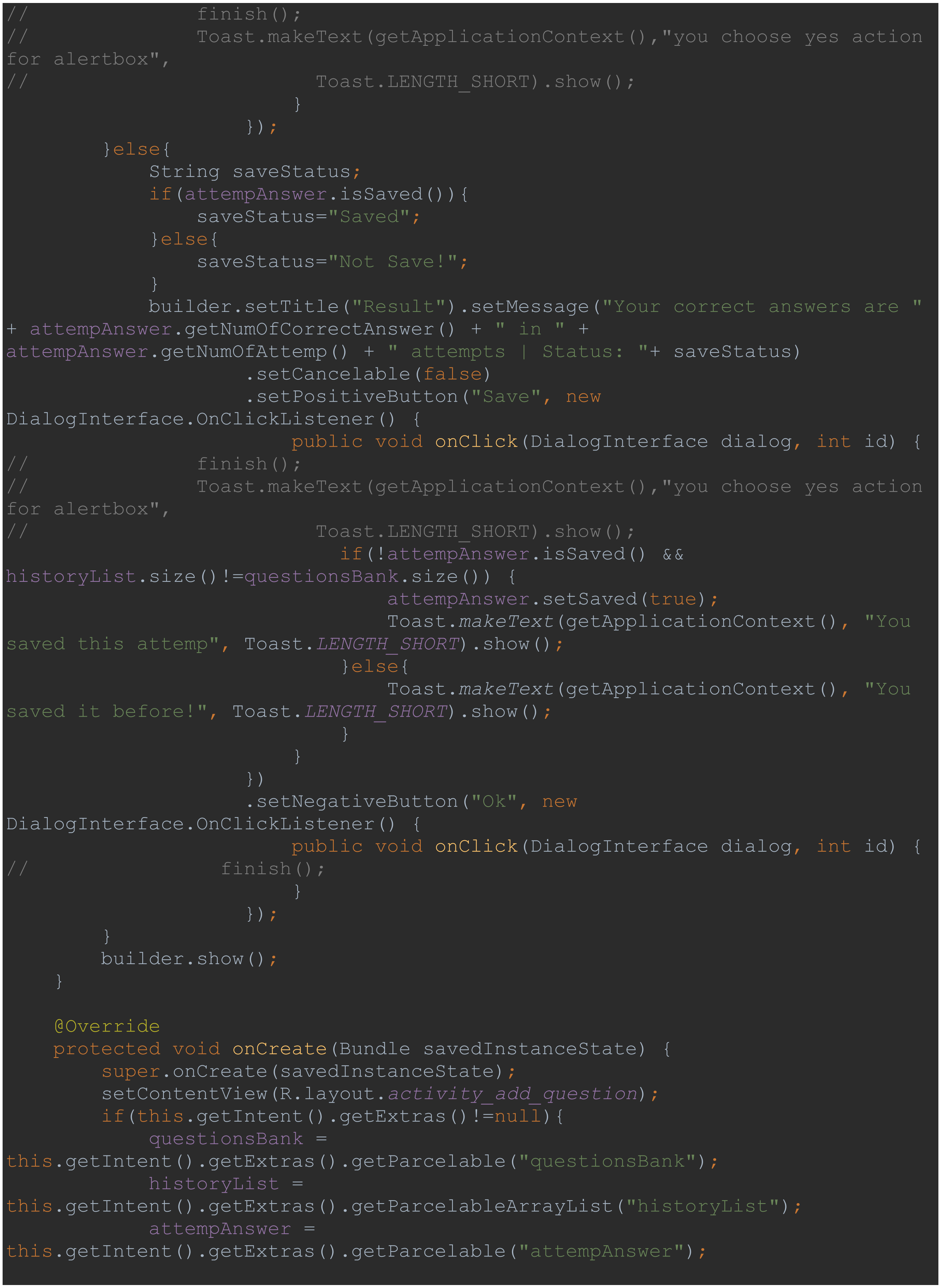
```
package com.example.quiz_assignment;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

public class AddQuestion extends AppCompatActivity {
    private boolean answer;
    QuestionBank questionsBank;
    ArrayList<HistoryAnswer> historyList = new ArrayList<>();
    AttempAnswer attempAnswer = new AttempAnswer();
    int indexQuestion = 0;
    int attempCount = 0;
    private StorageManager storageManager=new StorageManager();

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.app_menu, menu);
        return true;
    }

    @SuppressLint("NonConstantResourceId")
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                getTheAverage();
                return true;
            case R.id.item2:
                Toast.makeText(this, "Item2 was called",
Toast.LENGTH_SHORT).show();
                navigateToQuestionListView(new View(this));
                return true;
            case R.id.item3:
                Toast.makeText(this, "Item3 was called",
Toast.LENGTH_SHORT).show();
                attempAnswer.resetNew();
                attempCount=0;
                storageManager.saveAttempAnswer(this,attempAnswer);
                return true;
            case R.id.item4:
                Toast.makeText(this, "Edit Question was called",
Toast.LENGTH_SHORT).show();
                navigateToQuestionManager(new View(this));
                return true;
            case R.id.item5:
                navigateToHomePage(new View(this));
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        outState.putParcelable("questionsBank", questionsBank);
        outState.putParcelableArrayList("historyList", historyList);
        outState.putParcelable("attempAnswer", attempAnswer);
        outState.putInt("attempCount", attempCount);
        outState.putInt("indexQuestion", indexQuestion);
        super.onSaveInstanceState(outState);
    }

    public void navigateToHomePage(View v){
        Intent myIntent = new Intent(this, MainActivity.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    public void navigateToQuestionManager(View v) {
        Intent myIntent = new Intent(this, QuestionsManager.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    private void getTheAverage() {
        AlertDialog.Builder builder;
        builder = new AlertDialog.Builder(this);
        if(attempAnswer.isEmpty()){
            builder.setTitle("Result").setMessage("You do not have any
previous attempt!")
                    .setCancelable(false)
                    .setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
```



```
//              finish();
//              Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                          Toast.LENGTH_SHORT).show();
                        }
                    });
        }else{
            String saveStatus;
            if(attempAnswer.isSaved()){
                saveStatus="Saved";
            }else{
                saveStatus="Not Save!";
            }
            builder.setTitle("Result").setMessage("Your correct answers are "
+ attempAnswer.getNumOfCorrectAnswer() + " in " +
attempAnswer.getNumOfAttemp() + " attempts | Status: "+ saveStatus)
                    .setCancelable(false)
                    .setPositiveButton("Save", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//              finish();
//              Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                          Toast.LENGTH_SHORT).show();
                            if(!attempAnswer.isSaved() &&
historyList.size()!=questionsBank.size()) {
                                attempAnswer.setSaved(true);
                                Toast.makeText(getApplicationContext(), "You
saved this attemp", Toast.LENGTH_SHORT).show();
                            }else{
                                Toast.makeText(getApplicationContext(), "You
saved it before!", Toast.LENGTH_SHORT).show();
                            }
                        }
                    })
                    .setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                finish();
                        }
                    });
        }
        builder.show();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_question);
        if(this.getIntent().getExtras()!=null){
            questionsBank =
this.getIntent().getExtras().getParcelable("questionsBank");
            historyList =
this.getIntent().getExtras().getParcelableArrayList("historyList");
            attempAnswer =
this.getIntent().getExtras().getParcelable("attempAnswer");
```

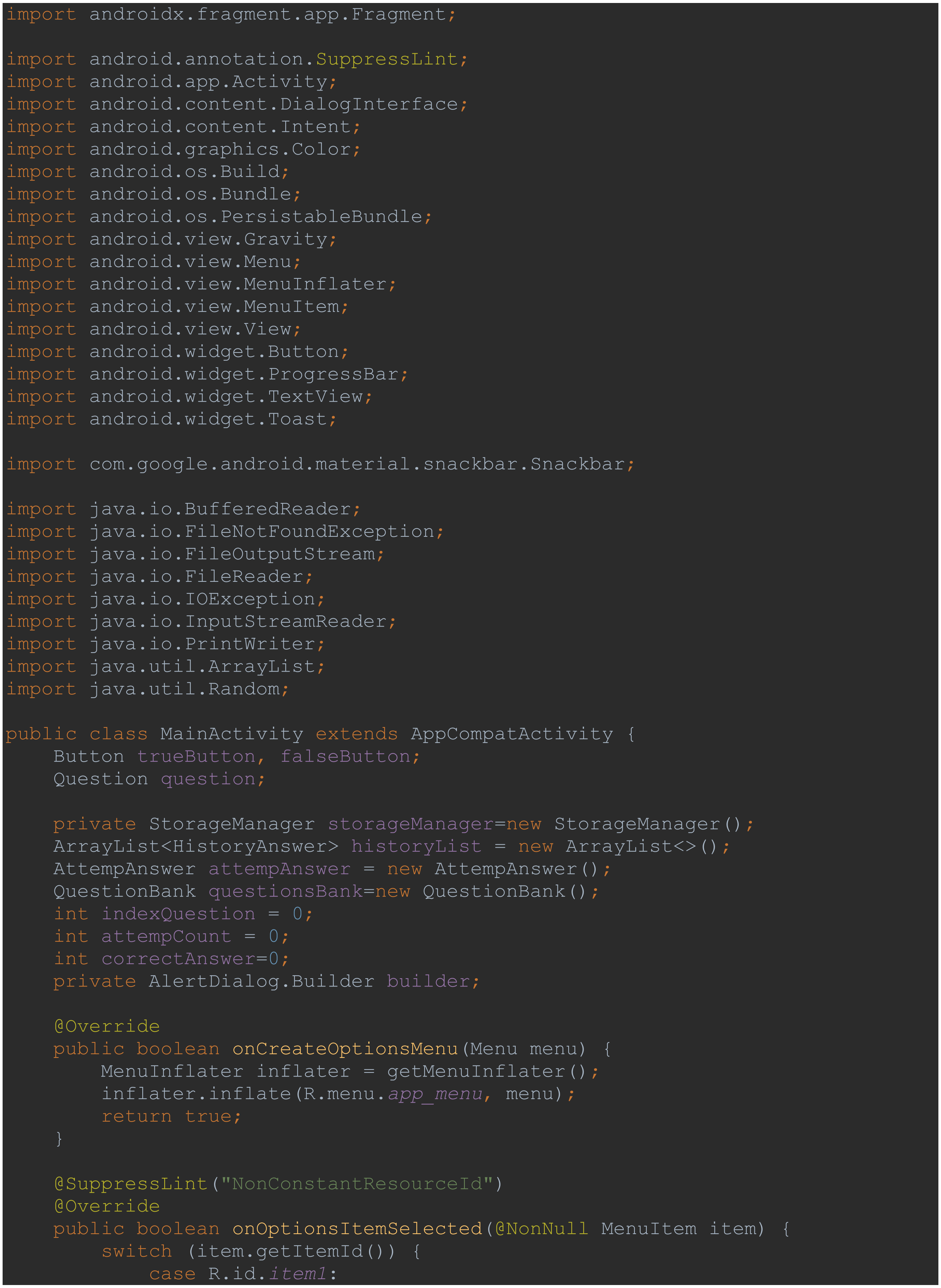
```
indexQuestion=this.getIntent().getExtras().getInt("indexQuestion");
            attempCount=this.getIntent().getExtras().getInt("attempCount");
        }
        if(savedInstanceState != null){
            questionsBank =
savedInstanceState.getParcelable("questionsBank");
            historyList =
savedInstanceState.getParcelableArrayList("historyList");
            attempAnswer = savedInstanceState.getParcelable("attempAnswer");
            indexQuestion=savedInstanceState.getInt("indexQuestion");
            attempCount=savedInstanceState.getInt("attempCount");
        }
        storageManager.load(this, questionsBank);
        setDefaultAnswer();
    }


    public void getAnswerValue(View v){
        RadioButton button= (RadioButton) v;
        String test=button.getText().toString();
        if(button.getText().toString().equals("true")){
            answer=true;
        }else{
            answer=false;
        }
    }

    public void onSubmit(View v){
        EditText input = (EditText) findViewById(R.id.question);
        if(input.getText().toString()==""){
            Toast.makeText(this, "Question cannot be empty",
Toast.LENGTH_SHORT).show();
        }else{
            if(input.getText().toString().length() <= 2){
                Toast.makeText(this, "Question must have at least 3
characters", Toast.LENGTH_SHORT).show();
            }else{
                Question newQuestion=new Question(input.getText().toString(),
answer);
                questionsBank.addQuestion(newQuestion);
                Toast.makeText(this, "Added new question!",
Toast.LENGTH_SHORT).show();
                input.setText("");
                setDefaultAnswer();
                storageManager.saveQuestions(this, questionsBank);
            }
        }
    }

    public void setDefaultAnswer(){
        RadioButton button = (RadioButton) findViewById(R.id.falseAnswer);
        button.setChecked(true);
    }

    public void navigateToQuestionListView(View v){
        Intent myIntent = new Intent(this, QuestionList.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }
}
```

```
package com.example.quiz_assignment;

import android.os.Parcel;
import android.os.Parcelable;

import java.util.ArrayList;

public class AttempAnswer implements Parcelable{
    private int numOfCorrectAnswer;
    private int numOfQuestions;
    private int numOfAttemp;
    private boolean isSaved;

    public AttempAnswer() {
        this.isSaved=false;
        this.numOfAttemp=0;
        this.numOfCorrectAnswer=0;
        this.numOfQuestions=0;
    }

    public boolean isEmpty(){
        return !isSaved && this.numOfAttemp == 0 && this.numOfCorrectAnswer
== 0 && this.numOfQuestions ==0;
    }

    public void increaseCorrect(){
        this.numOfCorrectAnswer++;
    }

    public void resetCorrect(){
        this.numOfCorrectAnswer=0;
    }

    public void increaseAttemp(){
        this.numOfAttemp++;
    }

    protected AttempAnswer(Parcel in) {
        numOfCorrectAnswer = in.readInt();
        numOfQuestions = in.readInt();
        numOfAttemp = in.readInt();
        isSaved = in.readByte() != 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeInt(numOfCorrectAnswer);
        dest.writeInt(numOfQuestions);
        dest.writeInt(numOfAttemp);
        dest.writeByte((byte) (isSaved ? 1 : 0));
    }

    @Override
    public int describeContents() {
        return 0;
    }

    public static final Creator<AttempAnswer> CREATOR = new
Creator<AttempAnswer>() {
        @Override
        public AttempAnswer createFromParcel(Parcel in) {
            return new AttempAnswer(in);
        }

        @Override
        public AttempAnswer[] newArray(int size) {
            return new AttempAnswer[size];
        }
    };

    public void resetNew(){
        this.isSaved=false;
        this.numOfAttemp=0;
        this.numOfCorrectAnswer=0;
        this.numOfQuestions=0;
    }

    public AttempAnswer(int numOfCorrectAnswer, int numOfQuestions, int
numOfAttemp, boolean isSaved) {
        this.numOfCorrectAnswer = numOfCorrectAnswer;
        this.numOfQuestions = numOfQuestions;
        this.numOfAttemp = numOfAttemp;
        this.isSaved = isSaved;
    }

    public StringBuilder getSb(){
        StringBuilder result= new StringBuilder();

result.append(this.numOfCorrectAnswer+","+this.numOfQuestions+","+this.numOfA
ttemp+","+this.isSaved);
        return result;
    }

    public int getNumOfCorrectAnswer() {
        return numOfCorrectAnswer;
    }

    public void setNumOfCorrectAnswer(int numOfCorrectAnswer) {
        this.numOfCorrectAnswer = numOfCorrectAnswer;
    }

    public int getNumOfQuestions() {
        return numOfQuestions;
    }

    public void setNumOfQuestions(int numOfQuestions) {
        this.numOfQuestions = numOfQuestions;
    }

    public int getNumOfAttemp() {
        return numOfAttemp;
    }

    public void setNumOfAttemp(int numOfAttemp) {
        this.numOfAttemp = numOfAttemp;
    }

    public boolean isSaved() {
        return isSaved;
    }

    public void setSaved(boolean saved) {
        isSaved = saved;
    }
}
```

