

**ANDROID BASED MOBILE APPLICATION
LEARN ENGLISH**

BY

**MD. TAUHID ISLAM
ID: 162-15-8107**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised
By

Mr. Aniruddha Rakshit

Senior Lecturer

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
JULY 2020**

APPROVAL

This Project/internship titled “**Android based mobile application Learn English**”, submitted by Md. Tauhid Islam, ID No: 162-15-8107 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 08 July 2020.

BOARD OF EXAMINERS



Dr. Syed Akhter Hossain
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Gazi Zahirul Islam
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

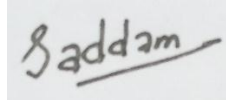
Internal Examiner



Abdus Sattar
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Saddam Hossain
Assistant Professor

Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

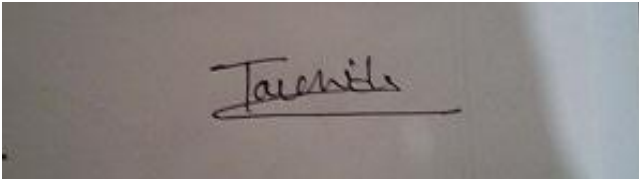
We hereby declare that, this project has been done by us under the supervision of **Mr. Aniruddha Rakshit, Senior Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Mr. Aniruddha Rakshit
Senior Lecturer
Department of CSE
Daffodil International University

Submitted by:



Md. Tauhid Islam
ID: 162-15-8107
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Mr. Aniruddha Rakshit, Senior Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Android Application” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement ,constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude Dr. Syed Akhter Hossain, Professor and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

We all know English as an international language, so to keep pace with the world. We need to learn English as well as our own language. Nowadays we all use mobile phones for various purposes. Mobile App Is an Easy way to connect a user to a phone. Since we can easily communicate with everyone through the mobile app, for this we have created a mobile app to teach English. Any user can download our application free from the Play Store. After Download you can easily learn English using our application. Our application is developed for beginners users. A new user can easily learn English grammar from basic to advance. You can use our apps offline online in two ways. We will add extra new features in the future.

TABLE OF CONTENTS

CONTENTS	PAGE NO
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
CHAPTER	
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcome	2
1.5 Report Layout	2
CHAPTER 2: BACKGROUND	3-4
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Studies	3
2.4 Scope of the Problem	4
2.5 Challenges	4
CHAPTER 3: REQUIREMENT SPECIFICATION	5-8
3.1 Business Process Modeling	5
3.2 Requirement Collection and Analysis	6

3.3 Use Case Modeling and Description	6
3.4 Logical Data Model	8
3.5 Design Requirements	8
CHAPTER 4: DESIGN SPECIFICATION	9-19
4.1 Front-end Design	9
4.2 Back-end Design	16
4.3 Interaction Design and UX	19
4.4 Implementation Requirements	19
CHAPTER 5: IMPLEMENTATION AND TESTING	20-21
5.1 Implementation of Database	20
5.2 Implementation of Front-end Design	20
5.3 Implementation of Interactions	20
5.4 Testing Implementation	21
5.5 Test Results and Reports	21
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	24
6.1 Discussion and Conclusion	24
6.2 Scope for Further Developments	24
REFERENCES	25
APPENDICES	26-29

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Block diagram of BPM	5
Figure 3.2: Use Case Diagram Administrator	7
Figure 3.3: User access use case diagram	7
Figure 3.4: Logical Data Model Diagram	8
Figure 4.1: (a) Home Activity Layout (b) Navigation Drawer Menu Layout	9
Figure 4.2: (a) Sentence Activity (b) Interrogative Activity Screen	10
Figure 4.3: (a) Part Of Speech Screen and Menu (b) Noun Screen	11
Figure 4.4: (a) Tense Activity Screen (b) Present Tense Screen and Menu	12
Figure 4.5: (a) Past Tense Screen and Menu (b) Past Indefinite Activity	13
Figure 4.6:(a) Future Tense Screen and Menu (b) Future Indefinite Activity	14
Figure 4.7: (a) Picture Dictionary View and (b) Translate Screen	15
Figure 4.8: Main Activity	16
Figure 4.9: Alphabet Model Java	16
Figure 4.10: Adapter Class Java	17
Figure 4.11: Menu Item XML	17
Figure 4.12: Button Design XML	18
Figure 4.13: Navigation Header XML	18
Figure 4.14: Main Activity Test	22
Figure 4.15: Part Of Speech Activity Test	23
Figure 4.16: Tense Activity Test	23

LIST OF TABLES

TABLES

PAGE NO

Table 5.1.1:Unit Test Case for Main Activity Java File

22

CHAPTER 1

Introduction

1.1 Introduction

Learn English is an android based Mobile application. You can learn English grammar very easily using these apps. In our application, various lessons of English Grammar are beautifully arranged.

Which programs are available in our app?

- Alphabet
- Sentence
- Part Of Speech
- Tense
- Article
- Picture Dictionary
- Translator

1.2 Motivation

There is no good application about English grammar in Bengali on Google Play. We created this application for learning English through the application of Bengali speaking people. If anyone can learn something through our application then we are successful.

1.3 Objectives

The main aim of building our application is to make education easy for everyone. We don't have to pay a fee to use the application; we have made it free so that everyone can use it easily. We want everyone to use our app and share it with others.

1.4 Expected Outcome

- Applications can be downloaded from anywhere.
- After downloading the application can be used offline Mode.
- You can study at any time
- High end low end will run on all types of devices
- Different words can be easily learned through Picture

1.5 Report Layout

Frist chapter we have already described motivation, purpose, and expected outcome of our app Project. In the second chapter, shortly discuss objections, involved works, the scope of the problem, and comparative studies of our App Project. Then we discuss more about the Requirement Specification of our project 3rd Chapter. In Chapter 4th we have displayed our application user interface, Front end back and Back end Design. Section 5th is an implementation and Testing Stage. In the last Chapter, 6th discuss about conclusion and Future developments. In the last part, we added references and Report related Information of Our Learn English app Project.

CHAPTER 2

Background

2.1 Introduction

At present most of the time people spend on mobile. There are many things that can be done with mobile now, such as: interacting with others, banking, reading, listening, etc. Since people are spending more time on mobiles and doing a lot of work with them, we think of teaching English to people through mobile applications. We create Learn English applications with this idea in mind. We hope that our application will benefit people.

2.2 Related Works

Google Android OS runs on a diversity of devices that have various screen sizes and pixel densities. The system execute fundamental scaling and resizing to adjust your user interface to various screens density, but there is more work you should do to ensure your UI gracefully adapts for each type of screen [1].

2.3 Comparative studies

The comparative system is often used in the early platform of the development of a branch of science. It can help the researcher, developer, and programmer to ascend from the initial level of exploratory case studies to a more advanced level of general theoretical models, invariances, such as causality or development [2]. Most people use the Hello English app to learn English but the Hello English app does not teach grammar so we are developing a different kind of app

2.4 Scope of the problem

Learn English app is a Learning stage “Android OS Based Mobile Application”. It is principle language for Front-End design develops will be XML and Back-end Development will be Java. We are using material design for interactive UI. We have designed it so that all types of users can easily use the application.

2.5 Challenges

There is a typical thing to play out a movement is an objection. When we are developing our LearnEnglish Application we additionally face a few problems. The 1st problem is the syllabus design. We are designing a simple syllabus so that beginners can learn English very easily. The second challenge is the offline app design. We keep a local database of all content for offline apps.

CHAPTER 3

Requirement specification

3.1 Business process Modeling

BPM is a process management method that displayed the process and Data of the system. BPM displayed a simple visual model using UML language [3]. If you are using BPN of your project management system you can better understand your process and Data. We are using (BPM) of our LearnEnglish App Project that displayed Our Application Process. In this data model start activity is a main process and Menu, items, buttons are sub process main processes that are control sub and other processes.

