

Syrian Arab Republic

Ministry of Higher Education

Syrian Virtual University

Bachelor of Information Technology (BAIT)



Mobile Programming

(IWB303)

Prepared by:

Name	id
Mahmoud Howa	135620
Salim Meini	134475

Term: F20

Class: C2

Tutor: Muhammad Mazen Almustafa

1. Database:

Admin table:

Id	Int	Primary key	AUTOINCREMENT
Name	Text		
Username	Text		UNIQUE
Password	Text		

Course table:

Id	Int	Primary key	AUTOINCREMENT
Type	Text		

Question table:

Id	Int	Primary key	AUTOINCREMENT
Text	Text		
Id answer	Int		
mark	Int		
Id course	Int		

Answer table:

Id	Int	Primary key	AUTOINCREMENT
Text	Text		
Status	Text		
Id question	int	Foreign key	

Exam table:

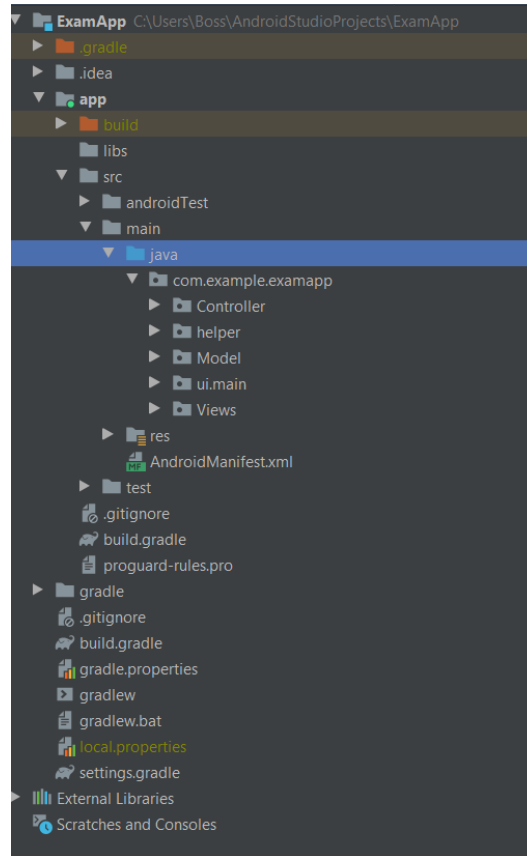
Id	Int	Primary key	AUTOINCREMENT
Student name	text		
Exam mark	Int		
Course name			

Student table:

Id	Int	Primary key	AUTOINCREMENT
Name	Text		
Username	Text		UNIQUE
Password	Text		

2. Project Structure

تم تقسيم المشروع بطريقة ال (MVC (Model, View, Controller)



- **Controller Package:**
 - Page Adapter: This class control of changing fragments in the Question Management Activity
 - Recycler View Adapter: This class is response of show the questions in recycler view
 - Exam Report Adapter: Response of show exam table data inside recycler view

- **Helper Package:**
 - Database Helper: This class contain functions to handle with database like get date or modify the data and create the database.
- **Model Package:**
 - Constant: This class contain the value of the theme to switch between themes depend on this value.
 - Data Admin: Contain: Constructors, Setter and Getter.
 - Data Answer: Contain: Constructors, Setter and Getter.
 - Data Course: Contain: Constructors, Setter and Getter.
 - Data Exam: Contain: Constructors, Setter and Getter.
 - Data Question: Contain: Constructors, Setter and Getter.
 - Data Student: Contain: Constructors, Setter and Getter.
 - Utils: Contain: The function setColorTheme() to switch between themes, And the functions that insert first important data to db.
- **Views Package:**
 - Activity Admin Dashboard.
 - Activity Courses.
 - Activity Question Management.
 - Add Question Activity.
 - Edit Question Activity.
 - Course1 (Fragment)
 - Course2 (Fragment)
 - Course3 (Fragment)
 - Course4 (Fragment)
 - Exam Activity.
 - Main Activity.
 - Result Score.
 - Setting Activity.
 - Exam Report Activity.

- **Main Activity:**

1- يتم إدخال المعلومات الأساسية إلى قاعدة البيانات والتحقق من عدم إدخالها أكثر من مرة

```
// Insert Data to DB
if(Constant.insertDataToDb == 0) {
    Data_Admin valuesAdm = new Data_Admin();
    valuesAdm.setName("mahmoud");
    valuesAdm.setUsername("admin");
    valuesAdm.setPassword("admin");
    new DatabaseHelper1( context: this).insertDataToAdmin(valuesAdm);
    Data_Student valuesStd = new Data_Student();
    valuesStd.setName("saleem");
    valuesStd.setUsername("student");
    valuesStd.setPassword("student");
    new DatabaseHelper1( context: this).insertDataToStudent(valuesStd);
    // Questions
    utils.insertFiveQuestionToCourseOne();
    utils.insertFiveQuestionToCourseTwo();
    utils.insertFiveQuestionToCourseThree();
    utils.insertFiveQuestionToCourseFour();
    // Answers
    utils.insertAnswersToQuestionsForCourseOne();
    utils.insertAnswersToQuestionsForCourseTwo();
    utils.insertAnswersToQuestionsForCourseThree();
    utils.insertAnswersToQuestionsForCourseFour();

    editor.putInt("data_is_inserted" , 1);
    editor.commit();
}
```