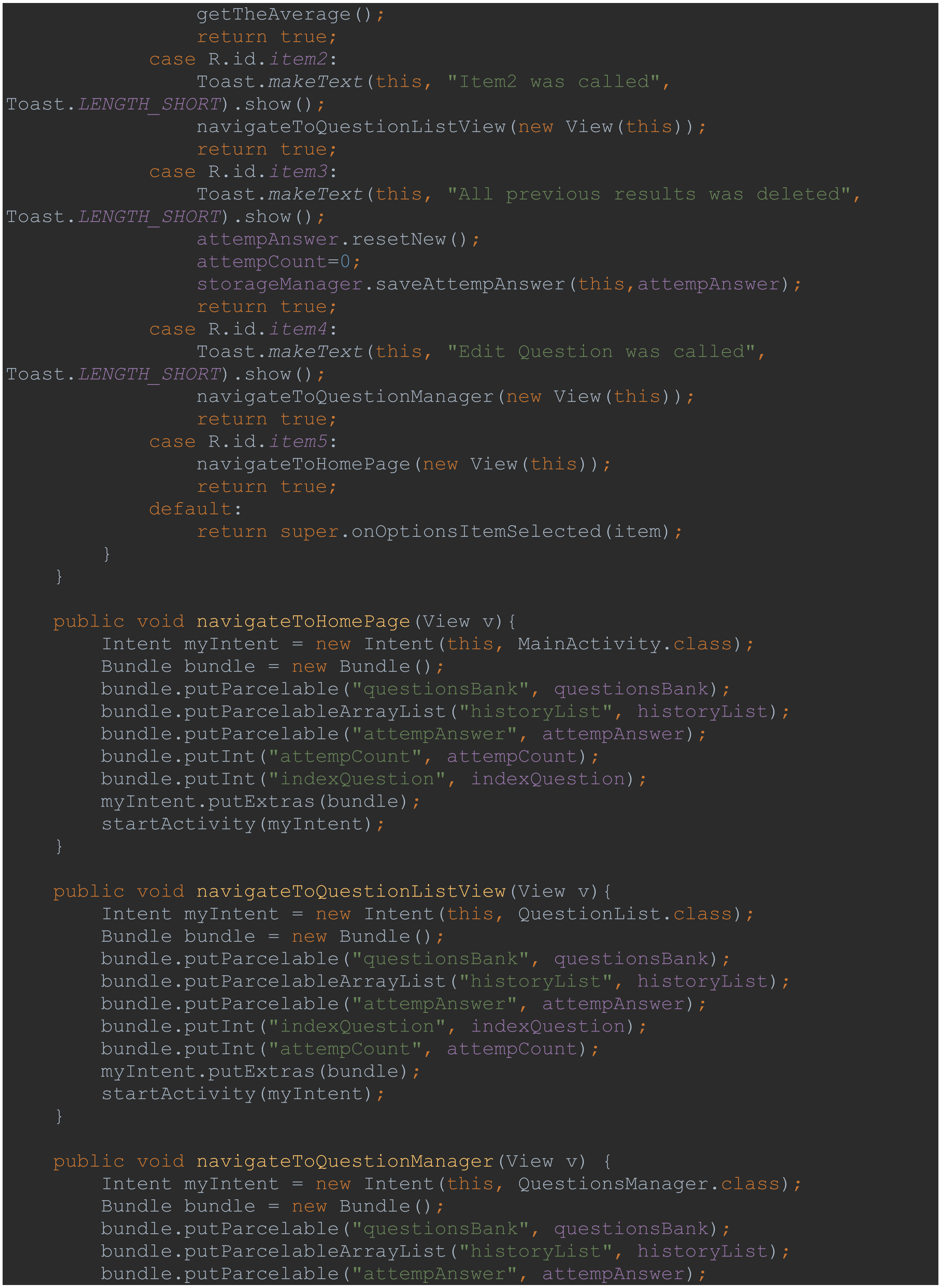
```
package com.example.quiz_assignment;

import android.os.Parcel;
import android.os.Parcelable;

public class HistoryAnswer implements Parcelable {
    private boolean userAnswer;
    private Question question;
    public HistoryAnswer(Question question, boolean userAnswer) {
        this.question=question;
        this.userAnswer=userAnswer;
    }

    protected HistoryAnswer(Parcel in) {
        userAnswer = in.readByte() != 0;
        question = in.readParcelable(Question.class.getClassLoader());
    }

    public static final Creator<HistoryAnswer> CREATOR = new
Creator<HistoryAnswer>() {
        @Override
        public HistoryAnswer createFromParcel(Parcel in) {
            return new HistoryAnswer(in);
        }

        @Override
        public HistoryAnswer[] newArray(int size) {
            return new HistoryAnswer[size];
        }
    };

    @Override
    public String toString() {
        return question.toString()+","+this.userAnswer;
    }

    public boolean getUserAnswer() {
        return userAnswer;
    }

    public Question getQuestion() {
        return question;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeByte((byte) (userAnswer ? 1 : 0));
        dest.writeParcelable(question, flags);
    }
}
```

```
package com.example.quiz_assignment;

import android.annotation.SuppressLint;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;


public class ItemAdapter extends
RecyclerView.Adapter<ItemAdapter.MyViewHolder> {
    ArrayList<Question> questionList;
    Context context;
    private final OnItemClickListener listener;

    public interface OnItemClickListener {
        void onItemClicked(Question question);
    }

    public ItemAdapter(Context context, ArrayList<Question> questionsBank,
OnItemClickListener onItemClickListener) {
        this.context=context;
        this.questionList=questionsBank;
        this.listener = onItemClickListener;
    }
    public static class MyViewHolder extends RecyclerView.ViewHolder{
        private TextView question;
        private TextView answer;

        public MyViewHolder(View view) {
            super(view);
            question =(TextView) view.findViewById(R.id.textView1);
            answer =(TextView) view.findViewById(R.id.textView2);
        }

        public TextView getQuestion() {
            return question;
        }
        public TextView getAnswer() {
            return answer;
        }
    }

    @NonNull
    @Override
    public ItemAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        LayoutInflater inflater=LayoutInflater.from(context);
        View view=inflater.inflate(R.layout.row, parent, false);
        View view2 = LayoutInflater.from(context)
                .inflate(R.layout.row, parent, false);
        return new MyViewHolder(view2);
    }

    @Override
    public void onBindViewHolder(@NonNull ItemAdapter.MyViewHolder holder,
int position) {

holder.getQuestion().setText(questionList.get(position).getQuestion());

holder.getAnswer().setText(String.valueOf(questionList.get(position).getAnswe
r()));
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                listener.onItemClicked(questionList.get(position));
            }
        });
    }

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

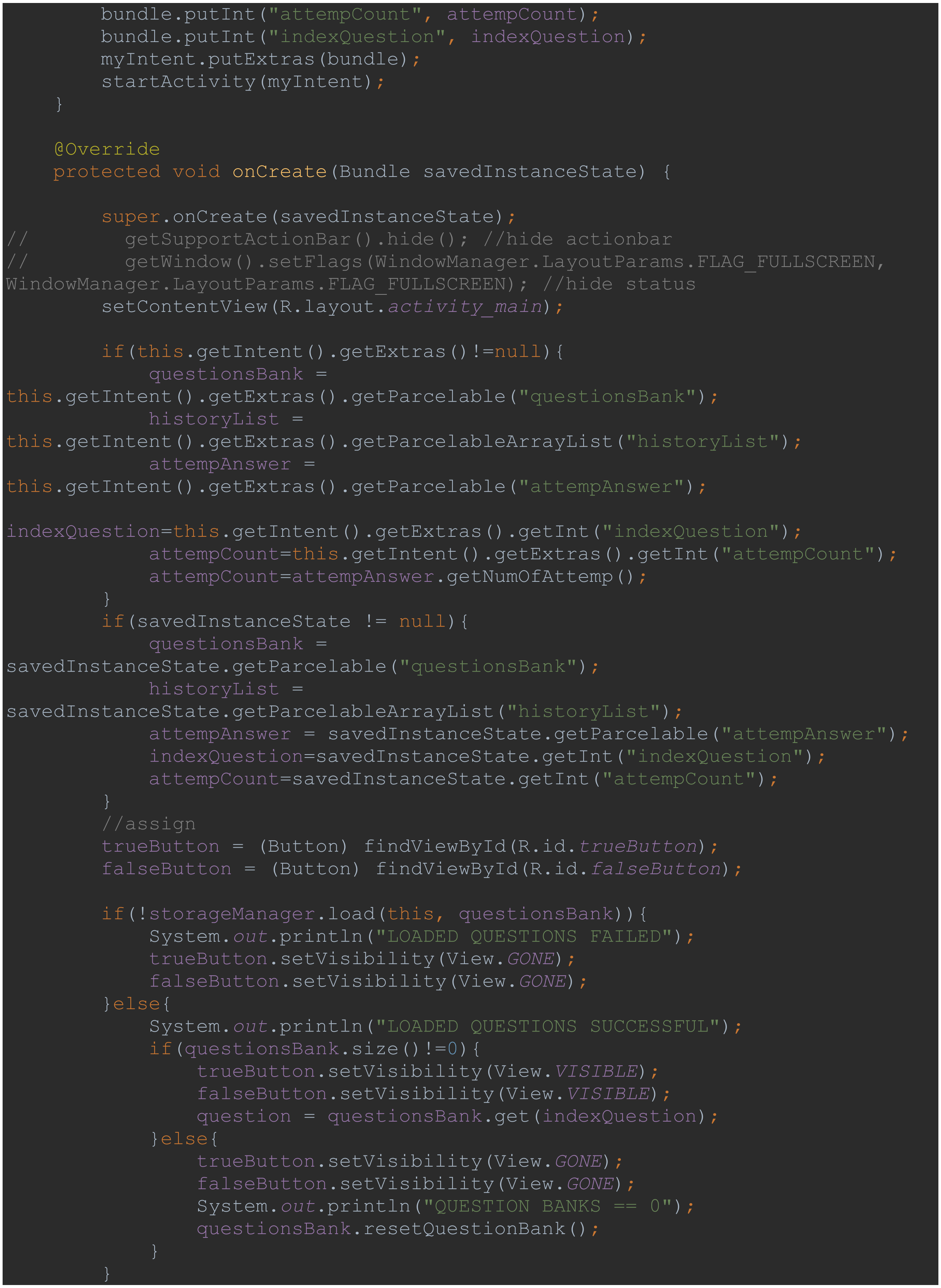
```
package com.example.quiz_assignment;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
```



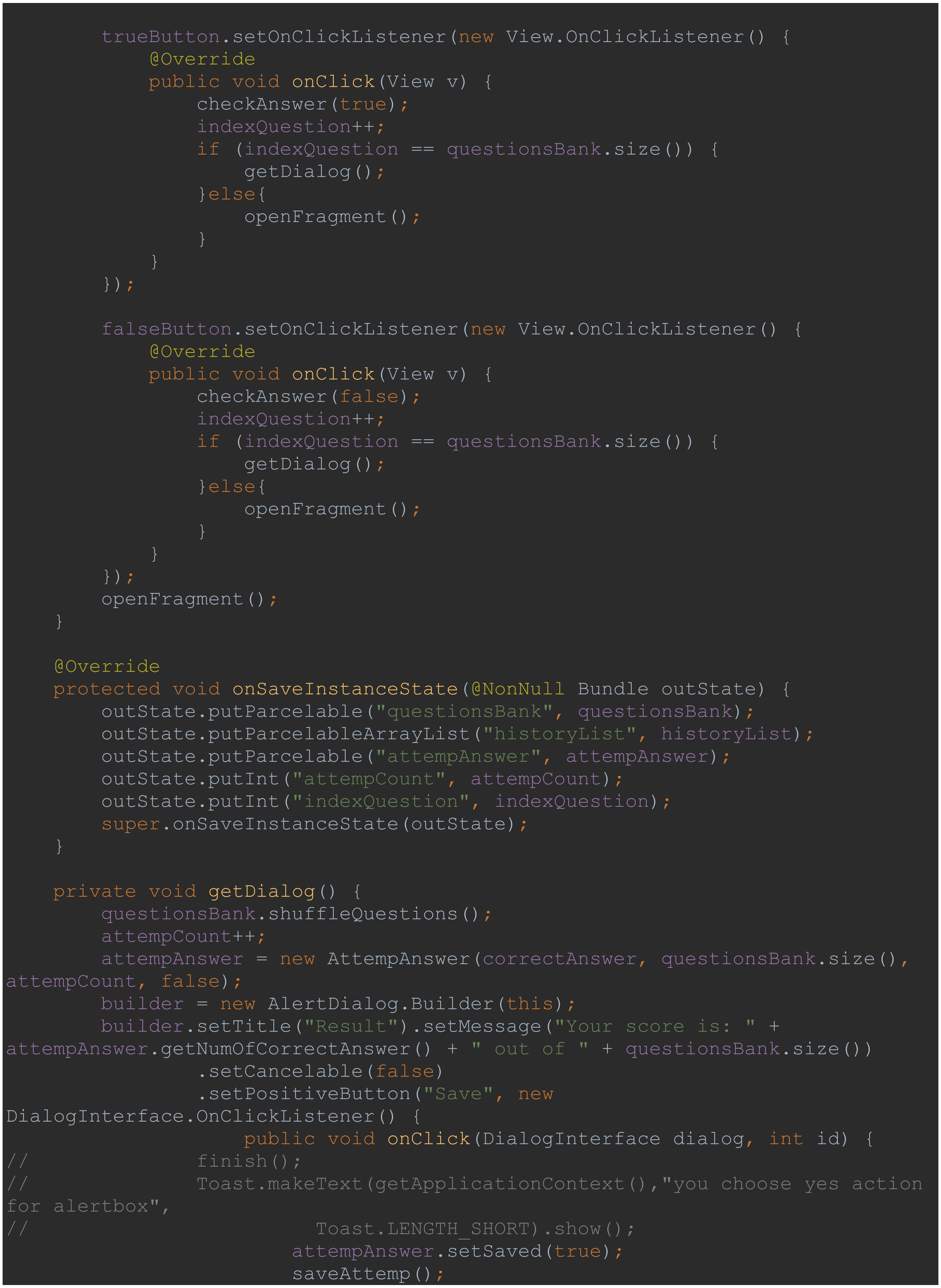
```java
import androidx.fragment.app.Fragment;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.os.PersistableBundle;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.material.snackbar.Snackbar;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Random;

