Design Document for «Kids learning» project

Author: Group 5

Auyeskhan Elmira

KurmanzhankyzyAkmaral

AbiyrMaira

Version	Date	Author	Change
0.1	12.03.2016	SM	InitialDocument

TableofContents

1. Introduction

1.1. PURPOSE

In this document, written description of the mobile and web application «Kids learning».

1.2. SCOPE

Our program for mobile and web application, associated with the server and a database.

1.3. DEFINITIONS, ACRONYMS, ABBREVIATIONS

Term	Description
Teacher	person who will be able to teach kids with help program (web andmobile)
Kids	type of the users who learn the materials
Game	kids can test what they learned
Questions	it will in part of game
Pictures	with help pictures kids can choose correct answer
Sound	button that read the questions

Grade	showsresultsofgame

1.4. DESIGN GOALS

1. Reliability. All the data on the database, and the server must be maintained and reliable.
2. Maintainability. If an error occurs on the server, it automatically has to report it and to repair.
3. Extensibility. Anyone who has a smart phone and the internet will be able to use our program.

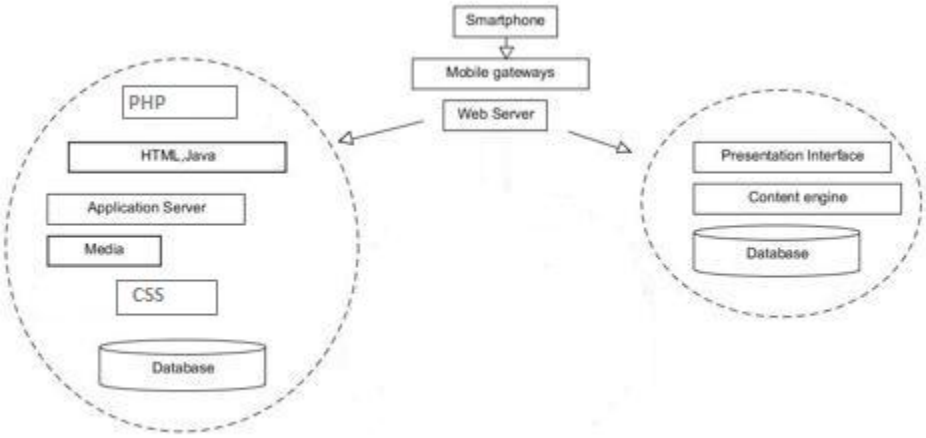
2. References

(Ifany)

3 Decomposition Description

3.1 MODULE DECOMPOSITION

Architectural diagram



3.1.1 InternetUsers (Smartphone) Description.

Anyone who has an Internet connection or if hasn't can learn any groups of this application(alphabet,colours,numbers) and can play the game!

3.1.2 WebServerDescription.

All data will be placed in the mobile applications and web applications. And user can find the information he/she was looking for or that it became interesting.

3.1.3 JSON ParsingDescription.

With this application we can all the data that shows on the site, show the same on a mobile application.

3.1.4 DatabaseDescription.

All data is stored in the database. Regarding the request to the server, the database provides data.

3.2 CONCURRENT PROCESS

3.2.1 InternetUsers (Smartphone) Description.

1. When you open the app the user can see the main page.
2. The user selects which wants to learn(training).
3. The user can check yourself playing the game(testing).
4. The user also can go back and choose next terms to learn.

3.2.2 WebServerDescription.

All data will be placed in the web application. There, the user can find the information for what looking for or that it became interesting.

3.2.3 JSON ParsingDescription.

With this application we can all the data that shows on the site, show the same on a mobile application.

3.2.4 DatabaseDescription.

All data is stored in the database. Regarding the request to the server, the database provides data.

3.3 DATA DECOMPOSITION

3.3.1 <Class 1>Description

3.3.2 <Class 2>Description

3.4 STATES

3.4.1 <State/System 1 >Description

3.4.2 <State/System 2>Description

4 DependencyDescription

4.1 INTERMODULE DEPENDENCIES

4.2 INTERPROCESS DEPENDENCIES

4.3 DATA DEPENDENCIES

5 InterfaceDescription

5.1 MODULE INTERFACE

5.1.1 WEB appcodeinterface

5.1.1.1 CreatingOurDatabase

First we are going to create our database which stores our data.

