



CENG491

Initial Design

Report

HANDE

Kadir Eray Dođanlar

Dođuř Kcgde

Baha Tosun

Hamide Hande Keskiner

Index

1. Introduction.....	6
1.1. Problem Definition	6
1.2. Purpose.....	6
1.3. Scope	6
1.4. Overview.....	6
1.5. Definitions, Acronyms and Abbreviations.....	6
1.6. References.....	6
2. System Overview.....	6
2.1. Last Module.....	7
2.2. Flat Text Module.....	7
2.3. Phrases Module.....	7
2.4. Icons Module	7
2.5. Database Module	7
2.6. Text-to-Speech Module.....	8
2.7. Logger Module.....	8
2.8. Settings Module.....	8
3. Design Considerations.....	8
3.1. Design Assumptions, Dependencies and Constraints.....	8
3.1.1. Time Constraints.....	8
3.1.2. Resource Constraints.....	8
3.1.3. Performance Constraints.....	8
3.1.4. Software Constraints.....	8
3.1.5. Hardware Constraints.....	9
3.2. Design Goals.....	9
3.2.1. Reliability	9
3.2.2. Functionality	9
3.2.3. Usability.....	9

4. Data Design.....	9
4.1. Data Description.....	9
4.2. ER Design.....	10
4.3. Database Schemas.....	10
4.4. Class Diagrams.....	12
4.4.1. Model Package	12
4.5. Data Description.....	14
5. System Architecture	15
5.1. Architectural design	15
5.2. Description of Components.....	16
5.2.1. Last Module.....	16
5.2.1.1. Processing Narrative for Last Module	16
5.2.1.2 Interface Description for Last Module	16
5.2.1.3 Processing Detail for Last Module	17
5.2.1.4 Dynamic Behavior.....	17
5.2.2. Flat Text Module.....	17
5.2.2.1. Processing Narrative for Flat Text Module	17
5.2.2.2 Interface Description for Flat Text Module.....	17
5.2.2.3 Processing Detail for Flat Text Module.....	17
5.2.2.4 Dynamic Behavior.....	18
5.2.3. Phrases Module.....	18
5.2.3.1. Processing Narrative for Phrases Module	18
5.2.3.2 Interface Description for Phrases Module.....	19
5.2.3.3 Processing Detail for Phrases Module.....	19
5.2.3.4 Dynamic Behavior.....	19
5.2.4. Icons Module	20
5.2.4.1. Processing Narrative for Icons Module.....	20

5.2.4.2. Interface Description for Icons Module.....	20
5.2.4.3. Processing Detail for Icons Module	20
5.2.4.4 Dynamic Behavior.....	20
5.2.5. Database Module	21
5.2.5.1. Processing Narrative for Database Module	21
5.2.5.2. Interface Description for Database Module	21
5.2.5.3. Processing Detail for Database Module	21
5.2.5.4. Dynamic Behavior.....	21
5.2.6. Text-to-Speech Module.....	21
5.2.6.1. Processing Narrative for Text-to-Speech Module	21
5.2.6.2. Interface Description for Text-to-Speech Module	21
5.2.6.3. Processing Detail for Text-to-Speech Module	21
5.2.6.4. Dynamic Behavior.....	22
5.2.7. Logger Module	22
5.2.7.1. Processing Narrative for Logger Module.....	22
5.2.7.2. Interface Description for Logger Module.....	22
5.2.7.3. Processing Detail for Logger Module.....	22
5.2.7.4 Dynamic Behavior.....	23
5.2.8. Settings Module	23
5.2.8.1. Processing Narrative for Settings Module.....	23
5.2.8.2. Interface Description for Settings Module.....	23
5.2.8.3. Processing Detail for Settings Module.....	24
5.2.8.4 Dynamic Behavior.....	24
5.3 Design Rationale.....	24
6. User Interface Design	25
6.1. Overview of User Interface	25
6.1.1. Main Page.....	25

6.1.2. Flat Text Screen.....	26
6.1.3. Phrases Screen.....	26
6.1.3.1. Add Phrase.....	26
6.1.3.2. Edit Phrase.....	27
6.1.3.3. Delete Phrase.....	27
6.1.4. Icons Screen.....	28
6.1.4.1. Add Icon.....	28
6.1.4.2. Edit Icon.....	29
6.1.4.3. Delete Icon.....	29
6.1.5. Menu.....	30
7. Time Planning (Gantt Chart).....	31
7.1. Term 1 Gantt Chart.....	31
7.2. Term 2 Gantt Chart.....	32
8. Libraries and Tools.....	32
9. Conclusion	33

1. Introduction

1.1. Problem Definition

The problem that we attend to solve is to provide a chance to explain what they want for speech-impaired people by giving a device that can speak for him/her. The product is going to help any people who can use touch screen android devices.

1.2. Purpose

This report provides the necessary definitions to conceptualize and further formalize the design of the software, of which its requirements and functionalities were summarized in the previous requirements analysis report. The aim is to provide a guide to a design that could be easily implemented by any designer reading this report.

1.3. Scope

In this project, mainly we are going to implement a text-to-speech engine and we are going to provide an interface for this engine to be used easily. For platform we are going to use touch screen tablet computers with android operating system. There will be sounds in application's database. The thing wanted to be express which can be a flat text, a pre-saved phrase or a picture explaining a certain event is going to be convert a text format and that format is going to combine with our sound database.

1.4. Overview

This document includes initial design report for AndroidSpeaker. First, an overview of the problem and the product are described. Then system overview and design considerations are presented to the audience. After declaring the data design of this project, system architecture will be clarified explicitly. Then user interface design and the detailed design of the AndroidSpeaker are explained clearly. And finally, time planning of the project will be given by this document.

1.5. Definitions, Acronyms and Abbreviations

SRS	: Software Requirement System
AAC	: Augmentative Alternative Communication
OS	: Operating System
IOS	: iPhone Operating System
App	: Application
USA	: United State of America
TTS	: Text to speech

2. System Overview

This system runs according to the data which is processed version of the user input. In the end module the application will just convert the text to speech by the system TTS engine. In this part conversion process is made according to the pre-saved user settings. Until the end state the program can get information from three different channels. The end part and the channels are represented the chart below.

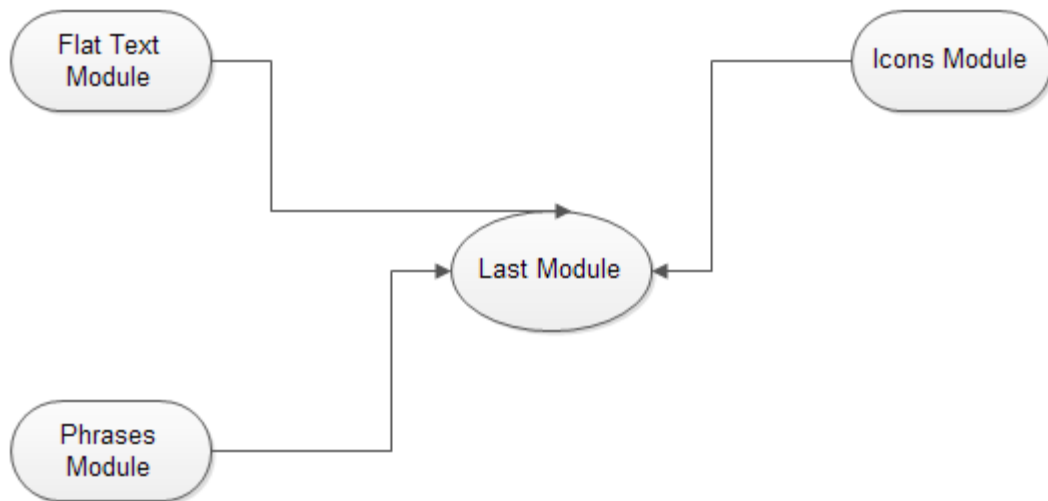


Figure -1-

2.1. Last Module

This module is a parent module in which data and the other information are kept in a certain order. In this state the processed last version of information from previous modules are collected and stored. This module packs them and sends to the text to speech engine module. Thus, we can say that the last module is the last module of getting text manner.

2.2. Flat Text Module

This module is a module provides to user a chance to type text. For this manner the system uses the keyboard of android and its features such as Swype manner. Main purpose of this part is to provide a module in which the end user can type what he/she wants easily and quickly as much as possible. Since it is just a flat text module there should not be any other thing that can distract the attention of the user. After text typing this module sends the data to last module.

2.3. Phrases Module

This module is a module provides to user a chance to select a pre-saved phrase instead of typing a frequently used phrase again and again. In this module the user can select one of the phrases from pre-saved phrases library, can combine two or more phrases or combine phrases with text, save new phrases or modify and update the old ones.

2.4. Icons Module

This module is to express the events which matches with the thing wanted to express. The aim is again the speed. For the times in which selecting a phrase can really be hard for the user, this module can be used.

2.5. Database Module

Main purpose of the database module is to save the syllables to convert to text to speech. Except for this purpose, it saves the images and its phrases with their information like usage count, saved date etc., phrases and their information and the user settings to make them permanent and not to take same setting every time.

2.6. Text-to-speech Module

This module runs after the last module and converts the processed version of data to text.

2.7. Logger

According to the coming data from modules, it saves the error message to the log file. The main purpose of this module is to provide technical data when an unexpected situation occurs.

2.8. Settings Module

This module is for getting and saving user settings. It provides an interface containing selection options to user and gets the user request and sends them to database module to save proper part. In the text-to-speech module while converting texts to speech these settings are taken by this module and provided to text-to-speech module.

3. Design Considerations

The system is designed with an Object-Oriented paradigm hence each module is presented by a class. Since each class represents a module, all the module classes are implemented by a Singleton Design Pattern.

3.1. Design Assumptions, Dependencies and Constraints

3.1.1. Time Constraints

Actually for this system there is no time constraint. The total system goes forward step by step by processing the modules. So the time depends on the user needs. However, since we are going to be implementing the system on Android Operating System we should obey some time constraints of Android. Not to bother the user Android pops up a window to user to terminate the application when that application runs a certain time and does not gives any activity result. This manner will be problematic for the system while last module is running the database module and getting the result of it. For these times we are going to use synchronized coding.

3.1.2. Resource Constraints

Some parts of the system will be implemented by using third party software; therefore, if proper could not be found the main system can fail. For this manner the device should have touch screen, camera and enough memory to store database values.

3.1.3. Performance Constraints

The main performance manner is depending on the acceptable real time performance. For this part the system's main approach will be improved after making real time tests.

3.1.4. Software Constraints

The system will be implementing by using Java Programming Language and compatible with the Android Operating System.

Except for this the system will be using SQLite for storing data. We will be using Android database connection manners to connect this database. For the syllables values we are going to get help form our sponsor Innova.

Logger module will be implementing by using logger interfaces of Java.

3.1.5. Hardware Constraints

Since our system will run on Android OS the user should have a tablet computer having Android OS.

