CENG491

Software Detailed Design Report

HANDE

Kadir Eray Doğanlar

Doğuş Küçükgöde

Baha Tosun

Hamide Hande Keskiner

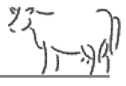
Table of Contents

1. Introduction 8
1.1. Problem Definition 8
1.2. Purpose 8
1.3. Scope 8
1.4. Overview 8
1.5. Definitions, Acronyms and Abbreviations 8
2. System Overview 8
2.1. Last Module 9
2.2. Flat Text Module 9
2.3. Phrases Module 9
2.4. Icons Module 10
2.5. Database Module 10
2.6. Text-to-speech Module 10
2.7. Logger 10
2.8. Settings Module 10
3. Design Considerations 10
3.1. Design Assumptions, Dependencies and Constraints 10
3.1.1. Time Constraints 10
3.1.2. Resource Constraints 10
3.1.3. Performance Constraints 11
3.1.4. Software Constraints 11
3.1.5. Hardware Constraints 11
3.2. Design Goals and Guidelines 11
3.2.1. Reliability 11
3.2.2. Functionality 11
3.2.3. Usability 11
4. Data Design 11
4.1. Data Description 11
4.1.1. ER Design 12
4.1.2. Database Schemas 12

4.1.3. Class Diagrams 14
Main Package 14
4.2. Data Dictionary 18
5. System Architecture 20
5.1. Architectural Design 20
5.2. Description of Components 21
5.2.1. Last Module 21
5.2.1.1. Processing Narrative for Last Module 21
5.2.1.2. Interface Description for Last Module 21
5.2.1.3. Processing Detail for Last Module 22
5.2.1.4. Dynamic Behavior for Last Module 22
5.2.2. Flat Text Module 22
5.2.2.1. Processing Narrative for Flat Text Module 22
5.2.2.2. Flat Text Module Interface Description 22
5.2.2.3. Flat Text Module Processing Detail 23
5.2.2.4. Dynamic Behavior Flat Text Module 23
5.2.3. Phrases Module 23
5.2.3.1. Processing Narrative for Phrases Module 24
5.2.3.2. Phrases Module Interface Description 24
5.2.3.3. Phrases Module Processing Detail 24
5.2.3.4. Dynamic Behavior for Phrases Module 24
5.2.4. Icons Module 25
5.2.4.1. Processing Narrative for Icons Module 25
5.2.4.2. Icons Module Interface Description 25
5.2.4.3. Icons Module Processing Detail 25
5.2.4.4. Dynamic Behavior Icons Module 25
5.2.5. Database Module 26
5.2.5.1. Processing Narrative for Database Module 26
5.2.5.2. Interface Description for Database Module 26
5.2.5.3. Processing Detail for Database Module 26
5.2.5.4. Dynamic Behavior 26
5.2.6. Text-to-Speech Module 26
5.2.6.1. Processing Narrative for Text-to-Speech Module 26

5.2.6.2. Text-to-Speech Interface Description 26
5.2.6.3. Text-to-Speech Processing Detail 26
5.2.6.4. Dynamic Behavior Text-to-Speech 27
5.2.7. Logger Module 27
5.2.7.1. Processing narrative for Logger Module 27
5.2.7.2. Logger Module Interface Description 27
5.2.7.3. Logger module Processing Detail 27
5.2.7.4. Dynamic Behavior Logger Module 28
5.2.8. Settings Module 28
5.2.8.1. Processing Narrative for Settings Module 28
5.2.8.2. Interface Description for Settings Module 28
5.2.8.3. Processing Detail for Settings Module 29
5.2.8.4. Dynamic Behavior 29
5.3. Design Rationale 29
5.4. Traceability of requirements 30
6. User Interface Design 30
6.1. Overview of User Interface 30
6.1.1. Main Page 30
6.1.2. Flat Text Screen 31
6.1.3. Phrases Screen 32
6.1.3.1. Add phrase 32
6.1.3.2. Edit Phrase 33
6.1.3.3. Delete Phrase 33
6.1.4. Icons Screen 34
6.1.4.1. Add icon 34
6.1.4.2. Edit Icon 34
6.1.4.3. Delete Icon 35
6.1.5. Menu 35
6.2. Screen Images 37
6.2.1. Flat Text Screen 37
6.2.2. Phrases Screen 38
6.2.3. Icons Screen 40
6.3. Screen Objects and Actions 41

7. Detailed Design 41
7.1. Last Module 41
7.1.1. Classification 41
7.1.2. Definition 41
7.1.3. Responsibilities 42
7.1.4. Constraints 42
7.1.5. Composition 42
7.1.6. Uses and Interactions 42
7.1.7. Resources 42
7.1.8. Processing 42
7.1.9. Interface/Exports 42
7.2. Flat Text Module 43
7.2.1. Classification 43
7.2.2. Definition 43
7.2.3. Responsibilities 43
7.2.4. Constraints 43
7.2.5. Composition 43
7.2.6. Uses and Interactions 43
7.2.7. Resources 43
7.2.8. Processing 43
7.2.9. Interface/Exports 43
7.3. Phrases Module 44
7.3.1. Classification 44
7.3.2. Definition 44
7.3.3. Responsibilities 44
7.3.4. Constraints 44
7.3.5. Composition 45
7.3.6. Uses/Interactions 45
7.3.7. Resources 45
7.3.8. Processing 45
7.3.9. Interface/Exports 45
7.4. Icons Module 45
7.4.1. Classification 45



7.4.2. Definition 45
7.4.3. Responsibilities 46
7.4.4. Constraints 46
7.4.5. Composition 46
7.4.6. Uses and Interactions 46
7.4.7. Resources 46
7.4.8. Processing 46
7.4.9. Exports 47
7.5. Database Module 47
7.5.1. Classification 47
7.5.2. Definition 47
7.5.3. Responsibilities 47
7.5.4. Constraints 47
7.5.5. Composition 47
7.5.6. Uses and Interactions 48
7.5.7. Resources 48
7.5.8. Processing 48
7.5.9. Interface/Exports 48
7.6. Flat Text Module 48
7.6.1. Classification 48
7.6.2. Definition 48
7.6.3. Responsibilities 48
7.6.4. Constraints 48
7.6.5. Composition 48
7.6.6. Uses and Interactions 49
7.6.7. Resources 49
7.6.8. Processing 49
7.6.9. Interface/Exports 49
7.7. Logger Module 49
7.7.1. Classification 49
7.7.2. Definition 49
7.7.3. Responsibilities 49
7.7.4. Constraints 49

7.7.5. Composition 49
7.7.6. Uses and Interactions 49
7.7.7. Resources 49
7.7.8. Processing 50
7.7.9. Interface/Exports 50
7.8. Settings Module 50
7.8.1. Classification 50
7.8.2. Definition 50
7.8.3. Responsibilities 50
7.8.4. Constraints 50
7.8.5. Composition 50
7.8.6. Uses and Interactions 50
7.8.7. Resources 50
7.8.8. Processing 50
7.8.9. Interface/Exports 51
8. Time Planning (Gannt Chart) 51
8.1. Term 1 51
8.2. Term 2 52
9. Libraries and Tools 52
10. Conclusion 53

1. Introduction

1.1. Problem Definition

The problem that we attend to solve is to provide a chance to explain what they want to say for speech-impaired people by giving a device that can speak for him/her. These people cannot talk with other people due to lots of reasons. In this manner they cannot communicate well. By providing a device which talking for these people, helping them is the main problem for us to make this project. Another problem is to provide this communication as possible as fast and healthy. Talking by using these devices is not that easy. Thus making a fluent project is another issue.

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 express which can be a flat text, a pre-saved phrase or a picture explaining a certain event is going to be converted 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 this subject, processing means that converting all kind of user inputs to final string. User inputs can

be just a string provided by the user or be one of the pre-saved icons or phrases. Moreover, the user can produce his input by combining these different kind of sending inputs parts. In the last module the application will just convert the final string to speech by using TTS module and database module. In this part conversion process is made according to the pre-saved user settings and by using the pre-saved syllables. In order to follow user's steps in application all modules communicates with logger module to save necessary data to text files. In the Figure -1- below the system overview diagram is representing general system overview.

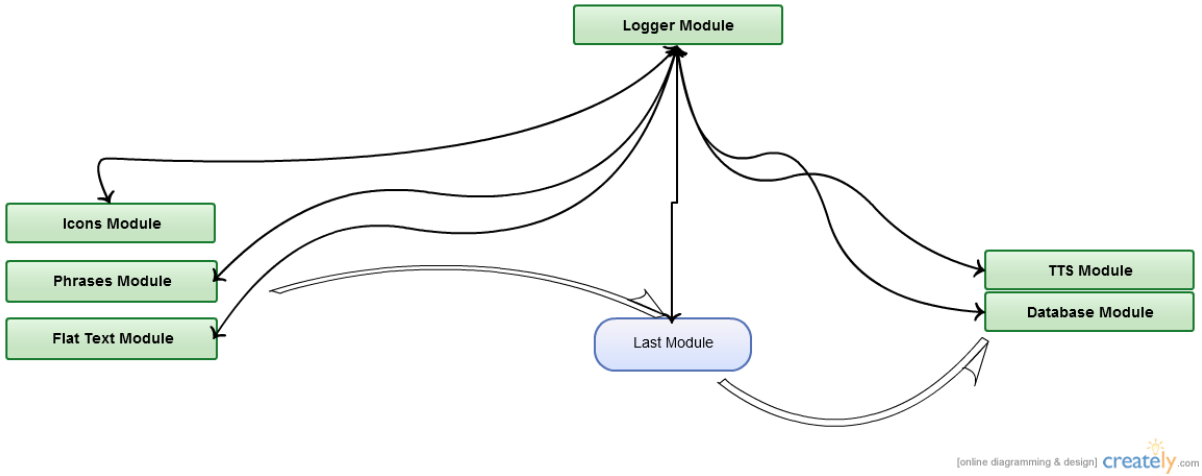


Figure -1- Collecting data from three different channels

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 get general usage of the user when an unexpected situation occurs. i.e. logger module keeps the every user step in application in a text file.

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 from 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 sad 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 sad 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.

4.1.1. ER Design

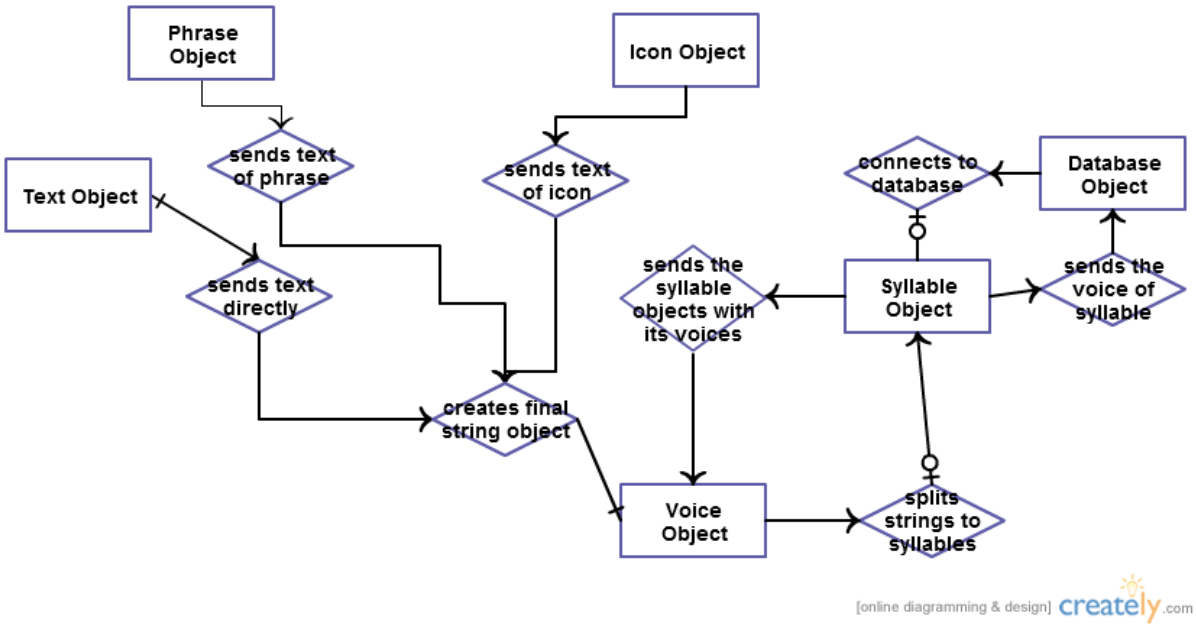


Figure -2- ER Diagram

4.1.2. 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 image path including a path in which the icon is kept.

