**Voice**

**A Mini-Project Report**

**Under**

**Project Workshop**

***Submitted by***

**SNEHA PAREKH (E016)**

**SALLY PETER (E020)**

**PARUL PUROHIT (E022)**

***Under The Guidance Of***

**MUDIT KAPOOR**

***In partial fulfillment for the award of the degree***

***Of***

**B.TECH**

**IN**

**COMPUTER SCIENCE**

**At**

**MPSTME, NMIMS, MUMBAI**

**APRIL, 2014**

**CERTIFICATE**

This is to certify that the project entitled “Voice” is the bonafide work carried out by

Sneha Parekh (E016)

Sally Peter (E020)

Parul Purohit (E022)

of B.tech (Computer Engineering), MPSTME (NMIMS), Mumbai, during the VI semester of the academic year 2013-14, in partial fulfillment of the requirements for the award of the Degree of Bachelors of Technology as per the norms prescribed by NMIMS. The mini-project work has been assessed and found to be satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MUDIT KAPOOR

Internal Mentor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner 1 Examiner 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dean

Dr. S. Y. Mhaiskar

**DECLARATION**

We,

Sneha Parekh (E016)

Sally Peter (E020)

Parul Purohit (E022)

B.tech (Computer Engineering), VI semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Student:

Name:   
Sneha Parekh (E016)

Sally Peter (E020)

Parul Purohit (E022)

Place: MUMBAI

Date: 26/4/14

**Acknowledgement**

We would like to express our gratitude and appreciation to all those who gave us the opportunity to accomplish this report. A special thanks to our project mentor, Mr. Mudit Kapoor, whose help, constant supervision, stimulating suggestions and encouragement, helped us to coordinate our project and achieve our targets.

We would also like to acknowledge with much appreciation the crucial role of our Faculty Head, Mrs. Supriya Agarwal, who helped in building our basic fundamentals about Android and kept a check on our work progress.

We are also obliged to our classmates and friends who helped us in solving errors, suggesting new ideas and giving constructive feedback.

Lastly, we would like to thank our Head Of Department, Dr. Dhirendra Mishra and our respectable Dean, Dr. S. Y. Mhaiskar, for giving us the wonderful opportunity to work on a project like this.

**Abstract**

The main purpose behind this app is to create a user friendly portal to communicate with the dumb and deaf. It focuses on creating cutting-edge, easy-to-use sign language learning tools, with a pictorial representation provided for each sign to help the user learn better.Our main goal is to help empower the handicapped people and bridge the discrimination among people in the society.

The logic and functionality of this application is written in java using the android sdk for eclipse. Our android app named “Voice”, helps a user learn basic sign language for mastering English alphabets, numbers and colours. After having learnt all this, the user can play a game and entertain himself/herself. Later on, the user can also take a quiz to test his knowledge.  
And finally, that image-data dictionary module helps the user to surf through a variety of words and learn the corresponding sign language for the same.The system is developed to allow translation from sign language to English and vice versa. Its application can be extended for all other languages. It can allow customized personal interaction between the user and the system. The system can be updated to include messenger and interaction forums to encourage personal tutoring. Interactive games will encourage users of all age-group to participate. To conclude, Voice is a very basic application and can be used by anybody to learn sign language independently.

**Organization of report**

A report makes easier to understand the project. Our project on android application named Voice is an application for security which consists of 4 modules.

The first module teaches basic sign language lessons. It is futher categorized into:

* Alphabets
* Numbers
* Colors

The second module is on game. This module is the most attractive and fun part of the whole application. Games module is in the form of a picture quiz that anyone can play after going through all the sub parts of the lessons.Games is a way of testing one’s knowledge of sign language but in a light hearted way. The inclusion of pictures as options makes it all the more amusing and fun learning experience.

The third module is on Quiz. After the user has accessed the lessons and games component of the system, he/she is ready to attempt the Quiz module to test his/her knowledge. The Quiz module is a set of Questions testing the user on the chapters covered previously. It is a multiple choice question series that generates a result in the end.

The fourth and final module is a Dictionary. The dictionary consists of two major classifications- Pictures and Videos.The pictures section consists of dictionary chapters such as family, action, food and a complete alphabet dictionary that states alphabet-word-picture. This photo section displays the image along with the pop up message displaying the text. The videos function makes it easier to memorize the signs.

**Table of contents**

|  |  |  |
| --- | --- | --- |
| **SR.NO.** | **Topic** | **PG.NO.** |
| 1 | List of Figures | 8 |
| 2 | List of Tables | 20 |
| 3 | Module 1: Lessons | 21 |
| 4 | Module 2: Game | 24 |
| 5 | Module 3: Quiz | 25 |
| 6 | Module 4: Dictionary | 27 |
| 7 | Conclusion | 29 |
| 8 | Future Scope | 30 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig.No.** | **Topic** | **PG.NO.** |
| 2.1 | Icon | 8 |
| 2.2 | Homepage | 8 |
| 2.3 | Menu | 10 |
| 2.4 | Lesson submenu | 10 |
| 2.5 | Number 1 | 11 |
| 2.6 | Number 7 | 12 |
| 2.7 | Alphabet scrollview 1 | 14 |
| 2.8 | Alphabet scrollview 1 | 14 |
| 2.9 | Alphabet A | 16 |
| 2.10 | Colors view1 | 18 |
| 2.11 | Colors view2 | 18 |
| 2.12 | Colors black | 18 |
| 2.13 | Colors yellow | 18 |
| 2.14 | Game 1 | 21 |
| 2.15 | Game 2 | 21 |
| 2.16 | Ouiz-Question 1 | 24 |
| 2.17 | Quiz-Question 2 | 24 |
| 2.18 | Score | 25 |
| 2.20 | Dictionary Main menu | 33 |
| 2.21 | Dictionary submenu | 33 |
| 2.22 | Complete dictionary 1 | 33 |
| 2.23 | Complete dictionary 2 | 33 |
| 2.24 | Item | 34 |
| 2.25 | Item | 34 |
| 2.26 | Video | 34 |

**Abbreviations**

**.apk file -**  Android application package file or apk file , is a single file containing the source code, the  resources used by your application and the manifest file.