2- عند ضغط المستخدم على زر ال login يتم التحقق من اسم المستخدم وكلمة المرور المدخلين في الحقول المطلوبة من خلال التعليمات التالية:

```
// Login Check
login.setOnClickListener(v -> {
    if (password.getText().toString().isEmpty() || username.getText().toString().isEmpty()) {
        Snackbar.make(activity, text: "Username or password not found", Snackbar.LENGTH_LONG).show();
    } else if (username.getText().toString().toLowerCase().equals("admin")) {

        boolean logged = new DatabaseHelper1( context: MainActivity.this).login_admin(username.getText().toString(),
            password.getText().toString());
        if (logged) {
            enterPageSound.start();
            startActivity(new Intent(getApplicationContext(), ActivityAdminDashboard.class));
        } else {
            Snackbar.make(activity, text: "Error in username or password", Snackbar.LENGTH_LONG).show();
        }
    } else if (username.getText().toString().toLowerCase().equals("student")) {

        boolean logged = new DatabaseHelper1( context: MainActivity.this).login_student(username.getText().toString(),
            password.getText().toString());
        if (logged) {
            enterPageSound.start();
            String selectedStudent = username.getText().toString();
            Intent i = new Intent(getApplicationContext(), ActivityCourses.class);
            i.putExtra( name: "Selected_Student_Id", selectedStudent);
            startActivity(i);
        } else {
            Snackbar.make(activity, text: "Error in username or password", Snackbar.LENGTH_LONG).show();
        }
    } else {
        Snackbar.make(activity, text: "Don't Forget Any Field", Snackbar.LENGTH_LONG).show();
    }
});
```

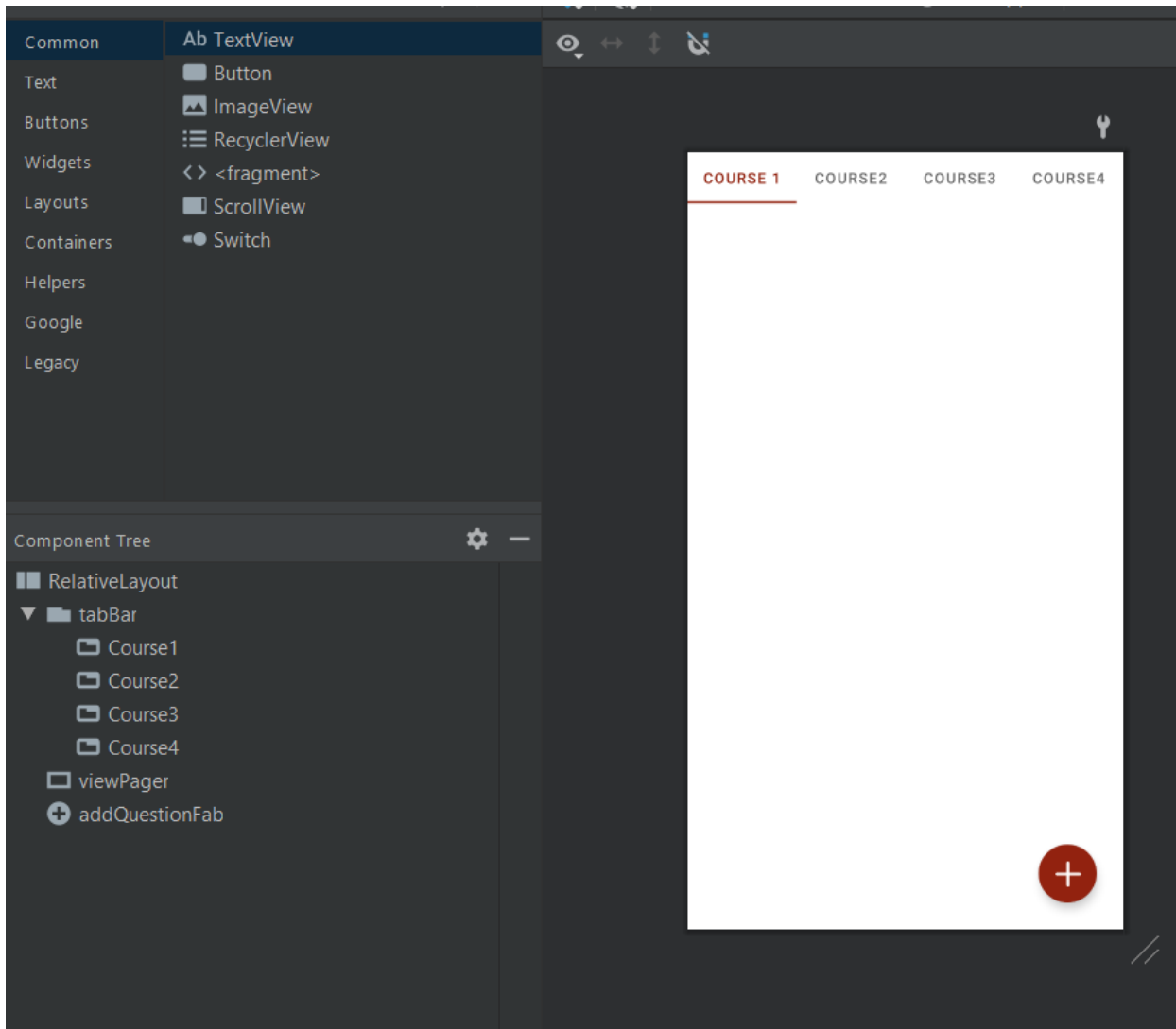
3- يتم نقل المستخدم الى النشاط Admin Dashboard إذا كان Admin ويتم نقله الى النشاط Courses Activity إذا كان Student