4.1.3. Class Diagrams

The system has 6 classes to achieve its goals. Just to keep them together we create one package called as main package.

Main Package

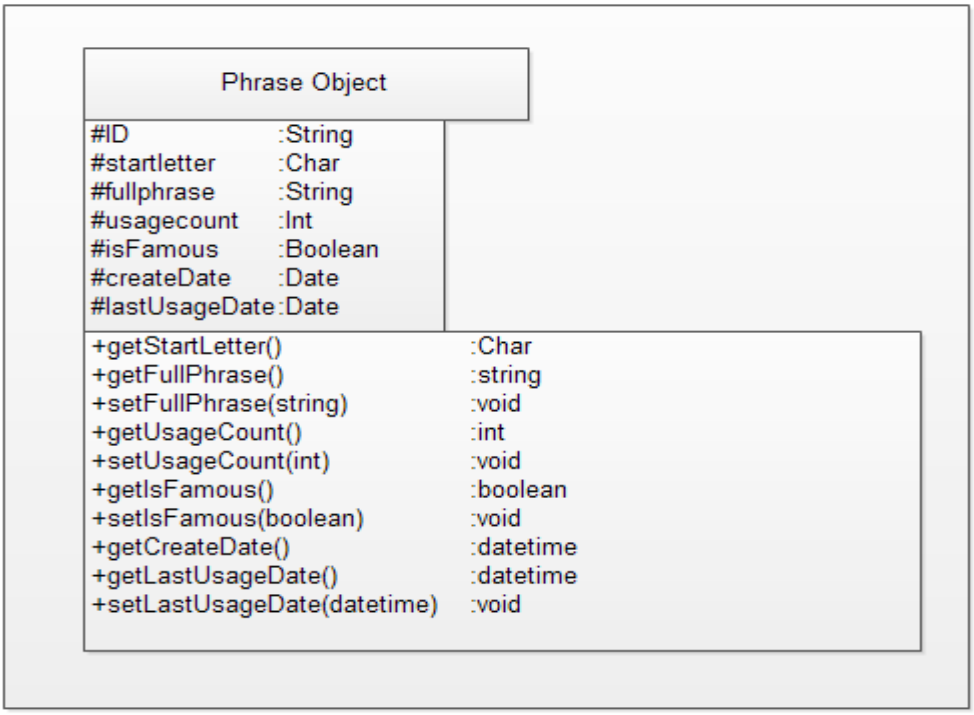


Figure -3-Phrase Class Representation

- getStartLetter() :It returns startletter field of class.
- getFullPhrase() :It returns fullphrase field of class.
- setFullPhrase(string) :It gets an element and sets it to fullphrase field of class.
- getUsageCount() :It returns the usageCount field of class.
- setUsageCount(int) :It gets an element and sets it to usageCount field of class.
- getIsFamous() :It returns isFamous element of class.
- setIsFamous(boolean) : It gets an element and sets it to isFamous element of class.
- getCreateDate() :It returns createDate field of class.
- getLastUsageDate() :It returns lastUsageDate field of class.
- setLastUsageDate(datetime) : It gets an element and sets it to lastUsageDate field of class.

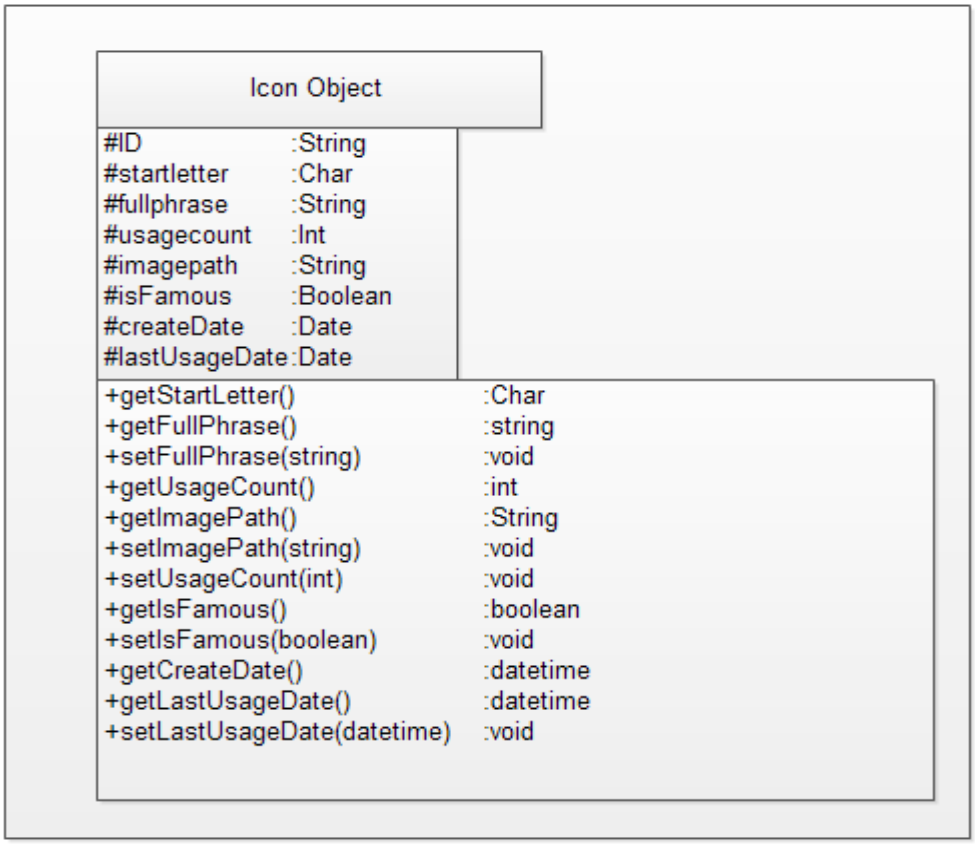


Figure -4- Icon Class Representation

- getStartLetter() :It returns startletter field of class.
- getFullPhrase() :It returns fullphrase field of class.
- setFullPhrase(string) :It gets an element and sets it to fullphrase field of class.
- getUsageCount() :It returns the usageCount field of class.
- setUsageCount(int) :It gets an element and sets it to usageCount field of class.
- getIsFamous() :It returns isFamous element of class.
- setIsFamous(boolean) : It gets an element and sets it to isFamous element of class.
- getCreateDate() :It returns createDate field of class.
- getLastUsageDate() :It returns lastUsageDate field of class.
- setLastUsageDate(datetime) :It gets an element and sets it to lastUsageDate field of class.
- getImagePath() :It returns imagePath field of class.
- setImagePath(string) : It gets an element and sets it to imagePath field of class.

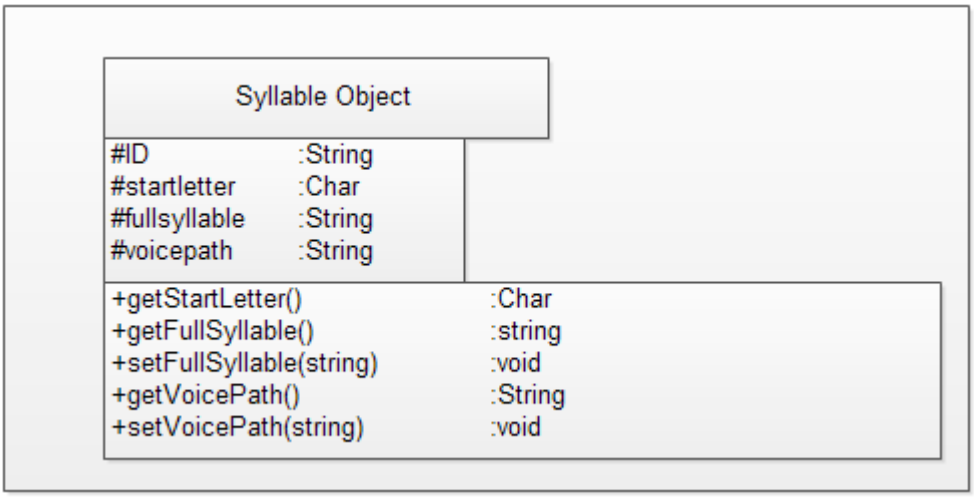


Figure -5- Syllable Class Representation

- getStartLetter() :It returns startLetter field of class.
- getFullSyllable() :It returns fullSyllable field of class.
- setFullSyllable (string) :It gets an element and sets it to fullSyllable field of class.
- getVoicePath() :It returns voicePath field of class.
- setVoicePath(string) : It gets an element and sets it to voicePath field of class.

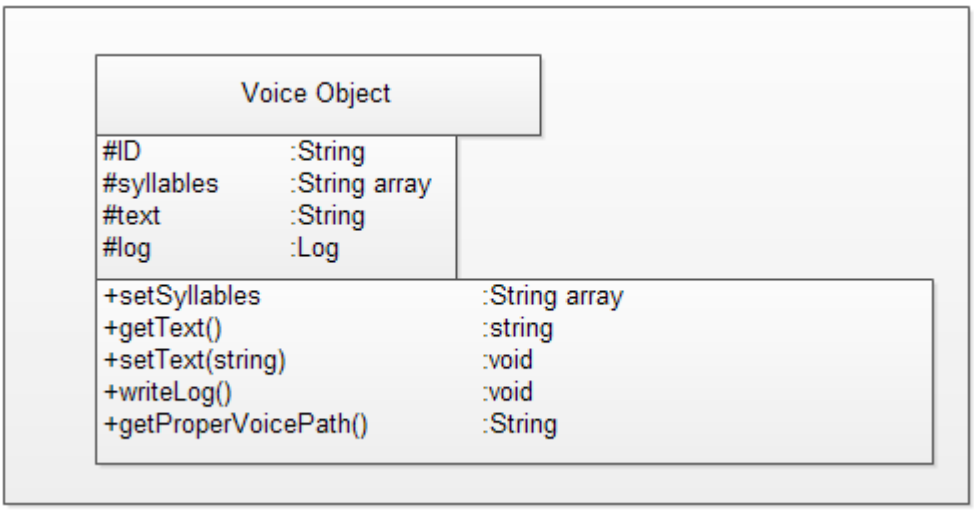


Figure -6- Voice Class Representation

- setSyllables(String array) : It gets an element and sets it to syllables field of class.
- getText() :It returns text field of class.
- setText (string) :It gets an element and sets it to text field of class.
- writeLog() :It writes log object to text files.
- getProperVoicePath(string) :It return a proper voice path by combining id and text values.

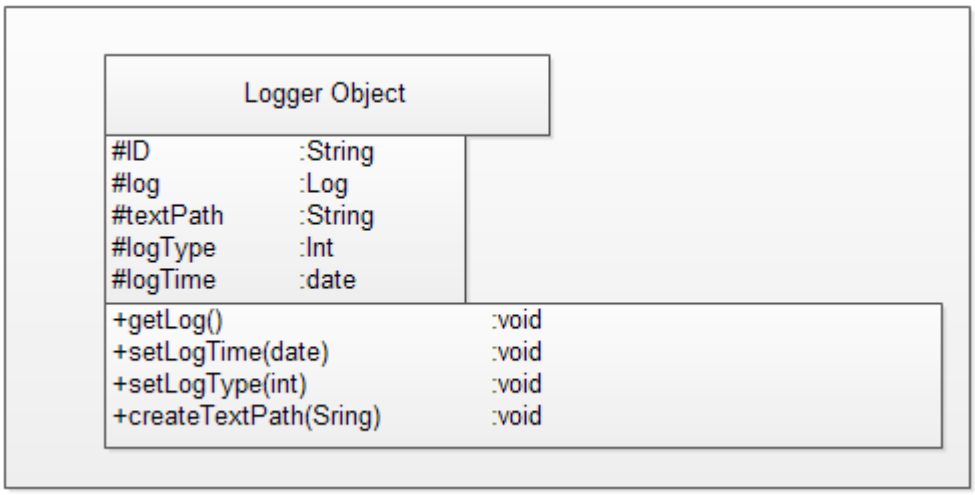


Figure -7- Logger Class Representation

- getLog() :It returns log field of class
- setLogTime(date) :It gets an element and sets it to logTime field of class.
- setLogType (string) :It gets an element and sets it to logType field of class.
- createTextPath() :It gets a string and combines ot with id field and sets that value to textPath field.

