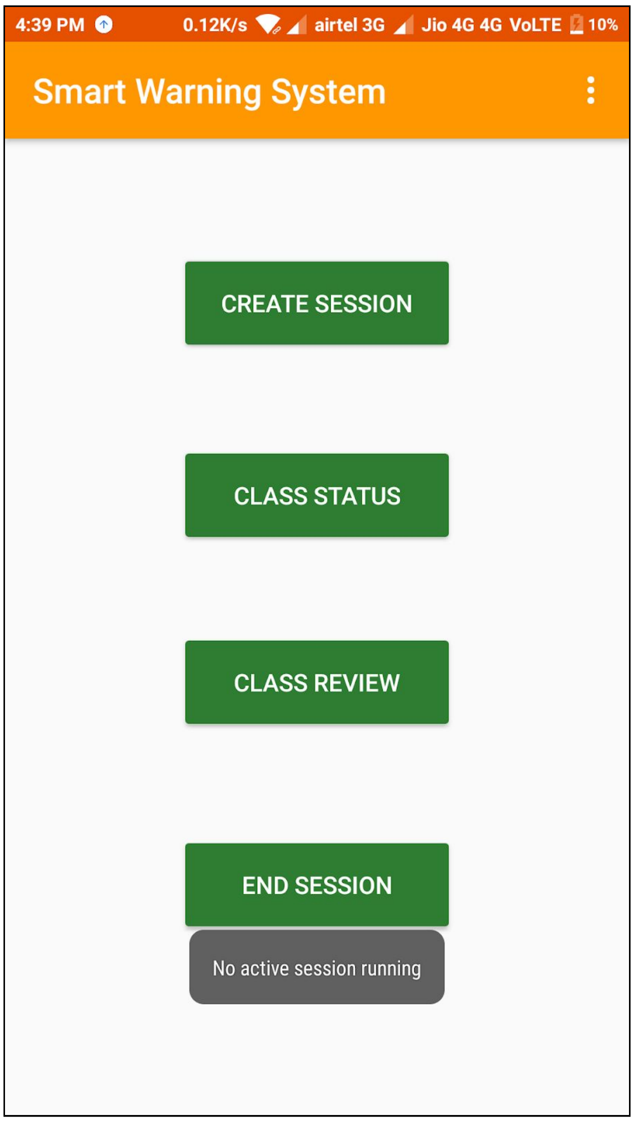
2. A session is running and no students have joined.



No Students Joined!

**Input**: Tap on Class Status Button
**Expected Output**: “No Students” message
**Observed Output**: As Expected

3. No session is active.



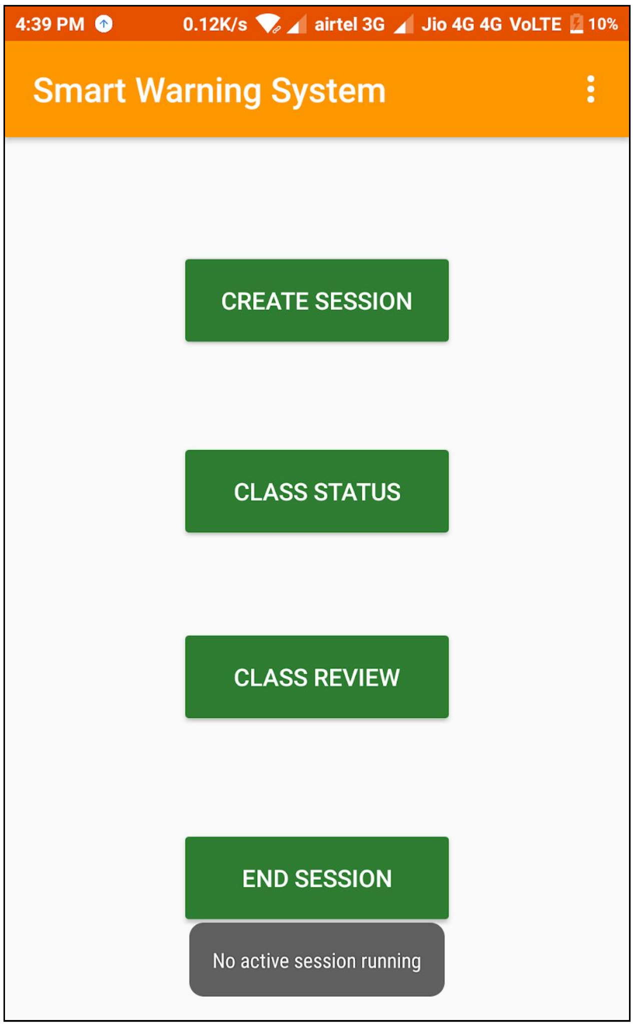
No active session running

**Input**: Tap on Class Status Button
**Expected Output**: Message saying no active session exists.
**Observed Output**: As Expected.

## 2.5 Class Review Module

Equivalence Classes:

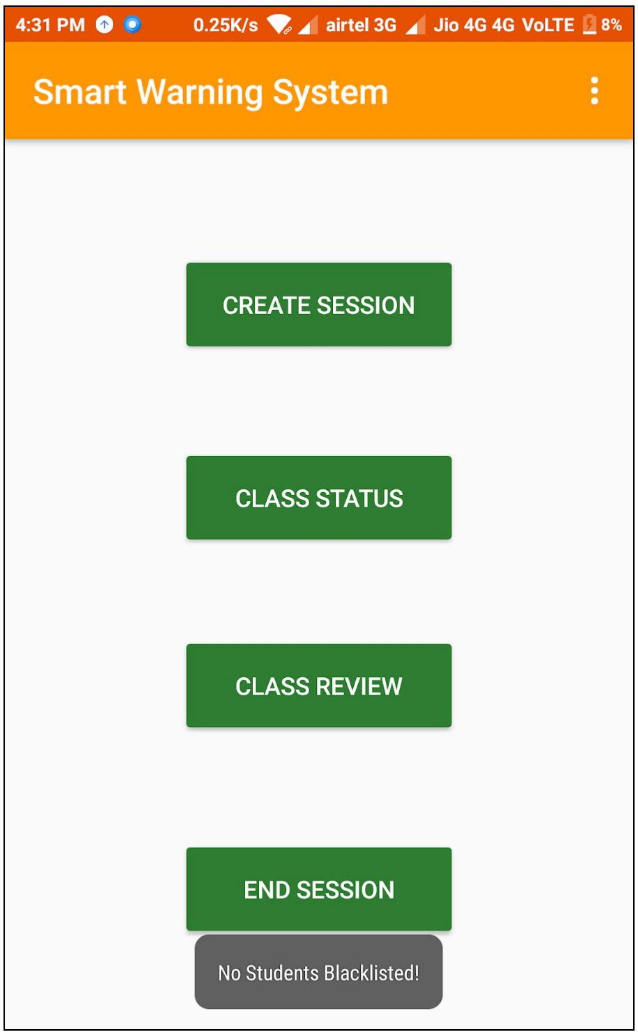
1. No session is active.



No active session running

**Input**: Tap on class Review Button
**Expected Output**: Message saying no active session exists.
**Observed Output**: As Expected.

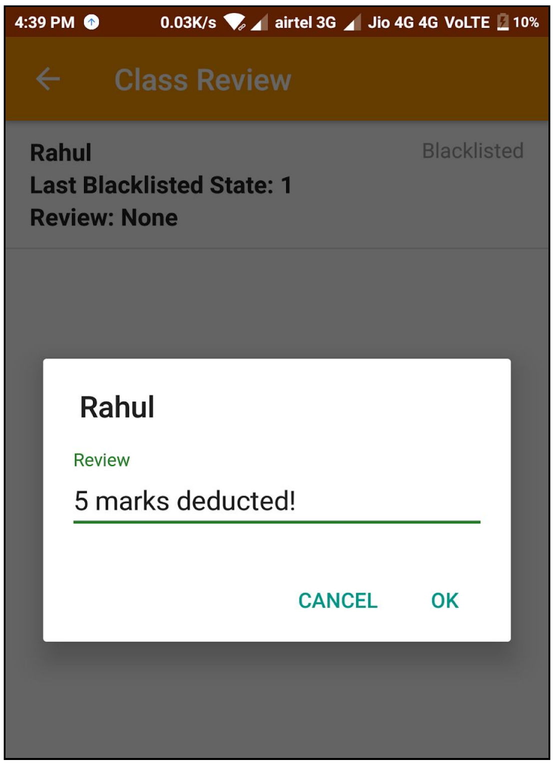
2. A session is running and no students have been blacklisted.



No Students Blacklisted!

**Input**: Tap on Class Review Button
**Expected Output**: Message saying there are no Blacklisted Students.
**Observed Output**: As Expected.

3. A session is running and at least one student has been blacklisted. Remark is entered.


4:39 PM 0.03K/s airtel 3G Jio 4G 4G VoLTE 10%

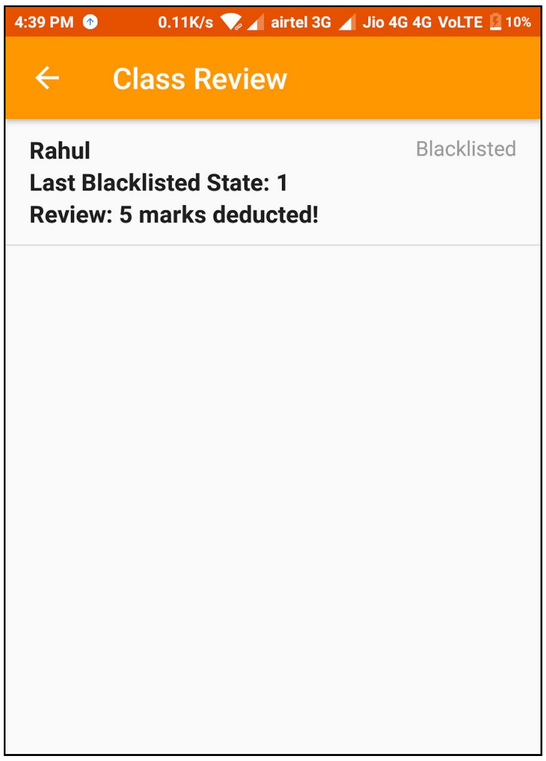
Class Review

Rahul
Blacklisted
Last Blacklisted State: 1
Review: None

Rahul

Review

5 marks deducted!

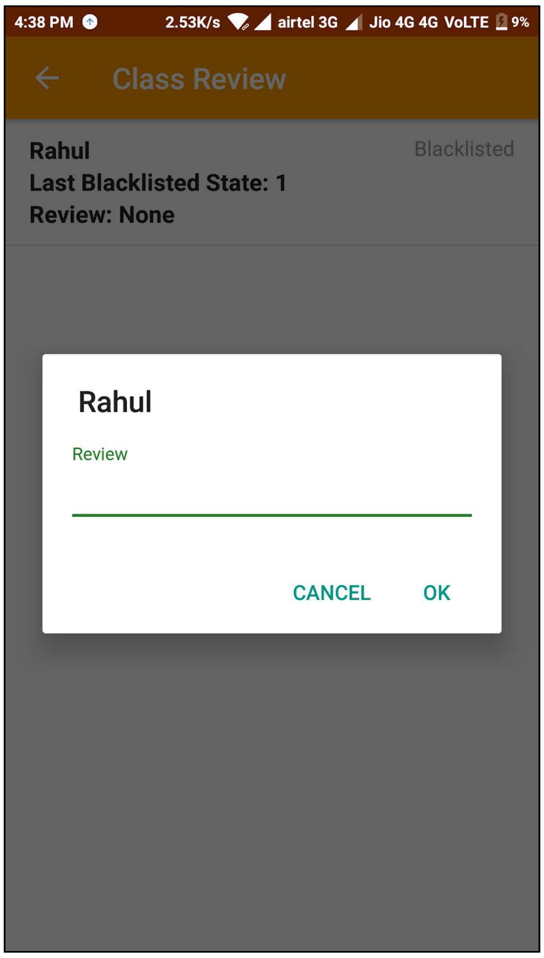

4:39 PM 0.11K/s airtel 3G Jio 4G 4G VoLTE 10%

Class Review

Rahul
Blacklisted
Last Blacklisted State: 1
Review: 5 marks deducted!

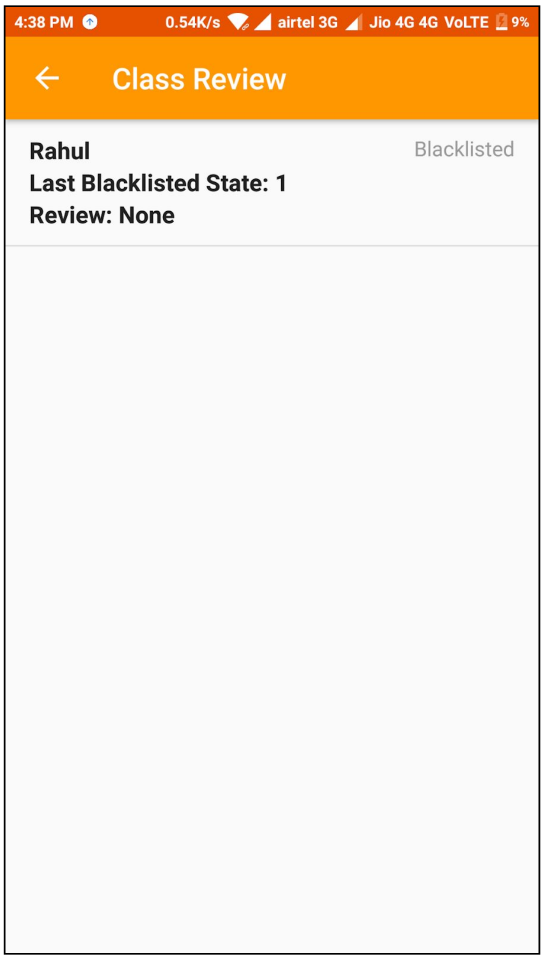
**Input**: Tap on class Review Button, Tap on Student, enter the remark “5 marks Deducted!”
**Expected Output**: Review is recorded next to Blacklisted Stuudent.
**Observed Output**: As Expected.