- **Admin Dashboard:**

يوجد في هذا النشاط أزرار لاختيار ما الذي يريد فعله المدير

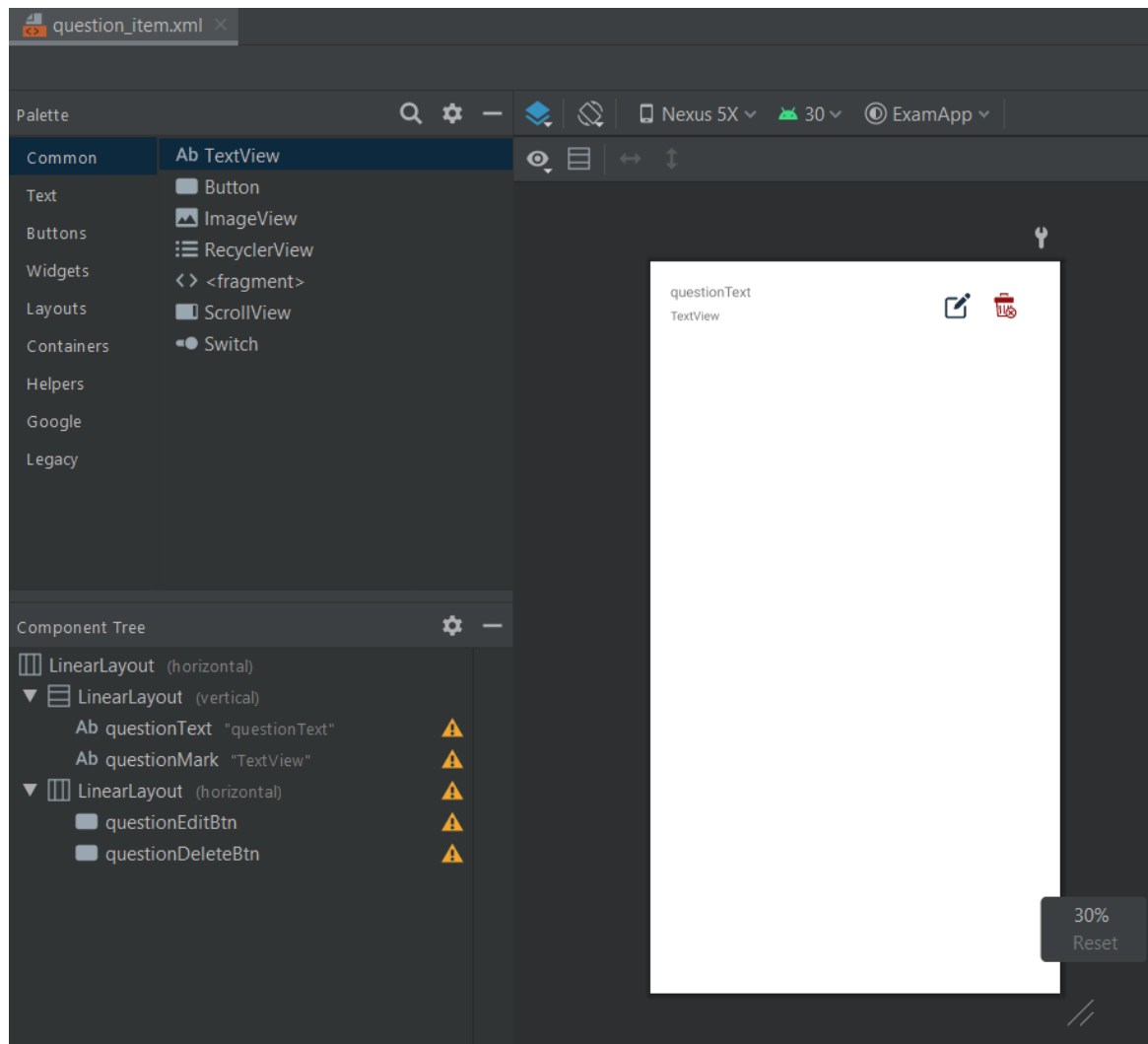
- **Question Management:**

يتم في هذه الواجهة عرض كافة الأسئلة الموجودة في جدول Question في قاعدة البيانات تم استخدام tab bar وانشاء 4 fragment ليتم التنقل بين الأسئلة والتصنيف حسب المادة والاداة view pager هي أداة تقوم بتعقب حركة اليد على الشاشة ليتمكن المستخدم من التنقل بين المواد من خلال السحب باليد الى اليمين او اليسار



