CENGONLINE Project Report

2016510001 Berk Adalı

2016510040 Elif Karataş

24.06.2020

Wednesday

Table of Contents

Project Requirements

Data Repository

SQLite is a relational database management system and also SQLite is suitable for Java. This project developed by Java programming language in Android Studio. So, SQLite is used in this project because of SQLite is one of suitable choices for saving the data.

Every data in CENGOnline, save in tables.

Student: Each student has own id that is autoincrement and unique. Name and surname are added by admin.

Teacher: Each teacher has own id that is autoincrement and unique. Name and surname are added by admin.

Course: Each course has own id that is autoincrement and unique. Course name, and students are added by teacher:

Assignment: Each assignment has own id that is autoincrement and unique. Course ID that is used to determine which course it belongs to, assignment name, assignment content, and due date are added by teacher.

Post: Each post has own id that is autoincrement and unique. Course ID that is used to determine which course it belongs to, post name, and post content are added by teacher.

Post Comments: Each post comment has own id that is autoincrement and unique. Course ID that is used to determine which course it belongs to, post ID that is used to determine which post it belongs to, comment content, name, and surname are added by teacher and student.

Announcement: Each announcement has own id that is autoincrement and unique. Course ID that is used to determine which course it belongs to, announcement name, and announcement content are added by teacher.

Message: Each message has own id that is autoincrement and unique. Receiver ID that is used to determine who will send the message, sender name, sender surname, receiver name, and receiver surname are added by teacher and student.

Submitted Works: Each submitted work has own id that is autoincrement and unique. Course ID that is used to determine which course it belongs to, Assignment ID that is used to determine which assignment it belongs to, and uploaded file are added by student.

Data repository in details:

```
public class Databasehelper extends SQLiteOpenHelper {
    public static final String STUDENT_TABLE = "STUDENT_TABLE";
    public static final String COLOUMN_STUDENT_ID = "STUDENT_ID";
    public static final String COLOUMN_NAME = "NAME";
    public static final String COLOUMN_STUDENT_NAME = "STUDENT_" + COLOUMN_NAME;
    public static final String COLOUMN_STUDENT_SURNAME = "STUDENT_SUR" + COLOUMN_NAME;

    public static final String TEACHER_TABLE = "TEACHER_TABLE";
    public static final String TEACHER_ID = "TEACHER_ID";
    public static final String TEACHER_NAME = "TEACHER_" + COLOUMN_NAME;
    public static final String TEACHER_SURNAME = "TEACHER_SUR" + COLOUMN_NAME;

    public static final String COURSE_TABLE = "COURSE_TABLE";
    public static final String COLOUMN_COURSE_ID = "COURSE_ID";
    public static final String COLOUMN_COURSE_NAME = "COURSE_" + COLOUMN_NAME;
    public static final String COLOUMN_STUDENT_1 = "STUDENT1";
    public static final String COLOUMN_STUDENT_2 = "STUDENT2";
    public static final String COLOUMN_STUDENT_3 = "STUDENT3";
```

