A Mini Project Report submitted to the Faculty of Computer Science and Engineering

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

Mrs. A. Lalitha Venkatesan
Associate Professor

By

Kavya Reddy. V (12R11A05E5)
Sudeepthi. S (12R11A05F7)
Vaishnavi. V (12R11A05G7)

Department of Computer Science & Engineering
Geethanjali College of Engineering & Technology
Cheeryal(V), Keesara(M), RR Dt. , Hyderabad - 501301
(Jawaharlal Nehru Technological University Hyderabad – 500085
Telangana
2015-2016

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Geethanjali College of Engineering & Technology

(Affiliated to J.N.T.U.H, Approved by AICTE, NEWDELHI)

Certificate

This is to certify that the Mini Project report entitled "**IDIOMIZE**" is a bonafide work done by Kavya Reddy. V **(12R11A05E5),** Sudeepthi. S **(12R11A05F7),** Vaishnavi. V **(12R11A05G7)**, in partial fulfillment of the requirements of the award for the degree of Bachelor of Technology in "**Computer Science and Engineering**" from Jawaharlal Nehru Technological University Hyderabad, Hyderabad during the year 2015-2016.

Mrs. A. Lalitha Venkatesan

Associate Professor in CSE

Dept of C.S.E

Internal Guide

Dr. D. S R Murthy

Professor & Head

Dept of C. S. E

External Examiner

ACKNOWLEDGEMENTS

We are greatly indebted to the authorities of Geethanjali College of Engineering and Technology, Cheeryal, R.R Dist, for providing us the necessary facilities to successfully carry put this mini project work titled “IDIOMIZE”.

Firstly, we thank and express our solicit gratitude to Dr. D. S. R. Murthy, Professor & Head, Department of CSE, Geethanjali College of Engineering and Technology, for his invaluable help and support which helped us a lot in successfully completing our mini project.

Secondly, we express our gratitude to Mrs. A. Lalitha venkatesan, Associate Professor in CSE, Internal guide, Geethanjali College of Engineering and Technology, for her suggestions and encouragement which helped us in the successful completion of our mini project.

We would like to express our sincere gratitude to Dr. S. Udaya Kumar, Principal, for providing the necessary infrastructure to complete our project.

Finally, we would like to express our heartfelt thanks to our parents who were very supportive both financially and mentally and for their encouragement to achieve our set goals.

Kavya Reddy. V
(12R11A05E5)
Sudeepthi. S
(12R11A05F7)
Vaishnavi. V
(12R11A05G7)

ABSTRACT

We propose to develop a tutorial app titled “Idiomize” that helps users to memorize various idioms. An idiom is a phrase or a fixed expression that has figurative meaning. This figurative meaning is different from the actual meaning. Almost every English person uses several idioms in the course of the conversation. Idioms help to make English a colorful language.

Hence we are aiming at creating a fun filled learning experience that helps the user get acquainted with most of the idioms so that he can use them in his daily conversation.

The app will contain different levels under idioms where each level will have a learning session and a test session. The complexity of the idioms keeps increasing at each level. The app will even contain different games like "logo quiz", "riddles", "jumbled words" which will refresh the user and act as stress-bursting games.

Hence this application is used as a tutorial to learn idioms. This application can be used any time anywhere and at any place without depending on any other source .It does not require internet to access this application.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 About the Project

This is a tutorial application mainly intended to improve basic English skills in an easy way through an android mobile. This application enables the user to learn few idioms in a fun filled way so as to remember them and apply it when necessary. Idioms are a part of the figurative language it makes a language sound very creative and yields a better impression when used hence it has a demand to be learnt and our application provides the user to learn the idioms and make the best use of it .This application has levels of about three with increase in complexity where in you can assets yourself and stand better by learning them, Some refreshing games are also included in this application which will help the user as stress buster. Logo quiz, jumbled 4 letter words, riddles, captions are all the games that fall under the category of refreshing games.

1.2 OBJECTIVE

IDIOMS: An idiom is a special feature, special phrasing, a peculiarity or a fixed expression that has a figurative, or sometimes literal meaning, Idioms in a context makes the context seem very nice and creative. Often the good writers use idiomatic expressions, and sometimes they have a different meaning.

We have provided a medium for the users to learn and use the idioms in a simple yet effective way. The correct usage of idioms and a better way of understanding it using examples will also be provided with which the user can be benefited.

2. SYSTEM ANALYSIS

2.1 Existing System

Till now most of us are aware of applications which have tutorials on vocabulary or other grammar some of which are “idioms and phrases” which have a similar kind of objective but has a simple way of transmitting it to the users which yield ineffective results

2.1 Proposed System

Our application focuses on the most creative and impressive phrases called idioms which when used creates a good impression on the person's interactive level of standards. Our app has few exciting games that have refreshing effects to the users.

2.3 Module description

Our application mainly consists of two pages one of which is idioms page and the other page is games page, both of which have few more sub pages in it. The idioms page has levels of about three with increase in complexity where in you can assist yourself and stand better by learning them and take a test to see where you stand.

2.4 System Configuration

Technologies Used

Language	: Java, android soap
IDE	: Eclipse
Operating System	: Windows 7/8

Hardware Requirements

RAM	: 1GB Ram and above
Hard Disk	: 50GB and above
Processor	: Dual Core and above

3. Literature Overview

3.1. Java

Java is an object-oriented language and it is similar to C++. Java is simplified to eliminate language features that cause common programming errors. Java source code files are compiled into a format called Byte code, which can then be executed by a Java interpreter. Features being

Platform Independent

1. Portable
2. Simple
3. Multithreaded
4. Robust
5. Object Oriented
6. Distributed
7. Secure
8. High performance
9. Integrated

3.2. Android

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

The Android SDK includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials .Currently supported development platforms include x86-architecture computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.8 or later, Windows XP or Vista. The officially supported integrated development environment (IDE) is Eclipse (3.2 or later) using the Android Development Tools (ADT) Plug-in, to create, build and debug Android applications.

3.2.1 About Native code

Libraries written in C and other languages can be compiled to ARM native code and installed, but the Native Development Kit is not yet officially supported by Google. Native classes can be called from Java code running under the Dalvik VM using the System load Library call, which is part of the standard Android Java classes.

