A Suite of Mobile Application based on Red5 Server

By

Aleena Khan

Muneeb Mohsin

Waseem Anjum Munir

Submitted to the Faculty of Computer Software Engineering Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfilment for the requirements of a B.E Degree in Computer Software Engineering

June 2012

ABSTRACT

Latest technology trends show that there has been an immense growth in the smart phone mobile industry. Numerous applications have taken over the smartphone's app markets that are capable of streaming media files effectively. Since this technology is new therefore there is immense space of innovations and creativity that can be applied to develop fascinating and useful applications.

There are quite a few streaming media servers available that provide efficient streaming of audio/video via smart phones as client. The Project undertaken is based on the development of a transmission Engine Using Red5 Server to stream media contents. The Red5 Server is a new Server in market. Not much experimentation has been done using it. Facebook has only recently used it for chatting applications. The project's objective is to demonstrate the functionality of the transmission Engine via creative applications not just for smartphones but for web too.
The Project's output will be ready to use Applications for deployment and use locally and ant Corporate sector.

CERTIFICATE OF CORRECTNESS AND APPROVAL

Certified that work contained in this project thesis "A SUITE OF MOBILE APPLICATION BASED ON RED5 SERVER" carried out by Aleena Khan, Muneeb Mohsin and Waseem Anjum under the supervision of Asst. Prof. Dr. Hammad Afzal for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

APPROVED BY

____

(SUPERVISOR NAME)

_____ DEPARTMENT

MCS

DATED: ________________

ACKNOWLEDGEMENTS

"ALL AND EVERY KIND OF PRAISE TO BE FOR ALMIGHTY ALLAH, WHO GUIDES US FROM DARKNESS TO LIGHT AND HELP US IN DIFFICULTIES"

First of all, we are grateful to Almighty Allah, His Kindness and Compassion that he made us able to complete this project. We pay special thanks to our teacher and supervisor Dr. Hammad Afzal for providing us guidance and specially the encouragement and moral support. We will like to express our gratitude for our teachers, who lead us from darkness to light. They were there for us whenever we needed them. Thank you for your support and co-operation without which this project would not have been possible

We are thankful to all of our friends for lively discussions, exchange of ideas and source of encouragement during the entire project. We are also thankful to our parents and family members for their perseverance and continued moral support, which helped us a lot in completing the project.

TABLE OF CONTENTS

Chapter 1: Introduction

1.1. Introduction

The project is based on the creation of a mobile based transmission engine that can record video/audio; stream it live or store it directly to a Red5 server (remotely hosted). The video/audio streamed can have various aspects and applications will be developed for demonstration of the transmission engine.

Latest technology trends show that there has been an immense growth in the smart phone mobile industry. Numerous applications have taken over the Smartphone's app markets that are capable of streaming media files effectively. Since this technology is new therefore there is immense space of innovations and creativity that can be applied to develop fascinating and useful applications.

There are quite a few streaming media servers available that provide efficient streaming of audio/video via smart phones as client. The primary objective of the project is to set up a client and server architecture, where Smart phones primarily with Android OS and a flash media web server Red5 will be used. The architecture set up will be used as an effective engine to facilitate to develop suite of smart phone applications. The applications will be used to demonstrate the working of the client/server architecture that is a new architecture recently introduced for streaming media files.

1.2. Background

Mobile phones are no longer just a tool for communication in our daily lives; they now act as a multi-purpose instrument that helps us with our daily activities. We often tend to spend unnecessary time on daily activities. To aid in this inconvenience in our daily lives, a mobile application could come in very handy for our day to day use. To develop such an application, it is necessary to identify a platform that would have the maximum reach and have a hassle free development.

Smart phones with open operating systems are getting popular with the passage of time. Increased exposure of open source Smartphones also increased the security risk. Android is one of the most popular open source operating system for mobile platforms.

Numerous applications have been made for this platform to facilitate the users of the smart phones. With the introduction of media streaming servers, developers are now innovating applications for the ease of users' to manage their day to day work and activities. One of the most booming fields in android Smart phones are the innovations made in audio video streaming. Various applications have been made and put up in Android Market. In accordance to the project the main purpose is to develop a suite of application to depict the capability of the transmission engine that will be configured and programmed accordingly.

1.3. Problems Addressed

A suite of Mobile Streaming applications offers solutions to make life easier on the move. The Enterprise Dictation application allows the people at corporate level to stay in touch and keep themselves posted of new events on the move. The FaceTime Application offers peer to peer caht using the services of a transmission engine of RED5 server and Speech to text app allows the user to use the converted text on the move in accordance to however they want to use it.

