ANDROID BALLOON GAME FOR CHILDREN

By

MOHANA SAKETH PULIPAKA

B.Tech, Jawaharlal Nehru Technological University, 2013

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Approved by:

Major Professor
Daniel Andresen

Abstract

Android balloon game application is an android application which helps in improving the children’s vocabulary power. This application provides a graphical user interface with words as questions and balloons as the options. The end user will have to select a balloon in order to select an answer. There are three modules in this application which are game mode, new game and high scores.

In the basic module questions like Alpha_bet are included. Also as an enhancement, the game is made interesting by giving the user three lives to clear the levels of the game. Three small balloons are included in the application which resemble the three lives and once an incorrect answer is selected, the green balloon will be replaced by a red balloon depicting an incorrect answer. Also the scores will be computed for every correct option that is selected by the user. The score is incremented by ten points for every correct answer.

And for the high score section, the game would maintain the list of top players. High score section will consist of the user’s name, score, time and date for the gameplay. Apart from that a next button is included which helps the user to skip a word which the user finds it difficult to answer. A hint option is also provided in the game which pronounces the word helping the user to guess the word correctly.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The implementation of this application would have not been possible without the guidance of my major professor, **Dr. Daniel Andresen**, as he has always supervised me and has provided valuable suggestions for improving this application in terms of user experience and functionality for which I am very grateful to him. I would also like to thank **Dr. Torben Amtoft** and **Dr. Mitchell Nielsen** for taking their valuable time off to be my committee members and providing me an opportunity to take courses under them.

I am grateful to **Dr. Gurdip Singh, Dr. Scott DeLoach** and the CIS department for providing me an opportunity to be a graduate teaching assistant during the course of my study. The experience that I have gained during this period was invaluable.

Finally, I would to thank my parents **Gopi Krishna Pulipaka** and **Padmaja Pulipaka** for making me what I am today. Special thanks to all of my friends for providing support and encouragement during the course of my study.

Chapter 1 - Introduction

A game application which targets the children providing them entertainment and practice with vocabulary is very popular trend and Android Balloon game is one of them. This application aims at providing a graphical user interface with questions and balloons as the options. The end user will have to select a balloon in order to select an answer.

In the main module questions like Alpha_bet are included. Simultaneously, the questions will become much complex depending on the difficulty of the game. Also as an enhancement, the game is made interesting by giving the user three lives to clear the levels of the game. Also the scores will be computed for every correct option that is selected by the user and for the high score section, the game would maintain the list of top players.

Android is one of the technologies that I have used for developing some of the projects while I was working with other interns during my internship. Since it was a collaborative environment, I was able to know only the basics of android development. Since then, I always had a penchant for learning android development. All these reasons inspired me to create an android application and learn more about the android development.

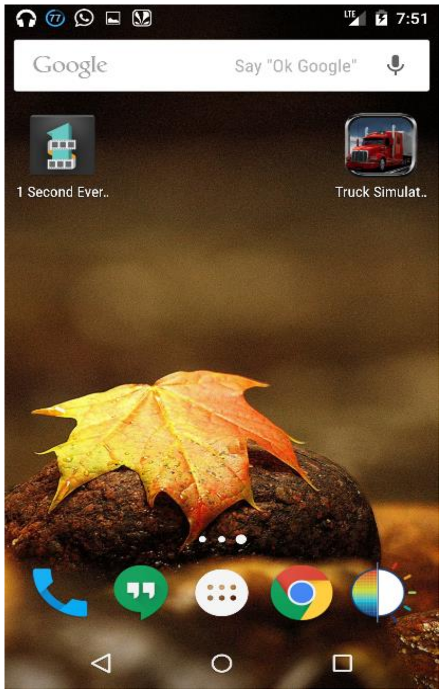
Some of the related work description of android framework its related technologies and requirements for the developing this android application have been included. Also, implementation of the project is discussed, followed by testing, conclusion and future work.

Chapter 2 - Android OS and related work

2.1 Android OS

Android is an operating system, based on Linux kernel which can be used on various types of touchscreen devices. Different types of android based devices are present in the market such as smartphones, tablets, televisions, wrist watches and automobiles. One of the main reason for the android popularity is because of its open source nature. The applications can be made available to other users by publishing the apps on google play store (Android Market). A large number of tutorials over the web makes it easy for anyone who wants to start android development. According to the statistics, Android has the largest installed base of all general-purpose operating systems. Android Inc. was founded in California, United States in October, 2003 by Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc.). It was then acquired by Google in 2005. [1]

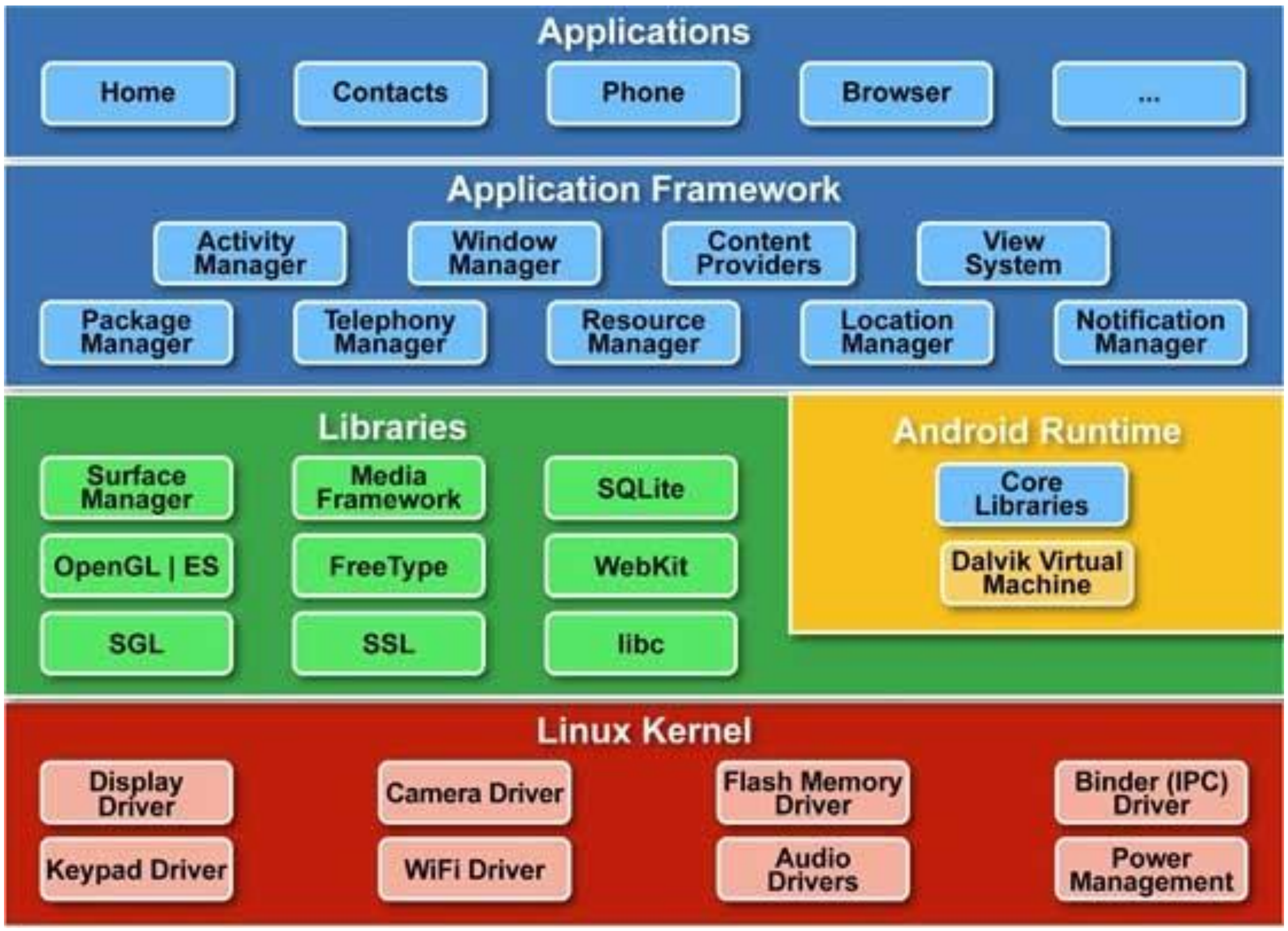
Figure 1 Android OS



2.2 Android Architecture

Android is a software stack of applications, an operating system, run-time environment, middleware, services and libraries.

Figure 2 Android Application Architecture [3]



2.2.1 Linux Kernel

From the figure 2.2.1, it is evident that Linux kernel is basically the bottom most layer of the Android architecture. The layer also has the drivers which are responsible for communicating with equivalent hardware. For example, the Linux kernel would consist of drivers to communicate with the hardware devices such as phone-screen, headphone, Bluetooth devices, etc. To be brief, Linux Kernel acts as a medium between the hardware and the other layers of the Android architecture.

2.2.2 Android Runtime and Libraries

Android runtime also consists of core java libraries which basically provide most of the functionalities available in java standard edition libraries. ART has replaced DVM from the android Lollipop version as ART has an edge over DVM in terms of performance.

Some of the important native libraries include the following:

Libraries consist of Android libraries which are needed for android development. These android libraries present various functionalities such as usage of database, widgets, networks and many more. Apart from the android libraries there are also native libraries which are used for graphics and for utilizing multimedia playback. Java libraries are also included so that developers may utilize the concepts of string manipulations and other tasks. Apart from these libraries, there are different frameworks available which can be used in order to work with media codecs and media formats. Android libraries also have the feature of displaying HTML content using a browser engine. Also, there are respective libraries which support database storage and manipulation. [3]

2.2.3 Application Framework

The applications that are developed by a developer basically utilize the components of application framework. It can be stated that, a developer generally uses the services offered by the application framework in order to create android applications. The various services offered by this framework content providers are the transfer of data between different applications, telephony manager, activity manager for controlling the application's lifecycle, resource manager, notification manager for managing the notifications, location manager for utilizing the location based services and many more.