- **Course1, Course2, Course3, Course4 Fragments:**

يوجد بداخل كل fragment أداة RecyclerView يتم عرض الأسئلة بداخلها
يتم استدعاء الأسئلة الى هذه الواجهات من خلال الربط بين
RecyclerView Adapter, Course fragment, databasehelper.getCourseQuestion()

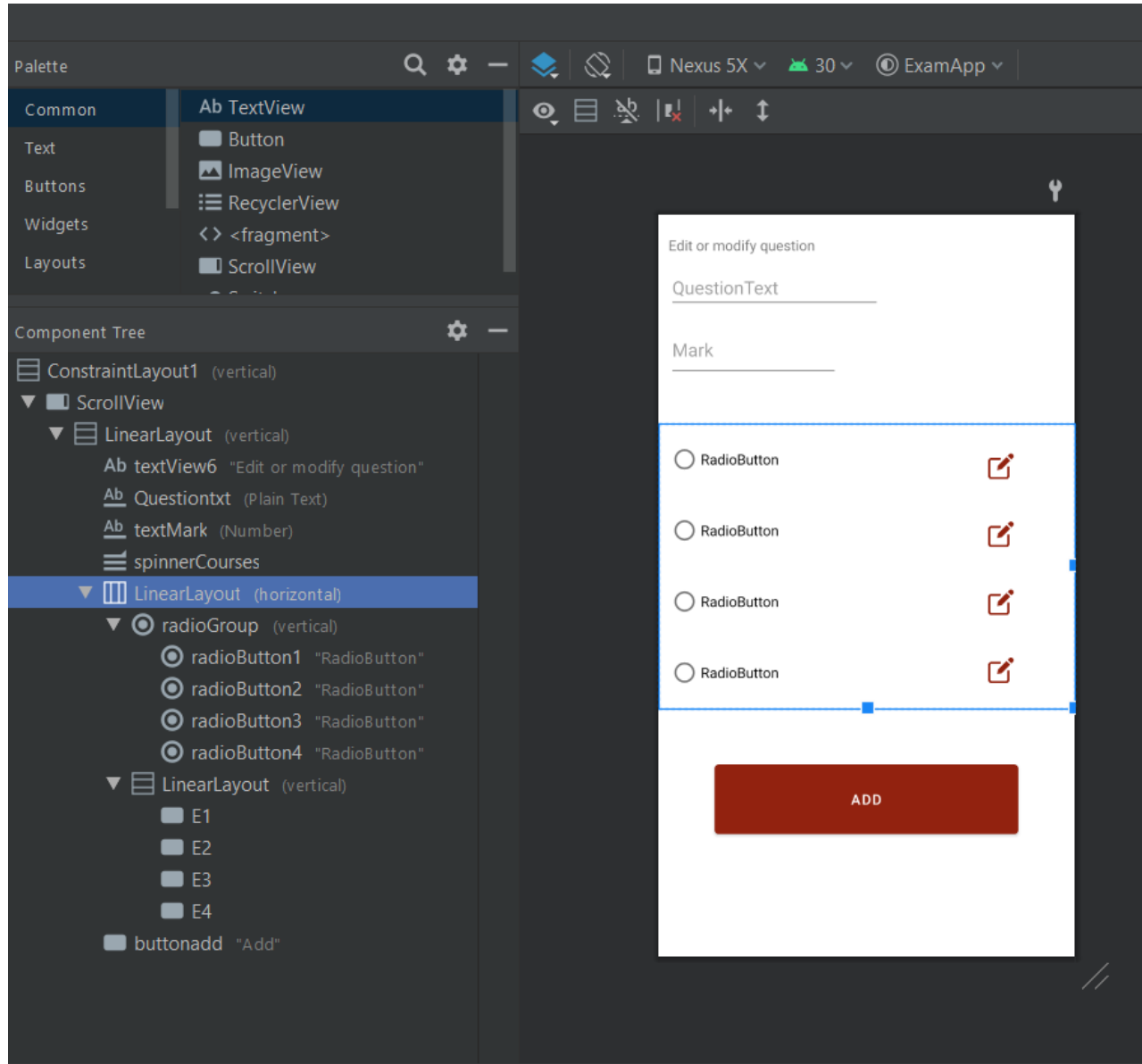


```
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    //Recycler
    course1_RecyclerView = (RecyclerView) v.findViewById(R.id.rvCourse1);
    LinearLayoutManager llm = new LinearLayoutManager(this.getActivity());
    course1_RecyclerView.setLayoutManager(llm);
    course1_RecyclerView.setItemAnimator(new DefaultItemAnimator());
    course1_RecyclerView.setHasFixedSize(true);
    // Get Course 1 Questions
    List<Data_Question> mydata = databaseHelper1.getCourse1Questions();

    //Adapter
    course1_RecyclerView.setAdapter(recyclerViewAdapter);
}
```

- **Add Question:**

يتم الانتقال الى هذا النشاط عند نقر المدير على زر الإضافة الموجود في صفحة إدارة الأسئلة
ويتم فيه تحديد المعلومات المطلوبة لإضافة السؤال والتحقق منها ومن ثم اضافتها الى قاعدة البيانات عند النقر
على زر Add