1.4. Goals and Objectives

Latest technology trends show that there has been an immense growth in the smart phone mobile industry. Numerous applications have taken over the Smartphone's app markets that are capable of streaming media files effectively. Since this technology is new therefore there is immense space of innovations and creativity that can be applied to develop fascinating and useful applications.

There are quite a few streaming media servers available that provide efficient streaming of audio/video via Smartphone as client. The primary objective of the project is to set up a client and server architecture, where Smart phones and a flash media web server Red5 will be used. The architecture set up will be used as an effective engine to facilitate to develop suite of smart phone streaming applications. The applications will be used to demonstrate the working of the client/server architecture that is a new architecture recently introduced for streaming media files.

1.5. Deliverables

- Phase I:
    - Requirement Analysis Report
    - Software Requirements Specifications (SRS)

- Phase II:
    - Detail Design document
    - Report on Setting up the transmission Engine
    - Prototype Dictation app
    - Prototype of Facetime app
    - Prototype Voice to Text app
- Phase III:
    - Speech to Text application development client end interface
    - Dictation application development web end module
    - Dictation app module for client end
    - Live chat app module web app
    - Live Chat app module client end
- Phase IV:
    - Review and Validation of modules
    - Implementation and Testing
    - Quality Assurance Document
- Phase V:
    - System Consolidation, Documentation, Training and Production Release.

Chapter 2: Literature Review

2.1. Literature Review

Mobile phones are no longer just a tool for communication in our daily lives; they now act as a multi-purpose instrument that helps us with our daily activities. We often tend to spend unnecessary time on daily activities. To aid in this inconvenience in our daily lives, a mobile application could come in very handy for our day to day use. To develop such an application, it is necessary to identify a platform that would have the maximum reach and have a hassle free development.

Smart phones with open operating systems are getting popular with the passage of time. Increased exposure of open source smart phones also increased the security risk. Android is one of the most popular open source operating system for mobile platforms.

Numerous applications have been made for this platform to facilitate the users of the smart phones. With the introduction of media streaming servers, developers are now innovating applications for the ease of users' to manage their day to day work and activities.

One of the most booming fields in android smart phones is the innovations made in audio video streaming. Various applications have been made and put up in Android Market. With respect to the project under discussion we have got three applications developed.

The first application is "Face Time" it's a live chat application that allows users to chat with their peers either via Smartphone or via web application. The second application is an Enterprise Application namely "Dictation app" this application allows users to publish video messages on a web interface maintained by the enterprise to keep the employees updated.

As far as the third voice to text application is concerned we have got varied applications available the most famous and highly rated application is "hand cent sms" that provides with the voice to English text functionality, other such applications include "chomp sms" and "go sms" etc. The application developed in this project will be converting the voice messages into English text to be used as desired by the users.

2.1.1. Introduction to Android

2.1.1.1 Android Framework

Android is a mobile operating system running on a Linux Kernel. It is a runtime environment released by Google which has a Java language based middleware. Android uses the Linux drivers, file system, memory management and process model. Android is widely seen as a worthy competitor to the iPhone and thus it has been adopted whole heatedly by mobile handset manufacturers such as HTC, Motorola, Samsung, LG and Dell.

Android architecture mainly consists of the Linux Kernel, Libraries, Android Runtime, Application Framework and lastly the native android applications.

Every Android application runs its own process, with its own instance of the Dalvik Virtual Machine.

Android includes a set of C/C++ libraries used by various components of the android system. These capabilities are exposed to the developers through the Android application framework. Android will ship with a set of native applications including an email client, SMS program, maps, contacts, browser and many others.

The Android SDK provides the tools and APIs necessary to begin developing applications that run on Android powered devices. It allows developers to create the application in Java language and control the device via the Java libraries created by Google.

In Android, all applications are treated equally. Therefore a user can create an application that replaces a native application. On the next page is a diagram of the Android framework.