**Dalvik -** Android’s own virtual machine (VM) that executes files in .dex format. The android programs written in java are  compiled by the Java compiler and converted into .dex format using the “dx” tool that comes with dalvik.

**.dex file -**  Dalvik Executable file or dex file , contains  the executable  instructions in a format optimized for efficient storage and execution by the Dalvik VM

**adb -**  Android Debug bridge, a command line debugging utility that comes with the Android SDK ( Software Development Kit)

**DDMS** -  Dalvik debug Monitor Service, a GUI tool for debugging the android app. Leverages the “adb” to connect to the emulator or device.

**JDK** - "Java Development Kit" an SDK for the java platform. It is needed to run the Android SDK.  
  
**JRE**- "Java Runtime Environment" a collection of binarys and files to allow java software to execute.

**OS** - Operating system, I.E. Windows Vista, LINUX or MAC or Android

**SQLite** - An embedded relational database management system contained in a relatively small (~275 kB) C programming library. It is multitasking concerning reads. Writes can be done only one-at-a-time. It is a popular choice for local/client storage on web browsers. It has many bindings to programming languages. It is arguably the most widely used database engine, as it is used today by several widespread browsers, operating systems, embedded systems among others

**MTP** - Short for Media Transfer Protocol. Allows file transfers between the Android device and a PC. Safer to use than USB Mass Storage, as it does not seize complete control of blocks on the Android Device; instead, it connects in a transactional (all-or-nothing transfer) method.

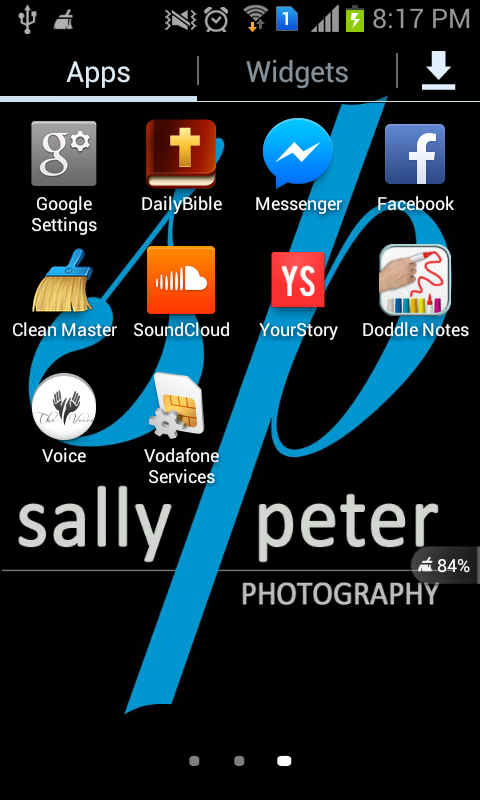
**Module 1: Lessons**

After installation the app icon looks like as shown in Fig 2.1.

The homepage is as shown in Fig. 2.2

Main Menu

Figure 2.1 Figure 2.2

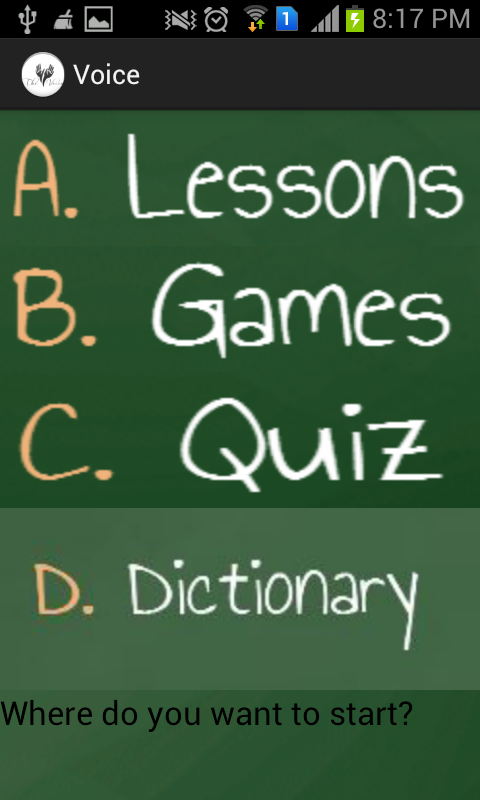
The system flow-chart diagram is as follows:

The first module helps to user to learn basic sign language translation for the following

* Numbers
* Alphabets
* Colors

On tapping on “Lessons” option in the main menu (Fig. 2.3), a sub-menu containing the above three options open as represented by the figure 2.4

Figure 2.3



**Numbers**

The Numbers aid the user in learning sign language of the digits 1-10. The user can use his two hands simultaneously to represent any digit between 11 to 99. Hence learning sign language of only 2 digits helps the user to represent any digit between 1-100.

A “viewFlipper” function has been used: A [ViewFlipper](http://developer.android.com/reference/android/widget/ViewFlipper.html)is a simple [ViewAnimator](http://developer.android.com/reference/android/widget/ViewAnimator.html) that will animate between two or more views that have been added to it. Only one child is shown at a time. If requested, can automatically flip between each child at a regular interval. A [ViewFlipper](http://developer.android.com/reference/android/widget/ViewFlipper.html)  can be used to slide views in and out of the user’s current view port .These Views slides with given appropriate Animation.

