

AboutActivity.java

```
package com.art.code.cplusplus.ui;

import
android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import com.art.code.cplusplus.R;

public class AboutActivity extends
AppCompatActivity {

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_about);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    }

    @Override

    public boolean onSupportNavigateUp() {

        finish();

        return true;

    }

}
```

AdminActivity.java

```
package com.art.code.cplusplus.ui;

import android.content.Intent;
```

```
import android.os.Bundle;

import android.os.Handler;

import android.support.annotation.NonNull;

import
android.support.design.widget.BottomNavigation
View;

import android.support.v4.app.Fragment;

import
android.support.v4.app.FragmentTransaction
;

import
android.support.v7.app.AppCompatActivity;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.widget.Toast;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.Question;

import
com.art.code.cplusplus.Model.Sample;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

public class AdminActivity extends
AppCompatActivity {

    BottomNavigationView
bottomNavigationView;
```

```

FragmentTransaction transaction;
selectedFragment =
SampleAdminFragment.newInstance();

@Override
break;

protected void onCreate(Bundle
case R.id.questions_menu:
savedInstanceState) {
selectedFragment =
super.onCreate(savedInstanceState);
QuestionAdminFragment.newInstance();

setContentView(R.layout.activity_admin);
break;

transaction =
}

getSupportFragmentManager().beginTransaction();
FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();

bottomNavigationView =
transaction.replace(R.id.admin_container,
selectedFragment);

(bottomNavigationView)
transaction.commit();

findViewById(R.id.nav_bottom);
return true;

bottomNavigationView.setOnNavigationItemSelectedListener(new
}

BottomNavigationView.OnNavigationItemSelectedListener {
});

@Override
setDefaultFragmentUsed();

public boolean
}

onNavigationItemSelectedListener(@NonNull
private void setDefaultFragmentUsed() {

MenuItem menuItem) {
transaction.add(R.id.admin_container,
ChapterAdminFragment.newInstance());

selectedFragment =
transaction.commit();

ChapterAdminFragment.newInstance();
}

break;
public void showSubChapterof(Chapter

case R.id.samples_menu:
chapter) {

```

```

        Fragment subChapterFragment =
SubChapterAdminFragment.newInstance(ch
apter);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransa
ction();

transaction.addToBackStack("chapters").repl
ace(R.id.admin_container,
subChapterFragment);

        transaction.commit();
    }

    public void showSubChapterDetail(SubChapter
subChapter) {

        DetailSubChapterFragment
detailSubChapterFragment = new
DetailSubChapterFragment();

detailSubChapterFragment.setSubChapter(s
ubChapter);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransa
ction();

transaction.addToBackStack("subchapter").r
eplace(R.id.admin_container,
detailSubChapterFragment);

        transaction.commit();
    }

    public void showSampleDetail(Sample
sample) {

        DetailSampleFragment sampleDetail =
new DetailSampleFragment();

        sampleDetail.setSample(sample);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransa
ction();

transaction.addToBackStack("sample").repla
ce(R.id.admin_container, sampleDetail);

        transaction.commit();
    }

    public void showQuestionsOf(Chapter
chapter) {

        Fragment listOfQuestions =
ListOfQuestionAdminFragment.newInstanc
e(chapter);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransa
ction();

transaction.addToBackStack("chapterList").r
eplace(R.id.admin_container,
listOfQuestions);

        transaction.commit();
    }

    public void showQuestionDetail(Question
question) {

        DetailQuestionFragment
detailQuestionFragment = new
DetailQuestionFragment();

```

```

detailQuestionFragment.setQuestion(question);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();

transaction.addToBackStack("question").replace(R.id.admin_container,
detailQuestionFragment);

        transaction.commit();
    }

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        MenuInflater inflater = new
MenuInflater(this);

        inflater.inflate(R.menu.main_menu,
menu);

        return true;
    }

    @Override

    public boolean onOptionsItemSelected(Menu item) {

        switch (item.getItemId()) {

            case R.id.logout:

LoginActivity.sharedPrefManager.logout();

                startActivity(new Intent(this,
LoginActivity.class));

```

```

            finish();

            break;

            case R.id.exit:

                System.exit(0);

            break;

            default:

                break;

        }

        return
super.onOptionsItemSelected(item);
    }
}

```

ChapterAdminFragment.java

```

package com.art.code.cplusplus.ui;

import android.app.AlertDialog;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingAction
nButton;

import android.support.v4.app.Fragment;

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

```

```

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickListener;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.ChapterAdminViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerAdapter;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseReference;

import
com.google.firebase.database.FirebaseDatabase;

import
com.google.firebase.database.ValueEventListener;

public class ChapterAdminFragment extends
Fragment implements
View.OnClickListener {

    View adminChapterFragment;

    FloatingActionButton fabTambahChapter;

    RecyclerView listOfChapters;

    RecyclerView.LayoutManager
layoutManager;

    TextView title;

    // ImageButton backBtn;

    FirebaseRecyclerAdapter<Chapter,
ChapterAdminViewHolder>
mChapterAdapter;

    FirebaseDatabase mDatabase;

    DatabaseReference chaptersReff;

```

```

DatabaseReference subChapterReff;

public static ChapterAdminFragment
newInstance() {

    ChapterAdminFragment
chapterAdminFragment = new
ChapterAdminFragment();

    return chapterAdminFragment;
}

@Override

public void onCreate(@Nullable Bundle
savedInstanceState) {

    super.onCreate(savedInstanceState);

    mDatabase =
DataRepository.getDBInstance();

    chaptersReff =
mDatabase.getReference("Chapters");

    subChapterReff =
mDatabase.getReference("SubChapters");
}

@Nullable

@Override

public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,

Bundle
savedInstanceState) {

    adminChapterFragment =
inflater.inflate(R.layout.fragment_chapter_a
dmin, container, false);

    fabTambahChapter =
(FloatingActionButton)
adminChapterFragment.findViewById(R.id.
fab_tambah_chapter);

    fabTambahChapter.setOnClickListener(this)
;

    listOfChapters = (RecyclerView)
adminChapterFragment.findViewById(R.id.
admin_chapter_container);

    title = (TextView)
adminChapterFragment.findViewById(R.id.
fragment_title);

    title.setText("Chapters");

    layoutManager = new
LinearLayoutManager(container.getContext(
));

    listOfChapters.setLayoutManager(layoutMa
nager);

    loadChapters();

    return adminChapterFragment;
}

private void loadChapters() {

    mChapterAdapter = new
FirebaseRecyclerAdapter<Chapter,
ChapterAdminViewHolder>(

    Chapter.class,

    R.layout.admin_item,

    ChapterAdminViewHolder.class,

```

```

        chaptersReff
    ) {
        @Override
        protected void
        populateViewHolder(ChapterAdminViewHo
        lder chapterAdminViewHolder, final
        Chapter chapter, int position) {
            String nameInList =
            chapter.getChapter() + ". " +
            chapter.getTitle();

            chapterAdminViewHolder.chapter_name.set
            Text(nameInList);

            chapterAdminViewHolder.edit_chapter_btn.
            setOnClickListener(new
            View.OnClickListener() {
                @Override
                public void onClick(View view)
            {
                showEditChapterDialog(chapter);
            }
        });

            chapterAdminViewHolder.del_chapter_btn.s
            etOnClickListener(new
            View.OnClickListener() {
                @Override
                public void onClick(View view)
            {
                showDeleteDialog(chapter);
            }
        });
    }

    chapterAdminViewHolder.setItemClickListe
    ner(new ItemClickListener() {
        @Override
        public void onClick(View view,
        int position, boolean isLongClick) {
            if
            (getLifecycle().getCurrentState()
            .isAtLeast(Lifecycle.State.STARTED)) {
                ((AdminActivity)
                getActivity()).showSubChapterof(chapter);
            }
        }
    });

    mChapterAdapter.notifyDataSetChanged();

    listOfChapters.setAdapter(mChapterAdapter
    );
}

private void showEditChapterDialog(final
Chapter chapter) {
    final Chapter editedChapter = chapter;

```



```

        }

        String chapterName = "ch" +
chapterNumber;

chaptersReff.child(chapterName).child("title
").setValue(newTitle);

mChapterAdapter.notifyDataSetChanged();

        editChapterDialog.dismiss();
    }
}

});

}

private void showDeleteDialog(Chapter
chapter) {

    final Chapter deletedchapter = chapter;

    final AlertDialog yesNoDialog = new
AlertDialog.Builder(getActivity())

        .setTitle("Delete " +
deletedchapter.getTitle())

        .setMessage("Deleting Chapter
also effect the sub chapters")

        .setNegativeButton(R.string.cancel, null)

        .setPositiveButton("Yes", null)

        .setIcon(R.drawable.ic_warning_black_24dp
)

        .create();

```

```

yesNoDialog.show();

yesNoDialog.getButton(AlertDialog.BUTT
ON_NEGATIVE).setOnClickListener(new
View.OnClickListener() {

    @Override

    public void onClick(View view) {

        yesNoDialog.dismiss();

    }

});

yesNoDialog.getButton(AlertDialog.BUTT
ON_POSITIVE).setOnClickListener(new
View.OnClickListener() {

    @Override

    public void onClick(View view) {

        String number =
String.valueOf(deletedchapter.getChapter());

        if (number.length() < 2) {

            number = '0' + number;

        }

        final String childNameChapter =
"ch" + number;

chaptersReff.child(childNameChapter).remo
veValue();

subChapterReff.addListenerForSingleValue
Event(new ValueEventListener() {

```

```

        @Override
        public void
onDataChange(DataSnapshot dataSnapshot)
{
        for (DataSnapshot ds :
dataSnapshot.getChildren()) {
                SubChapter onIterSub =
ds.getValue(SubChapter.class);
                Log.d("TAG",
"onDataChange: " + onIterSub.getChapter()
+ "; " + deletedchapter.getChapter());
                if (onIterSub.getChapter()
== deletedchapter.getChapter()) {
subChapterReff.child(ds.getKey()).removeV
alue();
                }
        }
}
@Override
public void
onCancelled(DatabaseError databaseError) {
}
});
mChapterAdapter.notifyDataSetChanged();
        yesNoDialog.dismiss();
}
});

```

```

}
@Override
public void onClick(View view) {
        if (view.getId() ==
R.id.fab_tambah_chapter) {
                LayoutInflater inflater =
this.getLayoutInflater();
                final View dialogView =
inflater.inflate(R.layout.add_edit_chapter_la
yout, null);
                final EditText new_chapter_title =
dialogView.findViewById(R.id.chapter_title
);
                final AlertDialog addChapterDialog
= new AlertDialog.Builder(getActivity())
                .setTitle(R.string.add_chapter)
                .setPositiveButton(R.string.save, null)
                .setNegativeButton(R.string.cancel, null)
                .setView(dialogView)
                .setIcon(R.drawable.ic_playlist_add_black_
24dp)
                .create();
                addChapterDialog.getWindow().setSoftInput
Mode(WindowManager.LayoutParams.SOF
T_INPUT_ADJUST_RESIZE);
                addChapterDialog.show();

```

```

addChapterDialog.getButton(android.support
t.v7.app.AlertDialog.BUTTON_POSITIVE)
.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View
view) {
            final String title =
new_chapter_title.getText().toString();
            if (title.isEmpty()) {
                new_chapter_title.setError("Chapter
title
can't be empty");
                return;
            } else {
                final Chapter
newChapter = new Chapter();
                chaptersReff.addListenerForSingleValueEve
nt(new ValueEventListener() {
                    @Override
                    public void
onDataChange(DataSnapshot dataSnapshot)
{
                        int i = 1;
                        int chapterCount =
i;
                        for (DataSnapshot
ds : dataSnapshot.getChildren()) {
                            Chapter
onInterCh = ds.getValue(Chapter.class);
                            if
(onInterCh.getChapter() == i) {
                                i++;
                                chapterCount
= i;
                            } else {
                                chapterCount
= i;
                            }
                            Log.d("TAG",
"onDataChange: " + chapterCount);
                            String chapterNum
= String.valueOf(chapterCount);
                            if
(chapterNum.length() < 2) {
                                chapterNum = '0'
+ chapterNum;
                            }
                            newChapter.setChapter(chapterCount);
                            newChapter.setTitle(title);
                            chaptersReff.child("ch"
+
chapterNum).setValue(newChapter);
                        }
                    }
                }
            }
        }
    }
);

