Design specification and data model for - Edu game Structured language - “Strukturēta valoda”- (STRL)

Team1
Vasilijs Malisevs
Roberts Pilaps
Anda Rozenfelde
Aygul Ismaiylova
Ieva Grada

Contents

Introduction 3
Requirements for development environment: 3
Requirements for development: 3
Design of code 5
Code examples 8
Design of educational game: 9
Icon 11
Technologies & databases 12
Model layer 12
Database design: 12
External database for the game 12
Internal database 17
Controller layer 17
View layer 17
Use cases 18
Chooses topic 19
Constructs phrases 19
Congratulation windows 20

Introduction

Educational game describes structured languages phrases in a game view for kids and adolescents with delay with speech development. The game consists of 140 phrases. The games phrases are based on structured language.

Requirements for development environment:

Issue Management system:	Slack	https://app.slack.com/client/TTQML6UUA/learning-slack
To do list:	GitHub	https://github.com/bazyl95/Structured-Language/projects/1
Code versioning system:	GitHub	https://github.com/bazyl95/Structured-Language/projects/1
Development IDE and Unit test IDE	Android Studio 3.5	https://developer.android.com/studio

Requirements for development:

Development intended to use Android Studio 3.5 for development, unit tests and instrumented tests. Android Studio uses MVC architecture. MVC Architecture uses Model in bottom layer (data), Controller in the middle layer (business logic of application) and View in the frontend layer (screen shoots).

Libraries used:

```
implementation 'androidx.appcompat:appcompat:1.1.0'
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'androidx.test:runner:1.2.0'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
implementation 'androidx.recyclerview:recyclerview:1.2.0-alpha01'
implementation 'com.googlecode.json-simple:json-simple:1.1'
implementation 'com.squareup.okhttp3:okhttp:4.4.0'
```

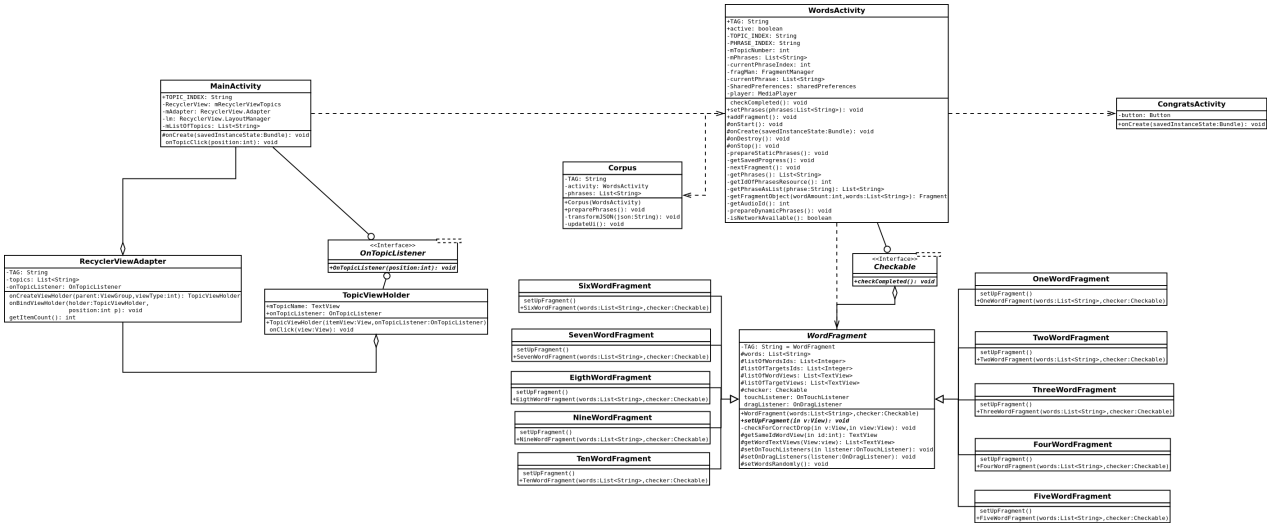
Libraries for test used:

```
testImplementation 'junit:junit:4.12'
testImplementation 'com.android.support.test:rules:0.5'
testImplementation 'com.android.support.test:runner:0.5'
androidTestImplementation 'androidx.test:runner:1.2.0'
androidTestImplementation('com.android.support.test.espresso:espresso-contrib:2.2') {
    // Necessary to avoid version conflicts
```

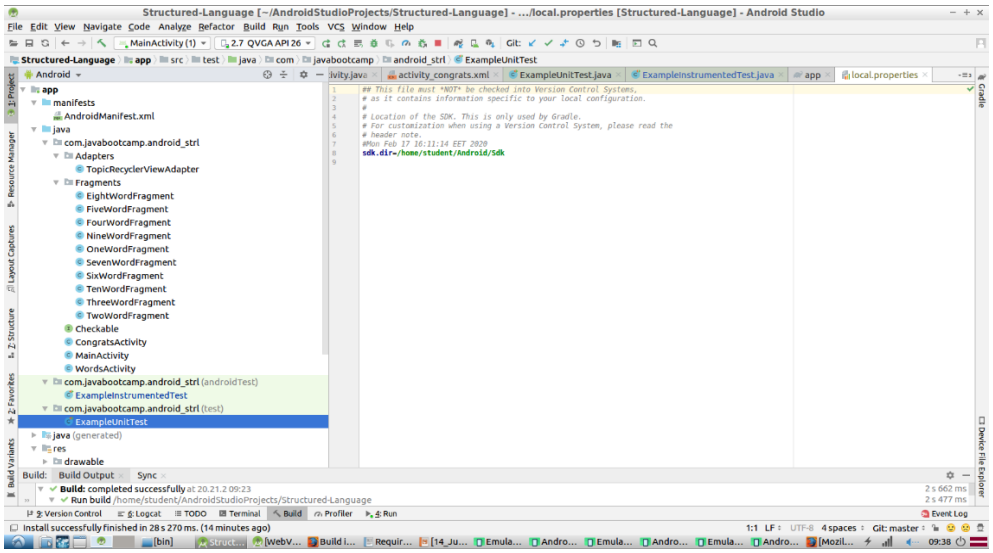
```
        exclude group: 'com.android.support', module: 'appcompat'
        exclude group: 'com.android.support', module: 'support-v4'
    exclude group: 'com.android.support', module: 'support-annotations'
                                    exclude module: 'recyclerview-v7'
                                                                    }
androidTestImplementation                       'androidx.test.espresso:espresso-core:3.1.0'
androidTestImplementation 'com.android.support.test:rules:1.0.2'
```

Design of code

Controller layer code – fragments and main class.The code is divided in ten fragments due to maximum 10 words in phrase. There are MainActivity, WordsActivity and CongratsActivity according to 3 use cases.