3.2. Design Goals and Guidelines

3.2.1. Reliability

Since the system does not have complicated algorithms and the system runs according to settings of the user the reliability manner is not that much problem. Only issue is for reliability manner is the text language and its conversion of speech. For this manner, if the user types another language word he will not be able to get the correct result.

3.2.2. Functionality

Mainly the functions of the system can be divided into two parts. For the main, it can be said that the system will convert the texts to speech. For this manner the system's TTS engine will just combine the syllables. There will be no accent or idiom. Second part is to create a text to send this TTS engine For this part in order to improve the usability there will be three options explaining in document's modules part. For these modules there will be sub-modules, again to increase the usability even these modules.

3.2.3. Usability

The usability of this system and its improvement is the main purpose of this application. As the system is a result of a specific need the system should answer this need so that the user, speech-impaired people, can easily use this system. For this manner, as it is said in 3.2.2. we are going to implement the system by modules and its sub-modules increasing the system speed and flow.

4. Data Design

4.1. Data Description

There will be 7 types of data objects in the system.

- **Phrase objects** : This object holds information about phrases which is used to explain actions. Objects have their text and id's.
- **Icon object** : This object holds information about icons which is used to explain action by help of icons. Objects have their text, id and a path describing image place associated with this icon.
- **Text object** : This object is an object to hold the specified text. Since text can come from different locations, from phrases, icons or flat user typed text, the application needs such kind of object to construct same object.
- **Voice object** : This object is an object to convert the specified text to speech. It takes a text as a field. Its voice method uses the syllable object.
- **Syllable object** : This object is used for converting texts to speech. It has strong communication with the database.

- **Logger object** : This object is an object to store the values of errors, warnings and developer specified messages to get rid of potential problems.
- **Database object** : This object is an object to create a bridge between the system and database.

4.2. ER Design

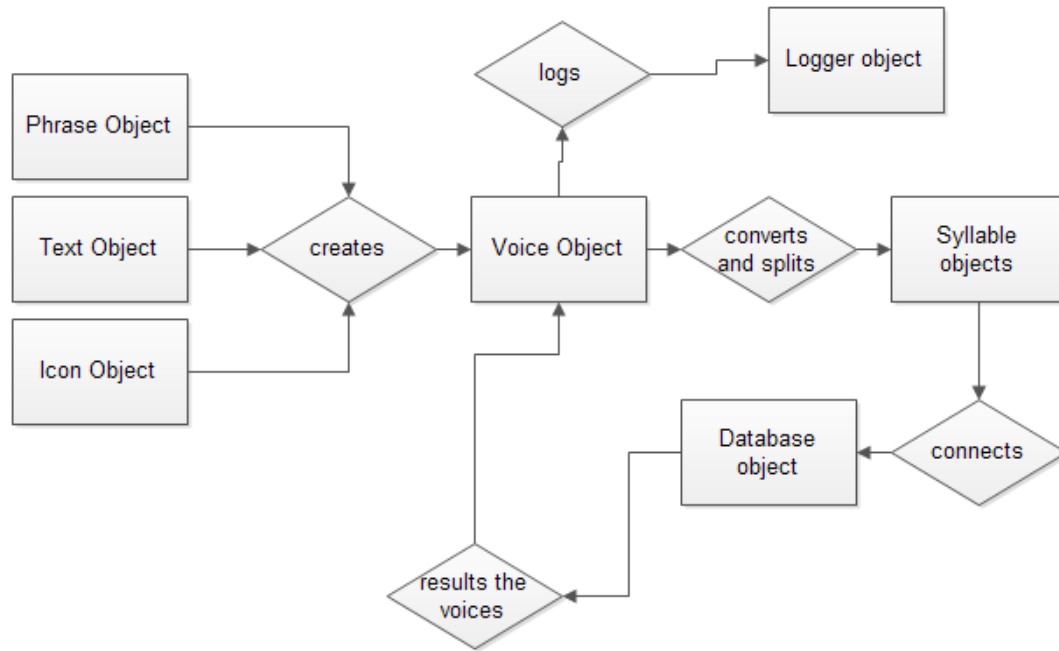


Figure -2-

4.3. Database Schemas

Syllable Table:

Field	Type	Null	Foreign Key	References
id	Varchar(20)	No	No	--
startletter	Varchar(1)	No	No	--
fullsyllable	Varchar(4)	No	No	--
voicepath	Varchar(50)	No	No	--

Syllable table is the table which holds all syllables in Turkish language and their voices. For search issue and its simplicity the system is going to store first letters of these syllables and a specific id for all of them.

Phrases Table:

Field	Type	Null	Foreign Key	References
id	Varchar(20)	No	No	--
startletter	Varchar(1)	No	No	--
fullphrase	Varchar(400)	No	No	--
usagecount	int	No	No	--
isFamous	Bit	Yes	No	--
createDate	Datetime	No	No	--
lastUsageDate	Datetime	No	No	--

In order to keep phrases and give chance to user to choose a phrase the system keeps phrases in the database. For search simplicity the system keeps first letter of phrases. In order to show the most popular phrases and last phrases the system keeps two columns for specify these.

Icons Table

Field	Type	Null	Foreign Key	References
id	Varchar(20)	No	No	--
startletter	Varchar(1)	No	No	--
fullphrase	Varchar(4)	No	No	--
imagepath	Varchar(50)	No	No	--
usagecount	Int	No	No	--
isFamous	bit	Yes	No	--
createDate	Datetime	No	No	--
lastUsageDate	Datetime	No	No	--

In order to implement icon module the system keeps necessary information in database. Different than the phrases table this one holds imagepath including a path in which the icon is kept.