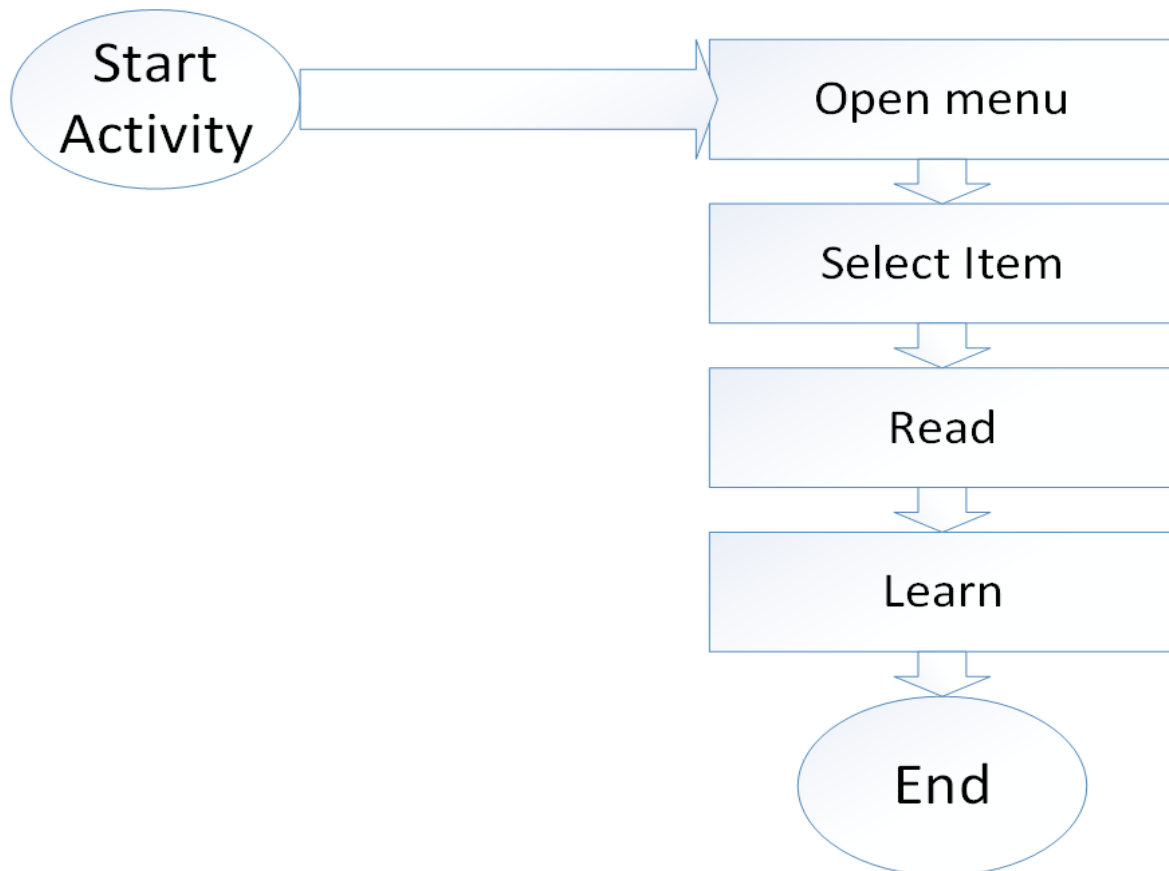


Figure 3.1: Block diagram of BPM

3.2 Requirement collection and Analysis

SDLC is the Process used to Software Company for their Software Development. Project managers, Business organizations, and Software Development Company use SDLC. The Requirement collection and Analysis is the 1st phase SDLC and it is the most important part of SDLC [3]. We are collecting our requirements interviewing methods. We are interviewing a lot of language learning user and know their requirement. After that analysis answer and Make a Plan for our Learn English App Project.

3.3 Uses-Case modeling & Description

Use case modeling diagram displayed the dynamic behavior of the system when all System is working. There are five types of common diagrams to model the dynamic nature of a system; use case diagram is one of them. Three items use to draw use case design of a system.

- Functionalities
- Actors
- Relationships

Functionalities to be represented as use case and actor to be represented a human user internal applications. We are used to use case Diagram Requirement analysis. Model of the system and reverse forward engineering [4].

Administrator use case Diagram displayed admin access LearnEnglish App Project.

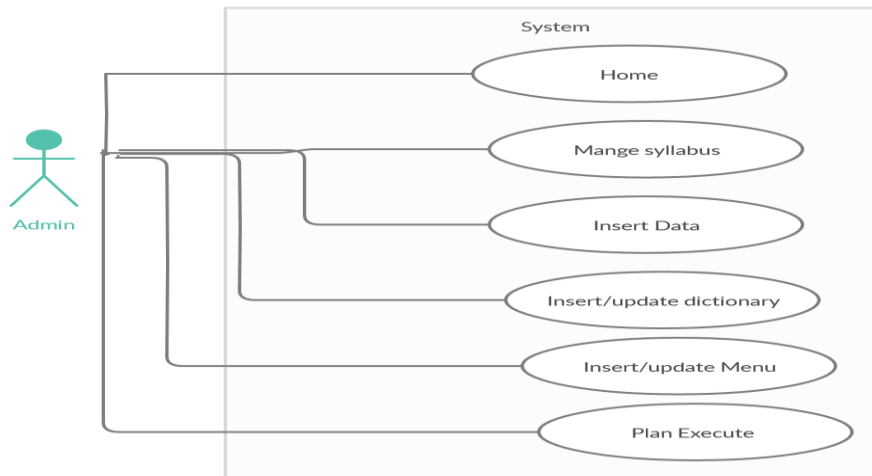


Figure 3.2: Use Case Diagram Administrator

User use case that is represented user action on LearnEnglish App. User able to access all user interface design part by part.

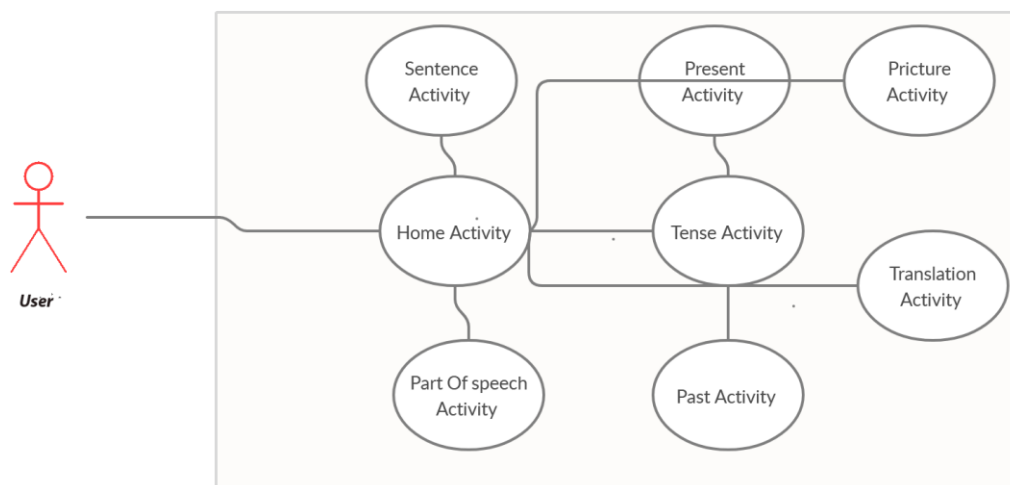


Figure 3.3: User access use case diagram

3.4 Logical Data Model

When open application at first Load Main Activity Java File then Load Other UL components.

s

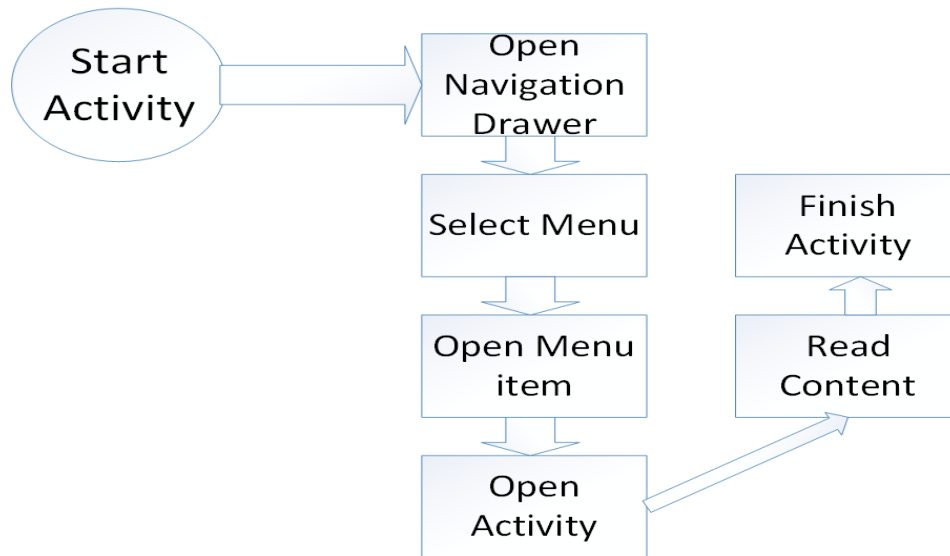


Figure 3.4: Logical Data Model Diagram

3.5 Design Requirements

Our application builds in android state so we are using the latest SDK and we are also using android Studio for design. Google developer Android studio support XML language so our application user interfaces design we are using XML and material design that language help to build to beautiful app interface. We are also implementing a new navigation app drawer UI that use to modern and latest application design. XML File, Graphics, and icon logo located Res Folder in Android Project.