3.2.2 Creating an android project

The ADT plug-in provides a New Project Wizard that you can use to quickly create a new Android project (or a project from existing \code). To create a new project:

- *Select File > New > Project.*
- *Select Android > Android Project, and click Next.*
- *Select the contents for the project:*
- *Enter a Project Name. This will be the name of the folder where your project is created.*
- *Under Contents, select Create new project in workspace. Select your project workspace location.*
- *Under Target, select an Android target to be used as the project's Build Target. The Build Target specifies which Android platform you'd like your application built against.*
- *Unless you know that you'll be using new APIs introduced in the latest SDK, you should select a target with the lowest platform version possible, such as Android 1.1.*
- *Under Properties, fill in all necessary fields.*

Enter an Application name. This is the human-readable title for your application — the name that will appear on the Android device.

i. *Enter a Package name. This is the package namespace (following the same rules as for packages in the Java programming language) where all your source code will reside.*

ii. *Select Create Activity (optional, of course, but common) and enter a name for your main Activity class.*

iii. *Enter a Min SDK Version. This is an integer that indicates the minimum API Level required to properly run your application. Entering this here automatically sets the minSdkVersion attribute in the <uses-sdk> of your Android Manifest file. If you're unsure of the appropriate API Level to use, copy the API Level listed for the Build Target you selected in the Target tab.*

iv. *Click Finish.*

3.2.3 To create an AVD with the AVD manager

- *Select Window > Android SDK and AVD Manager, or click the Android SDK and AVD Manager icon (a black device) in the Eclipse toolbar.*
- *In the Virtual Devices panel, you'll see a list of existing AVDs. Click New to create a new AVD.*
- *Fill in the details for the AVD.*
- *Give it a name, a platform target, an SD card image (optional), and a skin (HVGA is default).*
- *Click Create AVD.*

When you first run a project as an Android Application, ADT will automatically create a run configuration. The default run configuration will launch the default project Activity and use automatic target mode for device selection (with no preferred AVD).

3.2.4 To Create or Modify a Launch Configuration

Follow these steps as appropriate for your Eclipse version:

- *Open the run configuration manager.*
- *In Eclipse 3.3 ,select Run > Open Run Dialog (or Open Debug Dialog)*
- *In Eclipse 3.4 (Ganymede), select Run > Run Configurations (or Debug Configurations)*
- *Expand the Android Application item and create a new configuration or open an existing one.*

4. System Design

4.1 System Architecture

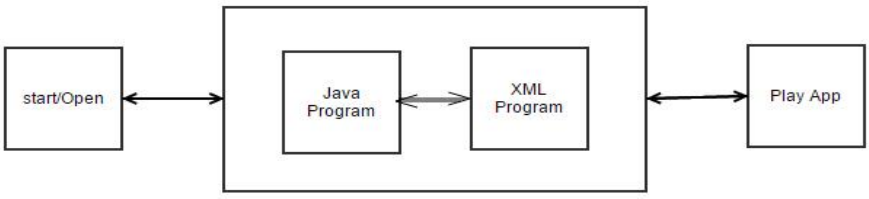


Fig: Architecture of Idiomize

Architecture diagram is a diagram of a system, in which the principle parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The block

diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.

System architecture of Idiomize has a front end displayed using xml and java is used for coding this application. The combination of both results in this application with effective designing and coding.

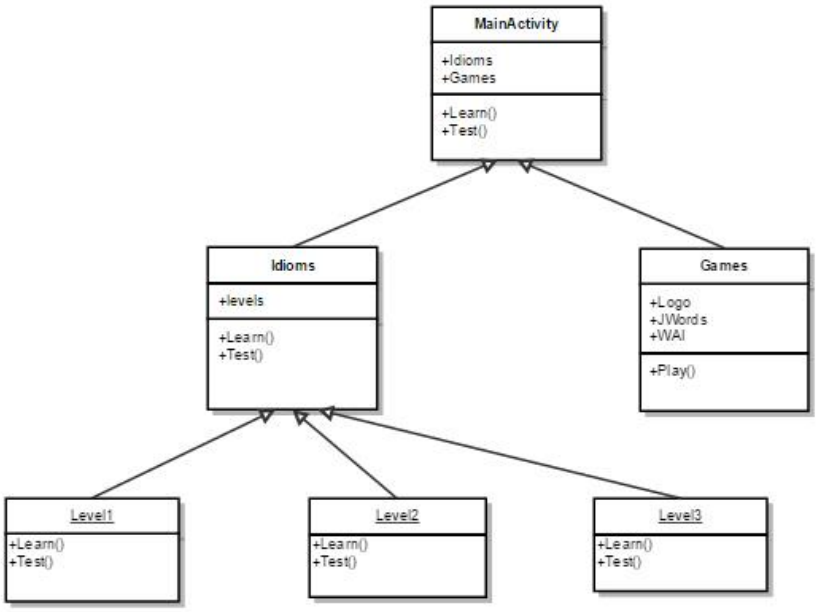
4.2 UML Diagrams

The unified modeling is a standard language for specifying, visualizing, constructing and documenting the system and its components is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure and control information about the systems.

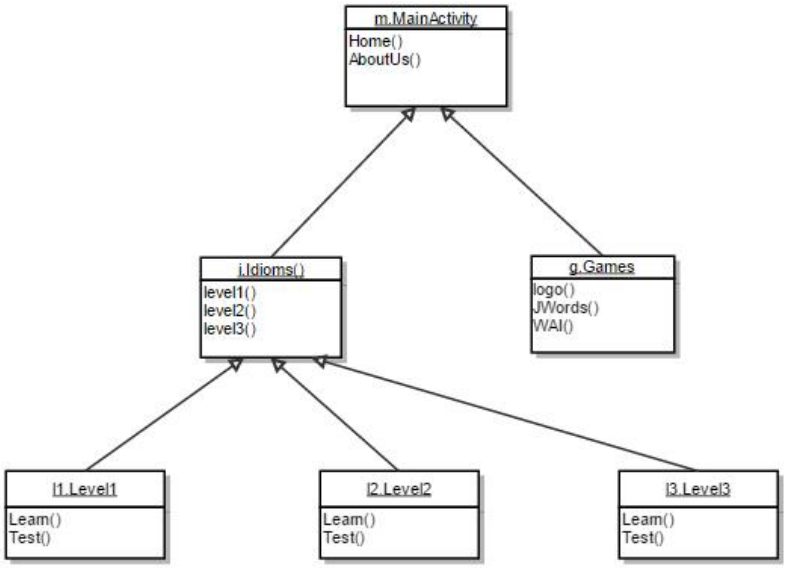
Depending on the development culture, some of these artifacts are treated more or less formally than others. Such artifacts are not only the deliverables of a project but also critical in controlling, measuring, and communicating about a system during its development and after its deployment.