The flipper function is made to detect human touch using the onClickListener command. With each touch/tap by humans anywhere on the screen the next digit is displayed as is represented by transitions from Fig. 2.5 to Fig. 2.6

Menu > Lessons > Numbers

Figure 2.4 Figure 2.5

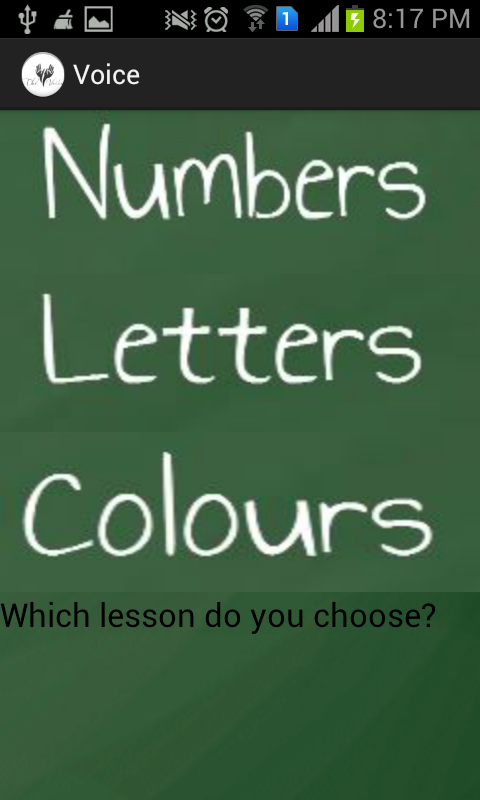
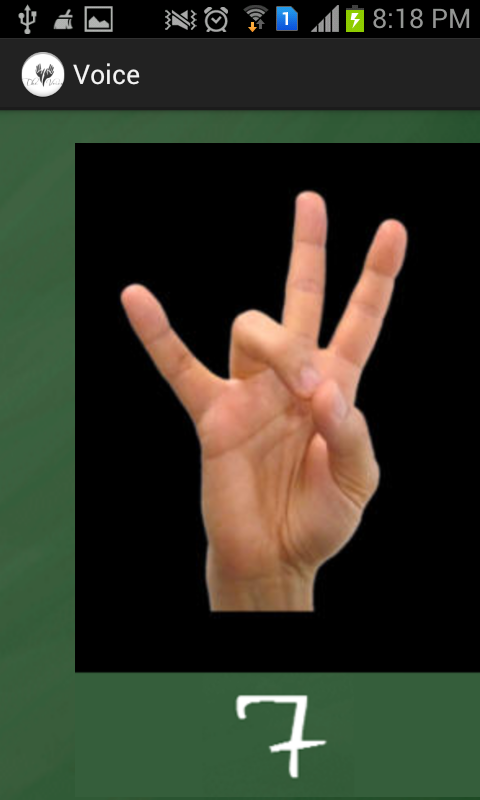
 

Figure 2.6



**Alphabets**

Tapping on the alphabet button on the lessons menu, displays a submenu (Fig. 2.7, Fig. 2.8), containing a scroll-list of buttons all the English alphabets from A-Z.

Any button can be clicked to display the sign-language translation for the corresponding alphabet. (Fig 2.9, Fig.2.10)

Back button of the phone can be clicked to go back to the sub menu containing all the alphabet buttons.

Back button when clicked again takes the user back to the “Lesson” menu.

26 button (from A-Z) have been created an a unique id has been associated to each one of them.

A switch case has been made use of to call the corresponding alphabet by its id whenever its button is tapped on.

findViewById function has been extensively made use of: A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components (buttons, text fields, etc.). The [ViewGroup](http://developer.android.com/reference/android/view/ViewGroup.html) subclass is the base class for layouts, which are invisible containers that hold other Views (or other ViewGroups) and define their layout properties.

setContentView function has also been made use of:

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window in which user can place the UI with [setContentView(View)](http://developer.android.com/reference/android/app/Activity.html" \l "setContentView(android.view.View)).

[onCreate(Bundle)](http://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) is where activities are initialized.Most importantly, here  [setContentView(int)](http://developer.android.com/reference/android/app/Activity.html" \l "setContentView(int)) is called, with a layout resource defining the UI, and using [findViewById(int)](http://developer.android.com/reference/android/app/Activity.html" \l "findViewById(int)) to retrieve the widgets in that UI that one needs to interact with programmatically.

Menu > Lessons > Letters

Figure 2.7 Figure 2.8

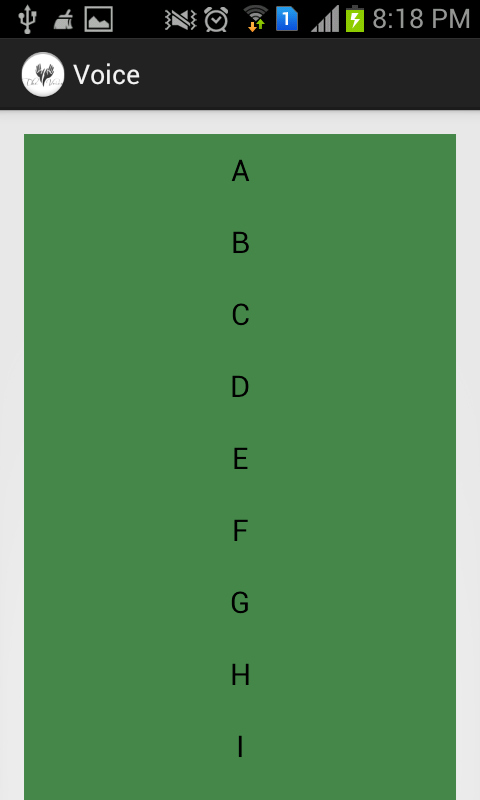
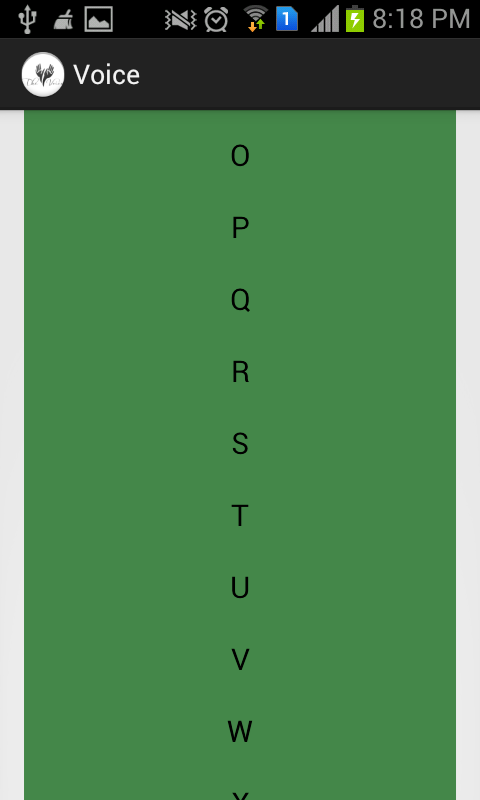
 