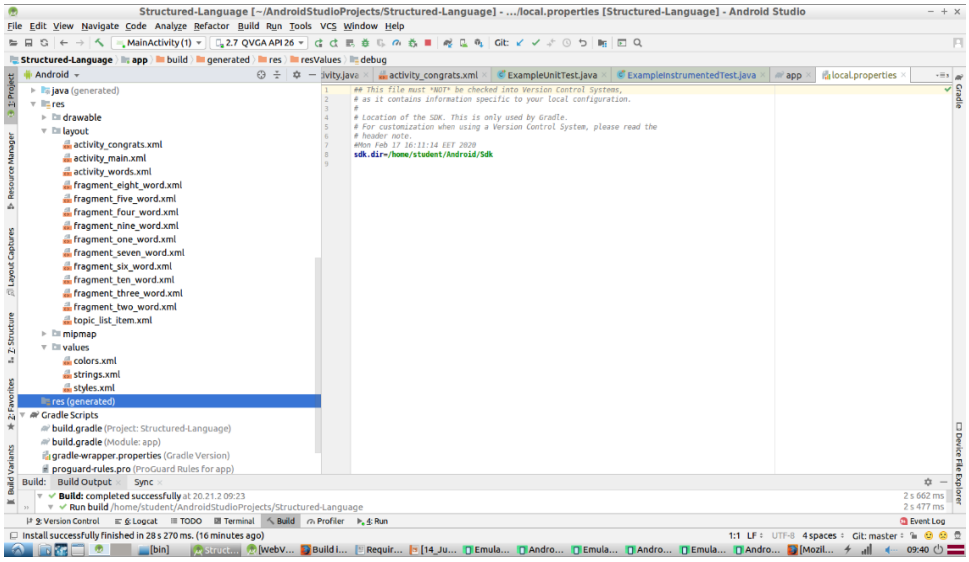


Picture 1 - class diagram



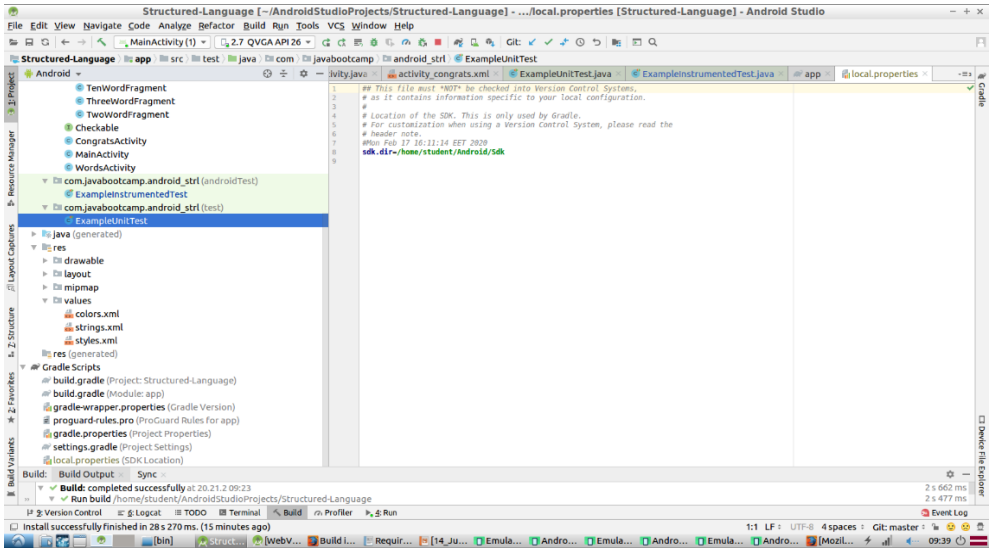
Picture 2 - code structure for Controller

View part – frontend of application. There are activity_main, activity_words and activity_congrats layouts according to 3 use cases. There are fragments layouts for each phrase from one to ten words.



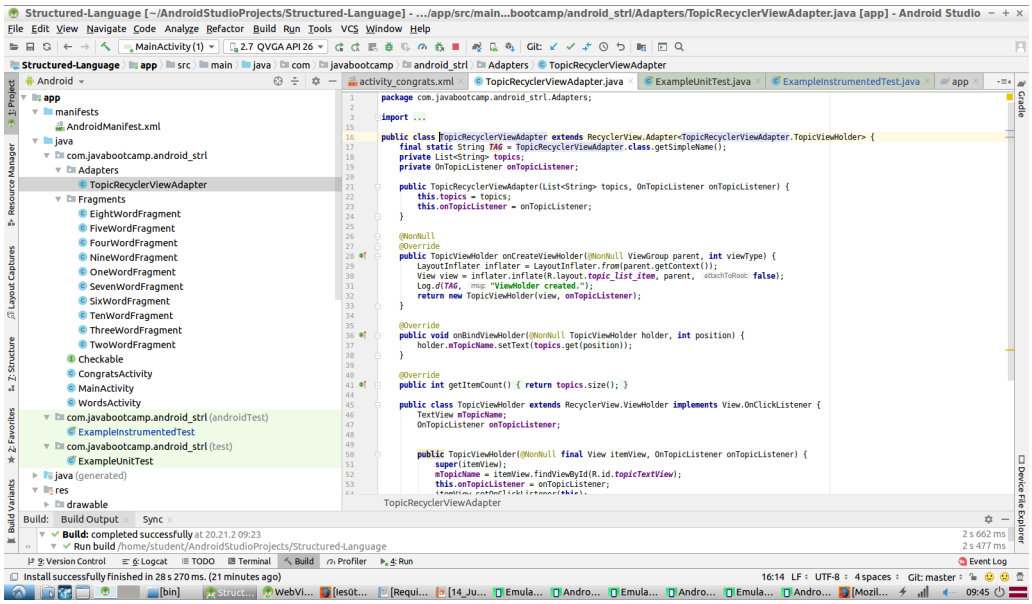
Picture 3 - code structure for View

Model layer and test part. Strings and styles storage.