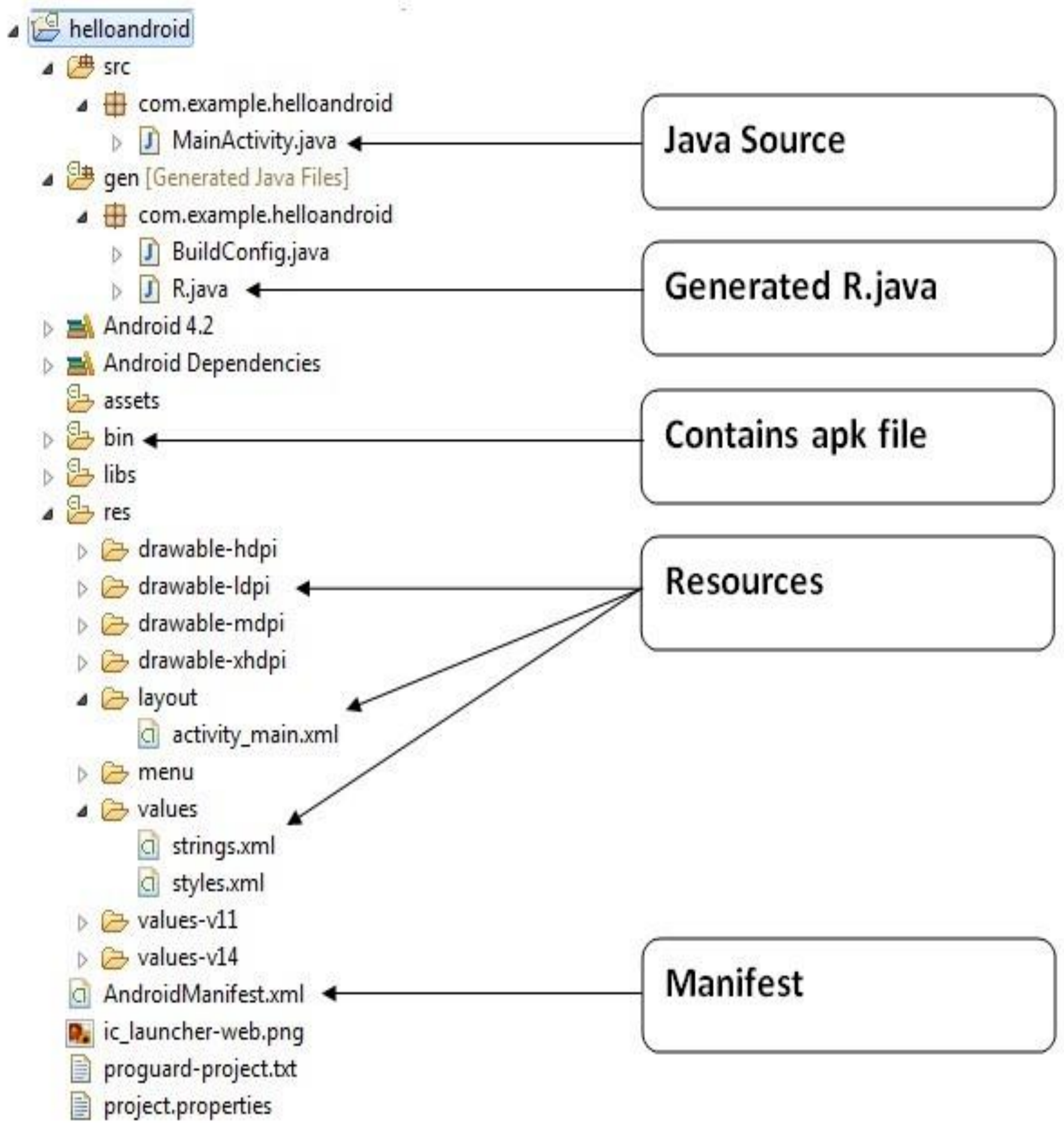
2.2.4 Applications

Applications are generally the default applications that are available with a new phone. These are also the applications which the user can install from play store. Calculator, Calendar, Camera, Contacts, SMS, Email, Music player are basically some of the applications that are pre-installed. [3]

2.3 Android Directory Structure

Default files and folders are generated when an android application is created in the eclipse IDE. The whole project represents the android directory and the files in this structure have to be created and manipulated based on the requirements. The following figure shows a screenshot of the android directory structure.

Figure 3 Android Directory Structure [4]



2.3.1 Components of the directory structure

The folder labelled as “src” contains java source files which are required to be created and modified depending on the functionality that is required. These java classes are generally called as activities which basically are nothing but the screen which the user sees when the application starts. The “gen” folder consists of auto generated java files which are not advisable to be modified.

The folder bin contains the “apk” file which can be used in order to install the application on the phone. These apk files are application packages that are downloaded from the play store.

The res folder consists of resources that are required to be used mainly for the UI manipulation of the application. The folder consists of draw able folders where images required for the application are placed. It also consists of layout folders where xml files are placed. These xml files basically represent the layout of a particular screen in the application. These xml files can also be auto generated and modified since there is an option of graphical layout from where the user can drag and drop specific components for the screen layout.

The AndroidManifest.xml file contains the package name, details about android version and name, different kinds of permissions that the application would require and also the activities contained in the application. It is the most important component of the entire directory since it contains the whole information about the application.

Chapter 3 - Requirement Analysis

The target audience or the target users for this application were children. Hence, initially I have contacted around 3-4 children in the age group of 10-15 to provide some requirements. These requirements were analyzed by me in order to derive the main functionalities of this application. One of the requirements was a rich user interface with some good animation effect. The application should cover a lot of screen space was another requirement. One of my nephew suggested to include a scoring section too, so that they can track their improvement. Another input that I have obtained was to include lives section.

The above were the requirements from the user's perspective. Apart from the inputs that I received from the children, I also had some ideas to include, such as a high score section which would track the top scores in the game and would display the scores in a list view with their name and their score. And then again my nephew suggested that it would be even more appropriate if date and time were also included. Also, I had asked all the children about the kind of phones they use and from that I had decided that the minimum version of the application should be android 3.2.

Some children advised that the game should include a difficulty section such as easy, medium and hard. Then these requirements were taken into consideration and the first version of the Balloon game was developed. The application was tested by the children in order to get some feedback about the game. My nephew suggested to include a button which can be used to skip a word which they feel they cannot answer. Also, Dr. Daniel Andresen suggested that the options should cover as much screen space as possible.

Dr. Andresen also suggested that a tip icon could be included in the application which would help the player to guess the missing letter accurately.

Hence, the following features were included in the final version of the application. A next button was included in the application that would skip the given word in the game. Then three small balloons as image views were included which represent the three lives that the user can have. On losing the three lives, the game is over. Also, in order to provide a competitive environment, I had included a high score section which would maintain a list of top players. It consists of score, name of the player and the timestamp on which the game was played.

A bulb icon is also included which would pronounce the word to help the user in guessing the missing letter accurately. A score section is provided which would increment the score by ten points for every correct answer. In order to provide a rich user interface, animations were provided on almost all the widgets used in the application. In this way, the final version of the application has been implemented to meet the requirements provided by the users or the target audience. The following are the hardware and the software requirements for the developer.

3.1 Requirement Specification

3.1.1 Software Requirements

Requirements for the development Environment:

Operating System: Windows XP or higher, Mac OS X

Development IDE: Eclipse (ADT plugin + Android SDK tools)

Database: SQLite

Running Application: Android Emulator or Android Device

3.1.2 Hardware Requirements

Requirements for the development Environment:

Processor: Intel Pentium IV or higher

RAM: 256 MB

Space on disk: 1GB or higher

Requirements for running the application:

Device: Android device with version 3.2 or higher

Space on disk: 5.0 MB

Chapter 4 - System Architecture and Design

4.1 System Architecture

The system architecture of the entire application is represented in the following diagram.