Figure 1.1 Identfying columns in database.

```
public Databasehelper(@Nullable Context context) { super(context, name: "ceng.db", factory: null, version: 1); }
@Override
public void onCreate(SQLiteDatabase db) {
    String table1= "CREATE TABLE " + STUDENT_TABLE + " (" + COLOUMN_STUDENT_ID + " INTEGER PRIMARY KEY AUTOINCI
    db.execSQL(table1);
    String table2= "CREATE TABLE " + TEACHER_TABLE + " (" + TEACHER_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,
    db.execSQL(table2);
    String table3= "CREATE TABLE " + COURSE_TABLE + " (" + COLOUMN_COURSE_ID + " INTEGER PRIMARY KEY AUTOINCRE
    db.execSQL(table3);
    String table4= "CREATE TABLE " + POST_TABLE + " (" + POST_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + CO
    db.execSQL(table4);
    String table5= "CREATE TABLE " + ANNOUNCMENT_TABLE + " (" + COLOUMN_ANNOUNCMENT_ID + " INTEGER PRIMARY KEY
    db.execSQL(table5);
    String table6= "CREATE TABLE " + ASSINGMENT_TABLE + " (" + COLOUMN_ASSINGMENT_ID + " INTEGER PRIMARY KEY A
    db.execSQL(table6);
    String table7= "CREATE TABLE " + MESSAGE_TABLE + " (" + COLOUMN_MESSAGE_ID + " INTEGER PRIMARY KEY AUTOINCI
    db.execSQL(table7);
    String table8= "CREATE TABLE " + SUBMITTED_WORK_TABLE + " (" + COLOUMN_SUBMITTED_ID + " INTEGER PRIMARY KE
    db.execSQL(table8);
    String table9= "CREATE TABLE " + POSTCOMMENT_TABLE + " (" + COLOUMN_POST_COMMENT_ID + " INTEGER PRIMARY KE
    db.execSQL(table9);
}
```

Figure 1.2 Creating tables in database.

```
public boolean addStudent(Student student) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLOUMN_STUDENT_NAME,student.getName());
    cv.put(COLOUMN_STUDENT_SURNAME,student.getSurname());
    long insert = db.insert(STUDENT_TABLE, nullColumnHack: null,cv);
    if (insert == -1){
        return false;
    }
    else{
        return true;
    }
}
public List<Student> geteveryone(){
    List<Student>returnlist=new ArrayList<>();
    List<Integer>student_number_list = new ArrayList<>();

    String queryString = "SELECT * FROM " + STUDENT_TABLE;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, selectionArgs: null);
    if (cursor.moveToFirst()){
        do {
            int studentID = cursor.getInt( columnIndex: 0);
            String student_name = cursor.getString( columnIndex: 1);
            String student_surname = cursor.getString( columnIndex: 2);
            Student student=new Student(studentID,student_name,student_surname);
            returnlist.add(student);
        }while (cursor.moveToNext());
```

Figure 1.3 Example of one of database functions.

IDE

In this project, Used Android Studio that made for developing android applications. Android Studio developed by Google and JetBrains. It's available for free.

Libraries

androidx.appcompat.app.AppCompatActivity
android.os.Bundle
android.text.TextUtils: Used for is.Empty method for checking inputs in text.
android.view.View: It was used to swich to a new screen by clicking.
android.widget.Button: Used for buttons.
android.widget.EditText: Used for input strings.
android.widget.Toast: Used for messages shown messages by application.
android.content.Intent: To open new page content and keep the data in it.
android.widget.AdapterView: Used for showing the data in Array list in list view.
android.widget.ArrayAdapter: Used for saving data in the database into the Array list.
android.widget.ListView: Used for showing the data in application.
android.util.Log: Used for controling the datas in Array list.

For database class:

android.database.sqlite.SQLiteDatabase
android.database.sqlite.SQLiteOpenHelper
android.content.ContentValues

android.content.Context
android.database.Cursor

Classes

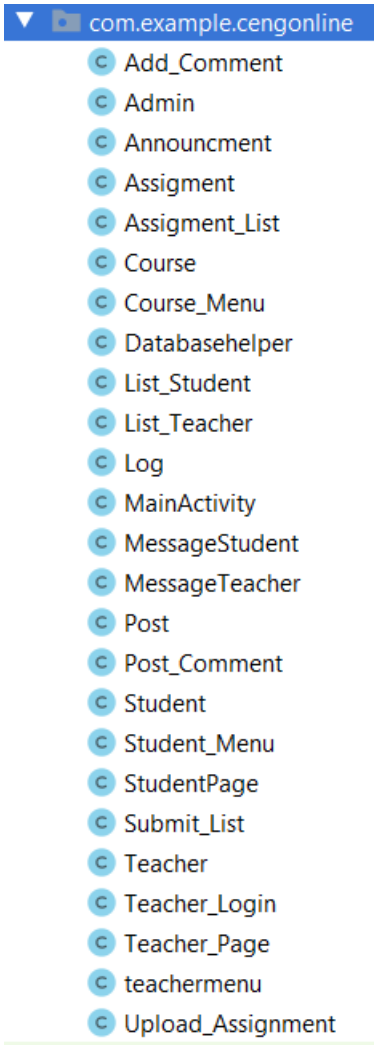

Figure 2.1 Classes of CENGOnline.