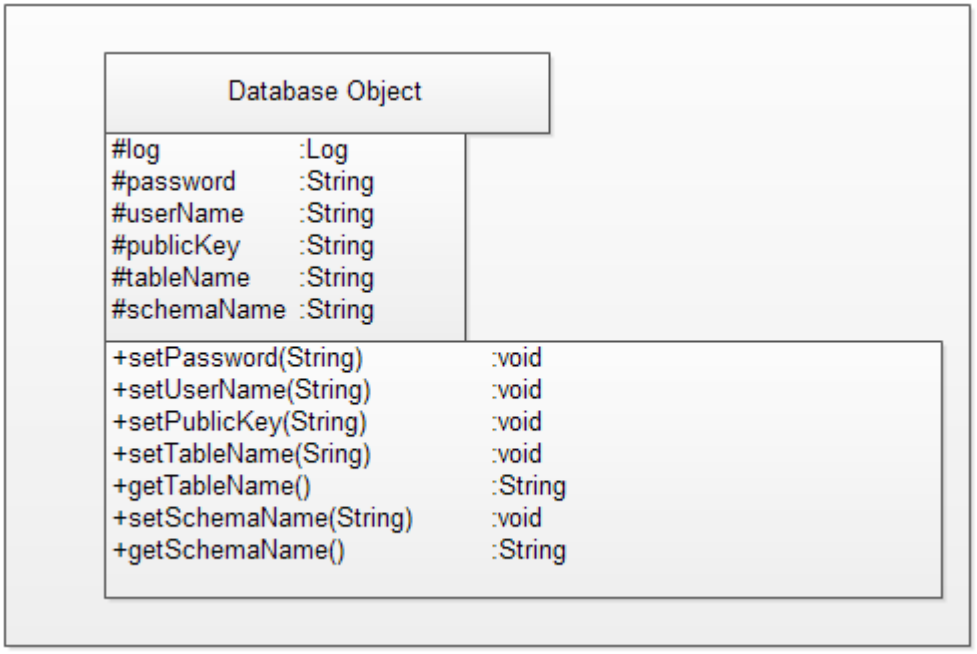


Figure -8- Database Class Representation

- setPassword() : It gets an element and sets it to password field of class.
- setUserName(date) :It gets an element and sets it to userName field of class.
- setPublicKey (string) :It gets an element and sets it to publicKey field of class.

- setTableName() :It gets an element and sets it to tableName field of class.
- getTableName() :It returns the tableName field of class.
- setSchemaName() : It gets an element and sets it to schemaName field of class.
- getSchemaName() :It returns schemaName field of class.

4.2. Data Dictionary

--Phrase Class data

DATA	TYPE	DESCRIPTION
ID	String	To specify the every object created from this class, id is a separator.
startLetter	Char	The start letter of phrase to simplify search manner.
fullPhrase	String	Whole phrase in order to send to the last module
usageCount	Int	It keeps how many times the phrase is used.
isFamous	Boolean	It keeps whether the phrase is assigned as famous or not.
createDate	Date	It keeps the creation date of the phrase.
lastUsageDate	Date	It keeps the last usage date of the phrase

--Icons Class Data

DATA	TYPE	DESCRIPTION
ID	String	To specify the every object created from this class, id is a separator.
startLetter	Char	The start letter of phrase of icon to simplify search manner.
fullPhrase	String	Whole phrase of icon in order to send to the last module
usageCount	Int	It keeps how many times the icon is used.
isFamous	Boolean	It keeps whether the icon is assigned as famous or not.
createDate	Date	It keeps the creation date of the icon.
lastUsageDate	Date	It keeps the last usage date of the icon
imagePath	String	It keeps the the path where the object's image is stored.

--Syllable Class Data

DATA	TYPE	DESCRIPTION
ID	String	To specify the every object created from this class, id is a separator.
startLetter	Char	The start letter of syllable to simplify search manner.
fullSyllable	String	The whole syllable
voicePath	String	The path in system where the mapped voice is stored.

--Logger Class Data

DATA	TYPE	DESCRIPTION
ID	String	To specify the every object created from this class, id is a separator.
log	Log	It is a log field of class. It is stored to see when it is needed.
textPath	String	A path where the log values is stored.
logType	Int	The type of log. There are lots of types to increase log affect.
logText	String	The text written in log. It is designed according to the developer needs.

--Voice Class Data

DATA	TYPE	DESCRIPTION
ID	String	To specify the every object created from this class, id is a separator.
Syllables	Char	Every syllable of the text is kept in this field.
Text	String	Whole text of voice.
Log	Log	Log object of class for writing them to texts.

--Database Class Data

DATA	TYPE	DESCRIPTION
log	Log	To specify the every object created from this class, id is a separator.
password	String	It stores password of database.
username	String	It stores username of database.
publicKey	String	It stores the publicKey of database.
tableName	String	It keeps the proper tableName of database.
schemaName	String	It keeps the proper schemaName of database.

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.

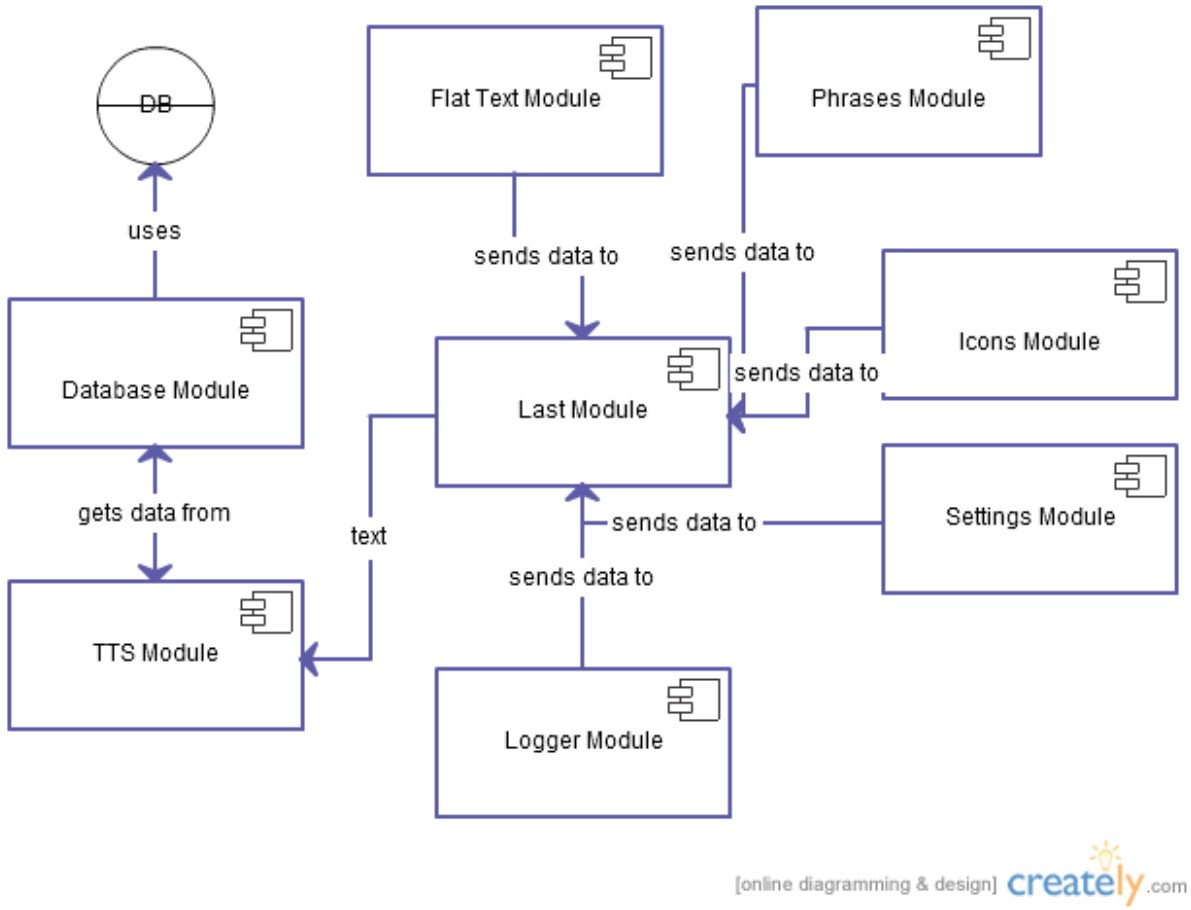


Figure -9- Component Diagram

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.

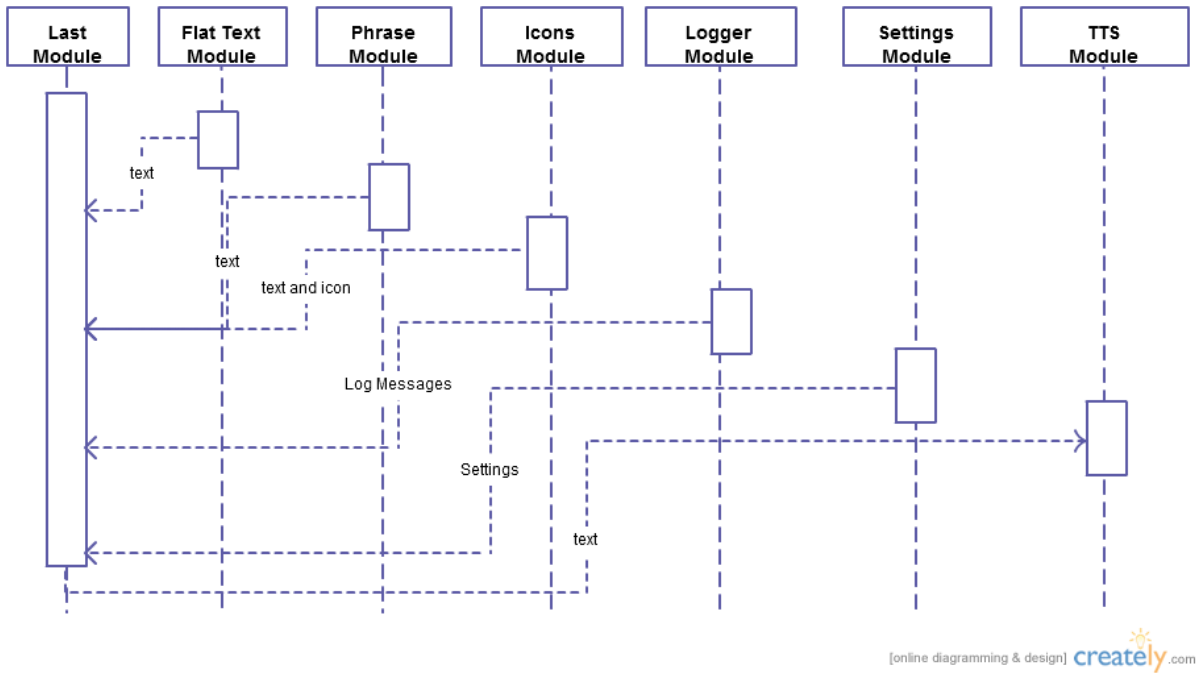


Figure -10- Last Module Sequence Diagram

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.