public class MainActivity extends AppCompatActivity {
    Button trueButton, falseButton;
    Question question;

    private StorageManager storageManager=new StorageManager();
    ArrayList<HistoryAnswer> historyList = new ArrayList<>();
    AttempAnswer attempAnswer = new AttempAnswer();
    QuestionBank questionsBank=new QuestionBank();
    int indexQuestion = 0;
    int attempCount = 0;
    int correctAnswer=0;
    private AlertDialog.Builder builder;

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.app_menu, menu);
        return true;
    }

    @SuppressLint("NonConstantResourceId")
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
```



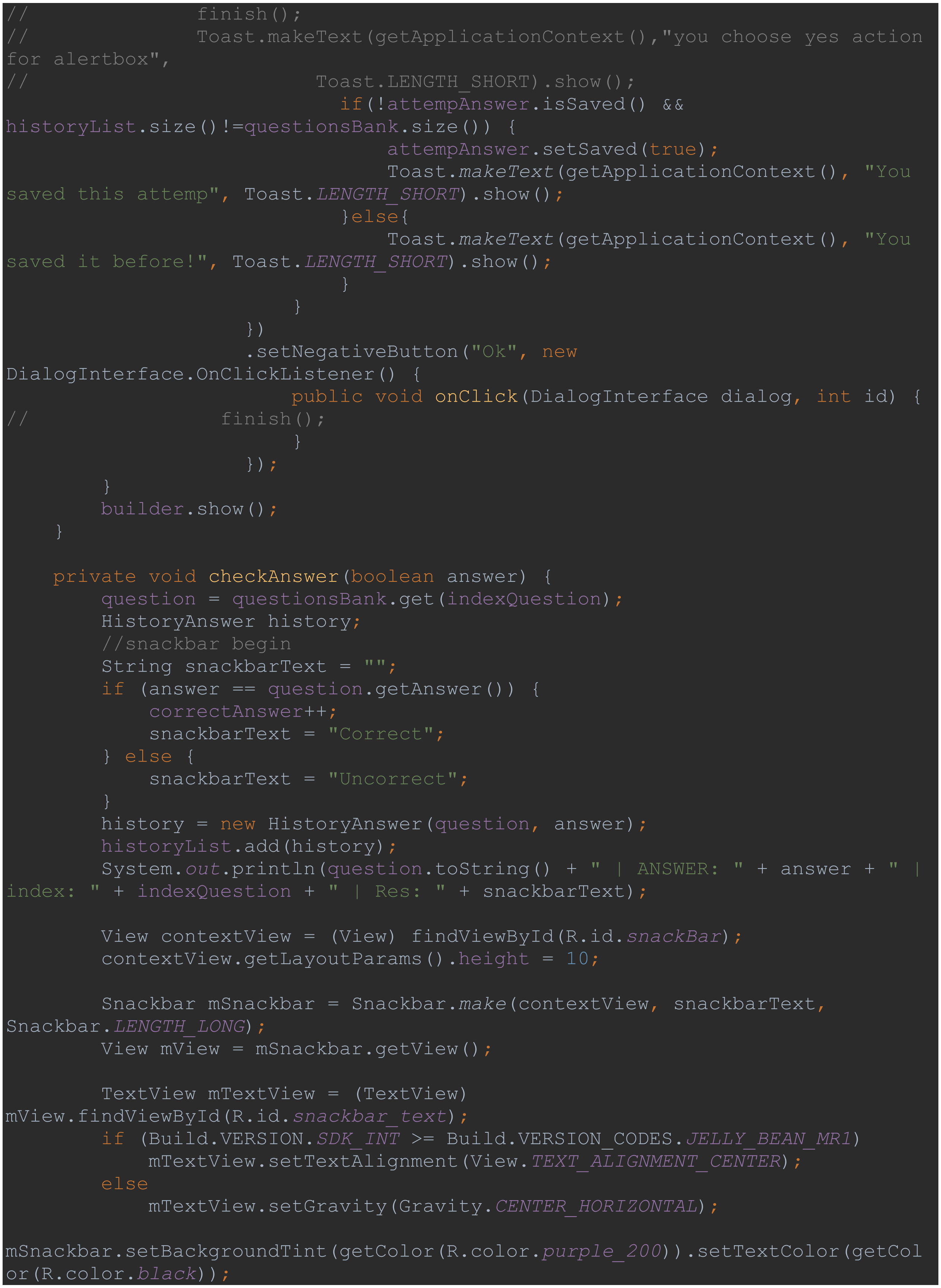
```
                getTheAverage();
                return true;
            case R.id.item2:
                Toast.makeText(this, "Item2 was called",
Toast.LENGTH_SHORT).show();
                navigateToQuestionListView(new View(this));
                return true;
            case R.id.item3:
                Toast.makeText(this, "All previous results was deleted",
Toast.LENGTH_SHORT).show();
                attempAnswer.resetNew();
                attempCount=0;
                storageManager.saveAttempAnswer(this,attempAnswer);
                return true;
            case R.id.item4:
                Toast.makeText(this, "Edit Question was called",
Toast.LENGTH_SHORT).show();
                navigateToQuestionManager(new View(this));
                return true;
            case R.id.item5:
                navigateToHomePage(new View(this));
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    public void navigateToHomePage(View v){
        Intent myIntent = new Intent(this, MainActivity.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    public void navigateToQuestionListView(View v){
        Intent myIntent = new Intent(this, QuestionList.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("indexQuestion", indexQuestion);
        bundle.putInt("attempCount", attempCount);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    public void navigateToQuestionManager(View v) {
        Intent myIntent = new Intent(this, QuestionsManager.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
```



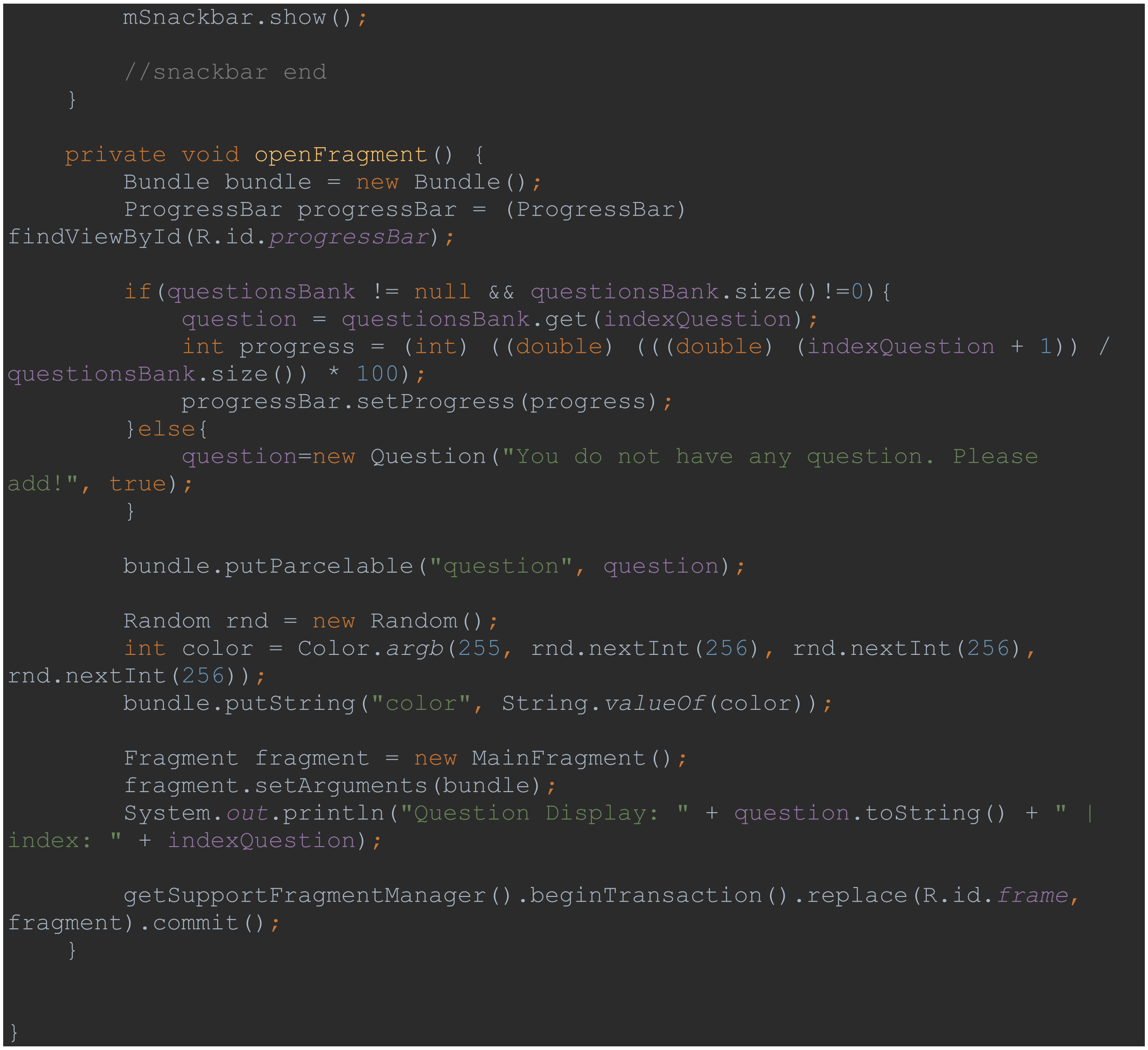
```
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
//          getSupportActionBar().hide(); //hide actionbar
//          getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN); //hide status
        setContentView(R.layout.activity_main);