```

```

        @Override
        public void
onCancelled(DatabaseError databaseError) {
            }
        });

mChapterAdapter.notifyDataSetChanged();

addChapterDialog.dismiss();
        }
    }
});

addChapterDialog.getButton(android.support
v7.app.AlertDialog.BUTTON_NEGATIV
E).setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View
view) {

addChapterDialog.dismiss();
            }
        });
    }
}
}

```

DetailQuestionFragment.java

```

package com.art.code.cplusplus.ui;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Model.Question;

import com.art.code.cplusplus.R;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

```

```

public class DetailQuestionFragment
extends Fragment implements
View.OnClickListener {

    private Question mQuestion;

    private FirebaseDatabase mDatabase;

    private DatabaseReference questionRef;

    private View fragmentView;

    private EditText question;

    private EditText correct_answer;

    private EditText other_answer_2;

    private EditText other_answer_1;

    private EditText other_answer_3;

    private Button save_btn;

    @Override

    public View onCreateView(LayoutInflater
inflater, ViewGroup container,

        Bundle

savedInstanceState) {

        fragmentView =
inflater.inflate(R.layout.fragment_detail_que
stion, container, false);

        question = (EditText)
fragmentView.findViewById(R.id.question_
in_fragment);

        correct_answer = (EditText)
fragmentView.findViewById(R.id.correct_a
nswer_in_fragment);

        other_answer_1 = (EditText)
fragmentView.findViewById(R.id.other_ans
wer_in_fragment_1);

        other_answer_2 = (EditText)
fragmentView.findViewById(R.id.other_ans
wer_in_fragment_2);

        other_answer_3 = (EditText)
fragmentView.findViewById(R.id.other_ans
wer_in_fragment_3);

        save_btn = (Button)
fragmentView.findViewById(R.id.save_que
stion_in_fragment);

        question.setText(mQuestion.getQuestion().is
Empty() ? "" : mQuestion.getQuestion());

        correct_answer.setText(mQuestion.getCorre
ctAnswer().isEmpty() ? "" :
mQuestion.getCorrectAnswer());

        other_answer_1.setText(mQuestion.getAns
werb().isEmpty() ? "" :
mQuestion.getAnswerb());

        other_answer_2.setText(mQuestion.getAns
wer().isEmpty() ? "" :
mQuestion.getAnswerc());

        other_answer_3.setText(mQuestion.getAns
werd().isEmpty() ? "" :
mQuestion.getAnswerd());

        save_btn.setOnClickListener(this);

```

```

        return fragmentView;
    }

    @Override

    public void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        mDatabase =
        DataRepository.getDBInstance();

        questionRef =
        mDatabase.getReference("Questions");
    }

    public void setQuestion(Question
    question) {

        this.mQuestion = question;
    }

    @Override

    public void onClick(View view) {

        if (view.getId() ==
        R.id.save_question_in_fragment) {

            final Question toSaved = mQuestion;

            if
            (question.getText().toString().isEmpty()) {

                question.setError("Question can't
                be empty");

                return;
            }
        }
    }

```

```

        if
        (correct_answer.getText().toString().isEmpty()
        ) {

            correct_answer.setError("Answer
            can't be empty");

            return;
        }

        if
        (other_answer_1.getText().toString().isEmpty()
        ) {

            other_answer_1.setError("Answer
            can't be empty");

            return;
        }

        if
        (other_answer_2.getText().toString().isEmpty()
        ) {

            other_answer_2.setError("Answer
            can't be empty");

            return;
        }

        if
        (other_answer_3.getText().toString().isEmpty()
        ) {

            other_answer_3.setError("Answer
            can't be empty");

            return;
        }

        String q =
        question.getText().toString();
    }

```

```

        String          ca          =
correct_answer.getText().toString();

        String          oa1         =
other_answer_1.getText().toString();

        String          oa2         =
other_answer_2.getText().toString();

        String          oa3         =
other_answer_3.getText().toString();

        toSaved.setQuestion(q);

        toSaved.setCorrectAnswer(ca);

        toSaved.setAnswera(ca);

        toSaved.setAnswerb(oa1);

        toSaved.setAnswerc(oa2);

        toSaved.setAnswerd(oa3);

questionRef.child(String.valueOf(toSaved.getId()))
.setValue(toSaved);

        Toast.makeText(getActivity(),
"Question detail successfully saved",
Toast.LENGTH_LONG).show();

getActivity().getSupportFragmentManager()
.popBackStack();
    }
}
}

DetailSampleFragment.java

package com.art.code.cplusplus.ui;

import android.os.Bundle;

import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Model.Sample;

import com.art.code.cplusplus.R;

import
com.google.firebase.database.DatabaseReference;

import
com.google.firebase.database.FirebaseDatabase;

public class DetailSampleFragment extends
Fragment implements
View.OnClickListener{

    View fragmentView;

    private Sample mSample;

    TextView judul, content;

    FirebaseDatabase mDatabase;

    DatabaseReference sampleRef;

    @Nullable

```

```

@Override

public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {

    fragmentView =
inflater.inflate(R.layout.fragment_detail,
container, false);

    judul = (TextView)
fragmentView.findViewById(R.id.fragment
_detail_title);

    content = (TextView)
fragmentView.findViewById(R.id.fragment
_detail_content);

    Button saveBtn = (Button)
fragmentView.findViewById(R.id.fragment
_detail_button);

    String judulnya = mSample.getTitle();

    judul.setText(judulnya);

content.setText(mSample.getContent());

    saveBtn.setOnClickListener(this);

    return fragmentView;
}

@Override

public void onCreate(@Nullable Bundle
savedInstanceState) {

    super.onCreate(savedInstanceState);

    mDatabase =
DataRepository.getDBInstance();

```

```

sampleRef =
mDatabase.getReference("Samples");
}

public void setSample(Sample sample){
    this.mSample = sample;
}

@Override

public void onClick(View view) {

    if (view.getId() ==
R.id.fragment_detail_button) {

        String cntn =
content.getText().toString();

        Sample newSample = new
Sample(mSample.getId(),
mSample.getTitle(), cntn);

sampleRef.child(String.valueOf(mSample.g
getId())).setValue(newSample);

        Toast.makeText(getActivity(),
"Sample detail successfully saved",
Toast.LENGTH_LONG).show();

getActivity().getSupportFragmentManager()
.popBackStack();

    }

}

}

DetailSubChapterFragment.java

package com.art.code.cplusplus.ui;

```



```

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import org.w3c.dom.Text;

public class DetailSubChapterFragment
extends Fragment implements
View.OnClickListener {

    View fragmentView;

    private SubChapter subChapter;

    TextView judul, content;

    FirebaseDatabase mDatabase;

    DatabaseReference subChapterRef;

    @Nullable

    @Override

    public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,

        Bundle
savedInstanceState) {

        fragmentView =
inflater.inflate(R.layout.fragment_detail,
container, false);

        judul = (TextView)
fragmentView.findViewById(R.id.fragment
_detail_title);

        content = (TextView)
fragmentView.findViewById(R.id.fragment
_detail_content);

        Button saveBtn = (Button)
fragmentView.findViewById(R.id.fragment
_detail_button);

        String judulnya =
subChapter.getChapter() + "." +
subChapter.getSubchapter() + ". " +
subChapter.getTitle();

        judul.setText(judulnya);

        content.setText(subChapter.getContent());

        saveBtn.setOnClickListener(this);

        return fragmentView;
    }

    @Override

    public void onCreate(@Nullable Bundle
savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);

        FirebaseDatabase.getInstance()
            .getReference("SubChapters")
            .child(subChapterRef)
            .setValue(new SubChapter(
                subChapter.getChapter(),
                subChapter.getSubchapter(),
                subChapter.getTitle(), cntn));

        Toast.makeText(getActivity(), "Sub
        chapter detail successfully saved",
        Toast.LENGTH_LONG).show();

        getActivity().getSupportFragmentManager()
            .popBackStack();

        public void onClick(View view) {

            if (view.getId() == R.id.fragment_detail_button) {

                String cntn = content.getText().toString();

                String ch = String.valueOf(subChapter.getChapter());

                while (ch.length() < 2) {

                    ch = '0' + ch;

                }

                String sb = String.valueOf(subChapter.getSubchapter());

                while (sb.length() < 3) {

                    sb = '0' + sb;

                }

                String key = "ch" + ch + "s" + sb;

                SubChapter newSubChapter = new
                SubChapter(subChapter.getChapter(),
                subChapter.getSubchapter(),
                subChapter.getTitle(), cntn);

                subChapterRef.child(key).setValue(newSub
                Chapter);

                Toast.makeText(getActivity(), "Sub
                chapter detail successfully saved",
                Toast.LENGTH_LONG).show();

                getActivity().getSupportFragmentManager()
                .popBackStack();

            }

        }

    }

}

HomeActivity.java

package com.art.code.cplusplus.ui;

import android.content.Intent;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.widget.LinearLayout;

import android.widget.Toast;

```

```

import com.art.code.cplusplus.R;

public class HomeActivity extends
    AppCompatActivity implements
    View.OnClickListener {

    private boolean
    doubleBackToExitPressedOnce = false;

    LinearLayout learning, samples, test,
    about;

    @Override

    protected void onCreate(Bundle
    savedInstanceState) {

        super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_home);

        learning = (LinearLayout)
    findViewById(R.id.learnings);

        samples = (LinearLayout)
    findViewById(R.id.samples);

        test = (LinearLayout)
    findViewById(R.id.tests);

        about = (LinearLayout)
    findViewById(R.id.about);

        learning.setOnClickListener(this);

        samples.setOnClickListener(this);

        test.setOnClickListener(this);

        about.setOnClickListener(this);
    }

    @Override

    public void onClick(View view) {

        switch (view.getId()) {

            case R.id.learnings:

                startActivity(new Intent(this,
                LearningActivity.class)

                .addFlags(Intent.FLAG_ACTIVITY_CLEA
                R_TOP |

                Intent.FLAG_ACTIVITY_SINGLE_TOP));

                break;

            case R.id.samples:

                startActivity(new Intent(this,
                SamplesActivity.class)

                .addFlags(Intent.FLAG_ACTIVITY_CLEA
                R_TOP |

                Intent.FLAG_ACTIVITY_SINGLE_TOP));

                break;

            case R.id.tests:

                startActivity(new Intent(this,
                TestActivity.class)

                .addFlags(Intent.FLAG_ACTIVITY_CLEA
                R_TOP |

                Intent.FLAG_ACTIVITY_SINGLE_TOP));

                break;

            case R.id.about:

```

```

        startActivity(new Intent(this,
AboutActivity.class)

        .addFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP |

Intent.FLAG_ACTIVITY_SINGLE_TOP));

        break;

    default:

        break;

    }

}

@Override

public void onBackPressed() {

    if (doubleBackToExitPressedOnce) {

        System.exit(0);

        return;

    }

    this.doubleBackToExitPressedOnce =
true;

    Toast.makeText(this,

        "Please click BACK again to exit",
Toast.LENGTH_SHORT).show();

    new Handler().postDelayed(new
Runnable() {

        @Override

        public void run() {