The UML addresses the documentation of a system's architecture and all of its details. The UML also provides a language for expressing requirements and for tests. Finally, the UML provides a language for modeling the activities of project planning and release management.

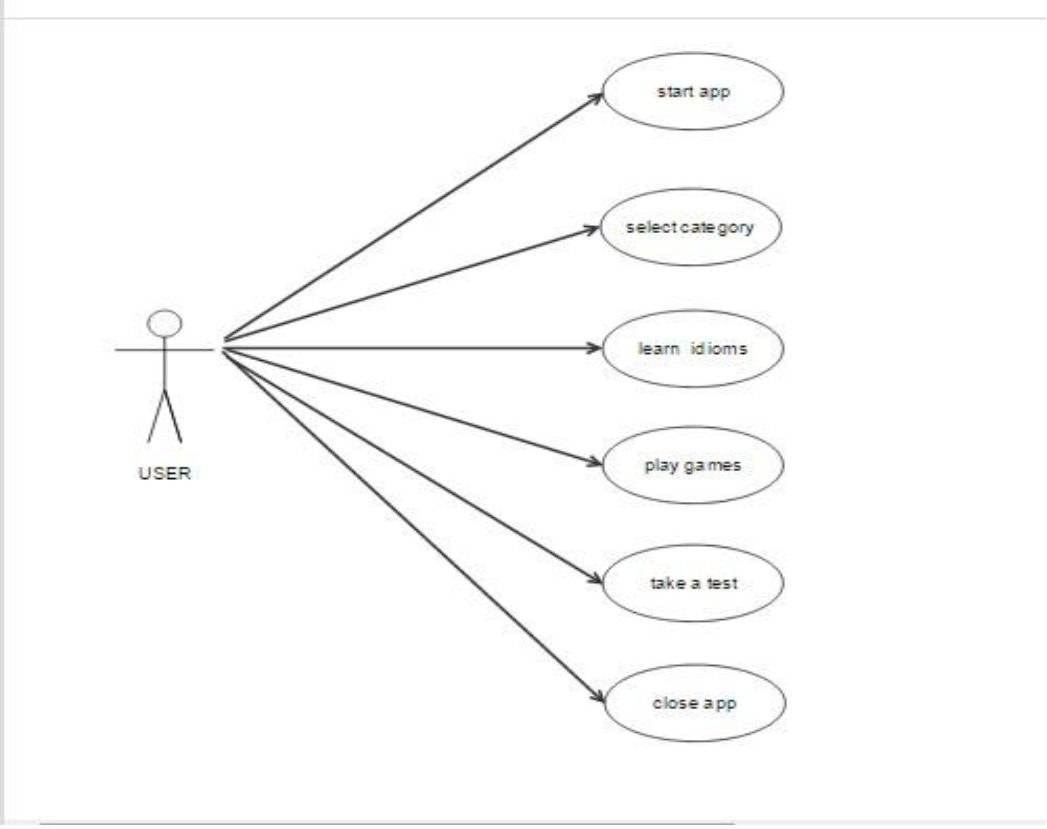
CLASS DIAGRAM



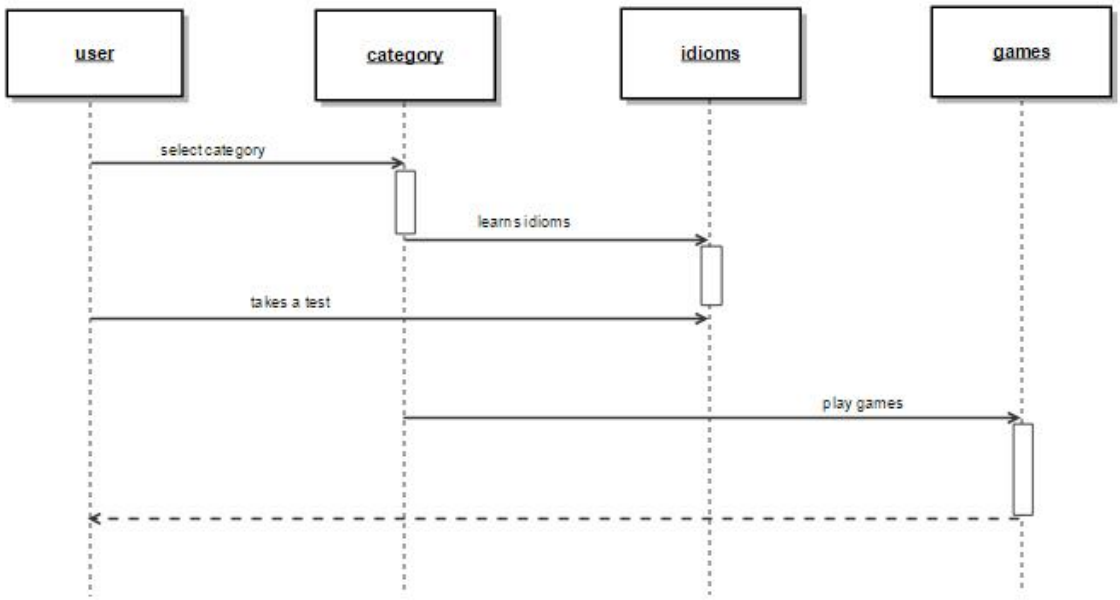
OBJECT DIAGRAM



USECASEDIAGRAM



SEQUENCE DIAGRAM



5. SAMPLE CODE

5.1 CODING

MAINACTIVITY.JAVA

```
package com.idiomize;
import android.app.Activity;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
public class MainActivity extends Activity {
MediaPlayer ourSong;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
ourSong= MediaPlayer.create(MainActivity.this, R.raw.introsong);
ourSong.start();
Thread timer=new Thread(){
public void run()
{
try
{
sleep(4000);
}
```

```
catch(InterruptedException e){
e.printStackTrace();
}
finally
{
Intent openHome= new Intent("com.idiomize.HOME");
startActivity(openHome);
}
}
};
timer.start();
}
protected void onPause()
{
super.onPause();
ourSong.release();
finish();
}
}
```

