ANDROID

Done By: Kishan Lutchman (812003797) & Javier Rojas (812003809)

Course: COMP 3275

Project: Student ID Roll Image Capture

Title: Developer Documentation

GitHub Link: https://github.com/Javier-Rojas/Android-project

Explanation of Code

Class Name: ScanAndSubmit.java

Description: Initiating a Scan

```
if(v.getId()==R.id.btn_scan){
    if(!packageManager.hasSystemFeature(PackageManager.FEATURE_CAMERA)){
        Toast.makeText(getApplicationContext(),"Your device has no camera!",Toast.LENGTH_LONG).show();
    }else {
        IntentIntegrator intIntegrator = new IntentIntegrator(this);
        intIntegrator.initiateScan();
    }
}
```

In the above code we are check to see if the user's mobile phone has a camera equipped so that they can scan the barcode. If the check returns false then the user is presented with a toast that tells them that their phone does not have a camera. If it does have a camera then an "IntentIntegrator" object is created and via this a scan is initiated.

Class Name: ScanAndSubmit.java

Description: Upon receiving a result from the Scan

```
public void onActivityResult(int requestCode, int resultCode, Intent intent) {//When barcode scanner receives data

    if(resultCode == RESULT_OK){
    IntentResult intResult=IntentIntegrator.parseActivityResult(requestCode,resultCode,intent);
    if(intResult!=null){
```

When the scan is finished, we must first check to see if the user interrupted the scanner during the scanning process (resultCode == RESULT_OK) if this is not the case, the user is presented with a toast stating that the process was interrupted by the user. If the scanning process completed successfully we must then check if anything was returned before we can begin processing it. That is why we check to see if the result is not equal to null (intResult! =null). If this statement returns true then we can begin analyzing the result and begin to process and upload the students ID to the database. If the statement is false, then the user is presented with a toast informing them that no data was received.

Class Name: ChangeUserPass.java

Description: Changes the default username and password

```
public void changeUserPass(View v) {
    EditText et = (EditText) findViewById(R.id.txt_user);//get the values for username,password
    EditText pt = (EditText) findViewById(R.id.txt_pass);
    String user = et.getText().toString();//convert the values for username,password to string and assign
    String pass = pt.getText().toString();

    if ((user.length() > 1) && (pass.length() > 1)) {
        Main.setUserPass(user, pass);
        Snackbar.make(v, "Username and Password successfully Updated", Snackbar.LENGTH_LONG)//snack bar long duration
                .setAction("undo", (v) -> {
                        String user = "admin";
                        String pass = "admin";
                        Main.setUserPass(user, pass);
                }).show();
    }
    else{
        Snackbar.make(v,"Please fill out correct values for username and password",Snackbar.LENGTH_SHORT)//Snack bar short duration
                .setAction("Action",null).show();
    }

}
```

The above code takes the text from the two edit texts (Username & Password) and stores them in two separate Strings. The code then ensures that they are not smaller than 2 characters long. It then calls a static method created within the Main.java class called”setUserPass “and then passes the two Strings to this method which then overwrites the username and password for the administrator within the app.

Class name: Student Provider

Description: creation of studentProvider class with setter and getter methods

```
// creation of student provider class
public class studentProvider {

    // declaration of attributes of a student provider object
    private String id;
    private String courseId;
    private String course;
    private String date;
    private String attend;
// constructor
    public studentProvider(String id,String courseId,String course,String time,String attend){
        this.id = id;
        this.courseId = courseId;
        this.course = course;
        this.date = time;
        this.attend = attend;
    }
```

The code above creates a student provider class to store the data retrieved from the database (id,CourseId,course,time,attend) and stores them as a studentProvider data structure.

Class:DeleteStudentRec

Description: deletes the record of a student for a particular course

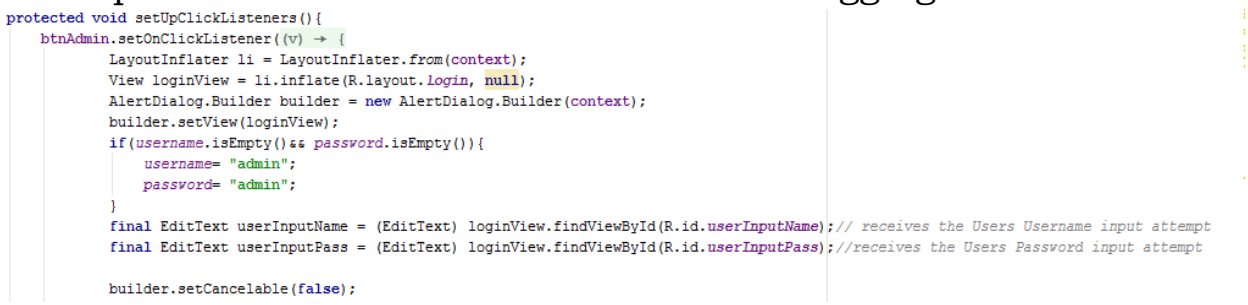
```
  public void deleteStudent(View v){
/ function executed when button delete is clicked
      EditText student = (EditText)findViewById(R.id.txt_StudID); // get input from view and set to variables
      EditText course = (EditText)findViewById(R.id.txt_Course);
      String s = student.getText().toString();
      String c = course.getText().toString();
      if(s.length()>1 && c.length()>1) {
          helper = new DBHelper(this);// instance helper created and database to writable
          db = helper.getWritableDatabase();
          // sql string to delete roll records from a particular student for a particular subject
          final String sql = "DELETE FROM " + RollContract.RollEntry.TABLE_NAME + " WHERE " + RollContract.RollEntry.STUDENT_ID + " = " + " '" + s +
                  " AND " + RollContract.RollEntry.COURSE_ID + " = " + " '" + c + "' " + ";";
          DialogBuilder(sql, db);
      }
      else {

          if(s.length()<=1){
              Snackbar.make(v,"Please enter an Id to delete",Snackbar.LENGTH_SHORT)//Snack bar short duration
                      .setAction("Action",null).show();
```

```
public void deleteAll(View v){ // function executed when button deleteAll is clicked
    helper = new DBHelper(this); // instance helper created and database to writable
    db = helper.getWritableDatabase();
    final String sql = "DELETE FROM " + RollContract.RollEntry.TABLE_NAME + ";"; // delete all sql statement
    DialogBuilder(sql,db);
```