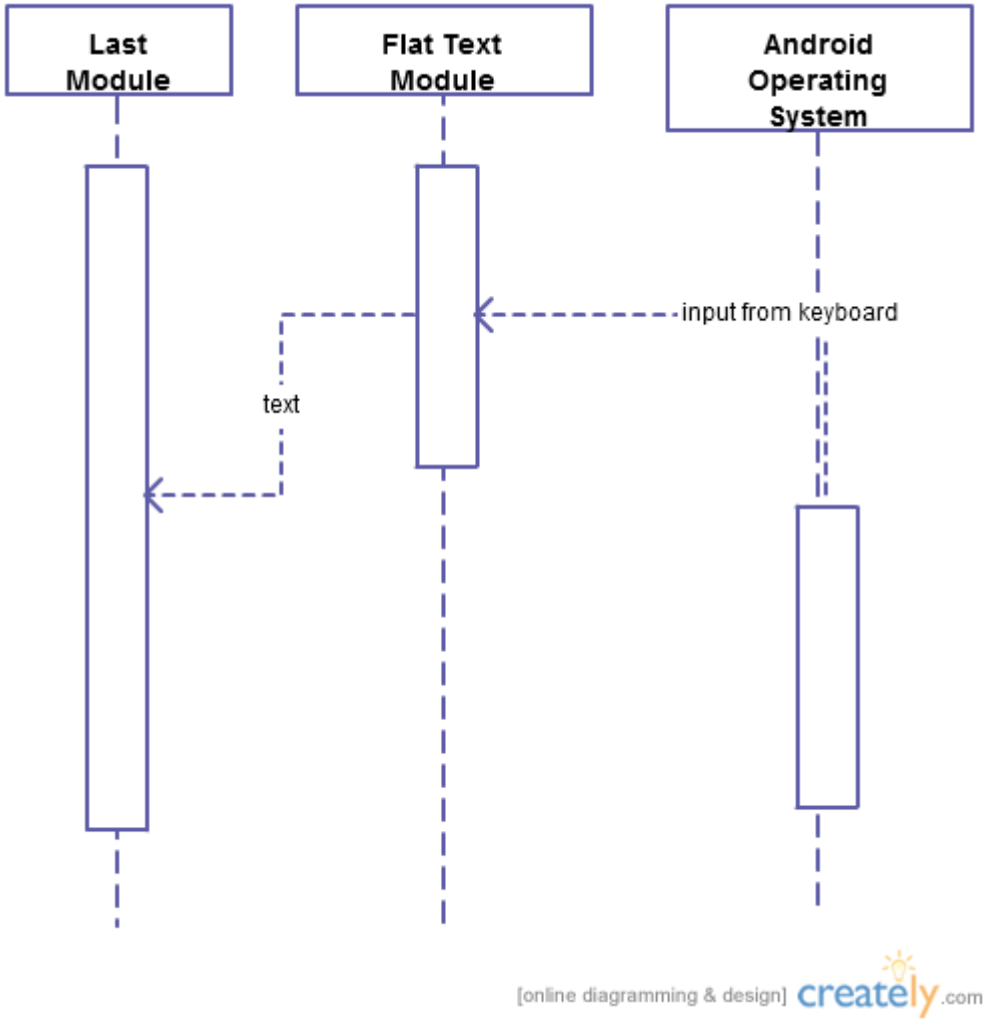


Figure -11- Flat Text Module Sequence Diagram

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

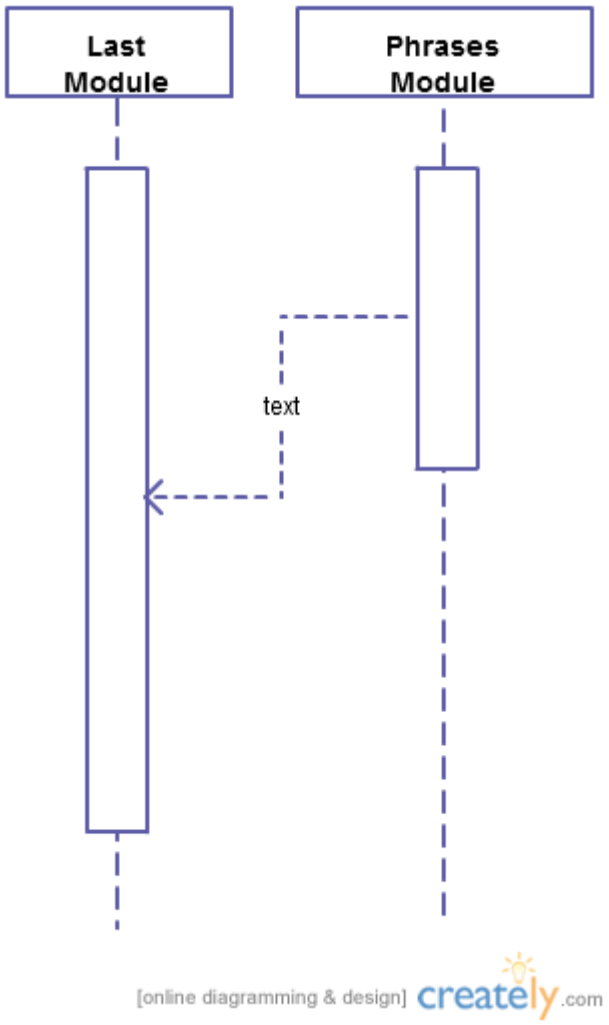


Figure -12- Phrases Module Sequence Diagram

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

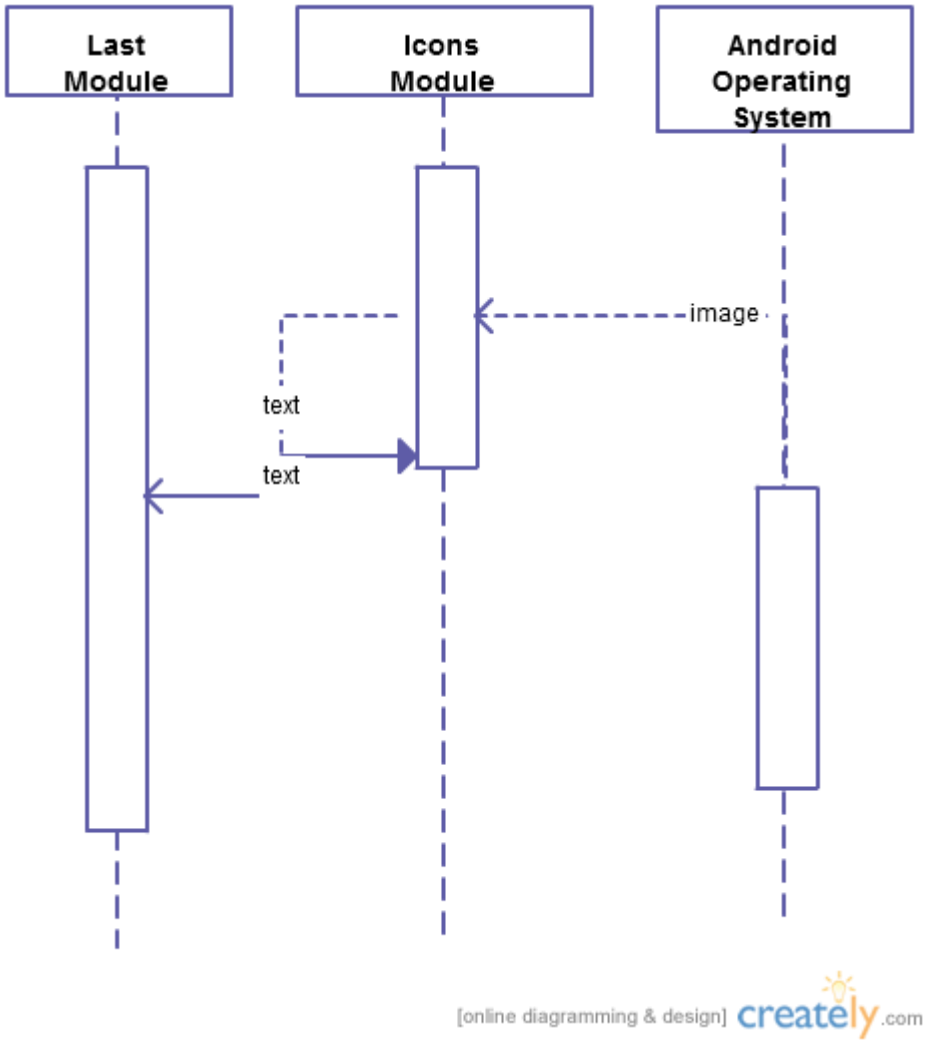


Figure -13- Icons Module Sequence Diagram

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

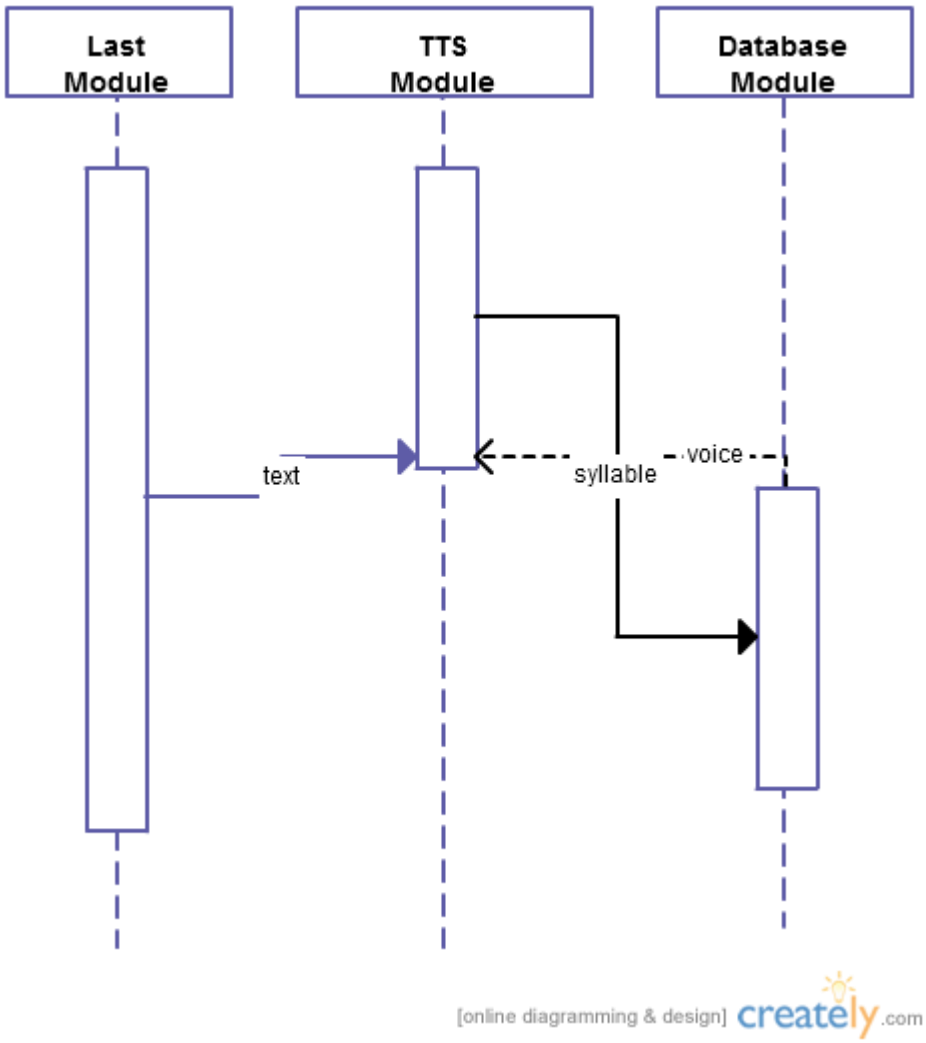


Figure -14- TTS Module Sequence Diagram

5.2.7. Logger Module

5.2.7.1. Processing narrative for Logger Module

Logger Module is responsible from 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