يتم بناء غرض من الصف `dataQuestion` ووضع كافة القيم المدخلة في الحقول في هذا الغرض ومن ثم اضافته إلى القاعدة عند استدعاء الدالة `insertDataToQuestion()`

```
Data_Question DQ = new Data_Question();
DQ.setText(Question);
DQ.setId_answer(RB);
DQ.setMark(Integer.parseInt(txtMark));
DQ.setId_course(idCourse);
long inserted = new DatabaseHelper1(getApplicationContext()).insertDataToQuestion(DQ);
```

```
// insert data to question table
public long insertDataToQuestion(Data_Question data){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put("text" , data.getText());
    cv.put("id_answer" , data.getId_answer());
    cv.put("mark", data.getMark());
    cv.put("id_course", data.getId_course());
    long id = db.insert(TABLE4_NAME , nullColumnHack: null , cv);
    db.close();
    return id;
}
```

```
// insert data to answer table
public long insertDataToAnswer(Data_Answer data){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put("text" , data.getText());
    cv.put("status" , data.getStatus());
    cv.put("id_question", data.getId_question());
    long id = db.insert(TABLE5_NAME , nullColumnHack: null , cv);
    db.close();
    return id;
}
```

- **Edit Question:**

عند قيام المدير بالضغط على زر التعديل نقوم ب جلب id السؤال المطلوب تعديله من ال recycler view adapter

```
RecyclerAdapter.java x
58     @Override
59     public void onBindViewHolder(@NonNull final MyViewHolder holder, final int position) {
60         final Data_Question data = question_List.get(position);
61         holder.questionText.setText(data.getText());
62         holder.questionMark.setText("Mark : " + data.getMark());
63         holder.delete.setOnClickListener(new View.OnClickListener() {
64             @Override
65             public void onClick(View v) { deleteData(position); }
66         });
67         holder.edit.setOnClickListener(new View.OnClickListener() {
68             @Override
69             public void onClick(View v) {
70                 Intent intent = new Intent(context, EditQuestionActivity.class);
71                 intent.putExtra( name: "position",String.valueOf(data.getId()));
72                 context.startActivity(intent);
73             }
74         });
75     }
76 }
77 }
78 }
79 }
```

ومن ثم نبني غرض ونسند له القيم الجديدة ونستدعي الدالة updateDataToQuestion

```
QuestionInfo = databaseHelper1.getQuestionId(position) ;
QuestionInfo.setText(Qtxt.getText().toString());
QuestionInfo.setId_answer(RBEdit);
QuestionInfo.setMark(Integer.parseInt(Mark.getText().toString()));

databaseHelper1.updateDataToQuestion(QuestionInfo);
```

```
// update data for question table
public int updateDataToQuestion(Data_Question data) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put("text" , data.getText());
    cv.put("id_answer" , data.getId_answer());
    cv.put("mark", data.getMark());
    String whereClause = "id =?";
    String whereArgs[] = {String.valueOf(data.getId())};
    return db.update(TABLE4_NAME, cv, whereClause, whereArgs);
}
```

- **Delete Question:**

عند ضغط المدير على زر حذف السؤال يتم استدعاء الدالة DeleteData الموجودة في Recycler View Adapter التي تمرر بداخلها Position و id السؤال وترسله إلى الدالة deleteDataFromQuestion

```
@Override
public void onBindViewHolder(@NonNull final MyViewHolder holder, final int position) {
    final Data_Question data = question_List.get(position);
    holder.questionText.setText(data.getText());
    holder.questionMark.setText("Mark : " + data.getMark());
    holder.delete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            deleteData(position);
        }
    });
}
```