```

```

        doubleBackToExitPressedOnce =
false;

        }

        }, 2000);

    }

    @Override

    public boolean

    onCreateOptionsMenu(Menu menu) {

        MenuInflater inflater = new
MenuInflater(this);

        inflater.inflate(R.menu.main_menu,
menu);

        return true;

    }

    @Override

    public boolean

    onOptionsItemSelected(MenuItem item) {

        switch (item.getItemId()) {

            case R.id.logout:

                LoginActivity.sharedPrefManager.logout();

                startActivity(new Intent(this,
LoginActivity.class));

                finish();

                break;

            case R.id.exit:

                System.exit(0);

                break;

```

```

        default:
            break;
    }

    return
    super.onOptionsItemSelected(item);
}
}

```

LearningActivity.java

```

package com.art.code.cplusplus.ui;

import android.support.v4.app.Fragment;

import
android.support.v4.app.FragmentTransaction
;

import
android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

public class LearningActivity extends
AppCompatActivity {

    FragmentTransaction transaction;

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_learning);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    if (savedInstanceState == null) {

        Fragment listChapterFragment =
ListChapterFragment.newInstance();

        transaction =
getSupportFragmentManager().beginTransaction();

        transaction.add(R.id.learning_container,
listChapterFragment);

        transaction.commit();

    }

}

@Override

public boolean onSupportNavigateUp() {

    finish();

    return true;

}

public void showSubChapterof(Chapter
chapter) {

    SubChapterFragment
subChapterFragment =
SubChapterFragment.newInstance(chapter);

    FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();

```

```

transaction.addToBackStack("chapterlist").r
eplace(R.id.learning_container,
        subChapterFragment);

transaction.commit();
}

public void
showSubChapterDetail(SubChapter
subChapter) {

    SubChapterDetailFragment
subChapterDetailFragment = new
SubChapterDetailFragment();

subChapterDetailFragment.setSubChapter(s
ubChapter);

    FragmentTransaction transaction =
getSupportFragmentManager().beginTransa
ction();

transaction.addToBackStack("subchapterlist
").replace(R.id.learning_container,
        subChapterDetailFragment);

transaction.commit();
}
}

```

ListChapterFragment.java

```

package com.art.code.cplusplus.ui;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

```

```

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingActio
nButton;

import android.support.v4.app.Fragment;

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

import
com.art.code.cplusplus.Model.Chapter;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.Chapter
AdminViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerA
dapter;

```

```

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

public class ListChapterFragment extends Fragment{

    private FirebaseDatabase mDatabase;

    private DatabaseReference mChapterRef;

    private View chapterListView;

    private RecyclerView listOfChapters;

    private
    FirebaseRecyclerAdapter<Chapter,
    ChapterAdminViewHolder>
    mChapterAdapter;

    public static ListChapterFragment
    newInstance() {

        ListChapterFragment
        listChapterFragment = new
        ListChapterFragment();

        return listChapterFragment;

    }

    @Override

    public void onCreate(@Nullable Bundle
    savedInstanceState) {

        super.onCreate(savedInstanceState);

        mDatabase =
        DataRepository.getDBInstance();

        mChapterRef =
        mDatabase.getReference("Chapters");

    }

    @Nullable
    @Override

    public View onCreateView(LayoutInflater
    inflater, @Nullable ViewGroup container,
    @Nullable Bundle savedInstanceState) {

        chapterListView =
        inflater.inflate(R.layout.fragment_chapter_a
        dmin, container, false);

        FloatingActionButton fab =
        (FloatingActionButton)
        chapterListView.findViewById(R.id.fab_ta
        mbah_chapter);

        fab.setVisibility(View.GONE);

        listOfChapters = (RecyclerView)
        chapterListView.findViewById(R.id.admin_
        chapter_container);

        TextView title = (TextView)
        chapterListView.findViewById(R.id.fragme
        nt_title);

        title.setText("Chapter");

        RecyclerView.LayoutManager
        layoutManager = new
        LinearLayoutManager(container.getContext(
        ));

        listOfChapters.setLayoutManager(layoutMa
        nager);

```

```

loadChapters();

return chapterListView;
}

private void loadChapters() {

    mChapterAdapter = new
    FirebaseRecyclerAdapter<Chapter,
    ChapterAdminViewHolder>(
        Chapter.class,
        R.layout.admin_item,
        ChapterAdminViewHolder.class,
        mChapterRef
    ) {

        @Override

        protected void
        populateViewHolder(ChapterAdminViewHo
        lder chapterAdminViewHolder, final
        Chapter chapter, int i) {

            String nameInList =
            chapter.getChapter() + ". " +
            chapter.getTitle();

            chapterAdminViewHolder.chapter_name.set
            Text(nameInList);

            chapterAdminViewHolder.edit_chapter_btn.
            setVisibility(View.GONE);

            chapterAdminViewHolder.del_chapter_btn.s
            etVisibility(View.GONE);

```

```

chapterAdminViewHolder.setItemClickListe
ner(new ItemClickListener() {

    @Override

    public void onClick(View view,
    int position, boolean isLongClick) {

        if
        (getLifecycle().getCurrentState()
        .isAtLeast(Lifecycle.State.STARTED)) {

            ((LearningActivity)
            getActivity()).showSubChapterof(chapter);

        }

    }

});

}

};

mChapterAdapter.notifyDataSetChanged();

listOfChapters.setAdapter(mChapterAdapter
);

}

}

```

ListOfQuestionAdminFragment.java

```

package com.art.code.cplusplus.ui;

import android.app.AlertDialog;

import android.arch.lifecycle.Lifecycle;

```



```

import android.os.Bundle;

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingAction
nButton;

import android.support.v4.app.Fragment;

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.Question;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.Questio
nViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerA
dapter;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import com.google.firebase.database.Query;

import
com.google.firebase.database.ValueEventLis
tener;

public class ListOfQuestionAdminFragment
extends      Fragment      implements
View.OnClickListener{

    View questionsListView;

    FloatingActionButton fabAddQuestion;

    RecyclerView listOfQuestions;

    RecyclerView.LayoutManager
layoutManager;

    TextView title;

    private int emptyId;

    private int chapterNumber;

    FirebaseDatabase mDatabase;

```

```

DatabaseReference questionRef;

FirebaseRecyclerAdapter<Question,
QuestionViewHolder> mQuestionsAdapter;

public static
ListOfQuestionAdminFragment
newInstance(Chapter chapter) {

    ListOfQuestionAdminFragment
listOfQuestionAdminFragment = new
ListOfQuestionAdminFragment();

    Bundle args = new Bundle();

    args.putInt("ch", chapter.getChapter());

listOfQuestionAdminFragment.setArgument
s(args);

    return listOfQuestionAdminFragment;
}

@Override

public void onCreate(@Nullable Bundle
savedInstanceState) {

    super.onCreate(savedInstanceState);

    mDatabase =
DataRepository.getDBInstance();

    questionRef =
mDatabase.getReference("Questions");

    Bundle args = getArguments();

    chapterNumber = args.getInt("ch");
}

@Nullable

@Override

```

```

public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {

    questionsListView =
inflater.inflate(R.layout.fragment_chapter_a
dmin, container, false);

    fabAddQuestion =
(FloatingActionButton)
questionsListView.findViewById(R.id.fab_t
ambah_chapter);

    fabAddQuestion.setOnClickListener(this);

    listOfQuestions = (RecyclerView)
questionsListView.findViewById(R.id.admi
n_chapter_container);

    title = (TextView)
questionsListView.findViewById(R.id.frag
ment_title);

    title.setText("Questions");

    layoutManager = new
LinearLayoutManager(container.getContext(
));

    listOfQuestions.setLayoutManager(layoutM
anager);

    loadQuestions();

    getEmptyId();

    return questionsListView;
}

private void getEmptyId() {

```

```

        }
    });
}

private void loadQuestions() {
    Query keyRef =
questionRef.orderByChild("chapter").equalTo(
chapterNumber);

    mQuestionsAdapter = new
FirebaseRecyclerViewAdapter<Question,
QuestionViewHolder>(
        Question.class,
        R.layout.question_item,
        QuestionViewHolder.class, keyRef
    ) {
        @Override
            protected void
populateViewHolder(QuestionViewHolder
questionViewHolder, final Question
question, int i) {
                String answer = "Answer : "
+question.getCorrectAnswer();

                questionViewHolder.question.setText(questi
on.getQuestion());

                questionViewHolder.correct_answer.setText
(answer);

                questionViewHolder.del_btn.setOnClickList
ener(new View.OnClickListener() {
                    @Override

```

```

questionRef.addListenerForSingleValueEve
nt(new ValueEventListener() {
    @Override
        public void
onDataChange(DataSnapshot dataSnapshot)
{
            int z = 1;
            int id = z;
            for (DataSnapshot ds :
dataSnapshot.getChildren()) {
                Question onIterQ =
ds.getValue(Question.class);
                if (onIterQ.getChapter() ==
chapterNumber) {
                    if (onIterQ.getId() == z) {
                        z++;
                        id = z;
                    } else {
                        id = z;
                    }
                }
            }
            emptyId = id;
        }
    @Override
        public void
onCancelled(DatabaseError databaseError) {

```

```

        public void onClick(View view)
    {
        showDeleteQuestionDialog(question);
    }
});

questionViewHolder.setItemClickListener(new ItemClickListener() {
    @Override
    public void onClick(View view, int position, boolean isLongClick) {
        if (getLifecycle().getCurrentState().isAtLeast(Lifecycle.State.STARTED)) {
            ((AdminActivity) getActivity()).showQuestionDetail(question);
        }
    }
});
}

mQuestionsAdapter.notifyDataSetChanged();

listOfQuestions.setAdapter(mQuestionsAdapter);
}

private void showDeleteQuestionDialog(Question question) {
    final Question deletedQuestion = question;

    final AlertDialog yesNoDialog = new AlertDialog.Builder(getActivity())
        .setTitle("Delete Question")
        .setMessage("Are you sure?")
        .setNegativeButton(R.string.cancel, null)
        .setPositiveButton("Yes", null)
        .setIcon(R.drawable.ic_warning_black_24dp)
        .create();

    yesNoDialog.show();

    yesNoDialog.getButton(AlertDialog.BUTTON_NEGATIVE).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            yesNoDialog.dismiss();
        }
    });

    yesNoDialog.getButton(AlertDialog.BUTTON

```

```

ON_POSITIVE).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String key =
String.valueOf(deletedQuestion.getId());
questionRef.child(key).removeValue();
mQuestionsAdapter.notifyDataSetChanged();
        yesNoDialog.dismiss();
    }
});
}
@Override
public void onClick(View view) {
    if (view.getId() ==
R.id.fab_tambah_chapter) {
        Question newQuestion = new
Question();
newQuestion.setChapter(chapterNumber);
        newQuestion.setId(emptyId);
        newQuestion.setQuestion("");
        newQuestion.setAnswera("");
        newQuestion.setAnswerb("");
        newQuestion.setAnswerc("");
        newQuestion.setAnswerd("");
        newQuestion.setCorrectAnswer("");
        if (getLifecycle().getCurrentState()
.isAtLeast(Lifecycle.State.STARTED)) {
            ((AdminActivity)
getActivity()).showQuestionDetail(newQues
tion);
        }
    }
}
}
}

ListSampleFragment.java
package com.art.code.cplusplus.ui;
import android.arch.lifecycle.Lifecycle;
import android.os.Bundle;
import android.support.annotation.Nullable;
import
android.support.design.widget.FloatingActio
nButton;
import android.support.v4.app.Fragment;
import
android.support.v7.widget.LinearLayoutMan
ager;
import
android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

import
com.art.code.cplusplus.Model.Sample;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.Sample
AdminViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerA
dapter;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

public class ListSampleFragment extends
Fragment{

    View listSampleView;

    TextView title;

    FloatingActionButton fab;

    RecyclerView listOfSamples;

    RecyclerView.LayoutManager
layoutManager;

    FirebaseDatabase mDatabase;

    DatabaseReference samplesRef;

    FirebaseRecyclerAdapter<Sample,
SampleAdminViewHolder> mAdapter;

    @Override

    public void onCreate(@Nullable Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        mDatabase =
DataRepository.getDBInstance();

        samplesRef =
mDatabase.getReference("Samples");

    }

    @Nullable

    @Override

    public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {

        listSampleView =
inflater.inflate(R.layout.fragment_chapter_a
dmin, container, false);

        title =
(listSampleView.findViewById(R.id.fragment_
t_title));

        fab =
(listSampleView.findViewById(R.id.fab_tam
bah_chapter));

        listOfSamples =
(listSampleView.findViewById(R.id.admin_
chapter_container));