To create a database:

```
CREATE TABLE IF NOT EXISTS `member` (
  `mem_id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `fname` varchar(30) NOT NULL,
  `lname` varchar(30) NOT NULL,
  `address` varchar(100) NOT NULL,
  `contact` varchar(30) NOT NULL,
  `picture` varchar(100) NOT NULL,
  `gender` varchar(10) NOT NULL,
  PRIMARY KEY(`mem_id`)
) ENGINE=MyISAMDEFAULT CHARSET=latin1 AUTO_INCREMENT=3;
```

Creating Our Form

Next step is to create a form and save it as index.php. To create a form, open your HTML code editor and paste the code below in the upper part of the document or above the html tag. The code below is use to disallow the user to go back as if it still logged in.

```
<?php
	//Start session
	session_start();
	//Unset the variables stored in session
	unset($_SESSION['SESS_MEMBER_ID']);
	unset($_SESSION['SESS_FIRST_NAME']);
	unset($_SESSION['SESS_LAST_NAME']);
?>
Paste the code bellow after the body tag of the HTML document
<form name="loginform" action="login_exec.php" method="post">
<table width="309" border="0" align="center" cellpadding="2" cellspacing="5">
<tr>
<td colspan="2">
		<!--the code bellow is used to display the message of the input
validation-->
		<?php

	if(isset($_SESSION['ERRMSG_ARR'])&&is_array($_SESSION['ERRMSG_ARR'])&&c
ount($_SESSION['ERRMSG_ARR'])>0){
			echo'<ul class="err">';
			foreach($_SESSION['ERRMSG_ARR']as$msg){
				echo'<li>',$msg,'</li>';
				}
			echo'</ul>';
			unset($_SESSION['ERRMSG_ARR']);
			}
		?>
```

```
      </td>
</tr>
<tr>
<td width="116"><div align="right">Username</div></td>
<td width="177"><input name="username" type="text" /></td>
</tr>
<tr>
<td><div align="right">Password</div></td>
<td><input name="password" type="text" /></td>
</tr>
<tr>
<td><div align="right"></div></td>
<td><input name="" type="submit" value="login" /></td>
</tr>
</table>
</form>
```

Creating our Connection

```
<?php
$mysql_hostname="localhost";
$mysql_user="root";
$mysql_password="";
$mysql_database="simple_login";
$prefix="";
$bd=mysql_connect($mysql_hostname,$mysql_user,$mysql_password) or die("Could not connect database");
mysql_select_db($mysql_database,$bd) or die("Could not select database");
?>
```

5.1.1.2 Interface between Android studio code and Server code

5.1.1.2.1

```
ImageViewButton = (ImageView)findViewById(R.id.yourButtonsId);

Button.setOnClickListener(newOnClickListener() {
publicvoidonClick(View v) {
Intentintent = newIntent();
intent.setAction(Intent.ACTION_VIEW);
intent.addCategory(Intent.CATEGORY_BROWSABLE);
intent.setData(Uri.parse("http://www.google.com"));
startActivity(intent);
   }
});
```