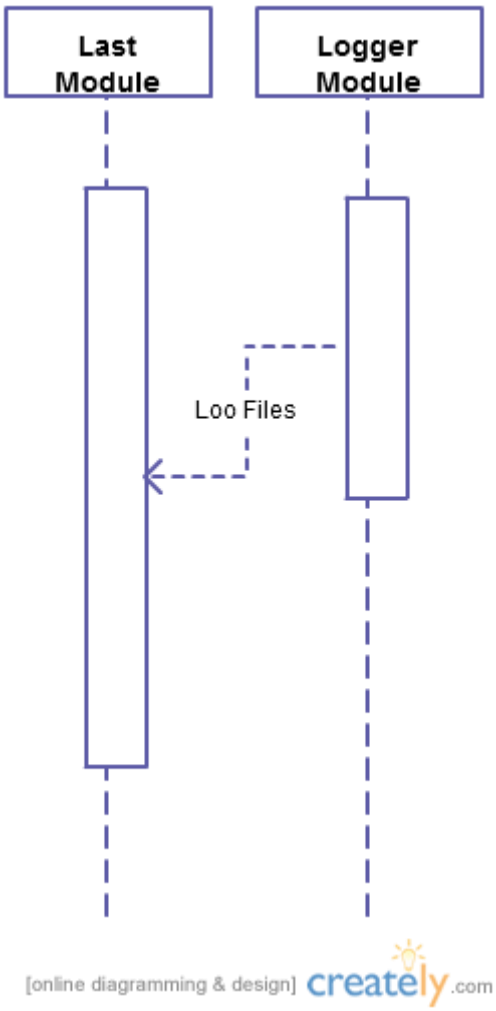


Figure -15- Logger Module Sequence Diagram

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

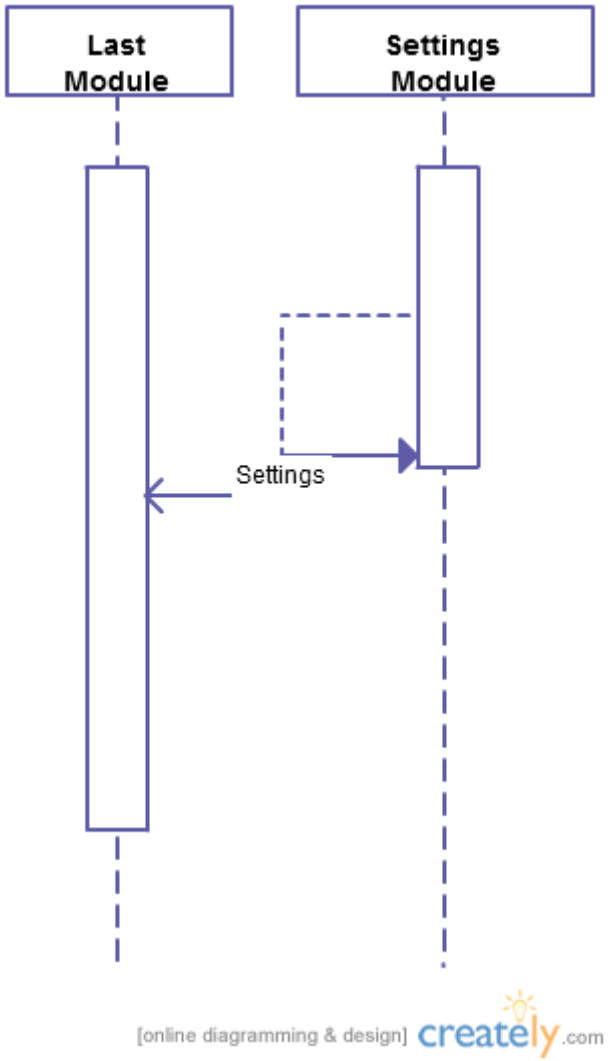


Figure -16- Settings Module Sequence Diagram

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.

5.4. Traceability of requirements

MODUL	DESIGN ELEMENT	DESCRIPTION
Flat Text Module	Voice Object	The flat text module is just for writing flat text. Voice object takes this string.
Phrases Module	Phrases Object, Voice Object	Phrases object extracts the text of phrase and sends it to voice object.
Icons Module	Icons Object, Voice Object	Icons object extracts the text of icon and sends it to voice object.
Settings Module	Database Object	Settings module just keeps user's settings in the database by using database object.
Logger Module	Logger Object	It is exactly matched with object. This module is consisting of the logger class.
Database Module	Database Object	Database object is for configuration of the database module.
Text-To-Speech Module	Syllable Object, Database Object	Syllables and its voices are the objects of text-to-speech module.

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.

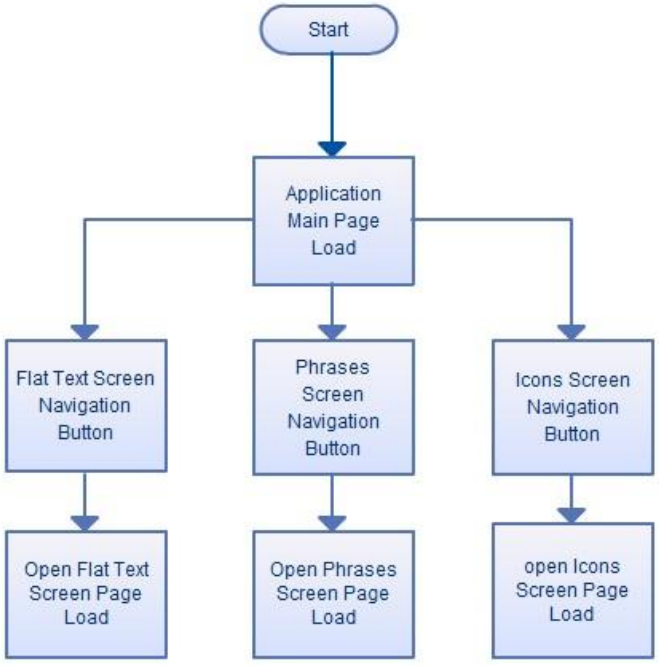


Figure -17- Main page flow

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.

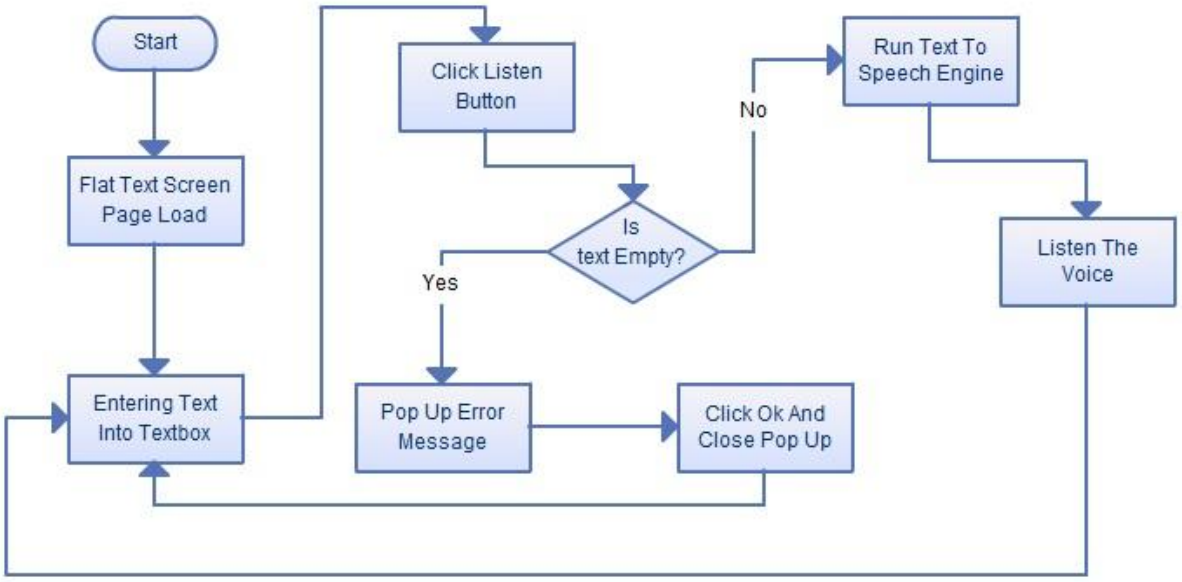


Figure -18-Flat text screen

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.

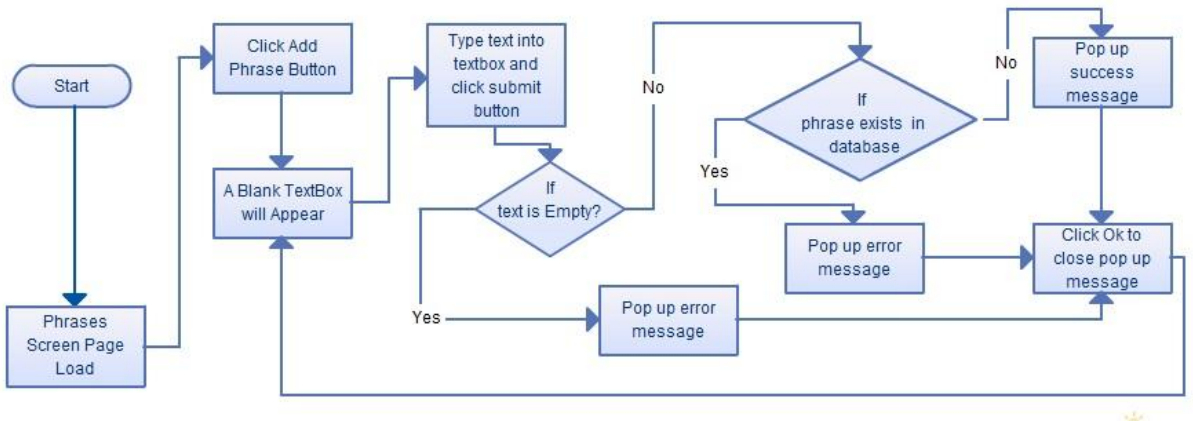


Figure -19- Add phrase flow of phrases screen

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.

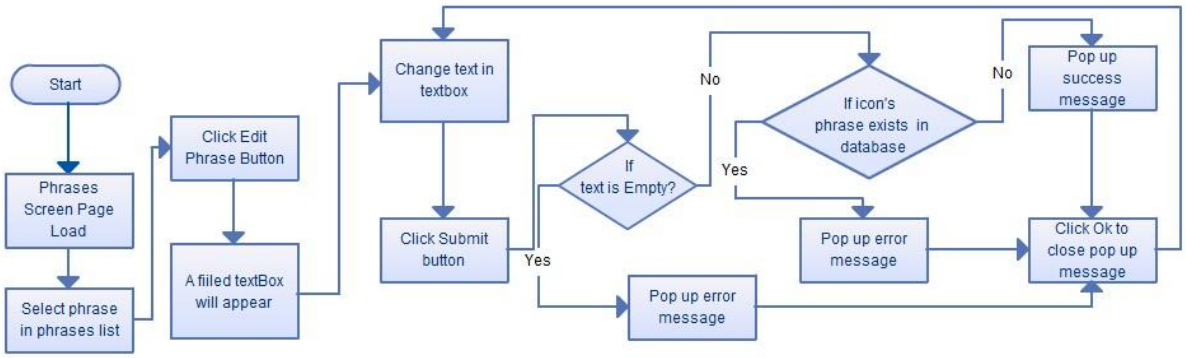


Figure -20- Edit phrase flow of phrases screen

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.

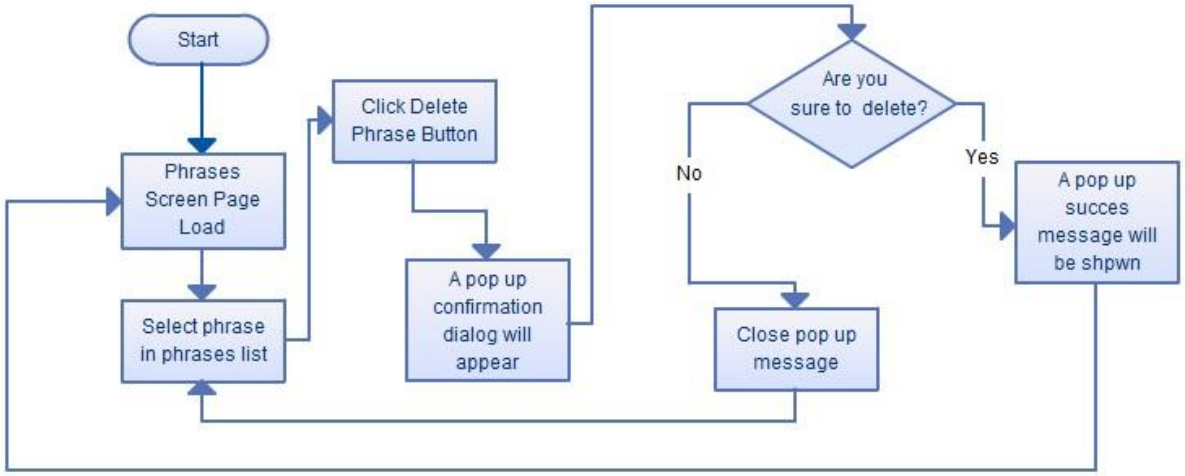


Figure -21- Delete phrase flow of phrases screen

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.