4. A session is running and at least one student has been blacklisted. Remark is not entered.



Rahul Blacklisted
Last Blacklisted State: 1
Review: None

Rahul
Review



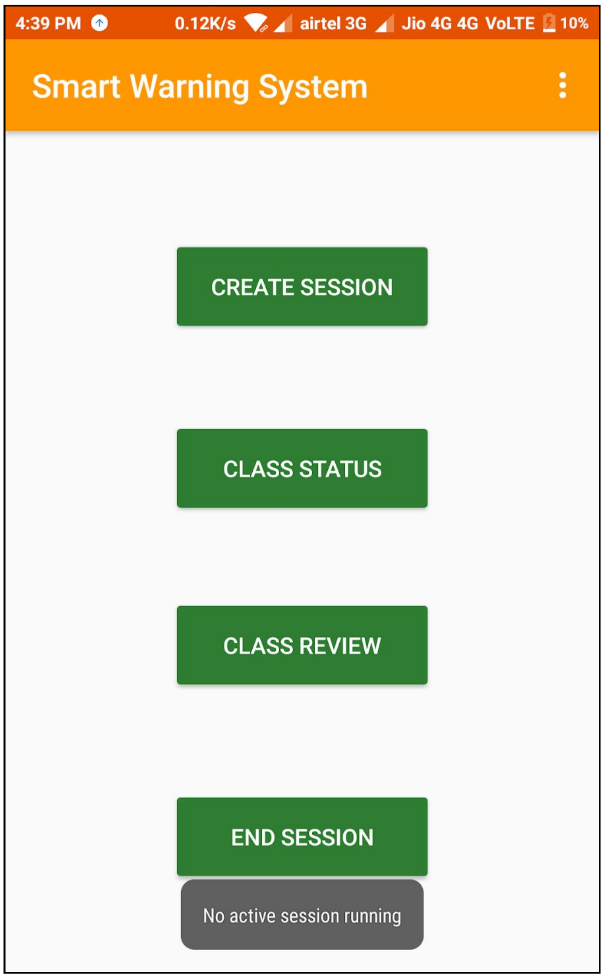
Rahul Blacklisted
Last Blacklisted State: 1
Review: None

**Input**: Tap on class Review Button, Tap on Student, Leave the Review field blank.
**Expected Output**: Review is recorded as “None” next to Blacklisted Student.
**Observed Output**: As Expected.

## 2.6 End Session Module

Equivalence Classes:

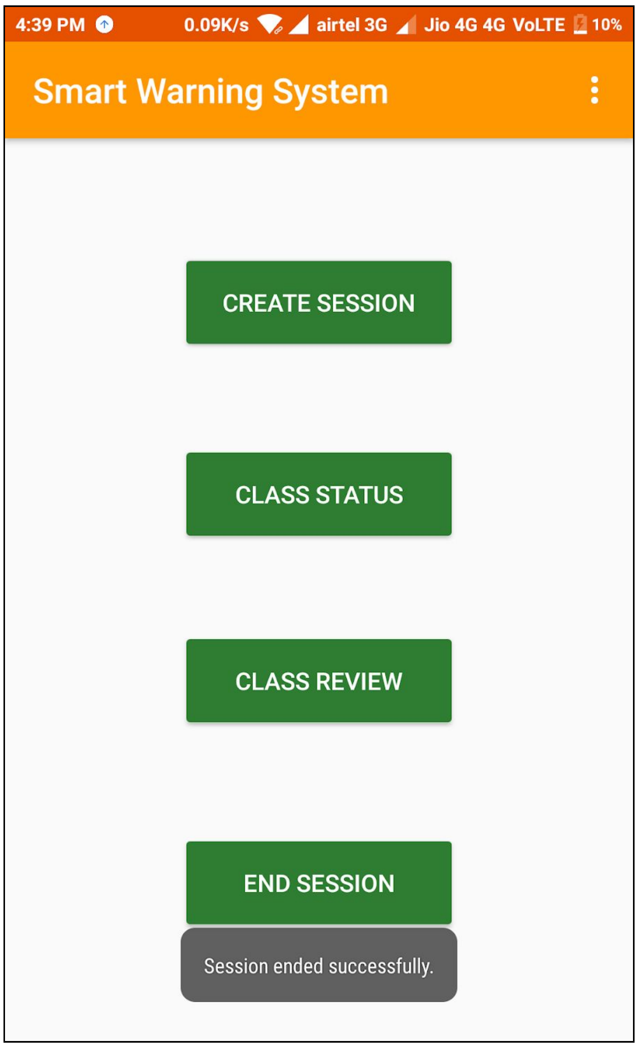
1. No session is active.



No active session running

**Input**: Tap on End Session Button
**Expected Output**: Message saying that no Session was running.
**Observed Output**: As Expected.

2. A session is currently running.



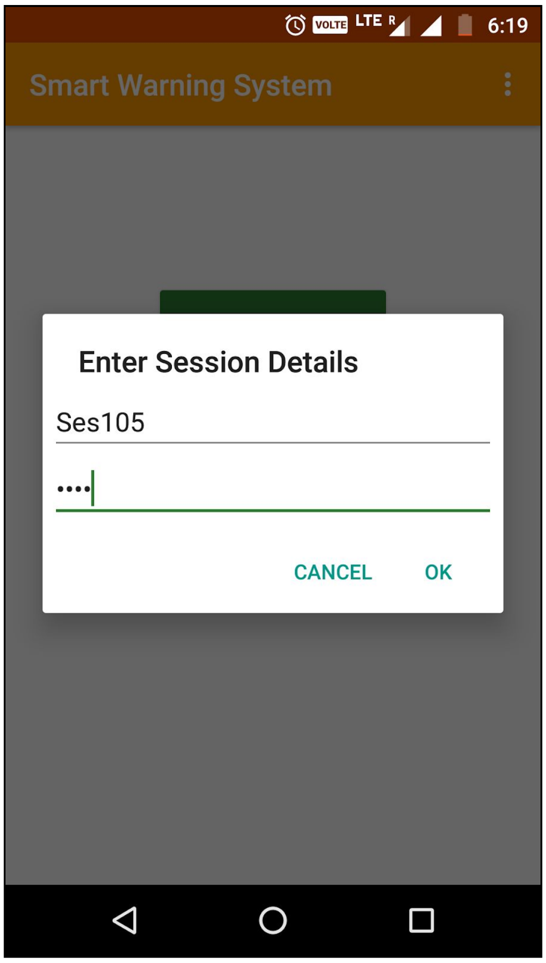
Session ended successfully.

**Input**: Tap on class End Session Button
**Expected Output**: Session is stopped with message..
**Observed Output**: As Expected.

## 2.7 Join Session Module

Equivalence Classes:

1. Both Session Name and Password are entered correctly.

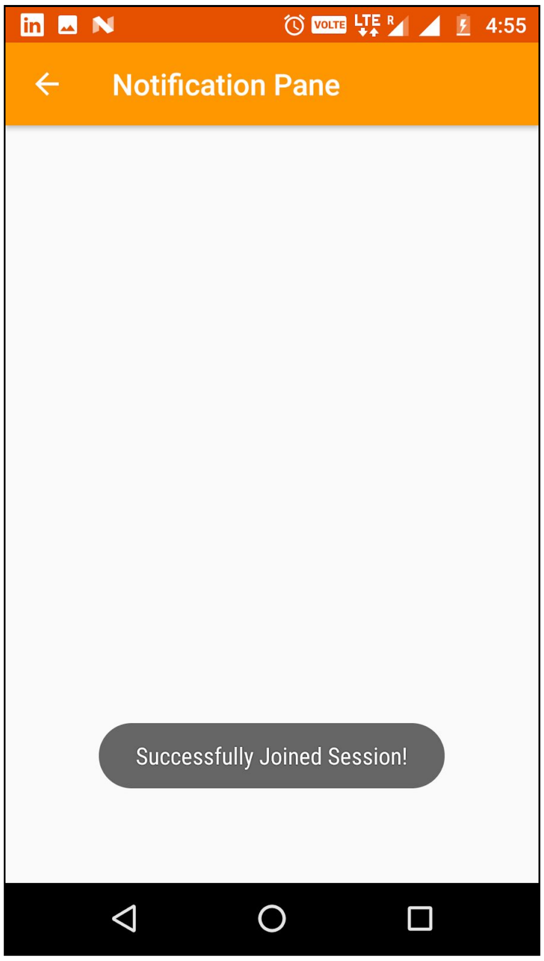


Smart Warning System

Enter Session Details

Ses105

••••

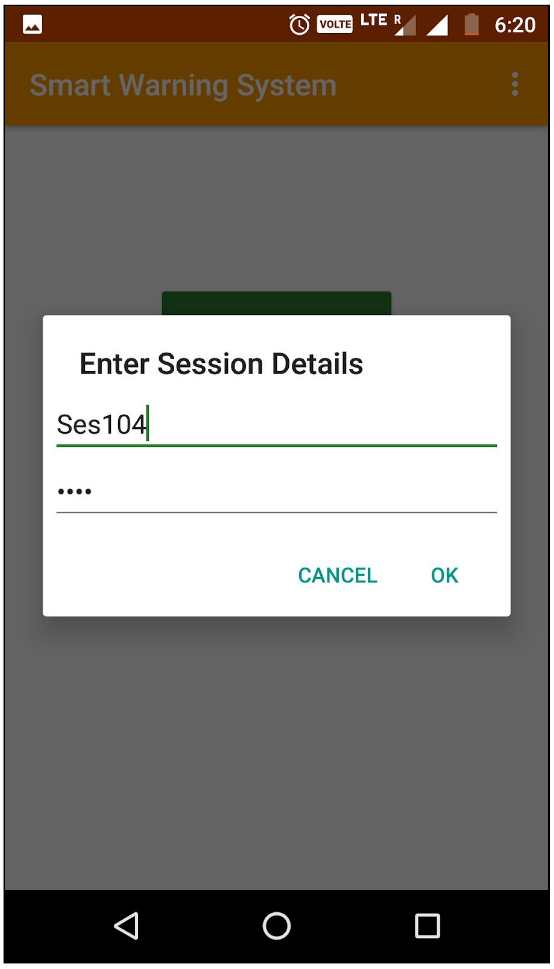


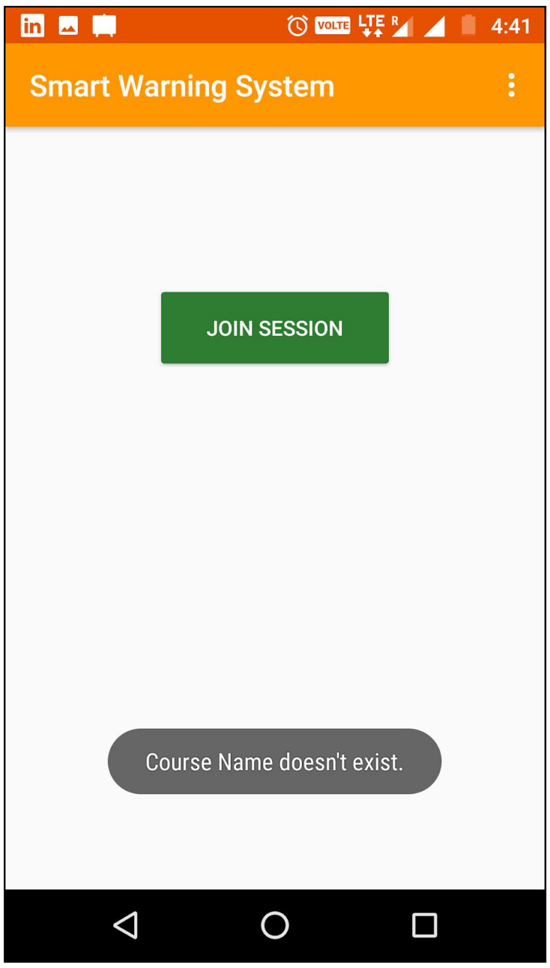
Notification Pane

Successfully Joined Session!

**Input**: Input “Ses105”, “abcd”
**Expected Output**: A new session is opened with message.
**Observed Output**: As Expected.

2. At least one out of Session Name and Password is entered wrong.

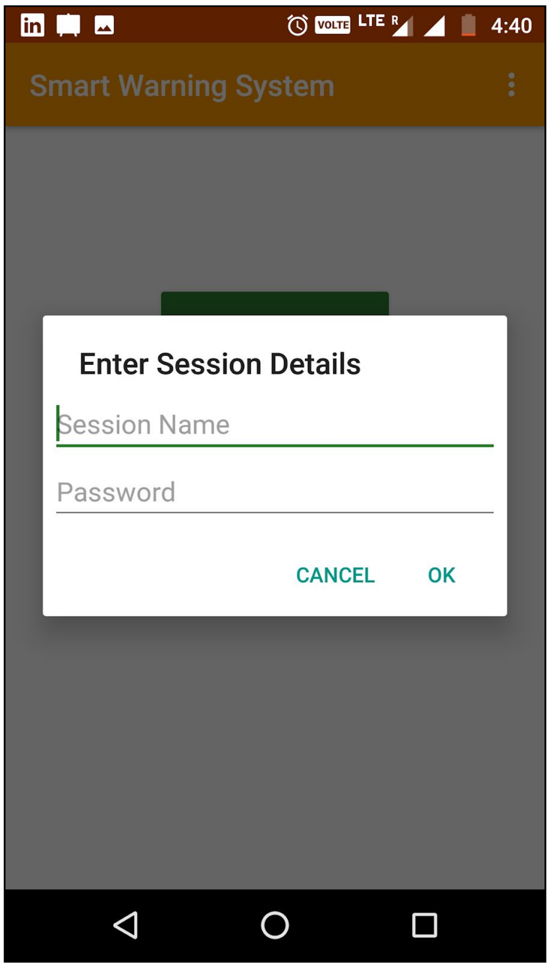


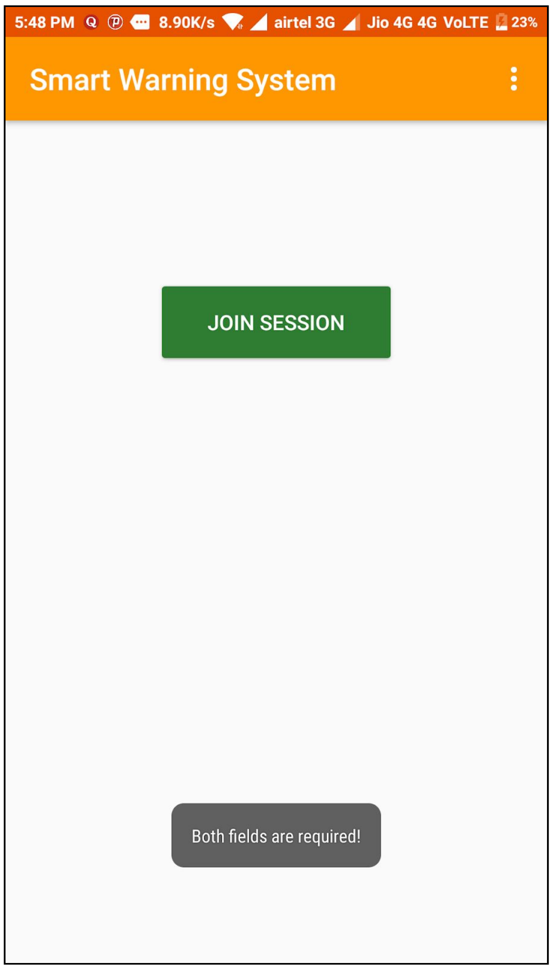


Course Name doesn't exist.

**Input**: Input “Ses104”, “abcd”
**Expected Output**: Error Message Informing that corresponding session doesn’t exist.
**Observed Output**: As Expected.