Figure 2.9 Figure 2.10

**Colors**

Tapping on the colours button on the lessons menu, displays a submenu (Fig. 2.11), containing a scroll-list of buttons containing a list of 12 colours.

Any button can be clicked to display the sign-language translation for the corresponding colour. (Fig 2.12, Fig.2.13)

Back button of the phone can be clicked to go back to the sub menu containing all the Colour List.

Back button when clicked again takes the user back to the “Lesson” menu.

12 button (for 12 colours) have been created and a unique id has been associated to each one of them.

A switch case has been made use of to call the corresponding colour by its id whenever its button is tapped on.

findViewById function has been extensively made use of: A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components (buttons, text fields, etc.). The [ViewGroup](http://developer.android.com/reference/android/view/ViewGroup.html) subclass is the base class for layouts, which are invisible containers that hold other Views (or other ViewGroups) and define their layout properties.

setContentView function has also been made use of:

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window in which user can place the UI with [setContentView(View)](http://developer.android.com/reference/android/app/Activity.html" \l "setContentView(android.view.View)).

[onCreate(Bundle)](http://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) is where activities are initialized.Most importantly, here  [setContentView(int)](http://developer.android.com/reference/android/app/Activity.html" \l "setContentView(int)) is called, with a layout resource defining the UI, and using [findViewById(int)](http://developer.android.com/reference/android/app/Activity.html" \l "findViewById(int)) to retrieve the widgets in that UI that one needs to interact with programmatically.

Menu > Lessons > Colors

Figure 2.11

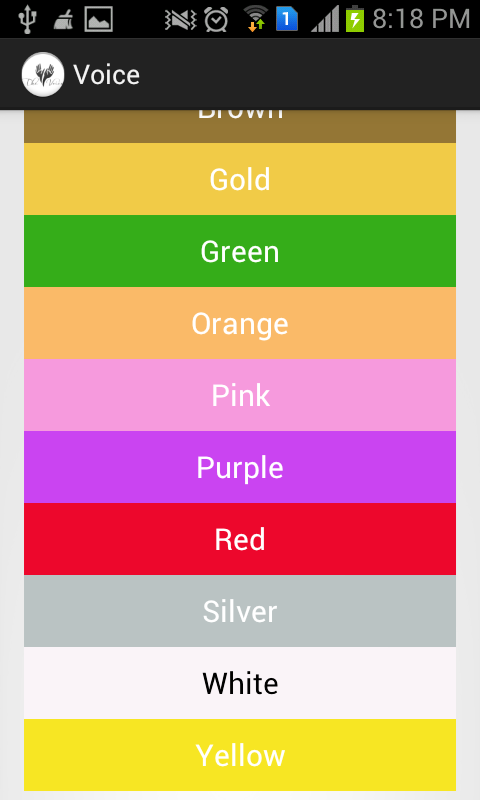


Figure 2.12 Figure 2.13

**Module 2: Games**

This module is the most attractive and fun part of the whole application. Games module is in the form of a picture quiz that anyone can play after going through all the sub parts of the lessons.  
Games is a way of testing one’s knowledge of sign language but in a light hearted way.

The inclusion of pictures as options makes it all the more amusing and fun learning experience.

The module starts with a question in the form of a word or a letter that is displayed, below which there are four pictures in the form of a square. The user has to tap on the right answer. A corresponding pop-up message will display as soon as the user taps on any button to tell whether the user has pressed the right option or not.

This module of games is different from the module of quiz in many ways.

* Firstly, the quiz is in the form of question and answers whereas games is a picture quiz
* Quiz displays the score of the user and rates him at the end, whereas game does not.
* Quiz will not specify which of the answers were correct or incorrect but in the game if on any particular question user taps on a wrong answer, a pop will show telling him its wrong, same as in the case of a correct one.

There are 5 questions that are included in all in the games module covering most of the categories of sign language out which 2 are based on knowledge of letters, one each are based on actions, Food and Family words.

Each question can be answered multiple no. of times. Each question has options in the form of four images, and there is a button for next question.

There are many ways in which Games module can be improvised for future use-

* We can add more questions and let the questions appear randomly.
* We can convert the current module from single-player format to multi-player format so that it becomes more attractive and competitive.
* We can even add levels of difficulties and sort questions according to them so that the user can play at a level at which he/she is comfortable

The following elements have been used in the functioning of the game module-

Grid Layout- A layout that places its children in a rectangular *grid*.

The grid is composed of a set of infinitely thin lines that separate the viewing area into *cells*. Throughout the API, grid lines are referenced by grid *indices*. A grid with N columns has N + 1 grid indices that run from 0 through N inclusive. Regardless of how GridLayout is configured, grid index 0 is fixed to the leading edge of the container and grid index N is fixed to its trailing edge (after padding is taken into account).

Toast Widget- A toast is a view containing a quick little message for the user. The toast class helps you create and show those.

When the view is shown to the user, appears as a floating view over the application. It will never receive focus. The user will probably be in the middle of typing something else. The idea is to be as unobtrusive as possible, while still showing the user the information you want them to see. Two examples are the volume control, and the brief message saying that your settings have been saved.