HOME.JAVA

```
package com.idiomize;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
```

```
import android.widget.Button;
public class Home extends Activity {
Button babout;
Button bstart;
@Override
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.home_layout);
babout=(Button)findViewById(R.id.baboutus);
bstart=(Button)findViewById(R.id.bgo);
babout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent openAbout= new Intent("com.idiomize.ABOUTUS");
startActivity(openAbout);
}
});
bstart.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent openGo =new Intent("com.idiomize.START");
startActivity(openGo);
```

```
}
});
}
}
```

HOME_LAYOUT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/homebg"
    android:orientation="vertical" >
     <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:text="Welcome"
        android:textSize="30dp" />
  <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
```

```
        android:layout_below="@+id/textView1"
        android:layout_marginTop="58dp"
        android:text="Discover Something New Everyday"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textSize="50dp" />
    <Button
        android:id="@+id/bgo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@+id/baboutus"
        android:layout_marginRight="51dp"
        android:text="Get Started" />
     <Button
        android:id="@+id/baboutus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="41dp"
        android:layout_marginRight="23dp"
        android:layout_toLeftOf="@+id/bgo"
        android:text="About Us" />
</RelativeLayout>
```

START.JAVA

```
package com.idiomize;
import android.animation.AnimatorInflater;
import android.animation.AnimatorSet;
import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.ImageView;
public class Start extends Activity{
Button idioms;
Button games;
Button exit;
MediaPlayer song;
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.start_layout);
ObjectAnimator cloudAnim = ObjectAnimator.ofFloat(findViewById(R.id.cloud), "x", -350);
cloudAnim.setDuration(3000);
cloudAnim.setRepeatCount(ValueAnimator.INFINITE);
cloudAnim.setRepeatMode(ValueAnimator.REVERSE);
cloudAnim.start();
idioms=(Button)findViewById(R.id.bidioms);
games=(Button)findViewById(R.id.bgames);
exit=(Button)findViewById(R.id.bexit);
song= MediaPlayer.create(Start.this, R.raw.correct);
idioms.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent openidioms= new Intent("com.idiomize.IDIOMS");
startActivity(openidioms);
}
});
games.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
```

```
Intent opengo= new Intent("com.idiomize.GO");
startActivity(opengo);
}
});
exit.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
finish();
System.exit(0);
}
});
}
}
```

IDIOMS.JAVA

```
package com.idiomize;
import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class idioms extends Activity implements OnClickListener{
```

```
Button level1,level2,level3;
Button back;

@Override
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.idioms_layout);
ObjectAnimator cloudAnim = ObjectAnimator.ofFloat(findViewById(R.id.cloud), "x", -350);
cloudAnim.setDuration(3000);
cloudAnim.setRepeatCount(ValueAnimator.INFINITE);
cloudAnim.setRepeatMode(ValueAnimator.REVERSE);
cloudAnim.start();
level1=(Button)findViewById(R.id.blevel1);
level2=(Button)findViewById(R.id.blevel2);
level3=(Button)findViewById(R.id.blevel3);
back=(Button)findViewById(R.id.bidiomsback);
level1.setOnClickListener(this);
level2.setOnClickListener(this);
level3.setOnClickListener(this);
back.setOnClickListener(this);
}
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
```

```
switch(v.getId())
{
case R.id.blevel1:
Intent open105=new Intent("com.idiomize.PAGE105");
startActivity(open105);
break;
case R.id.blevel2:
Intent openlevel2=new Intent("com.idiomize.LEVEL2");
startActivity(openlevel2);
break;
case R.id.blevel3:
Intent openlevel3=new Intent("com.idiomize.LEVEL3");
startActivity(openlevel3);
break;
case R.id.bidiomsback:
Intent opentest1=new Intent("com.idiomize.START");
startActivity(opentest1);
break;
}
}
}
```

LEVEL2.JAVA

```
package com.idiomize;
import android.animation.AnimatorInflater;
import android.animation.AnimatorSet;
```

```
import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

public class Level2 extends Activity{

Button learn,test,back;

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.level2_layout);

ImageView wheel = (ImageView)findViewById(R.id.star);

AnimatorSet wheelSet = (AnimatorSet) AnimatorInflater.loadAnimator(this,
R.animator.star_spin);

wheelSet.setTarget(wheel);

wheelSet.start();

learn=(Button)findViewById(R.id.blearn2);

test=(Button)findViewById(R.id.btest2);

back=(Button)findViewById(R.id.bl2back);

 learn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub
```

```
Intent openlearn2=new Intent("com.idiomize.LEARN2");
startActivity(openlearn2);
}
});
test.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent opentest2=new Intent("com.idiomize.L2TEST");
startActivity(opentest2);
}
});
back.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent openID=new Intent("com.idiomize.IDIOMS");
startActivity(openID);
}
});
}
}
```

LEVEL2_LAYOUT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@drawable/homebg"
  android:orientation="vertical" >
<Button
    android:id="@+id/bl2back"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="30dp"
    android:text="GO BACK TO LEVELS!!" />
<Button
    android:id="@+id/btest2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/bl2back"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="32dp"
    android:text="TAKE A TEST!!" />

  <Button
    android:id="@+id/blearn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="LEARN!!" />
<ImageView
        android:id="@+id/star"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="22dp"
        android:src="@drawable/star" />
</RelativeLayout>
```

GO.JAVA

```
package com.idiomize;
import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
```

```
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ViewFlipper;
public class Go extends Activity  {
 public int goal;
Button bjumbled;
Button bwhatami,logo;
@Override
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.go_layout);
ObjectAnimator cloudAnim = ObjectAnimator.ofFloat(findViewById(R.id.cloud), "x", -350);
cloudAnim.setDuration(3000);
cloudAnim.setRepeatCount(ValueAnimator.INFINITE);
cloudAnim.setRepeatMode(ValueAnimator.REVERSE);
cloudAnim.start();
bjumbled=(Button)findViewById(R.id.bjumbled);
bwhatami=(Button)findViewById(R.id.bwhatami);
logo=(Button)findViewById(R.id.blogo);
bjumbled.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
```

```
Intent openjumbled=new Intent("com.idiomize.JUMBLED");
startActivity(openjumbled);
}
});
    bwhatami.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent openwhatami=new Intent("com.idiomize.WHATAMI");
startActivity(openwhatami);
}
});
 logo.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Intent openlogo= new Intent("com.idiomize.LOGO");
startActivity(openlogo);
}
});
}
}
```