Parameters: None

Return Type: None

5.1.2 Androidstudiocodeinterface

5.1.2.1

```
importandroid.content.Intent;
importandroid.os.Bundle;
import android.support.v4.app.Fragment;
importandroid.view.LayoutInflater;
importandroid.view.View;
importandroid.view.ViewGroup;
importandroid.widget.Button;
importandroid.widget.ImageButton;

public class FragmentP extends Fragment {

private static Button b,b2,b3;
private static ImageButton n;
 // private static ImageButton k;

   @Override
public View onCreateView(LayoutInflaterinflater, ViewGroup container, Bundle savedInstanceState) {
      View view = inflater.inflate(R.layout.activity_main, container, false);
      //  finalMediaPlayermediaPlayer = MediaPlayer.create(getActivity().getApplication(),
R.raw.wkola);
      //  mediaPlayer.start();

      //  k = (ImageButton) view.findViewById(R.id.music);
      b = (Button) view.findViewById(R.id.bi);
      b2 = (Button) view.findViewById(R.id.ci);
      b3 = (Button) view.findViewById(R.id.di);
      n = (ImageButton) view.findViewById(R.id.anik);

b.setOnClickListener(new View.OnClickListener() {

         @Override

public void onClick(View view) {

            Intent intent1 = new Intent(getActivity().getApplication(), Alphabet.class);
startActivity(intent1);

         }


      });
b2.setOnClickListener(new View.OnClickListener() {

         @Override
```

```
public void onClick(View view) {

            Intent intent1 = new Intent(getActivity().getApplication(), Sandar.class);
startActivity(intent1);

        }


    });
b3.setOnClickListener(new View.OnClickListener() {

        @Override

public void onClick(View view) {

            Intent intent1 = new Intent(getActivity().getApplication(), Tuster.class);
startActivity(intent1);

        }


    });
n.setOnClickListener(new View.OnClickListener() {

        @Override

public void onClick(View view) {

            Intent intent1 = new Intent(getActivity().getApplication(), Anik.class);
startActivity(intent1);

        }


    });

    //k.setOnClickListener(new View.OnClickListener() {

     //  @Override

     //  public void onClick(View view) {

        //   if (mediaPlayer.isPlaying()) {
         //     mediaPlayer.pause();
         //     k.setImageResource(R.drawable.pause);
         //  }
         //  else {mediaPlayer.start();
          // k.setImageResource(R.drawable.play1);
      //  }}

   //  });
```

```
return view;
   }

}
```

5.1.3 Database interface

5.1.3.1 See sections 5.1.1.1 and 5.1.2.2

5.2 PROCESS INTERFACE

5.2.1 Android studio Main process

Description: This process shows all graphical interface of the system

5.2.1.1 Process is created when the application started

5.2.1.2 Terminated when applications close button is pressed

5.2.1.3 All other threads will be killed if this main thread stops

5.2.2 DatabaseListenerprocess

5.2.2.1 This thread is created after Main process acquires all information from Database

5.2.2.2 Database listener process interacts with panel Server process

5.2.2.3 This process will be terminated automatically if Main thread of process is killed

5.2.3 Serverprocess

5.2.2.1 This thread is created after Database Listener process

5.2.2.2 Mainly this thread interacts between Main thread and Database listener tread

5.2.2.3 Thread is terminated when one of other two threads terminates

5.2 PROCESS INTERFACE

5.2.1 Android studio Main process

Description: This process shows all graphical interface of the system

5.2.1.4 Process is created when the application started

5.2.1.5 Terminated when applications close button is pressed

5.2.1.6 All other threads will be killed if this main thread stops

5.2.2 DatabaseListenerprocess

5.2.2.4 This thread is created after Main process acquires all information from Database

5.2.2.5 Database listener process interacts with panel Server process

5.2.2.6 This process will be terminated automatically if Main thread of process is killed

5.2.3 Serverprocess

5.2.2.4 This thread is created after Database Listener process

5.2.2.5 Mainly this thread interacts between Main thread and Database listener tread

5.2.2.6 Thread is terminated when one of other two threads terminates

6 Detailed Design

[NOT REQUIRED]

7 Design Rationale

7.1 Training

7.1.1 Description

You can learn and play in the Application

7.1.2.1 Learn alphabets,numbers, colors are interesting.
7.1.2.2if necessary request to the site.
7.1.3 Request to text, image, audio.
7.1.3.1 Check the result
7.1.3.2 See your answers.

7.1.3.3 In website also do this functions.

7.1.3.4 Only study
7.1.3.5 Interface same as mobile app.

7.1.4 Resolution of Issue

We decided to create separate database connection class in case to improve maintainability.

7.3 Internet Users (Web)

Description.

7.3.1 Description

The user can make registration and learn

7.3.2 When you open the app the user can see the main page.

7.3.2.1 The user selects which wants to study.

7.3.4 Resolution of the issue