3. At least one out of Session Name and Password is left blank.

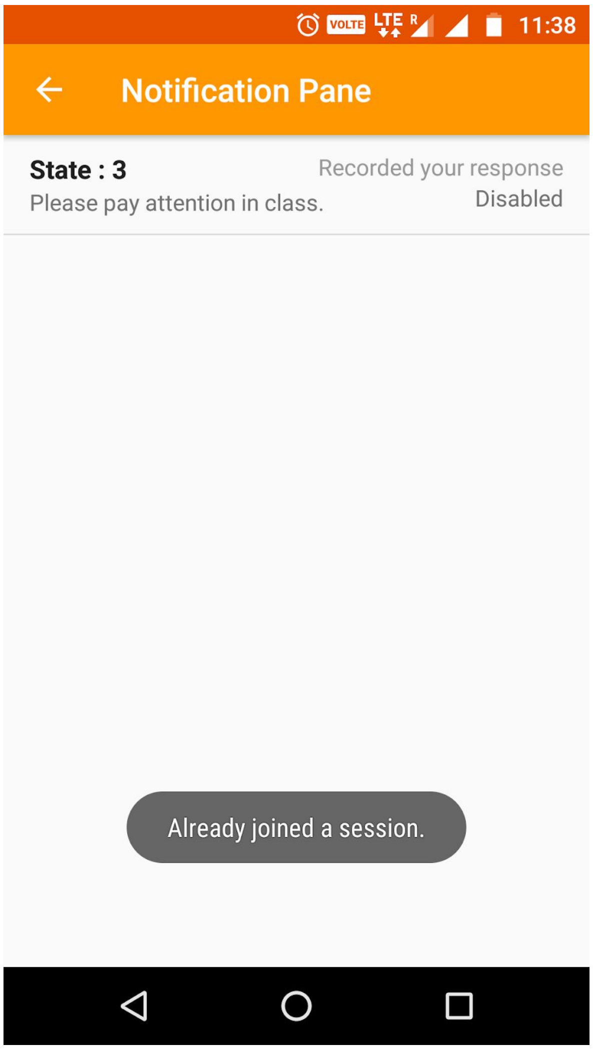




Both fields are required!

**Input**: Both fields are left Blank and OK is tapped.
**Expected Output**: Error message saying that both fields are required.
**Observed Output**: As Expected.

4. A session is already joined.



**State : 3**
Please pay attention in class.
Recorded your response
Disabled

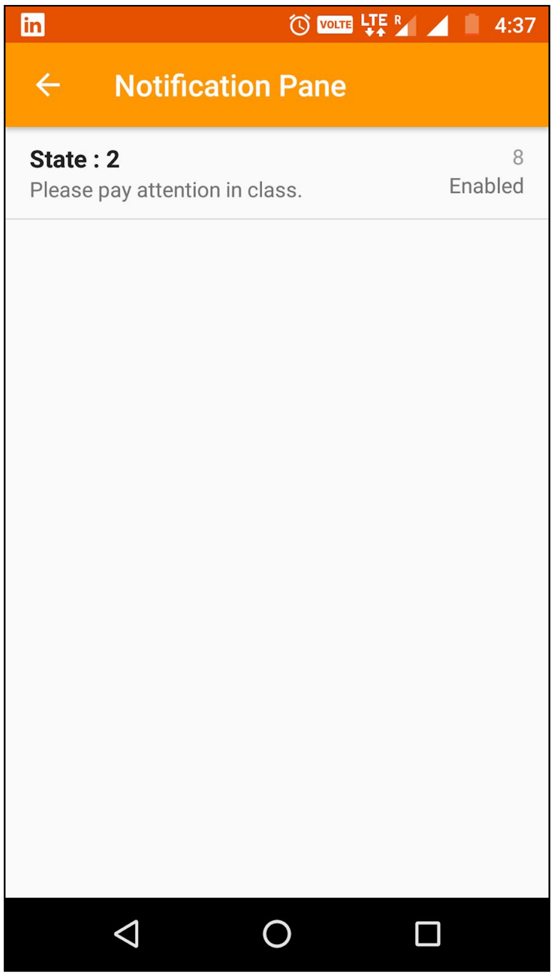
Already joined a session.

**Input**: Join Session is tapped
**Expected Output**: Notification window opens with Already Joined message.
**Observed Output**: As Expected.

## 2.8 Notification Module

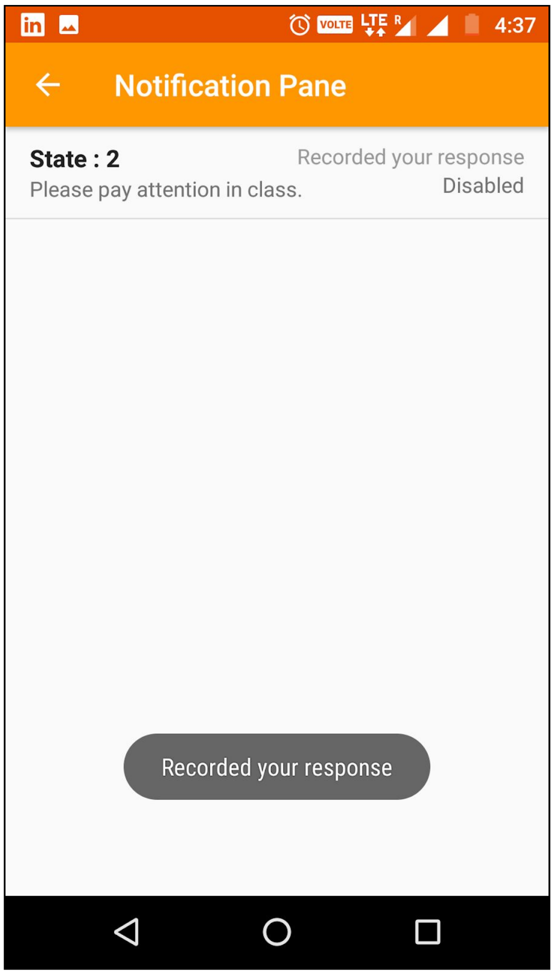
Equivalence Classes:

1. The received notification is tapped within 10 seconds.



Notification Pane

State : 2
Please pay attention in class.
8
Enabled



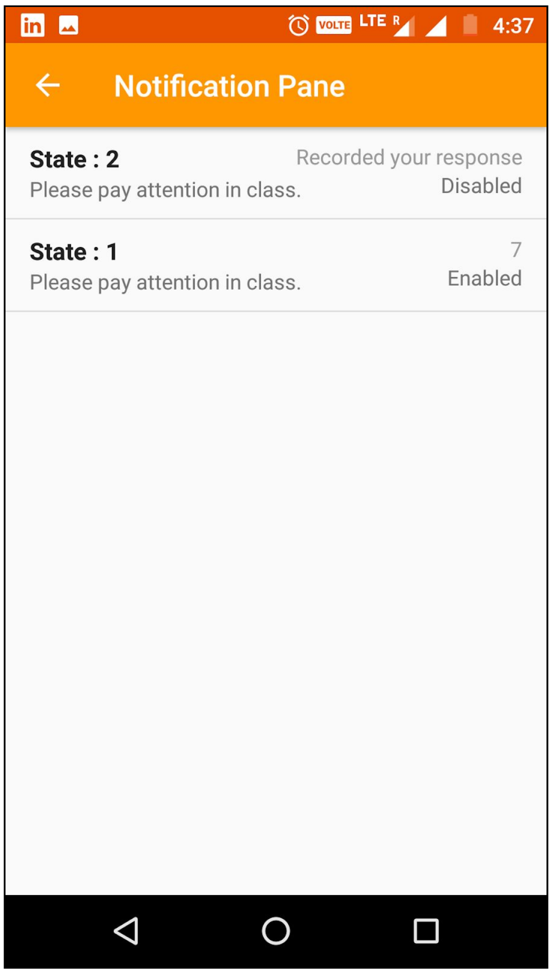
Notification Pane

State : 2
Please pay attention in class.
Recorded your response
Disabled

Recorded your response

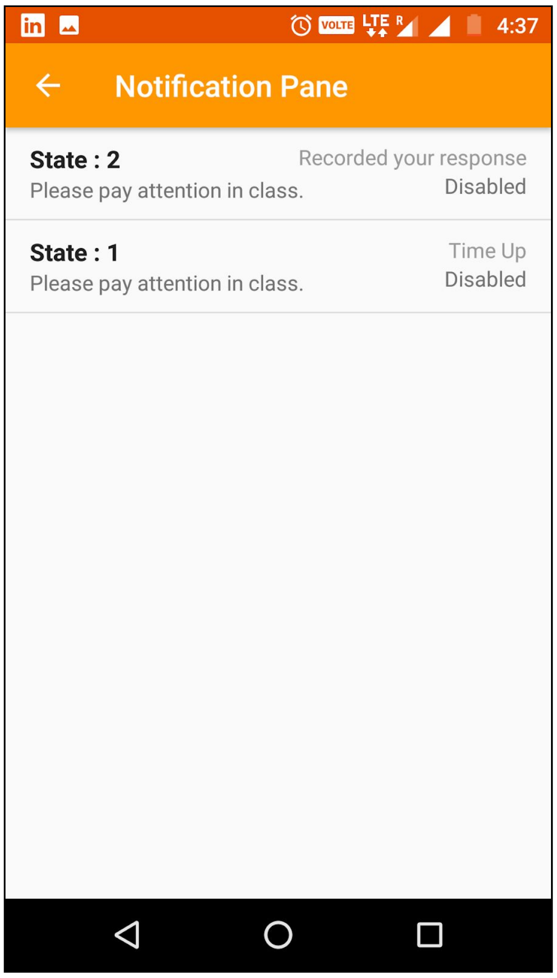
**Input**: Tap on an enabled notification.
**Expected Output**: The notification is disabled, with text that response is recorded as well as a message informing the same.
**Observed Output**: As Expected.

2. The received notification is not tapped within 10 seconds.


Notification Pane

**State : 2** Recorded your response
Please pay attention in class. Disabled

**State : 1** 7
Please pay attention in class. Enabled


Notification Pane

**State : 2** Recorded your response
Please pay attention in class. Disabled

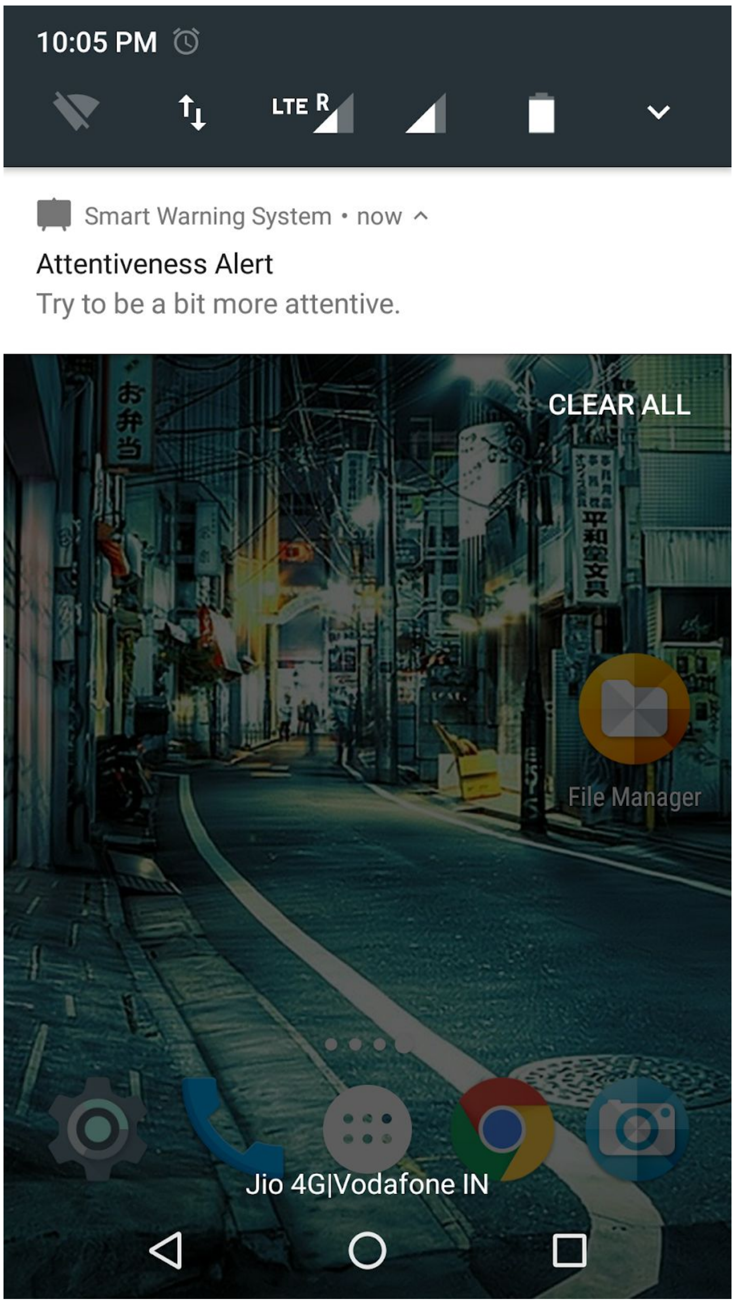
**State : 1** Time Up
Please pay attention in class. Disabled

**Input**: When a new Notification appears, wait for 11 seconds.
**Expected Output**: The notification is disabled with the message that time is over to tap the notification.
**Observed Output**: As Expected.