The easiest way to use this class is to call one of the static methods that constructs everything you need and returns a new Toast object.

Figure 2.14

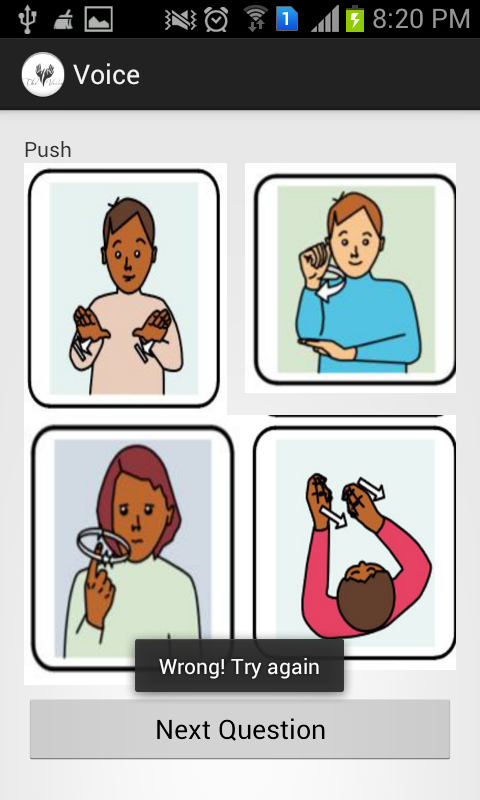
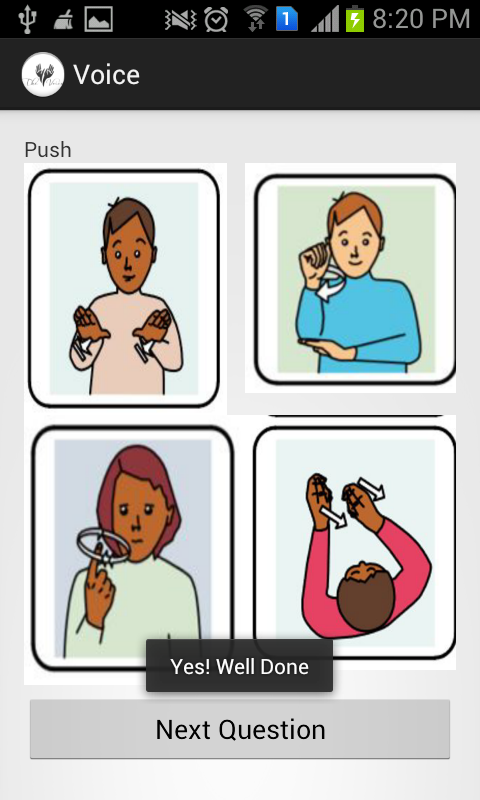


Figure 2.15



**Module 3: Quiz**

After the user has accessed the lessons and games component of the system, he/she is ready to attempt the Quiz module to test his/her knowledge. The Quiz module is a set of Questions testing the user on the chapters covered previously. It is a multiple choice question series that generates a result in the end.

Each question has three options to choose from and an image oriented question as shown in Figure 2.16 and Figure 2.17. The answer generated is displayed using star ratings and a pop up of the marks scored. It also displays a commentary message according to the marks scored by the user as shown in Figure 2.18.

A user can attempt the test any number of times to improve his/her score. Although a very basic implementation, it can be improvised in future in the following ways:

* Storing more questions in the database
* Random generation of questions from the database
* Creation of levels to qualify
* Storing user scores to create report card

To create, add data, modify and display data from the database, the following packages and classes have been used as shown in Figure 3.1

Package android.database.sqlite contains the SQLite database management classes that an application would use to manage its own private database. Applications use these classes to manage private databases.

Package android.database Contains classes to explore data returned through a content provider.

To manage data in a private database, we use the [android.database.sqlite](http://developer.android.com/reference/android/database/sqlite/package-summary.html) classes. These classes are used to manage the [Cursor](http://developer.android.com/reference/android/database/Cursor.html) object returned from a content provider query. Databases are usually created and opened with [openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory)](http://developer.android.com/reference/android/content/Context.html" \l "openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))

Public abstract class SQLiteOpenHelper makes it easy for [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) implementations to defer opening and upgrading the database until first use, to avoid blocking application startup with long-running database upgrades.

Android.database.cursor interface provides random read-write access to the result set returned by a database query. Cursor implementations are not required to be synchronized so code using a Cursor from multiple threads should perform its own synchronization when using the Cursor. Implementations should subclass [AbstractCursor](http://developer.android.com/reference/android/database/AbstractCursor.html)

Radio buttons widget is used to display the options for the questions. A radio button is a two-states button that can be either checked or unchecked. When the radio button is unchecked, the user can press or click it to check it. However, contrary to a [CheckBox](http://developer.android.com/reference/android/widget/CheckBox.html), a radio button cannot be unchecked by the user once checked. Radio buttons are normally used together in a [RadioGroup](http://developer.android.com/reference/android/widget/RadioGroup.html). When several radio buttons live inside a radio group, checking one radio button unchecks all the others.