Add_Comment: Contains methods of add comment to a post.
Admin: Contains methods of open teacher and student pages.
Announcment: Contains methods of add, edit, delete announcement.
Assigment: Contains methods of add, edit, delete assignment.
Assigment_List: Contains methods of list assignment.
Course: Contains methods of add, edit, delete course.
Course_Menu: Contains methods of list assignments, announcements, and posts of course.
Databasehelper: Contains methods of database.
List_Student: Contains methods of list students.
List_Teacher: Contains methods of list teacher.
Log: Contains methods of student login.
MainActivity: Contains methods of main menu.
MessageStudent: Contains methods of student inbox and send message.
MessageTeacher: Contains methods of teacher inbox and send message.

Post: Contains methods of add, edit, delete post.
Student: Contains methods of get, set.
Student_Menu: Contains methods of main student menu. (List courses that enrolled by teacher, message button, exc.)
StudentPage: Contains methods of add student.
Submit_List: Contains methods of list submitted works.
Teacher: Contains methods of get, set.
Teacher_Login: Contains methods of teacher login.
Teacher_Page: Contains methods of add teacher.
teachermenu: Contains methods of main teacher menu. (List courses, add buttons, messages, exc.)
Upload_Assignment: Contains method of submit work by student.

User Interface

Used Android Emulator that is provided by Android Studio for user interface in this project. It has xml codes for designing.

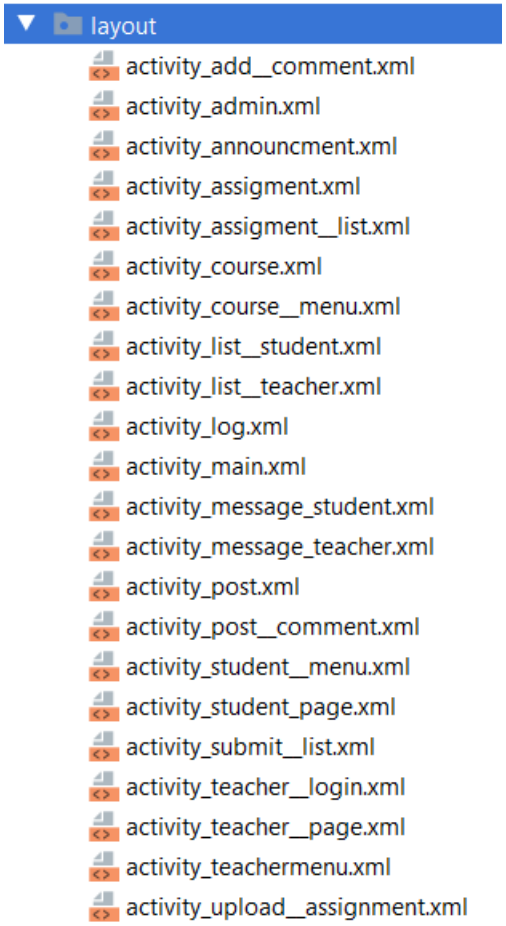

Figure 3.1 Xml files.