The code above gets the student name and course from an EditText txt_StudID and txt_Course and converts it to a string. If it is valid then it is concatenated to an sql delete statement then executed using the DialogBuilder function containing the db.exec() function. The function below deletes all records using the same DialogBuilder function.

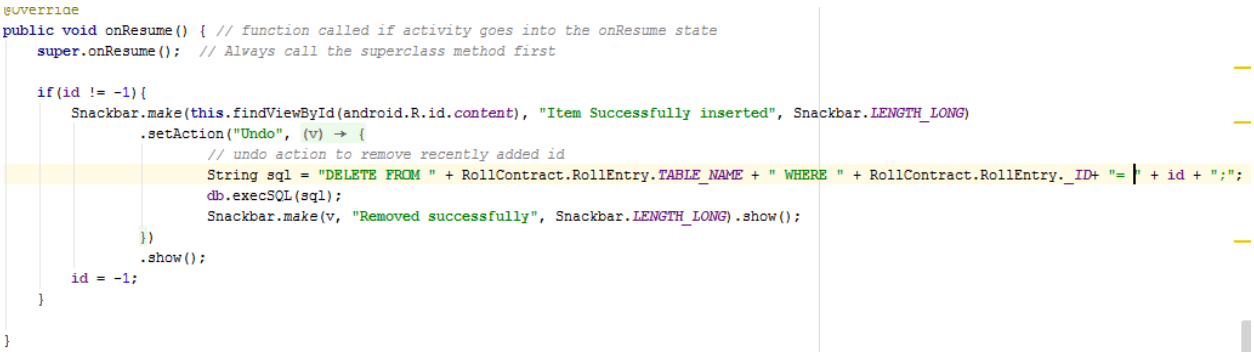
Class Name: Main

Description: create alert box for user admin logging.



```
protected void setUpClickListeners(){
    btnAdmin.setOnClickListener((v) -> {
            LayoutInflater li = LayoutInflater.from(context);
            View loginView = li.inflate(R.layout.login, null);
            AlertDialog.Builder builder = new AlertDialog.Builder(context);
            builder.setView(loginView);
            if(username.isEmpty()&& password.isEmpty()){
                username= "admin";
                password= "admin";
            }
            final EditText userInputName = (EditText) loginView.findViewById(R.id.userInputName);// receives the Users Username input attempt
            final EditText userInputPass = (EditText) loginView.findViewById(R.id.userInputPass);//receives the Users Password input attempt

            builder.setCancelable(false);
```

The above code uses a layout inflator login layout to enter username and password



```
@Override
public void onResume() { // function called if activity goes into the onResume state
    super.onResume();  // Always call the superclass method first

    if(id != -1){
        Snackbar.make(this.findViewById(android.R.id.content), "Item Successfully inserted", Snackbar.LENGTH_LONG)
                .setAction("Undo", (v) -> {
                        // undo action to remove recently added id
                        String sql = "DELETE FROM " + RollContract.RollEntry.TABLE_NAME + " WHERE " + RollContract.RollEntry._ID+ "= " + id + ";";
                        db.execSQL(sql);
                        Snackbar.make(v, "Removed successfully", Snackbar.LENGTH_LONG).show();
                })
                .show();
        id = -1;
    }

}
```

The above code allows the user to undo their submission of attendance in the event that they made the wrong course selection.

```
public void setUpSpinner(){//function to set up spinners
    Spinner spinner = (Spinner) findViewById(R.id.spinner);
Create an ArrayAdapter using the string array and a default spinner layout
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_spinner_item,courses);
Specify the layout to use when the list of choices appears
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
Apply the adapter to the spinner
    spinner.setAdapter(adapter);
    spinner.setOnItemSelectedListener(this);

    Spinner spinner2 = (Spinner) findViewById(R.id.spinner2);
    ArrayAdapter<CharSequence> adapter1 = ArrayAdapter.createFromResource(this,R.array.choices,androi
    );
Specify the layout to use when the list of choices appears
    adapter1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
Apply the adapter to the spinner
    spinner2.setAdapter(adapter1);
    spinner2.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
```

The above code sets up a spinner that will create a drop down menu that displays the various courses that a user can select from.

```
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener((view) -> {
            AlertDialog.Builder build = new AlertDialog.Builder(context);
            build.setTitle("Please enter a new course to add to the drop down menu");
Set up the input
            final EditText input = new EditText(context);
Specify the type of input expected; this, for example, sets the input as a password, and will mask the text
            input.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_CLASS_TEXT);
            build.setView(input);

Set up the buttons
            build.setPositiveButton("OK", (dialog, which) -> {
                    String text = input.getText().toString().toLowerCase();
                    if(text.contains("-")){
                        courses.add(text);
                    }
                    else{
                        Toast.makeText(context,"You did not enter a valid course",Toast.LENGTH_LONG).show();
                    }

            });
```

The code displayed above allows the user to enter their own course that will be added to the dropdown menu.