4.4. Class Diagrams

4.4.1. Model Package

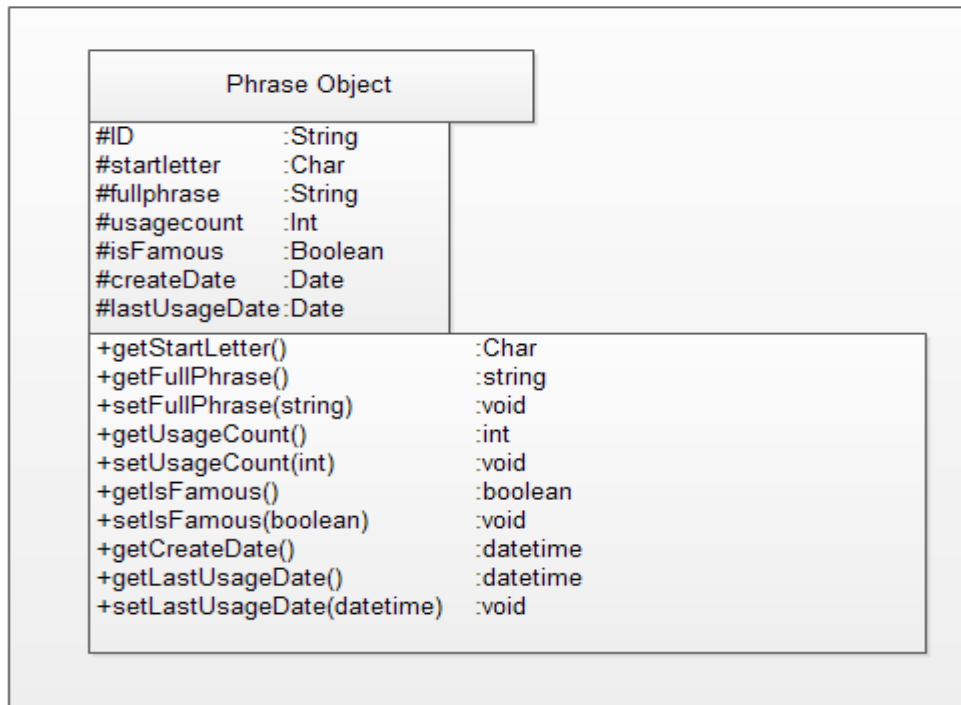


Figure -3-

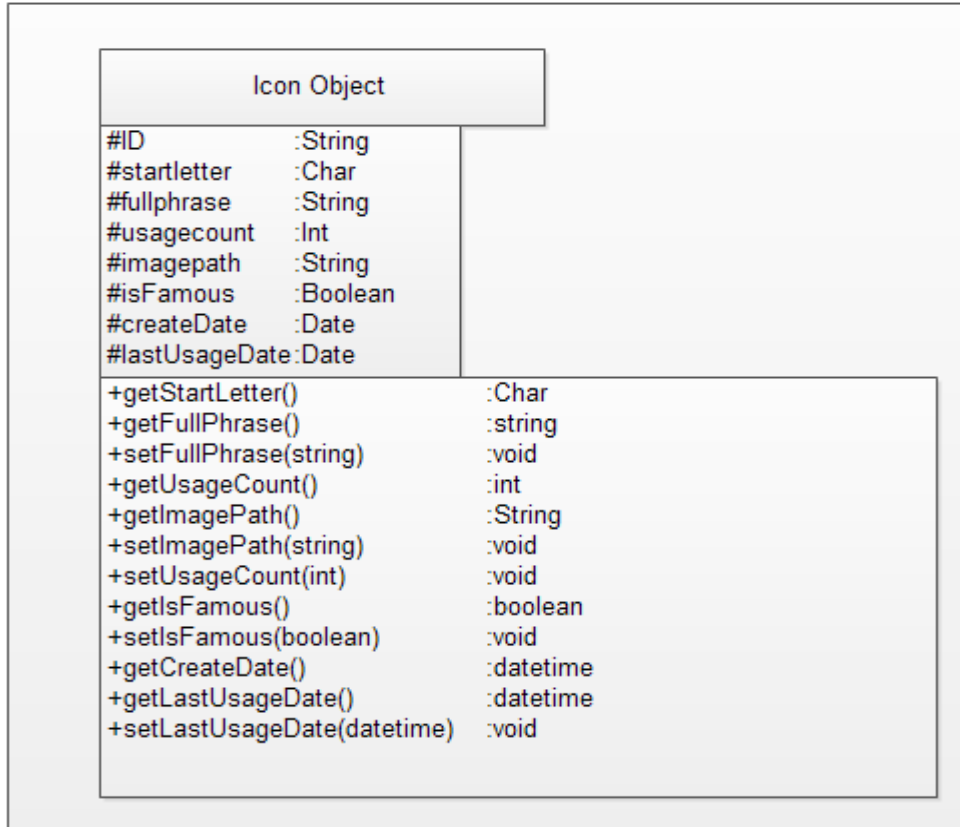


Figure -4-

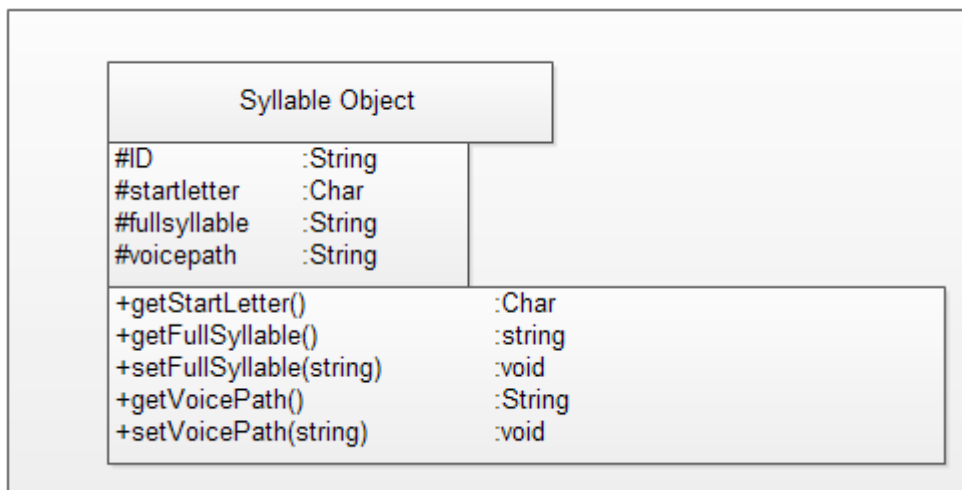


Figure -5-

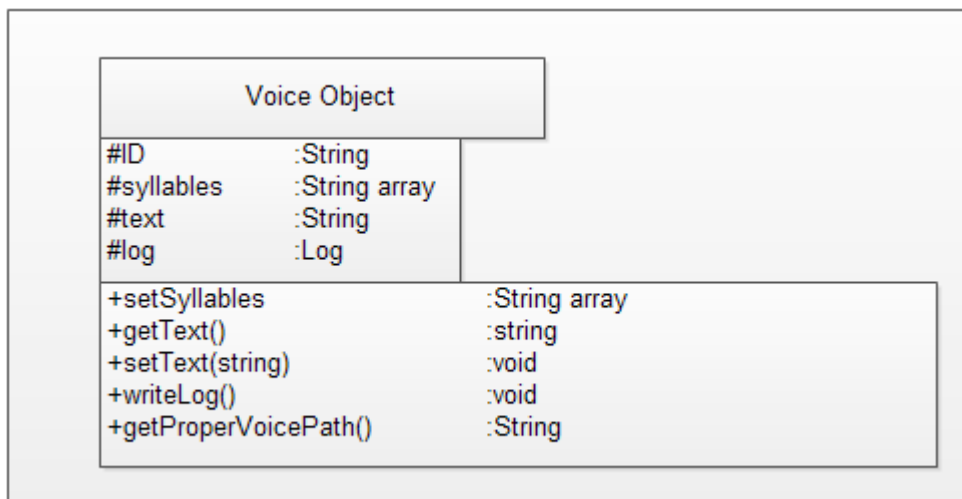


Figure -6-

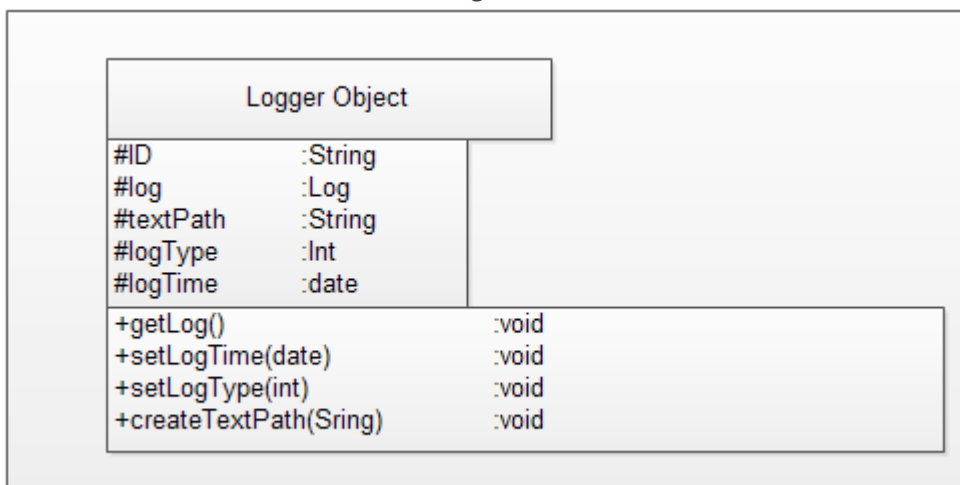


Figure -7-

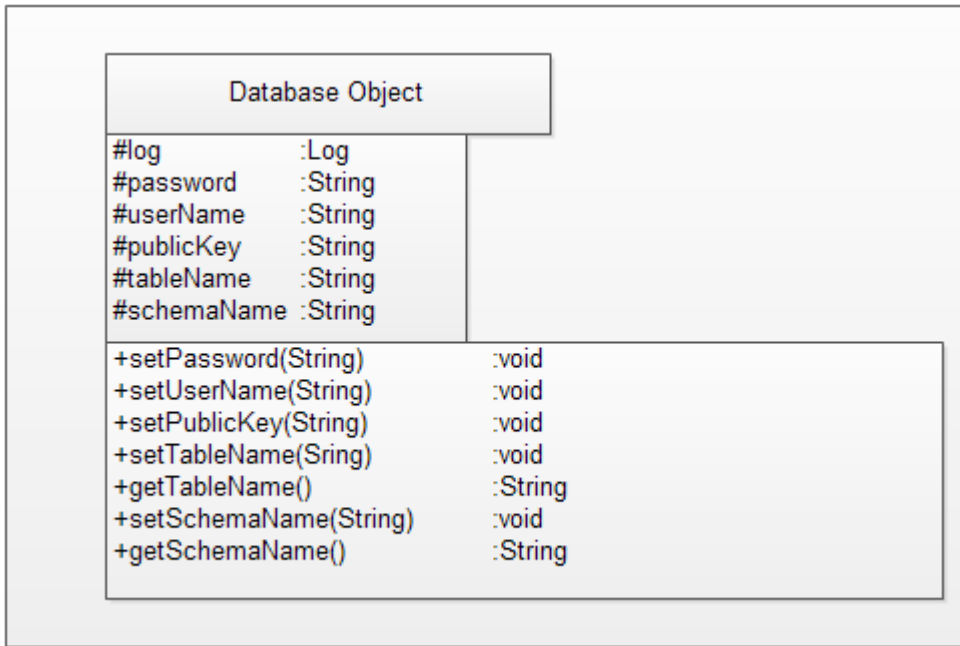


Figure -8-

4.5. Data Description

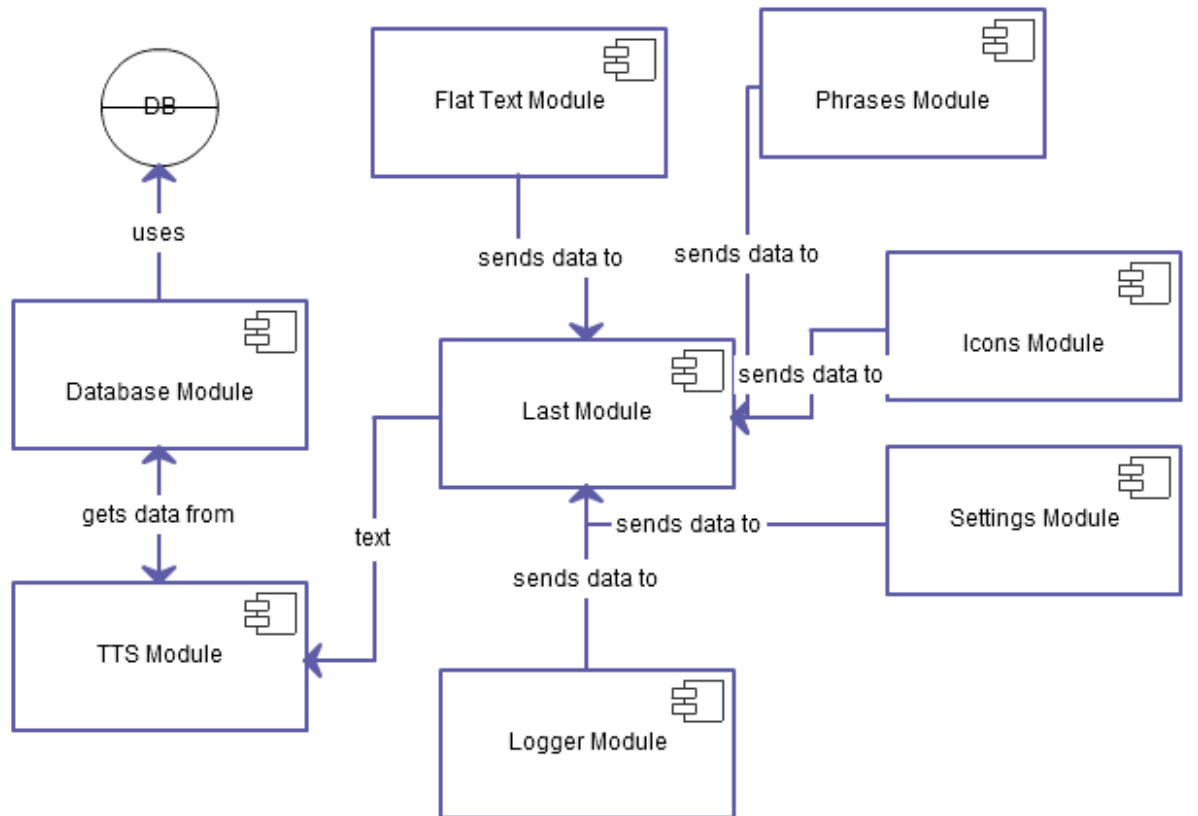
DATA	TYPE	DESCRIPTION
fullphrase	String	This data is the full phrase of phrases object. And it keeps full phrase of the user.
fullphrase	String	This data is the full phrase of icons object. And it keeps full phrase of the user.
fullsyllable	String	This data keeps full syllable. It helps combining these syllables and creating a voice text and voice object.
isFamous	boolean	This data is kept for phrase object and helping determining the user's famous phrases.
isFamous	boolean	This data is kept for icons object and helping determining the user's famous icons.
lastUsage	Date	This data is kept for phrase object and helping determining the user's last phrases.
lastUsage	Date	This data is kept for phrase object and helping determining the user's last icons.
log	String	This data is one of the most important data in the system. The system saves the all states

		of the application. While determining the errors and getting rid of the bugs the developers can use these logs.
userName	String	It keeps the username of the database.
password	String	It keeps the password of the database.
voice	String	It is for voice object and helps creating last real time voice

5. System Architecture

5.1. Architectural Design

The architectural design consists of 8 components. These are Last, Flat Text, Phrases, Icons, Database, Text-to-Speech, Logger and Settings modules. Text-to-Speech as being the most important part of the design is the responsible part for converting text to speech. This component cooperates with Database, Phrases and Icons modules. Namely, Text-to-Speech gets data that comes from Phrases and Icons processes those data and sends to result to the Database and take their equivalent voices then combines these voices appropriately. Logger module is responsible from maintenance of the system. All warning/error messages will be saved to the log files. This module enables us to update our application according to warning messages. Flat Text module enables user to type text using the keyboard of android system. After text typing this module sends the data to last module. Settings Module give chance to user make adjustment as he/she wants. Last module is the parent module of our system. In this state the processed last version of information from previous modules are collected and stored.