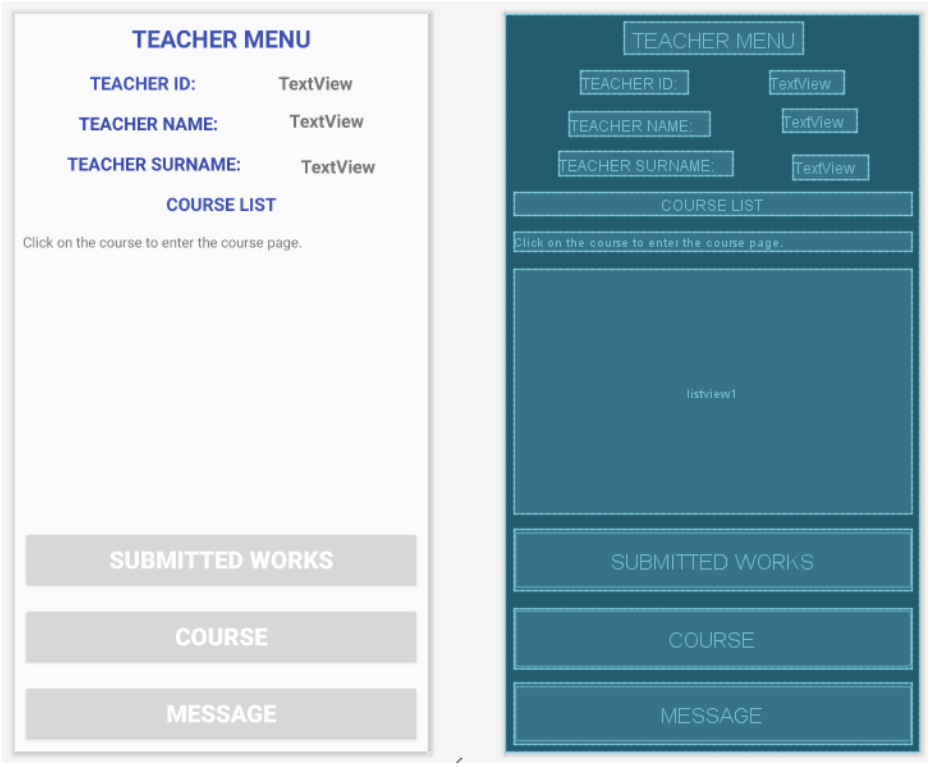

Figure 3.2 Design and Blueprint.

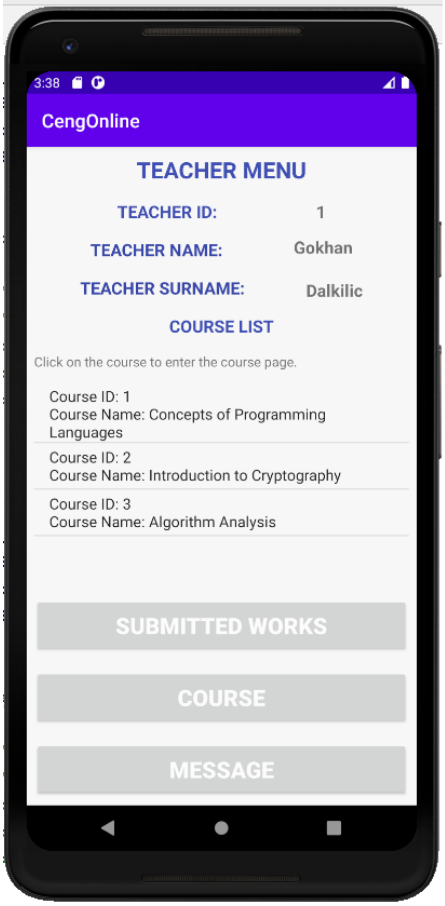

Figure 3.3 UI of teacher menu.

Abstract Data Type:

ADT is the name of the structure that regulates a group of data in computer science and the operations that can be performed on it. The word abstraction is that the data structure is a design and for the user, the inside of the structure is completely abstract, that the users do not require this data type to have knowledge about the application details.