```

```

title.setText(R.string.samples);

fab.setVisibility(View.GONE);

layoutManager = new
LinearLayoutManager(container.getContext(
));

```

```

listOfSamples.setLayoutManager(layoutMa
nager);

```

```

loadSamples();

```

```

return listSampleView;

```

```

}

```

```

private void loadSamples() {

```

```

    mAdapter = new
    FirebaseRecyclerAdapter<Sample,
    SampleAdminViewHolder>(
        Sample.class,
        R.layout.admin_item,
        SampleAdminViewHolder.class,
        samplesRef

```

```

    ) {

```

```

        @Override

```

```

        protected void
        populateViewHolder(SampleAdminViewHo
        lder sampleAdminViewHolder, final Sample
        sample, int i) {

```

```

        sampleAdminViewHolder.title.setText(samp
        le.getTitle());

```

```

        sampleAdminViewHolder.edit_btn.setVisibi
        lity(View.GONE);

```

```

        sampleAdminViewHolder.del_btn.setVisibil
        ity(View.GONE);

```

```

        sampleAdminViewHolder.setItemClickListe
        ner(new ItemClickListener() {

```

```

            @Override

```

```

            public void onClick(View view,
            int position, boolean isLongClick) {

```

```

                if

```

```

                (getLifecycle().getCurrentState()

```

```

                .isAtLeast(Lifecycle.State.STARTED)) {

```

```

                    ((SamplesActivity)

```

```

                    getActivity()).showSampleDetail(sample);

```

```

                }

```

```

            }

```

```

        });

```

```

    }

```

```

};

```

```

        mAdapter.notifyDataSetChanged();

```

```

        listOfSamples.setAdapter(mAdapter);

```

```

    }

```

```

}

```

LoginActivity.java

```

package com.art.code.cplusplus.ui;

```

```

import android.app.AlertDialog;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.support.v7.app.AlertDialog;

import
android.support.v7.app.AppCompatActivity;

import android.util.Patterns;

import android.view.LayoutInflater;

import android.view.View;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.Toast;

import
com.art.code.cplusplus.DataRepository;

import com.art.code.cplusplus.Model.User;

import com.art.code.cplusplus.R;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

```

```

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

import java.util.ArrayList;

import java.util.Date;

import java.util.Map;

public class LoginActivity extends
AppCompatActivity {

    public static SharedPreferences
sharedPrefManager;

    private FirebaseDatabase sInstance;

    private DatabaseReference userReference;

    final private int adminUAC = 0;

    final private int userUAC = 1;

    private ProgressDialog pDialogLogin,
pDialogRegister;

    EditText newUsername, newPassword,
newRePassword, newEmail;

    EditText username, password;

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);

```



```

        sharedPrefManager = new
SharedPrefManager(this);

        sInstance =
DataRepository.getDBInstance();

        userReference =
sInstance.getReference("Users");

        username =
findViewById(R.id.username_input);

        password =
findViewById(R.id.password_input);

        if (sharedPrefManager.getBoolean()
&& sharedPrefManager.getUAC() ==
userUAC) {

            userReference.child(sharedPrefManager.get
Value()).child("loginAt").setValue(new
Date());

            startActivity(new Intent(this,
HomeActivity.class));

            finish();

        } else if
(sharedPrefManager.getBoolean() &&
sharedPrefManager.getUAC() ==
adminUAC) {

            userReference.child(sharedPrefManager.get
Value()).child("loginAt").setValue(new
Date());

            startActivity(new Intent(this,
AdminActivity.class));

            finish();
    }

    public void signInAttempClick(View
view) {

        pDialogLogin = new
ProgressDialog(LoginActivity.this,
R.style.LoadingTheme);

        pDialogLogin.setMessage("Authenticating...
");

        pDialogLogin.setIndeterminate(false);

        pDialogLogin.setCancelable(true);

        pDialogLogin.show();

        final String username =
this.username.getText().toString();

        final String password =
this.password.getText().toString();

        if (!validateLoginRequest(username,
password)) {

            pDialogLogin.dismiss();

            return;

        }

        userReference.addListenerForSingleValueE
vent(new ValueEventListener() {

            @Override

            public void
onDataChange(DataSnapshot dataSnapshot)
{

```

```

        if (dataSnapshot.child(username).exists()) {
            if (!username.isEmpty()) {
                User loginUser =
                dataSnapshot.child(username).getValue(User.class);

                if (loginUser.getPassword().equals(password)) {
                    userReference.child(username).child("loginAt").setValue(new Date());

                    pDialogLogin.dismiss();

                    sharedPreferences.putBoolean(SharedPreferencesManager.sharedPrefLogin, true);

                    sharedPreferences.putString(SharedPreferencesManager.sharedPrefUser,
                    loginUser.getUsername());

                    sharedPreferences.putInt(SharedPreferencesManager.sharedPrefUAC, loginUser.getUac());

                    sharedPreferences.putInt(SharedPreferencesManager.sharedPrefHiScore,
                    loginUser.getHiScore());

                    if (loginUser.getUac() == userUAC) {
                        startActivity(new
                        Intent(LoginActivity.this,
                        HomeActivity.class));
                    }
                } else {
                    pDialogLogin.dismiss();

                    Toast.makeText(LoginActivity.this,
                    R.string.wrong_password,
                    Toast.LENGTH_SHORT).show();

                } else {
                    pDialogLogin.dismiss();

                    Toast.makeText(LoginActivity.this,
                    R.string.please_enter_username,
                    Toast.LENGTH_SHORT).show();

                } else {
                    pDialogLogin.dismiss();

                    Toast.makeText(LoginActivity.this,
                    R.string.username_not_exists,
                    Toast.LENGTH_SHORT).show();

                }
            }
        } else {
            if (loginUser.getUac() == adminUAC) {
                startActivity(new
                Intent(LoginActivity.this,
                AdminActivity.class));
            }
        }
    }
}

```

```

    }

    @Override

    public void
onCancelled(DatabaseError databaseError) {

    }

});

}

private boolean
validateLoginRequest(String username,
String password) {

    boolean valid = true;

    if (username.isEmpty() ||
!username.matches("[a-zA-Z][a-zA-Z0-
9\\_\\-\\.]{5,11}$")) {

this.username.setError(getString(R.string.val
id_username));

        valid = false;

    } else this.username.setError(null);

    if (password.isEmpty() ||
!password.matches("[a-zA-Z][a-zA-Z0-
9]{5,15}$")) {

this.password.setError(getString(R.string.val
id_password));

        valid = false;

    } else this.password.setError(null);

    return valid;

}

    public void signUpAttempClick(View
view) {

        LayoutInflater inflater =
this.getLayoutInflater();

        final View dialogView =
inflater.inflate(R.layout.sign_up_layout,
null);

        newUsername =
dialogView.findViewById(R.id.new_userna
me_input);

        newPassword =
dialogView.findViewById(R.id.new_passw
ord_input);

        newRePassword =
dialogView.findViewById(R.id.new_passw
ord_re_input);

        newEmail =
dialogView.findViewById(R.id.new_email_
input);

        final AlertDialog registerDialog = new
AlertDialog.Builder(LoginActivity.this)

                .setTitle(R.string.register_acc)

                .setMessage(R.string.fill_in_details)

                .setPositiveButton(R.string.sign_up, null)

                .setNegativeButton(R.string.cancel, null)

                .setView(dialogView)

```



```

R.string.username_already_exists,
Toast.LENGTH_LONG).show();

        return;
    } else {
        if
(dataSnapshot.getValue() != null) {
            if
(checkEmailonUsers((Map<String, Object>)
dataSnapshot.getValue(),
newUser.getEmail())) {

pDialogRegister.dismiss();

newEmail.setError("Email has been used");

//
Toast.makeText(LoginActivity.this, "Email
has been used",
Toast.LENGTH_LONG).show();

        return;
    }
}

pDialogRegister.dismiss();

userReference.child(newUser.getUsername()
).setValue(newUser);

Toast.makeText(LoginActivity.this,
R.string.account_registration_success,
Toast.LENGTH_SHORT).show();

registerDialog.dismiss();
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {
}

});

// registerDialog.dismiss();
}

});

registerDialog.getButton(AlertDialog.BUTTON
ON_NEGATIVE).setOnClickListener(
new View.OnClickListener() {
    @Override
    public void onClick(View view)
    {
        registerDialog.dismiss();
    }
});

private boolean
checkEmailonUsers(Map<String, Object>
users, String email) {
    boolean hasEmail = true;

```

```

        ArrayList<String> emailList = new
ArrayList<>();

        for (Map.Entry<String, Object> entry :
users.entrySet()) {

            Map user = (Map) entry.getValue();

            emailList.add((String)
user.get("email"));

        }

        if (!emailList.contains(email)) {

            hasEmail = false;

        }

        return hasEmail;

    }

    private boolean
validateRegisterRequest(String username,
String password, String rePassword, String
email) {

        boolean valid = true;

        if (username.isEmpty() ||
!username.matches("[a-zA-Z][a-zA-Z0-
9\\_\\-\\.]{5,11}$")) {

            newUsername.setError(getString(R.string.va
lid_username));

            valid = false;

        } else newUsername.setError(null);

        if (password.isEmpty() ||
!password.matches("[a-zA-Z][a-zA-Z0-
9]{5,15}$")) {

            newPassword.setError(getString(R.string.val
id_password));

            valid = false;

        } else newPassword.setError(null);

        if (rePassword.isEmpty() ||
!rePassword.equals(password)) {

            newRePassword.setError(getString(R.string.
valid_repassword));

            valid = false;

        } else newPassword.setError(null);

        if (email.isEmpty() ||
!Patterns.EMAIL_ADDRESS.matcher(email
).matches()) {

            newEmail.setError(getString(R.string.valid_
email));

            valid = false;

        } else newEmail.setError(null);

        return valid;

    }

    public static class SharedPrefManager {

        public static final String
sharedPrefName = "attempLogin";

        public static final String
sharedPrefLogin = "LoggedIn";

        public static final String sharedPrefUser
= "username";

```

```

    public static final String
sharedPrefUAC = "UAC";

    public static final String
sharedPrefHiScore = "HiScore";

    SharedPreferences sharedPreferences;

    static SharedPreferences.Editor editor;

    public SharedPrefManager(Context
context) {

        sharedPreferences =
context.getSharedPreferences(sharedPrefNa
me,

        Context.MODE_PRIVATE);

        editor = sharedPreferences.edit();

    }

    public void putString(String key, String
value) {

        editor.putString(key, value);

        editor.commit();

    }

    public void putInt(String key, int value)
{

        editor.putInt(key, value);

        editor.commit();

    }

    public void logout() {

        editor.clear();

        editor.commit();

    }

    public void putBoolean(String key,
boolean value) {

        editor.putBoolean(key, value);

        editor.commit();

    }

    public String getValue(){

        return
sharedPreferences.getString(sharedPrefUser,
"");

    }

    public int getHiScore() {

        return
sharedPreferences.getInt(sharedPrefHiScore,
0);

    }

    public int getUAC() {

        return
sharedPreferences.getInt(sharedPrefUAC,
0);

    }

    public boolean getBoolean() {

        return
sharedPreferences.getBoolean(sharedPrefLo
gin, false);

    }

}
}