Ratingstar widget is used to display the quiz results. A RatingBar is an extension of SeekBar and ProgressBar that shows a rating in stars. The user can touch/drag or use arrow keys to set the rating when using the default size RatingBar. The number of stars set (via [setNumStars(int)](http://developer.android.com/reference/android/widget/RatingBar.html" \l "setNumStars(int)) or in an XML layout) will be shown when the layout width is set to wrap content (if another layout width is set, the results may be unpredictable). Menu > Quiz

Figure 2.16 Figure 2.17

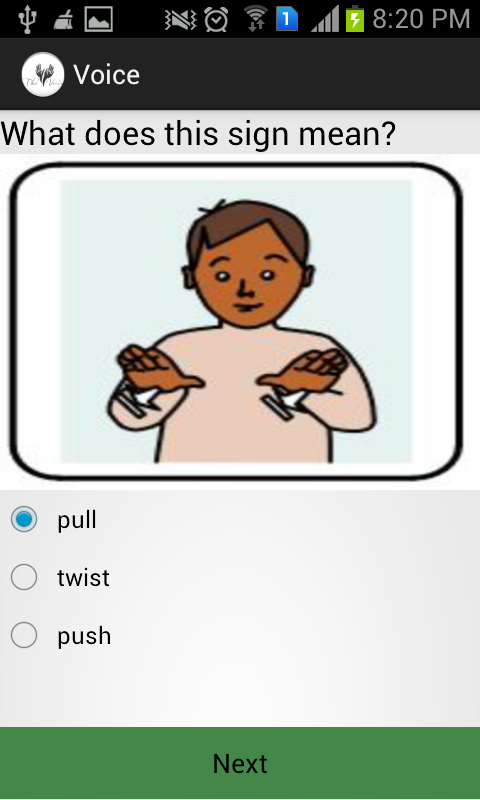
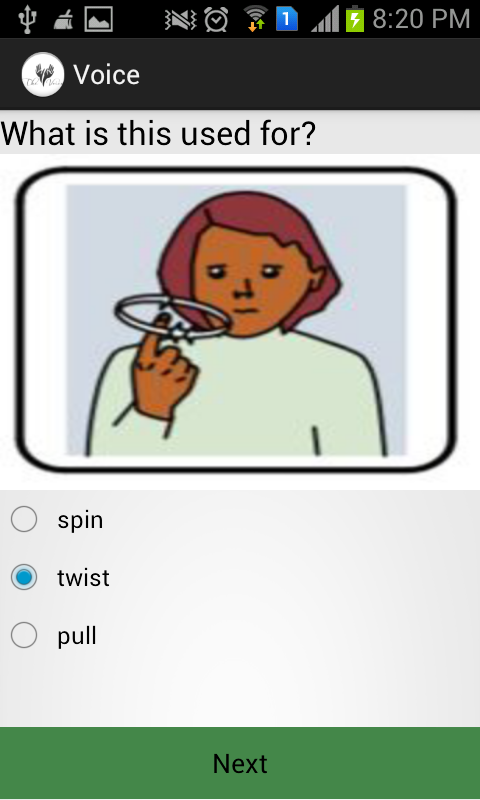
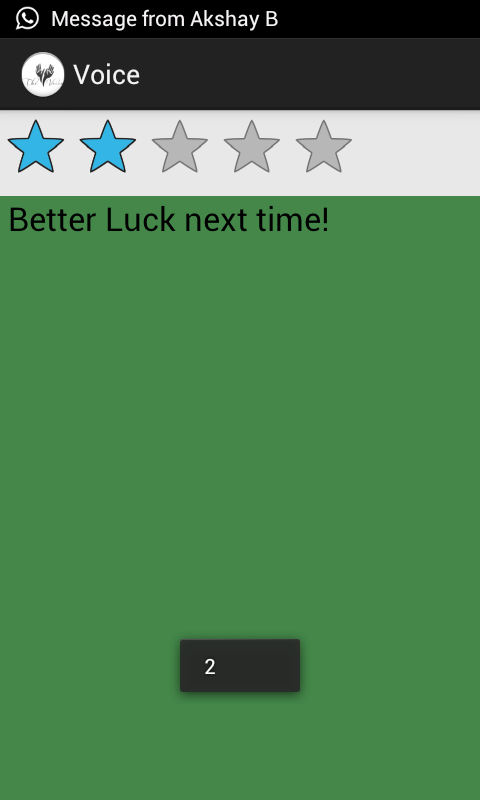
 

Figure 2.18



**Module 4: Dictionary**

A dictionary is collection of words in one or more specific languages, often listed alphabetically, with usage information, definitions etc. A dictionary is a very vital part of the sign language application as it provides the basic knowledge a person requires to communicate. The dictionary consists of two major classifications- Pictures and Videos as shown in Figure 2.18. The pictures section consists of dictionary chapters such as family, action, food and a complete alphabet dictionary that states alphabet-word-picture as shown in Figure 2.19-2.25. This photo section displays the image along with the pop up message displaying the text.

The videos section serves the following purposes

* For the disabled who are trying to learn pronunciation of words
* For use of audio and video feature to encourage children to learn
* Song videos to make it easier to memorize the signs

The videos when clicked start playing the video with media controller features as shown in Figure 2.26

Public interface expandableListAdapter is used to display the drop down menu list for dictionary. It is an adapter that links a [ExpandableListView](http://developer.android.com/reference/android/widget/ExpandableListView.html) with the underlying data. The implementation of this interface will provide access to the data of the children (categorized by groups), and also instantiate [View](http://developer.android.com/reference/android/view/View.html)s for children and groups.

Public abstract [Object](http://developer.android.com/reference/java/lang/Object.html) getChild (int groupPosition, int childPosition) gets the data associated with the given child within the given group.

Parameters:

|  |  |
| --- | --- |
| groupPosition | the position of the group that contains the child |
| childPosition | the position of the child (for which the View is returned) within the group |

A dialog is a small window that prompts the user to make a decision or enter additional information. A dialog does not fill the screen and is normally used for modal events that require users to take an action before they can proceed. This is used to display the dictionary image on clicking of an item in the list menu.

Public class VideoView displays a video file. The VideoView class can load images from various sources (such as resources or content providers), takes care of computing its measurement from the video so that it can be used in any layout manager, and provides various display options such as scaling and tinting. This is used to display the three videos in video dictionary.