As sample abstract data types, linked list, stack, queue, complex numbers, clusters can be counted.[1]

In this project, Array list is used as abstract data type.

```
final Databasehelper databaseHelper=new Databasehelper( context: Student_Menu.this);
 List<String> everyone=databaseHelper.StudentCourseListele1(Temp);
 List<String>everyone2=databaseHelper.StudentCourseListele2(Temp);
 List<String>everyone3=databaseHelper.StudentCourseListele3(Temp);
```

Figure 4.1 Use of array list in Student_Menu class.

Foreach Loop:

```
final List<String> everyone=databaseHelper.PostListele(Temp);
final ArrayAdapter postadapter = new ArrayAdapter<String>( con

for (String item: everyone) {
    Log.d(TAG, item);
}
```

Figure 5.1 Example of using foreach loop in Course_Menu class.

Switch-Case Condition:

```
public class Clik implements View.OnClickListener{

    //CRTL + O
    @Override
    public void onClick(View v) {

        switch (v.getId()){
            case R.id.btn_admin:
                openAdmin();
                break;
            case R.id.btn_teacher:
               openLoginTeacher();
                break;
            case R.id.btn_student:
                openLoginStudent();
                break;
            default:
        }
    }
}
```

Figure 6.1 Use of switch-case condition as clicking buttons in MainActivity.

Named Constants

A Named Constant is an identifier that represents a permanent value. The value of a variable may change during the execution of a program; but a Named Constant (or just Constant) represents permanent data that never changes.[2]

```
public static final String STUDENT_TABLE = "STUDENT_TABLE";
public static final String COLOUMN_STUDENT_ID = "STUDENT_ID";
```

Figure 7.1 Use of named constants in Databasehelper class.

Method Overloading

With method overloading, multiple methods can have the same name with different parameters.

```
public List<String>SurnameSearch1(String x){
    List<String>veriler=new ArrayList<~>();
    SQLiteDatabase db = this.getWritableDatabase();
    String[]stun={COLOUMN_SENDER_NAME ,COLOUMN_SENDER_SURNAME,COL
    Cursor cursor = db.query(MESSAGE_TABLE,stun, selection: RECEIVER
    while (cursor.moveToNext()){
        veriler.add( "Sender Name: " + cursor.getString( columnInd
    }
    return veriler;
}

///////////////method Overloading
public List<String>SurnameSearch1(String x,String y){
    List<String>veriler=new ArrayList<~>();
    SQLiteDatabase db = this.getWritableDatabase();
    String[]stun={COLOUMN_MESSAGE_ID,COLOUMN_SENDER_NAME ,COLOUMN
    Cursor cursor = db.query(MESSAGE_TABLE,stun, selection: RECEIVER
    while (cursor.moveToNext()){
        veriler.add( "Sender Name: " + cursor.getString( columnInd
    }
    return veriler;
}
```

Figure 8.1 Use of method overloading in Databasehelper class.

Incomplete Tasks:

Inheritance and associative arrays are not used. Because the whole project was started with the use of focusing SQL, it could not be changed afterwards.

Additional Improvements

1. Adding student and teacher by admin.
2. Listing student and teacher by admin.

Problems Encountered

Android Studio was a program we just learned. So we spent a lot of time in the learning phase. We tried to do as much as possible at a basic level. We had a hard time deciding where to save the data.

Since Android Studio uses a lot of RAM while working, some problems arose during the testing stages. We searched the internet for these problems and solved the problem.

Contribution of Each Group Member

Each group member collaborated at all stages and showed equal dedication. These phases are project plan, database usage, methods, Android Studio learning research, design, report and presentation.

References

[1] http://bilgisayarkavramlari.sadievrenseker.com/2007/05/04/abstract-data-type-adt-soyut-veri-tipleri/

[2] https://www.therevisionist.org/software-engineering/java/tutorials/named-constants/#:~:text=A%20Named%20Constant%20is%20an,type%20of%20Java%20%E2%80%9Cconstant%E2%80%9D.