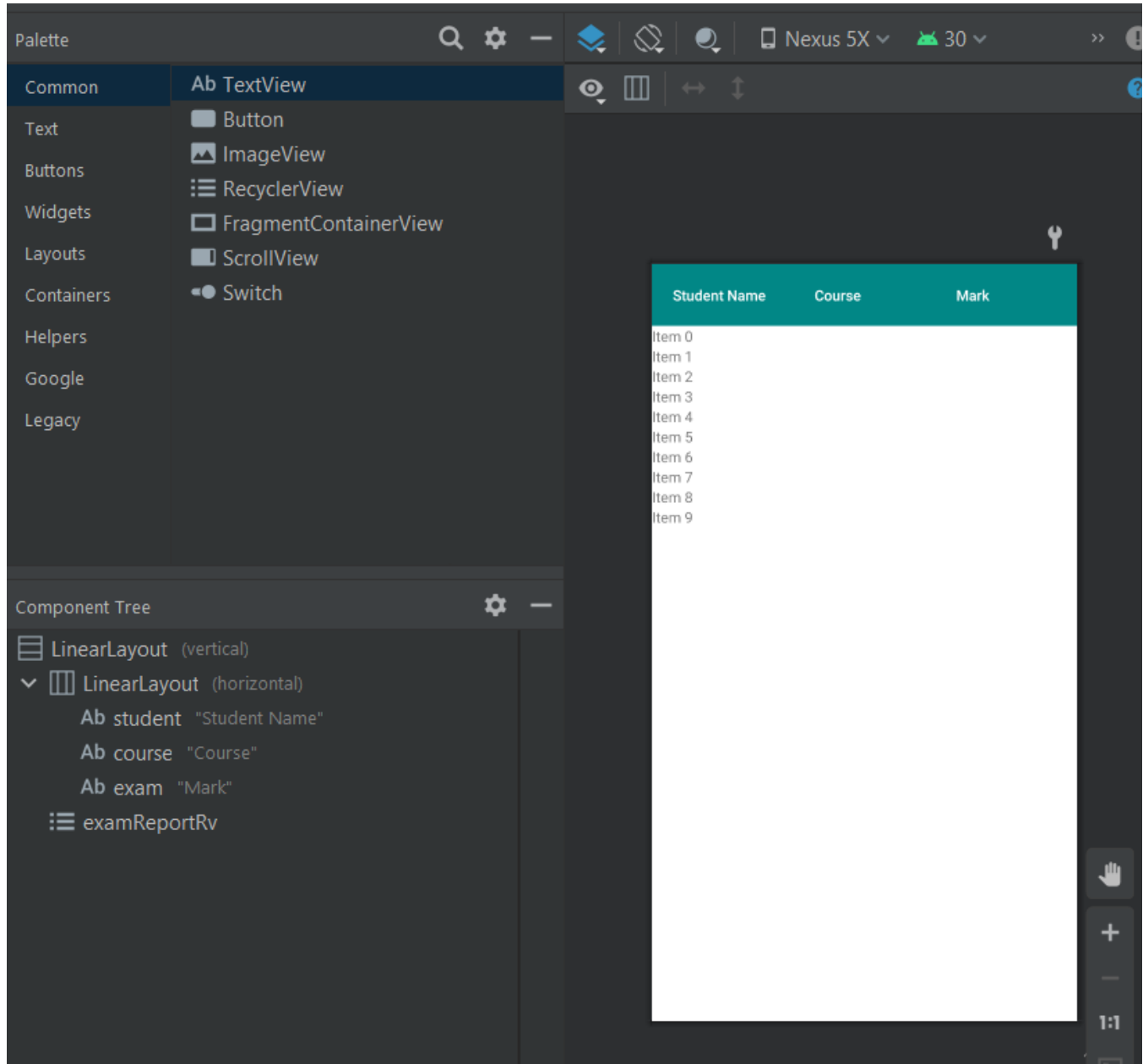
حذف الأسئلة والتحقق من إبقاء 5 أسئلة على الأقل

```
// Delete question
public void deleteData(int position){
    if (question_List.size() > 5) {
        int course_id = question_List.get(position).getId_course();
        if (course_id == 0) {
            DBH.deleteDataFromQuestion(question_List.get(position));
            question_List.remove(position);
            Course1.notifyAdapter();
        }
        if (course_id == 1) {
            DBH.deleteDataFromQuestion(question_List.get(position));
            question_List.remove(position);
            Course2.notifyAdapter();
        }
        if (course_id == 2) {
            DBH.deleteDataFromQuestion(question_List.get(position));
            question_List.remove(position);
            Course3.notifyAdapter();
        }
        if (course_id == 3) {
            DBH.deleteDataFromQuestion(question_List.get(position));
            question_List.remove(position);
            Course4.notifyAdapter();
        }
    } else (Toast.makeText(context, text: "Sorry questions can't be less than 5 ",Toast.LENGTH_LONG)).show();
}
```

```
// delete data from question table
public void deleteDataFromQuestion(Data_Question data){
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE4_NAME , whereClause: "id" + " =?",
        new String[]{String.valueOf(data.getId())});
    db.close();
}
```

- **Exam report activity:**

يتم في هذا النشاط عرض بيانات الامتحانات التي قدمها الطلاب وجلب هذا البيانات في مصفوفة أغراض من نوع الصف Data Exam وذلك بعد جلبها من قاعدة البيانات عن طريق الدالة `getAllDataExam()` ومن ثم عرض هذه البيانات في `RecyclerView`



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_exam_report);
    //Recycler
    examReportRv = findViewById(R.id.examReportRv);
    LinearLayoutManager llm = new LinearLayoutManager( context: this);
    examReportRv.setLayoutManager(llm);
    examReportRv.setItemAnimator(new DefaultItemAnimator());
    examReportRv.setHasFixedSize(true);
    //Adapter
    databaseHelper2 = new DatabaseHelper1( context: this);
    ExamRecyclerViewAdapter = new ExamReportAdapter( context: this,databaseHelper2.getAllExamData(),databaseHelper2);
    examReportRv.setAdapter(ExamRecyclerViewAdapter);
}

```

```

@NonNull
@NotNull
@Override
public ExamReportAdapter.ExamViewHolder onCreateViewHolder(@NonNull @NotNull ViewGroup parent, int viewType) {
    View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.exam_item,parent, attachToRoot: false);
    return new ExamViewHolder(v);
}

@Override
public void onBindViewHolder(@NonNull @NotNull ExamViewHolder holder, int position) {
    final Data_Exam data = exam_List.get(position);
    holder.nameStudent.setText(data.getNamestudent());
    holder.courseName.setText(data.getNamecourse());
    holder.examMark.setText("Mark: " + data.getMark());
}