## 2.9 Notification Module

Equivalence Classes:

1. Generated state is from 1-4


10:05 PM

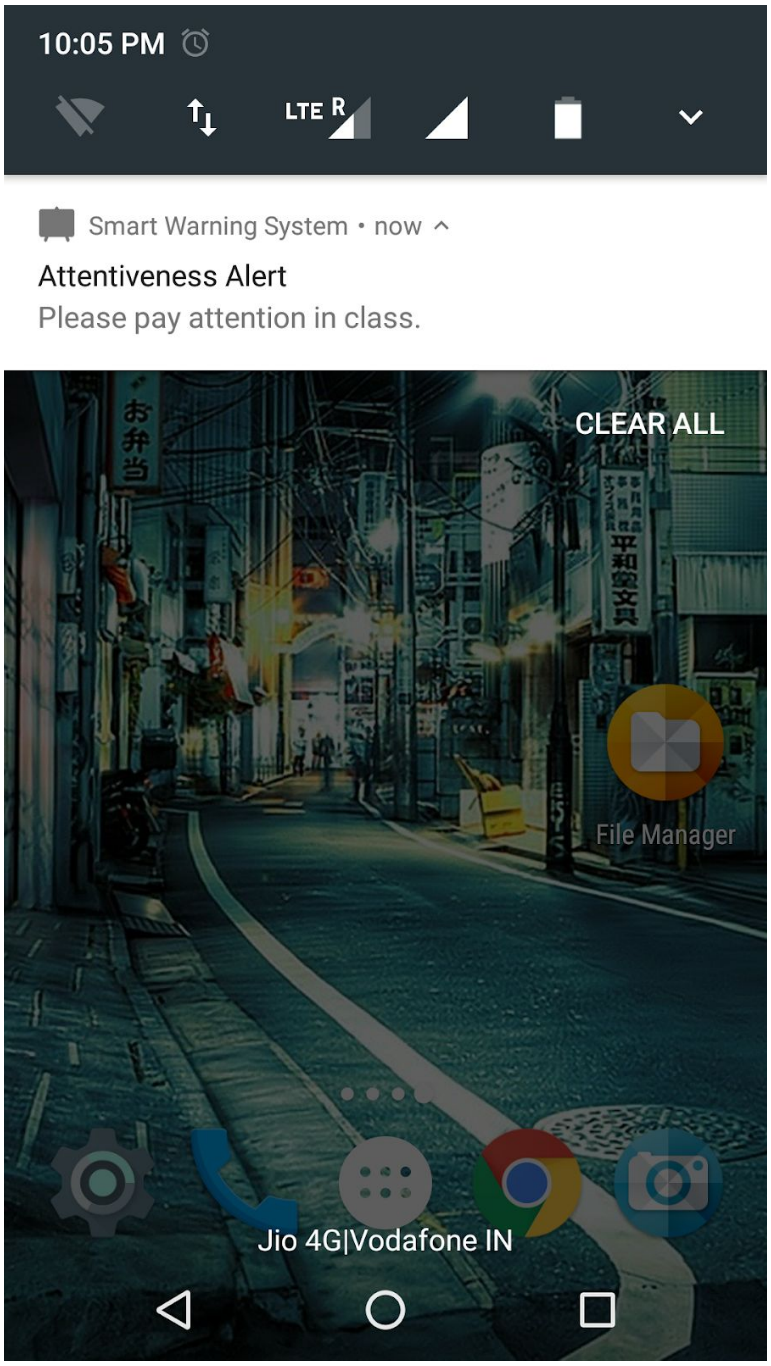
Smart Warning System • now

Attentiveness Alert

Try to be a bit more attentive.

**Input**: Generated state that has the value 3.
**Expected Output**: The notification with the message “Try to be a bit more attentive”
**Observed Output**: As Expected

2. Generated state is from 5-7


10:05 PM

Smart Warning System • now

Attentiveness Alert

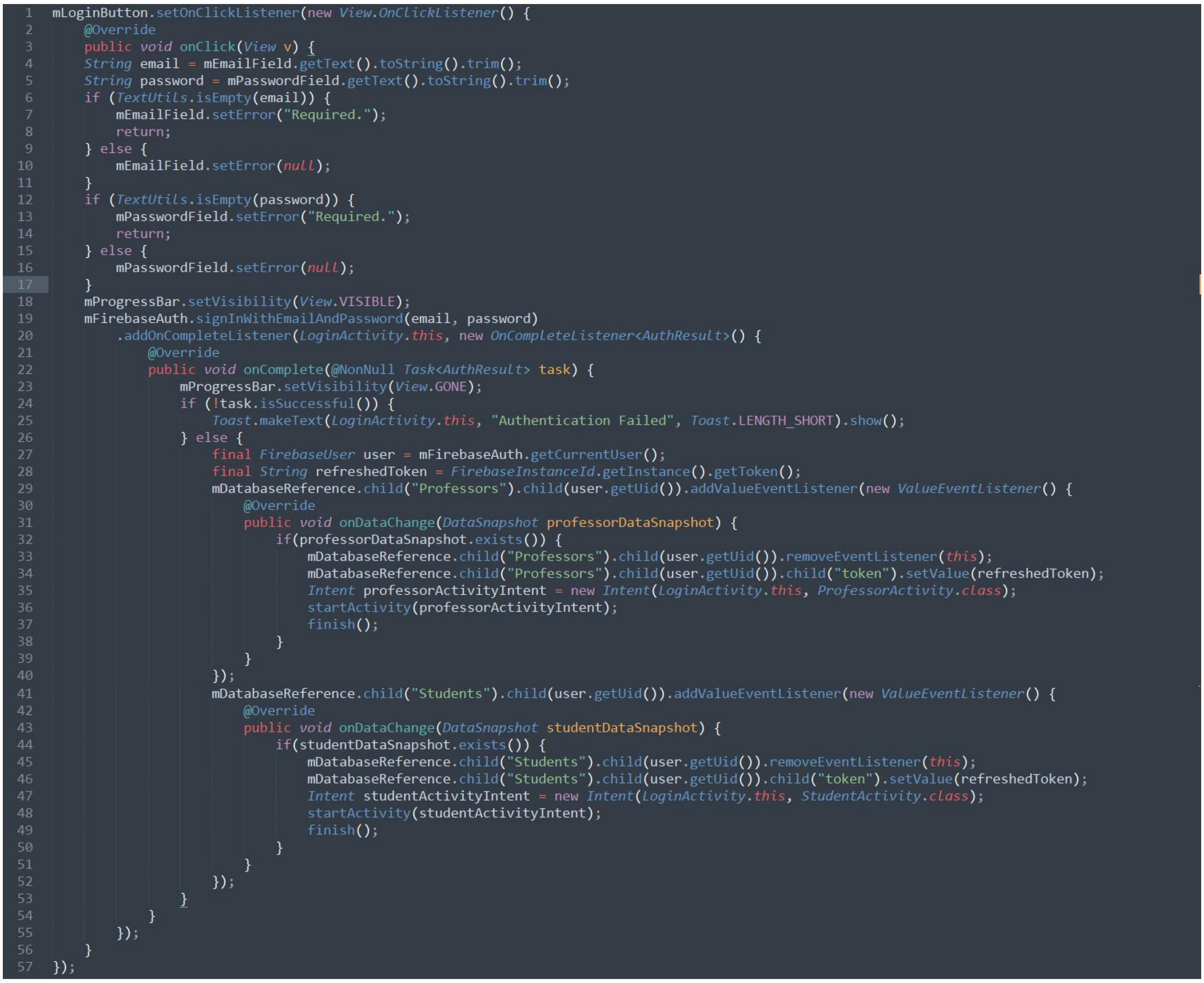
Please pay attention in class.

**Input**: Generated state that has the value 6.
**Expected Output**: The notification with the message “Please pay attention in class”.
**Observed Output**: As Expected

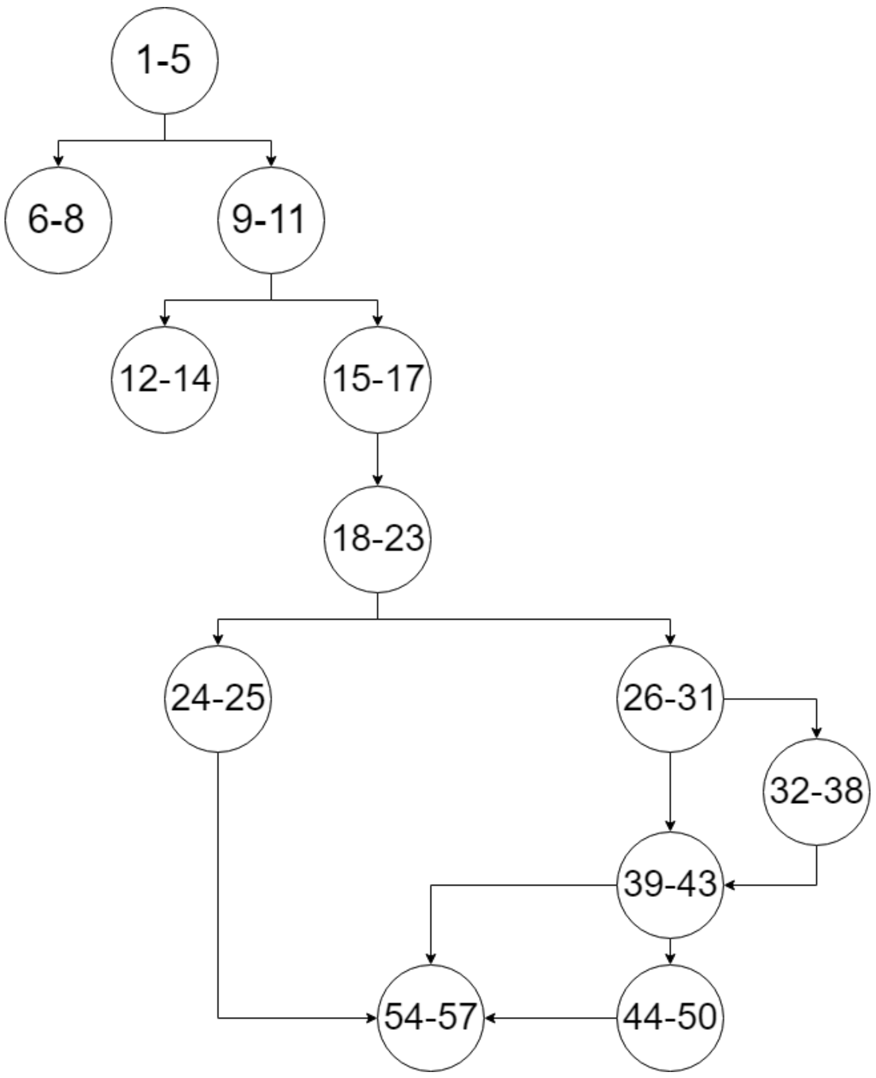
3. Generated state is from 8-10
**Input**: Generated state that has the value 9.
**Expected Output**: The state is recorded into the database and visible to professor in class status.
**Observed Output**: As Expected

# 3. White Box Testing

## 3.1 Login Module



```java
mLoginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    String email = mEmailField.getText().toString().trim();
    String password = mPasswordField.getText().toString().trim();
    if (TextUtils.isEmpty(email)) {
        mEmailField.setError("Required.");
        return;
    } else {
        mEmailField.setError(null);
    }
    if (TextUtils.isEmpty(password)) {
        mPasswordField.setError("Required.");
        return;
    } else {
        mPasswordField.setError(null);
    }
    mProgressBar.setVisibility(View.VISIBLE);
    mFirebaseAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                mProgressBar.setVisibility(View.GONE);
                if (!task.isSuccessful()) {
                    Toast.makeText(LoginActivity.this, "Authentication Failed", Toast.LENGTH_SHORT).show();
                } else {
                    final FirebaseUser user = mFirebaseAuth.getCurrentUser();
                    final String refreshedToken = FirebaseInstanceId.getInstance().getToken();
                    mDatabaseReference.child("Professors").child(user.getUid()).addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot professorDataSnapshot) {
                            if(professorDataSnapshot.exists()) {
                                mDatabaseReference.child("Professors").child(user.getUid()).removeEventListener(this);
                                mDatabaseReference.child("Professors").child(user.getUid()).child("token").setValue(refreshedToken);
                                Intent professorActivityIntent = new Intent(LoginActivity.this, ProfessorActivity.class);
                                startActivity(professorActivityIntent);
                                finish();
                            }
                        }
                    });
                    mDatabaseReference.child("Students").child(user.getUid()).addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot studentDataSnapshot) {
                            if(studentDataSnapshot.exists()) {
                                mDatabaseReference.child("Students").child(user.getUid()).removeEventListener(this);
                                mDatabaseReference.child("Students").child(user.getUid()).child("token").setValue(refreshedToken);
                                Intent studentActivityIntent = new Intent(LoginActivity.this, StudentActivity.class);
                                startActivity(studentActivityIntent);
                                finish();
                            }
                        }
                    });
                }
            }
        });
    }
});
```

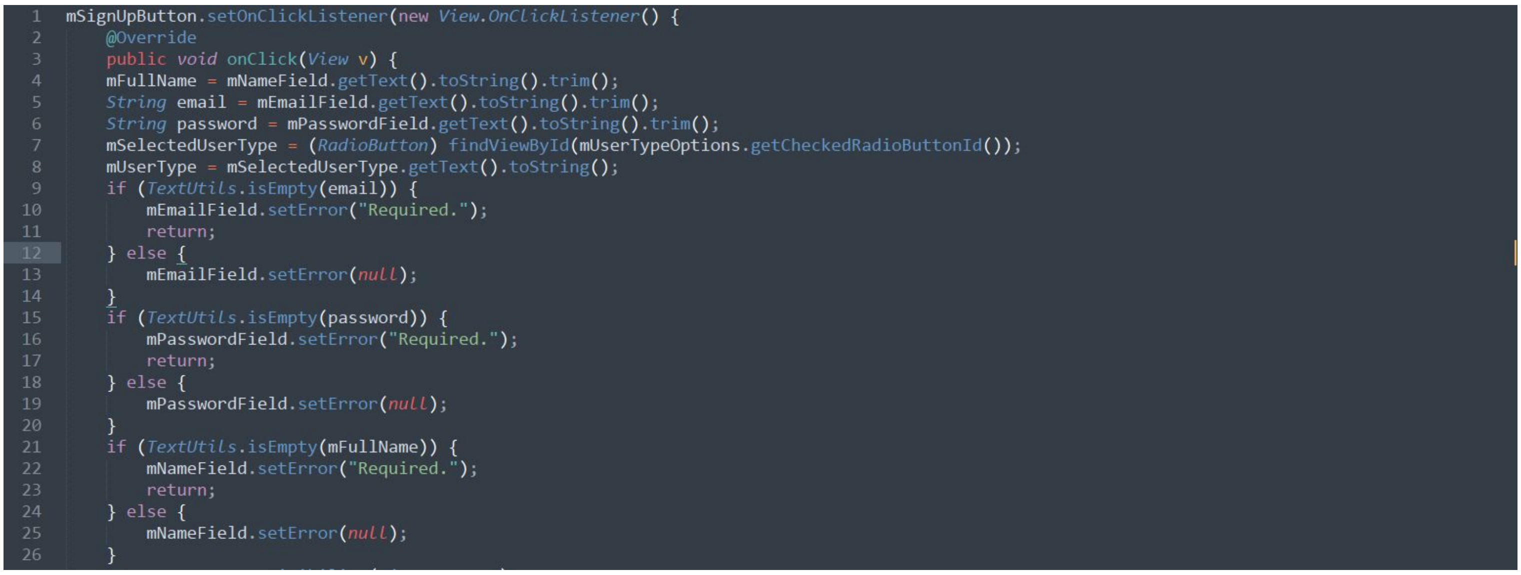


**Path 1 test case** :
**Input**: “jatingoyal412@gmail.com”, “abcdefgh”
Path Followed :
(1-5) -> (9-11) -> (15-17) -> (18-23) -> (26-31) -> (32-38) -> (39-43) -> (54-57)