[online diagramming & design] .com

Figure -9- Component Diagram

5.2. Description of Components

5.2.1. Last Module

All modules of the system except TTS Module and Database Module connected to this module. Namely all information gathering from user come to this module and processed. Last Module provides connection with TTS module.

5.2.1.1. Processing Narrative for Last Module

Last Module is responsible for getting data from Flat Text Module, Phrases Module, Icons Module, Settings Module and Logger module. All data get into last module processed according to their types. Data comes from Flat Text Module, Phrases Module and Icons Module sends to TTS Module to converting texts to voice.

5.2.1.2. Interface Description for Last Module

The interface of this module as follows;

Data Interface: This interface provides an interface to the system to carry data between the modules. All the data through the system will flow over this interface.

Time interface: This interface provides time information to the system. Time interface ticks all the system at the appropriate times.

5.2.1.3. Processing Detail for Last Module

All the communications is done over this module except Database and TTS communication. In other words the modules cannot communicate with each other they only communicate with the last module. All the information is that needs to be transmitted somewhere is hold in the last module. So when a module wants data to be transmitted somewhere it writes the data to the appropriate port of the core module. Then core module sends the data to the appropriate module.

5.2.1.4. Dynamic Behavior for Last Module

Last module has interaction with Flat Text Module, Phrase Module, Icons Module, Logger Module, Settings Module, TTS Module. The relation can be explained like, Last Module gets data from modules that interacted with Last module and sends data to TTS Module.

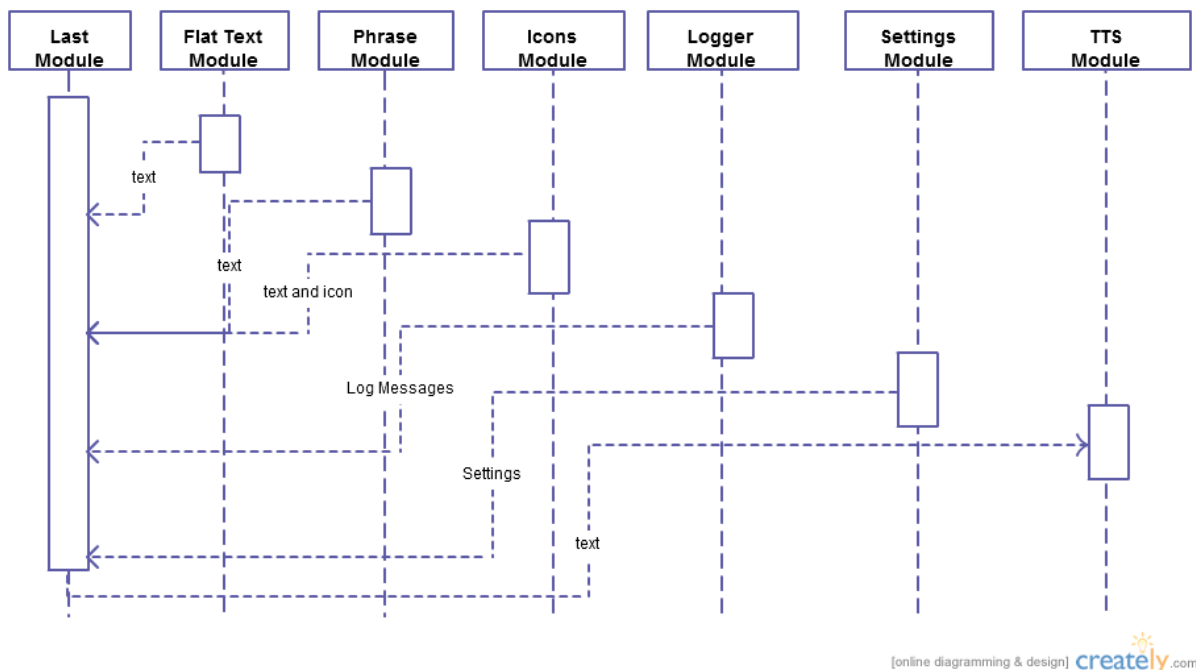


Figure -10- Last Module Sequence Diagram

5.2.2. Flat Text Module

5.2.2.1. Processing Narrative for Flat Text Module

Flat Text module is responsible from getting text input to user. For this manner the system uses the keyboard of android and its features such as Swype manner. This module can be used anywhere that requires typing.

5.2.2.2. Flat Text Module Interface Description

Flat Text Module has interface with last module. Flat Text Module sends data to Last Module. The input interface of this module is android operating system.

5.2.2.3. Flat Text Module Processing Detail

Flat Text Module enables user to type text. These texts are send to Last Module, therefore; there is an interaction with Flat Text Module and Last Module.

5.2.2.4. *Dynamic Behavior Flat Text Module*

Flat Text Module has interaction with Last Module.

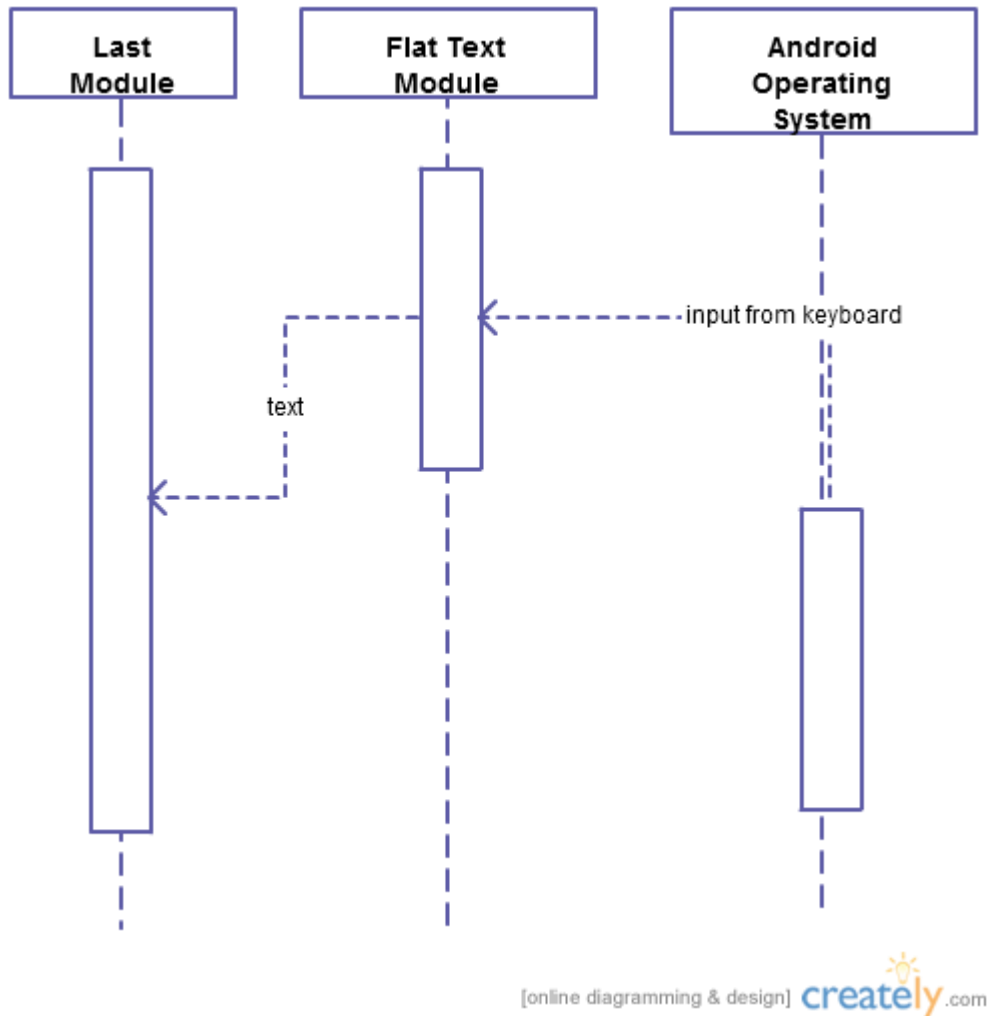


Figure -11- Flat Text Module Sequence Diagram

5.2.3. **Phrases Module**

Phrases module is designed to provide user select pre-saved phrases .User can choose appropriate pre-saved phrase from the pre-saved phrases library and also combine two or more phrases or phrases with text, save new phrases or modify and update the old ones. This part of the application constitutes one of the most powerful properties of the system.

5.2.3.1. *Processing Narrative for Phrases Module*

Phrases module is responsible from organizing phrases and occupations related phrases. It sends data to the last module and does not connect other modules. It uses pre-saved phrases library.

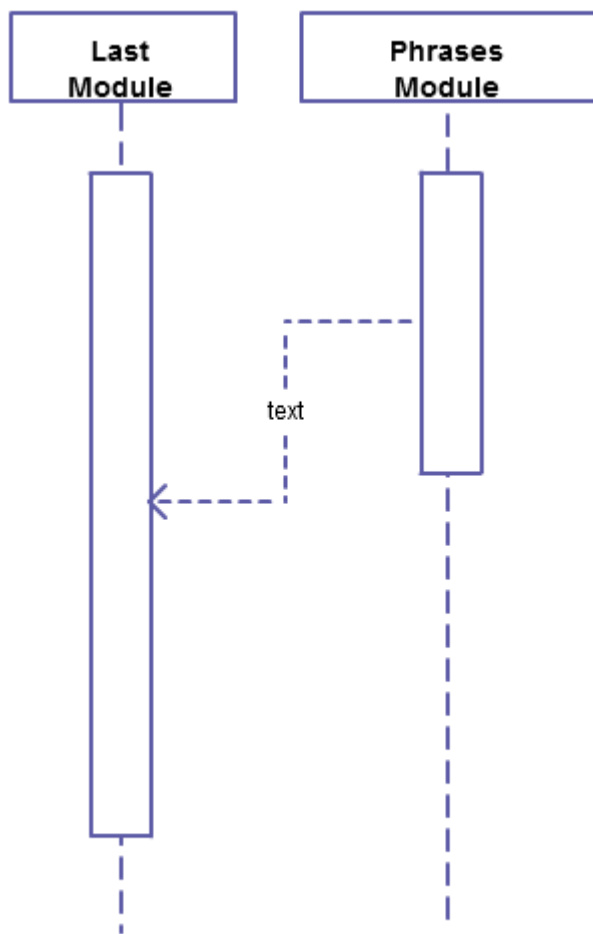
5.2.3.2. *Phrases Module Interface Description*

Phrases module has interface with last module and android device. Phrases module sends data to last module. Its interface is similar to flat text module.