GO_LAYOUT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/homebg"
    android:orientation="vertical"
     >
<include layout="@layout/header" />
 <Button
        android:id="@+id/bwhatami"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/bjumbled"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="43dp"
        android:text="WHAT AM I?? :P" />
  <Button
        android:id="@+id/bjumbled"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="JUMBLED :D" />
 <Button
        android:id="@+id/blogo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_above="@+id/bjumbled"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="27dp"
        android:text="logo" />
 <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true”
  android:src="@drawable/grass" />
  <ImageView
        android:id="@+id/cloud"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:src="@drawable/cloud" />
</RelativeLayout>
```

LOGO.JAVA

```
package com.idiomize;
import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
```

```
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
public class Logo extends Activity{
TextView whatquestion;
EditText whatanswer;
Button go;
TextView whatcorrect;
Button nextquestion;
int questions[];
String answers[];
ImageView imageview;
int q;
int id;
MediaPlayer ourSong;
MediaPlayer ourSong1;
ImageView correctsm,wrongsm;
@Override
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.logo_layout);
init();
```

```
}
public void init() {
  questions=new int[]{getResources().getIdentifier("com.idiomize:drawable/dominos" , null,
null),
 getResources().getIdentifier("com.idiomize:drawable/nike" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/twitter" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/adidas" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/blackberry" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/gucci" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/ing" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/jaguar" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/lacoste" , null, null),
 getResources().getIdentifier("com.idiomize:drawable/pringles" , null, null)};
 answers=new String[]{"dominos","nike","twitter","adidas","blackberry","gucci","ing",
 "jaguar","lacoste","pringles"};
 q=-1;
  whatanswer=(EditText)findViewById(R.id.whatanswer);
 go=(Button)findViewById(R.id.whatcheck);
 whatcorrect=(TextView)findViewById(R.id.whatcorrect);
 nextquestion=(Button)findViewById(R.id.whatnext);
 imageview=(ImageView)findViewById(R.id.imagequestion);
 correctsm=(ImageView)findViewById(R.id.correctsm);
 wrongsm=(ImageView)findViewById(R.id.wrongsm);
 showQuestion();
 go.setOnClickListener(new View.OnClickListener() {
```

```
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
checkAnswer();
}
});
 nextquestion.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
showQuestion();
}
});
 }
private void showQuestion() {
// TODO Auto-generated method stub
q++;
if(q==questions.length)
{
q=0;
}
imageview.setImageResource(questions[q]);
whatanswer.setText("");
whatcorrect.setText("");
}
```

```
public boolean isCorrect(String ans)
{
return(ans.equalsIgnoreCase(answers[q]));
}
public void checkAnswer()
{
String ans=whatanswer.getText().toString();
if(isCorrect(ans))
{   correctsm.setVisibility(View.VISIBLE);
    wrongsm.setVisibility(View.GONE);
ourSong= MediaPlayer.create(Logo.this, R.raw.correct);
ourSong.start();
whatcorrect.setText("You're right!! :D");
}
if(!isCorrect(ans))
{
if(ans.equals(""))
{
Toast.makeText(getApplicationContext(),"please enter the answer",
Toast.LENGTH_SHORT).show();
}
else
{ wrongsm.setVisibility(View.VISIBLE);
 correctsm.setVisibility(View.GONE);
ourSong1= MediaPlayer.create(Logo.this, R.raw.wrong);
```

```
ourSong1.start();

whatcorrect.setText("wrong ans..correct ans is:" + answers[q]);

}}

}

}
```

LOGO_LAYOUT.XML

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/homebg"

    android:orientation="vertical" >

     <include layout="@layout/header" />

   <EditText

        android:id="@+id/whatanswer"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_centerHorizontal="true"

        android:layout_centerVertical="true"

        android:ems="10" />

   <Button

        android:id="@+id/whatcheck"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_above="@+id/whatnext"
```

```
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="22dp"
    android:text="CHECK!!" />
<Button
    android:id="@+id/whatnext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/whatcheck"
    android:layout_alignParentBottom="true"
    android:text="Next Logo!!" />
<ImageView
    android:id="@+id/imagequestion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/whatanswer"
    android:layout_marginLeft="50dp"
    android:layout_marginBottom="32dp"
    android:src="@drawable/nike" />
  <ImageView
    android:id="@+id/correctsm"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/whatanswer"
    android:layout_marginTop="14dp"
```

```
        android:src="@drawable/correctsmiley"
        android:visibility="gone"
         />
    <ImageView
        android:id="@+id/wrongsm"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignTop="@+id/correctsm"
        android:layout_toRightOf="@+id/whatnext"
        android:src="@drawable/wrongsm"
        android:visibility="gone" />

     <TextView
       android:id="@+id/whatcorrect"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/whatcheck"
        android:layout_below="@+id/whatanswer"
        android:layout_marginTop="38dp"
        android:text="crct" />
</RelativeLayout>
```

6. Testing

6.1 Testing

Software testing is a critical element of software quality and assurance which represents ultimate review of specifications, design and coding. Testing is an exposure of the system to trial input to see whether it produces correct output or not.

6.2 Types of Testing

Testing Activities

1. Inspecting Components: This finds faults in the individual component through the manual inspection of its source code.
2. Unit Testing: This find faults by isolating an individual component using test stubs and drivers and by exercising the components using a test case.
3. Integration Testing: This finds faults by integrating several components together. System testing focuses on the complete system, its functional and non-functional requirements and its target environment.

Unit Testing

Unit testing focuses on the building blocks of the software system, that is, objects and subsystems. They are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall test activities, allowing us to focus on smaller units of the system. Unit testing makes it easier to pinpoint and correct faults that are given by few computers involved in this test. Unit testing allows parallelism in the testing activities in which each component can be tested independently of one another.