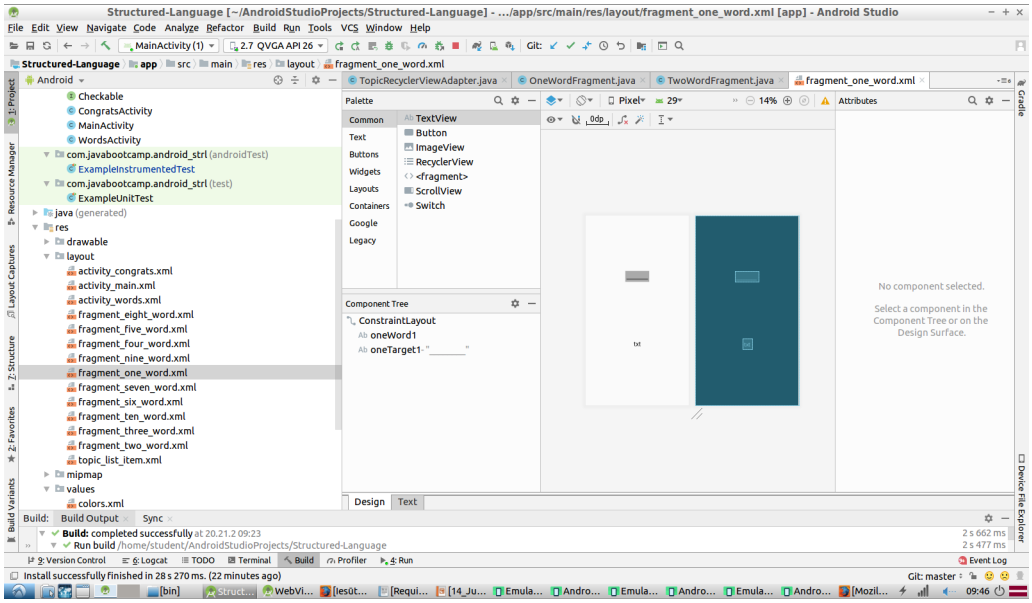


Picture 4 - code structure for Model

Code examples

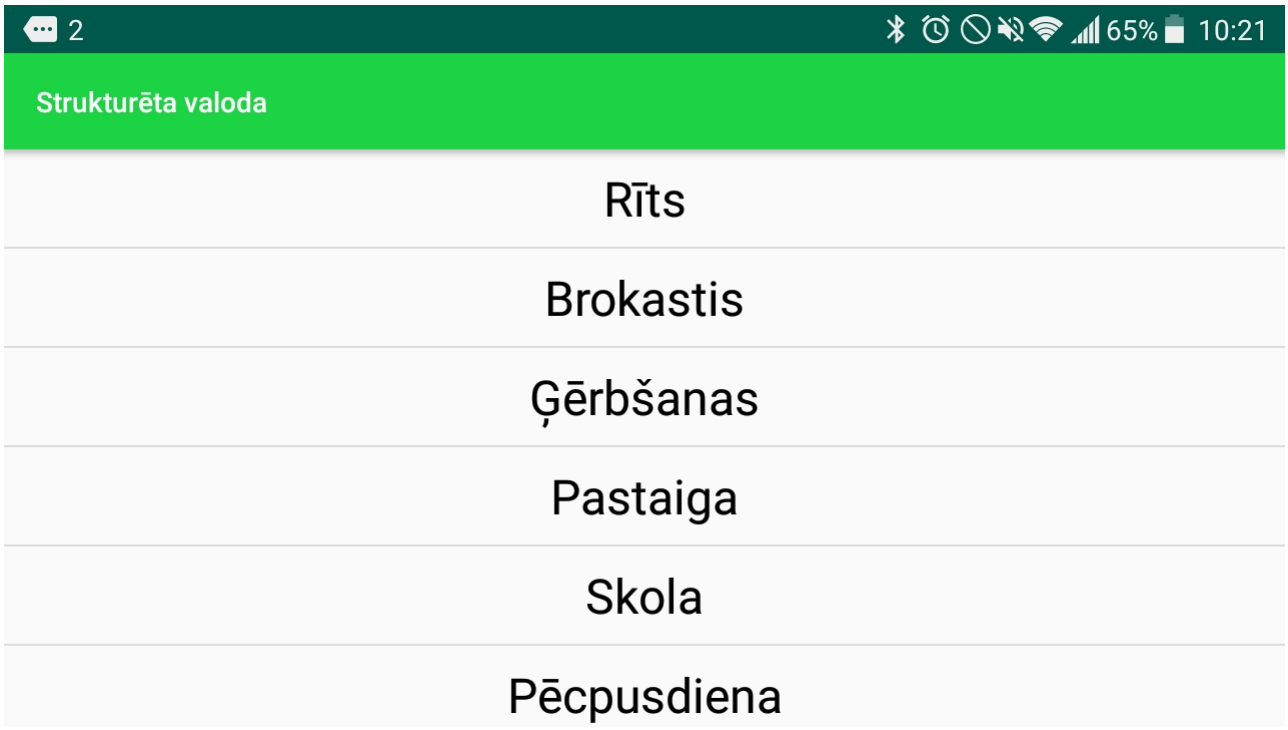