```

QuestionAdminFragment.java

```

package com.art.code.cplusplus.ui;

import android.app.AlertDialog;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingActio
nButton;

import android.support.v4.app.Fragment;

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.Question;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.Chapter
AdminViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerA
dapter;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

import java.util.List;

public class QuestionAdminFragment
extends Fragment {

    View adminChapterFragment;

    FloatingActionButton fabTambahChapter;

    RecyclerView listOfChapters;

```



```

RecyclerView.LayoutManager
layoutManager;

TextView title;

List<Question> listOfQuestion;

FirebaseRecyclerAdapter<Chapter,
ChapterAdminViewHolder>
mChapterAdapter;

FirebaseDatabase mDatabase;

DatabaseReference chaptersRef;

DatabaseReference questionRef;

public static QuestionAdminFragment
newInstance() {

    QuestionAdminFragment
questionAdminFragment = new
QuestionAdminFragment();

    return questionAdminFragment;
}

@Override

public void onCreate(@Nullable Bundle
savedInstanceState) {

    super.onCreate(savedInstanceState);

    mDatabase =
DataRepository.getDBInstance();

    chaptersRef =
mDatabase.getReference("Chapters");

    questionRef =
mDatabase.getReference("Questions");
}

}

@Nullable

@Override

public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,

Bundle
savedInstanceState) {

    adminChapterFragment =
inflater.inflate(R.layout.fragment_chapter_a
dmin, container, false);

    fabTambahChapter =
(FloatingActionButton)
adminChapterFragment.findViewById(R.id.
fab_tambah_chapter);

    fabTambahChapter.setVisibility(View.GON
E);

    listOfChapters = (RecyclerView)
adminChapterFragment.findViewById(R.id.
admin_chapter_container);

    title = (TextView)
adminChapterFragment.findViewById(R.id.
fragment_title);

    title.setText("Chapters");

    listOfQuestion = null;

    layoutManager = new
LinearLayoutManager(container.getContext(
));

    listOfChapters.setLayoutManager(layoutMa
nager);

    loadChapters();
}

```

```

        return adminChapterFragment;
    }

    private int qCount(final int cha) {

        int r = 0;

        listOfQuestion.clear();

        questionRef.addListenerForSingleValueEvent(
            new ValueEventListener() {

                @Override

                public void
                onDataChange(DataSnapshot dataSnapshot)
                {

                    for (DataSnapshot ds :
                    dataSnapshot.getChildren()) {

                        Question newQFromDb =
                        ds.getValue(Question.class);

                        if (newQFromDb.getChapter()
                        == cha) {

                            listOfQuestion.add(newQFromDb);

                        }

                    }

                }

                @Override

                public void
                onCancelled(DatabaseError databaseError)
                {}

            });

        r = listOfQuestion.size();
    }

```

```

        return r;
    }

    private void loadChapters() {

        mChapterAdapter = new
        FirebaseRecyclerAdapter<Chapter,
        ChapterAdminViewHolder>(

            Chapter.class,

            R.layout.admin_item,

            ChapterAdminViewHolder.class,

            chaptersRef

        ) {

            @Override

            protected void
            populateViewHolder(ChapterAdminViewHo
            lder chapterAdminViewHolder,

                final Chapter
                chapter, int position) {

                String nameInList = "Bab " +
                chapter.getChapter() + " - " +
                chapter.getTitle();

                chapterAdminViewHolder.chapter_name.set
                Text(nameInList);

                chapterAdminViewHolder.edit_chapter_btn.
                setVisibility(View.GONE);

                chapterAdminViewHolder.del_chapter_btn.s
                etVisibility(View.GONE);
            }
        }
    }

```

```

chapterAdminViewHolder.setItemClickListener(new ItemClickListener() {
    @Override
    public void onClick(View view,
int position, boolean isLongClick) {
        if
(getLifecycle().getCurrentState()
.isAtLeast(Lifecycle.State.STARTED)) {
            ((AdminActivity)
getActivity()).showQuestionsOf(chapter);
        }
    }
});
}
};

mChapterAdapter.notifyDataSetChanged();

listOfChapters.setAdapter(mChapterAdapter
);
}
}

```

ResultFragment.java

```

package com.art.code.cplusplus.ui;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

```

```

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import com.art.code.cplusplus.Model.User;

import com.art.code.cplusplus.R;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

public class ResultFragment extends
Fragment {
    private int score = 0;

    private FirebaseDatabase mDb;

```

```

private DatabaseReference usrRef;

public static ResultFragment
newInstance(int score) {

    ResultFragment resultFragment = new
ResultFragment();

    Bundle args = new Bundle();

    args.putInt("sc", score);

    resultFragment.setArguments(args);

    return resultFragment;

}

@Override

public void onCreate(@Nullable Bundle
savedInstanceState) {

    super.onCreate(savedInstanceState);

    mDb =
DataRepository.getDBInstance();

    usrRef = mDb.getReference("Users");

    Bundle args = getArguments();

    score = args.getInt("sc");

}

@Nullable

@Override

public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,

Bundle
savedInstanceState) {

```

```

View resView =
inflater.inflate(R.layout.fragment_result,
container, false);

TextView value = (TextView)
resView.findViewById(R.id.value);

Button fin = (Button)
resView.findViewById(R.id.finish_button);

value.setText(String.valueOf(score));

fin.setOnClickListener(new
View.OnClickListener() {

@Override

public void onClick(View view) {

final int test;

final String username =
LoginActivity.sharedPrefManager.getValue(
);

usrRef.addListenerForSingleValueEvent(ne
w ValueEventListener() {

@Override

public void
onDataChange(DataSnapshot dataSnapshot)
{

for (DataSnapshot ds :
dataSnapshot.getChildren()) {

User onIterUsr =
ds.getValue(User.class);

if
(onIterUsr.getUsername().equals(username))
{

```

```

        int userHiScore =
onIterUsr.getHiScore();

        if (userHiScore < score)
{

usrRef.child(username).child("hiScore").set
Value(score);

LoginActivity.sharedPrefManager.putInt(LoginActivity.SharedPrefManager.sharedPrefHiScore, score);

        }
    }
}

@Override

public void
onCancelled(DatabaseError databaseError) {

}

});

if (getLifecycle().getCurrentState()

.isAtLeast(Lifecycle.State.STARTED)) {

        ((TestActivity)
getActivity()).endTest();

    }

}

});

return resView;

```

```

    }
}

SampleAdminFragment.java

package com.art.code.cplusplus.ui;

import android.app.AlertDialog;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingAction
nButton;

import android.support.v4.app.Fragment;

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

```

```

import
com.art.code.cplusplus.Model.Sample;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.Sample
AdminViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerA
dapter;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

public class SampleAdminFragment extends
Fragment implements
View.OnClickListener{

    View adminSampleFragment;

    TextView title;

    RecyclerView listOfSamples;

    RecyclerView.LayoutManager
layoutManager;

    FloatingActionButton fabButton;

    FirebaseRecyclerAdapter<Sample,
SampleAdminViewHolder>
mSampleAdapter;

    FirebaseDatabase mDatabase;

    DatabaseReference samplesRef;

    public static SampleAdminFragment
newInstance() {

        SampleAdminFragment
sampleAdminFragment = new
SampleAdminFragment();

        return sampleAdminFragment;

    }

    @Override

    public void onCreate(@Nullable Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        mDatabase =
DataRepository.getDBInstance();

        samplesRef =
mDatabase.getReference("Samples");

    }

    @Nullable

    @Override

    public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,

        Bundle
savedInstanceState) {

```

```

        adminSampleFragment =
inflater.inflate(R.layout.fragment_chapter_a
admin, container, false);

        fabButton = (FloatingActionButton)
adminSampleFragment.findViewById(R.id.f
ab_tambah_chapter);

        fabButton.setOnClickListener(this);

        listOfSamples = (RecyclerView)
adminSampleFragment.findViewById(R.id.
admin_chapter_container);

        title = (TextView)
adminSampleFragment.findViewById(R.id.f
ragment_title);

        title.setText("Samples");

        layoutManager = new
LinearLayoutManager(container.getContext(
));

listOfSamples.setLayoutManager(layoutMa
nager);

        loadAllSamples();

        return adminSampleFragment;
    }

    private void loadAllSamples() {

        mSampleAdapter = new
FirebaseRecyclerViewAdapter<Sample,
SampleAdminViewHolder>(

            Sample.class,
R.layout.admin_item,

```

```

SampleAdminViewHolder.class,
samplesRef

    ) {

        @Override

        protected void
populateViewHolder(SampleAdminViewHo
lder sampleAdminViewHolder, final Sample
sample, int i) {

            sampleAdminViewHolder.title.setText(samp
le.getTitle());

            sampleAdminViewHolder.edit_btn.setOnCli
ckListener(new View.OnClickListener() {

                @Override

                public void onClick(View view)
                {

                    showEditSampleDialog(sample);

                }

            });

            sampleAdminViewHolder.del_btn.setOnCli
ckListener(new View.OnClickListener() {

                @Override

                public void onClick(View view)
                {

                    showDeleteSampleDialog(sample);

```

```

    }
});

sampleAdminViewHolder.setItemClickListener(new ItemClickListener() {

    @Override

    public void onClick(View view,
int position, boolean isLongClick) {

        if
(getLifecycle().getCurrentState()

.isAtLeast(Lifecycle.State.STARTED)) {

            ((AdminActivity)
getActivity()).showSampleDetail(sample);

        }

    }

});

}

};

mSampleAdapter.notifyDataSetChanged();

listOfSamples.setAdapter(mSampleAdapter)
;

}

private void
showEditSampleDialog(Sample sample) {

    final Sample editedSample = sample;

```

```

        LayoutInflater inflater =
this.getLayoutInflater();

        final View dialogView =
inflater.inflate(R.layout.add_edit_chapter_la
yout, null);

        final EditText edit_sample_title =
dialogView.findViewById(R.id.chapter_title
);

        edit_sample_title.setText(editedSample.getT
itle());

        final AlertDialog editSampleDialog =
new AlertDialog.Builder(getActivity())

            .setTitle(R.string.sample_title)

            .setPositiveButton(R.string.save,
null)

            .setNegativeButton(R.string.cancel, null)

            .setView(dialogView)

            .setIcon(R.drawable.ic_mode_edit_black_24
dp)

            .create();

        editSampleDialog.getWindow().setSoftInput
Mode(WindowManager.LayoutParams.SOF
T_INPUT_ADJUST_RESIZE);

        editSampleDialog.show();

        editSampleDialog.getButton(AlertDialog.B

```



```

        UTTON_NEGATIVE).setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    editSampleDialog.dismiss();
                }
            });

editSampleDialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if
(edit_sample_title.getText().toString().equals(editedSample.getTitle())) {

edit_sample_title.setError("Click cancel to cancel changes");

        return;
    } else {
        String newTitle =
edit_sample_title.getText().toString();

        String key =
String.valueOf(editedSample.getId());

samplesRef.child(key).child("title").setValue
(newTitle);

mSampleAdapter.notifyDataSetChanged();

        editSampleDialog.dismiss();
    }
});

private void
showDeleteSampleDialog(Sample sample) {
    final Sample deletedSample = sample;

    final AlertDialog yesNoDialog = new
AlertDialog.Builder(getActivity())
        .setTitle("Delete " +
deletedSample.getTitle())
        .setMessage("Are you sure?")
        .setNegativeButton(R.string.cancel, null)
        .setPositiveButton("Yes", null)
        .setIcon(R.drawable.ic_warning_black_24dp
)
        .create();

    yesNoDialog.show();

    yesNoDialog.getButton(AlertDialog.BUTTON_NEGATIVE).setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            yesNoDialog.dismiss();
        }
    });
}