5.2.3.3. *Phrases Module Processing Detail*

Phrases module should get data from pre-saved phrases library, then sends it to the last module. Last module packs and sends to the text to speech engine module .

5.2.3.4. *Dynamic Behavior for Phrases Module*



[online diagramming & design] 

Figure -12- Phrases Module Sequence Diagram

5.2.4. Icons Module

5.2.4.1. Processing Narrative for Icons Module

Icons module is responsible for processes about Icons namely, listing icons, adding new icon, deleting icon and updating icon. Icons Module also has interaction with camera. Photos taken by camera can be used as icon.

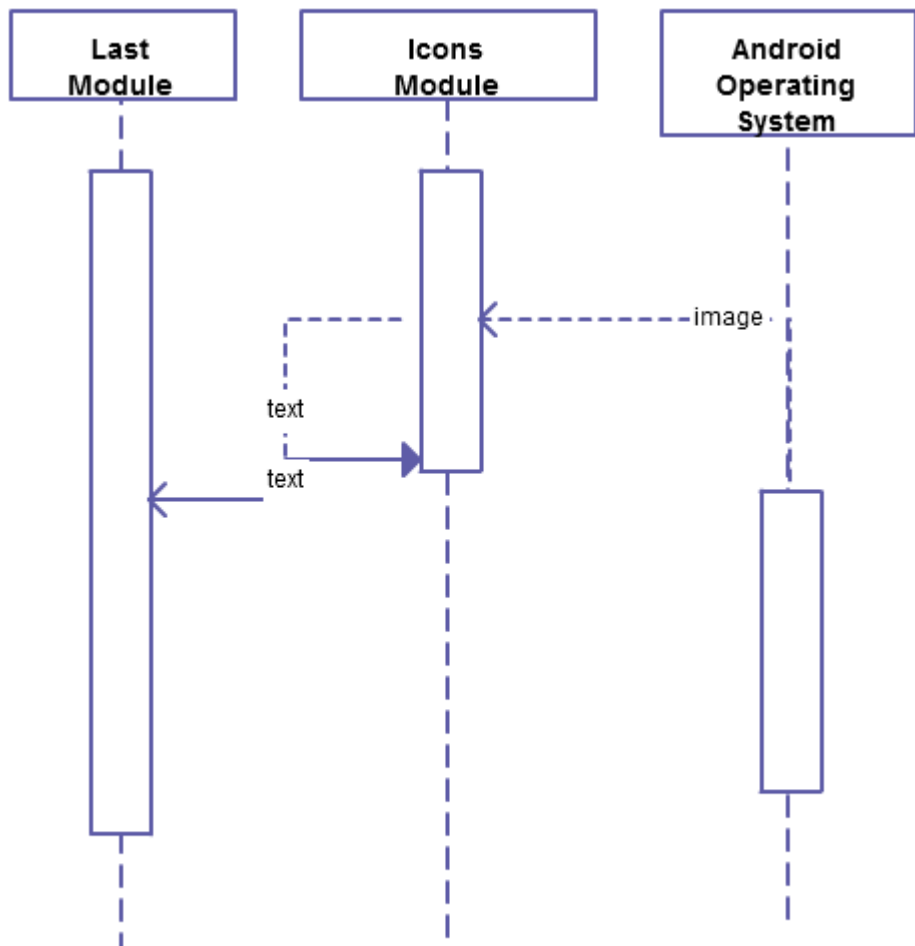
5.2.4.2. Icons Module Interface Description

Icons module has interface with Last Module since the icons send to the Last Module to be processed. Also Icons Module has interface with device's camera to adding new icon.

5.2.4.3. Icons Module Processing Detail

Icons Module list icons that are pre-reserved at database. While adding new icon Icons Module uses Flat Text Module and device's camera. All data are collected at Last Module then processed here. Icons Module to delete or updating an icon interacts with database through Last Module.

5.2.4.4. Dynamic Behavior Icons Module



[online diagramming & design] 

Figure -13- Icons Module Sequence Diagram

5.2.5. Database Module

5.2.5.1. *Processing Narrative for Database Module*

This module builds a bridge between database management system and last module. This module only communicates with last module. This means that, if any module wants to execute a database operation, it must communicate last module first then database module.

5.2.5.2. *Interface Description for Database Module*

Interface of database module consists of model package. Class diagrams of these packages are presented in the class diagrams section of data design. Phrase class, icon class, syllable class, voice class, logger class, database class are the classes of model package.

5.2.5.3. *Processing Detail for Database Module*

Classes from model package are mapped into database tables. Every class object is mapped to an identical table. Save, delete, update operations are done by these class instances.

5.2.5.4. *Dynamic Behavior*

Data Design section explains the dynamic behavior of this module.

5.2.6. Text-to-Speech Module

5.2.6.1. *Processing Narrative for Text-to-Speech Module*

TTS module is responsible for converting texts to speech. First of all, data received from Last Module splitting into syllable then these syllable sending to database to getting equivalent voices. Database sends equivalent voices to TTS then these voices combine. Finally, we can get response voices to text.

5.2.6.2. *Text-to-Speech Interface Description*

TTS has interface between Last Module since all text send to TTS from Last Module. Also TTS has interface between Database Module to access database.

5.2.6.3. *Text-to-Speech Processing Detail*

When program needs to convert text to speech, TTS Module makes process on text gathered from Last module. Then, interaction between database gives us equivalent voices of text gathered from Last Module.

5.2.6.4. *Dynamic Behavior Text-to-Speech*

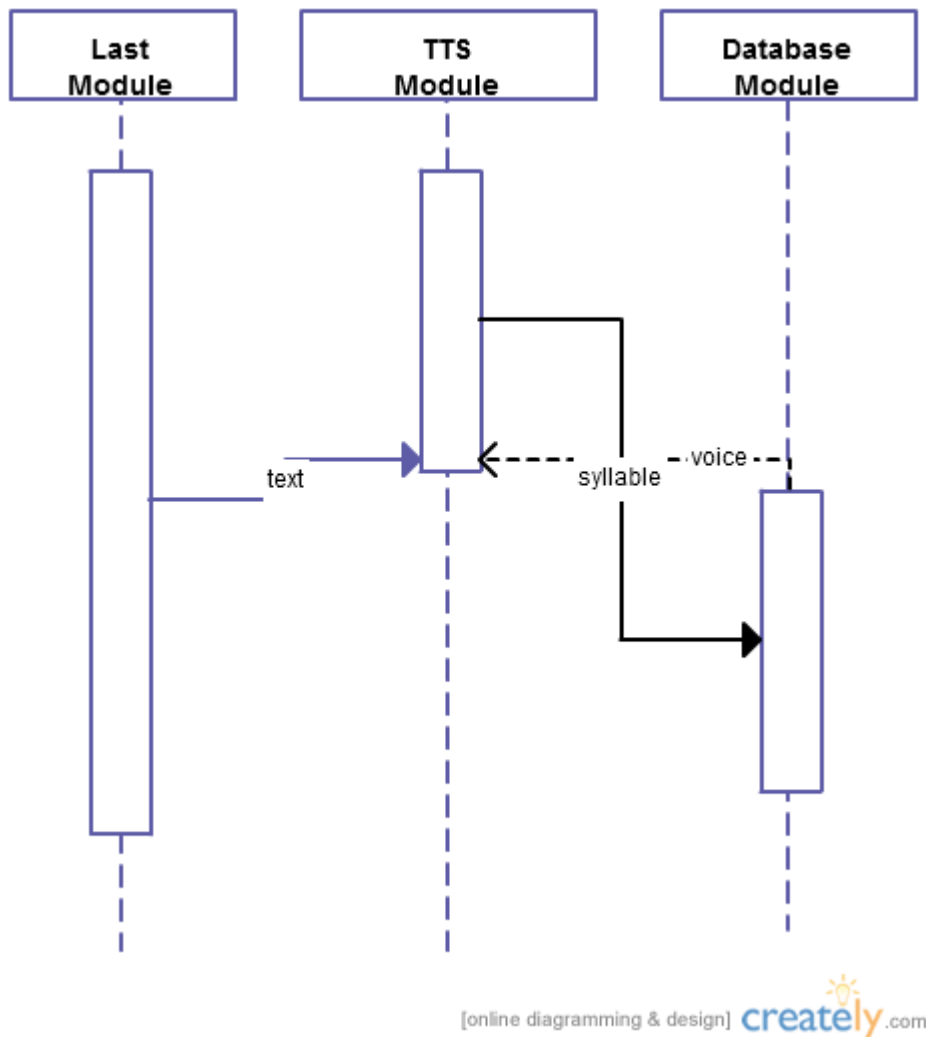


Figure – TTS Module Sequence Diagram

5.2.7. **Logger Module**

5.2.7.1. *Processing narrative for Logger Module*

Logger Module is responsible for saving error messages to the log file. System brings error messages from other modules to Last Module then they send to Logger Module to be processed.

5.2.7.2. *Logger Module Interface Description*

Logger Module has interface between Last Module.

5.2.7.3. *Logger module Processing Detail*

Logger Module keeps log files to present when they are called. When the request comes, log files are received from database through Last module.

5.2.7.4. *Dynamic Behavior Logger Module*

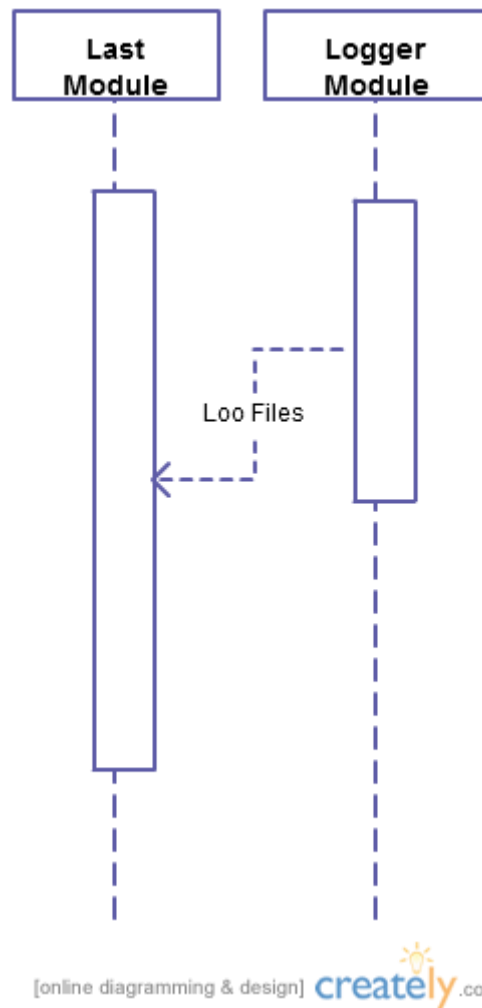


Figure -14- Logger Module Sequence Diagram

5.2.8. **Settings Module**

This Module is for getting and saving user settings. It provides an interface selection options to user and gets the user request and sends them to database module to save proper part.