        if(this.getIntent().getExtras()!=null){
            questionsBank =
this.getIntent().getExtras().getParcelable("questionsBank");
            historyList =
this.getIntent().getExtras().getParcelableArrayList("historyList");
            attempAnswer =
this.getIntent().getExtras().getParcelable("attempAnswer");

indexQuestion=this.getIntent().getExtras().getInt("indexQuestion");
            attempCount=this.getIntent().getExtras().getInt("attempCount");
            attempCount=attempAnswer.getNumOfAttemp();
        }
        if(savedInstanceState != null){
            questionsBank =
savedInstanceState.getParcelable("questionsBank");
            historyList =
savedInstanceState.getParcelableArrayList("historyList");
            attempAnswer = savedInstanceState.getParcelable("attempAnswer");
            indexQuestion=savedInstanceState.getInt("indexQuestion");
            attempCount=savedInstanceState.getInt("attempCount");
        }
        //assign
        trueButton = (Button) findViewById(R.id.trueButton);
        falseButton = (Button) findViewById(R.id.falseButton);

        if(!storageManager.load(this, questionsBank)){
            System.out.println("LOADED QUESTIONS FAILED");
            trueButton.setVisibility(View.GONE);
            falseButton.setVisibility(View.GONE);
        }else{
            System.out.println("LOADED QUESTIONS SUCCESSFUL");
            if(questionsBank.size()!=0){
                trueButton.setVisibility(View.VISIBLE);
                falseButton.setVisibility(View.VISIBLE);
                question = questionsBank.get(indexQuestion);
            }else{
                trueButton.setVisibility(View.GONE);
                falseButton.setVisibility(View.GONE);
                System.out.println("QUESTION BANKS == 0");
                questionsBank.resetQuestionBank();
            }
        }
```



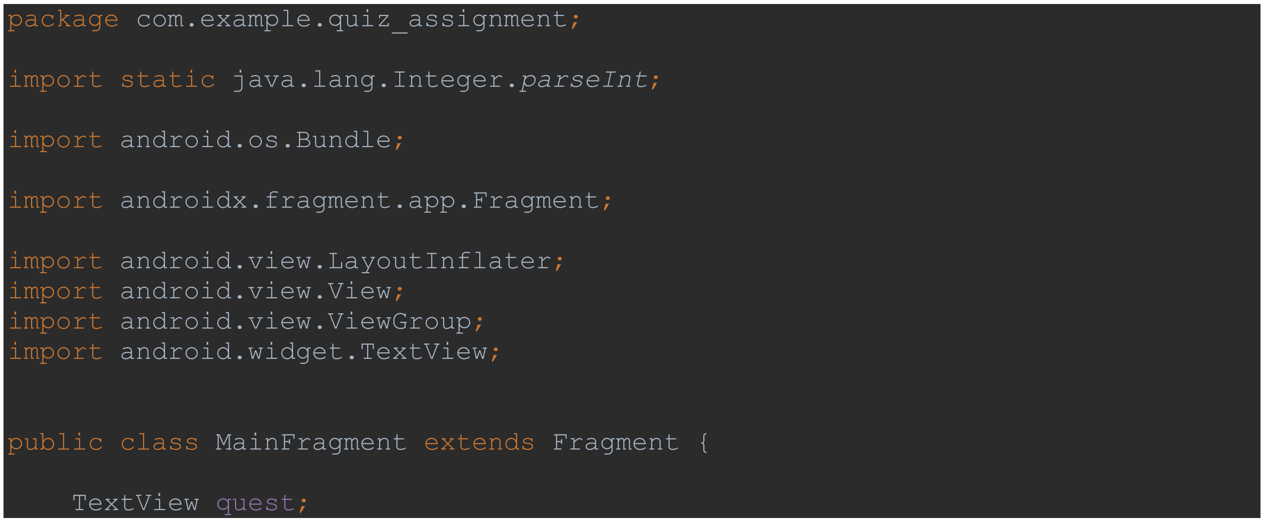
```
        trueButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                checkAnswer(true);
                indexQuestion++;
                if (indexQuestion == questionsBank.size()) {
                    getDialog();
                }else{
                    openFragment();
                }
            }
        });

        falseButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                checkAnswer(false);
                indexQuestion++;
                if (indexQuestion == questionsBank.size()) {
                    getDialog();
                }else{
                    openFragment();
                }
            }
        });
        openFragment();
    }

    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        outState.putParcelable("questionsBank", questionsBank);
        outState.putParcelableArrayList("historyList", historyList);
        outState.putParcelable("attempAnswer", attempAnswer);
        outState.putInt("attempCount", attempCount);
        outState.putInt("indexQuestion", indexQuestion);
        super.onSaveInstanceState(outState);
    }

    private void getDialog() {
        questionsBank.shuffleQuestions();
        attempCount++;
        attempAnswer = new AttempAnswer(correctAnswer, questionsBank.size(),
attempCount, false);
        builder = new AlertDialog.Builder(this);
        builder.setTitle("Result").setMessage("Your score is: " +
attempAnswer.getNumOfCorrectAnswer() + " out of " + questionsBank.size())
                .setCancelable(false)
                .setPositiveButton("Save", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
//              finish();
//              Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                        Toast.LENGTH_SHORT).show();
                        attempAnswer.setSaved(true);
                        saveAttemp();
                        Toast.makeText(getApplicationContext(), "You saved
this attemp. Let's Restart!", Toast.LENGTH_SHORT).show();
                        indexQuestion = 0;
                        correctAnswer=0;
                        historyList=new ArrayList<>();
                        openFragment();
                    }
                })
                .setNegativeButton("Ignore", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
//                    finish();
                    attempAnswer.setSaved(false);
                    saveAttemp();
                    Toast.makeText(getApplicationContext(), "You ignored it.
Let's Restart!", Toast.LENGTH_SHORT).show();
                    indexQuestion = 0;
                    correctAnswer=0;
                    historyList=new ArrayList<>();
                    openFragment();
                    }
                });
        builder.show();
    }
    public void saveAttemp(){
        storageManager.saveAttempAnswer(this,attempAnswer);
    }

    private void getTheAverage() {
        builder = new AlertDialog.Builder(this);
        if(attempAnswer.isEmpty()){
            builder.setTitle("Result").setMessage("You do not have any
previous attempt!")
                    .setCancelable(false)
                    .setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                finish();
//                Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                          Toast.LENGTH_SHORT).show();
                        }
                    });
        }else{
            String saveStatus;
            if(attempAnswer.isSaved()){
                saveStatus="Saved";
            }else{
                saveStatus="Not Save!";
            }
            builder.setTitle("Result").setMessage("Your correct answers are "
+ attempAnswer.getNumOfCorrectAnswer() + " in " +
attempAnswer.getNumOfAttemp() + " attempts | Status: "+ saveStatus)
                    .setCancelable(false)
                    .setPositiveButton("Save", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
```



```
//                finish();
//                Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                          Toast.LENGTH_SHORT).show();
                            if(!attempAnswer.isSaved() &&
historyList.size()!=questionsBank.size()) {
                                attempAnswer.setSaved(true);
                                Toast.makeText(getApplicationContext(), "You
saved this attemp", Toast.LENGTH_SHORT).show();
                            }else{
                                Toast.makeText(getApplicationContext(), "You
saved it before!", Toast.LENGTH_SHORT).show();
                            }
                        }
                    })
                    .setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                  finish();
                        }
                    });
        }
        builder.show();
    }

    private void checkAnswer(boolean answer) {
        question = questionsBank.get(indexQuestion);
        HistoryAnswer history;
        //snackbar begin
        String snackbarText = "";
        if (answer == question.getAnswer()) {
            correctAnswer++;
            snackbarText = "Correct";
        } else {
            snackbarText = "Uncorrect";
        }
        history = new HistoryAnswer(question, answer);
        historyList.add(history);
        System.out.println(question.toString() + " | ANSWER: " + answer + " |
index: " + indexQuestion + " | Res: " + snackbarText);

        View contextView = (View) findViewById(R.id.snackBar);
        contextView.getLayoutParams().height = 10;

        Snackbar mSnackbar = Snackbar.make(contextView, snackbarText,
Snackbar.LENGTH_LONG);
        View mView = mSnackbar.getView();

        TextView mTextView = (TextView)
mView.findViewById(R.id.snackbar_text);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR1)
            mTextView.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
        else
            mTextView.setGravity(Gravity.CENTER_HORIZONTAL);

mSnackbar.setBackgroundTint(getColor(R.color.purple_200)).setTextColor(getCol
or(R.color.black));
```

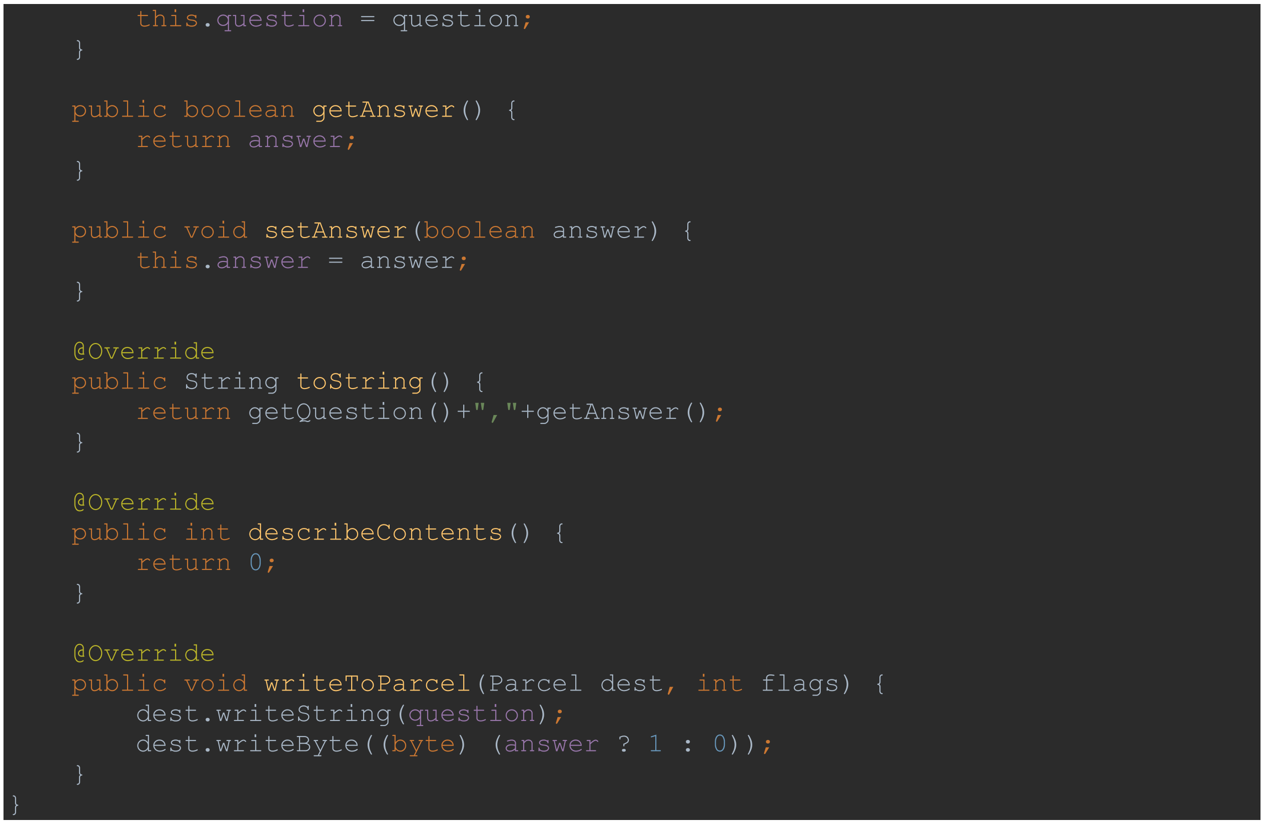


```
        mSnackbar.show();

        //snackbar end
    }

    private void openFragment() {
        Bundle bundle = new Bundle();
        ProgressBar progressBar = (ProgressBar)
findViewById(R.id.progressBar);

        if(questionsBank != null && questionsBank.size()!=0){
            question = questionsBank.get(indexQuestion);
            int progress = (int) ((double) (((double) (indexQuestion + 1)) /
questionsBank.size()) * 100);
            progressBar.setProgress(progress);
        }else{
            question=new Question("You do not have any question. Please
add!", true);
        }

        bundle.putParcelable("question", question);

        Random rnd = new Random();
        int color = Color.argb(255, rnd.nextInt(256), rnd.nextInt(256),
rnd.nextInt(256));
        bundle.putString("color", String.valueOf(color));

        Fragment fragment = new MainFragment();
        fragment.setArguments(bundle);
        System.out.println("Question Display: " + question.toString() + " |
index: " + indexQuestion);

        getSupportFragmentManager().beginTransaction().replace(R.id.frame,
fragment).commit();
    }