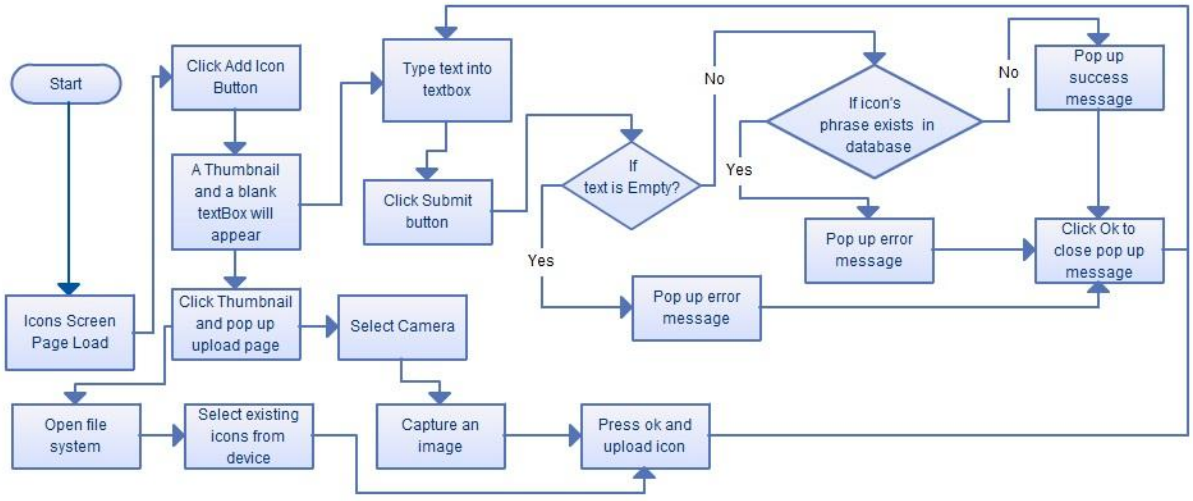


Figure -22- Add icon flow of icon screen

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.

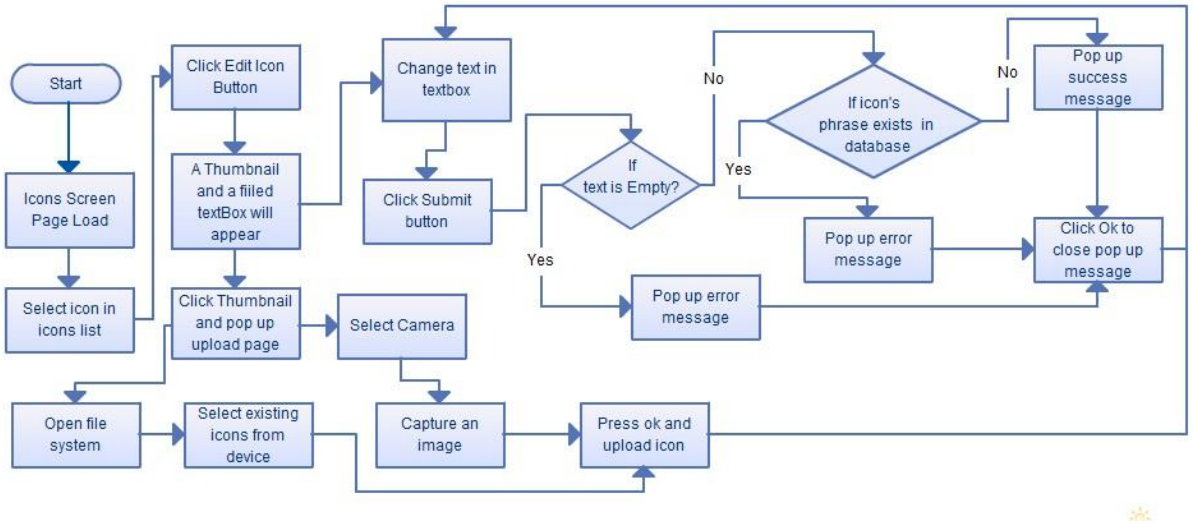


Figure -23- Edit icon flow of icon screen

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.

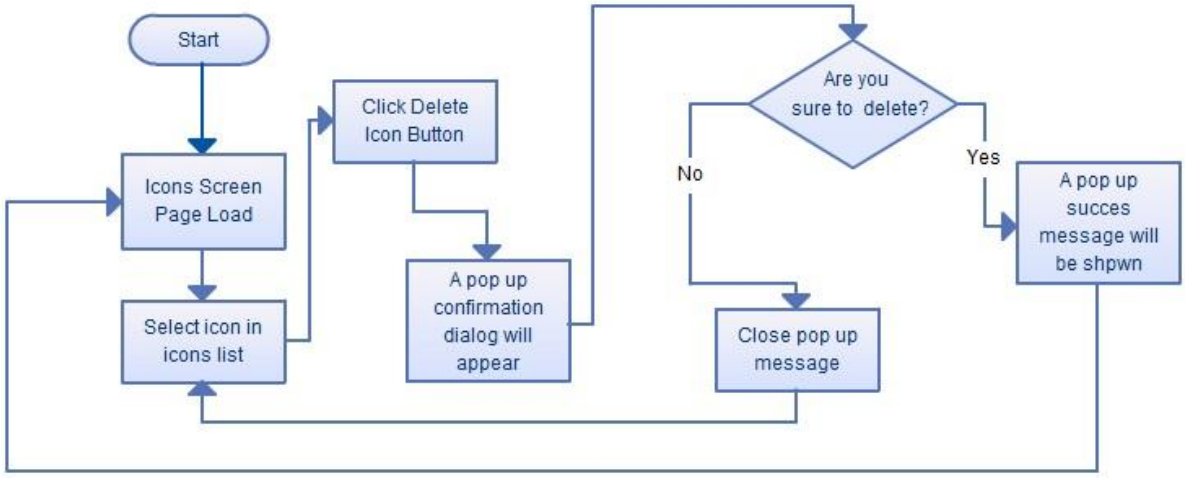


Figure -24- Delete icon flow of icon screen

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.