Menu > Dictionary > Pictures

Figure 2.18 Figure 2.19

Figure 2.20 Figure 2.21

Figure 2.22 Figure 2.23

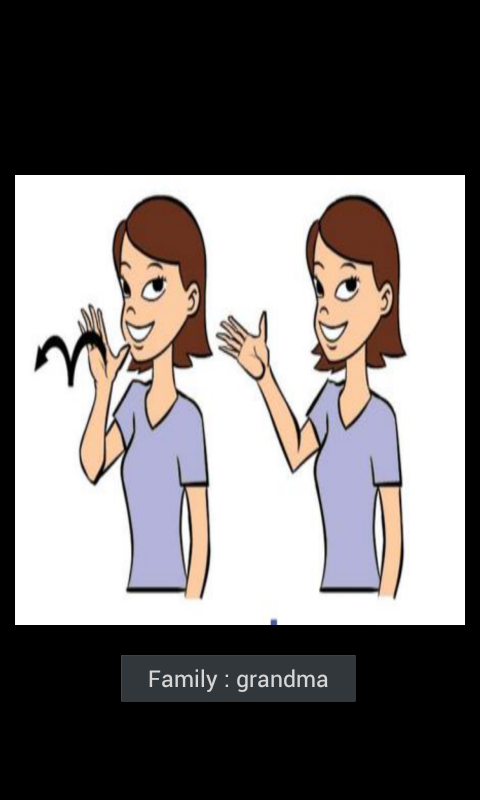
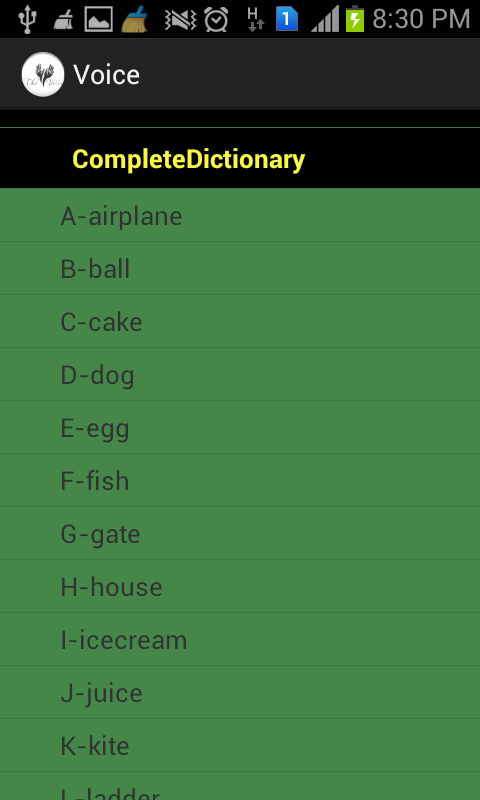
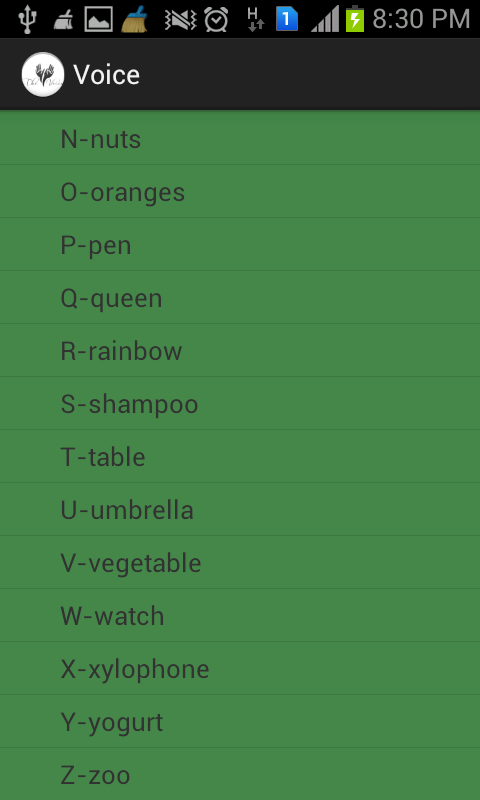
 

Figure 2.24 Figure 2.25

Menu > Dictionary > Videos

Figure 2.26

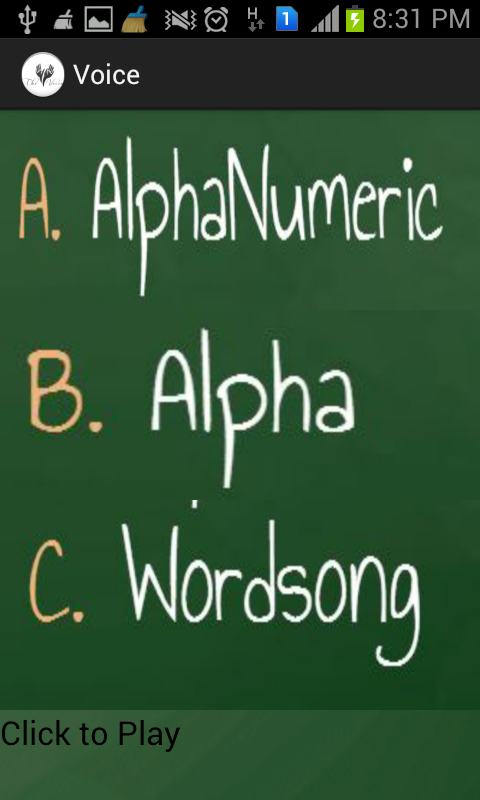
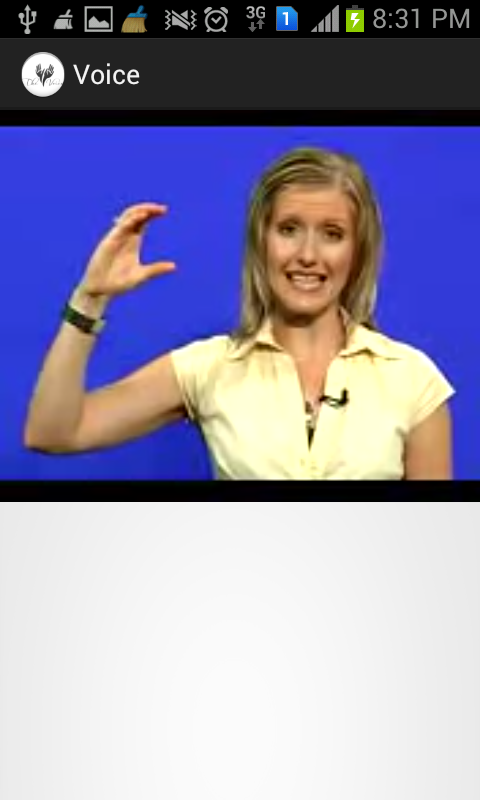


Figure 2.27



**Conclusion**

**Android Development** is the best area of smart-phone technologies which encourages android application developer and designers to provide customers precisely what they search for under their common genus associated with curiosity.