```

```

    }
});

yesNoDialog.getButton(AlertDialog.BUTTON
ON_POSITIVE).setOnClickListener(new
View.OnClickListener() {

    @Override

    public void onClick(View view) {

samplesRef.child(String.valueOf(deletedsam
ple.getId())).removeValue();

mSampleAdapter.notifyDataSetChanged();

        yesNoDialog.dismiss();
    }
});
}

@Override

public void onClick(View view) {

    if (view.getId() ==
R.id.fab_tambah_chapter) {

        LayoutInflater inflater =
this.getLayoutInflater();

        final View dialogView =
inflater.inflate(R.layout.add_edit_chapter_la
yout, null);

        final EditText title =
dialogView.findViewById(R.id.chapter_title
);

        final AlertDialog addSampleDialog
= new AlertDialog.Builder(getActivity())

            .setTitle("Add New Sample")

            .setPositiveButton(R.string.save, null)

            .setNegativeButton(R.string.cancel, null)

            .setView(dialogView)

            .setIcon(R.drawable.ic_playlist_add_black_
24dp)

            .create();

        addSampleDialog.getWindow().setSoftInput
Mode(WindowManager.LayoutParams.SOF
T_INPUT_ADJUST_RESIZE);

        addSampleDialog.show();

        addSampleDialog.getButton(AlertDialog.B
UTTON_NEGATIVE).setOnClickListener(
new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                addSampleDialog.dismiss();

            }

        });

        addSampleDialog.getButton(AlertDialog.B
UTTON_POSITIVE).setOnClickListener(ne
w View.OnClickListener() {

```

```

@Override
    public void onClick(View view) {
        if (title.getText().toString().isEmpty()) {
            title.setError("Sub chapter title can't be empty");
            return;
        } else {
            final Sample newSample = new Sample();
            samplesRef.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    int y = 1;
                    int id = y;
                    for (DataSnapshot ds : dataSnapshot.getChildren()) {
                        Sample onIterSample = ds.getValue(Sample.class);
                        if (onIterSample.getId() == y) {
                            y++;
                            id = y;
                        } else {
                            id = y;
                        }
                    }
                }
            });
        }
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {
}

});

mSampleAdapter.notifyDataSetChanged();
addSampleDialog.dismiss();
}
}
});
}
}

SampleDetailFragment.java
package com.art.code.cplusplus.ui;

```

```

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

```

```

import
com.art.code.cplusplus.Model.Sample;

import com.art.code.cplusplus.R;

public class SampleDetailFragment extends
Fragment {

    private Sample mSample;

    @Nullable

    @Override

    public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,
Bundle savedInstanceState) {

        View v =
inflater.inflate(R.layout.detail_for_user,
container, false);

        TextView title = (TextView)
v.findViewById(R.id.detail_title);

        TextView content = (TextView)
v.findViewById(R.id.detail_content);

        title.setText(mSample.getTitle());

```

```

content.setText(mSample.getContent());

        return v;
    }

    public void setSample(Sample sample) {

        this.mSample = sample;
    }
}

```

SamplesActivity.java

```

package com.art.code.cplusplus.ui;

import
android.support.v4.app.FragmentTransaction
;

import
android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import
com.art.code.cplusplus.Model.Sample;

import com.art.code.cplusplus.R;

public class SamplesActivity extends
AppCompatActivity {

    FragmentTransaction transaction;

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_samples);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    if (savedInstanceState == null) {

        ListSampleFragment
listSampleFragment = new
ListSampleFragment();

        transaction =
getSupportFragmentManager().beginTransaction();

transaction.add(R.id.sample_container,
listSampleFragment);

        transaction.commit();

    }

}

@Override

public boolean onSupportNavigateUp() {

    finish();

    return true;

}

public void showSampleDetail(Sample
sample) {

    SampleDetailFragment
sampleDetailFragment = new
SampleDetailFragment();

sampleDetailFragment.setSample(sample);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();

transaction.addToBackStack("samplelist").replace(R.id.sample_container,
sampleDetailFragment);

        transaction.commit();

    }

}

SplashActivity.java

package com.art.code.cplusplus.ui;

import android.content.Intent;

import
android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Toast;

import
com.art.code.cplusplus.Model.Chapter;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

```

```

import
com.google.firebase.database.ValueEventLis
tener;

import java.util.ArrayList;

import java.util.List;

public class SplashActivity extends
AppCompatActivity {

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        FirebaseDatabase.getInstance().setPersistenc
eEnabled(true);

        startActivity(new Intent(this,
LoginActivity.class));

        finish();

    }

}

```

StartTestFragment.java

```

package com.art.code.cplusplus.ui;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

```

```

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Model.Question;

import com.art.code.cplusplus.Model.User;

import com.art.code.cplusplus.R;

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

import java.util.Collections;

import java.util.List;

public class StartTestFragment extends
Fragment implements
View.OnClickListener {

    Button start;

    int hiScore;

    FirebaseDatabase mDb;

    private DatabaseReference usrRef;

```

```

public static StartTestFragment
newInstance() {
    StartTestFragment startTestFragment =
new StartTestFragment();

    return startTestFragment;
}

@Override

public View onCreateView(LayoutInflater
inflater, ViewGroup container,

Bundle
savedInstanceState) {
    View v =
inflater.inflate(R.layout.fragment_start_test,
container, false);

    TextView scoreTv = (TextView)
v.findViewById(R.id.current_hi_score);

    String crsc = hiScore != 0 ?
getString(R.string.currentHiScore) +
String.valueOf(hiScore) :
getString(R.string.neverTakeTest);

    scoreTv.setText(crsc);

    start = (Button)
v.findViewById(R.id.test_button_start);

    start.setOnClickListener(this);

    return v;
}

@Override

public void onCreate(@Nullable Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);

    hiScore =
LoginActivity.sharedPrefManager.getHiScore();

    mDb =
DataRepository.getDBInstance();

    usrRef = mDb.getReference("Users");
}

@Override

public void onClick(View view) {
    if (view.getId() ==
R.id.test_button_start) {
        ((TestActivity)getActivity()).startTest();
    }
}
}

SubChapterAdminFragment.java

package com.art.code.cplusplus.ui;

import android.app.AlertDialog;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

import android.os.Parcelable;

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingAction
Button;

import android.support.v4.app.Fragment;

```

```

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

import
com.art.code.cplusplus.ViewHolder.SubCha
pterViewHolder;

import
com.firebase.ui.database.FirebaseRecyclerA
dapter;

```

```

import
com.google.firebase.database.DataSnapshot;

import
com.google.firebase.database.DatabaseError
;

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import com.google.firebase.database.Query;

import
com.google.firebase.database.ValueEventLis
tener;

import java.io.Serializable;

public class SubChapterAdminFragment
extends Fragment implements
View.OnClickListener {

    View adminSubChapterFragment;

    int chapterNumber;

    FloatingActionButton
fabTambahSubChapter;

    RecyclerView listOfSubChapters;

    RecyclerView.LayoutManager
layoutManager;

    TextView title;

    EditText new_sub_chapter_title;

    FirebaseDatabase mDatabase;

    DatabaseReference subChapterReff;

```



```

        FirebaseRecyclerAdapter<SubChapter,
        SubChapterViewHolder>
        mSubchapterAdapter;

        public static SubChapterAdminFragment
        newInstance(Chapter chapter) {

            SubChapterAdminFragment
            subChapterAdminFragment = new
            SubChapterAdminFragment();

            Bundle args = new Bundle();

            args.putInt("ch", chapter.getChapter());

            subChapterAdminFragment.setArguments(a
            rgs);

            return subChapterAdminFragment;
        }

        @Override

        public void onCreate(@Nullable Bundle
        savedInstanceState) {

            super.onCreate(savedInstanceState);

            mDatabase =
            DataRepository.getDBInstance();

            subChapterReff =
            mDatabase.getReference("SubChapters");

            Bundle args = getArguments();

            chapterNumber = args.getInt("ch");
        }

        @Nullable

        @Override

```

```

        public View onCreateView(LayoutInflater
        inflater, @Nullable ViewGroup container,

            Bundle
            savedInstanceState) {

            adminSubChapterFragment =
            inflater.inflate(R.layout.fragment_chapter_a
            dmin, container, false);

            fabTambahSubChapter =
            (FloatingActionButton)
            adminSubChapterFragment.findViewById(
            R.id.fab_tambah_chapter);

            fabTambahSubChapter.setOnClickListener(t
            his);

            listOfSubChapters = (RecyclerView)
            adminSubChapterFragment.findViewById(
            R.id.admin_chapter_container);

            title = (TextView)
            adminSubChapterFragment.findViewById(
            R.id.fragment_title);

            title.setText("Sub-Chapters");

            layoutManager = new
            LinearLayoutManager(container.getContext(
            ));

            listOfSubChapters.setLayoutManager(layout
            Manager);

            loadSubChapter();

            return adminSubChapterFragment;
        }

        private void loadSubChapter() {

```

```

        Query query = Query.whereEqualTo("chapter", chapterNumber);
        subChapterReff.orderByChild("chapter").equalTo(chapterNumber);

        mSubchapterAdapter = new
        FirebaseRecyclerAdapter<SubChapter,
        SubChapterViewHolder>(
            SubChapter.class,
            R.layout.admin_item,
            SubChapterViewHolder.class, keyRef
        ) {
            @Override
            protected void
            populateViewHolder(SubChapterViewHolder subChapterViewHolder, final SubChapter
            subChapter, int pos) {
                String subChapterName =
                subChapter.getChapter() + "." +
                subChapter.getSubchapter() + ". " +
                subChapter.getTitle();

                subChapterViewHolder.sub_chapter_name.setText(subChapterName);

                subChapterViewHolder.del_sub_chapter_btn.setOnClickListener(new
                View.OnClickListener() {
                    @Override
                    public void onClick(View view)
                    {
                        showDeleteSubChapterDialog(subChapter);
                    }
                });
            }
        };

        subChapterViewHolder.edit_sub_chapter_btn.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view)
            {
                showEditSubChapterDialog(subChapter);
            }
        });

        subChapterViewHolder.setItemClickListener(new ItemClickListener() {
            @Override
            public void onClick(View view,
            int position, boolean isLongClick) {
                if
                (getLifecycle().getCurrentState()
                .isAtLeast(Lifecycle.State.STARTED)) {
                    ((AdminActivity)
                    getActivity()).showSubChapterDetail(subChapter);
                }
            }
        });
    }
}

```

```

};

mSubchapterAdapter.notifyDataSetChanged();

listOfSubChapters.setAdapter(mSubchapterAdapter);
}

private void showEditSubChapterDialog(SubChapter subChapter) {
    final SubChapter editedSubChapter = subChapter;

    LayoutInflater inflater = this.getLayoutInflater();

    final View dialogView = inflater.inflate(R.layout.add_edit_chapter_layout, null);

    final EditText edit_sub_chapter_title = dialogView.findViewById(R.id.chapter_title);

    edit_sub_chapter_title.setText(editedSubChapter.getTitle());

    final AlertDialog editSubChapterDialog = new AlertDialog.Builder(getActivity())
        .setTitle(R.string.edit_subchapter)
        .setPositiveButton(R.string.save, null)

        .setNegativeButton(R.string.cancel, null)

        .setView(dialogView)

        .setIcon(R.drawable.ic_mode_edit_black_24dp)

        .create();

    editSubChapterDialog.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_ADJUST_RESIZE);

    editSubChapterDialog.show();

    editSubChapterDialog.getButton(AlertDialog.BUTTON_NEGATIVE).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            editSubChapterDialog.dismiss();
        }
    });

    editSubChapterDialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (edit_sub_chapter_title.getText().toString().equals(editedSubChapter.getTitle())) {