5.2.8.1. *Processing Narrative for Settings Module*

Settings module is responsible from personalization of the application. There is 4 options in the settings module. They are “application settings”, “feedback” , “about us”, “help” . User can change the settings by pressing the menu button .The input are received via the device and data are processed in the setting module , then It sends data to the last module and does not connect other modules.

5.2.8.2. *Interface Description for Settings Module*

Settings module has interface with android OS. Phrases module sends data to last module. Four options are shown. Also it has interface with Last Module since all data processed at Settings Module send to Last Module.

5.2.8.3. Processing Detail for Settings Module

This module gets input from and processes it and then sends to the last module. In the text-to-speech module while converting texts to speech these settings are taken by this module and provided to text-to-speech module.

5.2.8.4. Dynamic Behavior

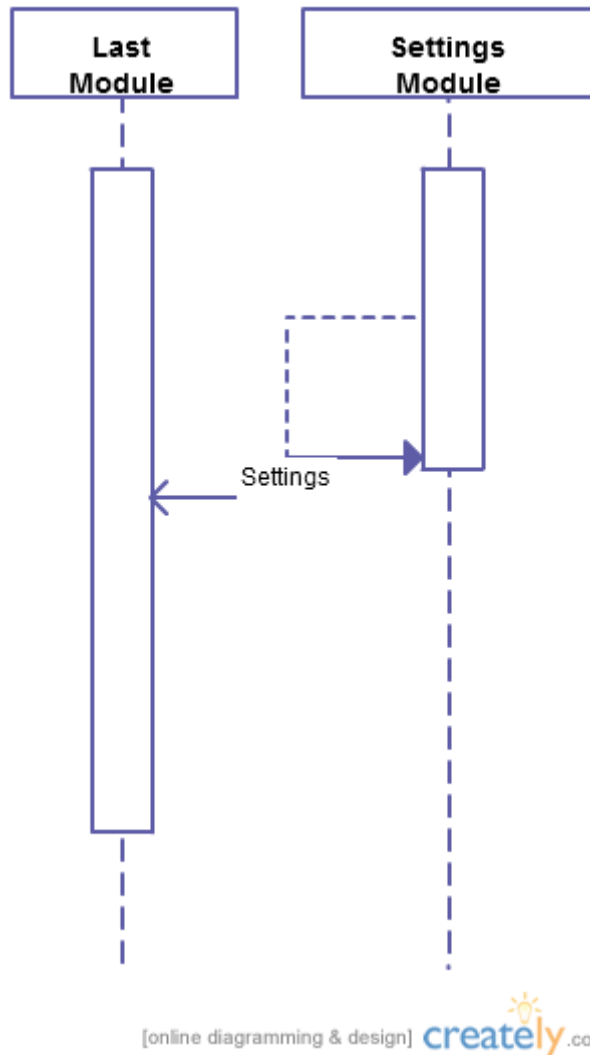


Figure -15- Settings Module Sequence Diagram

5.3. Design Rationale

We decided to 8 part composition since we need 8 main functions. For these functions we created 8 modules. These are last module, flat text module, phrases module, icons module, database module, text-to-speech module, logger module and settings module. Last module collects and stored last version of information from previous modules, then packs and sends to the text to speech engine module. Flat text module provides a chance for user to type text, then sends the data to the last module. Phrases module uses pre-saved phrases which are saved by user, so user can easily select phrases instead of typing phrases again and

again. Icons module matches the icon and the related event. Database module saves images and its phrases with their information like usage count, saved date etc. Text-to-speech module runs after the last module and converts the processed version of data to text. Logger module saves the error message to the log file. Settings module gets and sets user settings. While developing these modules we use flow-based programming which is a particular form of dataflow programming. The purpose of this type of implementation is simple. The developer can add and remove modules with a little cost. We create our own text to speech engine if we countered unexpected and insoluble problems then we use text to speech engine of "GOOGLE" by using "GOOGLE" text to speech API.

6. User Interface Design

6.1. Overview of User Interface

6.1.1. Main Page

In application main page, there will be three buttons namely flat text screen, phrases and icons. Flat text screen button navigates user to flat screen page which he/she can directly write text and listen. Phrases button can navigate user to phrases screen which he/she can choose a phrase and listen it or save/delete/edit phrase. Icons button navigates user to icons screen which he/she can choose an icon and listen it's text or save/delete/edit icon with related to its text.

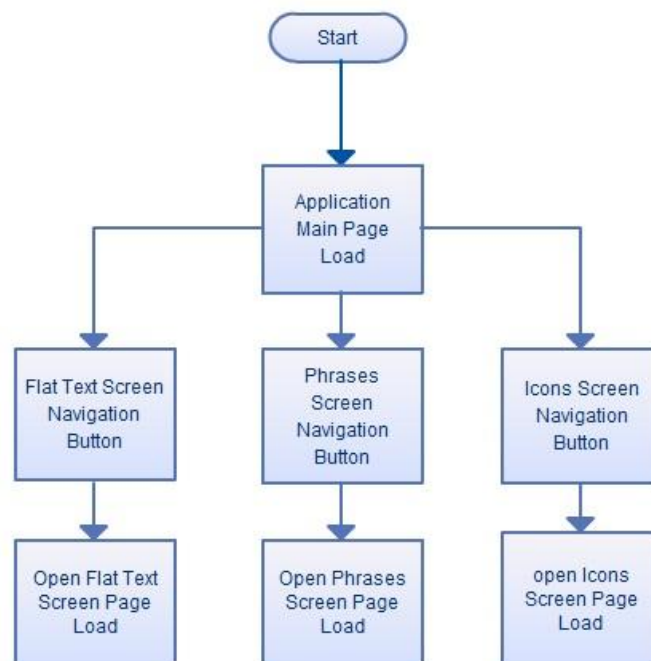


Figure -16-

6.1.2. Flat Text Screen

In flat text screen there will be a text box and a button and also a picture of a man or a woman which is selected from settings page. This page can be navigated from main page of application with flat text button from main screen. When in flat text screen, user can enter text into text box and click the listen button to run the text to speech engine. Program first controls the input, if it is empty then a pop up message will be shown and says that “you must enter some text”. If text is not empty, then the text will be sent to text to speech engine and it returns the related voice.

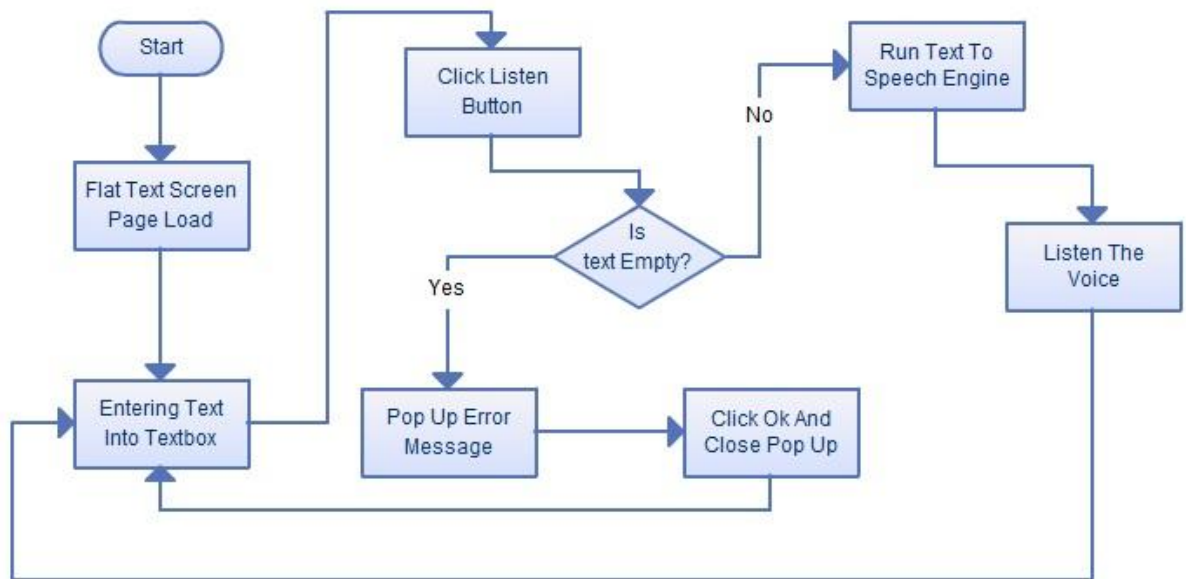


Figure -17-

6.1.3. Phrases Screen

In phrases screen, with on load function of page, there will be list of phrases and three buttons namely “add phrase”, “edit phrase” and “delete phrase”. Except for add button, other buttons are disabled at first. Selecting item from the list, edit and delete buttons will be enabled.

6.1.3.1. Add phrase

In adding phrase section, clicking add phrase button, a blank textbox and a submit button will be shown at the top of the list. User enters new phrase into the textbox and clicks submit button. Application first controls the emptiness of text if text is empty then an error message will be shown. If text is not empty then text will be controlled of existence of database. If phrase exists in database then error message will be shown. If phrase does not exist in database then phrase will be saved in the database and a success message will be shown in pop up.

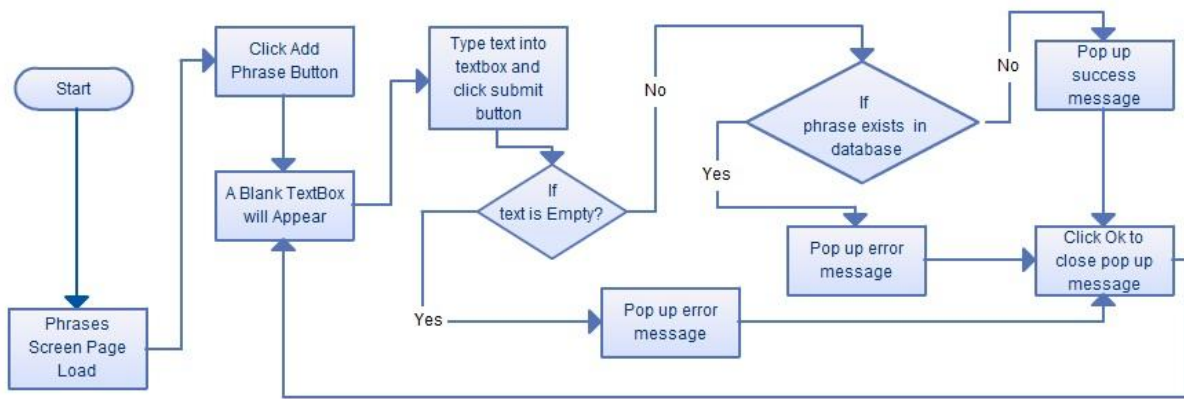


Figure -18-

6.1.3.2. Edit Phrase

In editing phrase section, selecting a phrase from list and clicking edit phrase button, a filled textbox and a submit button will be shown at the top of the list. User changes phrase in the textbox and clicks submit button. Application first controls the emptiness of text if text is empty then an error message will be shown. If text is not empty then text will be controlled of existence of database. If phrase exists in database then error message will be shown. If phrase does not exist in database then phrase will be saved in the database and a success message will be shown in pop up.