CHAPTER 4

Design Specifications

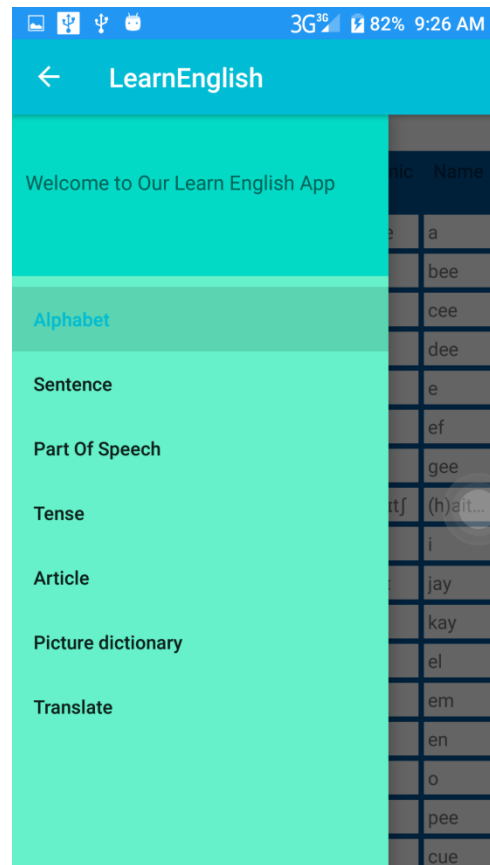
4.1 Front End Design

Learn English android App Front End Design Interface build in XML Language. Home and Navigation Drawer Menu Layout located in res folder Learn English app Project.



sNo	Capital Letter	Small Letter	Phonic	Name
1	A	a	eɪ, æ	a
2	B	b	bi:	bee
3	C	c	si	cee
4	D	d	di	dee
5	E	e	i	e
6	F	f	ɛf	ef
7	G	g	dʒi:	gee
8	H	h	(h)ertʃ	(h)ait...
9	I	i	aɪ	i
10	J	j	dʒeɪ	jay
11	K	k	keɪ	kay
12	L	l	el	el
13	M	m	ɛm	em
14	N	n	ɛn	en
15	O	o	ou	o
16	P	p	pi	pee
17	Q	q	kju	cue

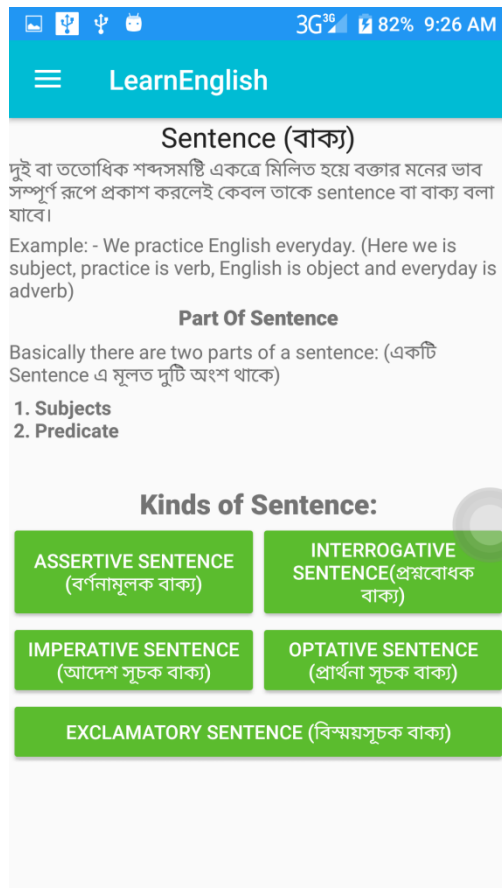
(a)



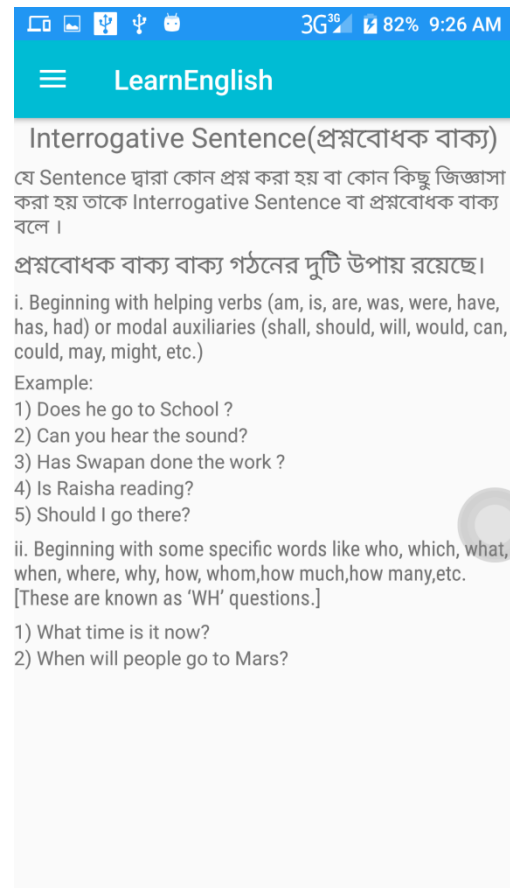
(b)

Figure 4.1: (a) Home Activity Layout (b) Navigation Drawer Menu Layout

Sentence Activity and Integrative Layout XML design. Its build-in XML and Google material design that activity displayed about sentence Content.



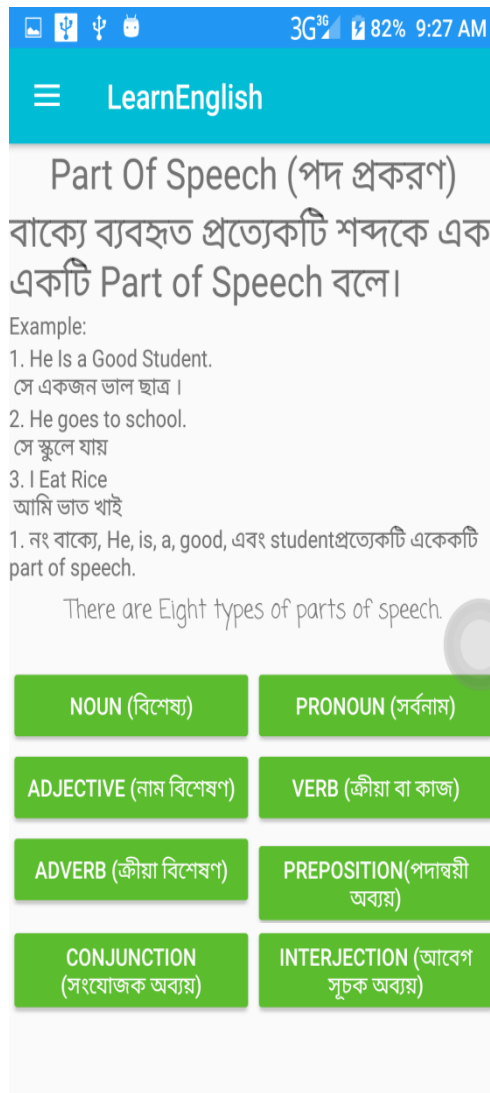
(a)



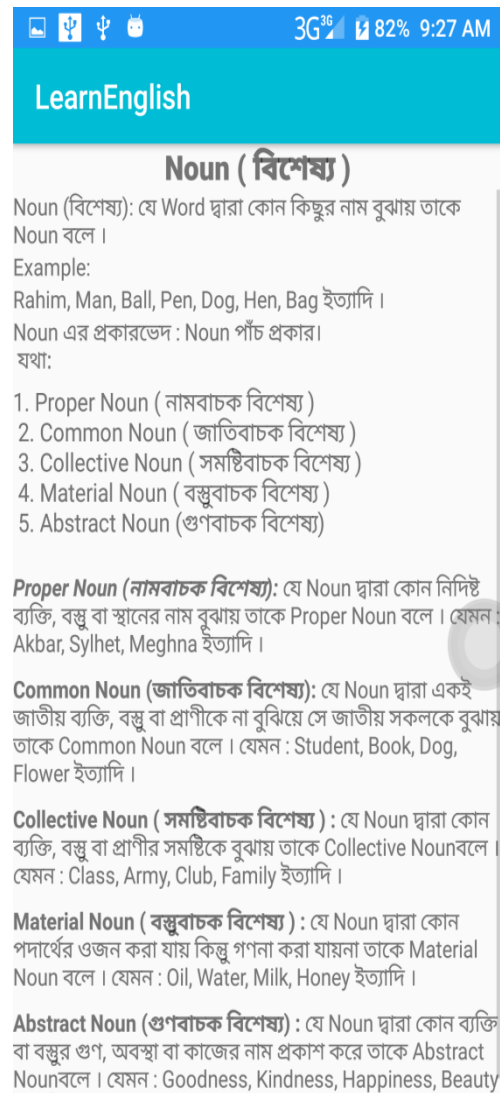
(b)

Figure 4.2: (a) Sentence Activity (b) Interrogative Activity Screen

In this Step we discuss part of speech and every part in Part of speech design.