}
```

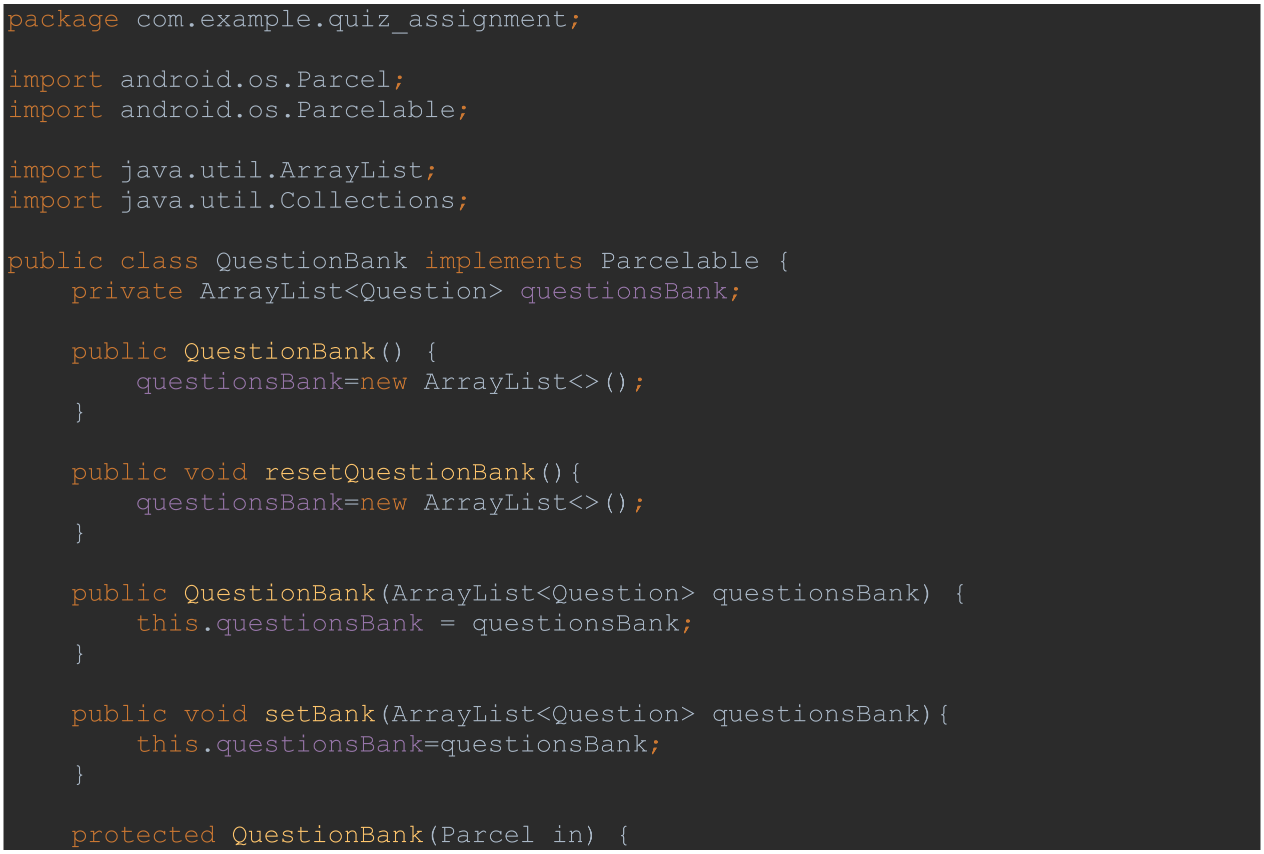


```
package com.example.quiz_assignment;

import static java.lang.Integer.parseInt;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;


public class MainFragment extends Fragment {

    TextView quest;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view=inflater.inflate(R.layout.fragment_main, container, false);

        quest = (TextView) view.findViewById(R.id.quest);

        if(getArguments() != null){
            int color=parseInt(getArguments().getString("color"));
            quest.setBackgroundColor(color);
            Question question = getArguments().getParcelable("question");
            quest.setText(question.getQuestion());
        }

        return view;
    }
}
```

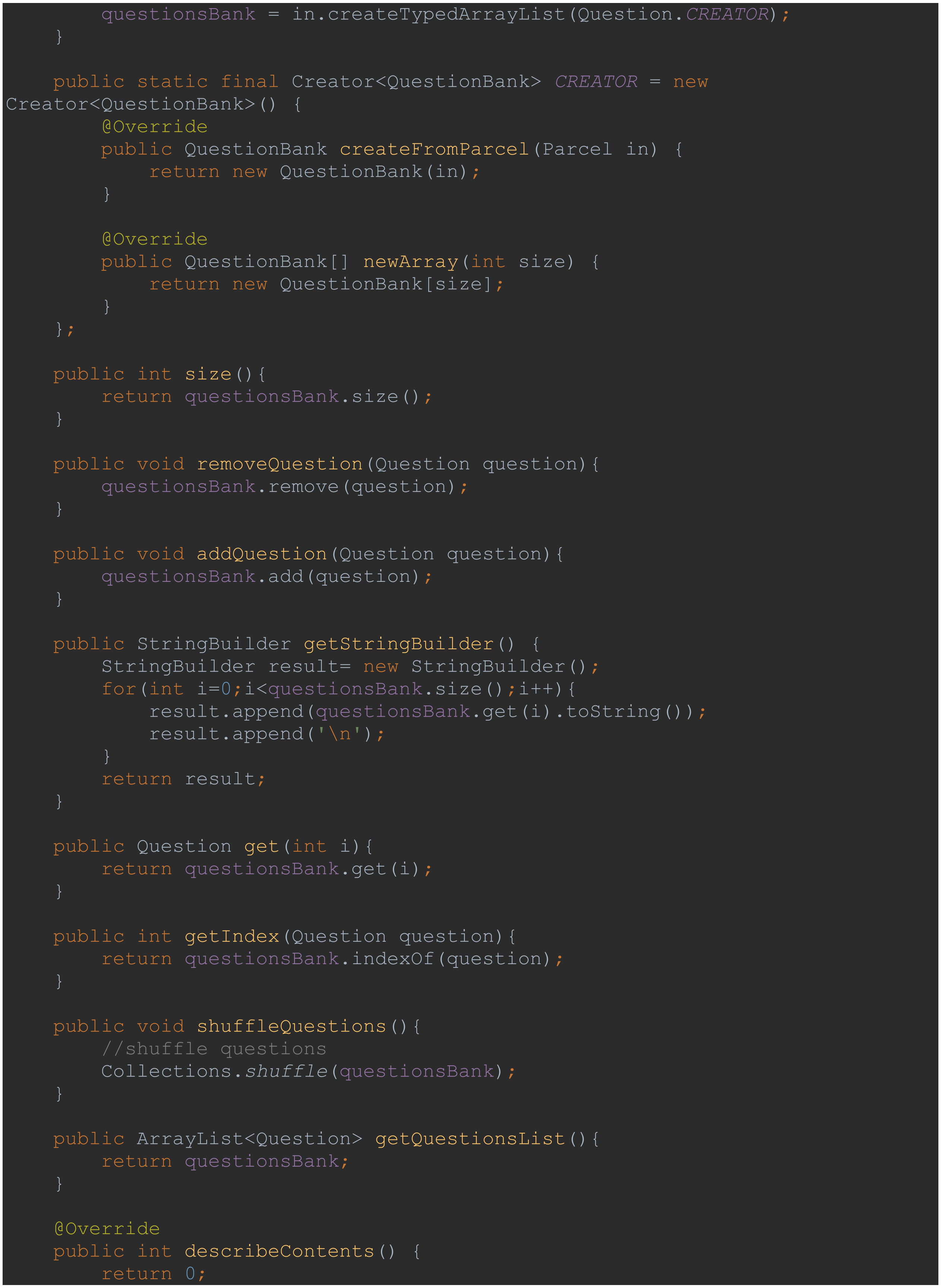
```
package com.example.quiz_assignment;

import android.os.Parcel;
import android.os.Parcelable;

public class Question implements Parcelable {
    private String question;
    private boolean answer;

    public Question(String question, boolean answer) {
        this.question = question;
        this.answer = answer;
    }

    protected Question(Parcel in) {
        question = in.readString();
        answer = in.readByte() != 0;
    }

    public static final Creator<Question> CREATOR = new Creator<Question>() {
        @Override
        public Question createFromParcel(Parcel in) {
            return new Question(in);
        }

        @Override
        public Question[] newArray(int size) {
            return new Question[size];
        }
    };

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
```



```
        this.question = question;
    }

    public boolean getAnswer() {
        return answer;
    }

    public void setAnswer(boolean answer) {
        this.answer = answer;
    }

    @Override
    public String toString() {
        return getQuestion()+","+getAnswer();
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(question);
        dest.writeByte((byte) (answer ? 1 : 0));
    }
}
```



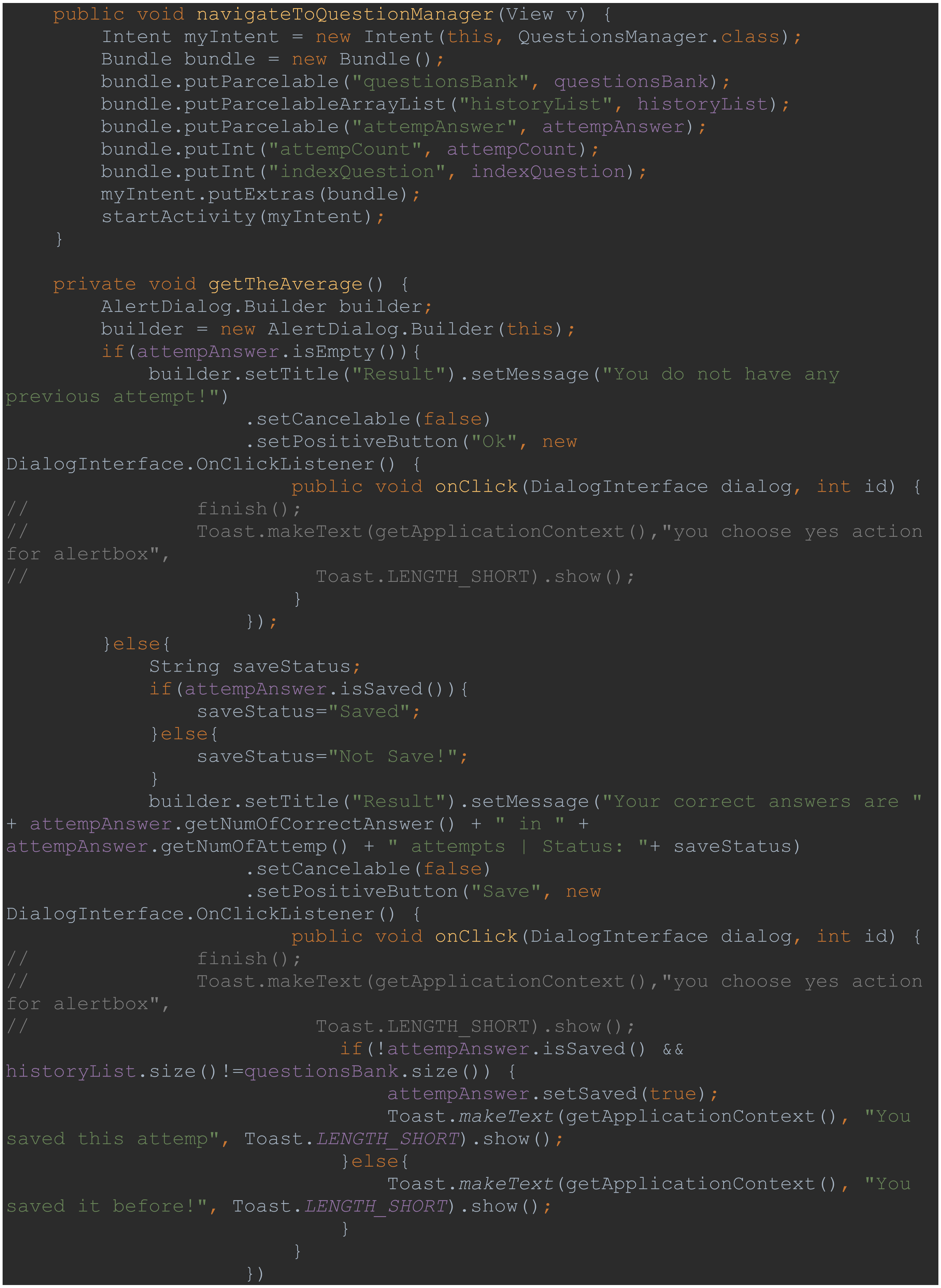
```
package com.example.quiz_assignment;

import android.os.Parcel;
import android.os.Parcelable;

import java.util.ArrayList;
import java.util.Collections;

public class QuestionBank implements Parcelable {
    private ArrayList<Question> questionsBank;

    public QuestionBank() {
        questionsBank=new ArrayList<>();
    }

    public void resetQuestionBank(){
        questionsBank=new ArrayList<>();
    }

    public QuestionBank(ArrayList<Question> questionsBank) {
        this.questionsBank = questionsBank;
    }

    public void setBank(ArrayList<Question> questionsBank){
        this.questionsBank=questionsBank;
    }

    protected QuestionBank(Parcel in) {
```



```
        questionsBank = in.createTypedArrayList(Question.CREATOR);
    }

    public static final Creator<QuestionBank> CREATOR = new
Creator<QuestionBank>() {
        @Override
        public QuestionBank createFromParcel(Parcel in) {
            return new QuestionBank(in);
        }