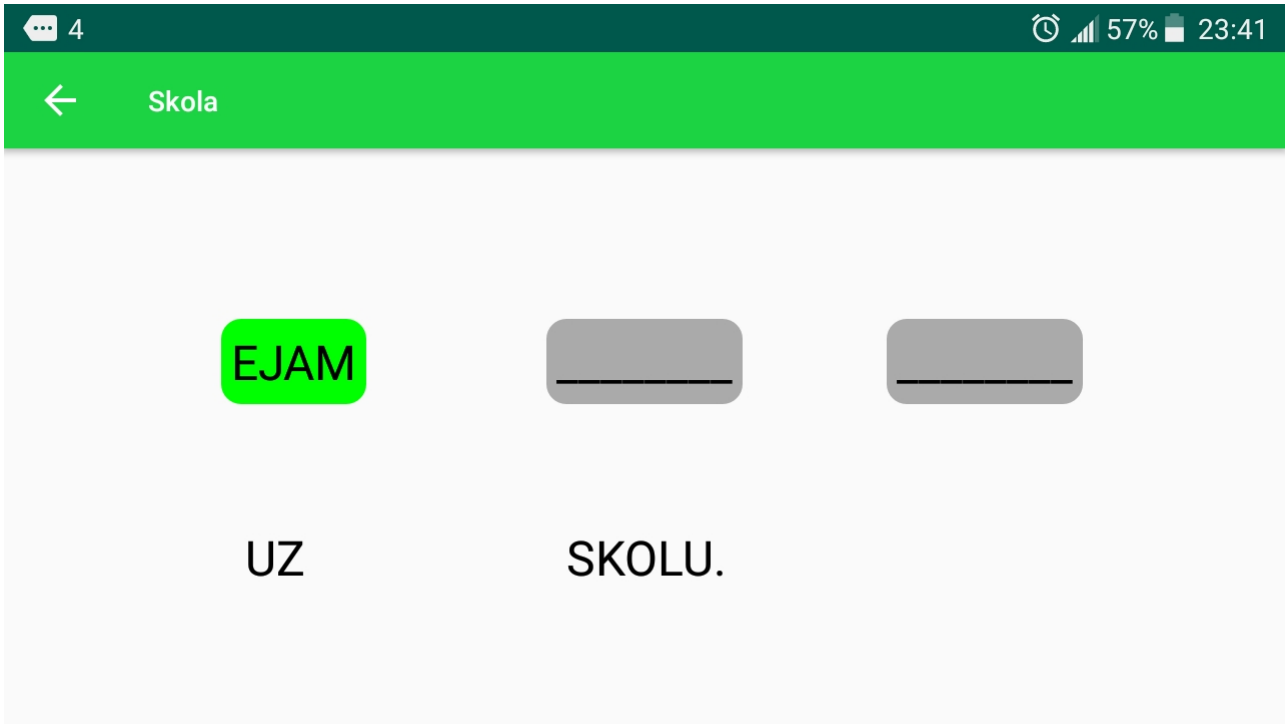
Picture 5

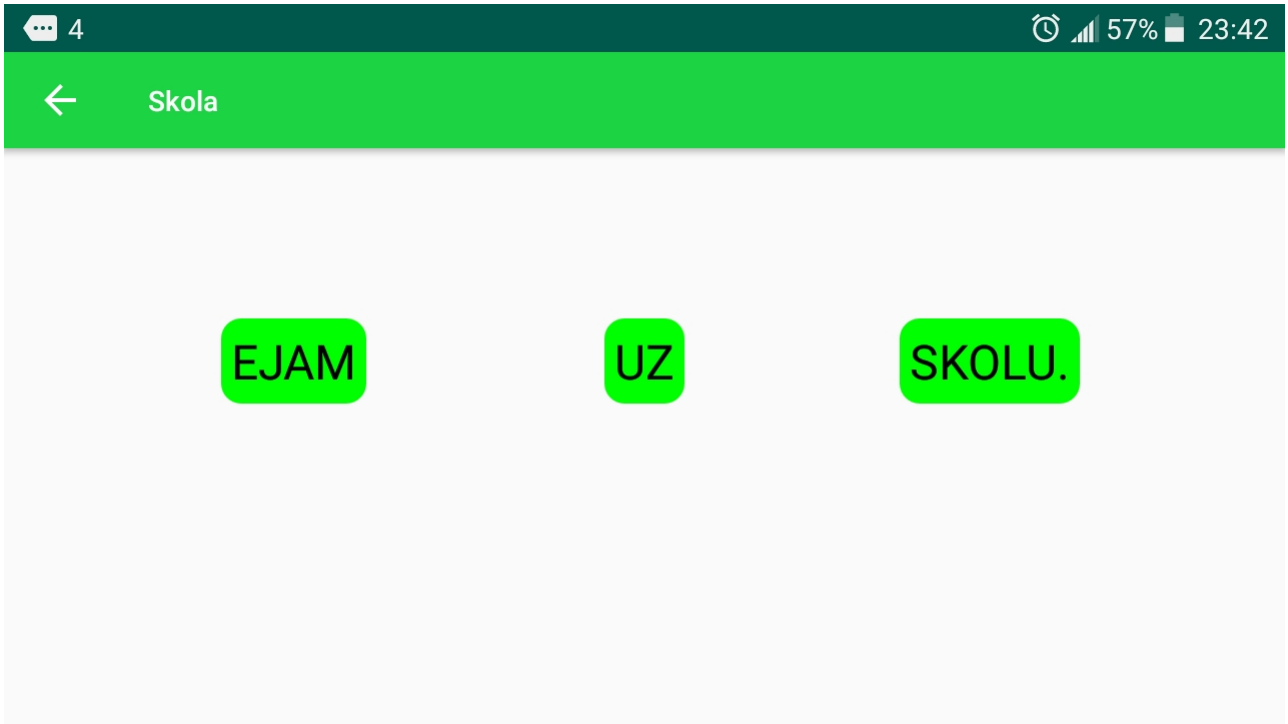