The specific components for unit testing are chosen from the object model and the system decomposition of the system. In principle, all the objects developed during the development process should be tested which is often not feasible because of time and budget

Many unit testing techniques have been devised. Some of them are**:**

Test Case Design

The design of tests for software and other engineering products can be as challenging as the initial design of the product. Test case methods provide the developer with a systematic approach to testing. Moreover, these methods provide a mechanism that can help to ensure the completeness of tests and provide the highest like hood for uncovering errors in software.

Any Engineered product can be tested in either of the two ways:

1. Knowing the specified function that a product has been designed to perform, tests can be conducted. These tests demonstrate whether each function is fully operational and at the same time searches for errors in each function.
2. Knowing the internal workings of a product, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components hence been adequately exercised.

Test case design methods are divided into two types:

1. White-box testing
2. Black-box testing

White-box testing

White –box testing, sometimes called glass-box testing is a test, case designed method that uses the control structure of the procedural design to derive test cases. Using white-box testing methods, the s/w engineer can derive test cases that guarantee that all independent paths within a module have been exercised at least once. Exercise all logical decisions on their true and false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structures to ensure their validity.

Basis path testing is a white-box testing technique. The basis path method enables the test case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set are guaranteed to exercise every statement in the program at least one time during testing.

6.3.2 Black-box testing

Black-box testing ,also called behavioral testing, focuses on the functional requirements of the s/w. Black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of a program. It is a complementary approach that is likely to uncover a different class of errors that white-box methods could not.

Black-box testing attempted to find errors in the following categories.

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external data base access.
- Behavior or performance errors.
- Initialization and termination errors.

Black-box testing purposely disregards control structure; attention is focused on information domain. By applying black-box techniques, we derive a set of cases that satisfies the criteria test cases that reduce, by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing. Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specified test.

6.3 Test cases

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Larger test cases may also contain prerequisite states or steps, and descriptions. A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spreadsheet, database or other common repository.

Test Case 1

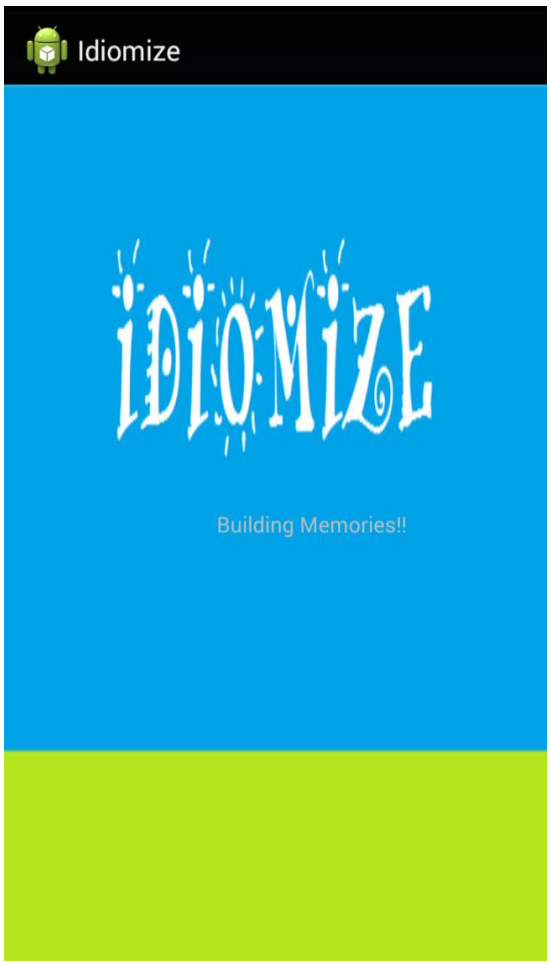
Test case 1: Test idioms	**Priority (H, L):** High
Test Objective: display correct answer	
Test Description: correct answers to the idioms will be displayed	
Requirements Verified: Yes	
Test Environment: Application must be deployed in android mobile phone or emulator.	
Test Setup/Pre-Conditions:	
Actions	**Expected Results**
The user will learn idioms.	Successfully displaying
Pass: Yes Conditions pass: Yes	Fail: No
Problems / Issues: NIL	
Notes: Successfully Executed	

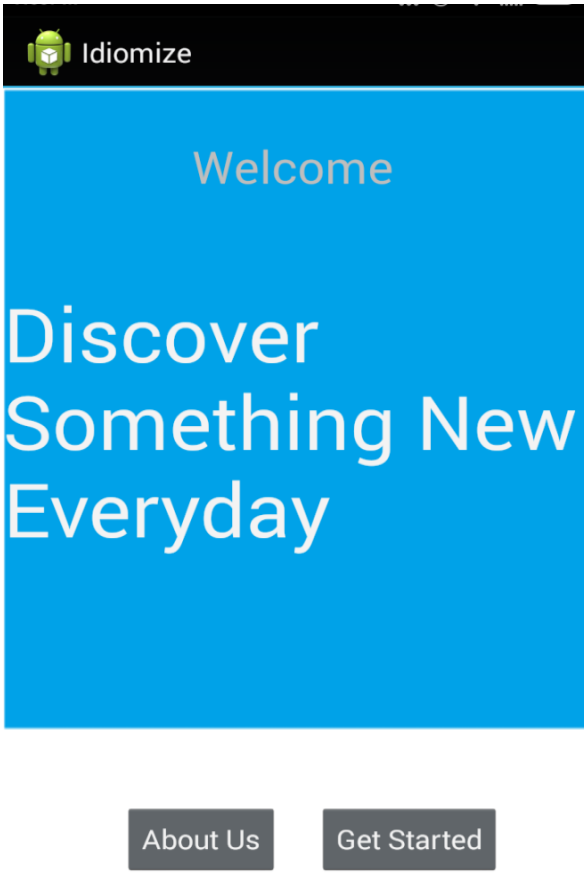
Test case 2:

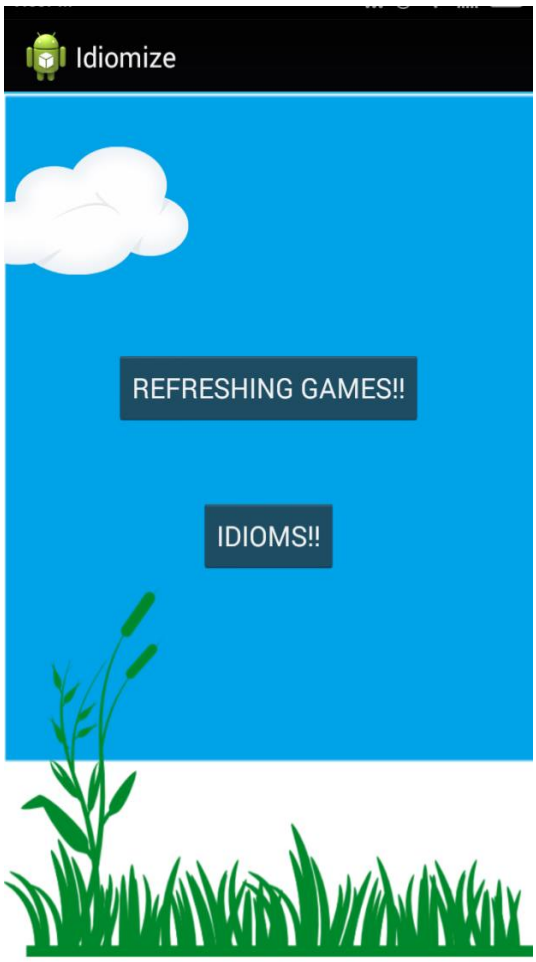
Test case 2: if an empty field is checked	**Priority (H, L):** High
Test Objective: To prompt a message saying please enter the answer	
Test Description: User enters answers in the field	
Requirements Verified: Yes	
Test Environment: Application must be deployed in android mobile phone or emulator.	
Test Setup/Pre-Conditions:	
Actions	**Expected Results**
The user plays games	Successfully playing games
Pass: Yes Conditions pass: Yes	Fail: No
Problems / Issues: NIL	
Notes: Successfully Executed	

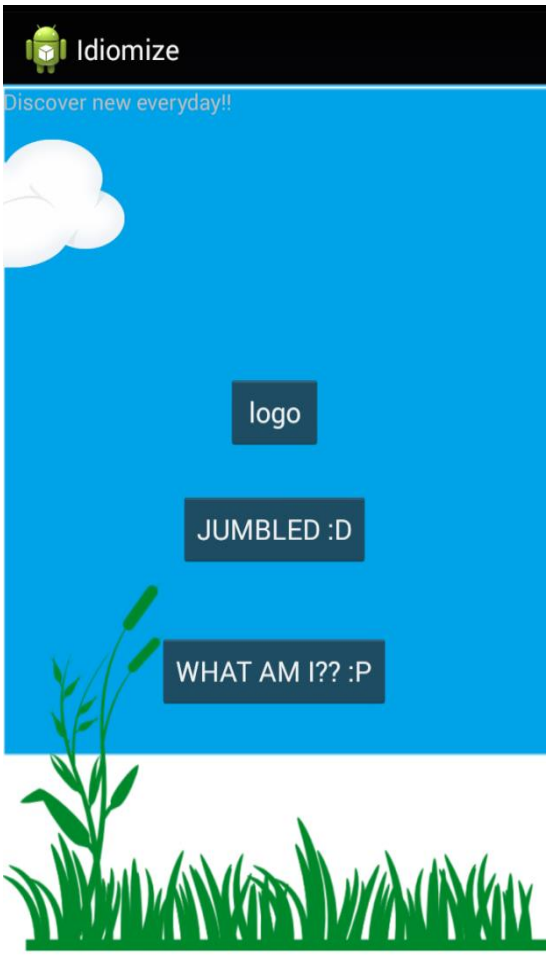
Test case 3

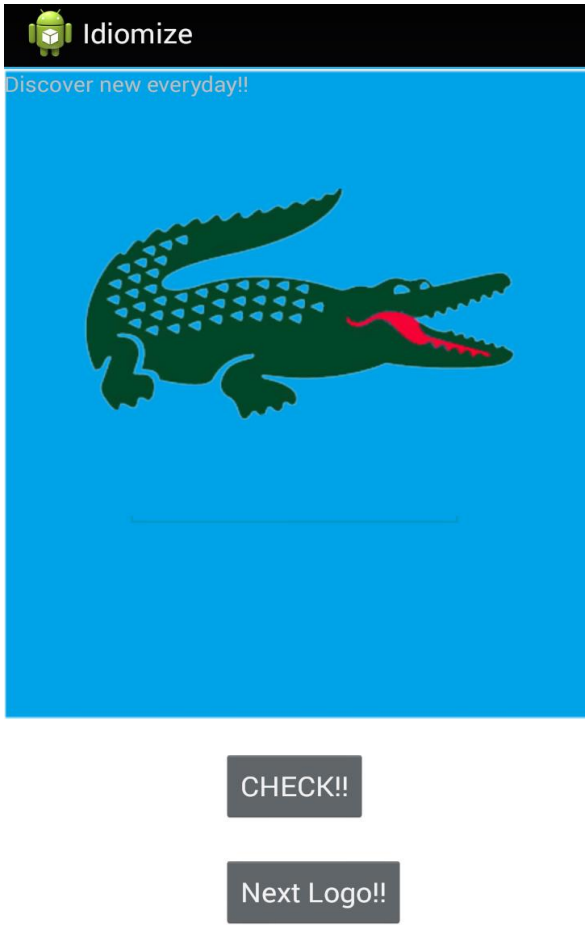
Test case 3: To see if it checks wrong answers	**Priority (H, L):** High
Test Objective: To display the correct answers if answered wrong	
Test Description: correct idioms will be displayed	
Requirements Verified: Yes	
Test Environment: Application must be deployed in android mobile phone or emulator.	
Test Setup/Pre-Conditions:	
Actions	**Expected Results**
To check the correct answers	Displaying the correct answers on test of idioms
Pass: Yes Conditions pass: Yes	Fail: No
Problems / Issues: NIL	
Notes: Successfully Executed	