        @Override
        public QuestionBank[] newArray(int size) {
            return new QuestionBank[size];
        }
    };

    public int size(){
        return questionsBank.size();
    }

    public void removeQuestion(Question question){
        questionsBank.remove(question);
    }

    public void addQuestion(Question question){
        questionsBank.add(question);
    }

    public StringBuilder getStringBuilder() {
        StringBuilder result= new StringBuilder();
        for(int i=0;i<questionsBank.size();i++){
            result.append(questionsBank.get(i).toString());
            result.append('\n');
        }
        return result;
    }

    public Question get(int i){
        return questionsBank.get(i);
    }

    public int getIndex(Question question){
        return questionsBank.indexOf(question);
    }

    public void shuffleQuestions(){
        //shuffle questions
        Collections.shuffle(questionsBank);
    }

    public ArrayList<Question> getQuestionsList(){
        return questionsBank;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeTypedList(questionsBank);
    }
}
```

```
package com.example.quiz_assignment;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

import java.util.ArrayList;

public class QuestionList extends AppCompatActivity {
    private StorageManager storageManager=new StorageManager();
    ArrayList<HistoryAnswer> historyList;
    AttempAnswer attempAnswer = new AttempAnswer();
    QuestionBank questionsBank = new QuestionBank();
    int indexQuestion = 0;
    int attempCount = 0;

    RecyclerView recyclerList;
    ItemAdapter adapter;

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.app_menu, menu);
        return true;
    }

    @SuppressLint("NonConstantResourceId")
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                getTheAverage();
                return true;
            case R.id.item2:
                Toast.makeText(this, "Item2 was called",
Toast.LENGTH_SHORT).show();
```

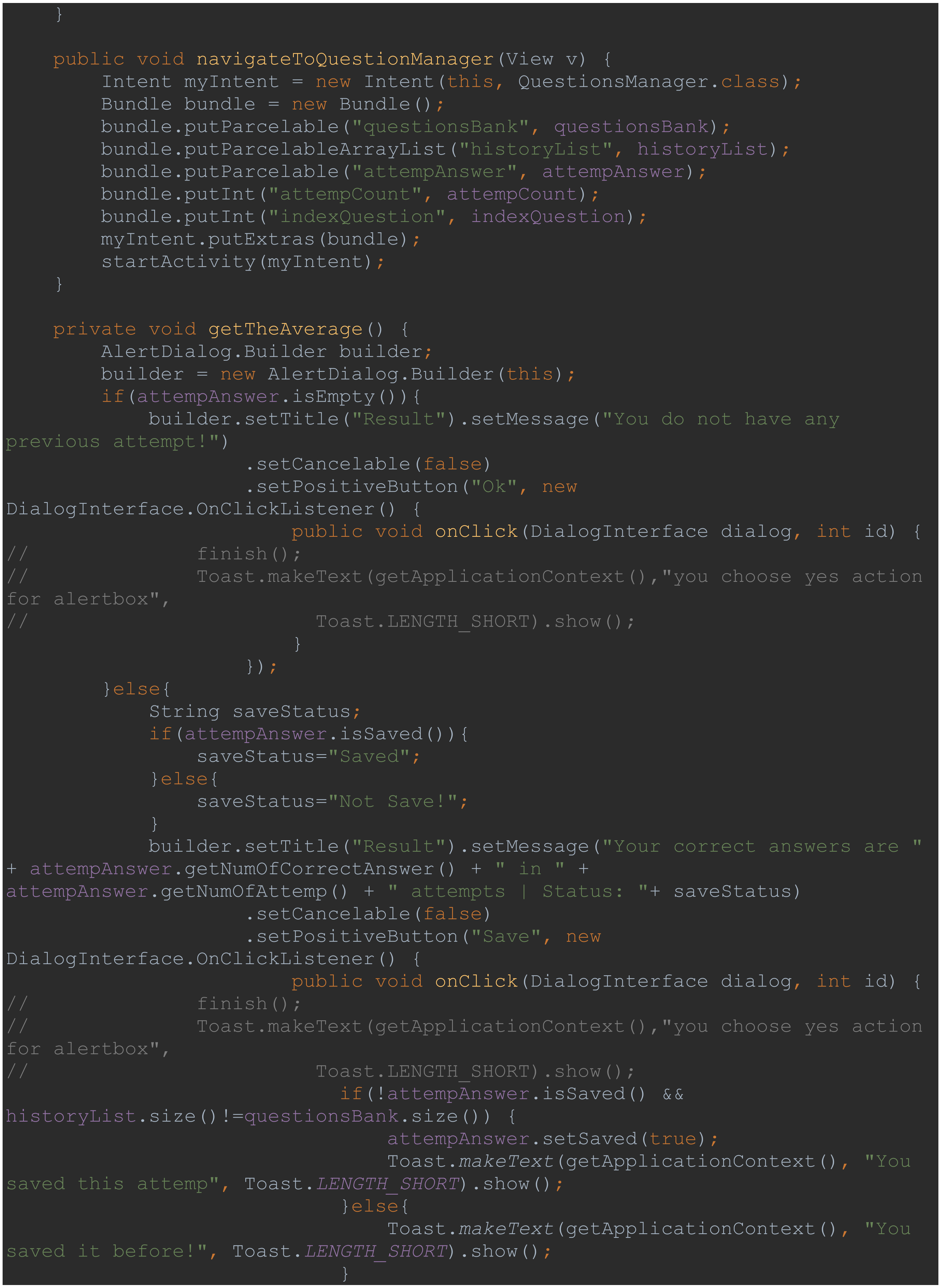
```
                navigateToQuestionListView(new View(this));
                return true;
            case R.id.item3:
                Toast.makeText(this, "Item3 was called",
Toast.LENGTH_SHORT).show();
                attempAnswer.resetNew();
                attempCount=0;
                storageManager.saveAttempAnswer(this,attempAnswer);

                return true;
            case R.id.item4:
                Toast.makeText(this, "Edit Question was called",
Toast.LENGTH_SHORT).show();
                navigateToQuestionManager(new View(this));
                return true;
            case R.id.item5:
                navigateToHomePage(new View(this));
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        outState.putParcelable("questionsBank", questionsBank);
        outState.putParcelableArrayList("historyList", historyList);
        outState.putParcelable("attempAnswer", attempAnswer);
        outState.putInt("attempCount", attempCount);
        outState.putInt("indexQuestion", indexQuestion);
        super.onSaveInstanceState(outState);
    }

    public void navigateToHomePage(View v){
        Intent myIntent = new Intent(this, MainActivity.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    public void navigateToQuestionListView(View v){
        Intent myIntent = new Intent(this, QuestionList.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }
```



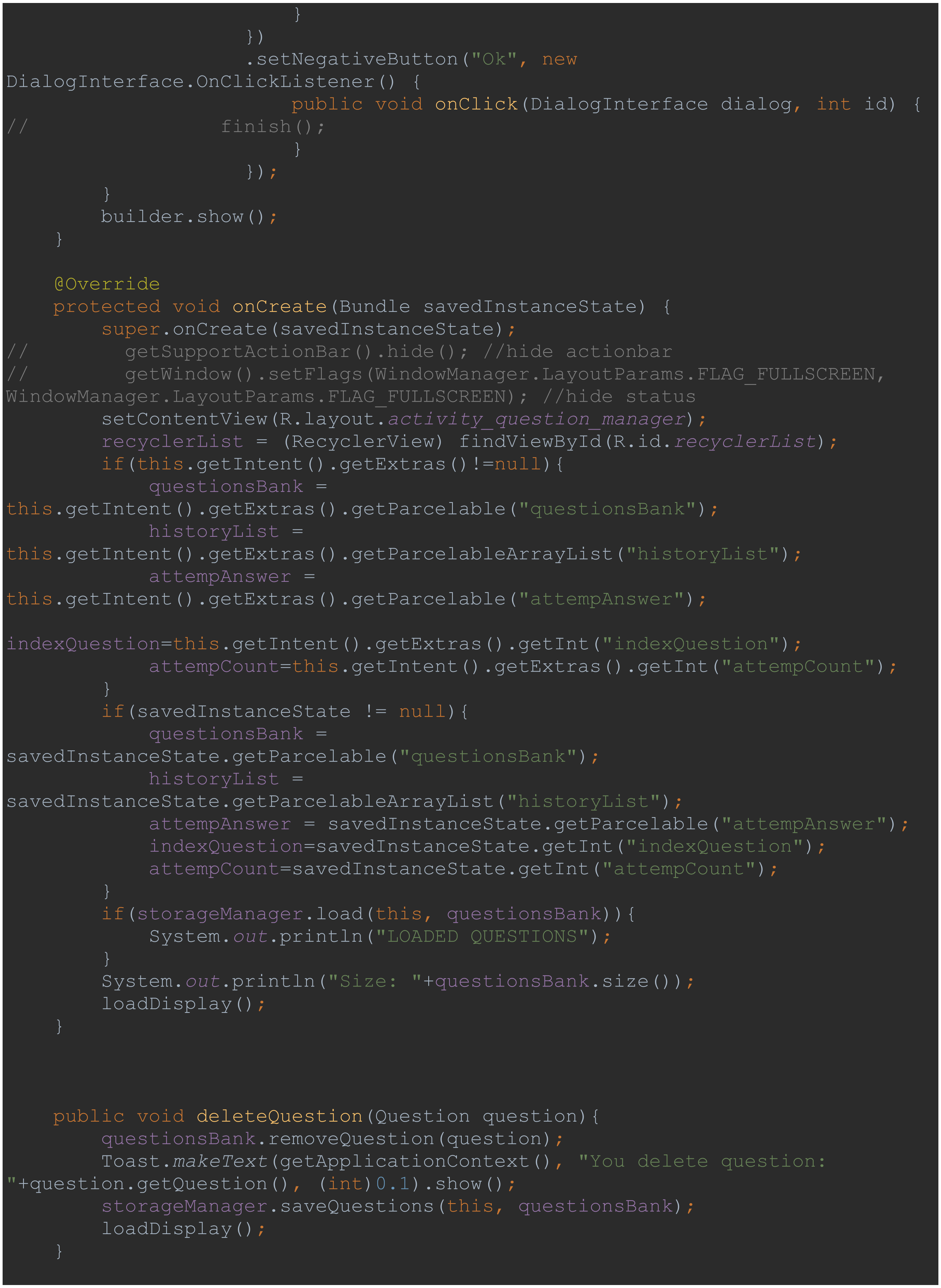
```
    public void navigateToQuestionManager(View v) {
        Intent myIntent = new Intent(this, QuestionsManager.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    private void getTheAverage() {
        AlertDialog.Builder builder;
        builder = new AlertDialog.Builder(this);
        if(attempAnswer.isEmpty()){
            builder.setTitle("Result").setMessage("You do not have any
previous attempt!")
                    .setCancelable(false)
                    .setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                finish();
//                Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                          Toast.LENGTH_SHORT).show();
                        }
                    });
        }else{
            String saveStatus;
            if(attempAnswer.isSaved()){
                saveStatus="Saved";
            }else{
                saveStatus="Not Save!";
            }
            builder.setTitle("Result").setMessage("Your correct answers are "
+ attempAnswer.getNumOfCorrectAnswer() + " in " +
attempAnswer.getNumOfAttemp() + " attempts | Status: "+ saveStatus)
                    .setCancelable(false)
                    .setPositiveButton("Save", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                finish();
//                Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                          Toast.LENGTH_SHORT).show();
                            if(!attempAnswer.isSaved() &&
historyList.size()!=questionsBank.size()) {
                                attempAnswer.setSaved(true);
                                Toast.makeText(getApplicationContext(), "You
saved this attemp", Toast.LENGTH_SHORT).show();
                            }else{
                                Toast.makeText(getApplicationContext(), "You
saved it before!", Toast.LENGTH_SHORT).show();
                            }
                        }
                    })
```
]]>

```
                    .setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                  finish();
                        }
                    });
        }
        builder.show();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//        getSupportActionBar().hide(); //hide actionbar
        setContentView(R.layout.activity_question_list);
        recyclerList = (RecyclerView) findViewById(R.id.recyclerView);
        if(this.getIntent().getExtras()!=null){
            questionsBank =
this.getIntent().getExtras().getParcelable("questionsBank");
            historyList =
this.getIntent().getExtras().getParcelableArrayList("historyList");
            attempAnswer =
this.getIntent().getExtras().getParcelable("attempAnswer");

indexQuestion=this.getIntent().getExtras().getInt("indexQuestion");
            attempCount=this.getIntent().getExtras().getInt("attempCount");
        }
        if(savedInstanceState != null){
            questionsBank =
savedInstanceState.getParcelable("questionsBank");
            historyList =
savedInstanceState.getParcelableArrayList("historyList");
            attempAnswer = savedInstanceState.getParcelable("attempAnswer");
            indexQuestion=savedInstanceState.getInt("indexQuestion");
            attempCount=savedInstanceState.getInt("attempCount");
        }
        if(storageManager.load(this, questionsBank)){
            System.out.println("LOADED QUESTIONS");
        }