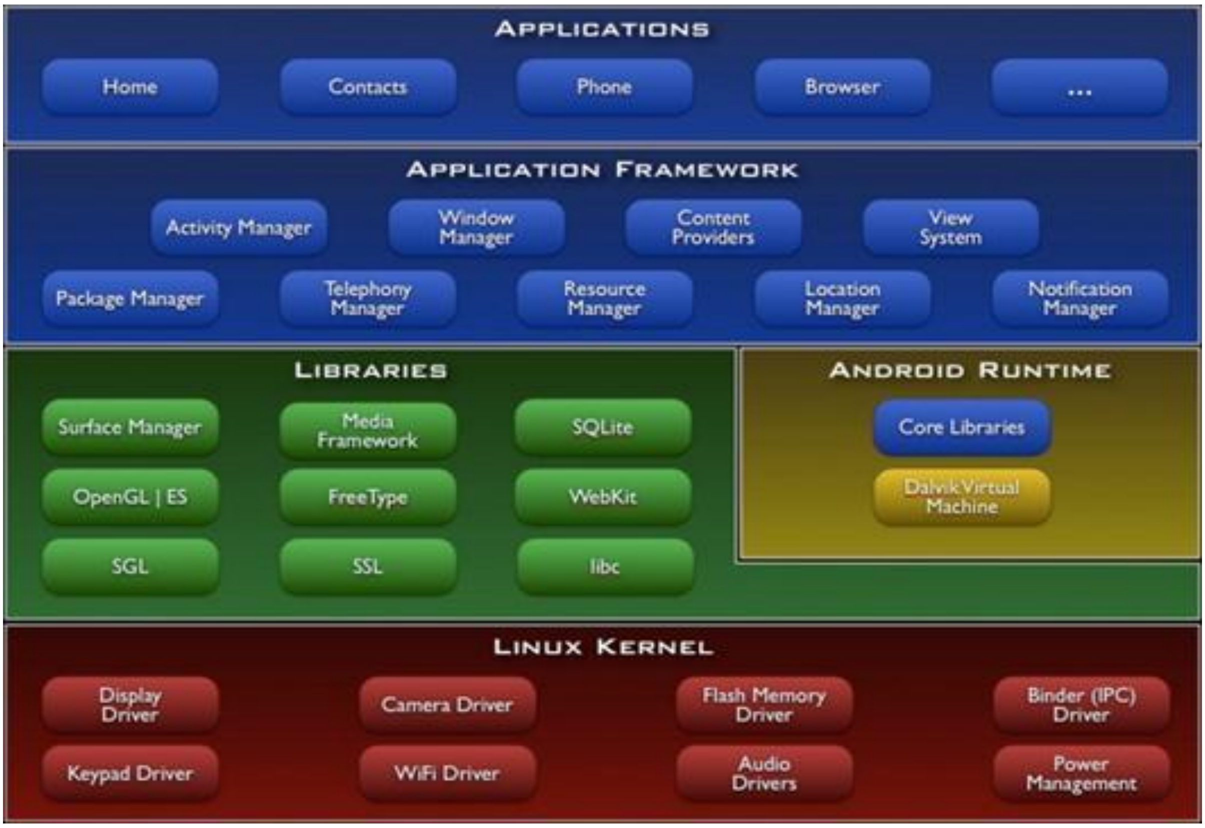


Figure 3 Android Frameworks (1)

2.1.1.2 Android Distribution

Over the last four years, Google has released various Android versions. While majority of the phones on the market have the latest version of the OS loaded, many consumers who have bought an Android phone on contract still have an older version of Android. The charts below show an accurate representation of the distribution of the different versions of OS owned by consumers.

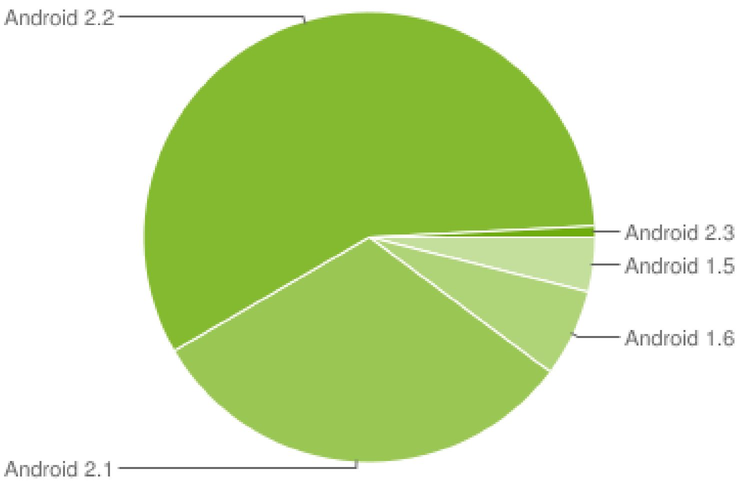


Figure 4 Android Distribution Chart (2)

From the chart above we can observe that there are a larger number of people who have the oldest version of Android (1.5) compared to the latest (2.3).