Owing to the popularity of Android, Mobile Apps development industries are considering Android Application Development as one of the best remunerative business opportunities. The need to hire knowledgeable mobile application developer is intense.

Android Programming language is very easy to learn and app development is cost effective this put Android into extraordinary use. Developer can develop Apps in different ways as per their need and wish.

Many mobile Apps development industries are considering **Android Application Development** as one of the best business opportunities, for this they need to hire a lot of knowledgeable **mobile application developer** in future. This adds a big sign of scope of mobile Apps in future.

In the current job market of mobile application development, the need for inventive App developers is huge and still increasing.

With the guidance of our teachers and mentors, we were able to create a basic sign language application that helps children and teachers to learn general and widely used words.  
 We included many modules in the applications such as lessons, quiz, games and Dictionary

Lessons module has all the tutorials related to letters numbers and colours.

Quiz is a question-answer type module that even displays scores in the end

Games module is a fun picture quiz that is mainly meant for children.

Finally Dictionary is a complete reference to all general categories of words for sign language. It has sub parts as a picture dictionary and a video dictionary.

As it was our first android application we had a lot to learn. We learnt many new terms such as intent, we learnt about the structure of eclipse, what is the significance of each folder in any android application.

We learnt all the basic steps that are required in building any android application, such as downloading adt bundle and its packages, creating a new android application, how to use text boxes, buttons, and various other controls and defining their properties.

We learnt how to create an in-built database, how to use different widgets as well.

Thus, creating this application gave us a really significant insight towards sign language as a necessity in today’s world as well as how to use its significance and convert it into an application that will be very useful for android users for years to come.

**FUTURE SCOPE**

Mobile Application Development is the future of Software Development.

In the last one or two year the users of mobile phones has rapidly increased and counting is still on. India stands second in the number of active mobile phones in the world. Out of the six billion mobile phones in the world, around one billion is being used in India (70% of our current population of India). Around 6 million subscribers every month join.

People using smart-phones demands for better applications and update for existing one, which in turn huge scope of android mobile application development in India

Many mobile Apps development industries are considering Android Application Development as one of the best business opportunities, for this they need to hire a lot of knowledgeable mobile application developer in future. This adds a big sign of scope of mobile Apps in future.

When this is the future scope of android applications in India, and after seeing how on a large scale Android applications are making their mark among Indians then there is a need for applications that are specific to Indian regions and should be made especially for them.

Sign Language is a very important part of any deaf and dumb person and even for those who want to learn this language and communicate through it. Having an app for this language facilitates many individuals around the world. But when it comes to India, people may not prefer current sign language apps that are available.

There are many reasons for that. Firstly, all the sign language applications that are available are mainly based on American or british sign language, or as a matter of fact, any other foreign language, making it difficult for any Indian to learn and teach using such apps, especially in rural areas where regional languages are spoken.

Therefore there is a rising need for an INDIAN SIGN LANGUAGE APPLICATION in future that would help Indians in all corners of not only in India but in the world to sign in Indian Languages, and no longer depend on British or American sign language that are otherwise difficult to understand by Indian kids.

But creating an Indian sign language has its own difficulties, because till date there is no proper standard for signing in Hindi or any other Indian language. There are Institutions that teach according to their own language, as a result people are more comfortable signing in their languages. The need is to reach out to such Institutions and nationalize their sign language standards using android applications so that Indians all around the world can make use of it.  
  
A website on Indian Sign Language would help greatly in achieving this goal, as developers can easily use resources such as pictures, videos, data etc.

As far as future of our app is concerned there can be many ways in which it can be improvised and made better looking at the current scenario of android applications in India.

The current system that we have developed allows translation from sign language to English and vice versa. And it includes both American and British sign language pictures. We can extend its application to all other languages, giving special preference to Indian languages as the main focus of our app was to make it for people living in India.

The system currently includes interaction between the user and the system through lessons, quiz, games and full picture and video dictionary. The system can be updated to include messenger and interaction forums to encourage personal tutoring.

We can even create modules of small conversation preferably in video format so that words that are most spoken are covered in form of group interaction as well.

In the current application we have included 3 general sign language videos, the system can be updated to include gif files of all parts of sign languages mainly those that include more than one hand movement and are difficult to understand by just pictures.

The application even aims towards providing downloadable materials offline so that the users can access the resources even if they do not have internet connection. This will help greatly in rural areas where even network connection is a huge problem in India.

Future of android is beyond imagination. It has opened a new stream of technological advancements. And interestingly, there seems no end to experimenting and evolving new applications and mobile phones using this innovative and dynamic technology. In years to come the usage of this technology is bound to increase because of its flexibility to develop applications of different kinds. There has been an unexpected rise in the demands of Android developers and in future this demand is expected to increase with time. With each year passing, we are expected to have more and more developers joining the pool of talent because as it is an open source platform, the scope to create something new is infinite. It totally depends on the imagination and skills of the developers that what they wish to create out of this amazing technology.

Customers are always eager to look for something different and new and android is the key to generate something unique all the time. If we have an idea, we can surely convert it into an interesting app using this diversified technology. The future of android is bright because it can be used on different avenues to create new things.