        System.out.println("Size: "+questionsBank.size());

        adapter = new ItemAdapter(this, questionsBank.getQuestionsList(), new
ItemAdapter.OnItemClickListener() {
            @Override
            public void onItemClicked(Question question) {
                navigateToQuestionDetail(question);
            }
        });
        recyclerList.setAdapter(adapter);
        recyclerList.setLayoutManager(new LinearLayoutManager(this));
    }

    public void navigateToQuestionDetail(Question question){
        int indexQuestion=questionsBank.getIndex(question);
        Intent myIntent = new Intent(this, MainActivity.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }
}
```

```
package com.example.quiz_assignment;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

import java.util.ArrayList;

public class QuestionsManager extends AppCompatActivity {
    QuestionBank questionsBank;
    ArrayList<HistoryAnswer> historyList = new ArrayList<>();
    AttempAnswer attempAnswer = new AttempAnswer();
    int indexQuestion = 0;
    int attempCount = 0;

    RecyclerView recyclerList;
    ItemAdapter adapter;
    StorageManager storageManager=new StorageManager();
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.app_menu, menu);
        return true;
    }


    @SuppressLint("NonConstantResourceId")
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                getTheAverage();
                return true;
            case R.id.item2:
                Toast.makeText(this, "Item2 was called", 
Toast.LENGTH_SHORT).show();
                navigateToQuestionListView(new View(this));
                return true;
            case R.id.item3:
                Toast.makeText(this, "Item3 was called", 
Toast.LENGTH_SHORT).show();
                attempAnswer.resetNew();
                attempCount=0;
                storageManager.saveAttempAnswer(this,attempAnswer);

                return true;
            case R.id.item4:
                Toast.makeText(this, "Edit Question was called", 
Toast.LENGTH_SHORT).show();
                navigateToQuestionManager(new View(this));
                return true;
            case R.id.item5:
                navigateToHomePage(new View(this));
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        outState.putParcelable("questionsBank", questionsBank);
        outState.putParcelableArrayList("historyList", historyList);
        outState.putParcelable("attempAnswer", attempAnswer);
        outState.putInt("attempCount", attempCount);
        outState.putInt("indexQuestion", indexQuestion);
        super.onSaveInstanceState(outState);
    }

    public void navigateToHomePage(View v){
        Intent myIntent = new Intent(this, MainActivity.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    public void navigateToQuestionListView(View v){
        Intent myIntent = new Intent(this, QuestionList.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
```



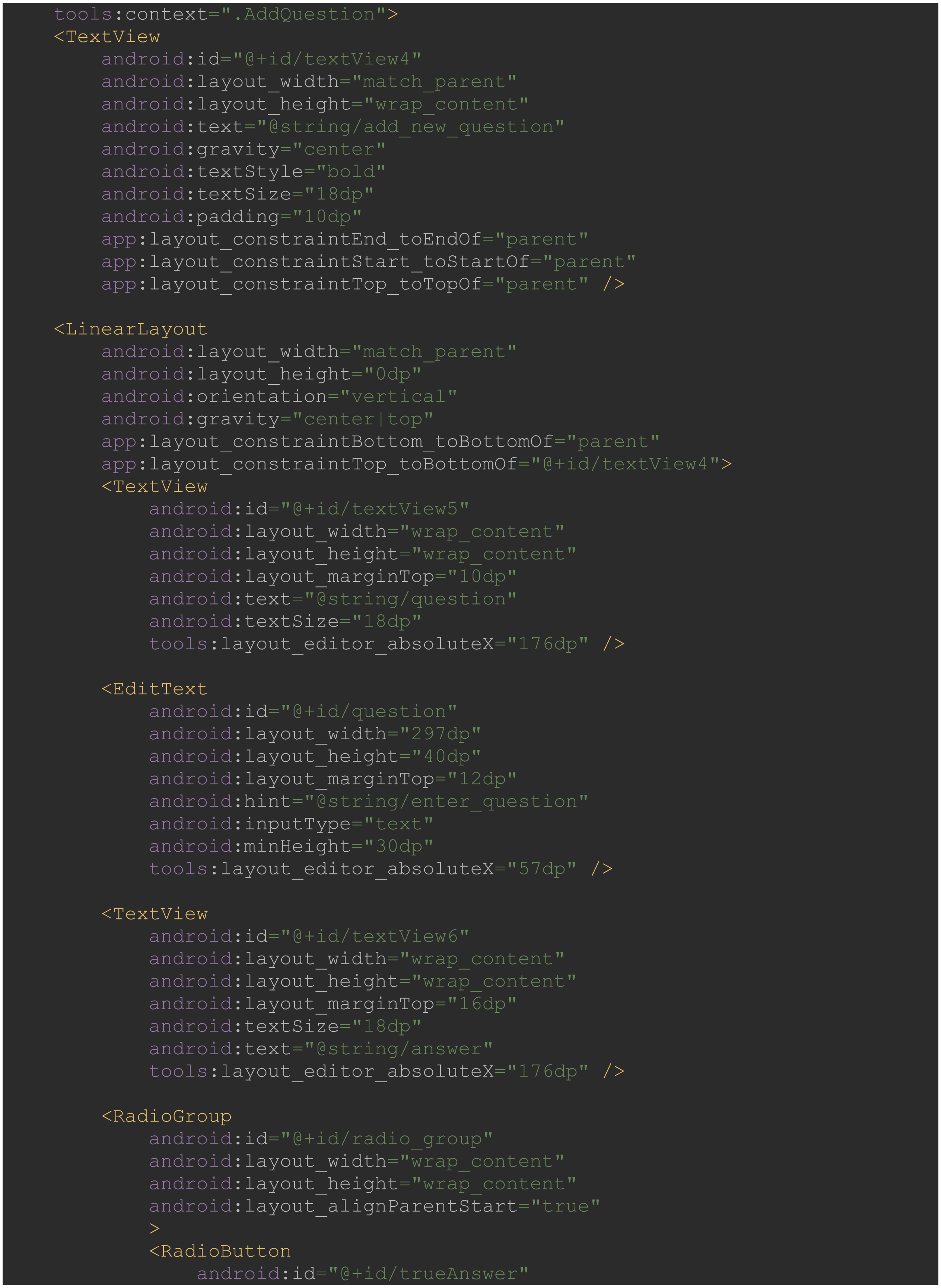
```
    }

    public void navigateToQuestionManager(View v) {
        Intent myIntent = new Intent(this, QuestionsManager.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    private void getTheAverage() {
        AlertDialog.Builder builder;
        builder = new AlertDialog.Builder(this);
        if(attempAnswer.isEmpty()){
            builder.setTitle("Result").setMessage("You do not have any
previous attempt!")
                    .setCancelable(false)
                    .setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//              finish();
//              Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                        Toast.LENGTH_SHORT).show();
                        }
                    });
        }else{
            String saveStatus;
            if(attempAnswer.isSaved()){
                saveStatus="Saved";
            }else{
                saveStatus="Not Save!";
            }
            builder.setTitle("Result").setMessage("Your correct answers are "
+ attempAnswer.getNumOfCorrectAnswer() + " in " +
attempAnswer.getNumOfAttemp() + " attempts | Status: "+ saveStatus)
                    .setCancelable(false)
                    .setPositiveButton("Save", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//              finish();
//              Toast.makeText(getApplicationContext(),"you choose yes action
for alertbox",
//                        Toast.LENGTH_SHORT).show();
                            if(!attempAnswer.isSaved() &&
historyList.size()!=questionsBank.size()) {
                                attempAnswer.setSaved(true);
                                Toast.makeText(getApplicationContext(), "You
saved this attemp", Toast.LENGTH_SHORT).show();
                            }else{
                                Toast.makeText(getApplicationContext(), "You
saved it before!", Toast.LENGTH_SHORT).show();
                            }
```



```
                        }
                    })
                    .setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
//                  finish();
                        }
                    });
        }
        builder.show();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//        getSupportActionBar().hide(); //hide actionbar
//        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN); //hide status
        setContentView(R.layout.activity_question_manager);
        recyclerList = (RecyclerView) findViewById(R.id.recyclerList);
        if(this.getIntent().getExtras()!=null){
            questionsBank =
this.getIntent().getExtras().getParcelable("questionsBank");
            historyList =
this.getIntent().getExtras().getParcelableArrayList("historyList");
            attempAnswer =
this.getIntent().getExtras().getParcelable("attempAnswer");

indexQuestion=this.getIntent().getExtras().getInt("indexQuestion");
            attempCount=this.getIntent().getExtras().getInt("attempCount");
        }
        if(savedInstanceState != null){
            questionsBank =
savedInstanceState.getParcelable("questionsBank");
            historyList =
savedInstanceState.getParcelableArrayList("historyList");
            attempAnswer = savedInstanceState.getParcelable("attempAnswer");
            indexQuestion=savedInstanceState.getInt("indexQuestion");
            attempCount=savedInstanceState.getInt("attempCount");
        }
        if(storageManager.load(this, questionsBank)){
            System.out.println("LOADED QUESTIONS");
        }
        System.out.println("Size: "+questionsBank.size());
        loadDisplay();
    }



    public void deleteQuestion(Question question){
        questionsBank.removeQuestion(question);
        Toast.makeText(getApplicationContext(), "You delete question:
"+question.getQuestion(), (int)0.1).show();
        storageManager.saveQuestions(this, questionsBank);
        loadDisplay();
    }
    public void addNewQuestion(View v){
        System.out.println("CREATE NEW QUESTION");
        Intent myIntent = new Intent(this, AddQuestion.class);
        Bundle bundle = new Bundle();
        bundle.putParcelable("questionsBank", questionsBank);
        bundle.putParcelableArrayList("historyList", historyList);
        bundle.putParcelable("attempAnswer", attempAnswer);
        bundle.putInt("attempCount", attempCount);
        bundle.putInt("indexQuestion", indexQuestion);
        myIntent.putExtras(bundle);
        startActivity(myIntent);
    }

    public void loadDisplay(){
        adapter = new ItemAdapter(this, questionsBank.getQuestionsList(), new
ItemAdapter.OnItemClickListener() {
            @Override
            public void onItemClicked(Question question) {
                deleteQuestion(question);
            }
        });
        recyclerList.setAdapter(adapter);
        recyclerList.setLayoutManager(new LinearLayoutManager(this));
    }
}
```

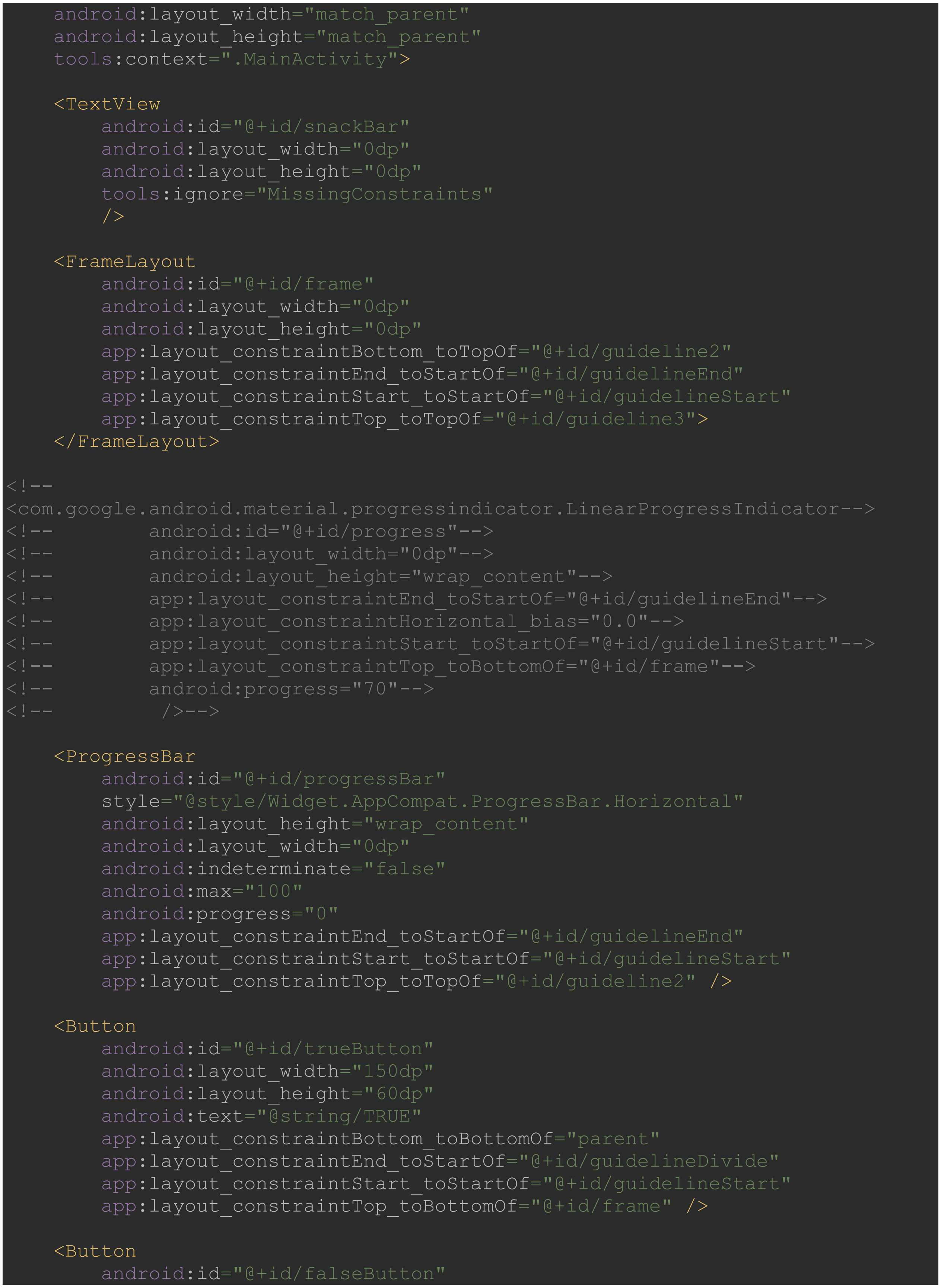
```
package com.example.quiz_assignment;