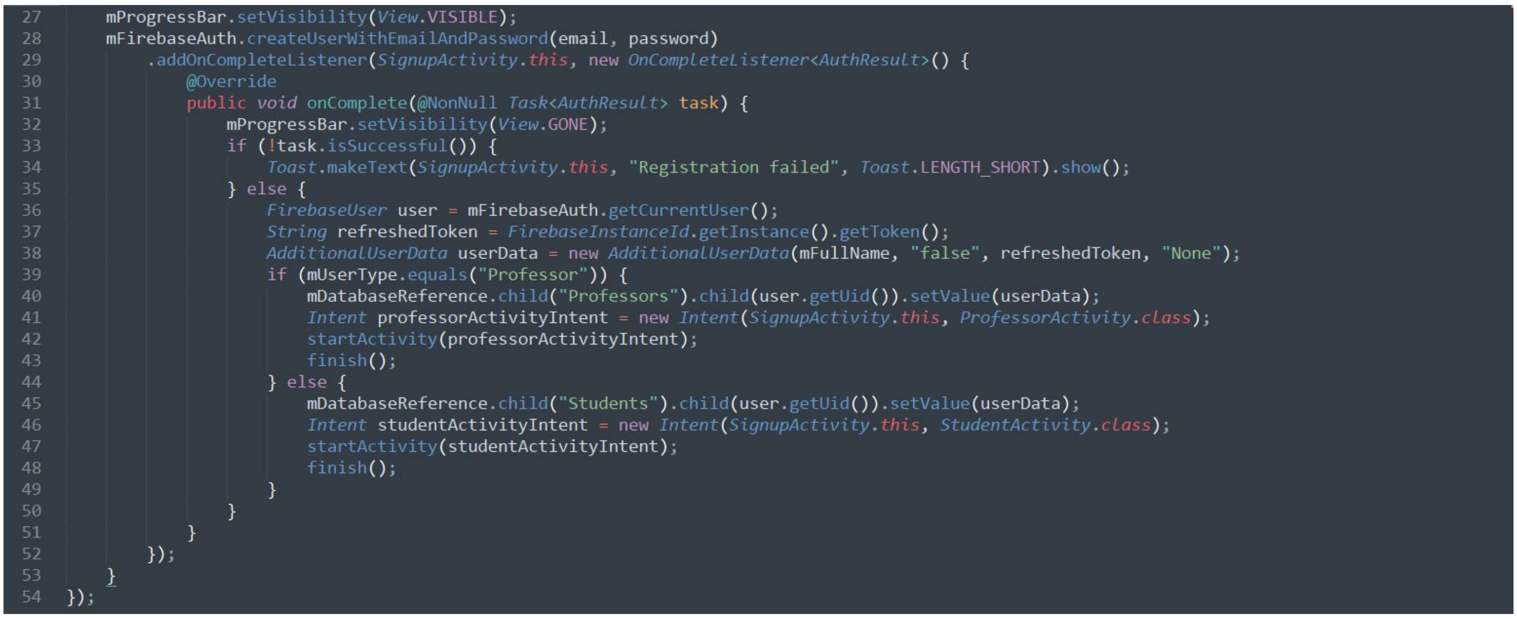
**Path 2 test case** :
**Input**: “jatingoyal412@gmail.com”, “abcxyzsd”
Path Followed :
(1-5) -> (9-11) -> (15-17) -> (24-25) -> (54-57)

**Path 3  test case** :
**Input**:  “{Empty}”, “abcxyzsd”
Path Followed :
(1-5) -> (6-8)

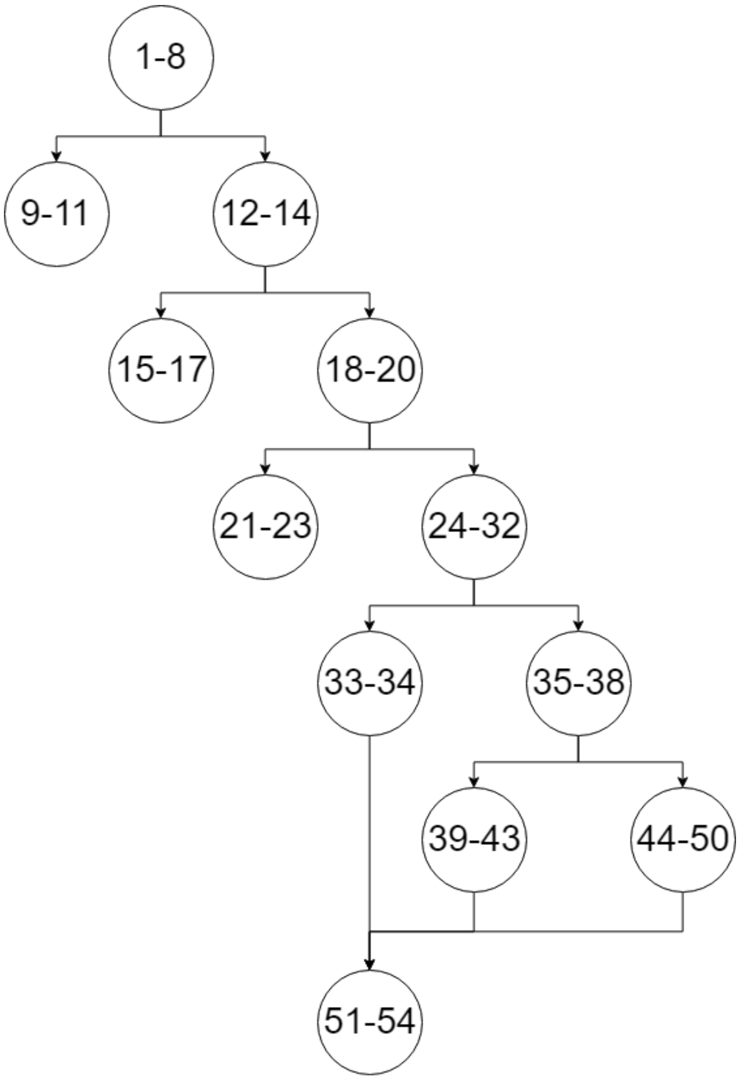
## 3.2 Registration Module



```java
mSignUpButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    mFullName = mNameField.getText().toString().trim();
    String email = mEmailField.getText().toString().trim();
    String password = mPasswordField.getText().toString().trim();
    mSelectedUserType = (RadioButton) findViewById(mUserTypeOptions.getCheckedRadioButtonId());
    mUserType = mSelectedUserType.getText().toString();
    if (TextUtils.isEmpty(email)) {
        mEmailField.setError("Required.");
        return;
    } else {
        mEmailField.setError(null);
    }
    if (TextUtils.isEmpty(password)) {
        mPasswordField.setError("Required.");
        return;
    } else {
        mPasswordField.setError(null);
    }
    if (TextUtils.isEmpty(mFullName)) {
        mNameField.setError("Required.");
        return;
    } else {
        mNameField.setError(null);
    }
```



```java
    mProgressBar.setVisibility(View.VISIBLE);
    mFirebaseAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(SignupActivity.this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                mProgressBar.setVisibility(View.GONE);
                if (!task.isSuccessful()) {
                    Toast.makeText(SignupActivity.this, "Registration failed", Toast.LENGTH_SHORT).show();
                } else {
                    FirebaseUser user = mFirebaseAuth.getCurrentUser();
                    String refreshedToken = FirebaseInstanceId.getInstance().getToken();
                    AdditionalUserData userData = new AdditionalUserData(mFullName, "false", refreshedToken, "None");
                    if (mUserType.equals("Professor")) {
                        mDatabaseReference.child("Professors").child(user.getUid()).setValue(userData);
                        Intent professorActivityIntent = new Intent(SignupActivity.this, ProfessorActivity.class);
                        startActivity(professorActivityIntent);
                        finish();
                    } else {
                        mDatabaseReference.child("Students").child(user.getUid()).setValue(userData);
                        Intent studentActivityIntent = new Intent(SignupActivity.this, StudentActivity.class);
                        startActivity(studentActivityIntent);
                        finish();
                    }
                }
            }
        });
    }
});
```



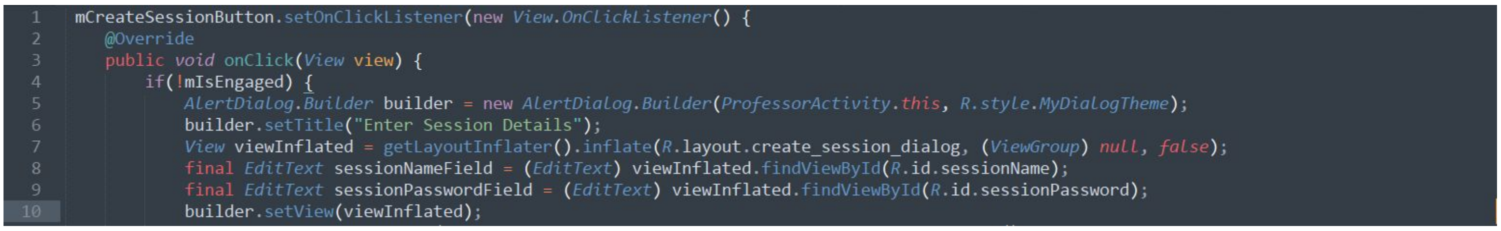
**Path 1 test case** :
**Input**: “Jatin Goyal”, “jatingoyal412@gmail.com”, “abcdefgh”, Professor
Path Followed :
(1-8) -> (12-14) -> (18-20) -> (24-32) -> (35-38) -> (39-43) ->(51-54)

**Path 2 test case** :
**Input**: “Jatin Goyal”, “jatingoyal412”, “abcdefgh”
Path Followed :
(1-8) -> (12-14) -> (18-20) -> (24-32) -> (33-34) -> (51-54)

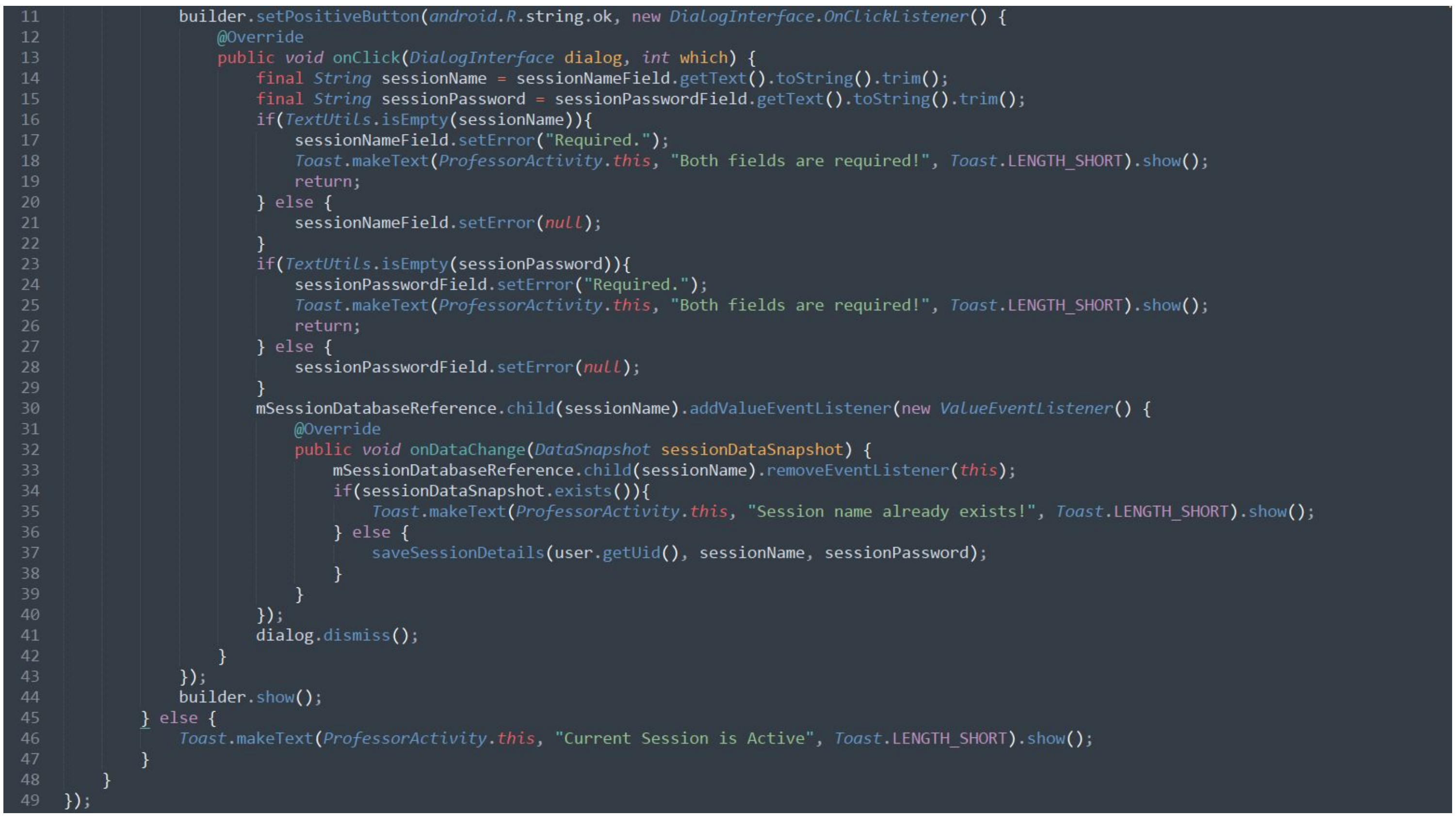
**Path 3 test case** :
**Input**: “Jatin Goyal”, “jatingoyal412@gmail.com”, “abcd”
Path Followed :
(1-8) -> (12-14) -> (18-20) -> (24-32) -> (33-34) -> (51-54)

**Path 4 test case** :
**Input**: “Jatin Goyal”, “jatingoyal412@gmail.com”, “{Empty}”
Path Followed :
(1-8) -> (12-14) -> (15-17)