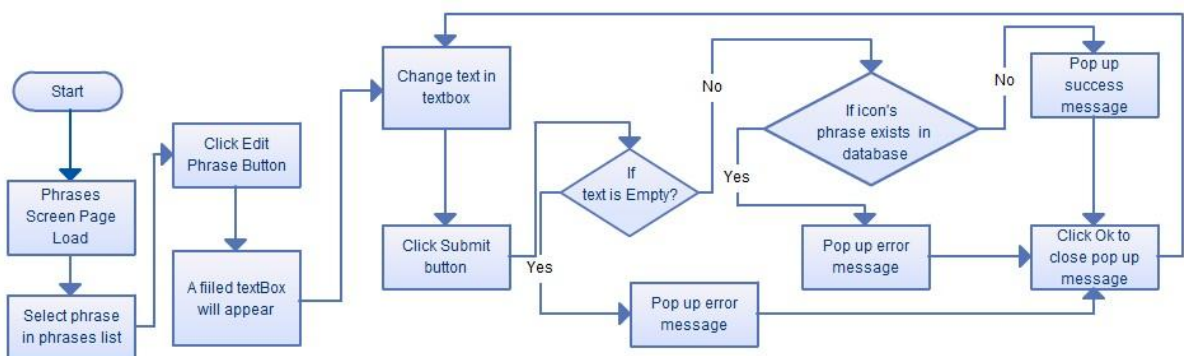


Figure -19-

6.1.3.3. Delete Phrase

In deleting phrase section, with selecting a phrase from list and clicking delete phrase button, a confirmation dialog box will be shown as pop up. Selecting ok, another pop up will be shown including the success message of deleting phrase. Selecting cancel will show the list with initial state.

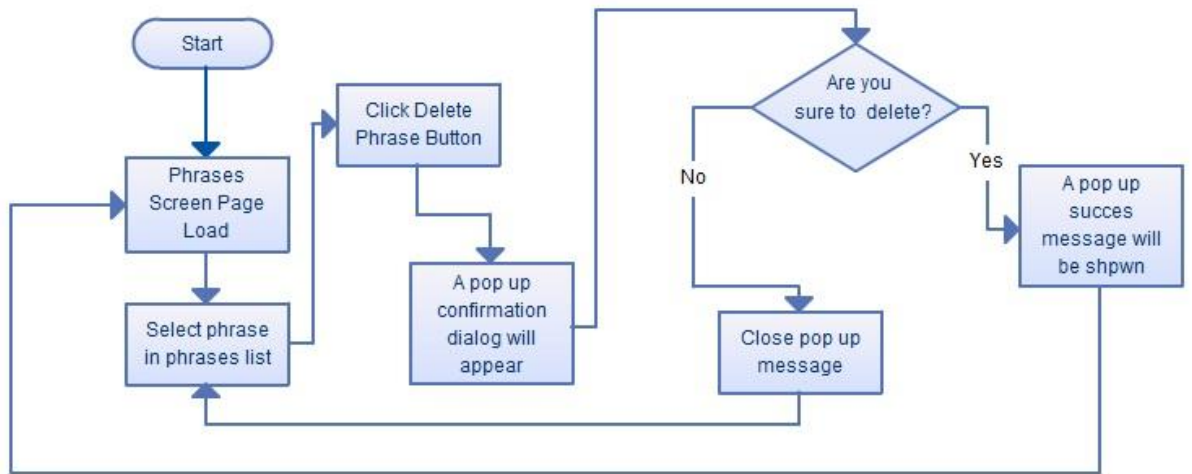


Figure -20-

6.1.4. Icons Screen

In icons screen, with on load function of page, there will be list of icons and three buttons namely “add icon”, “edit icon” and “delete icon”. Except for add button, other buttons are disabled at first. Selecting item from the list, edit and delete buttons will be enabled.

6.1.4.1. Add icon

In adding icon section, clicking add icon button, a thumbnail and a blank textbox and a submit button will be shown at the top of the list. User clicks the thumbnail and upload page will be shown. One can choose two options; from camera, from device. Selecting camera opens the device’s camera and he/she can capture from it. If user chooses the other option, file system will be open and he/she can select icon directly from device.

User enters new phrase into the textbox and clicks submit button. Application first controls the emptiness of text if text is empty then an error message will be shown. If text is not empty then text will be controlled of existence of database. If phrase exists in database then error message will be shown. If phrase does not exist in database then phrase will be saved in the database and a success message will be shown in pop up.

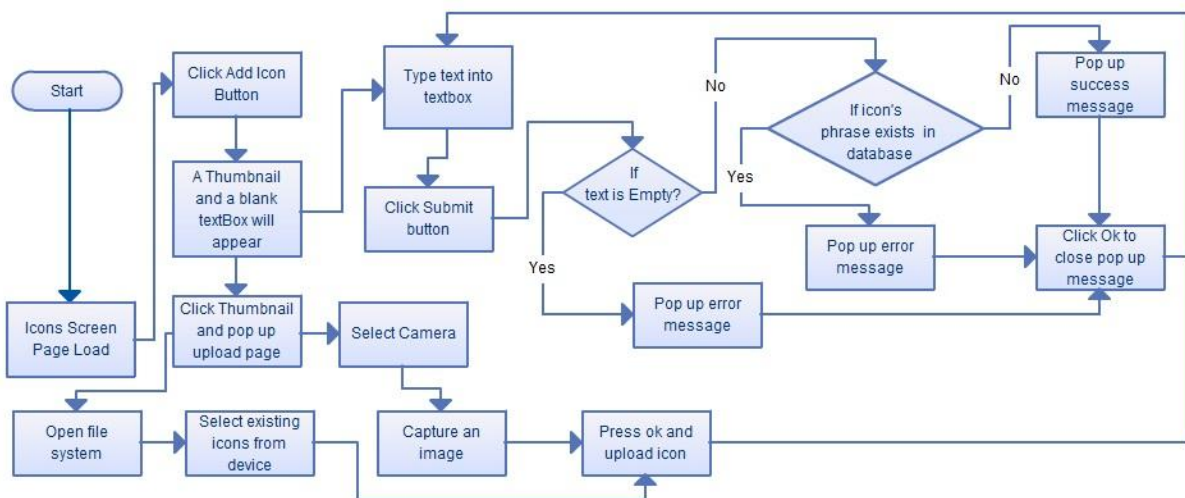


Figure -21-

6.1.4.2. Edit Icon

In editing icon section, selecting a phrase from list and clicking edit icon button, a thumbnail, a filled textbox and a submit button will be shown at the top of the list. User clicks the thumbnail and upload page will be shown. One can choose two options; from camera, from device. Selecting camera opens the device's camera and he/she can capture from it. If user chooses the other option, file system will be open and he/she can select icon directly from device.

User changes phrase in the textbox and clicks submit button. Application first controls the emptiness of text if text is empty then an error message will be shown. If text is not empty then text will be controlled of existence of database. If phrase exists in database then error message will be shown. If phrase does not exist in database then phrase will be saved in the database and a success message will be shown in pop up.

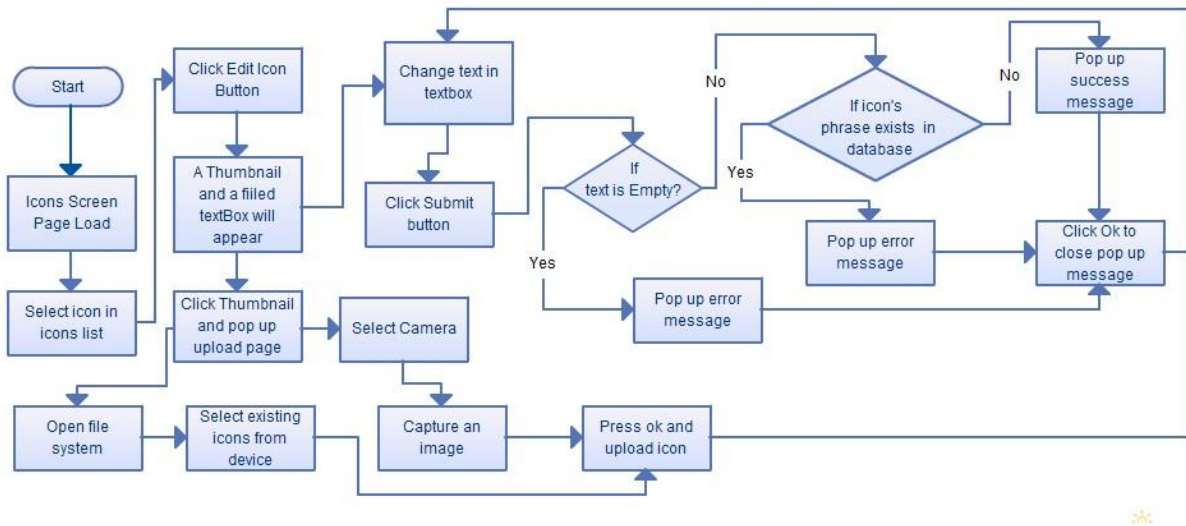


Figure -22-

6.1.4.3. Delete Icon

In deleting icon section, with selecting an icon from list and clicking delete icon button, a confirmation dialog box will be shown as pop up. Selecting ok, another pop up will be shown including the success message of deleting phrase. Selecting cancel will show the list with initial state.

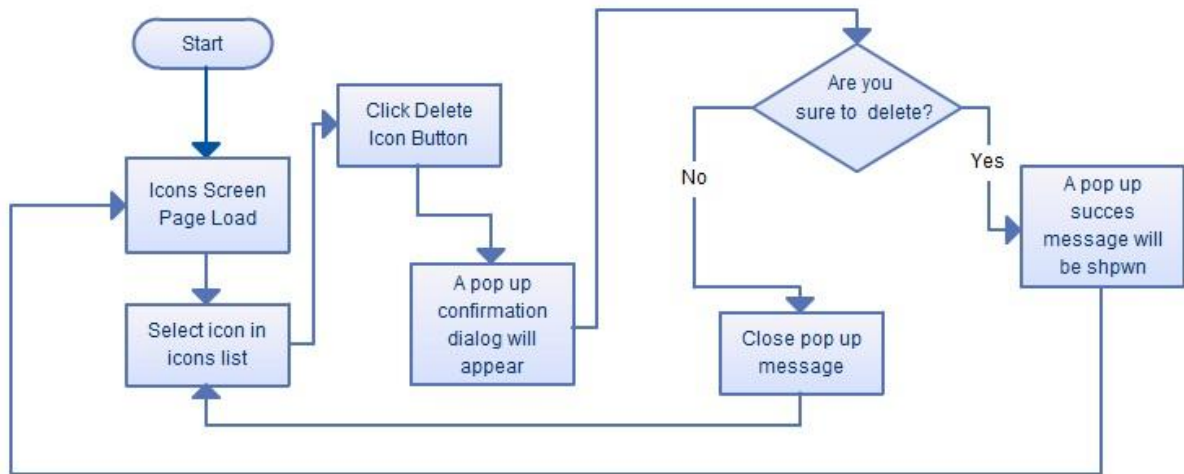


Figure -23-

6.1.5. Menu

Pressing the menu button in user's android device will show four options to his/her namely "application settings", "feedback", "about us", "help".