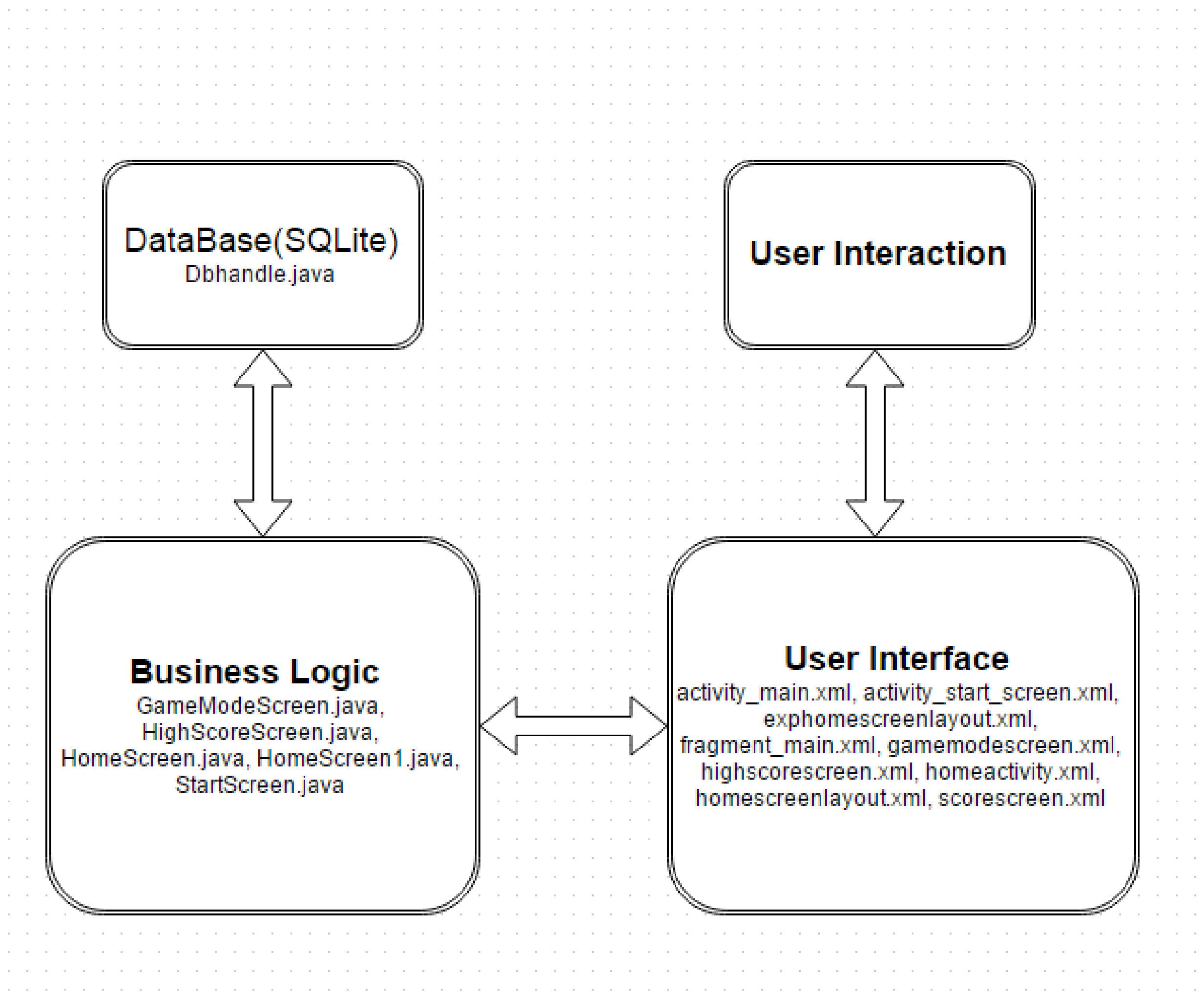

DataBase(SQLite)
Dbhandle.java
User Interaction
Business Logic
GameModeScreen.java, HighScoreScreen.java, HomeScreen.java, HomeScreen1.java, StartScreen.java
User Interface
activity_main.xml, activity_start_screen.xml, exphomescreenlayout.xml, fragment_main.xml, gamemodescreen.xml, highscorescreen.xml, homeactivity.xml, homescreenlayout.xml, scorescreen.xml

Figure 4 System Architecture of the application

The above diagram depicts the entire architecture of the application. The application can be broken down into three components. The Business logic, User Interface and the Database component. User interaction is nothing but the user interacting the application by giving inputs via touch and observing the outputs which are the operations that are performed by the application.

As soon as the user starts the application, the game loads up and the user is provided with two options. The buttons are high scores and new game. When the high scores is selected, the user will be transferred to the high scores page, where the user will be able view the scores in the descending order. For each entry, the user will be able to view the name of the player, the score and the date and time on which the game is played. Also for the top scorer a medal will be included.

When the player selects new game, the player can select the difficulty level for the game. And like all the games the user has an option to choose the difficulty level of the game from the three options “easy”, “medium” and “difficult”. And on selecting the difficulty the game begins. The player will be displayed with the word with missing letter and he/she will be able to pick the correct letter from the options. Three lives are provided and also a hint option is provided which pronounces the word.

Generally, User interface contains the xml files which are presented to the user in the form of graphical user interface. These xml files uses resources such as menu, string values and images to act as a presentation layer. These xml files are placed under the layouts folder of res folder in the android project directory. These xml files consists of various attributes for the different widgets and text fields such color, font size, alignment, etc.

The Business interface consists of java files which are nothing but activities in the application. These activities provide the backend logic for the presentation layers. Each activity will contain xml file which are to be inflated on the application device. These java files are basically placed under the “src” folder of the android directory structure. The database component consists of a java file which is responsible for interacting with the SQLite database. The developer will have the liberty of writing SQL statement for the creation, manipulation and deletion of data.

4.2 System Design

System design of this application could be explained in a better way using the UML diagrams which act as a blueprint for any application. Basically, using the UML diagrams one can visualize the entire system design of the application. The following are UML diagrams which depict the system design of the application

4.2.1 Use case Diagram

A use case diagram represents the dynamic behavior of a system. A use case diagram generally consists of actors and use cases. The diagram depicts the interaction between the actors and their functionalities. The actions of the user are nothing but the features that are supported by this application. Below is a clear depiction of a use case diagram. [5]

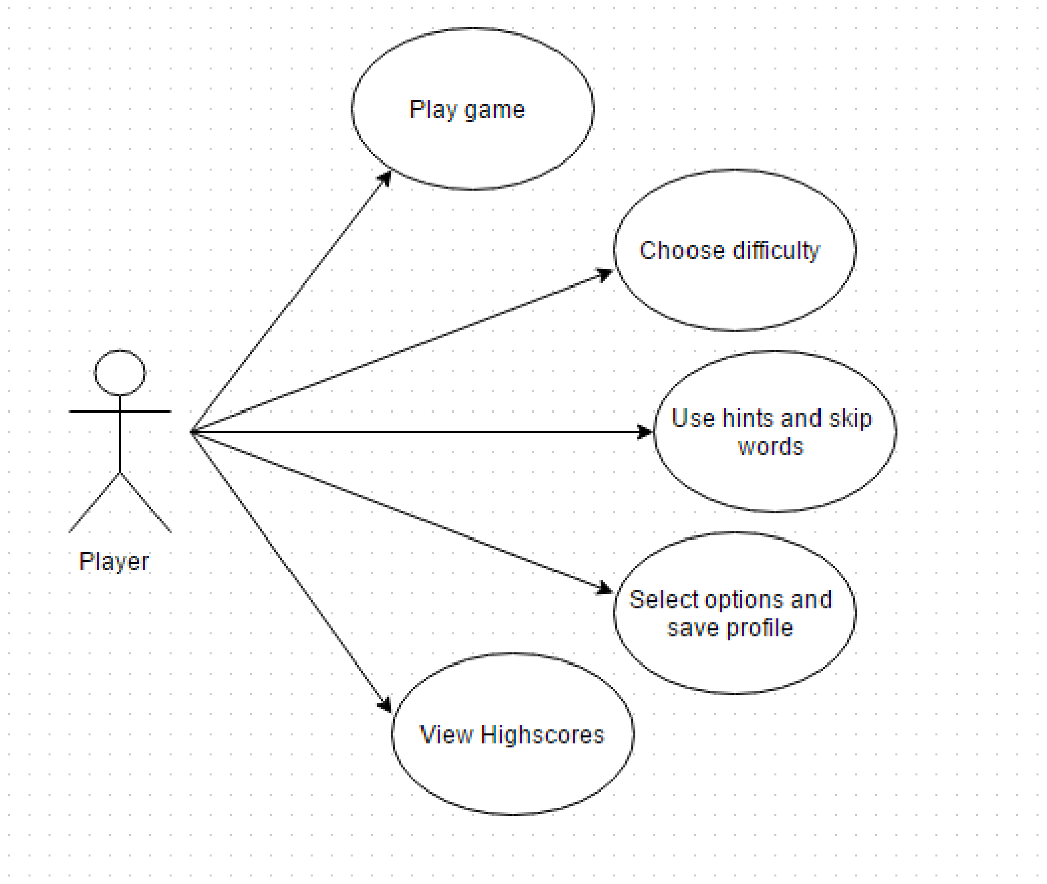


Figure 5 Use Case Diagram

In the above use case diagram, the actor is generally the player who interacts with the application and the use cases are nothing but the features or the functions that are supported by this application. So, the actor can basically utilize the functions such as playing the game, choosing the difficulty of the game, using hints whenever the word seems difficult, skipping the word by using the “next” button, selecting options, saving profile when all the three lives are lost and viewing the high scores.

4.2.2 Activity Diagram

Activity diagrams depicts the flow of an application. They are basically the diagrams which present the control flow of the whole application. The activity diagrams represent the transition from one operation to another operation. The below activity diagram depicts the dynamic behavior of the application. [5]