@Override
public int getItemCount() { return exam_List.size(); }

public class ExamViewHolder extends RecyclerView.ViewHolder {
    public TextView nameStudent,courseName,examMark;
    public ExamViewHolder(@NonNull View v) {
        super(v);
        nameStudent = v.findViewById(R.id.studentName);
        courseName = v.findViewById(R.id.courseName);
        examMark = v.findViewById(R.id.examMark);
    }
}
}

```


- **Activity Courses:**

بعد ان يتم التسجيل الدخول كطالب يتم الانتقال الى هذا النشاط ليختار الطالب المادة التي يريد اجراء اختبار بها ويتم استدعاء الدالة start exam التي تنقل الطالب الى النشاط Exam Activity وترسل المتحول selectedCourse لكي يتم تحديد الكورس الذي سيتم جلب الأسئلة له

```
// Start Exam //
public void startExam(View v){
    int selectedCourse = v.getId();
    String nameStudent = position;
    Intent i = new Intent( packageContext: ActivityCourses.this,ExamActivity.class);
    i.putExtra( name: "Selected_Course_Id", selectedCourse);
    i.putExtra( name: "Selected_Student_Id", nameStudent);
    startActivity(i);
}
```

- **Exam Activity:**

جلب 5 أسئلة من قاعدة البيانات ووضعها في question List وهي غرض <DataQuestion> list

```
Bundle bundle = getIntent().getExtras();
coursePositionId = bundle.getInt( key: "Selected_Course_Id");
studentPosition = bundle.getString( key: "Selected_Student");
databaseHelper1 = new DatabaseHelper1( context: this);
currentQuestion = new Data_Question();

if(coursePositionId == R.id.courseOneBtn){ questionList = databaseHelper1.getCourse1Questions(); }
if(coursePositionId == R.id.courseTwoBtn){ questionList = databaseHelper1.getCourse2Questions(); }
if(coursePositionId == R.id.courseThreeBtn){ questionList = databaseHelper1.getCourse3Questions(); }
if(coursePositionId == R.id.courseFourBtn){ questionList = databaseHelper1.getCourse4Questions(); }
questionCountTotal = 5;
```

```

nextBtnQuestion1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        nextSound.start();
        if (!answered) {
            if (RR1.isChecked() || RR2.isChecked() || RR3.isChecked() || RR4.isChecked()) {
                checkAnswer();
                Toast.makeText(context: ExamActivity.this, text: "Done", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(context: ExamActivity.this, text: "Please select an answer", Toast.LENGTH_LONG).show();
            }
        } else {
            showNextQuestion();
        }
    }
});

```

```

// Previous Btn
prvBtnQuestion1 = findViewById(R.id.prvBtnQuestion1);
final MediaPlayer previousSound = MediaPlayer.create(context: this, R.raw.previous);
prvBtnQuestion1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        previousSound.start();
        showPreviousQuestion();
    }
});

```

- **Setting Activity:**

نافذة الإعدادات تحتوي على ثلاث خيارات لونية عند اختيار ثيم لوني يتم ارسال id الزر المختار الى الدالة setColorTheme التي تقوم بتحديد الثيم المطلوبة من بين الثيمات التي اضفناها مسبقاً في ملف Themes.xml

```
// Red Theme
<style name="RedTheme" parent="Theme.ExamApp">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/red</item>
    <item name="colorPrimaryDark">@color/darkBlue</item>
</style>
// Dark Blue Theme
<style name="DarkBlueTheme" parent="Theme.ExamApp">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/darkBlue</item>
    <item name="colorPrimaryDark">@color/red</item>
</style>
// Teal Theme
<style name="tealTheme" parent="Theme.ExamApp">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/teal_700</item>
    <item name="colorPrimaryDark">@color/darkBlue</item>
</style>
```

```
public void onClick(View v){
    utils.setColorTheme(v);

    editor.putInt("theme",Constant.theme);
    editor.commit();

    Intent intent = new Intent( packageContext: SettingActivity.this, MainActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(intent);
}
```

```
public class Utils {  
    private static int sTheme;  
    public final static int THEME_DEFAULT = 0;  
    public final static int THEME_DARKBLUE = 1;  
    public final static int THEME_TEAL = 2;  
  