```

```

edit_sub_chapter_title.setError("Click cancel
to cancel changes");

        return;
    } else {

        String      newTitle      =
edit_sub_chapter_title.getText().toString();

        String      chapterNumber  =
String.valueOf(editedSubChapter.getChapter
());

        String      subchapterNumber =
String.valueOf(editedSubChapter.getSubcha
pter());

        while (chapterNumber.length()
< 2) {

            chapterNumber  = '0' +
chapterNumber;

        }

        while
(subchapterNumber.length() < 3) {

            subchapterNumber = '0' +
subchapterNumber;

        }

        String      key      = "ch" +
chapterNumber + "s" + subchapterNumber;

subChapterReff.child(key).child("title").setV
alue(newTitle);

mSubchapterAdapter.notifyDataSetChanged
();

```

```

editSubChapterDialog.dismiss();

        }

        }

    });

}

private void
showDeleteSubChapterDialog(SubChapter
subChapter) {

    final SubChapter deletedSubChapter =
subChapter;

    final AlertDialog yesNoDialog = new
AlertDialog.Builder(getActivity())

        .setTitle("Delete " +
deletedSubChapter.getTitle())

        .setMessage("Are you sure?")

        .setNegativeButton(R.string.cancel, null)

        .setPositiveButton("Yes", null)

        .setIcon(R.drawable.ic_warning_black_24dp
)

        .create();

    yesNoDialog.show();

yesNoDialog.getButton(AlertDialog.BUTT
ON_NEGATIVE).setOnClickListener(new
View.OnClickListener() {

    @Override

```

```

        public void onClick(View view) {
            yesNoDialog.dismiss();
        }
    });

    yesNoDialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener(new
    View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String number =
            String.valueOf(deletedSubChapter.getChapter());

            String subNumber =
            String.valueOf(deletedSubChapter.getSubchapter());

            while (number.length() < 2) {
                number = '0' + number;
            }

            while (subNumber.length() < 3) {
                subNumber = '0' + subNumber;
            }

            final String childNameSubChapter
            = "ch" + number + "s" + subNumber;

            subChapterReff.child(childNameSubChapter
            ).removeValue();

            mSubchapterAdapter.notifyDataSetChanged();

            yesNoDialog.dismiss();
        }
    });
}

@Override
public void onClick(View view) {
    if (view.getId() ==
    R.id.fab_tambah_chapter) {
        LayoutInflater inflater =
        this.getLayoutInflater();

        final View dialogView =
        inflater.inflate(R.layout.add_edit_chapter_la
        yout, null);

        new_sub_chapter_title =
        dialogView.findViewById(R.id.chapter_title
        );

        final AlertDialog
        addSubChapterDialog = new
        AlertDialog.Builder(getActivity())

        .setTitle(R.string.add_subchapter)

        .setPositiveButton(R.string.save, null)

        .setNegativeButton(R.string.cancel, null)

        .setView(dialogView)

```

```

        return;
    } else {
        final SubChapter
newSubChapter = new SubChapter();

subChapterReff.addListenerForSingleValue
Event(new ValueEventListener() {
    @Override
    public void
onDataChange(DataSnapshot dataSnapshot)
{
        int y = 1;
        int subchnum = y;
        for (DataSnapshot ds :
dataSnapshot.getChildren()) {
            SubChapter onIterSub
= ds.getValue(SubChapter.class);
            if
(onIterSub.getChapter() == chapterNumber)
{
                if
(onIterSub.getSubchapter() == y) {
                    y++;
                    subchnum = y;
                } else {
                    subchnum = y;
                }
            }
        }
    }
}

.setIcon(R.drawable.ic_playlist_add_black_
24dp)
        .create();

addSubChapterDialog.getWindow().setSoftI
nputMode(WindowManager.LayoutParams.
SOFT_INPUT_ADJUST_RESIZE);

addSubChapterDialog.show();

addSubChapterDialog.getButton(AlertDialo
g.BUTTON_NEGATIVE).setOnClickListener {
    @Override
    public void onClick(View view) {

addSubChapterDialog.dismiss();
    }
});

addSubChapterDialog.getButton(AlertDialo
g.BUTTON_POSITIVE).setOnClickListener {
    @Override
    public void onClick(View view) {
        if
(new_sub_chapter_title.getText().toSting().i
sEmpty()) {
            new_sub_chapter_title.setError("Sub chapter
title can't be empty");
        }
    }
}

```

```

        String chNum =
String.valueOf(chapterNumber);

        while (chNum.length() <
2) {

            chNum = '0' +
chNum;

        }

        String subchNumber =
String.valueOf(subchnum);

        while
(subchNumber.length() < 3) {

            subchNumber = '0' +
subchNumber;

        }

        String sub_chapter_title
=
new_sub_chapter_title.getText().toString();

        Log.d("TAG",
"onDataChange: "+ sub_chapter_title);

newSubChapter.setChapter(chapterNumber)
;

newSubChapter.setSubchapter(subchnum);

newSubChapter.setTitle(sub_chapter_title);

newSubChapter.setContent("");

subChapterReff.child("ch" + chNum + "s" +
subchNumber).setValue(newSubChapter);

```

```

    }

    @Override

    public void

onCancelled(DatabaseError databaseError) {

    }

});

mSubchapterAdapter.notifyDataSetChanged()

();

addSubChapterDialog.dismiss();

    }

}

});

}

}

}

```

SubChapterDetailFragment.java

```

package com.art.code.cplusplus.ui;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

```

```

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

public class SubChapterDetailFragment
extends Fragment{

    private SubChapter mSubChapter;

    public void setSubChapter(SubChapter
mSubChapter) {

        this.mSubChapter = mSubChapter;

    }

    @Nullable

    @Override

    public View onCreateView(LayoutInflater
inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {

        View v =
inflater.inflate(R.layout.detail_for_user,
container, false);

        TextView title = (TextView)
v.findViewById(R.id.detail_title);

        TextView content = (TextView)
v.findViewById(R.id.detail_content);

        title.setText(mSubChapter.getTitle());

        content.setText(mSubChapter.getContent());

        return v;

    }

}

```

SubChapterFragment.java

```

package com.art.code.cplusplus.ui;

import android.arch.lifecycle.Lifecycle;

import android.os.Bundle;

import android.support.annotation.Nullable;

import
android.support.design.widget.FloatingActio
nButton;

import android.support.v4.app.Fragment;

import
android.support.v7.widget.LinearLayoutMan
ager;

import
android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import
com.art.code.cplusplus.DataRepository;

import
com.art.code.cplusplus.Interface.ItemClickL
istener;

import
com.art.code.cplusplus.Model.Chapter;

import
com.art.code.cplusplus.Model.SubChapter;

import com.art.code.cplusplus.R;

```



```

import com.art.code.cplusplus.ViewHolder.SubChapterViewHolder;

import com.firebase.ui.database.FirebaseRecyclerAdapter;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;

public class SubChapterFragment extends Fragment {

    private FirebaseDatabase mDatabase;

    private DatabaseReference subchapterRef;

    private int chapterNumber;

    private RecyclerView listOfSubchapter;

    private RecyclerView.LayoutManager layoutManager;

    private FirebaseRecyclerAdapter<SubChapter, SubChapterViewHolder> mAdapter;

    public static SubChapterFragment newInstance(Chapter chapter) {

        SubChapterFragment subChapterFragment = new SubChapterFragment();

        Bundle args = new Bundle();

        args.putInt("ch", chapter.getChapter());

        subChapterFragment.setArguments(args);

        return subChapterFragment;
    }

    @Override

    public void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        mDatabase = FirebaseDatabase.getInstance();

        subchapterRef = mDatabase.getReference("SubChapters");

        Bundle args = getArguments();

        chapterNumber = args.getInt("ch");
    }

    @Nullable

    @Override

    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

        View subchapterView = inflater.inflate(R.layout.fragment_chapter_admin, container, false);

        FloatingActionButton fab = (FloatingActionButton) subchapterView.findViewById(R.id.fab_tambah_chapter);

        fab.setVisibility(View.GONE);

```

```

        listOfSubchapter = (RecyclerView)
subchapterView.findViewById(R.id.admin_
chapter_container);

        TextView title = (TextView)
subchapterView.findViewById(R.id.fragme
nt_title);

        title.setText("Sub Chapter");

        layoutManager = new
LinearLayoutManager(container.getContext(
));

listOfSubchapter.setLayoutManager(layout
Manager);

        loadSubChapter();

        return subchapterView;
    }

    private void loadSubChapter() {

        Query keyRef =
subchapterRef.orderByChild("chapter").equ
alTo(chapterNumber);

        mAdapter = new
FirebaseRecyclerViewAdapter<SubChapter,
SubChapterViewHolder>(

            SubChapter.class,
R.layout.admin_item,
SubChapterViewHolder.class, keyRef
) {

            @Override

            protected void
populateViewHolder(SubChapterViewHolde

```

```

r subChapterViewHolder, final SubChapter
subChapter, int i) {

            String name =
subChapter.getChapter() + "." +
subChapter.getSubchapter() + ". " +
subChapter.getTitle();

            subChapterViewHolder.sub_chapter_name.s
setText(name);

            subChapterViewHolder.edit_sub_chapter_bt
n.setVisibility(View.GONE);

            subChapterViewHolder.del_sub_chapter_bt
n.setVisibility(View.GONE);

            subChapterViewHolder.setItemClickListene
r(new ItemClickListener() {

                @Override

                public void onClick(View view,
int position, boolean isLongClick) {

                    if
(getLifecycle().getCurrentState()

.isAtLeast(Lifecycle.State.STARTED)) {

                        ((LearningActivity)
getActivity()).showSubChapterDetail(subCh
apter);

                    }

                }

            });

```

```

        }
    };

    mAdapter.notifyDataSetChanged();

    listOfSubchapter.setAdapter(mAdapter);
}
}

```

TestActivity.java

```

package com.art.code.cplusplus.ui;

import android.content.DialogInterface;

import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.widget.Toast;

import com.art.code.cplusplus.DataRepository;

import com.art.code.cplusplus.Model.Question;

import com.art.code.cplusplus.R;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError
;

```

```

import
com.google.firebase.database.DatabaseRefer
ence;

import
com.google.firebase.database.FirebaseDatab
ase;

import
com.google.firebase.database.ValueEventLis
tener;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

public class TestActivity extends
AppCompatActivity {

    private static final String TAG = "teg";

    public static List<Question>
listOfQuestions;

    FirebaseDatabase mDatabase;

    DatabaseReference mQuestionRef;

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_test);

        mDatabase =
DataRepository.getDBInstance();

        mQuestionRef =
mDatabase.getReference("Questions");

        listOfQuestions = new ArrayList<>();