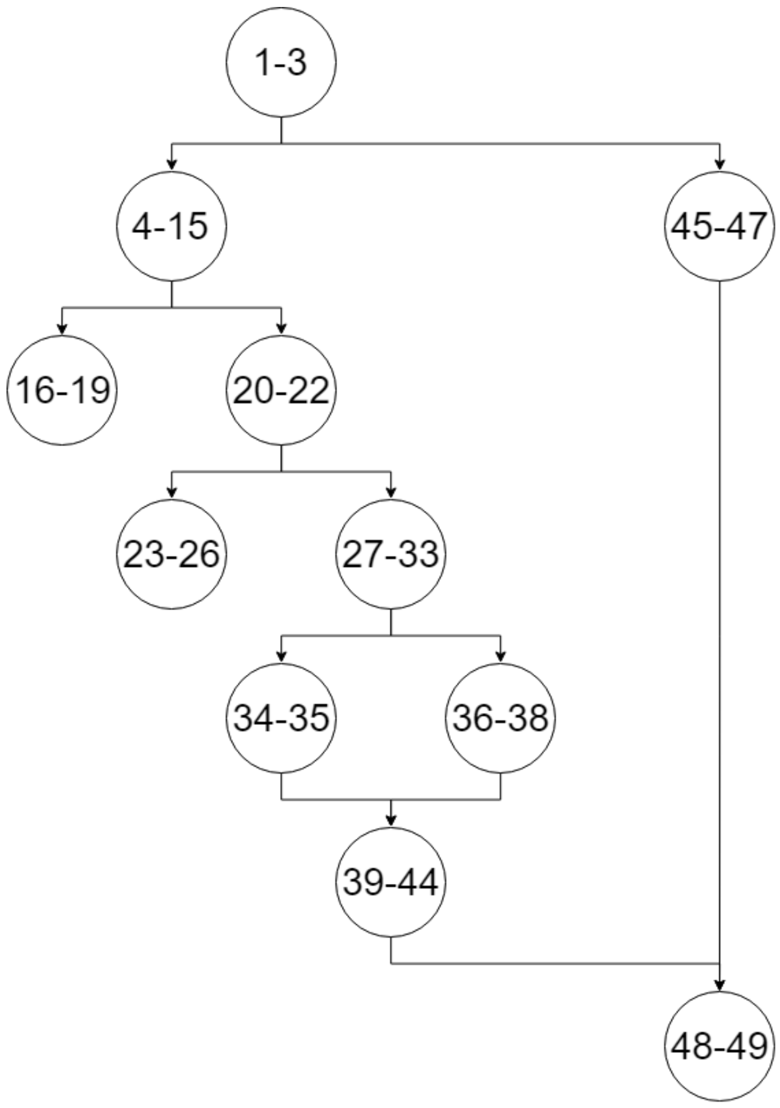
## 3.3 Create Session Module



```
mCreateSessionButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(!mIsEngaged) {
            AlertDialog.Builder builder = new AlertDialog.Builder(ProfessorActivity.this, R.style.MyDialogTheme);
            builder.setTitle("Enter Session Details");
            View viewInflated = getLayoutInflater().inflate(R.layout.create_session_dialog, (ViewGroup) null, false);
            final EditText sessionNameField = (EditText) viewInflated.findViewById(R.id.sessionName);
            final EditText sessionPasswordField = (EditText) viewInflated.findViewById(R.id.sessionPassword);
            builder.setView(viewInflated);
```



```
            builder.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    final String sessionName = sessionNameField.getText().toString().trim();
                    final String sessionPassword = sessionPasswordField.getText().toString().trim();
                    if(TextUtils.isEmpty(sessionName)){
                        sessionNameField.setError("Required.");
                        Toast.makeText(ProfessorActivity.this, "Both fields are required!", Toast.LENGTH_SHORT).show();
                        return;
                    } else {
                        sessionNameField.setError(null);
                    }
                    if(TextUtils.isEmpty(sessionPassword)){
                        sessionPasswordField.setError("Required.");
                        Toast.makeText(ProfessorActivity.this, "Both fields are required!", Toast.LENGTH_SHORT).show();
                        return;
                    } else {
                        sessionPasswordField.setError(null);
                    }
                    mSessionDatabaseReference.child(sessionName).addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot sessionDataSnapshot) {
                            mSessionDatabaseReference.child(sessionName).removeEventListener(this);
                            if(sessionDataSnapshot.exists()){
                                Toast.makeText(ProfessorActivity.this, "Session name already exists!", Toast.LENGTH_SHORT).show();
                            } else {
                                saveSessionDetails(user.getUid(), sessionName, sessionPassword);
                            }
                        }
                    });
                    dialog.dismiss();
                }
            });
            builder.show();
        } else {
            Toast.makeText(ProfessorActivity.this, "Current Session is Active", Toast.LENGTH_SHORT).show();
        }
    }
});
```



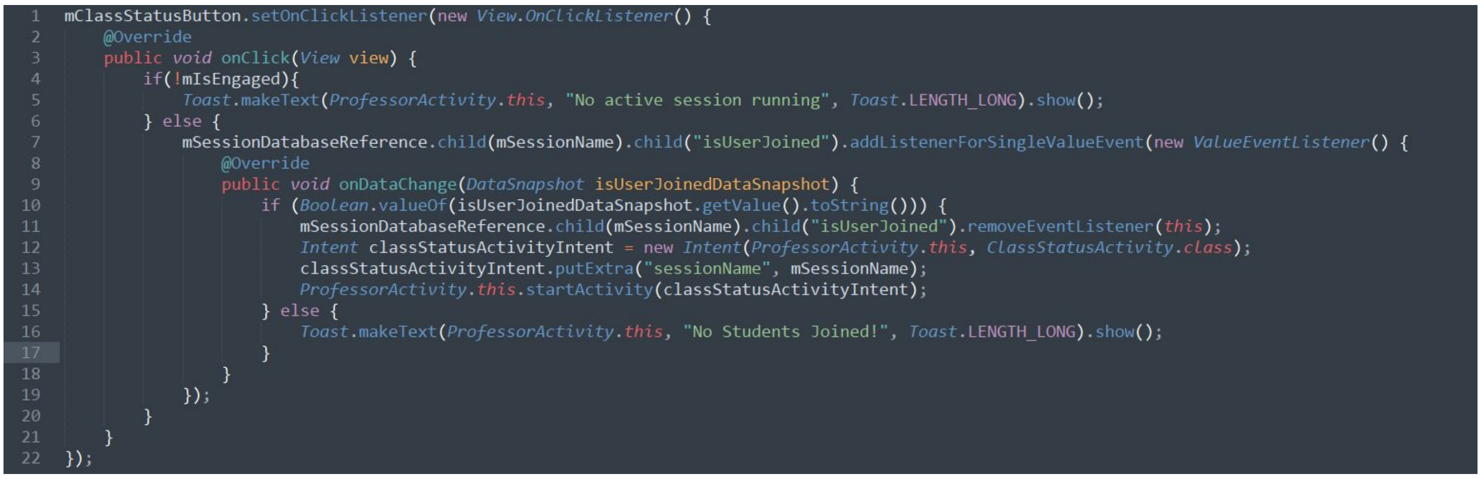
**Path 1 test case:**
**Input**: “Ses101”, “abcd”
Path Followed:
(1-3) -> (4-15) -> (20-22) -> (27-33) -> (36-38) -> (39-44) -> (48-49)

**Path 2 test case:**
**Input**: “Ses101”, “abcd”
sessionDataSnapshot is not null
Path Followed:
(1-3) -> (4-15) -> (20-22) -> (27-33) -> (34-35) -> (39-44)

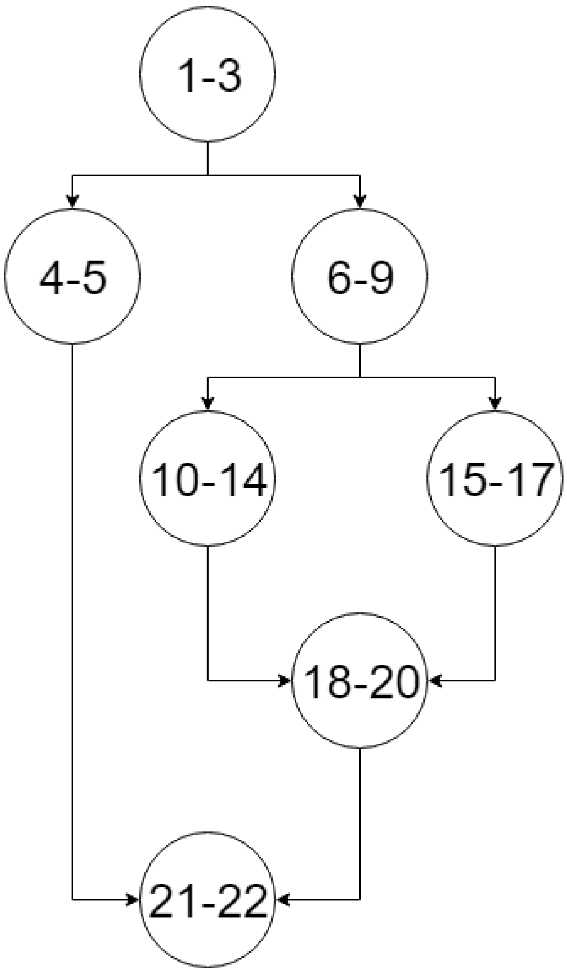
**Path 3 test case:**
**Input**: “{Empty}”, “abcd”
Path Followed:
(1-3) -> (4-15) -> (16-19)

**Path 4 test case:**
**Input**: “Ses101”, “abcd”
mIsEngaged = true
Path Followed:
(1-3) -> (45-47) -> (48-49)

## 3.4 Class Status Module



```
1  mClassStatusButton.setOnClickListener(new View.OnClickListener() {
2      @Override
3      public void onClick(View view) {
4          if(!mIsEngaged){
5              Toast.makeText(ProfessorActivity.this, "No active session running", Toast.LENGTH_LONG).show();
6          } else {
7              mSessionDatabaseReference.child(mSessionName).child("isUserJoined").addListenerForSingleValueEvent(new ValueEventListener() {
8                  @Override
9                  public void onDataChange(DataSnapshot isUserJoinedDataSnapshot) {
10                     if (Boolean.valueOf(isUserJoinedDataSnapshot.getValue().toString())) {
11                         mSessionDatabaseReference.child(mSessionName).child("isUserJoined").removeEventListener(this);
12                         Intent classStatusActivityIntent = new Intent(ProfessorActivity.this, ClassStatusActivity.class);
13                         classStatusActivityIntent.putExtra("sessionName", mSessionName);
14                         ProfessorActivity.this.startActivity(classStatusActivityIntent);
15                     } else {
16                         Toast.makeText(ProfessorActivity.this, "No Students Joined!", Toast.LENGTH_LONG).show();
17                     }
18                 }
19             });
20         }
21     }
22 });
```

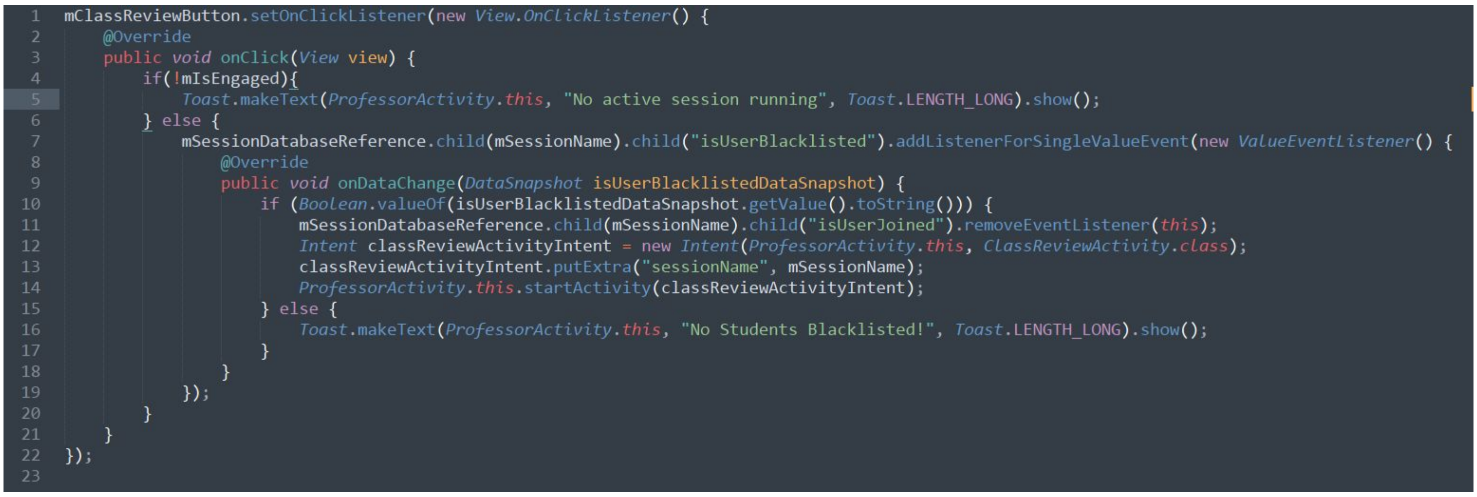


**Path 1 test case:**
mIsEngaged = true
Path Followed:
(1-3) -> (4-5) -> (21-22)

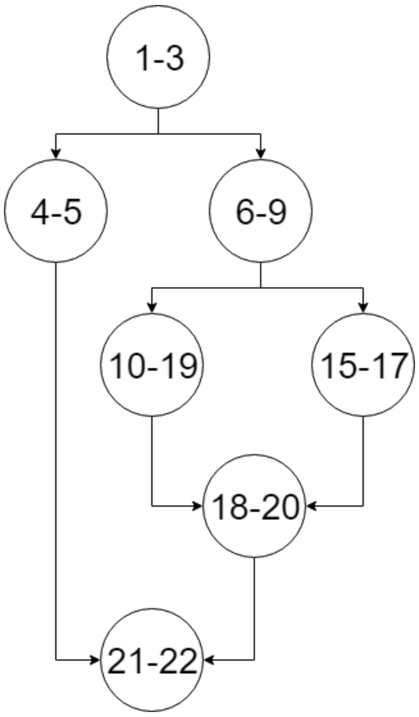
**Path 2 test case:**
mIsEngaged = false, isUserJoined = false
Path Followed:
(1-3) -> (6-9) -> (15-17) -> (18-20) -> (21-22)

**Path 3 test case:**
mIsEngaged = false, isUserJoined = true
Path Followed:
(1-3) -> (6-9) -> (10-14) -> (18-20) -> (21-22)

## 3.5 Class Review Module



```
 1  mClassReviewButton.setOnClickListener(new View.OnClickListener() {
 2      @Override
 3      public void onClick(View view) {
 4          if(!mIsEngaged){
 5              Toast.makeText(ProfessorActivity.this, "No active session running", Toast.LENGTH_LONG).show();
 6          } else {
 7              mSessionDatabaseReference.child(mSessionName).child("isUserBlacklisted").addListenerForSingleValueEvent(new ValueEventListener() {
 8                  @Override
 9                  public void onDataChange(DataSnapshot isUserBlacklistedDataSnapshot) {
10                      if (Boolean.valueOf(isUserBlacklistedDataSnapshot.getValue().toString())) {
11                          mSessionDatabaseReference.child(mSessionName).child("isUserJoined").removeEventListener(this);
12                          Intent classReviewActivityIntent = new Intent(ProfessorActivity.this, ClassReviewActivity.class);
13                          classReviewActivityIntent.putExtra("sessionName", mSessionName);
14                          ProfessorActivity.this.startActivity(classReviewActivityIntent);
15                      } else {
16                          Toast.makeText(ProfessorActivity.this, "No Students Blacklisted!", Toast.LENGTH_LONG).show();
17                      }
18                  }
19              });
20          }
21      }
22  });
23
```