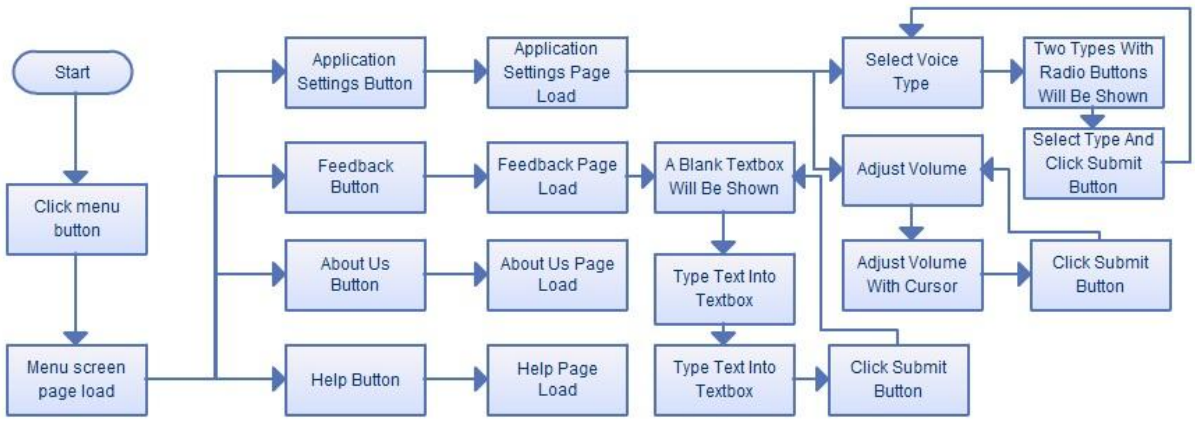


Figure -25- Menu flow

6.2. Screen Images

6.2.1. Flat Text Screen

Figure -26- Screenshot of Flat Text Screen

6.2.2. Phrases Screen

Figure -27- Screenshot of Phrases Screen #1

Figure -28- Screenshot of Phrases Screen #2

6.2.3. Icons Screen

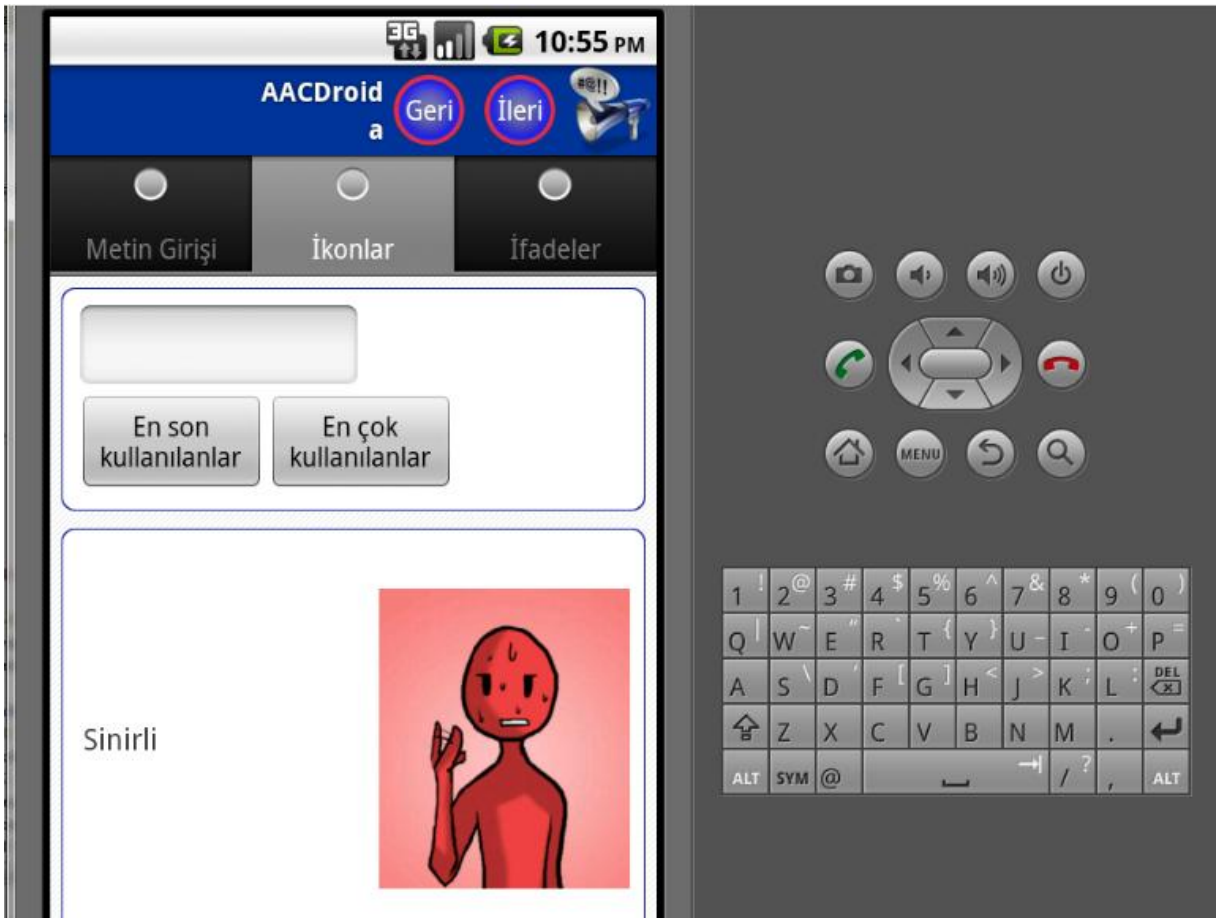


Figure -29- Screenshot of Icons Screen #1

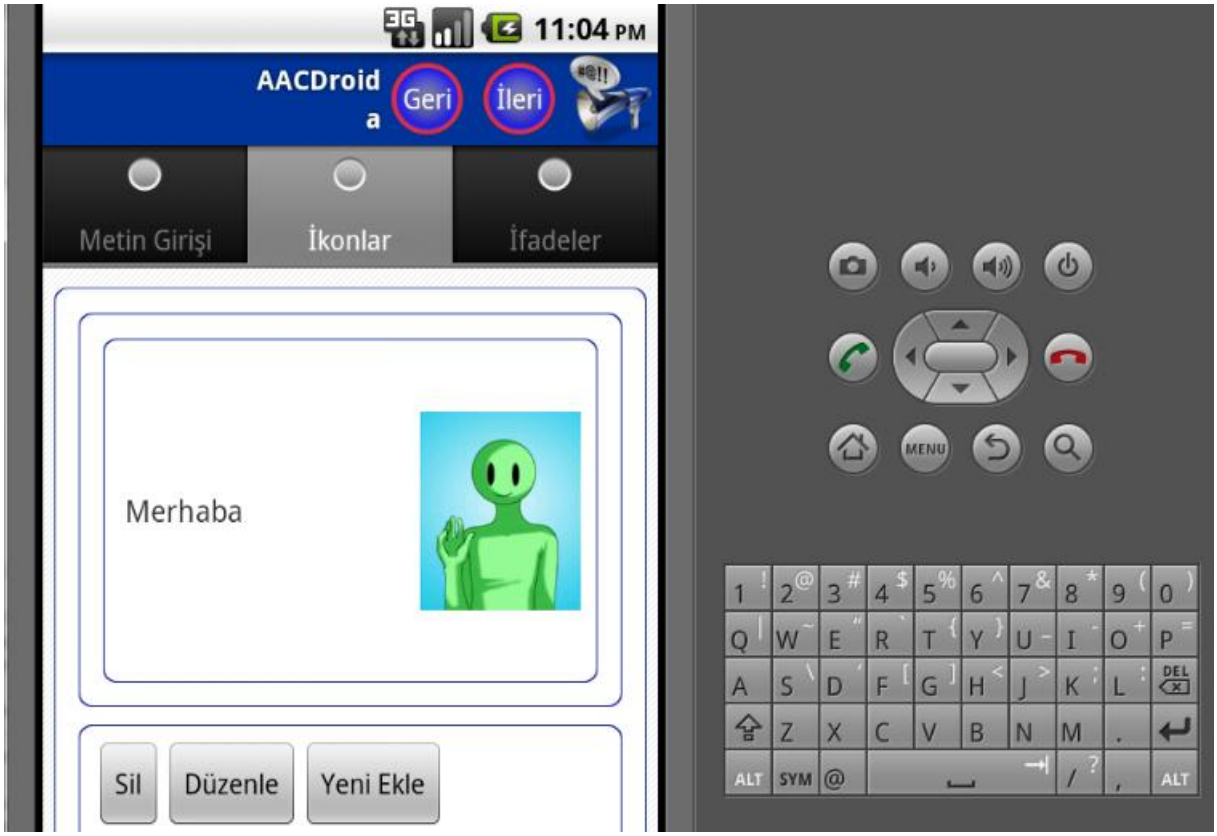


Figure -30- Screenshot of Icons Screen #2

6.3. Screen Objects and Actions

On all of the screens, there will be fragments tab pager object which enables user to select the page view. One can navigate flat text screen, icons screen and phrases screen anywhere. Also there is a button at the right top of all screens which enables user listen texts. All other objects are specified in 6.1 in detail.

7. Detailed Design

7.1. Last Module

7.1.1. Classification

Last Module is a module of the whole system.

7.1.2. Definition

Last module is a parent module enables communication between other modules. Namely, it gets data from Flat Text Module, Phrases Module, Icons Module, Settings Module and Logger Module then sends them to TTS Module or Database Module. In addition to that, module collects data from TTS Module or Database Module to send related module.

7.1.3. Responsibilities

This module has two responsibilities. It provides communication whole system between TTS Module and Database Module. First of all, this module sends all the text data to be converted sound to TTS module then gets equivalent the sound data from TTS Module. These sound data are distributed to related module. Secondly, this module provides communication between whole system and Database Module. Indeed, it gives us to chance to reach icons and phrases database. Then we get data from Database or send it to store.

7.1.4. Constraints

There is only one constraint of this module. It is about the timing of the system. Since main role of this component is providing communication between whole system. It should be handle getting data from one or more module at a reasonable time or sending data to one or more module at a reasonable.

Since this project is an augmented reality project, it responses in at most 2 seconds. Considering TTS and database interactions processes time, this module should be operate in less than 1 second.

7.1.5. Composition

There are two subcomponent of this module. Data Flow subcomponent is responsible for communication between this module and others. Database Handler is responsible from database connection.

7.1.6. Uses and Interactions

Last Module hold whole system together; therefore, Last Module has interaction with all other modules. Last Module is used by other modules to reach TTS and Database. None of the modules can directly arrive TTS module they should use Last Module for this purpose. Data Flow subcomponent performs this communication. In addition to that no modules can directly reach Database Module except from TTS Module. All database issues achieved on Last Module Database Handler subcomponent perform this action.

7.1.7. Resources

Predefined connection rules are needed to construct the communication protocols for the whole system. These rules are used by DataFlow subcomponent.

DatabaseHandler subcomponent uses SQLite. But there is no header file to integrate SQLite with proejct since android will handle it.

7.1.8. Processing

In processing, since this module is responsible for communication with other modules , data flow is the main process on this module. Connecting the database will be achieved by means of SQLite by making queries.

7.1.9. Interface/Exports

The set of services:

- Gets text objects

- Gets icon objects
- Gets phrases objects
- Sends text objects to TTS
- Sends icon an phrases objects to Database
- Provides the connection between other components and database

7.2. Flat Text Module

7.2.1. Classification

Flat Text module is a module of the whole system.

7.2.2. Definition

Flat Text Module gets text input from user by typing text through the keyboard of android.User can manage this module instead he requires.

7.2.3. Responsibilities

Flat Text module is responsible from getting text input from user.For this manner the system uses the keyboard of android and its features such as Swype manner.

7.2.4. Constraints

There is only one constraint of this module. It is about the timing of the system. Since main role of this component is providing communication between user and last module. It should be handle getting data from user at a reasonable time and sending data to last module at a reasonable and vice versa.Since this project augmented reality project, it responses in at most 2 seconds. Considering TTS and database interactions processes time, this module should be operate in less than 1 second.

7.2.5. Composition

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.Flat Text module has no other subcomponent.

7.2.6. Uses and Interactions

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

7.2.7. Resources

Flat Text module directly interacts with user who may types texts through the keyboard of android device.

7.2.8. Processing

In processing, since this module provides direct interaction between user and Last module , the module must get input as text from user and deliver the data to the Last module.

7.2.9. Interface/Exports

The set of services:

- Gets text input from user
- Delivers data to the Last module
- Provides the connection between user and Last module

7.3. Phrases Module

7.3.1. Classification

Phrases module is a sub-module of the system.

7.3.2. Definition

Phrases module is a sub-module which can communicate with last module. This module enables user input to get and send the last module. One can select phrases objects from a certain list then send to the last module to execute in text to speech engine. The certain list can be in two ways; once is frequently used and the other one is favorite user saved list. User can also add new phrase, edit existing ones or delete unwanted ones.

7.3.3. Responsibilities

This module has two responsibilities. One of them is operation on phrases and the other one is sending data to last module and getting data from last module.

There are four operations on phrases. These are adding, editing, deleting and searching phrases.

To add new phrases, user enters a new phrase and sends to last module. Last module does the necessary operation of adding new phrase. If conditions are satisfied last module saves a new entry for phrases table.

To edit existing phrases, user selects a phrase and changes and sends to the last module. Last module does the necessary operations on editing existing phrase. If conditions are satisfied last module updates related row in phrases table.

To delete unwanted phrases, user selects a phrase from list and sends deleting request to the last module. Last module does the necessary operations on deleting unwanted phrase. After that if conditions are satisfied, last module finds the related row in phrases table then deletes it.

Phrases can be searched in three ways. There exist two lists which are frequently used and favorite list. If user chooses one of two, the request will be sent to last module and data will be got from related table in database. The other way of search is user defined. User enters a text and sends it to last module. Last module sends a query which is a certain criteria including user defined text and returns the result.

7.3.4. Constraints

There is only one constraint for this module; timing. All operations should be done in a reasonably time since this project is a real time projects. System should be response in at most 2 seconds. Hence, sending phrase object to Last Module, then getting equivalent voice object should not take more than 2 seconds.

Operations on phrases; adding, deleting, editing and searching take a while. For, phrases objects are sent to Last Module, then, they are sent to Database Module. After, updating database according to data came from. However, it should not pass 3 seconds.

7.3.5. Composition

There is only one subcomponent of this module. DataFlow is responsible from sending data to Last Module and getting data from Last Module.

7.3.6. Uses/Interactions

Phrases Module is responsible from phrases operations. DataFlow is enables the interaction between Last Module and Phrases Module. Also it enables interaction between Phrases Module and Database Module or TTS Module indirectly.

7.3.7. Resources

There is no specific library to use in this Module except from Android library.

7.3.8. Processing

There are two main processing on this module. These are operations on phrases and data flow between Last Module.

Phrases operations are adding, deleting, editing and searching. These operations are handled through Last Module. While adding, deleting, editing and searching phrase object is sent to last Module. It is examined where it should transfer. After that, it transferred to Database Module, the operations are done according to process.

Data flow is the responsible from all data flow between Phrases Module and Last Module. Phrases objects are sent to Last Module to send TTS Module. After TTS Module converts phrase objects to sound objects, They are sent back to Phrases module through Last Module.

7.3.9. Interface/Exports

The set of services:

- Sends phrase objects to the Last Module
- Gets icons objects from Last Module
- Gets voices objects from Last Module
- Gets list of phrases from Last Module
- Sends list of phrases to Last Module

7.4. Icons Module

7.4.1. Classification

Icons Module is a sub module of the system.

7.4.2. Definition

Icons module enables to process on icons. Namely, adding new icon, deleting icon, updating icon and listing icon. Also Icons module give chance to keep frequently used icons and favourite icons list.

7.4.3. Responsibilities

This module has two main responsibilities. One of them is operation on icons and the other one is sending icon objects to Last Module and getting icon objects from Last Module.

Icons operations are adding new icon, deleting icon, updating icon and listing icons. First of all, adding new icon is integration a related text to an icon then it saves. While saving, new object sends to Last Module then Last Module transfers it to Database Module. In Database Module, object is inserted to the database. Secondly, deleting icon deletes selected object from database. This can be happen via Last Module. Thirdly, updating icons is composition of adding new icon and deleting icon operations. Last Module also let this operation happen by providing connection between Database Module. Finally, listing icons composed of three sub operations. These are making related icons list, listing frequently used icons and favorite icons. Making related icons list keeps together in a list selected icons. For example, food list includes pasta, cake, toast and etc.. Frequently used icons list keeps most frequently used icons. These objects are selected by self-regulated. Favorite icons list keeps selected icons in a list. All listing operations can be achieved with Last Module since database connection.

7.4.4. Constraints

Timing is the only constraint of this module. Since the AAC-Droid is communication device, it should be give response in reasonable time. Therefore, selecting object and play it should be achieved in at most 1 second. However, adding new icon, deleting icon, updating icon and listing icon may take a while. In spite of everything, it should not be exceed 3 seconds.

7.4.5. Composition

There is only one subcomponent of this module. DataFlow is responsible from sending data to Last Module and getting data from Last Module.

7.4.6. Uses and Interactions

Icons Module is responsible from icons operations. DataFlow is enables the interaction between Last Module and Icons Module. Also it enables interaction between Icons Module and Database Module or TTS Module indirectly.

7.4.7. Resources

Icons module use device camera while adding new icon or updating icon. Indeed, user captures a photo using device camera while program is running, then use it as a icon. Therefore, we can get input from camera. Because of that we should use android hardware camera package. Apart from that, there is no specific library used by this module.

7.4.8. Processing

As it is described earlier, this module is responsible from operation on icons and data flow with Last Module. Adding, deleting, updating and listing operations are done in this module by using database. While adding, deleting, updating and listing, module sends data to Last Module then Last Module connect to Database Module and reach database. After that, database can be changed according to process.

Icons Module sends the text data to convert sound. This process can be handled via Last Module since Last Module provides connection to TTS Module. After the text data sent to the Last Module, they are transferred to the TTS Module. TTS Module convert text to speech then transferred back it to the Last Module. Voices gathered from Last module are sent to Icons module. DataFlow subcomponent of Last Module and Icons Module make this process happen.

7.4.9. Exports

The set of services:

- Sends icons object to the Last Module
- Gets icons object from Last Module
- Gets voices object from Last Module
- Gets captured image from device camera
- Gets list of icons from Last Module
- Sends list of icons to Last Module

7.5. Database Module

7.5.1. Classification

Database Module is a module of the system.

7.5.2. Definition

This module builds a bridge between database management system and last module.Other modules uses database over this module by connection of last module. In other words Database module only communicates with last module. This means that, if any module needs to execute a database operation, it must communicate last module first and then communicate database module. Therefore, whole system uses database from the last module.