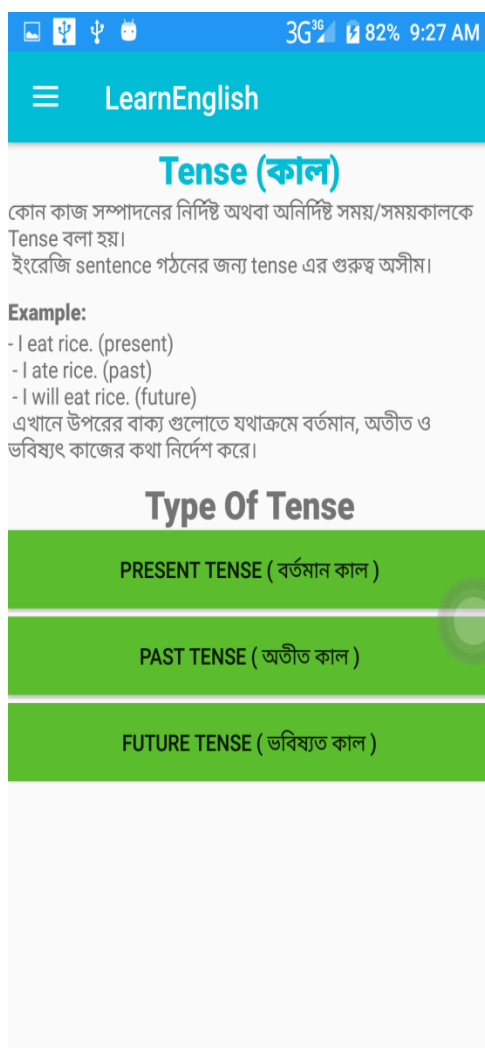


(a)

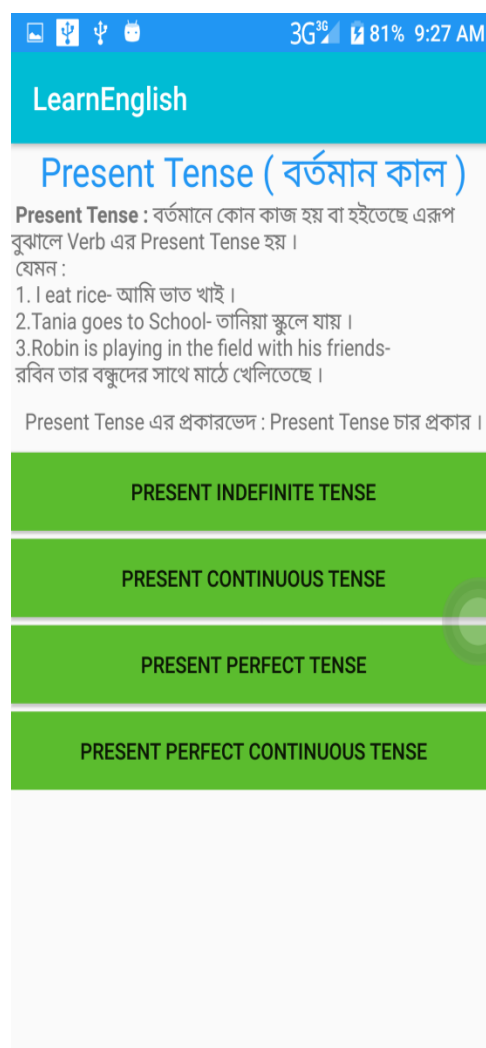


(b)

Figure 4.3: (a) Part Of Speech Screen and Menu (b) Noun Screen

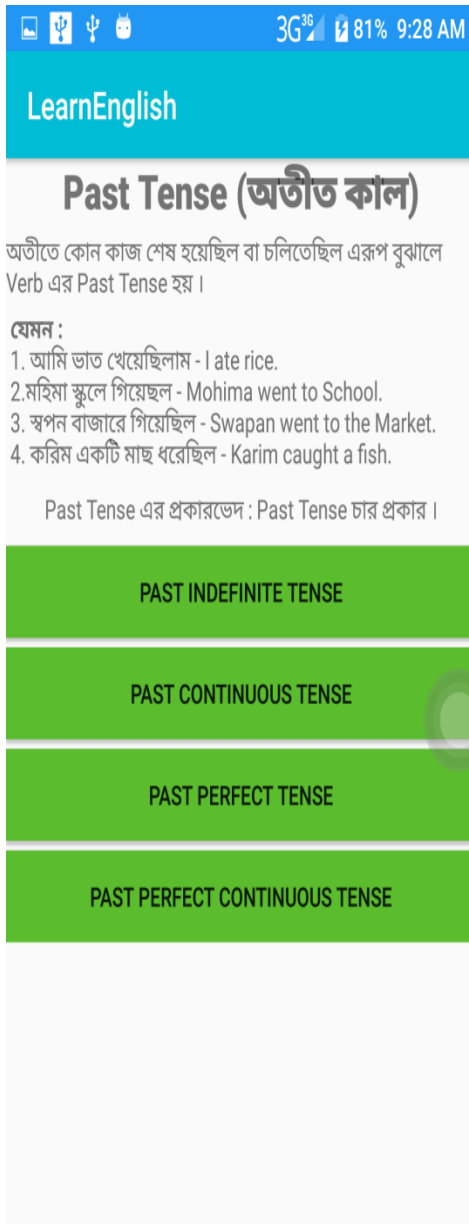


(a)

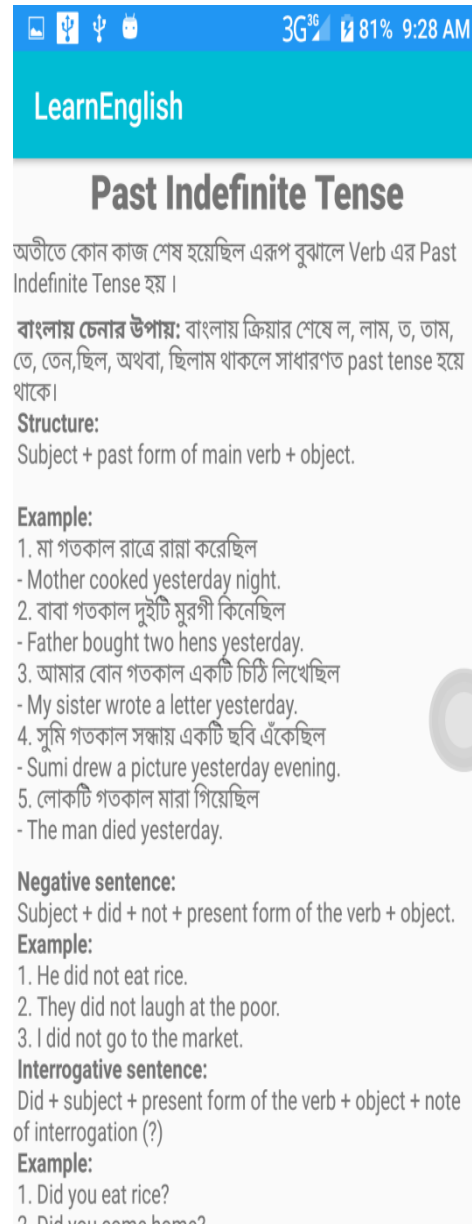


(b)

Figure 4.4: (a) Tense Activity Screen (b) Present Tense Screen and Menu

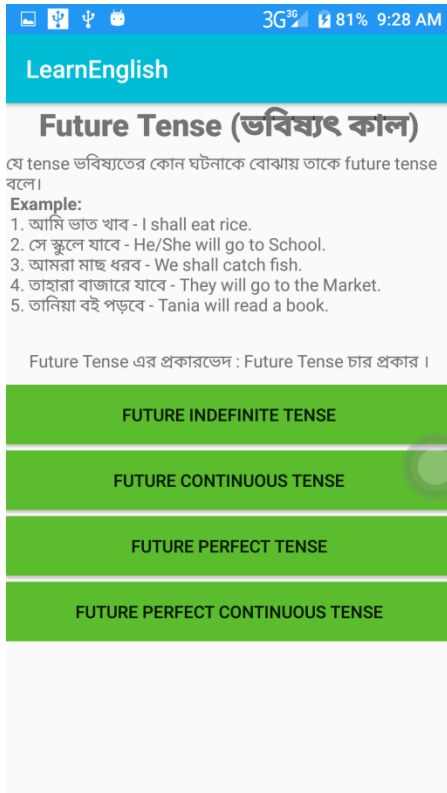


(a)