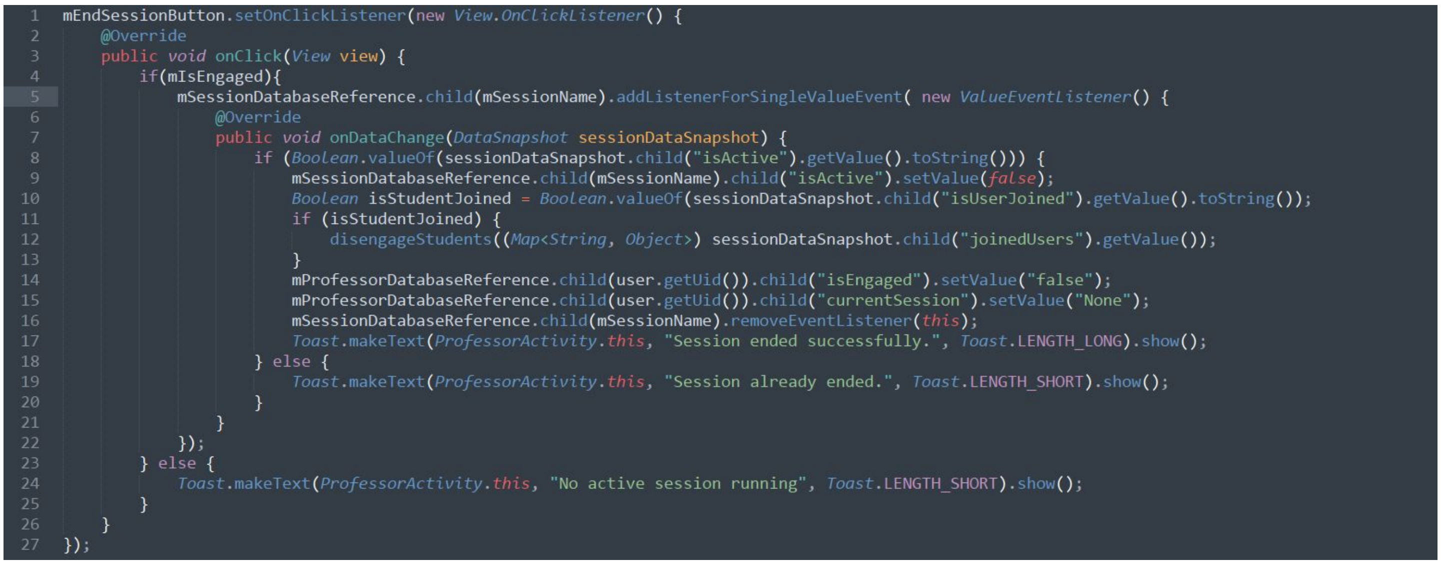


**Path 1 test case:**
mIsEngaged = true
Path Followed:
(1-3) -> (4-5) -> (21-22)

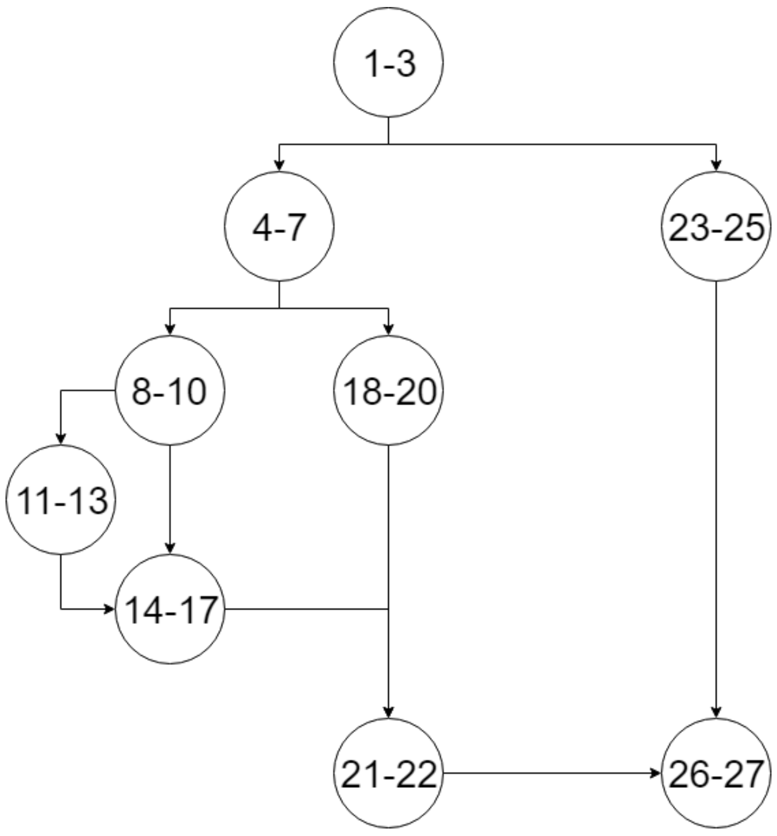
**Path 2 test case:**
mIsEngaged = false, UserJoined = false
Path Followed:
(1-3) -> (6-9) -> (15-17) -> (18-20) -> (21-22)

**Path 3 test case:**
mIsEngaged = false, UserJoined = true
Path Followed:
(1-3) -> (6-9) -> (10-14) -> (18-20) -> (21-22)

## 3.6 End Session Module



```
1   mEndSessionButton.setOnClickListener(new View.OnClickListener() {
2       @Override
3       public void onClick(View view) {
4           if(mIsEngaged){
5               mSessionDatabaseReference.child(mSessionName).addListenerForSingleValueEvent( new ValueEventListener() {
6                   @Override
7                   public void onDataChange(DataSnapshot sessionDataSnapshot) {
8                       if (Boolean.valueOf(sessionDataSnapshot.child("isActive").getValue().toString())) {
9                           mSessionDatabaseReference.child(mSessionName).child("isActive").setValue(false);
10                          Boolean isStudentJoined = Boolean.valueOf(sessionDataSnapshot.child("isUserJoined").getValue().toString());
11                          if (isStudentJoined) {
12                              disengageStudents((Map<String, Object>) sessionDataSnapshot.child("joinedUsers").getValue());
13                          }
14                          mProfessorDatabaseReference.child(user.getUid()).child("isEngaged").setValue("false");
15                          mProfessorDatabaseReference.child(user.getUid()).child("currentSession").setValue("None");
16                          mSessionDatabaseReference.child(mSessionName).removeEventListener(this);
17                          Toast.makeText(ProfessorActivity.this, "Session ended successfully.", Toast.LENGTH_LONG).show();
18                      } else {
19                          Toast.makeText(ProfessorActivity.this, "Session already ended.", Toast.LENGTH_SHORT).show();
20                      }
21                  }
22              });
23          } else {
24              Toast.makeText(ProfessorActivity.this, "No active session running", Toast.LENGTH_SHORT).show();
25          }
26      }
27  });
```



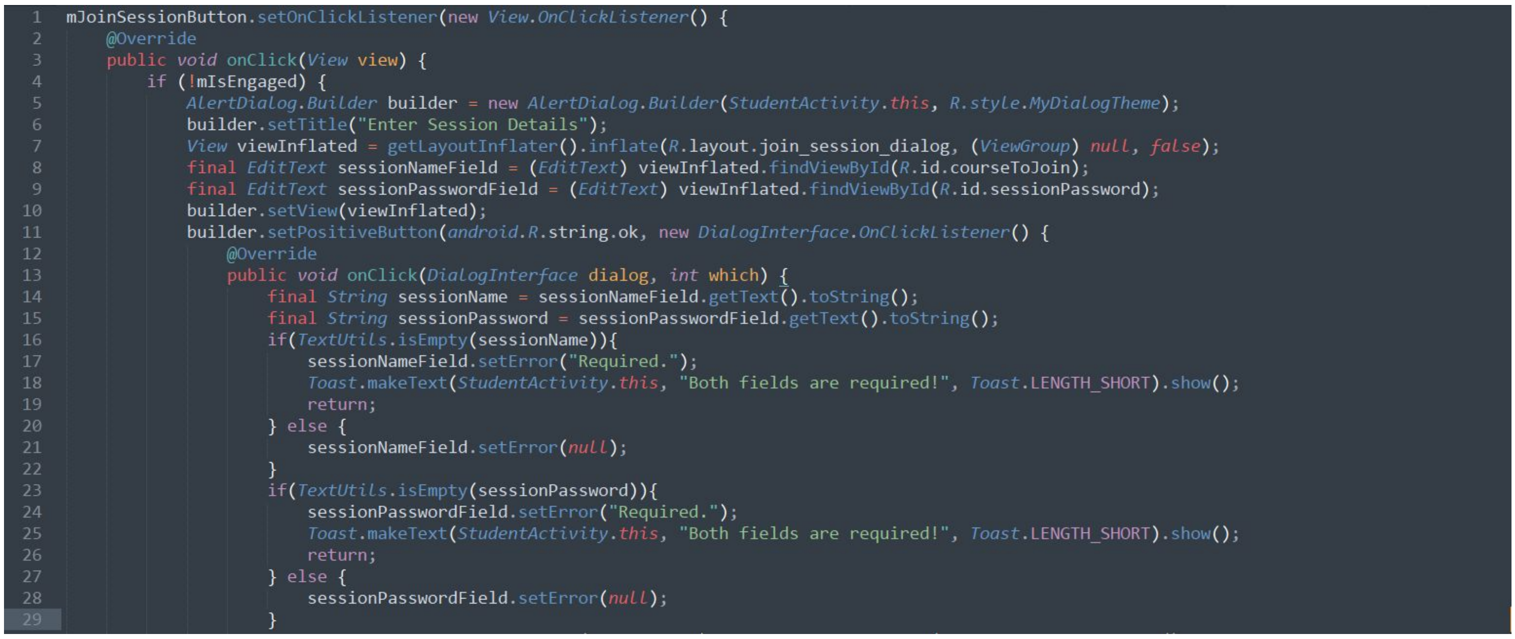
**Path 1 test case:**
**Input** : mIsEngaged = false
Path Followed:
(1-3) -> (23-25) -> (26-27)

**Path 2 test case:**
**Input :** isStudentJoined = false
mIsEngaged = true
Path Followed:
(1-3) -> (4-7) -> (8-10) -> (14-17) -> (21 - 22) -> (26-27)

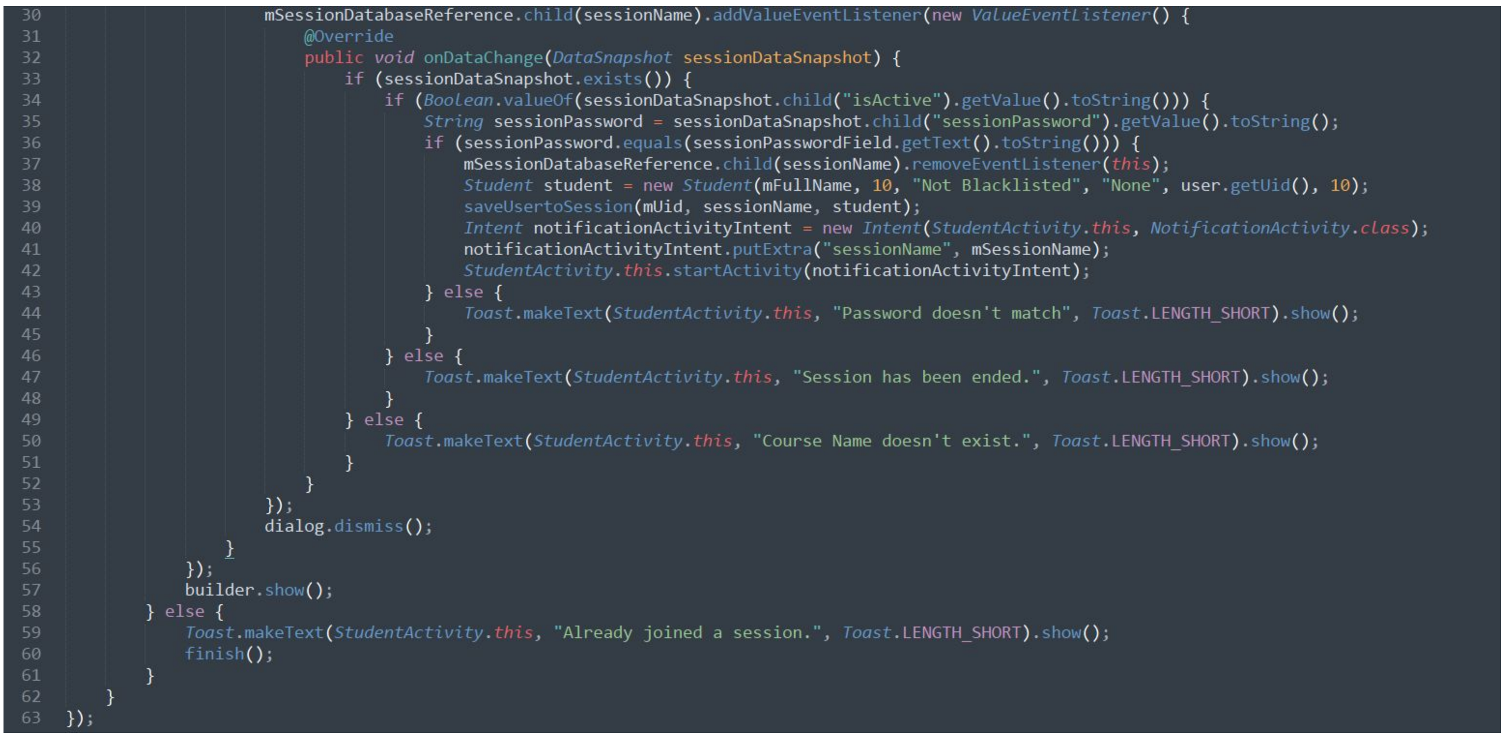
**Path 3 test case:**
**Input :** isStudentJoined = true
mIsEngaged = true
Path Followed:
(1-3) -> (4-7) -> (8-10) -> (11-13) -> (14-17) -> (21-22) -> (26-27)

**Path 4 test case:**
**Input :** isActive flag of current session = false
mIsEngaged = true
Path Followed:
(1-3) -> (4-7) -> (18-20) -> (21-22) -> (26-27)