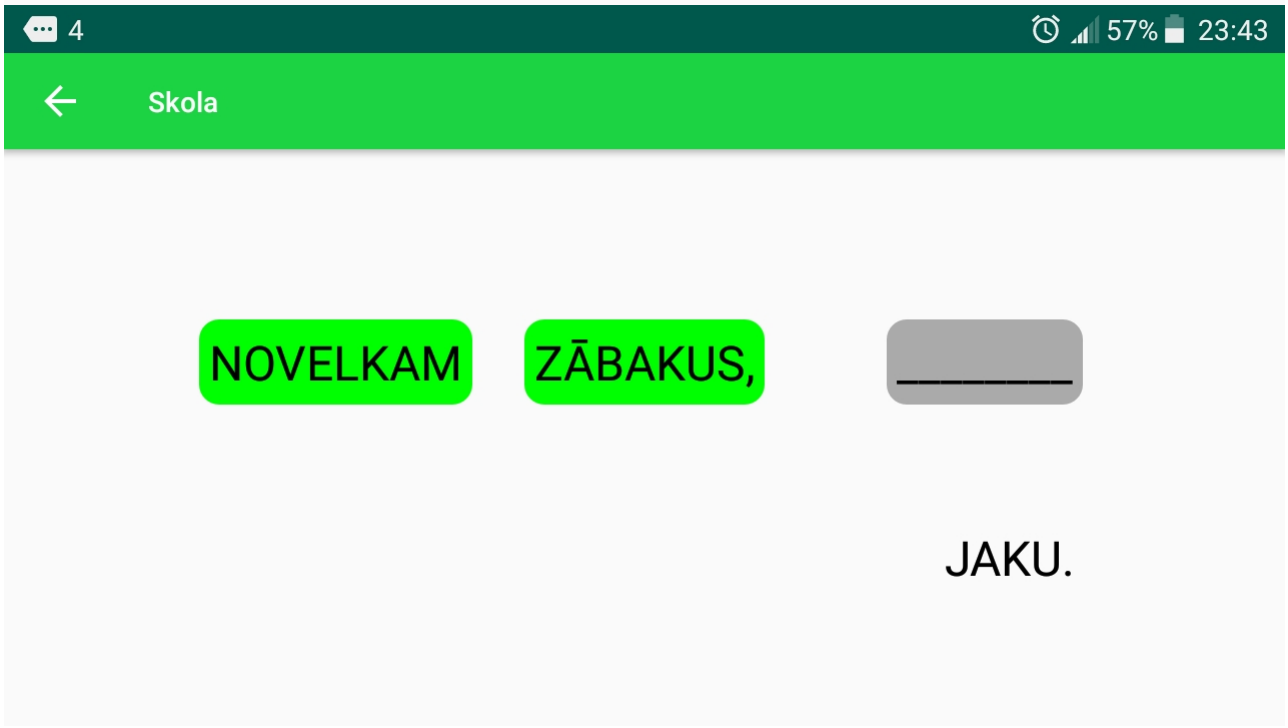
Picture 6

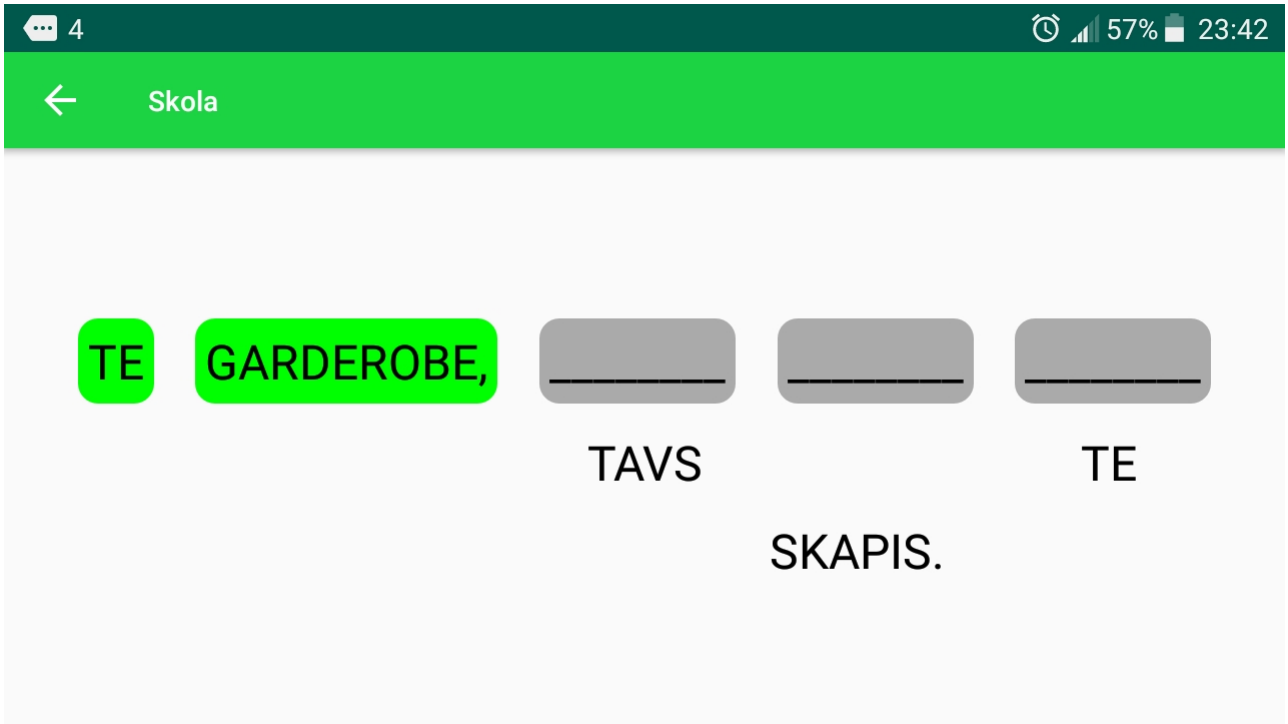
Design of educational game:

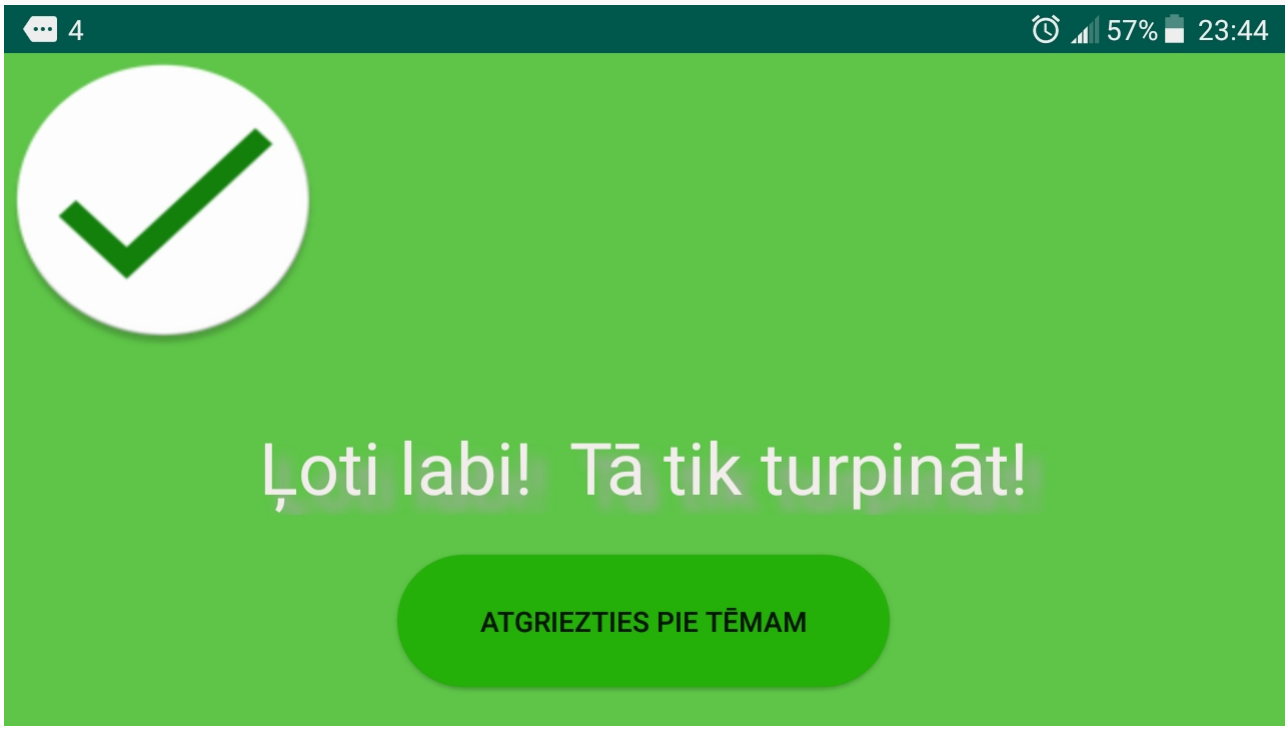





4
57% 23:42
Skola
EJAM
UZ
SKOLU.


4
57% 23:43
Skola
NOVELKAM
ZĀBAKUS,
_____
JAKU.





Icon

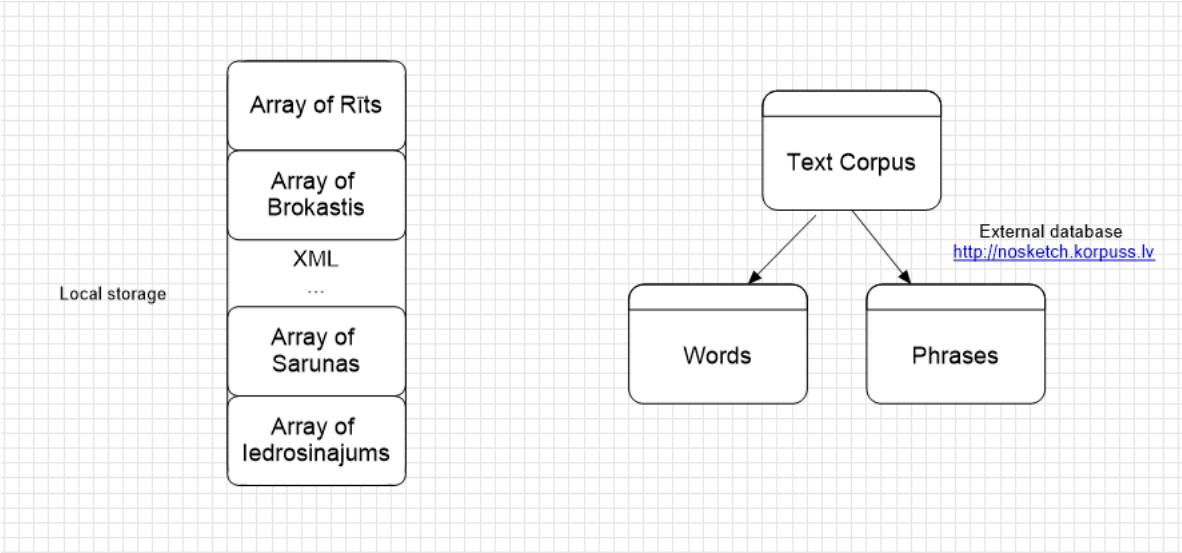
Icon is created in Android Asset Studio. In Asset Studio icon formats are defined for different phones.

Technologies & databases

Model layer

Database design:



Picture 7

External database for the game

Search tags in LVK2018	https://www.clarin.lv/attachments/LVK2018_meklesana.pdf
The game intends to work with external database:	http://nosketch.korpuss.lv/run.cgi/first_form
The full list of databases are:	http://www.korpuss.lv/id/LVTB
Document about word database structure	https://www.apgads.lu.lv/fileadmin/user_upload/lu_portal/apgads/PDF/Valoda-nozime-forma/VNF-10/vnf_10-12_Levane_Petrova.pdf
Lemma definition	https://en.wikipedia.org/wiki/Lemma
Word frequency documentation	https://odo.lv/Blog/160513