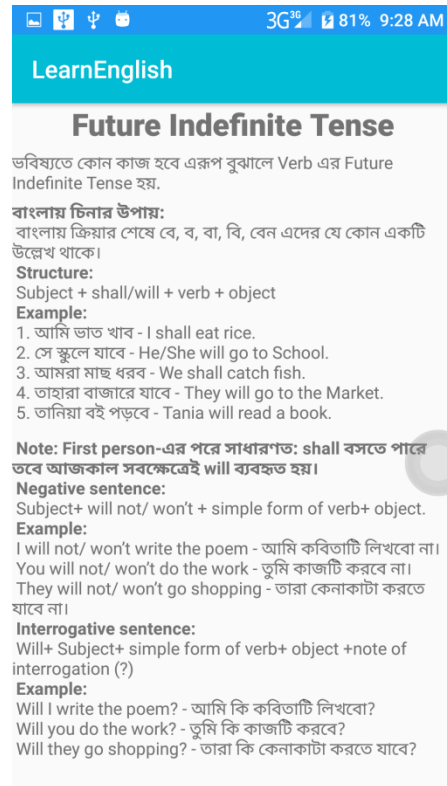


(b)

Figure 4.5: (a) Past Tense Screen and Menu (b) Past Indefinite Activity



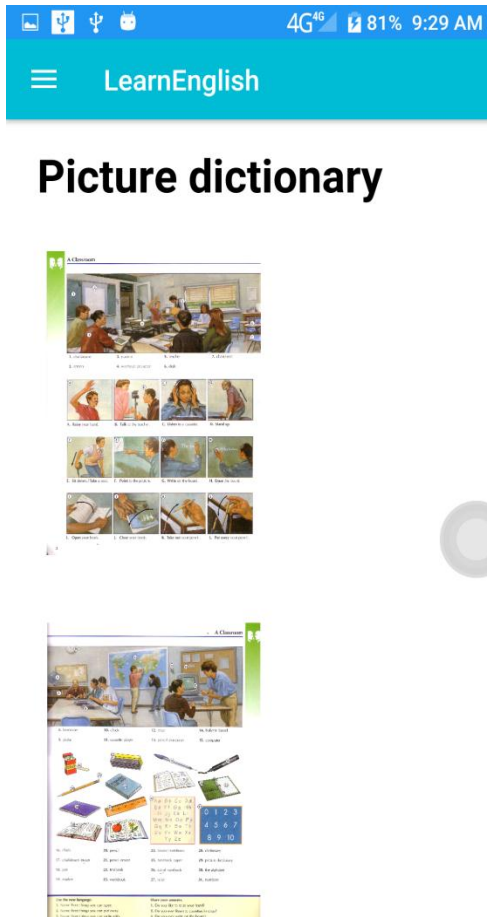
(a)



(b)

Figure 4.6: (a) Future Tense Screen and Menu (b) Future Indefinite Activity

Picture dictionary Front end Design Interface.it option help to learn new World see picture.



(a)



(b)

Figure 4.7: (a) Picture Dictionary View and (b) Translate Screen

4.2 Back End Design

We are using Java Language and Android SDK Our Application Backend Development.

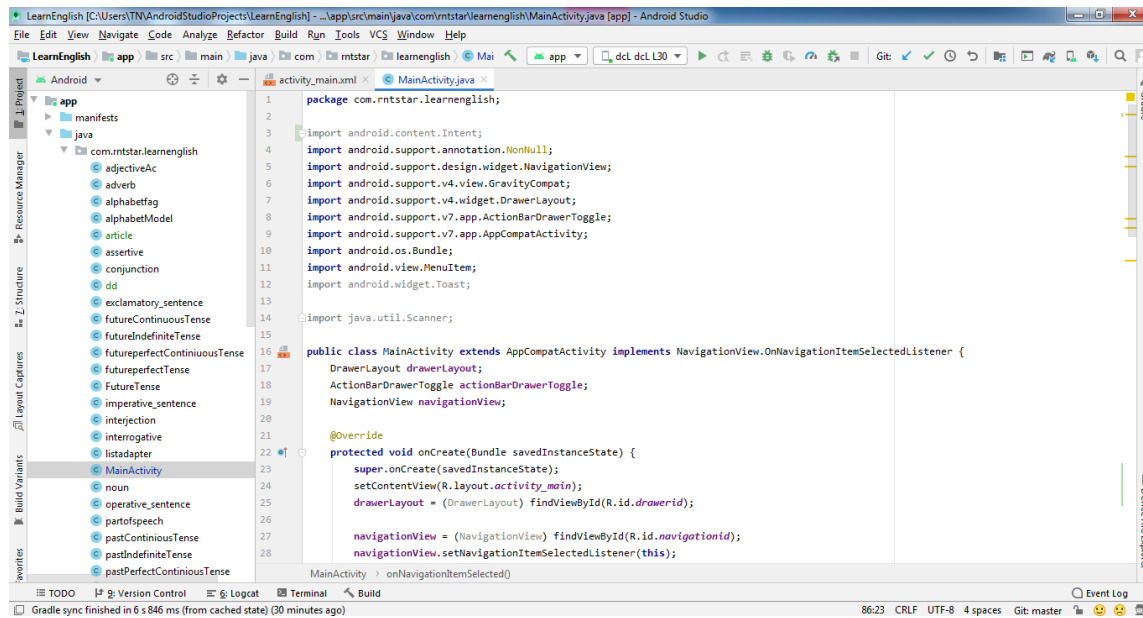


Figure 4.8: Main Activity

Alphabet Model java Class represented a Model Alphabet Activity Layout and collect data from database to show Alphabet Activity.