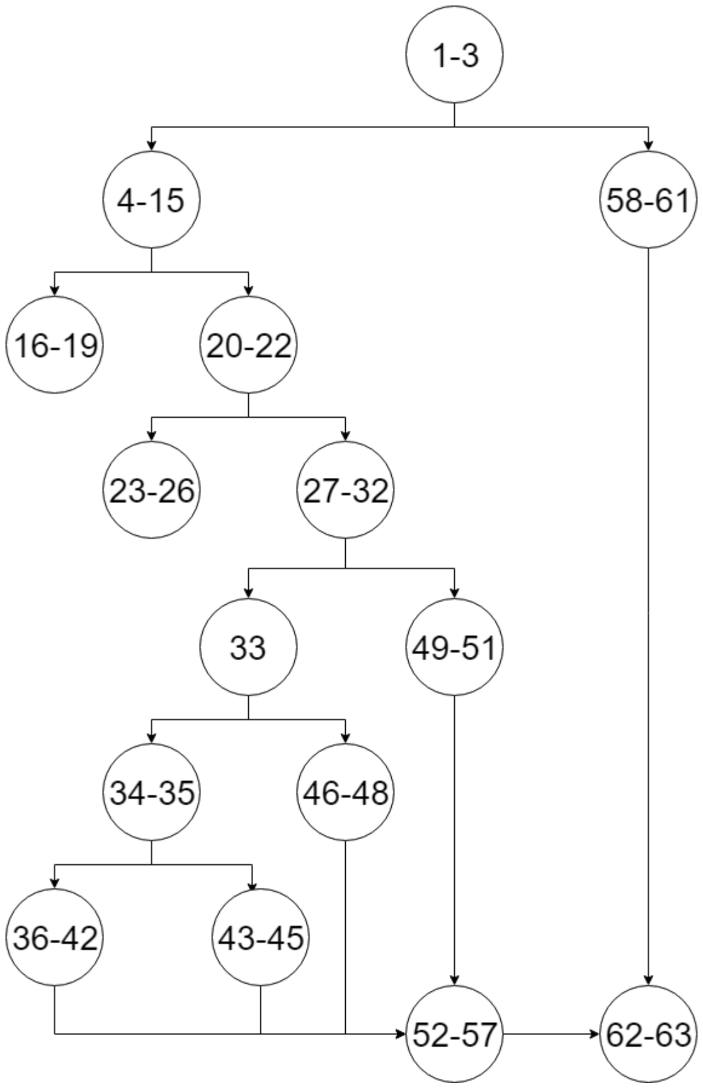
## 3.7 Join Session Module



```java
mJoinSessionButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (!mIsEngaged) {
            AlertDialog.Builder builder = new AlertDialog.Builder(StudentActivity.this, R.style.MyDialogTheme);
            builder.setTitle("Enter Session Details");
            View viewInflated = getLayoutInflater().inflate(R.layout.join_session_dialog, (ViewGroup) null, false);
            final EditText sessionNameField = (EditText) viewInflated.findViewById(R.id.courseToJoin);
            final EditText sessionPasswordField = (EditText) viewInflated.findViewById(R.id.sessionPassword);
            builder.setView(viewInflated);
            builder.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    final String sessionName = sessionNameField.getText().toString();
                    final String sessionPassword = sessionPasswordField.getText().toString();
                    if(TextUtils.isEmpty(sessionName)){
                        sessionNameField.setError("Required.");
                        Toast.makeText(StudentActivity.this, "Both fields are required!", Toast.LENGTH_SHORT).show();
                        return;
                    } else {
                        sessionNameField.setError(null);
                    }
                    if(TextUtils.isEmpty(sessionPassword)){
                        sessionPasswordField.setError("Required.");
                        Toast.makeText(StudentActivity.this, "Both fields are required!", Toast.LENGTH_SHORT).show();
                        return;
                    } else {
                        sessionPasswordField.setError(null);
                    }
```



```java
                    mSessionDatabaseReference.child(sessionName).addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot sessionDataSnapshot) {
                            if (sessionDataSnapshot.exists()) {
                                if (Boolean.valueOf(sessionDataSnapshot.child("isActive").getValue().toString())) {
                                    String sessionPassword = sessionDataSnapshot.child("sessionPassword").getValue().toString();
                                    if (sessionPassword.equals(sessionPasswordField.getText().toString())) {
                                        mSessionDatabaseReference.child(sessionName).removeEventListener(this);
                                        Student student = new Student(mFullName, 10, "Not Blacklisted", "None", user.getUid(), 10);
                                        saveUsertoSession(mUid, sessionName, student);
                                        Intent notificationActivityIntent = new Intent(StudentActivity.this, NotificationActivity.class);
                                        notificationActivityIntent.putExtra("sessionName", mSessionName);
                                        StudentActivity.this.startActivity(notificationActivityIntent);
                                    } else {
                                        Toast.makeText(StudentActivity.this, "Password doesn't match", Toast.LENGTH_SHORT).show();
                                    }
                                } else {
                                    Toast.makeText(StudentActivity.this, "Session has been ended.", Toast.LENGTH_SHORT).show();
                                }
                            } else {
                                Toast.makeText(StudentActivity.this, "Course Name doesn't exist.", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                    dialog.dismiss();
                }
            });
            builder.show();
        } else {
            Toast.makeText(StudentActivity.this, "Already joined a session.", Toast.LENGTH_SHORT).show();
            finish();
        }
    }
});
```



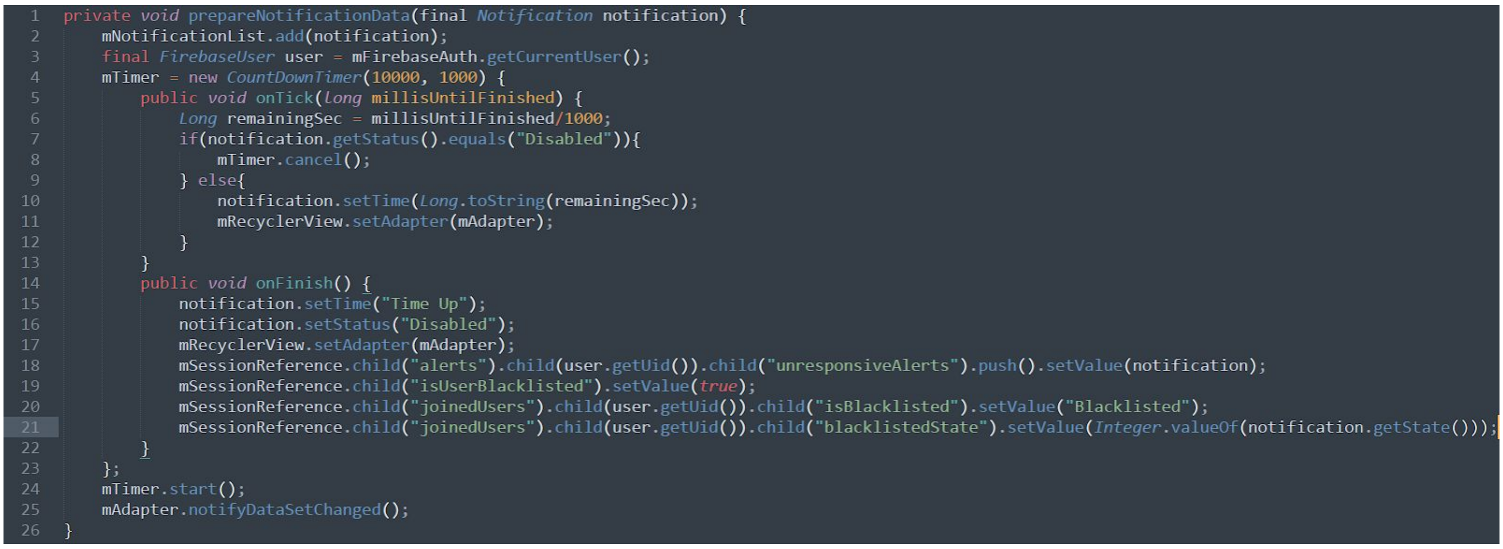
**Path 1 test case**:
**Input**: Input “Ses105”, “abcd”
Path Followed:
(1-3) -> (4-15) -> (20-22) -> (27-32) -> (33) -> (34-35) -> (36-42) -> (52-57) -> (62-63)

**Path 2 test case:**
**Input**: Input “Ses104”, “abcd”
sessionDataSnapshot.exists() = false
Path Followed:
(1-3) -> (4-15) -> (20-22) -> (27-32) -> (49-51) -> (52-57) -> (62-63)

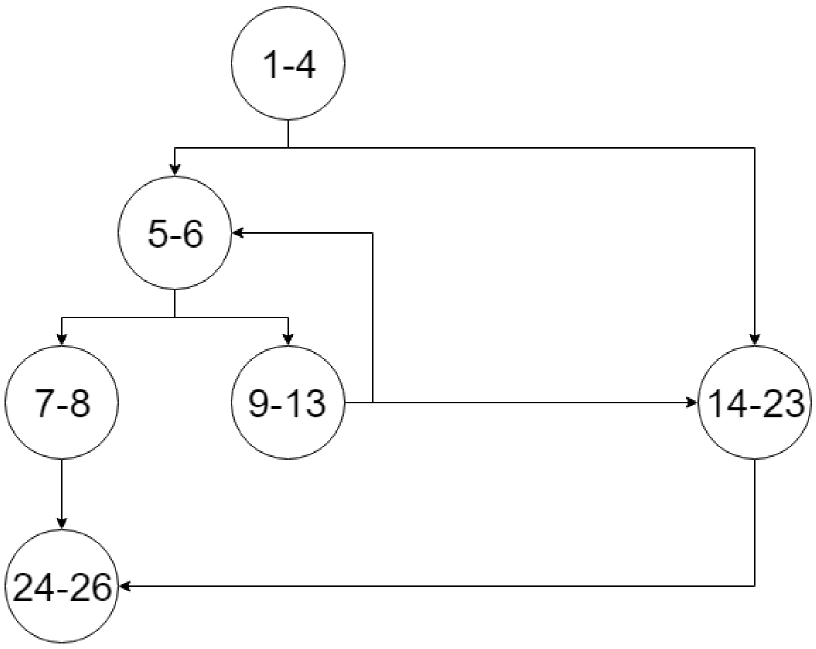
**Path 3 test case:**
**Input:** TextUtils.isEmpty(sessionName) = true
Path Followed:
(1-3) -> (4-15) -> (16-19)

**Path 4 test case:**
**Input:** mIsEngaged = true
Path Followed:
(1-3) -> (58-61) -> (62-63)

## 3.8 Notification Module



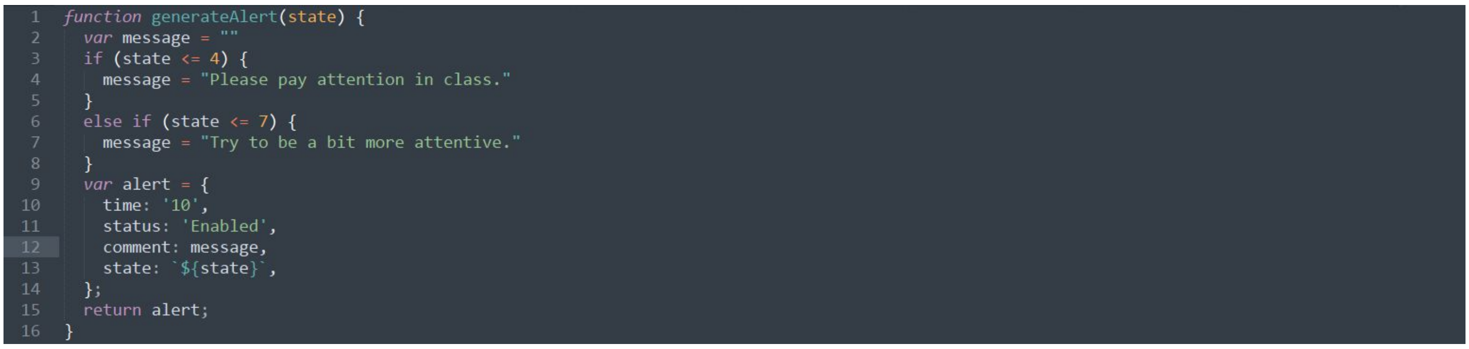
```
1  private void prepareNotificationData(final Notification notification) {
2      mNotificationList.add(notification);
3      final FirebaseUser user = mFirebaseAuth.getCurrentUser();
4      mTimer = new CountDownTimer(10000, 1000) {
5          public void onTick(long millisUntilFinished) {
6              Long remainingSec = millisUntilFinished/1000;
7              if(notification.getStatus().equals("Disabled")){
8                  mTimer.cancel();
9              } else{
10                 notification.setTime(Long.toString(remainingSec));
11                 mRecyclerView.setAdapter(mAdapter);
12             }
13         }
14         public void onFinish() {
15             notification.setTime("Time Up");
16             notification.setStatus("Disabled");
17             mRecyclerView.setAdapter(mAdapter);
18             mSessionReference.child("alerts").child(user.getUid()).child("unresponsiveAlerts").push().setValue(notification);
19             mSessionReference.child("isUserBlacklisted").setValue(true);
20             mSessionReference.child("joinedUsers").child(user.getUid()).child("isBlacklisted").setValue("Blacklisted");
21             mSessionReference.child("joinedUsers").child(user.getUid()).child("blacklistedState").setValue(Integer.valueOf(notification.getState()));
22         }
23     };
24     mTimer.start();
25     mAdapter.notifyDataSetChanged();
26 }
```



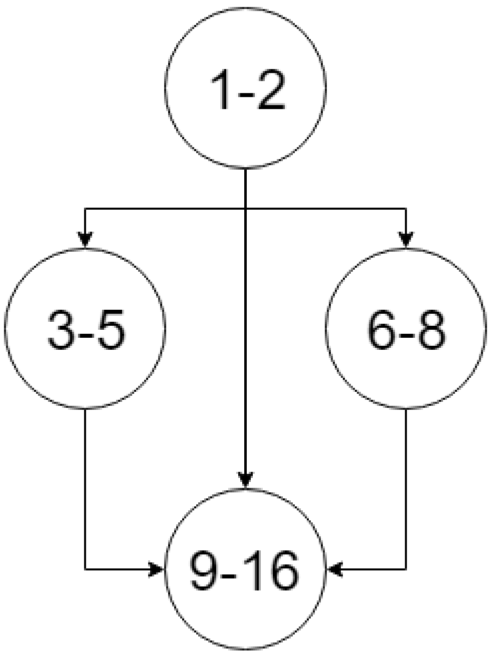
(Part of the path enclosed in square brackets run in loop some finite times)
**Path 1 test case:**
**Input** : Student does tap the notification at 7 seconds, thus
notification.getStatus() = Enabled for first 3 iterations then becomes false..
**Path Followed**:
(1-4) -> [(5-6) -> (9-13)] -> (5-6) -> (7-8) -> (24-26) *loop runs 3 times

**Path 3 test case:**
**Input**: Student does not tap the notification in initial 10 seconds,
notification.getStatus() = Enabled in each iteration
**Path Followed**:
(1-4) -> [(5-6) -> (9-13)] -> (14-23) -> (24-26) *loop runs 10 times.

## 3.9 Send Alert Module



```
1  function generateAlert(state) {
2    var message = ""
3    if (state <= 4) {
4      message = "Please pay attention in class."
5    }
6    else if (state <= 7) {
7      message = "Try to be a bit more attentive."
8    }
9    var alert = {
10     time: '10',
11     status: 'Enabled',
12     comment: message,
13     state: `${state}`,
14   };
15   return alert;
16 }
```



**Path 1 test case:**
**Input**: State = 9;
**Path Followed**:
(1-2) -> (9-16)

**Path 2 test case:**
**Input**: State = 1;
**Path Followed**:
(1-2) -> (3-5) -> (9-16)

**Path 3 test case:**
**Input**: State = 6;
**Path Followed**:
(1-2) -> (6-8) -> (9-16)