How to connect to external database

```
package com.javabootcamp.android_strl.externalModel;

import android.util.Log;

import com.javabootcamp.android_strl.WordsActivity;

import org.jetbrains.annotations.NotNull;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

public class Corpus {
    private static final String TAG = Corpus.class.getSimpleName();
    private WordsActivity activity;
    private List<String> phrases;

    public Corpus(WordsActivity activity) {
        this.activity = activity;
    }

    public void preparePhrases() {
        String url =
"http://nosketch.korpuss.lv/run.cgi/view?q=atag%2C[lemma%3D%22vi%C5%86%C5%A1%22|lemma%3D%22vi%C5%86a%22|lemma%3D%22es%22][tag%3D%22v...p.*%22][lemma%3D%22m%C4%81ja%22|lemma%3D%22skola%22];fromp=1;corpname=LVK2018&attrs=word%2Ctag%2Clemma&ctxattrs=word&structs=p%2Cg&refs=%3Ddoc.id;format=json";
        OkHttpClient client = new OkHttpClient();
        Request req = new Request.Builder()
                .url(url)
                .build();

        Call call = client.newCall(req);
        call.enqueue(new Callback() {
            @Override
            public void onFailure(@NotNull Call call, @NotNull IOException e)
```

```
{
                Log.e(TAG, "Response failed.", e);
            }

            @Override
            public void onResponse(@NotNull Call call, @NotNull Response
response) throws IOException {
                String json = response.body().string();
                if (response.isSuccessful()) {
                    transformJSON(json);
                } else {
                    Log.e(TAG, "Response was not succesfull");
                    activity.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            activity.finish();
                        }
                    });
                }
            }
        });
    }

    private void transformJSON(String json) {
        JSONParser parser = new JSONParser();
        phrases = new ArrayList<>();
        try {
            JSONObject jObject = (JSONObject) parser.parse(json);
            JSONArray lineItems = (JSONArray) jObject.get("Lines");
            for (Object lineObject : lineItems) {
                JSONObject jsonLineItem = (JSONObject) lineObject;
                JSONArray kwic = (JSONArray) jsonLineItem.get("Kwic");
                String phrase = "";
                for (int i = 0; i < kwic.size(); i += 2) {
                    JSONObject kwicItem = (JSONObject) kwic.get(i);
                    phrase += kwicItem.get("str").toString();
                }
                phrase = phrase.toUpperCase();
                if (phrase.charAt(0) == ' ') {
                    StringBuilder sb = new StringBuilder();
                    for (int i = 1; i < phrase.length(); i++) {
                        sb.append(phrase.charAt(i));
                    }
                    phrase = sb.toString();
                }
                if (!phrases.contains(phrase)) {
                    phrases.add(phrase);
                }
            }
            updateUi();
        } catch (ParseException e) {
            Log.e(TAG, "JSON Exception caught: ", e);
```

```
        } catch (Exception e) {
            Log.e(TAG, "Exception caught: ", e);
        }
    }

    private void updateUi() {
        activity.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                activity.setPhrases(phrases);
                activity.getSavedProgress();
                activity.addFragment();
            }
        });
    }
}
```

Returned JSON String from external database request.


```
{
  "concsize": 6697,
  "gdex_scores": [],
  "numofpages": 335,
  "finished": 1,
  "api_version": "open-3.105.1",
  "Sort_idx": [],
  "q": ["aword,[lc=\"roka\" | lemma_lc=\"roka\"]"],
  "Desc": [{
    "op": "Query",
    "tourl": "q=aword%2C%5Blc%3D%22roka%22+%7C+lemma_lc%3D%22roka%22%5D",
    "size": 6697,
    "nicearg": "roka",
    "arg": "[lc=\"roka\" | lemma_lc=\"roka\"]",
    "rel": 544.95
  }],
  "fromp": 1,
  "fullsize": 6697,
  "port": 45648,
  "sc_strcts": [
    [
      "doc",
      "Dokuments"
    ],
    [
      "p",
      "Rindkopa"
    ],
    [
```

```
      "s",
      "Teikums"
    ],
    [
      "g",
      "g"
    ]
  ],
  "nextlink": "fromp=2",
  "relsize": 545,
  "lastlink": "fromp=335",
  "Lines": [
    {
      "Tbl_refs": ["s31"],
      "ref": "s31",
      "leftsize": 46,
      "leftspace": "    ",
      "Left": [{
        "str": "ka tas viss koncentrējoties tikai izpildvaras",
        "class": ""
      }],
      "hitlen": "",
      "Kwic": [{
        "str": " rokās",
        "class": "col0 coll"
      }],
      "toknum": 4693,
      "Right": [
        {
          "str": ".",
          "class": ""
        },
        {
          "str": "<\/p><p>",
          "class": "strc"
        },
        {
          "str": "Līdzīgu viedokli pauda arī Latvijas",
          "class": ""
        }
      ],
      "rightspace": "            ",
      "linegroup": "_",
      "rightsize": 38
    },
    {
      "Tbl_refs": ["s36"],
```

```
"ref": "s36",
"leftsize": 49,
"leftspace": "",
"Left": [{
  "str": "pārstāvji, gan arī cilvēki jau ar sankcijām \u201e uz",
  "class": ""
}],
"hitlen": "",
"Kwic": [{
  "str": " rokām",
  "class": "col0 coll"
}], …
```


Internal database

Data can be stored as:

Shared preferences	Store private, primitive data in key-value pairs.
Internal (private) storage	Store files that are meant for your app's use only, either in dedicated directories within an internal storage volume or different dedicated directories within external storage. Use the directories within internal storage to save sensitive information that other apps shouldn't access.Data should be written on the Android file system
External storage	Connection to external database using API.
Sqlite database	Store structured data in a private database using the Room persistence library.

Controller layer

Implementation on recycleview for list of topics in edu game.	https://developer.android.com/guide/topics/ui/layout/recyclerview