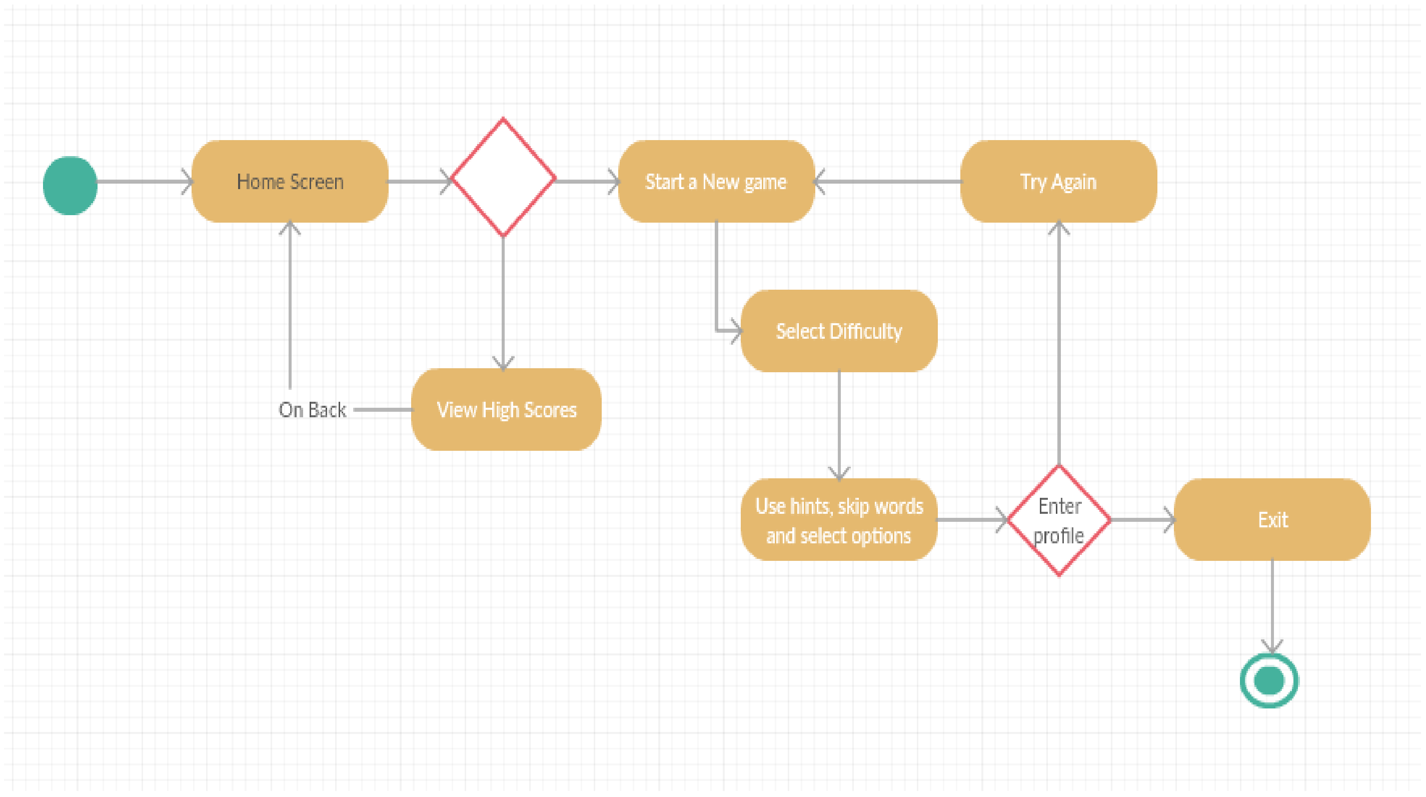


Figure 6 Activity Diagram

The above activity diagram illustrates the control of the application. The activity starts when the player starts the application. Then he/she has two choices of viewing the high scores and starting a new game. The player then, can select the difficulty and start the game. At the end the player will have an option of saving his/her profile, exiting the game or even trying the game again. At the exit the activity ends.

4.2.3 Class Diagram

The class diagram illustrates the static view of the application unlike the use-case and activity diagrams. It can be said that the class diagram would basically depict the entire structure of the system, like a blueprint. It gives information regarding the system's classes, attributes, methods and the relationships between these.

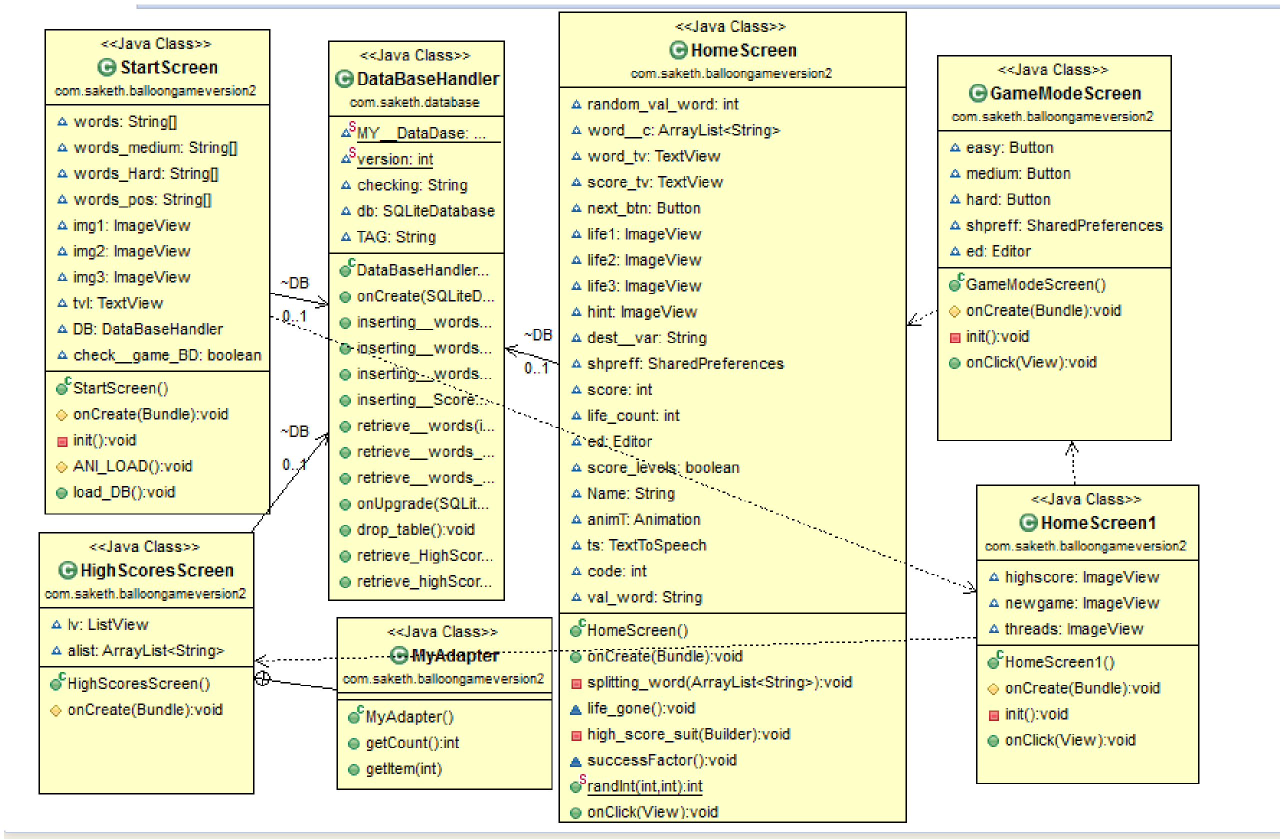


Figure 7 Class Diagram

The above class diagram represents the entire class structure of the android application. It is clear that there are seven classes on the whole. As soon as the application is started, the user generally experiences the SplashScreen.java class. The class inflates an xml file for the presentation layer, but the main logic is instilled in these java files.

StartScreen.java consists of three image views, one text view and one progress bar. This is basically the screen where the game loads the words into the database. The three image views are used for displaying the balloons. The text view for displaying the name of the game and a progress bar is also included. After the game loads up, the control of the application finally shifts to the HomeScreen1.java.

In the HomeScreen1.java, the three image views are used for displaying the “New Game” and “High Score” options. The third image view is used for displaying the threads. Different animations are loaded for displaying both the options. For “High Score”, TranslateAnimation is used and methods like setDuration(), setFillAfter() were applied. On selecting the “New Game” the android display is redirected to the activity GameModeScreen.java. Also the Home Screen will also be directed to HighScoresScreen.java when the user selects High Score.

In the HighScoresScreen.java, the class consists of an array list and a list view. The entire page itself is a list view with all the scores in it where the xml file inflated for the graphical user interface is highscoresscreen.xml. The array list retrieves the scores from the database. And for each row the file scorescreen.xml is inflated. Now for a row three text views are used which are nothing but the date and time when the game was played, the name of the player and the score of the player. When the high scores are displayed, a medal is always displayed for the high scorer.

On selecting a new game, the activity GameModeScreen.java is started and the activity consists of three buttons which represent the level difficulties. Also, the concept of shared preferences is used for carrying different kinds of information to the next activity. Now the layout inflated for this activity is gamemodescreen.xml. And on selecting an option the game now shifts to HomeScreen.java.

Now the activity HomeScreen.java is started. This activity consists of text views, buttons, image views, shared preferences, text to speech and array lists. Exphomescreenlayout.xml is inflated for this activity. A text view is used at the bottom of the screen for displaying the score of the game. A linear layout consisting of three image views are used for representing the lives of the player. A "next" button for skipping the current word, a text view for displaying the word and finally an image view for "hint" which basically spells out the word. Since the lives are image views, for every wrong answer on of the green life will be replaced by an image view which is a red life. Using the shared preferences the value of the difficulty of the game are passed on to this activity, where depending upon the difficulty the words are retrieved from the database and are stored in an array list. Each word is then randomly split for the missing letter and then a "?" is placed on the missing letter. Then random letters are generated. If the correct letter is selected then the score is incremented by calling the method successFactor() and displaying a toast message. And if incorrect letter is selected then the method life_gone() is called and a toast message with the correct letter is displayed.

When the life_gone() method is called the life count value is decremented and the shared preference is updated, also if the life count reaches zero then all the images views of the lives are set to "red" and soon after this an alert is created by using AlertDialog.Builder(), where a message of losing the game is displayed. Also an option of trying again is provided within the alert by using DialogInerface.OnClickListener(). If the user chooses to try again then he will be redirected to HomeScreen.java. Also the user has to option of exiting the game. The functionalities of "try" and "exit" are set basically by using setPositiveButton() and setNegativeButton(), where try is the positive and exit is the negative button.

Now, when a player loses with the high score, the game asks the player using an alert dialogue to enter their name for the purpose of displaying their name in the high sores. Here also an AlertDialog.Builder() is used and a message displaying "You Lost with High Score" is displayed. Then again, a DialogInterface.OnClickListener() is used for storing the name of the player, where the name of the player, high score and the date the game was played are inserted into the database. Also, a successFactor() method is used to incrementing the current score every time the players selects the correct letter, a TextToSpeech functionality is also used for pronouncing the word.

Chapter 5 - Android Application Components

An android application is basically built by utilizing the application components of the android framework. These components act as the Android API for developing mobile applications. The following are the application components in Android.

5.1 Activities

Activities are nothing but the screens in the Android which provide user interface. A single screen can be described as a single activity. There can be multiple activities in an android application. An activity can be designated as main activity which will be launched when the application is started. Each activity is inflated with an XML view which contains user interface components like text boxes, buttons, labels, images etc.