import static android.content.Context.MODE_PRIVATE;

import android.app.Activity;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;

public class StorageManager {
    private static final String QUESTION_FILE_NAME = "questions.csv";
    private static final String HISTORY_ANSWER_FILE_NAME = "answer.csv";

    public void saveAttempAnswer(Activity activity, AttempAnswer attempList)
{
        FileOutputStream fos = null;
        try {
            fos = activity.openFileOutput(HISTORY_ANSWER_FILE_NAME,
MODE_PRIVATE);
            StringBuilder sb = new StringBuilder();
            sb.append(attempList.getSb());
            fos.write(sb.toString().getBytes());
            System.out.println("SAVED " + HISTORY_ANSWER_FILE_NAME);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (fos != null) {
                try {
                    fos.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public boolean loadAttempAnswer(Activity activity,AttempAnswer 
attempAnswer) {
        String line = "";
        FileInputStream fis=null;
        try{
            fis= activity.openFileInput(HISTORY_ANSWER_FILE_NAME);
            InputStreamReader isr=new InputStreamReader(fis);
            BufferedReader br=new BufferedReader(isr);
            StringBuilder sb = new StringBuilder();
            while((line = br.readLine())!=null){
                String[] data = line.split(",");
                String correct = data[0];
                String question = data[1];
                String attemp = data[2];
                String isSaved = data[3];
                boolean ans;
                if (isSaved == "True") {
                    ans = true;
                } else {
                    ans = false;
                }
                attempAnswer = new AttempAnswer(Integer.parseInt(correct), 
Integer.parseInt(question), Integer.parseInt(attemp), ans);
            }
            return true;
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            return false;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
    }

    public void saveQuestions(Activity activity, QuestionBank questionsBank) 
{
        FileOutputStream fos = null;
        try {
            fos = activity.openFileOutput(QUESTION_FILE_NAME, MODE_PRIVATE);
            StringBuilder sb = new StringBuilder();
            sb = questionsBank.getStringBuilder();
            fos.write(sb.toString().getBytes());
            System.out.println("SAVED " + QUESTION_FILE_NAME);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (fos != null) {
                try {
                    fos.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public boolean load(Activity activity,QuestionBank questionsBank) {
        String line = "";
        questionsBank.resetQuestionBank();
        ArrayList<Question> questionsList = new ArrayList<>();

        FileInputStream fis=null;
        try{
            fis= activity.openFileInput(QUESTION_FILE_NAME);
            InputStreamReader isr=new InputStreamReader(fis);
            BufferedReader br=new BufferedReader(isr);
            StringBuilder sb = new StringBuilder();
            while((line = br.readLine())!=null){
                String[] data = line.split(",");
                String question = data[0];
                String answer = data[1];
                boolean ans;
                if (answer.equals("true")) {
                    ans = true;
                } else {
                    ans = false;
                }
                Question tempQuestion = new Question(question, ans);
                questionsList.add(tempQuestion);
            }
            questionsBank.setBank(questionsList);
            return true;
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            return false;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```



```
    tools:context=".AddQuestion">
    <TextView
        android:id="@+id/textView4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/add_new_question"
        android:gravity="center"
        android:textStyle="bold"
        android:textSize="18dp"
        android:padding="10dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:orientation="vertical"
        android:gravity="center|top"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView4">
        <TextView
            android:id="@+id/textView5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:text="@string/question"
            android:textSize="18dp"
            tools:layout_editor_absoluteX="176dp" />

        <EditText
            android:id="@+id/question"
            android:layout_width="297dp"
            android:layout_height="40dp"
            android:layout_marginTop="12dp"
            android:hint="@string/enter_question"
            android:inputType="text"
            android:minHeight="30dp"
            tools:layout_editor_absoluteX="57dp" />

        <TextView
            android:id="@+id/textView6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:textSize="18dp"
            android:text="@string/answer"
            tools:layout_editor_absoluteX="176dp" />

        <RadioGroup
            android:id="@+id/radio_group"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            >
            <RadioButton
                android:id="@+id/trueAnswer"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:onClick="getAnswerValue"
                android:text="@string/True" />

            <RadioButton
                android:id="@+id/falseAnswer"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:onClick="getAnswerValue"
                android:text="@string/False" />
        </RadioGroup>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:backgroundTint="#4073AF"
            android:onClick="onSubmit"
            android:text="@string/submit"
            />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:gravity="center"
        android:orientation="horizontal"
        android:onClick="navigateToQuestionListView"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="8dp"
            android:layout_weight="0.5"
            android:text="@string/see_question_list"
            android:textSize="20dp" />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.5"
            android:src="@drawable/ic_list" />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```



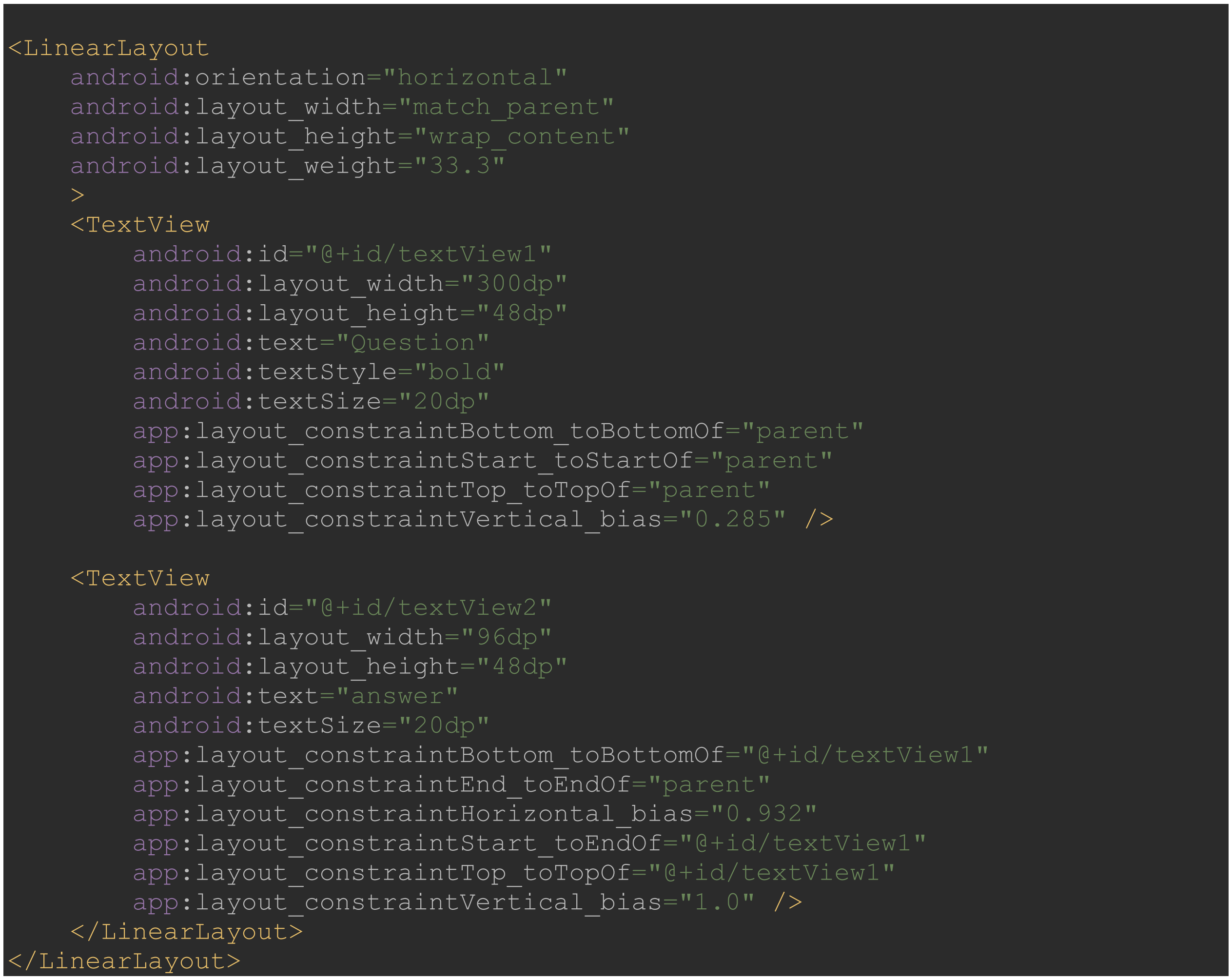
```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/snackBar"
        android:layout_width="0dp"
        android:layout_height="0dp"
        tools:ignore="MissingConstraints"
        />

    <FrameLayout
        android:id="@+id/frame"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@+id/guideline2"
        app:layout_constraintEnd_toStartOf="@+id/guidelineEnd"
        app:layout_constraintStart_toStartOf="@+id/guidelineStart"
        app:layout_constraintTop_toTopOf="@+id/guideline3">
    </FrameLayout>

<!--
<com.google.android.material.progressindicator.LinearProgressIndicator-->
<!--        android:id="@+id/progress"-->
<!--        android:layout_width="0dp"-->
<!--        android:layout_height="wrap_content"-->
<!--        app:layout_constraintEnd_toStartOf="@+id/guidelineEnd"-->
<!--        app:layout_constraintHorizontal_bias="0.0"-->
<!--        app:layout_constraintStart_toStartOf="@+id/guidelineStart"-->
<!--        app:layout_constraintTop_toBottomOf="@+id/frame"-->
<!--        android:progress="70"-->
<!--         />-->

    <ProgressBar
        android:id="@+id/progressBar"
        style="@style/Widget.AppCompat.ProgressBar.Horizontal"
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:indeterminate="false"
        android:max="100"
        android:progress="0"
        app:layout_constraintEnd_toStartOf="@+id/guidelineEnd"
        app:layout_constraintStart_toStartOf="@+id/guidelineStart"
        app:layout_constraintTop_toTopOf="@+id/guideline2" />

    <Button
        android:id="@+id/trueButton"
        android:layout_width="150dp"
        android:layout_height="60dp"
        android:text="@string/TRUE"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/guidelineDivide"
        app:layout_constraintStart_toStartOf="@+id/guidelineStart"
        app:layout_constraintTop_toBottomOf="@+id/frame" />

    <Button
        android:id="@+id/falseButton"
        android:layout_width="150dp"
        android:layout_height="60dp"
        android:text="@string/FALSE"
        app:layout_constraintBottom_toBottomOf="@+id/trueButton"
        app:layout_constraintEnd_toStartOf="@+id/guidelineEnd"
        app:layout_constraintStart_toStartOf="@+id/guidelineDivide"
        app:layout_constraintTop_toTopOf="@+id/trueButton"
        app:layout_constraintVertical_bias="0.0" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.02" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guidelineEnd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.96" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guidelineDivide"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.5" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guidelineStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.04" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.7" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".QuestionList">
    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/question_list"
        android:textStyle="bold"
        android:textSize="25dp"
        android:gravity="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintTop_toBottomOf="@+id/textView3"
        tools:layout_editor_absoluteX="0dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".QuestionsManager">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#949398"
        android:gravity="center"
        android:text="@string/question_bank"
        android:textColor="#F4DF4E"
        android:textSize="25dp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:gravity="center"
        android:orientation="horizontal"
        android:onClick="addNewQuestion"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="8dp"
            android:layout_weight="0.5"
            android:text="@string/add_new_question"
            android:textSize="20dp" />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.5"
            android:src="@drawable/ic_add_icon" />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/quest"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Question"
        android:layout_gravity="center"
        android:background="#005000"
        android:textColor="#ffffff"
        android:textSize="30dp"
        android:padding="20dp"
        />

</FrameLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:weightSum="100">
```



```xml
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="33.3"
    >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="300dp"
        android:layout_height="48dp"
        android:text="Question"
        android:textStyle="bold"
        android:textSize="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.285" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="96dp"
        android:layout_height="48dp"
        android:text="answer"
        android:textSize="20dp"
        app:layout_constraintBottom_toBottomOf="@+id/textView1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.932"
        app:layout_constraintStart_toEndOf="@+id/textView1"
        app:layout_constraintTop_toTopOf="@+id/textView1"
        app:layout_constraintVertical_bias="1.0" />
    </LinearLayout>
</LinearLayout>
```