    public void setColorTheme(View v) {  
        switch (v.getId()) {  
            case R.id.redBtn:  
                Constant.theme = R.style.RedTheme;  
                break;  
            case R.id.darkBlueBtn:  
                Constant.theme = R.style.DarkBlueTheme;  
                break;  
            case R.id.tealBtn:  
                Constant.theme = R.style.tealTheme;  
                break;  
            default:  
                Constant.theme = R.style.Theme_ExamApp;  
                break;  
        }  
    }  
}
```

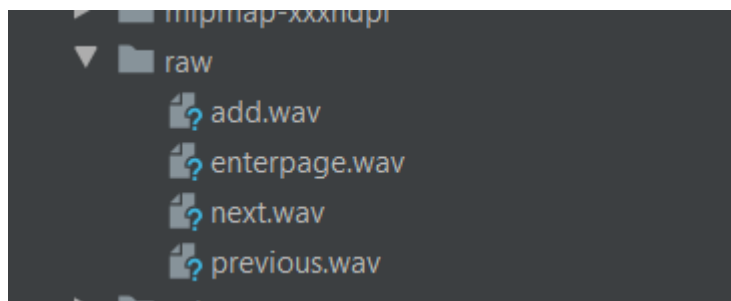
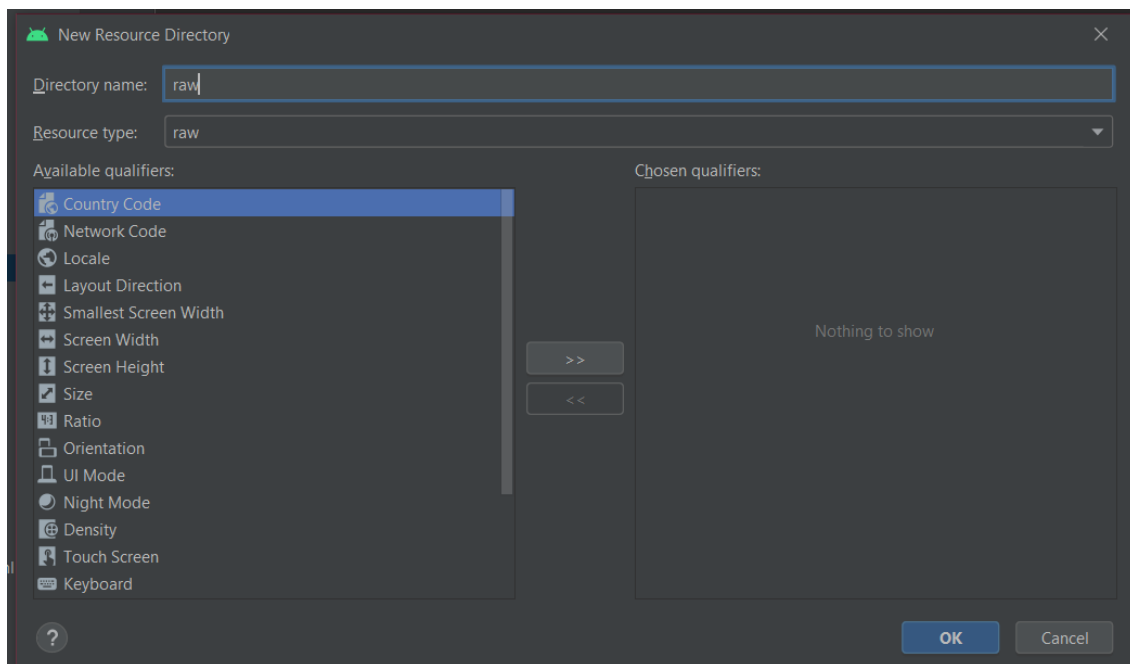
إذا كيف يتم تطبيق الثيم في كل النشاطات؟؟

عند انشاء النشاط نقوم بإنشاء shared preferences يجلب لنا قيمة ال theme المرسله من نشاط الإعدادات ونستدعي الدالة setTheme ونمرر لها هذه القيمة ليتم اختيار الثيم لهذا النشاط

```
//      Start  Set Theme
app_preferences = PreferenceManager.getDefaultSharedPreferences( context: this);
appTheme = app_preferences.getInt( key: "theme", defValue: 0);
setTheme(appTheme);

//      End set Theme
```

- الأصوات: قمنا ببناء مجلد Android Recourse Directory من نوع raw ليحوي بداخله كافة الأصوات المستخدمة في التطبيق



في النشاط الذي نريد تشغيل صوت به نبني غرض من نوع MediaPlayer ونمرر له المقطع الصوتي الذي نريده

```
final MediaPlayer enterPageSound = MediaPlayer.create(this,R.raw.enterpage);
```

ومن ثم عندما نريد تشغيل هذا الصوت عند حدث النقر على زرٍ ما نستدعي الدالة start() لهذا الغرض

```
enterPageSound.start();()
```

• كتم الأصوات:

في نشاط الإعدادات وضعنا زر من نوع switch وانشأنا الحدث setOnCheckedChangeListener

```
// Mute Sound
switchMute.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked){
            AudioManager amanager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
            amanager.setStreamMute(AudioManager.STREAM_MUSIC, state: true);
            editor.putInt("muteSwitch", 1);
            editor.commit();
        }
        else {
            AudioManager amanager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
            amanager.setStreamMute(AudioManager.STREAM_MUSIC, state: false);
            editor.putInt("muteSwitch", 0);
            editor.commit();
        }
    }
});
}
```

يتم ارسال القيمة 1 او 0 في shared preferences الى نفس هذا النشاط ليتم جلبها والتحقق منها لتطبيق كتم الصوت او عدم تطبيقه وحفظ هذه الحالة في حال اغلاق وفتح التطبيق

```
// Mute Status
switchStatus = app_preferences.getInt( key: "muteSwitch", defValue: 0);
setContentView(R.layout.activity_setting);
switchMute = findViewById(R.id.switchMute);
if (switchStatus == 1) {
    switchMute.setChecked(true);
}
```

3. المراجع

- مقرر المادة
- قناة بغداد الجديدة، دورة برمجة تطبيقات اندرويد [Link](#)
- <https://developer.android.com>
- دورة الاولى Android Studio كامل من الصفر وللإحتراف قناة Muhammed Essa