In about us section there will be explanation about the developers and contributors.

In help section there will be a text explaining the usage, frequently asked questions and general possible problems by heading of them.

In feedback section there will be a text box to write the ideas or defect's explanations and a button named as "submit" in order to send them to the application technical support team.

In application settings section there will be radio buttons to select voice type among the man voice and woman voice. In order to adjust the volume there will be a cursor.

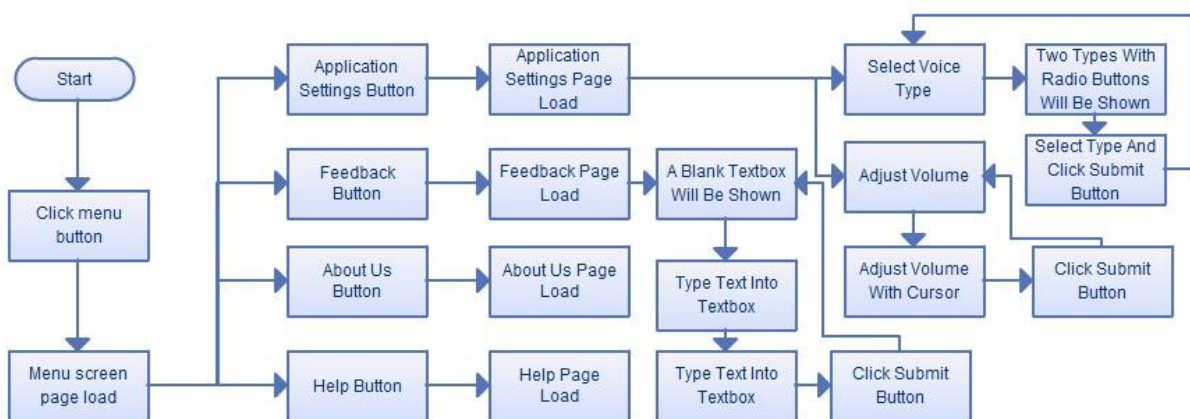


Figure -24-

7. Time Planning (Gantt Chart)

The gantt chart that shows the basic time schedule of the project is as follows with the tasks and dates included.

7.1. Term 1

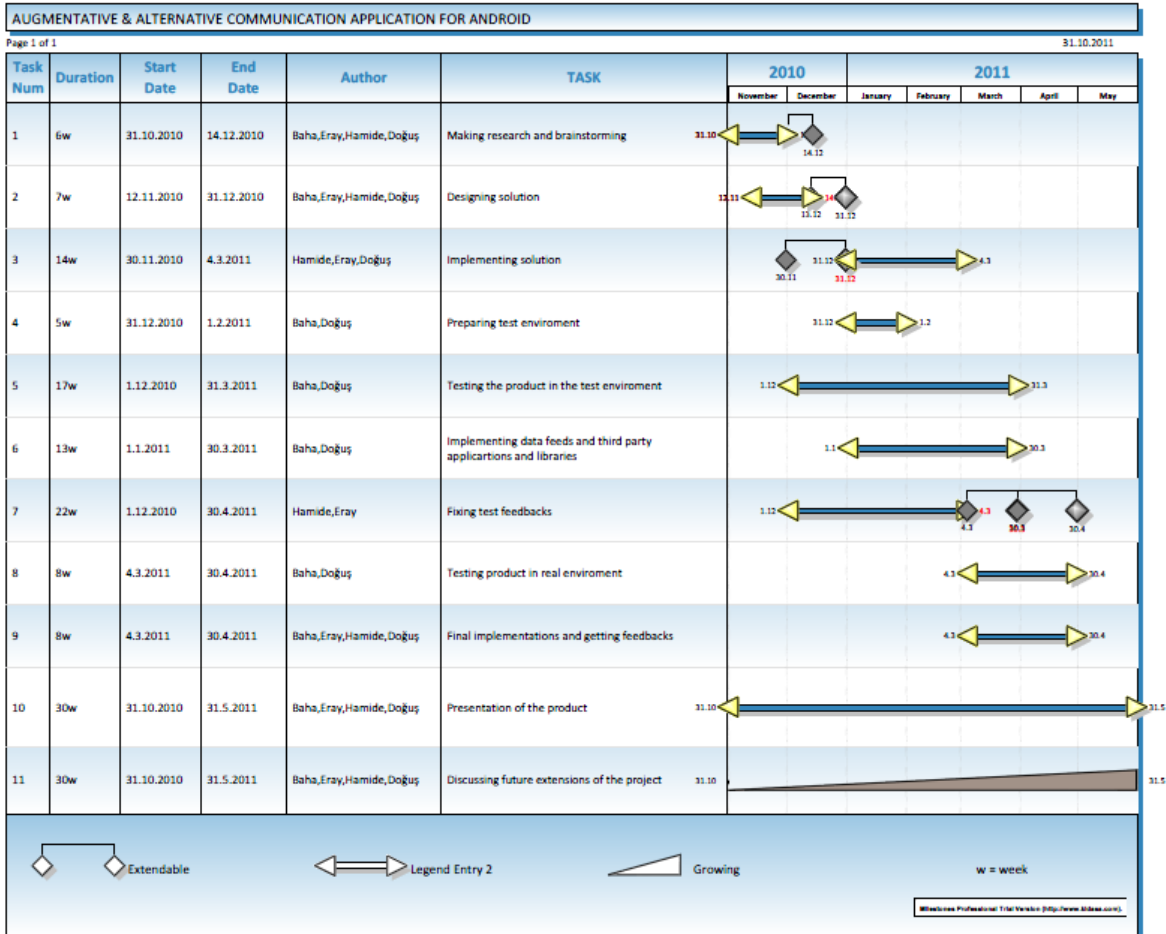


Figure -25- Gantt Chart for Term 1

7.2. Term 2

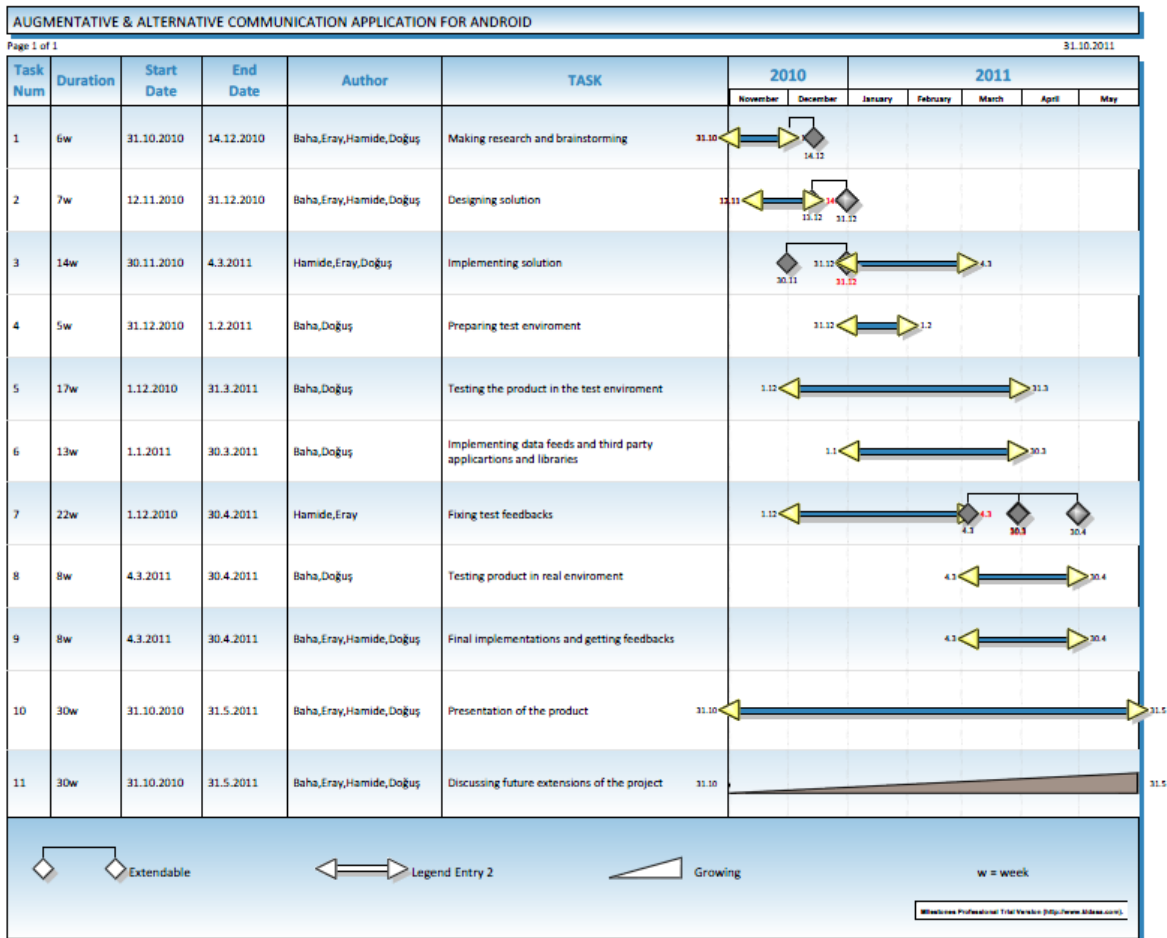


Figure -26- Gantt Chart for Term 2

8. Libraries and Tools

Eclipse: It is an IDE that we are going to use for design. It is really compatible with Java programming language, and with the library provided by Minder to use BCID. Therefore we have chosen this IDE to design the system.

Java: Because of the libraries provided of use, we are going to use Java.

MotoDev Studio: It is a development platform, it includes some useful design tools that we are going to use while developing our android application.

MotoDev Emulator: Since Innova does not provide us any device to test our project while developing application, then we need to use emulator. Motodev emulator is suitable for us.

SQLite Database: SQLiteDatabase has methods to create, delete, execute SQL commands, and perform other common database management tasks. Therefore, we will use SQLite Database in this project.

9. Conclusion

In conclusion, the definition, purpose and scope of the project are given in the Initial Design Report for Augmentative and alternative communication application for android. The possible design that we desire to have and other constraints that can be encountered are explained. We also decided which tools and the libraries will be used while developing the project. Entity relationship diagrams, data flow models, interface features, class diagrams, possible use cases, gantt chart are given within the document. In the schedule of the project the occupations that we will have are also given.