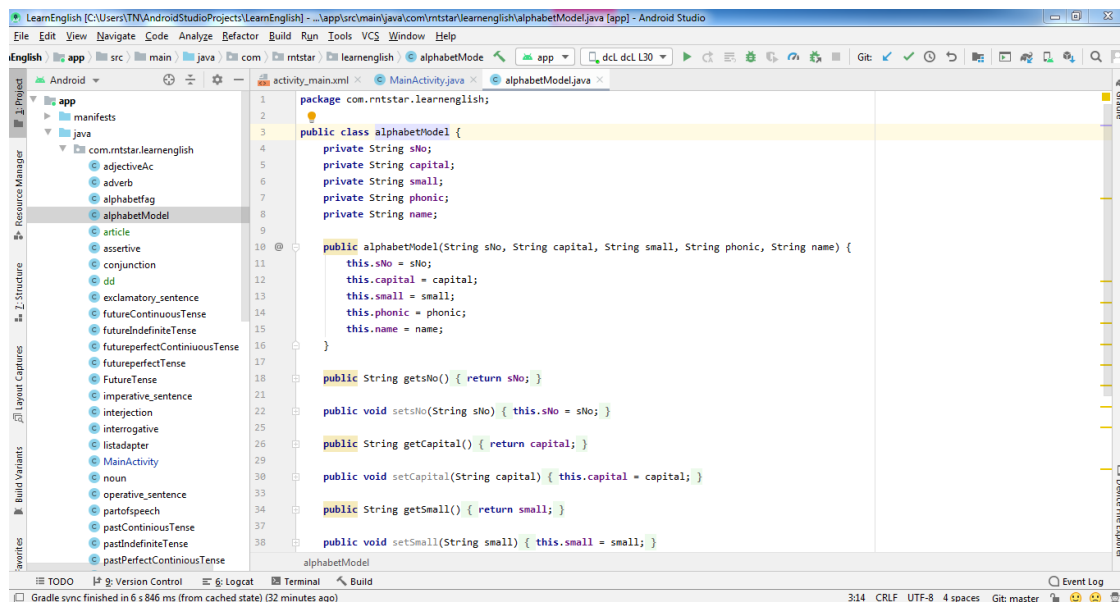


Figure 4.9: Alphabet Model Java

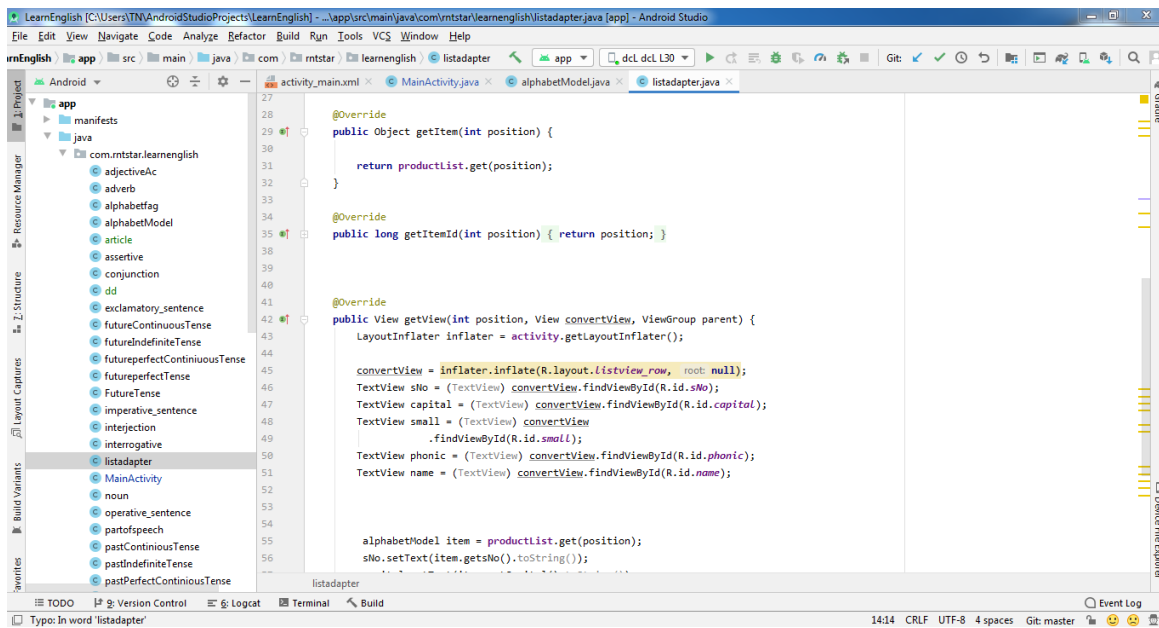


Figure 4.10: Adapter Class Java

Application Menu item that displayed menu LearnEnglish App. Menu item provides Navigation app menu Application app Drawer.

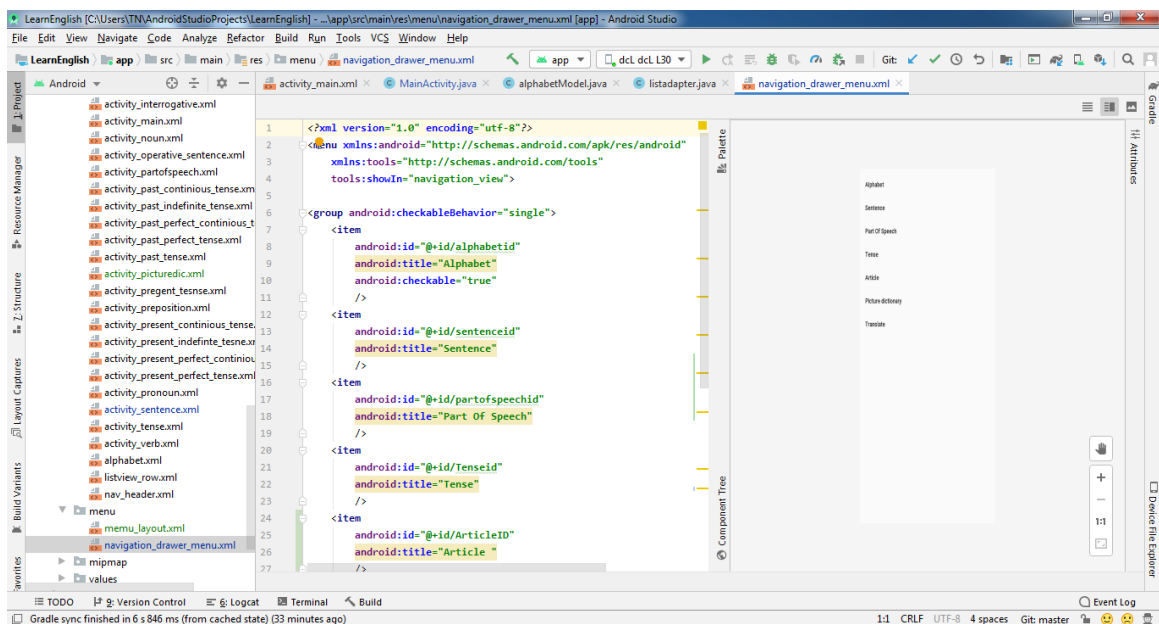


Figure 4.11: Menu Item XML

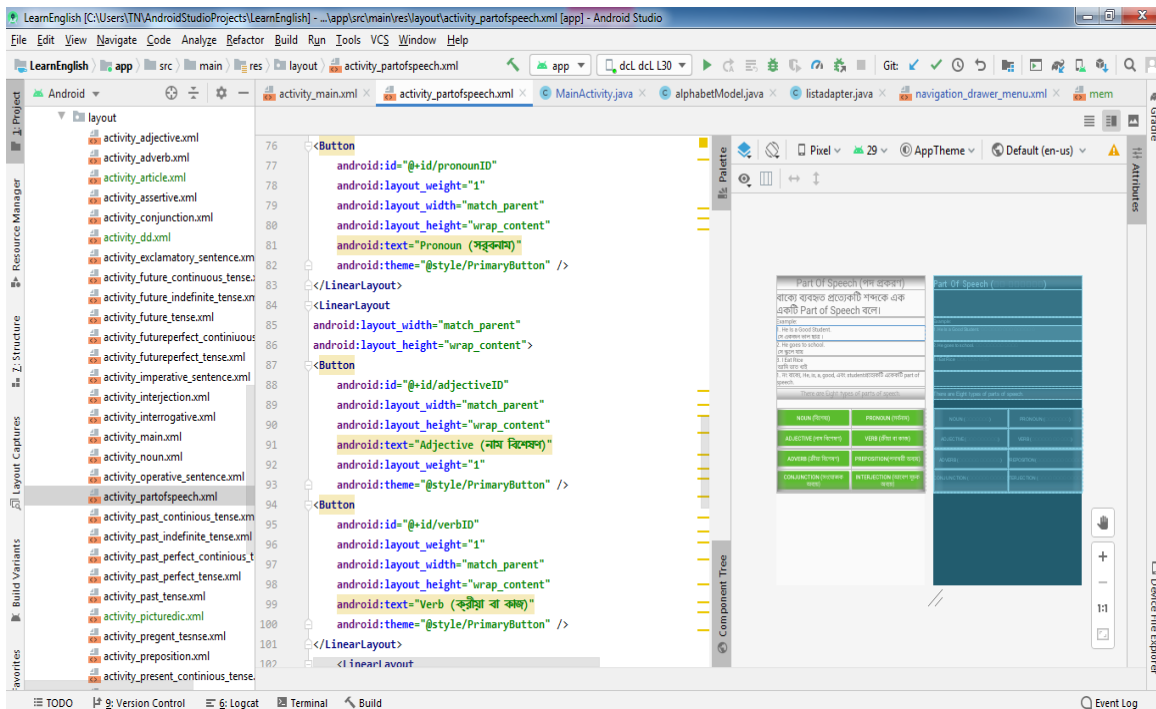


Figure 4.12: Button Design XML

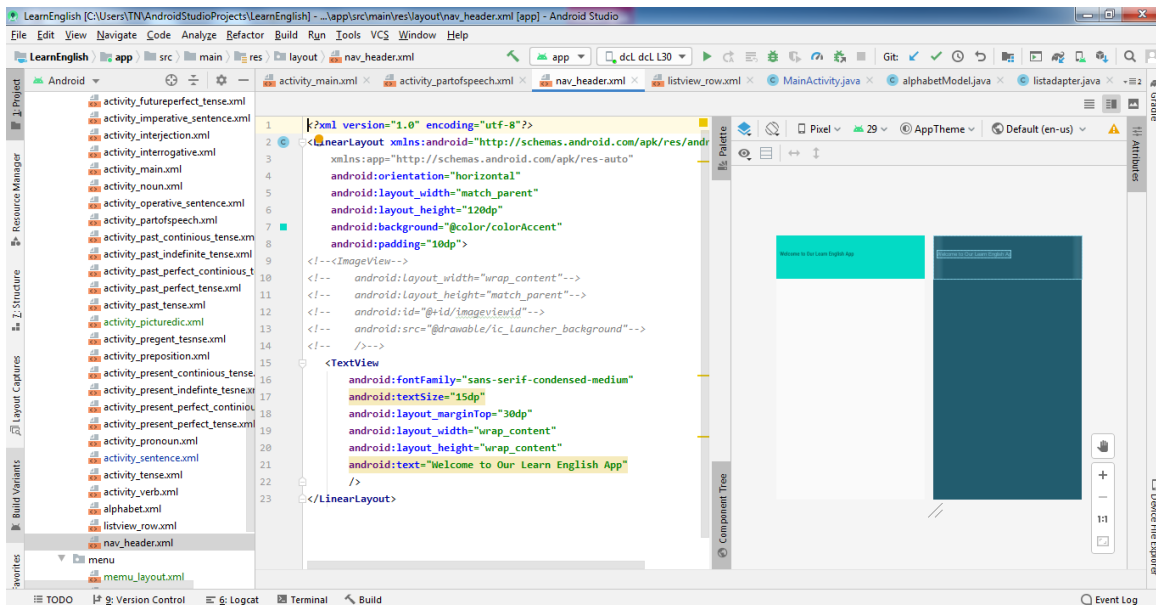


Figure 4.13: Navigation Header XML

4.3 Interaction Design and UX

A lot of people think that Interaction design and UX design are the same but not really the same because Interaction designs relation between user and system and other things UX or user experience design is the overall experience on system or product [5]. When we build our application we are focusing on UX and interaction design. our application firstly responds to user interaction because we are using sensors and the latest API Google material design in our system. Example: when the user rotates their hand application automatic rotate screen. We've developed user experiences with feedback from various kinds of users using our application.