And android activity can be managed by using different call back methods such as Create, Start, Resume, Pause, Stop and Destroy. The activity will be in differently displayed depending upon the method it is in. Example of an activity is provided below. [6] [7]

```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be
"paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

5.2 Services

Services are generally the components that run in the background of an application. The services can best explained with an example of a music player. For example, when the user selects the music track from the player, the user can minimize the music player and the music will be still playing in the background while the user interacts with the other application. Services basically doesn't contain a user interface, they act as the background processing components for the frontend activities. They also find their usage in the data processing. The services can actually be started either by an activity or by another service in the application. [6]

5.3 Content Providers

Content providers allow the concept of data sharing among vivid applications. A data in one application can be used and modified by another application is one of the important uses of content providers. A client server architecture can also be simulated by the usage of content providers, all the centralized can be placed in a server and the client can then interact with the server in order to add, modify, delete and also to query the data by utilizing the concept of content providers. The concept of content providers is also used in many contact based applications where an application might be required to access the contacts in mobile phone. Such applications can be made possible by content providers. There are many built in content providers that can be used in Android. [6]

5.4 Broadcast Receivers

Broadcast Receivers can be used in order to instill the concept of notifications. A broadcast message acts as a notification to other activities or screens in the application. A broadcast message is illustrated with the help of intents. Broadcast receivers are nothing but the subclasses of the BroadCastReceiver.java class. The broadcast receivers will generally act as the event listeners for the broadcast messages that are transferred from the other activities. Broadcast receivers are also generated by the android system if the android system generates any broadcast messages. Some of the examples include indication if the battery is low or battery is full. Broadcast receivers also finds their use in case of Play store. Whenever an app is downloaded from play store, a broadcast message is broadcasted and the broadcast receiver will install the application in the mobile phone. [6]

5.5 Intents

In order to go from one activity to another, Android uses the concept of Intents. Intents are used for the dynamic behavior of the activities. An application can also be started from an intent. The intents can be utilized in any of these concepts such as services and broadcast receivers. The Intents are generally passed as an object to the method startActivity() in order to launch a new activity. Similarly, the intents are passed to startService() in order to launch a service. For broadcasting the message to the broadcast receivers the intents are passed into sendBroadcast() method. Also, there are two types of Intents that are supported by android they are explicit and implicit intents. [6]

5.6 Android Manifest File

AndroidManifest.xml is an important component in the entire directory structure. It also contains information about the entire package including various activities and services. In brief, the manifest file generally gives about the android minimum version. It gives an insight about the android icon and its label.

Apart from that the manifest file could also be used for specifying the activities, services and broadcast receivers in the application. An activity could also be initialized as a main activity which would load on startup of the application from manifest file by using the concept of intent filter.

Also, the manifest file is responsible for granting different kinds of permissions for the installation of application. Some of the permissions include accessing the internet, accessing the contact list, etc. The manifest file for this application is provided below.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.saketh.balloongameversion2"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="10"
        android:targetSdkVersion="19" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>

    <application
        android:allowBackup="true"
        android:largeHeap="true"
        android:icon="@drawable/ballon"
        android:label="@string/app_name"
        >
        <activity
            android:name="com.saketh.balloongameversion2.StartScreen"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="HomeScreen" >
        </activity>
        <activity android:name="HomeScreen1" >
        </activity>
        <activity android:name="GameModeScreen" >
        </activity>
        <activity android:name="HighScoresScreen" >
        </activity>
    </application>

      </manifest>
```

Chapter 6 - Implementation of the Application

The implementation of this project can be divided into three different components. The front end where most of the layouts are nothing but the xml files. Each screen on the android application is nothing but an xml file which are located in the layout folder under the res folder. These xml files consist of different number of widgets such as Progress bars, Text views, Image views, etc. For each of these elements different properties are available that can be modified for fulfilling the requirements of the project. Some the properties consist of height, width, alignment, source for image views, default text, color, etc. Different layouts are used such as linear layouts, relative layouts, etc. Also, all the image resources are placed inside the drawable folder.

As soon as the application starts activity_splash_screen.xml is displayed. This file consists of three balloons which are nothing but three image views with some properties regarding alignment, id, height and width. The method AnimationUtils.loadAnimation() is used to apply animations to the three image views. A text view is used to display the title of the game and the properties applied to it consists of text, width, height, text style, text color and text appearance. For displaying the loading icon, a progress bar widget is used where it holds some layout properties such as width, height, below and center horizontal. Also, as an addition a text can also be displayed for the progress bar. And all these widgets are placed in a relative layout.

Now, for displaying the home page of the application, the xml file homeactivity.xml is inflated. Three image views are used in this application, where two are used for the “new game” and “high scores” buttons. The third image view is the thread which comes along with the new game button. The property used for setting the new game button below the thread button is “layout_below”. The method TranslateAnimation() is used for providing animations to the high score button and for the new game button, the animation is provided using the method AnimationUtils.loadAnimation().

On selecting a new game, the game inflates gamemodescreen.xml. Here, three buttons are used for easy, medium and hard. For the appearance, some of the properties that are applied to these buttons are marginLeft, marginRight, marginTop, text and text color. Since these button are supposed to be aligned in a vertical way, a linear layout is used for holding these buttons.

If the selected option was high scores, then HighScoresScreen.xml is inflated where a list view is included which would display three text views in each row and for the row the file scorescreen.xml is inflated which would consist of three text views and one image view. The text views have the properties of marginLeft, marginTop, text color, text and text appearance. Another text view is also used for displaying the date and time which has its layout below the text view which displays the score. Visibility property for the image view is set as “gone” by default in the xml file. The medal is made visible only for the top scorer by using the method .setVisibility().

After selecting the game mode, the main part of the game starts up where the user has to start selecting the correct answers. In this part, the file exphomescreenlayout.xml is inflated. Many widgets are utilized in this application. Altogether, two linear layouts are used in this application, one linear layout consists of a text view, a button and an image view. The image view represents the hint icon and it uses the layout properties of alignParentLeft and centerVertical. When the hint icon is pressed, the image is replaced by another image when pressed and then voice_HintOperation() is called which utilizes the text to speech engine concept in android making it possible to pronounce the word for the user. Another component in this linear layout is a button, which is used for skipping the word. This button “Next” has a text color, text style and layout alignment. The last component in this layout is the text view in which the word with the missing letter is displayed, it also consists of different layout properties such as centerHorizontal, centerVertical, text, text appearance and text color. Apart from this layout, another linear layout is also used where three image views are used for depicting the lives of a player. These image views also include animations. Whenever a wrong answer is selected the image view will be replaced by another red image view. Also at the bottom of the screen a text view is provided which shows the current score of the user. Its text style is italic with bold. The method successFactor() is used whenever the score is updated.

6.1 Graphical User Interface

The following are the screenshots of the application on an android mobile phone.

6.1.1 Application Loading Screen

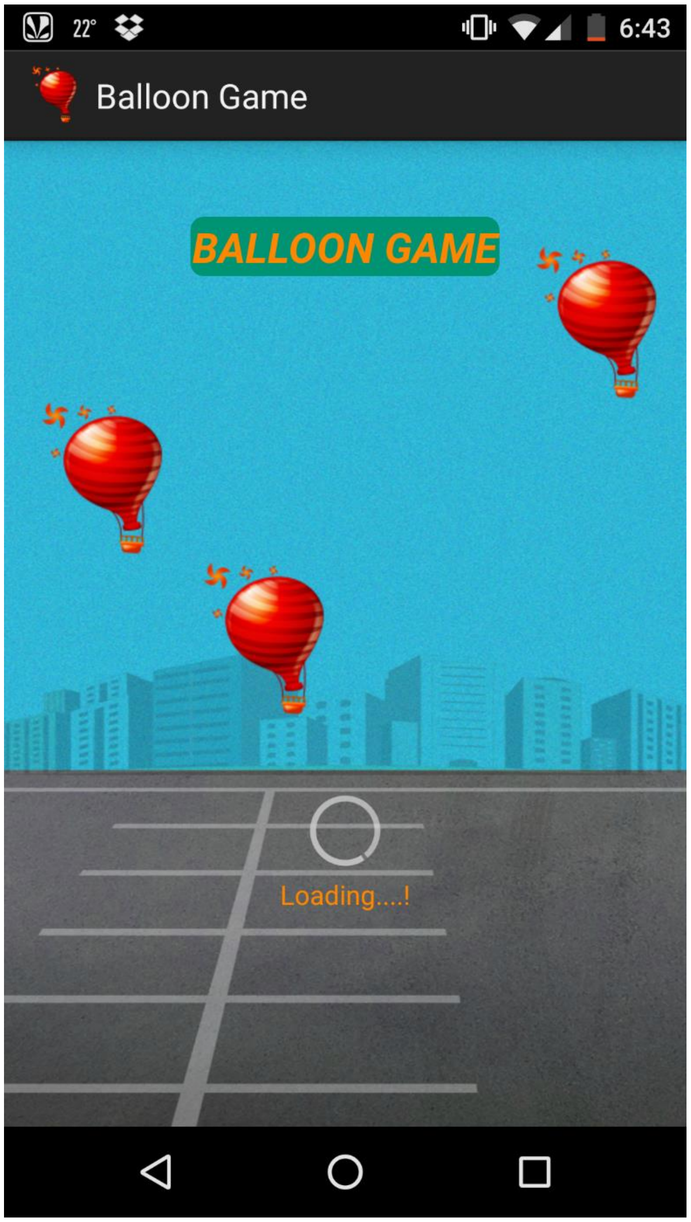


Figure 8 Application Loading Screen

This is the first screen that the user experiences when the application icon is clicked. In this screen, the application loads until the words for the balloon game are loaded into the SQLite Database.