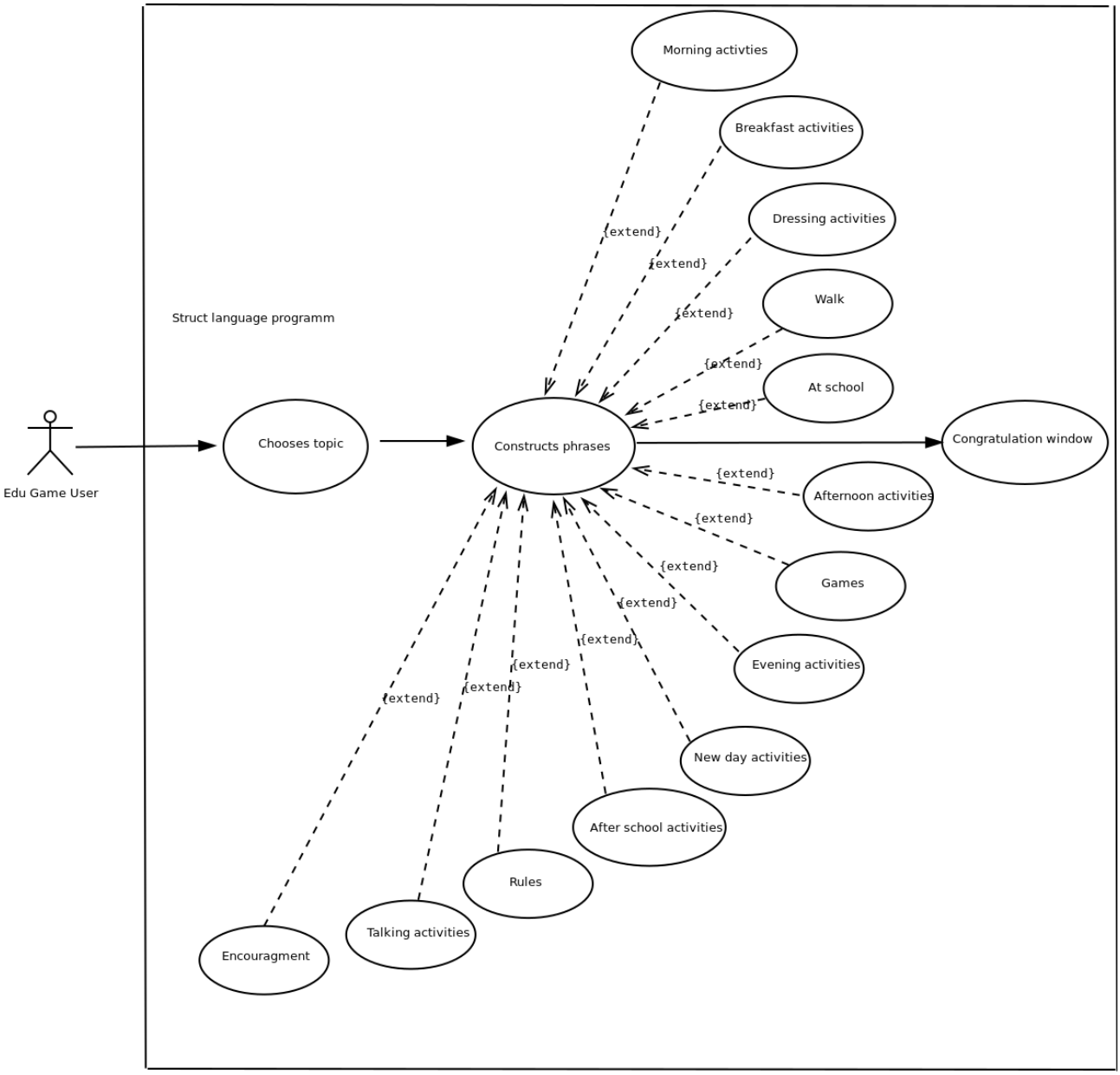
View layer

Fragments usage to organise the frontend phrases scope on the screen	https://developer.android.com/guide/components/fragments

Drag and drop interaction	https://developer.android.com/guide/topics/ui/drag-drop https://stackoverflow.com/questions/37549806/drag-and-drop-view-not-visible-after-drag-and-drop-finished
Constraint layout usage	https://developer.android.com/training/constraint-layout

Use cases

There are 3 activities realized in all application (for use case – Choose topic and Congratulation window). The middle Activity is realized using Fragments (use case – Constructs phrases).



Chooses topic

UseCase realized using Activity with RecyclerView in View layer. The app should show the topics,the app needs to display a scrolling list of elements.The RecyclerView widget is a more advanced and flexible version of ListView.

As internal data storage is used string.xml file.

Constructs phrases

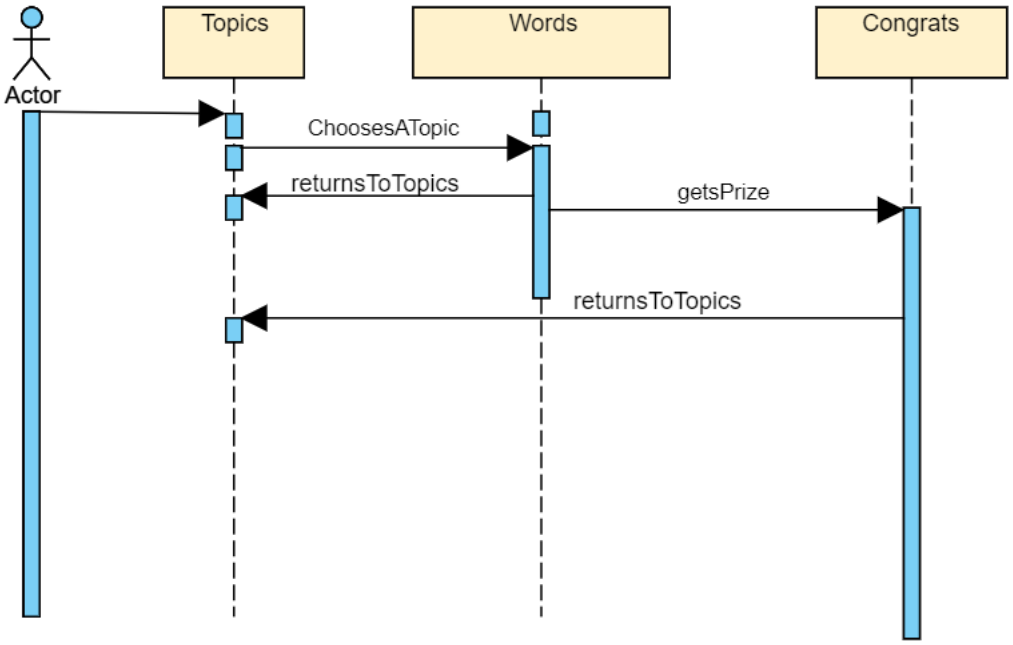
Use case realized using Activity with Fragments. A Fragment is a piece of an application's user interface or behavior that can be placed in an Activity. Interaction

with fragments is done through FragmentManager, which can be obtained via Activity#getFragmentManager() and Fragment#getFragmentManager(). All following Use cases are realized through Fragments.

Morning activities, Breakfast activities, Dressing activities, Walk, At school, Afternoon activities, Games, Evening activities, New day activities, After school activities, Rules, Talking activities, Encouragement.

Congratulation windows

An Activity is realized as congratulation windows. In congratulation windows there are congratulations for the ame player.



Picture 9