4.4 Implementation Requirements

Implementation is a construction part of a system. When you work on execution, you have to a lot of work on an idea, design, method, and other related information about your project. Our mobile application LearnEnglish build an open-source Android platform. The tools and technology are given below to develop our LearnEnglish android application.

These are the:

- Android SDK
- Google develop Android Studio
- Java
- SQLite.
- HTML

CHAPTER 5

Implementation and testing

5.1 Implementation and Database

We are using SQLite open-source relational database in our LearnEnglish App Project. It is attached to Google android OS by default so there is no extra database setup. SQLiteOpenHelper class supplies the functionality to use the SQLite database. The android.database.sqlite. we are using the SQLiteOpenHelper class used to database creation and version management systems. For performing any database operation, you have to provide the implementation of onCreate() and onUpgrade() methods of SQLiteOpenHelper class [6]. We keep all the data of our application in the local database so that the data stored without an internet connection can be viewed by our application.

5.2 Implementation of front-end Design

We are developing our LearnEnglish application in two parts. one is Front end and the other is Backend Design. Front end interface is a visual part and back end part work logical and functional work of a system. We are using Liner layout, list view, Fragment, button, Text view, Table layout, and navigation app drawer build user interface. All user interface front end design generates XML. We are also using images, Graphics, and material for interactive UI Design.

5.3 Implementation of Interactions

Hardware and software combination is An Implementation of Interactions that element provides a way for smartphone users to accomplish a single task. For real Life example: when you use your smartphone click an activity go to next activity back to the previous activity, clicking a button, inter number pressing key, gesture, or inter your own commend. An interaction is a single task with user order or scenarios [7]. When we are building our application always keep and our mind user Interactions and Implementation of our app.

5.4 Testing Implementation

Testing the main aim is to recognize and Errors. When we build a system then we test the system and identify faults and solve the problem. Testing help to improve software quality and performance. different types of testing show different types of test results [8].

Various kinds of Software testing in below

Functional Test types List:

- Unit Testing
- Integration Testing
- System Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance Testing

Non-functional test types List:

- Performance Testing
- Compatibility Testing
- Install Testing
- Reliability Testing
- Security Testing

5.5 Test Results and Reports

Unit Testing Reports

The test is a bug finding method. When our application builds complete then we are starting to test our application and searching the bug and solve the error. We are testing our app various types of bug finding methods and solve problems and error. We are also testing our app junit4 and JVM. Junit4 frameworks help to easy bug fixing that tools are very helpful.

Table 5.1.1: Unit Test Case for Main Activity Java File

SN	1
Name of test Case	Main Activity XML Test Case
Sample Input Test Case	Action Menu after that runs Main Activity.java
output Result	Main Activity GUI should display in below
Actual output Test Case	Login GUI displayed as expected.
Remarks:	pass

Espresso Test:

Espresso Test Recorder is a new feature released in Android Studio 2.2 Preview 3 that makes it easy to generate automated UI tests by recording your own interactions on a device so you don't have to actually write any test code [9]

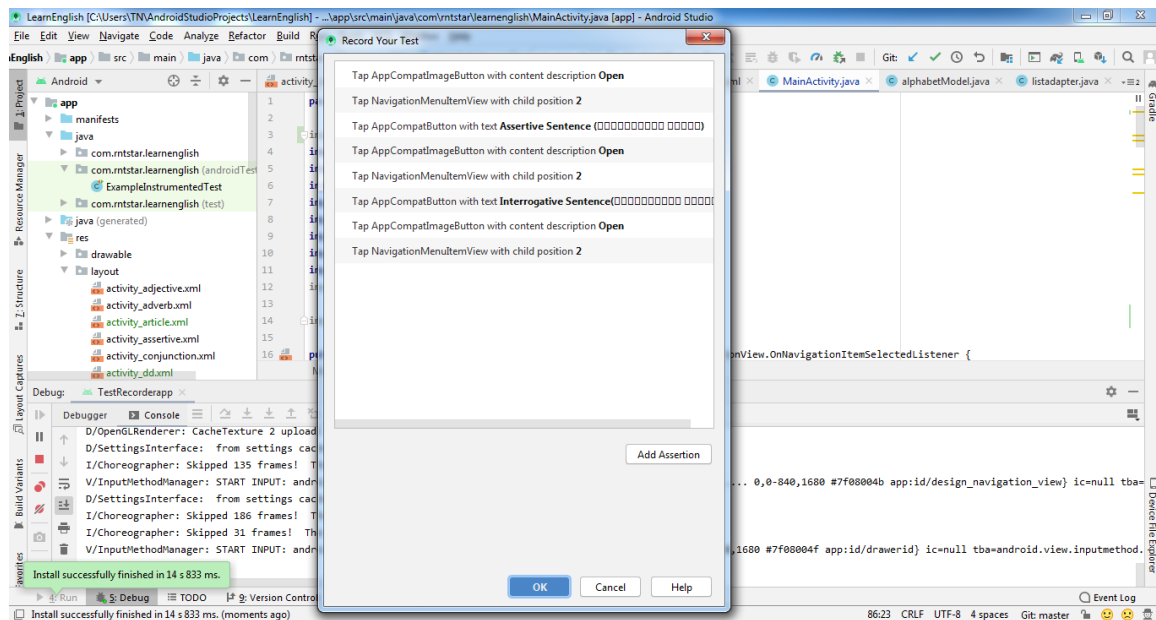


Figure 4.14: Main Activity Test

Part of speech activity junit4 test bug finding methods that are find bug and report.