```

```

loadQuestion();

if (savedInstanceState == null) {
    showStartMenu();
}

}

public void showStartMenu() {
    Fragment startTest =
StartTestFragment.newInstance();

getSupportFragmentManager().beginTransaction().add(R.id.test_container,
startTest).commit();
}

private void loadQuestion() {

mQuestionRef.addListenerForSingleValueEvent(new ValueEventListener() {

    @Override

    public void
onDataChange(DataSnapshot dataSnapshot)
{
        for (DataSnapshot ds :
dataSnapshot.getChildren()) {
            Question newQFromDb =
ds.getValue(Question.class);

listOfQuestions.add(newQFromDb);
        }
    }
}

}

@Override

public void
onCancelled(DatabaseError databaseError)
{}

});

}

public void showTestResult(int score) {
    Fragment resultFragment =
ResultFragment.newInstance(score);

getSupportFragmentManager().beginTransaction().replace(R.id.test_container,
resultFragment).commit();
}

public void startTest() {

Collections.shuffle(listOfQuestions);

    Fragment testFragment =
TestFragment.newInstance();

getSupportFragmentManager().beginTransaction().replace(R.id.test_container,
testFragment).commit();
}

@Override

public void onBackPressed() {

    Fragment currentFragment =
getSupportFragmentManager().findFragmentById(R.id.test_container);

    if (currentFragment instanceof
StartTestFragment) {

```

```

        super.onBackPressed();
    } else if (currentFragment instanceof
TestFragment) {
        AlertDialog alertDialog = new
AlertDialog.Builder(this)

.setTitle("Warning").setMessage("Score will
not saved,\nsure want to quit test?")

.setNegativeButton(R.string.cancel, new
DialogInterface.OnClickListener() {
            @Override
            public void
onClick(DialogInterface dialogInterface, int
i) {
                dialogInterface.dismiss();
            }
        })

        .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
            @Override
            public void
onClick(DialogInterface dialogInterface, int
i) {
                endTest();
            }
        })

        .create();
    }
}

```

```

    }
    public void endTest() {
        finish();
    }
}

```

TestFragment.java

```

package com.art.code.cplusplus.ui;

import android.arch.lifecycle.Lifecycle;
import android.graphics.Color;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ProgressBar;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

import
com.art.code.cplusplus.Model.Question;
import com.art.code.cplusplus.R;
import java.util.ArrayList;

```

```

import java.util.Collections;

import java.util.Enumeration;

import java.util.List;

import android.content.ContentValues.TAG;

public class TestFragment extends Fragment
implements View.OnClickListener,
RadioGroup.OnCheckedChangeListener {

    private List<Question> questionList;

    private List<String> answersList;

    private ProgressBar progressBar;

    private TextView progressBarText;

    private TextView testQuestion;

    private RadioButton answerA, answerB,
answerC, answerD;

    private RadioGroup answerGroup;

    private ImageButton before, next;

    private int questionSize;

    private int progressValue, index;

    private Button fin_btn;

    private View testView;

    public static TestFragment newInstance()
{
        TestFragment testFragment = new
TestFragment();

        return testFragment;
    }

```

```

@Override

    public void onCreate(@Nullable Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        questionList = new
ArrayList<Question>(TestActivity.listOfQu
estions);

        index = 0;

        progressValue = 0;

        answersList = new ArrayList<>();

        questionSize = questionList.size() > 50
? 50 : questionList.size();

        for (int l = 0; l < questionSize; l++) {

            answersList.add(l, "");

        }

    }

@Override

    public View onCreateView(LayoutInflater
inflater, ViewGroup container,

        Bundle
savedInstanceState) {

        testView =
inflater.inflate(R.layout.fragment_test,
container, false);

        progressBar = (ProgressBar)
testView.findViewById(R.id.progress_bar_t
est);

```

```

        progressBarText = (TextView)
testView.findViewById(R.id.progress_bar_t
ext);

        testQuestion = (TextView)
testView.findViewById(R.id.test_question);

        answerA = (RadioButton)
testView.findViewById(R.id.answer_a);

        answerB = (RadioButton)
testView.findViewById(R.id.answer_b);

        answerC = (RadioButton)
testView.findViewById(R.id.answer_c);

        answerD = (RadioButton)
testView.findViewById(R.id.answer_d);

        answerGroup = (RadioGroup)
testView.findViewById(R.id.answer_group)
;

        before = (ImageButton)
testView.findViewById(R.id.test_before);

        next = (ImageButton)
testView.findViewById(R.id.test_next);

        fin_btn = (Button)
testView.findViewById(R.id.test_finish);

        before.setOnClickListener(this);

        next.setOnClickListener(this);

        fin_btn.setOnClickListener(this);

answerGroup.setOnCheckedChangeListener
(this);

        runTest();

        return testView;
    }

    private void runTest() {

        if (index == 0) {

            before.setVisibility(View.GONE);

        } else {

            before.setVisibility(View.VISIBLE);

        }

        if (index == questionSize - 1) {

            next.setVisibility(View.GONE);

            fin_btn.setVisibility(View.VISIBLE);

        } else {

            next.setVisibility(View.VISIBLE);

            fin_btn.setVisibility(View.GONE);

        }

        testQuestion.setText(questionList.get(index)
.getQuestion());

        List<String> answers = new
ArrayList<String>();

        answers.add(questionList.get(index).getAns
wera());

        answers.add(questionList.get(index).getAns
werb());

        answers.add(questionList.get(index).getAns
werc());
    }

```

```

answers.add(questionList.get(index).getAnswer());
Collections.shuffle(answers);
answerA.setText(answers.get(0));
answerB.setText(answers.get(1));
answerC.setText(answers.get(2));
answerD.setText(answers.get(3));
if (!answersList.get(index).equals("")) {
    answerA.setEnabled(false);
    answerB.setEnabled(false);
    answerC.setEnabled(false);
    answerD.setEnabled(false);
    String storedAnswer =
answersList.get(index);
    for (int r = 0; r <
answerGroup.getChildCount(); r++) {
        RadioButton rdb = (RadioButton)
answerGroup.getChildAt(r);
        if
(rdb.getText().toString().equals(storedAnswer)) {
            rdb.setChecked(true);
            if
(!storedAnswer.equals(questionList.get(index).getAnswera())) {
                for (int l=0; l <
answerGroup.getChildCount(); l++) {
                    RadioButton rbt =
(RadioButton) answerGroup.getChildAt(l);
                    if
(rbt.getText().toString().equals(questionList.get(index).getAnswera())) {
                        wrongAnsSelected(rdb,
rbt);
                    }
                }
            }
        } else {
            answerA.setEnabled(true);
            answerB.setEnabled(true);
            answerC.setEnabled(true);
            answerD.setEnabled(true);
        }
        progressValue = index + 1;
        progressBar.setMax(questionSize);
        progressBar.setProgress(progressValue);
        String valueText = progressValue + "/"
+ questionSize;
        progressBarText.setText(valueText);
    }
}
@Override

```



```

public void onClick(View view) {
    switch (view.getId()) {
        case R.id.test_next:
            if
(answerGroup.getCheckedRadioButtonId()
!= -1) {
                int    selectedRadioId    =
answerGroup.getCheckedRadioButtonId();
                String    selectedAnswer    =
((RadioButton)
testView.findViewById(selectedRadioId)).g
getText().toString();
                answersList.set(index,
selectedAnswer);
            }
            index++;
            answerGroup.check(-1);
            resetRB();
            runTest();
            break;
        case R.id.test_before:
            if
(answerGroup.getCheckedRadioButtonId()
!= -1) {
                int    selectedRadioId    =
answerGroup.getCheckedRadioButtonId();
                String    selectedAnswer    =
((RadioButton)
testView.findViewById(selectedRadioId)).g
getText().toString();
                answersList.set(index,
selectedAnswer);
            }
            index--;
            answerGroup.check(-1);
            resetRB();
            runTest();
            break;
        case R.id.test_finish:
            if
(!answersList.get(index).equals("")) {
                String    storedAnswer    =
answersList.get(index);
                for (int r = 0; r <
answerGroup.getChildCount(); r++) {
                    RadioButton    rdb    =
(RadioButton) answerGroup.getChildAt(r);
                    if
(rdb.getText().toString().equals(storedAnsw
er)) {
                        rdb.setChecked(true);
                    }
                }
            }
            if
(answerGroup.getCheckedRadioButtonId()
!= -1) {
                int    selectedRadioId    =
answerGroup.getCheckedRadioButtonId();

```

```

        String selectedAnswer =
((RadioButton)
testView.findViewById(selectedRadioId)).get
getText().toString();

        answersList.set(index,
selectedAnswer);
    }

    int score = calculatesscore();

    score = (int) (((double) score
/(double) questionSize ) * 100);

    if (getLifecycle().getCurrentState()
.isAtLeast(Lifecycle.State.STARTED)) {

        ((TestActivity)
getActivity()).showTestResult(score);
    }

    break;
}

private void resetRB() {

answerA.setBackgroundColor(Color.WHIT
E);

answerB.setBackgroundColor(Color.WHIT
E);

answerC.setBackgroundColor(Color.WHIT
E);

```

```

answerD.setBackgroundColor(Color.WHIT
E);

    answerA.setTextColor(Color.BLACK);
    answerB.setTextColor(Color.BLACK);
    answerC.setTextColor(Color.BLACK);
    answerD.setTextColor(Color.BLACK);
}

private int calculatesscore() {

    int correctAnswer = 0;

    int idx = 0;

    int maxidx = questionSize;

    while (idx < maxidx) {

        if
(answerList.get(idx).equals(questionList.ge
t(idx).getCorrectAnswer())) {

            correctAnswer++;

        }

        idx++;

    }

    return correctAnswer;
}

private void wrongAnsSelected(RadioButton rbf,
RadioButton rbt) {

    rbf.setBackgroundColor(getResources().get
Color(R.color.red));

```

```

        rbf.setTextColor(Color.WHITE);
    }

    private void trueAnsSelected(RadioButton
    rbt) {

        rbt.setBackgroundColor(getResources().get
        Color(R.color.green));

        rbt.setTextColor(Color.WHITE);
    }

    private void trueAnsSelected(RadioButton
    rbt) {

        rbt.setBackgroundColor(getResources().get
        Color(R.color.green));

        rbt.setTextColor(Color.WHITE);
    }

    @Override

    public void
    onCheckedChanged(RadioGroup
    radioGroup, int id) {

        if (id == R.id.answer_a) {

            resetRB();

            if
            (checkAnswer().equals(answerA.getText().t
            oString())) {

                trueAnsSelected(answerA);
            } else {

                String corr = checkAnswer();

                for (int r = 0; r <
                answerGroup.getChildCount(); r++) {

                    RadioButton rdb =
                    (RadioButton) answerGroup.getChildAt(r);

                    if
                    (rdb.getText().toString().equals(corr)) {

                        wrongAnsSelected(answerB,
                        rdb);
                    }
                }
            }
        }
    }

```

```

        disableRadioBtn();
    }

    if (id == R.id.answer_c) {

        resetRB();

        if
        (checkAnswer().equals(answerC.getText().t
        oString())) {

            trueAnsSelected(answerC);

        } else {

            String corr = checkAnswer();

            for (int r = 0; r <
            answerGroup.getChildCount(); r++) {

                RadioButton rdb =
                (RadioButton) answerGroup.getChildAt(r);

                if
                (rdb.getText().toString().equals(corr)) {

                    wrongAnsSelected(answerC,
                    rdb);

                }

            }

            disableRadioBtn();

        }

        if (id == R.id.answer_d) {

            resetRB();

            if
            (checkAnswer().equals(answerD.getText().t
            oString())) {

                trueAnsSelected(answerD);

            } else {

                String corr = checkAnswer();

                for (int r = 0; r <
                answerGroup.getChildCount(); r++) {

                    RadioButton rdb =
                    (RadioButton) answerGroup.getChildAt(r);

                    if
                    (rdb.getText().toString().equals(corr)) {

                        wrongAnsSelected(answerD,
                        rdb);

                    }

                }

                disableRadioBtn();

            }

        }

        private String checkAnswer() {

            Question q = questionList.get(index);

            return q.getCorrectAnswer();

        }

        private void disableRadioBtn() {

            answerA.setEnabled(false);

            answerB.setEnabled(false);

            answerC.setEnabled(false);

            answerD.setEnabled(false);

        }
    }
}

```