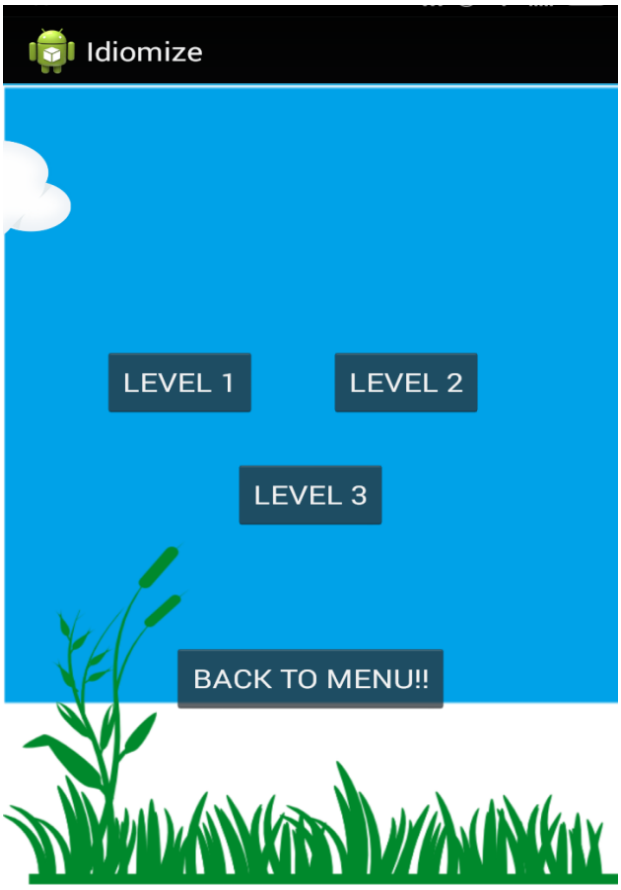

Idiomize
IDIOMIZE
Building Memories!!

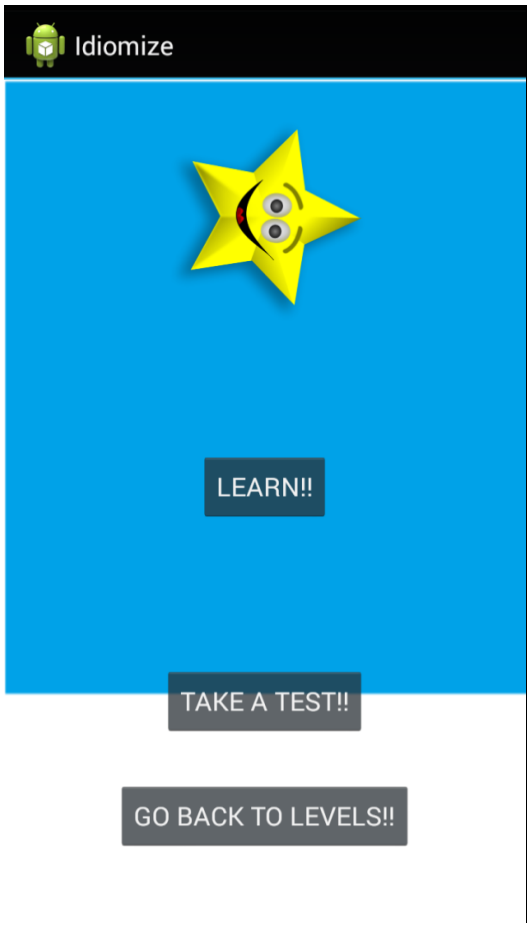

Idiomize
Welcome
Discover
Something New
Everyday
About Us
Get Started

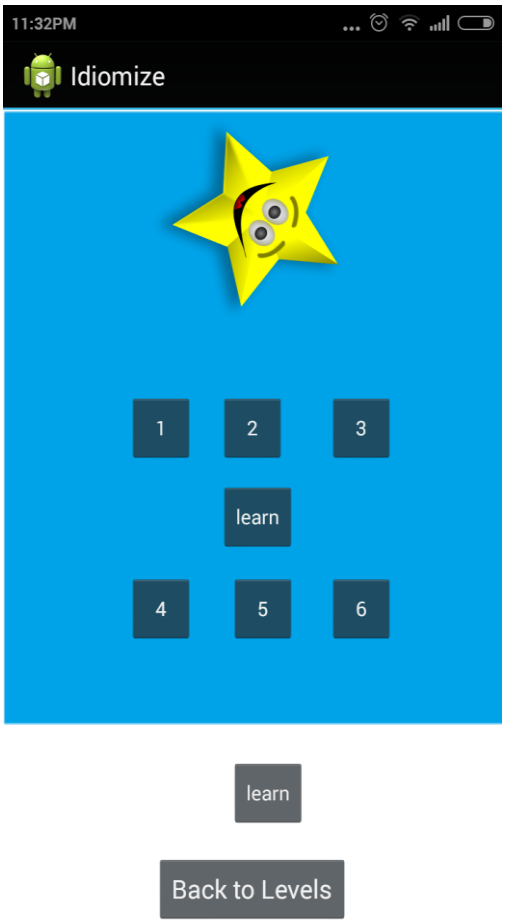

Idiomize
REFRESHING GAMES!!
IDIOMS!!


Idiomize
Discover new everyday!!
logo
JUMBLED :D
WHAT AM I?? :P


Idiomize
Discover new everyday!!
CHECK!!
Next Logo!!


Idiomize
LEVEL 1
LEVEL 2
LEVEL 3
BACK TO MENU!!


Idiomize
LEARN!!
TAKE A TEST!!
GO BACK TO LEVELS!!


11:32PM
Idiomize
1
2
3
learn
4
5
6
learn
Back to Levels

Idiomize

idiom:Cut the ground from under feet

meaning:When you cut the ground from under someone's feet, you do something which weakens their position

sentence:When team India hit more than 350 runs in the ODI, they cut the ground from under the opponent's feet

next

Idiomize

_ _a_e / y_u_ / _a_l /-Spending a lot of time and energy doing a lot of things but actually achieving too little

wrong

->

wrong ans..correct ans is:Chase your tail

Go back to Menu!!

8. Conclusion

8.1 Conclusion

This application is used as a tutorial to learn idioms. This application can be used any time anywhere and at any place without depending on any other source .It does not require internet to access this application and it is a tool developed for android platform. This makes this application efficient, convenient and easy to use along with providing maximum user satisfaction which is the key aspect for any developer.

8.2 Future Enhancements

The future enhancements that can be done to this project are updating features such as providing images to learn the idioms, adding daily goals to it, adding even more exciting games and new interesting idioms.

9. References

1. Johnathan Simons, Headfirst Android Development.

2. Wei-Meng-Le**e,** Beginning Android Application Development 4.

www.android.com http://developer.android.com/index.html

www.stackoverflow.com http://stackoverflow.com