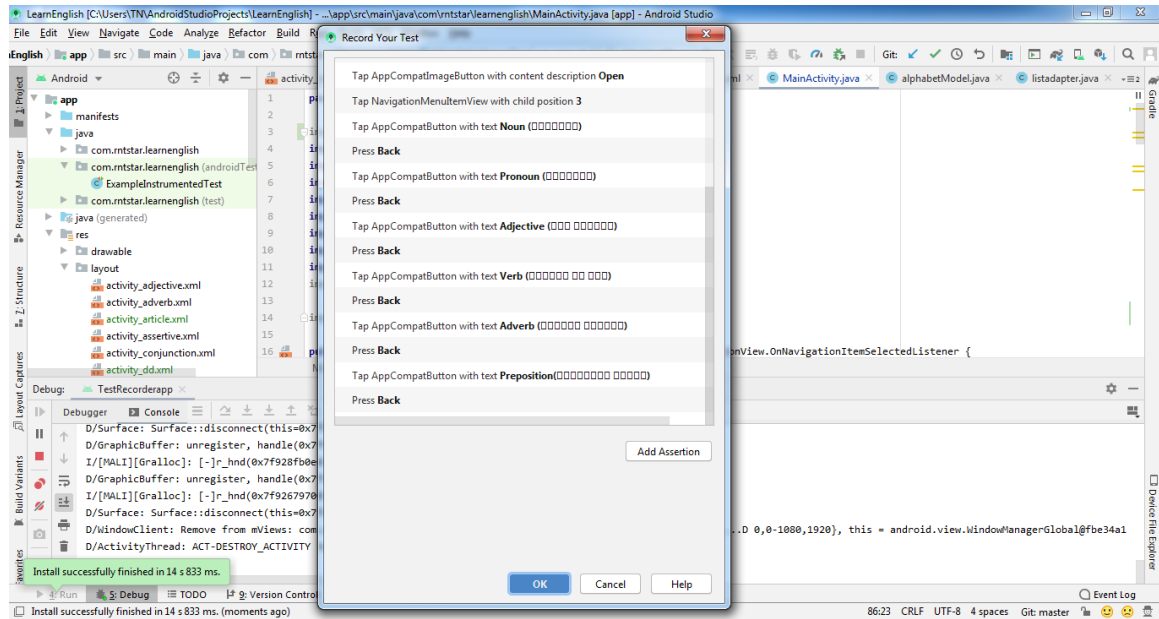


Figure 4.15: Part Of Speech Activity Test

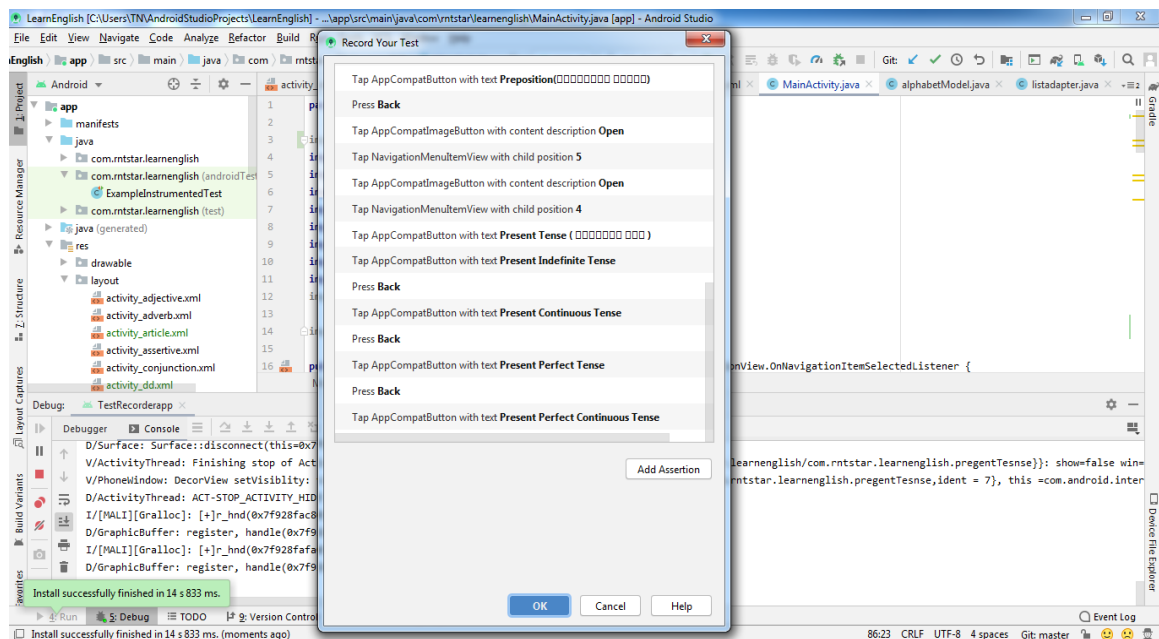


Figure 4.16: Tense Activity Test

CHAPTER 6

Conclusion and future scope

6.1 Discussion and conclusion

Learning English is not difficult. We can easily learn English if we want to. The use of the English language is increasing day by day. If you want to learn something from the internet, you will get a lot of content in English from your language. You will benefit if you learn English. We think that our application will help you a little in the field of English language learning. When you use our app, you will finish one lesson and go to another. Don't forget to let us know your opinion. Based on your feedback, our application will be beautifully developed.

6.2 Scope for Future Developments

We have huge thoughts about the app in the future. In the next version we will add quiz Vocabulary game Chabot and new some feature. We always think of the user. We will analyze the user feedback and add new features to our application.

REFERENCES

- [1] Screen compatibility overview, available at<<https://developer.android.com/guide/practices/screens_support>> Access on 6 June' 2020 at 12:17 PM
- [2] Comparative Study, available at<< <http://www2.uiah.fi/projekti/metodi/172.htm>>> Access on 6 June' 2020 at 12:20 PM
- [3] Requirement Gathering & Analysis Phase in SDLC, available at<< <https://study.com/academy/lesson/requirement-gathering-analysis-phase-in-sdlc.html>>> Access on 6 June' 2020 at 12:25 PM
- [4] Use Case Diagrams, available at <<https://www.tutorialspoint.com/uml/uml_use_case_diagram.html>> Access on 6 June' 2020 at 03:14 AM
- [5] Interaction Design and UX, available at << <https://www.interaction-design.org/literature/article/what-is-interaction-design> |>> Access on 6 June' 2020 at 12:35 PM
- [6] Android Tutorial, available at <<<https://www.javatpoint.com/android-sqlite-tutorial>>> Access on 6 June' 2020 at 12:40 PM
- [7] Implementation of Interactions, available at << https://en.wikipedia.org/wiki/Interaction_technique>> Access on 6 June' 2020 at 12:47 PM
- [8] Implementation Testing, available at << <https://www.softwaretestinghelp.com/types-of-software-testing>>> Access on 6 June' 2020 at 12:49 PM
- [9] Espresso Test Recorder, available at << <https://developer.android.com/studio/test/espresso-test-recorder>>> Access on 6 June' 2020 at 12:52 PM

APPENDICES

Appendix A:

```
package com.rntstar.learnenglish;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MenuItem;
import android.widget.Toast;

import java.util.Scanner;

public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
    DrawerLayout drawerLayout;
    ActionBarDrawerToggle actionBarDrawerToggle;
    NavigationView navigationView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        drawerLayout = (DrawerLayout) findViewById(R.id.drawerid);

        navigationView = (NavigationView) findViewById(R.id.navigationid);
        navigationView.setNavigationItemSelectedListener(this);

        actionBarDrawerToggle = new ActionBarDrawerToggle(this, drawerLayout,
R.string.open, R.string.close);
        drawerLayout.addDrawerListener(actionBarDrawerToggle);
        actionBarDrawerToggle.syncState();
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        if (savedInstanceState == null) {

getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new
alphabetfrag()).commit();
            navigationView.setCheckedItem(R.id.alphabetid);
        }
    }
}
```

```
}
```

```
@Override  
public void onBackPressed() {  
    if (drawerLayout.isDrawerOpen(GravityCompat.START)){  
        drawerLayout.closeDrawer(GravityCompat.START);  
    }else {  
        super.onBackPressed();  
    }  
}
```

```
}
```

```
@Override  
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
    if (actionBarDrawerToggle.onOptionsItemSelected(item)){  
        return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

```
@Override  
public boolean onOptionsItemSelected(@NonNull MenuItem menuItem) {  
    switch (menuItem.getItemId()){  
        case R.id.alphabetid:
```

```
getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new  
alphabetfrag()).commit();  
        navigationView.setCheckedItem(R.id.alphabetid);  
        break;
```

```
        case R.id.sentenceid:
```

```
getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new  
sentence()).commit();  
        break;  
        case R.id.partofspeechid:
```

```
getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new  
partofspeech()).commit();  
        break;  
        case R.id.Tenseid:
```

```
getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new
```

```

Tense()).commit();
        break;
        case R.id.vocGame:

getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new
dd()).commit();
        break;
        case R.id.ArticleID:

getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new
article()).commit();
        break;
        case R.id.picture_dictionaryID:

getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new
picturedic()).commit();
        break;
    }
    drawerLayout.closeDrawer(GravityCompat.START);
    return true;
}
}

```

Alphabet Model Code:

```

package com.rntstar.learnenglish;

public class alphabetModel {

    private String sNo;
    private String capital;
    private String small;
    private String phonic;
    private String name;

    public alphabetModel(String sNo, String capital, String small, String phonic, String
name) {

        this.sNo = sNo;
        this.capital = capital;
        this.small = small;
        this.phonic = phonic;
    }
}

```

```

this.name = name;
    }
    public String getsNo() {

        return sNo;
    }

    public void setsNo(String sNo) {
        this.sNo = sNo;
    }
    public String getCapital() {
        return capital;
    }

    public void setCapital(String capital) {
        this.capital = capital;
    }

    public String getSmall() {
        return small;
    }

    public void setSmall(String small) {
        this.small = small;
    }
    public String getPhonic() {

        return phonic;
    }

    public void setPhonic(String phonic) {

        this.phonic = phonic;
    }

    public String getName() {

        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}

```

Learn English App

ORIGINALITY REPORT

28%	%	%	28%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	9%
2	Submitted to University of East London Student Paper	4%
3	Submitted to Varsity College Student Paper	2%
4	Submitted to University of Northampton Student Paper	2%
5	Submitted to University of Plymouth Student Paper	2%
6	Submitted to University College London Student Paper	1%
7	Submitted to Middle East College of Information Technology Student Paper	1%
8	Submitted to Colorado Technical University Online Student Paper	1%

9	Submitted to University of Bradford Student Paper	1 %
10	Submitted to Nottingham Trent University Student Paper	1 %
11	Submitted to University of Wollongong Student Paper	1 %
12	Submitted to University of Ulster Student Paper	1 %
13	Submitted to City of Glasgow College Student Paper	1 %
14	Submitted to Aston University Student Paper	<1 %
15	Submitted to Southeast Community College Student Paper	<1 %
16	Submitted to Institute of Development Management Student Paper	<1 %
17	Submitted to Central Queensland University Student Paper	<1 %
18	Submitted to University of Westminster Student Paper	<1 %
19	Submitted to Higher College of Technology Student Paper	<1 %

20	Submitted to Arab Open University Student Paper	<1 %
----	--	------

21	Submitted to Nanyang Technological University Student Paper	<1 %
----	--	------

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		