7.5.3. Responsibilities

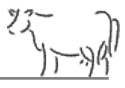
The purpose of this module is to make it possible to communicate the other modules with the database. It manages the receiving and storing information to the database according to the signals coming from the other modules. Database module achieves this purpose by building a bridge between database management system and last module.

7.5.4. Constraints

This module is completely dependent on the other modules. It assumes the last module always transforms the correct information to be kept or to be delivered.

7.5.5. Composition

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.



7.5.6. Uses and Interactions

Model package consists of classes that are used for interacting with the database. They represent the entity structure of the database. When data is requested from the database, related class instances will be created for related database tables and will be filled with the requested information to be used for the requested execution. Last module will be used for communicating with the database, i.e. saving/retrieving/updating/deleting/inserting data.

7.5.7. Resources

Database Module interacts with a SQLite DBMS. SQLite is a powerful, open source object-relational database system. JDBC(The Java Database Connectivity API) will be used to provide an appropriate connection between the system and the database. It is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases, SQL databases and other tabular data sources.

7.5.8. Processing

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.

7.5.9. Interface/Exports

- Gets data from Last module
- Sends icon an phrases objects to Last Module

7.6. Flat Text Module

7.6.1. Classification

Flat Text module is a module of the whole system.

7.6.2. Definition

Flat Text Module gets text input from user by typing text through the keyboard of android. User can manage this module instead he requires.

7.6.3. Responsibilities

Flat Text module is responsible from getting text input from user. For this manner the system uses the keyboard of android and its features such as Swype manner.

7.6.4. Constraints

There is only one constraint of this module. It is about the timing of the system. Since main role of this component is providing communication between user and last module. It should be handle getting data from user at a reasonable time and sending data to last module at a reasonable and vice versa.Since this project augmented reality project, it responses in at most 2 seconds. Considering TTS and database interactions processes time, this module should be operate in less than 1 second.

7.6.5. Composition

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.Flat Text module has no other subcomponent.

7.6.6. Uses and Interactions

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

7.6.7. Resources

Flat Text module directly interacts with user who may types texts through the keyboard of android device.

7.6.8. Processing

In processing, since this module provides direct interaction between user and Last module, the module must get input as text from user and deliver the data to the Last module.

7.6.9. Interface/Exports

The set of services:

- Gets text input from user
- Delivers data to the Last module
- Provides the connection between user and Last module

7.7. Logger Module

7.7.1. Classification

Logger module is sub module of the system.

7.7.2. Definition

Logger module gives a chance to technicians to see the problem of the system by supplying log files to them. By exploring these files, one can see easily which warning message is given a lot. Then it can easily be predicted where the problem is.

7.7.3. Responsibilities

Logger module is responsible from 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.

7.7.4. Constraints

There is one data type for the Logger module. It is from type of String in java. It is for holding error/or warning messages. Moreover there will be a static array of string and integers to hold all types of error/warning messages with priority ids.

7.7.5. Composition

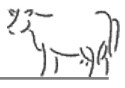
Since there is only one component, it is the main component of the sub module. Logger module has interactions with Last module.

7.7.6. Uses and Interactions

Module gets warning or error message as string. It is the only interaction between Last module and Logger module.

7.7.7. Resources

Since Java file I/O operations are needed, File class of java.io.File library is needed.,



7.7.8. Processing

When message comes to the module, it is checked with the array that is specified before, and according to priority id found in array, it is saved to the file that has present date. If there is no log file for present date, first it is created. After writing operation it is closed.

7.7.9. Interface/Exports

Logger module has only interface with Last module. It gets error/warning message as data type of string. After saving it to the log file, it sends Boolean true value to the Last module.

7.8. Settings Module

7.8.1. Classification

Settings module is sub module of the system.

7.8.2. Definition

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.

7.8.3. Responsibilities

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.

7.8.4. Constraints

There is no special constraint with this module.

7.8.5. Composition

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.

7.8.6. Uses and Interactions

Settings module has an interaction with Last module. This module sends data to Last module.

7.8.7. Resources

For the user setting storage this module uses the database of the system. In the system SQLite database is used. Since there will not be lots of users settings whole value about this settings will be stored in just one table.

7.8.8. Processing

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.

7.8.9. Interface/Exports

Settings module has only interface with Last module. It gets user personal preferences .

8. Time Planning (Gannt Chart)

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

8.1. Term 1

AACDroid First Term Gantt Chart

Page 1 of 1

Start Date	End Date	TASK
1.10.2011	15.12.2011	Making research and brainstroming
1.10.2011	15.1.2012	Learning android enviroment and programming with java
12.11.2011	1.1.2012	Designing solution
15.12.2011	31.1.2012	Implementing graphical user interface
15.11.2011	31.1.2012	Implementing part of user logics

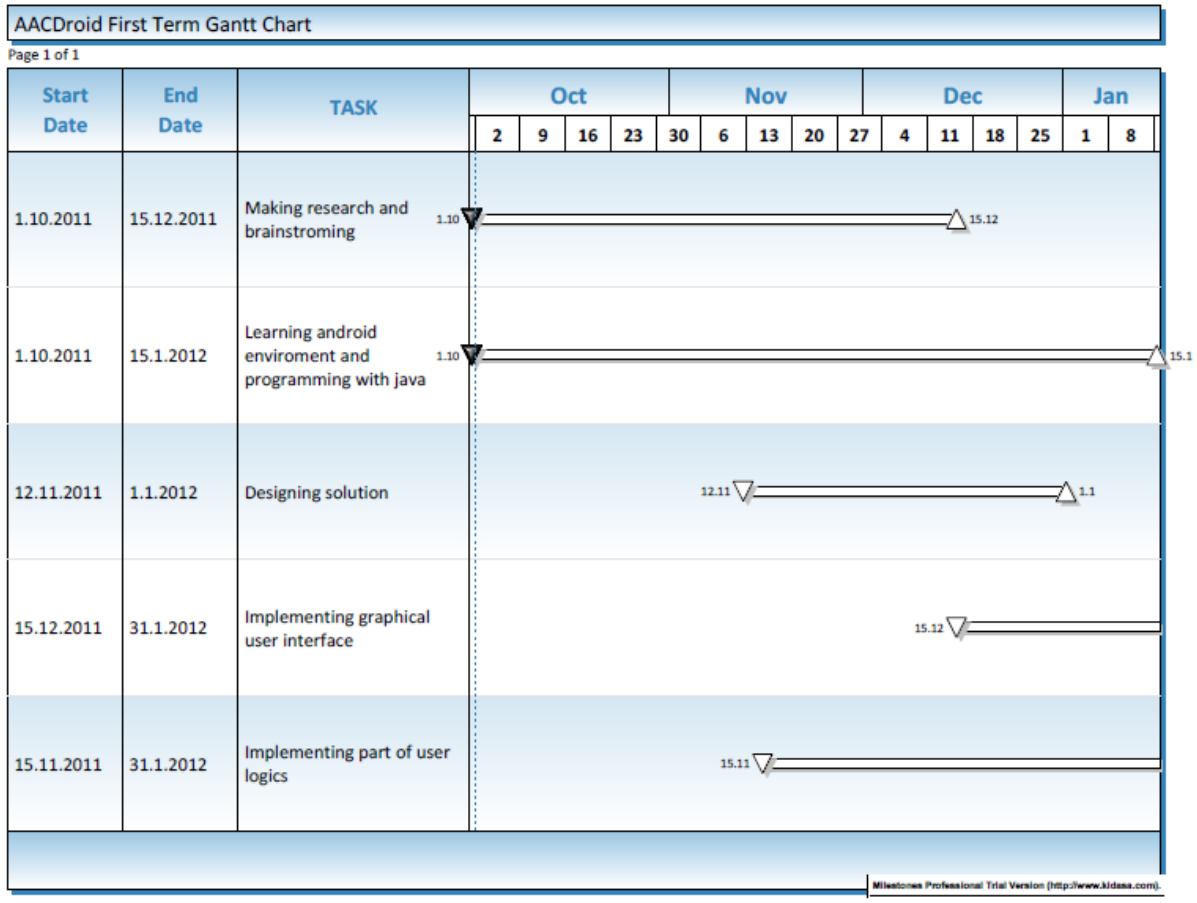


Figure -31- Gannt Chart for Term 1

8.2. Term 2

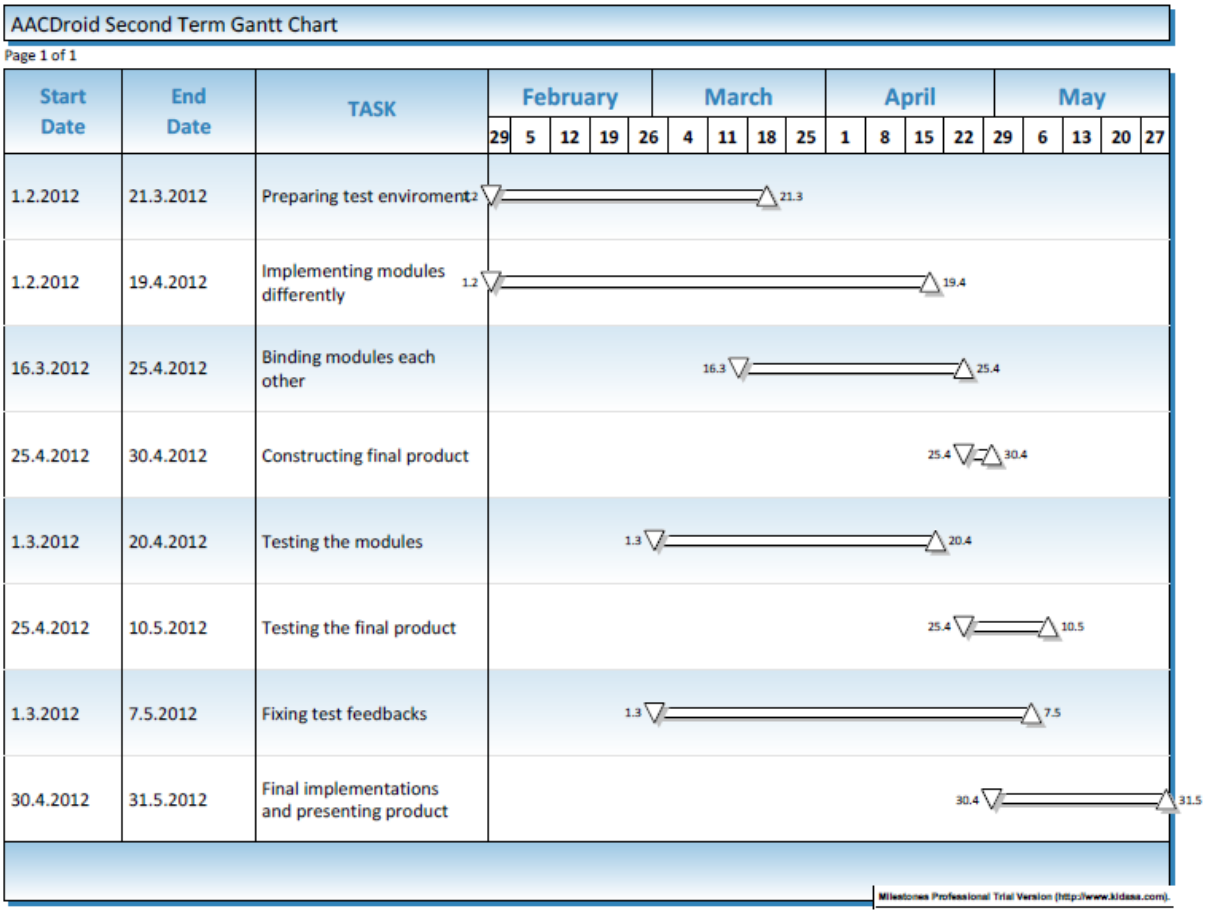


Figure -32- Gannt Chart for Term 2

9. Libraries and Tools

Android Library: It is a library formed by contribution of Google, Android environment and all the developers of Android. In order to develop applications and use the device's properties this library provides main paths. Just by adding this library to the java development kit the developer can use all the features of Android.

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 the 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.

10. Conclusion

In conclusion, the definition, purpose and scope of the project are given in the Detail 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. As the most important part, in this document, detailed design is introduced. In this manner the implementation of the product is clearly described.