6.1.2 Home Screen

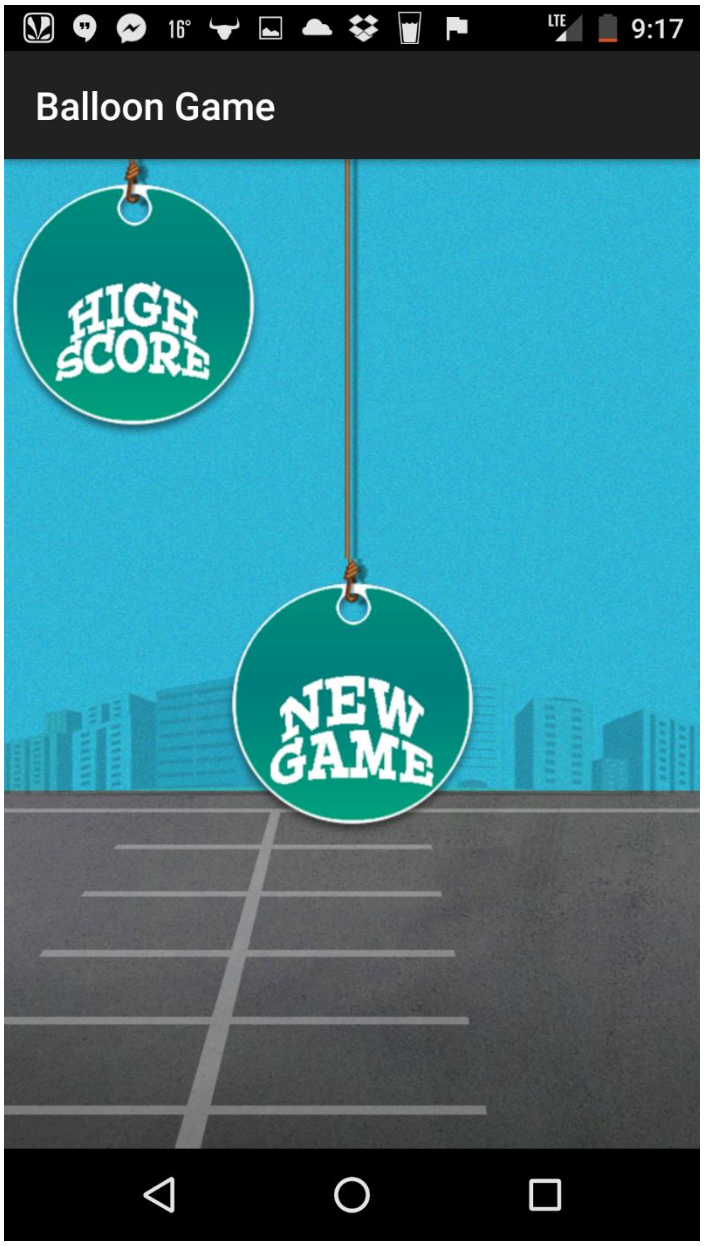


Figure 9 Home Screen

As soon as the word are loaded into the database. The Home Screen is displayed. The name of the activity is HomeScreen1.java. In this screen, the high score will consist of no entries if the user is playing for the first time. And on selecting the new game, the user will directed to the main application game. Here both the options are animated for the high score and new game and these are nothing but two image views. A third image view is also used for the thread on top of the new game option.

6.1.3 Game Mode Screen

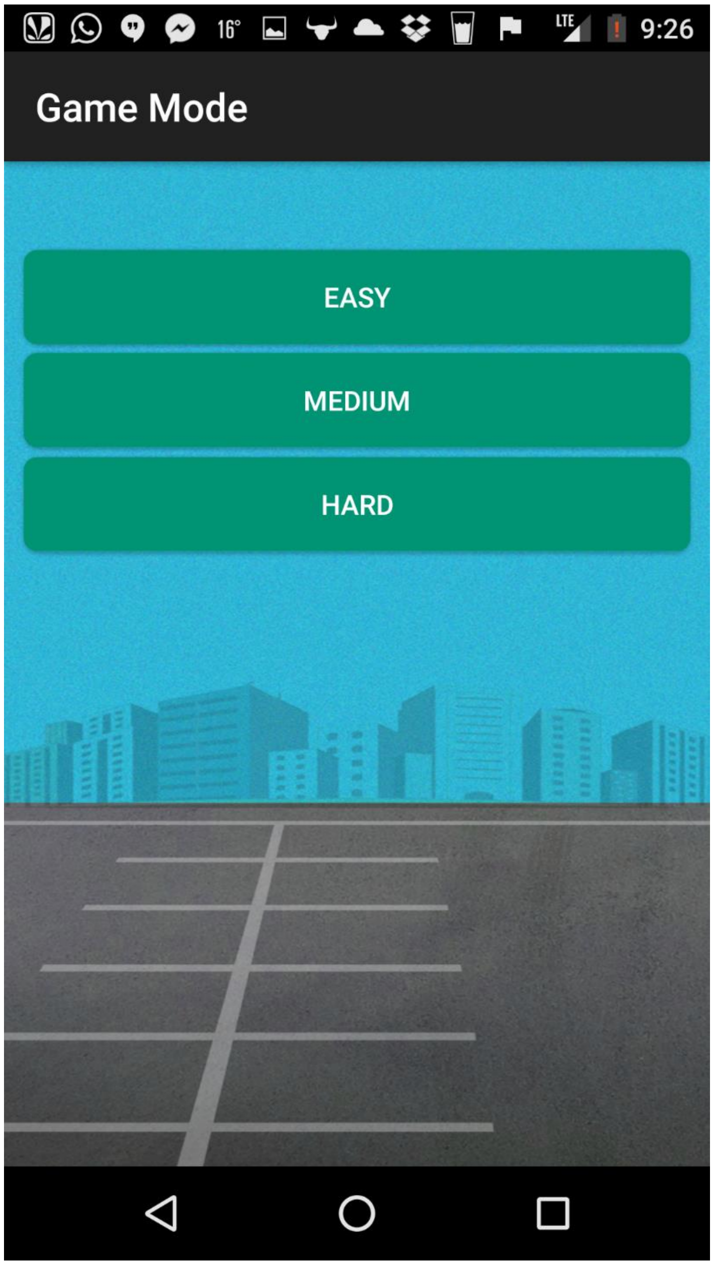


Figure 10 Game Mode Screen

On clicking the new game button, the button, the user is provided with three buttons which depict the difficulty of the application. The three buttons have been provided with animation and the buttons float on the screen. GameModeScreen.java is the name of the activity for this page. Also the concept of shared preferences is used in order to pass the values of lives from this activity to the main game activity.

6.1.4 Main Balloon Game Screen

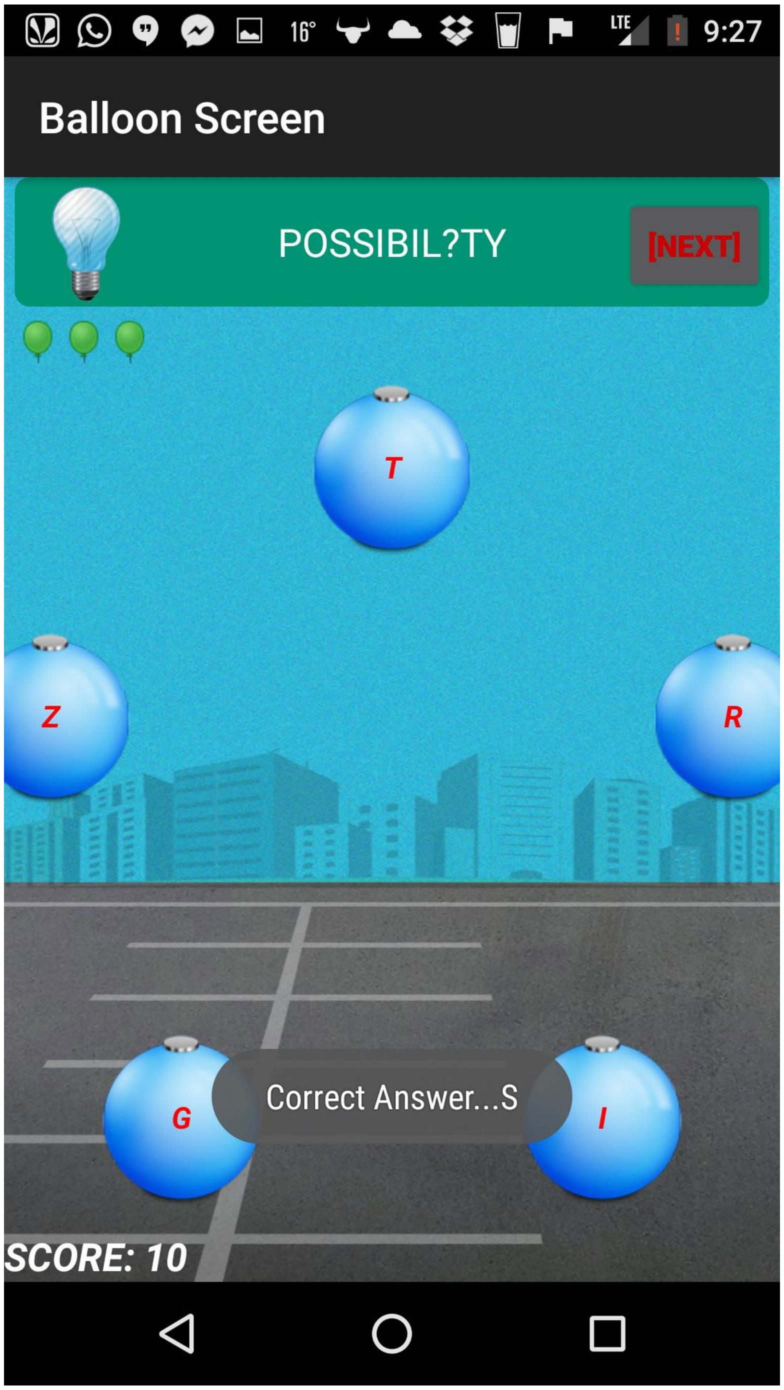
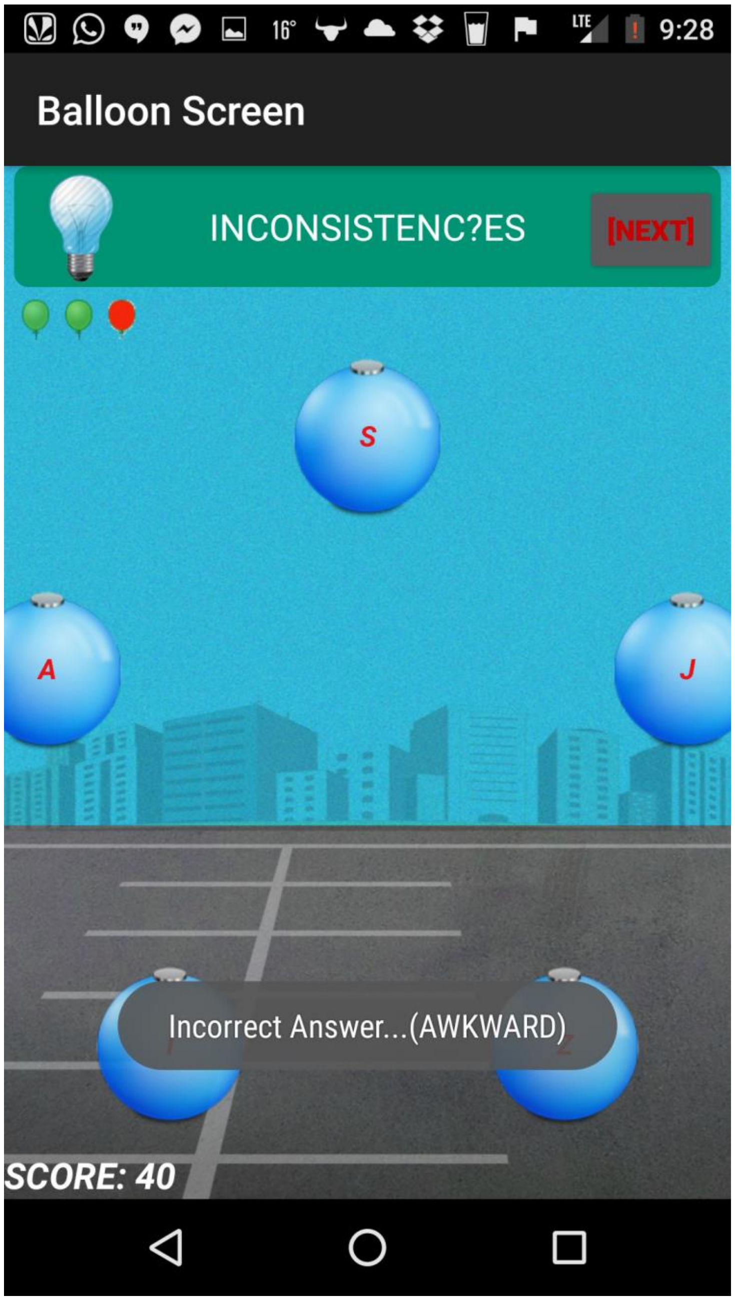


Figure 11 Main Balloon Game Screen

The above screen is the actual representation of the main game screen. The three small green balloons are the lives. Whenever the user selects a correct answer a toast message will be displayed with the missing letter. Also the score which is located at the bottom left will be incremented with ten points. Also there is a next button which will give the use the ability to skip a word.

6.1.5 Incorrect Answer Game Screen

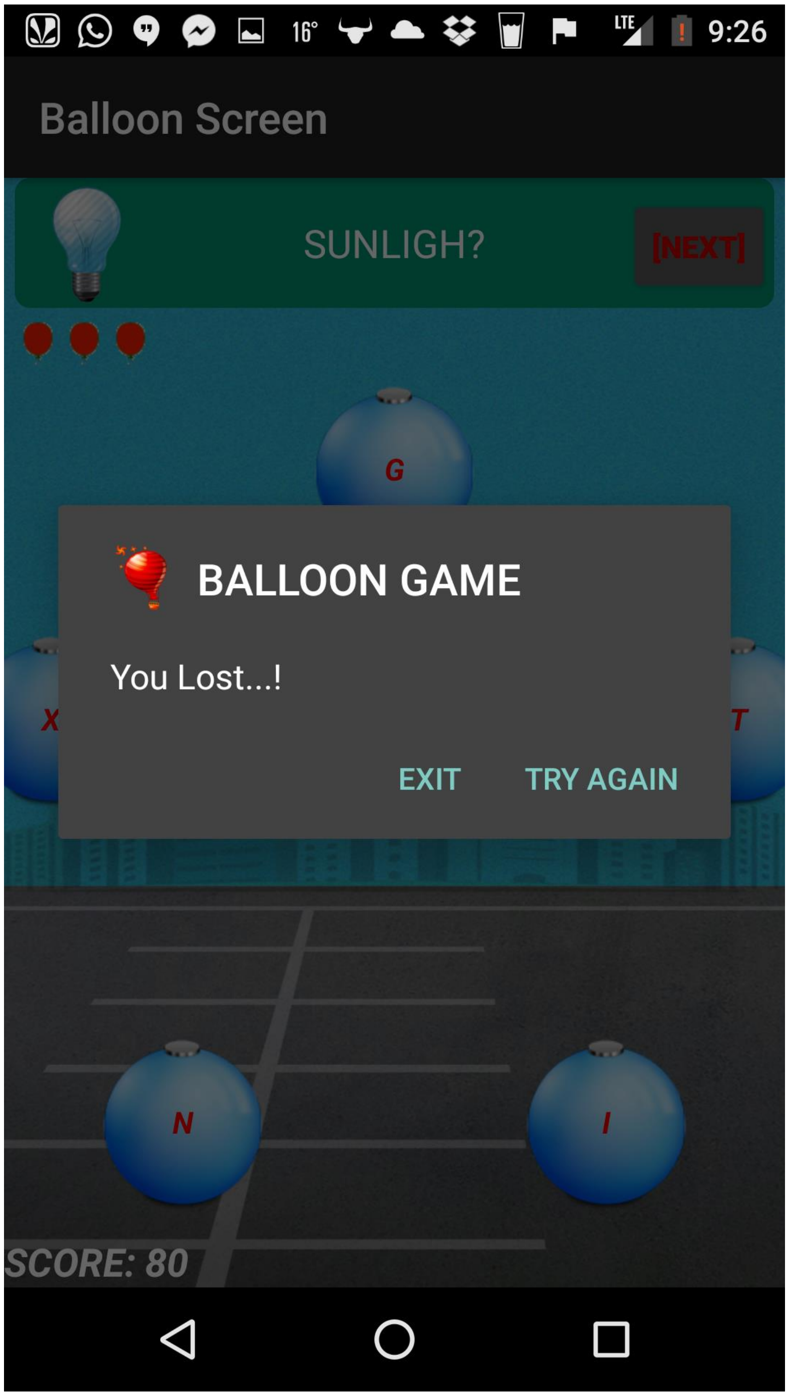


Incorrect Answer...(AWKWARD)

Figure 12 Incorrect Answer Game Screen

Now when the user selects a wrong answer, the user loses a life. And in order to make that visible, a green balloon in the screen will be replaced by a small red balloon and no points will be added to the score. Also, as soon as these things happen, a toast message is displayed stating that the selected answer is incorrect. It also displays the correct answer.

6.1.6 Game Over Screen



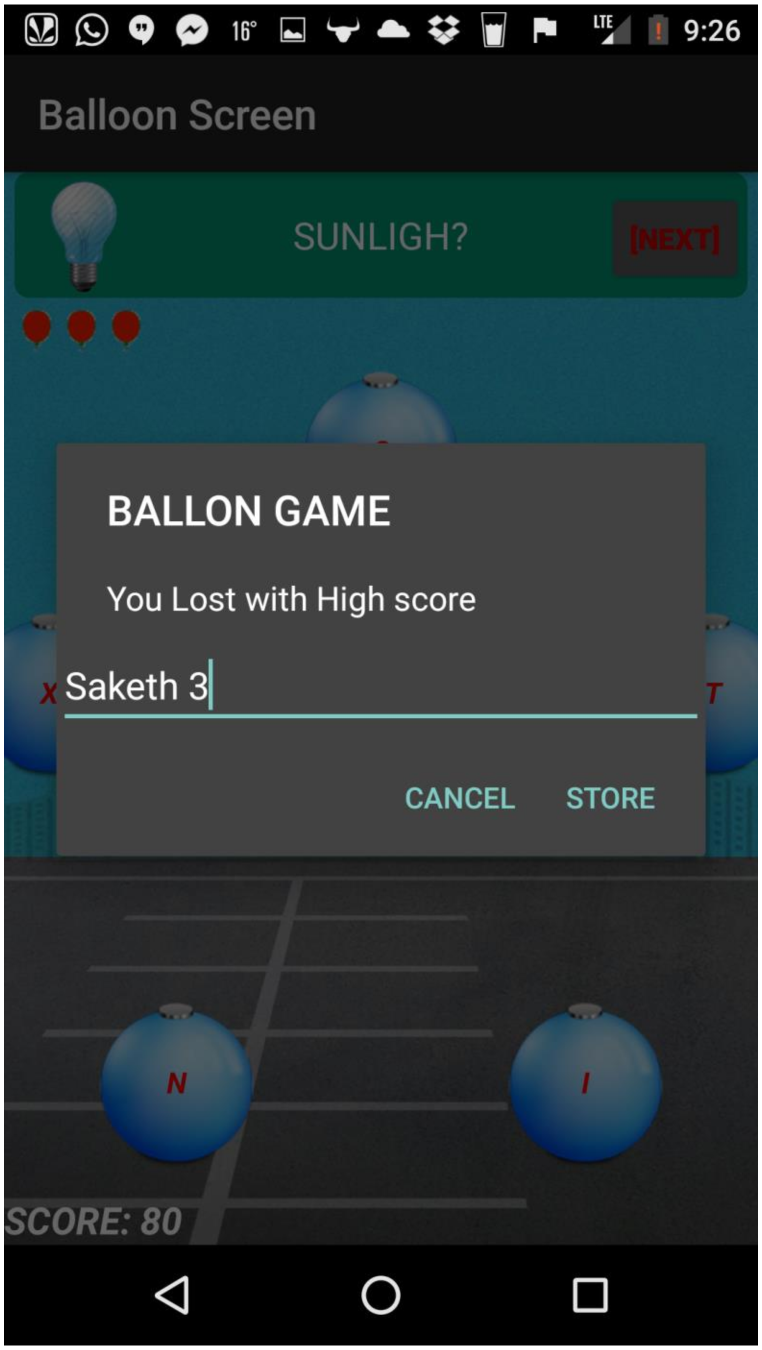
BALLOON GAME

You Lost...!

Figure 13 Game Over Screen

Now if the user loses all the three lives by selecting wrong answers, a game over message will be displayed. All the small green balloons will be replaced by the red balloons as an indication for game over. Also, there will be two options which are provided for the user. One of them would be to directly exit out of the game and the second one will be to try the game again.

6.1.7 Game Over High Score Screen



BALLON GAME

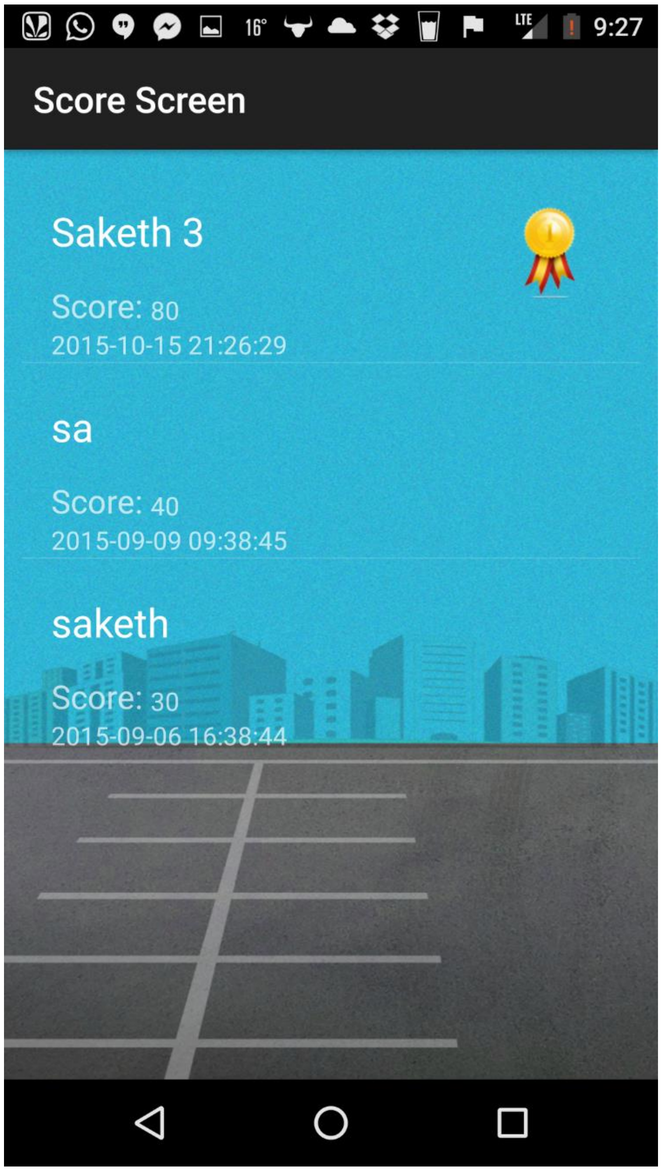
You Lost with High score

Saketh 3

Figure 14 Game Over High Score Screen

There are two different ways in which the user's game might end. The first one is a normal game over, where the user can try again or can exit. If the user happens to be the top scorer or the high scorer when all the lives are lost, then the incorrect answer toast message is displayed. Shortly after, a prompt a displayed where the user can enter their name to be displayed in the high score screen. The user also has an option of cancelling this prompt. Now apart from these features a tip feature is also included which when clicked would pronounce the word in order to help the user in guessing the missing letter correctly.

6.1.8 High Score Screen



Saketh 3

Score: 80
2015-10-15 21:26:29

sa

Score: 40
2015-09-09 09:38:45

saketh

Score: 30
2015-09-06 16:38:44

Figure 15 High Score Screen

Now, another main component of the application is the high score screen which is nothing but a list view of all the scores in the descending order. Each and every score listed here would generally comprise of the name of the user, the score they have achieved, the date on which the game has been played and also the time stamp. Also, a medal will be displayed adjacent to the top scorer or the high scorer of the game.

Chapter 7 - Testing

Testing an android application is an important aspect of delivering a reliable and robust application. Also, testing allows us to find any errors or missing requirements in the application. Using software testing one can evaluate the software from the requirements point of view. There are many kinds of testing an android application such as unit testing, integration testing, compatibility testing, etc. [8]

7.1 Unit Testing

The unit testing is done by the developers before supplying the software to the QA team. Unit testing is the process of separating small units of the application and then test them. For this application, the features of this application have been tested individually and then the test results are documented in the following table. [8]

Table 1 Unit Testing

S.no	Test Case	Expected Result	Result
1	On load of Main Activity	Display three floating balloons, a progress bar and a text view which displays Balloon Game. Also loads up the home screen.	Pass
2	On load of Home Screen	Two options, high score and new game are displayed with animations.	Pass
3	On click of High Score	Loads up the High score screen and displays empty list, since the game has not been played yet.	Pass

4	On click of New Game	Game mode screen is loaded. Displays three floating options for the difficulty of the game.	Pass
5	On selecting the Difficulty	Balloon Screen is loaded	Pass
6	On load of Balloon Screen	A tip icon is displayed, the word with the missing letter and the next button to skip the word is displayed. Three small green balloons are floating. Score and the options are displayed.	Pass
7	On selection of a correct answer	The score is incremented by ten points. The small green balloons are not replaced. A toast message is displayed. The next word is displayed.	Pass
8	On selection of wrong answer	The score remains unchanged. A small red balloon replaces the green balloon since the user loses a life. A toast message is displayed with the correct answer.	Pass
9	On selection of next button	The current word is replaced by a new word for the user to guess.	Pass
10	On selection of tip button	The current word is pronounced in order to provide a hint for the user.	Pass

11	On selection of three incorrect answers	A prompt is displayed with two options of exiting the game or trying the game again. Also displays a message in the prompt.	Pass
12	On selection of try again and exit	On selection of exit, the users comes out of the application. On selection of try again, the user goes back to the game mode screen.	Pass
13	On answering three incorrect answers with high score	A prompt message is displayed which states that the user has lost with high score. A place for the user to enter their name along with two options are displayed for the user which are store and cancel.	Pass
14	On selection of store	The applications stores the name of the user for the high score and takes back the user to game mode screen.	Pass
15	On selection of cancel	The application doesn’t store the name of the user and exits out of the game.	Pass
16	On selection of High Score	The High Score screen is loaded.	Pass
17	On load of High Score screen	A list view is displayed with scores in the descending order. A medal is displayed adjacent to the top scorer. The name of the scorer, the score, the date and the time at which the game has been played is displayed.	Pass

The application was also checked for compatibility issues. The application was installed on many other devices such as Nexus 5, HTC one mini, HTC M8, Galaxy S5, Galaxy S6 edge, Samsung Note 3, Samsung Note 5, etc. Also, the application was tested on different configurations of android virtual devices present in the ADT bundle.

Also, I have asked some of my friends to use the application for some time and provide me with some constructive criticism. One of the suggestion that I have got was improving the UI. Apart from that, they have reported that application runs without any lag. Also, two versions of the application were created based on the suggestions. It was the second version that has been made final. One such great suggestion that I received was the introduction of a hint feature which is nothing but the bulb button present in the application which was suggested by major professor Dr. Daniel Andresen. This hint feature pronounces the word to help the user. Overall, the application is responsive and also has rich user interface.

Chapter 8 - Conclusion and Future Work

Developing the Android Balloon game helped me gain a good amount of knowledge in the android development. Since this application is intended for children, figuring out the requirements was an essential part of the project. It was really a great experience for me in having the opportunity to learn the animations in Android. Another important aspect that I have got to learn from this application was figuring out the system design. Since this application is entirely based on the text to speech engine, I had the opportunity of learning that API. Also, learning the concept of android shared preferences helped me a lot in implementing some of the tasks where some data has to be passed from one activity to another activity without the use of a database. Apart from all these, building this application also enhanced my understanding of developing an android application which uses the SQLite database for the management of data.

There can be some enhancements added to this application in the future. One of the enhancement could be a timer based module, where the user has to answer as many questions as possible in a given amount of time such as a minute or two. And then the game would also consist of another high score screen under the timer based category. Apart from the English words, the application can also include basic mathematics in the future. Some of the operations that can be included are addition, subtraction, multiplication and division of numbers. These above future enhancements can make this application much more helpful for the children who can also improve their mathematical skills.

Chapter 9 - Bibliography

[1] "Android (Operating System)", Online, March 23, 2015

http://en.wikipedia.org/wiki/Android_%28operating_system%29

[2] "Android Architecture", Online, March 23, 2015

http://www.techdesignforums.com/edasource/images/68/esd1004_mentor1_large.jpg

[3] "Android Architecture", Online, April 2, 2015

http://www.eazytutz.com/android/android-architecture/

[4] "Android Director Structure", Online, April 2, 2015

http://www.javatpoint.com/internal-details-of-hello-android-example

[5] "UML Diagrams", Online, April 4, 2015

http://www.smartdraw.com/uml-diagram/

[6] "Android Application Components", Online, April 6, 2015

http://www.tutorialspoint.com/android/android_application_components.htm

[7] "Android Activity Example", Online, April 8, 2015

http://developer.android.com/guide/components/activities.html

[8] "Software Testing", Online, April 8, 2015

https://en.wikipedia.org/wiki/Software_testing