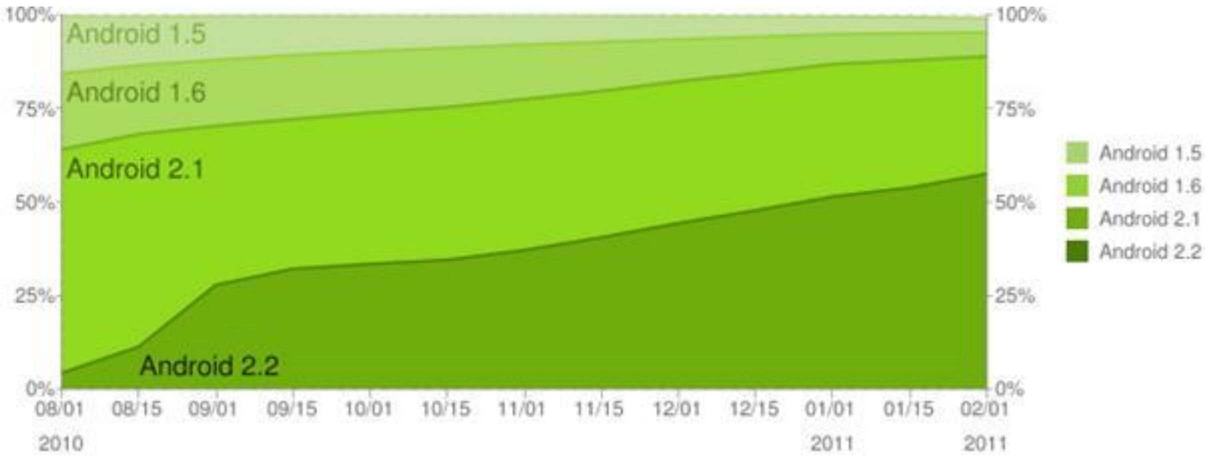


Figure 5 Android Distribution Vs Time (2)

Figure 3 shows us the trend in the distribution of the various versions of Android over the past 1 year. It is seen that although more and more consumers are moving to the latest version of Android there is still a slight percentage holding on to the older versions.

Thus when creating an application, the developer must take into consideration these facts and decide if he wants to cater to all consumers or a certain target market.

2.1.1.3 Android Fundamentals

Activity:

An activity is the most basic building block of an application. An application can typically have more than one activity. A more general description of an activity would be an activity corresponds to a UI screen. When a user switches to another activity, the current activity would be placed in a paused state.

Service:

Service is a background task which executes without a UI. Media Player, Download Managers would be examples of a Service.

Content Provider:

Components which share data, typically abstract access to application database (e.g. Contacts).

Broadcast Receivers:

Broadcast Receivers are codes which are executed for system wide broadcast messages, such as battery low. They are a way of responding to external notifications such as alarms. Applications can create and broadcast their own Intents as well.

Intents:

Intents are asynchronous communication mechanisms between components such as activities.

2.1.2 RED5 Server

Red5 is an open source media server that delivers live video/audio/data to a client application. In most cases that client happens to be the Flash Player. Since the server is licensed under the LGPL, companies have the liberty to use Red5 in proprietary products. The main restriction placed on Red5 is that any modification to the original source code must be donated back to the project. This ensures that the project continues to thrive with patches and helps us deliver a more stable product to the community.

Red5 was conceived with the intention of providing a third party streaming solution that would come without all of the expensive licensing costs of other comparable servers, a few developers including John Grden, Chris Allen and Luke Hubbard got together and started a discussion on what it would take to build something that could be used by the masses. Today, Red5 is used by companies around the world and continues to thrive not only as a streaming server but also as a gaming and application server.

Red5 can be used for a number of different applications. From a high level, the first use of Red5 typically deals with streaming video. However, Red5 can be used for anything from an MMO gaming engine, to a full-fledged application server. Its strengths lie in the core infrastructure. Built using Java and accompanying libraries like MINA's networking library and Spring's inversion of control project, Red5 is prized for its modification and extensibility. Most open source projects don't get to see as much modification, but Red5 has seen many internal changes due to the wide array of uses that our developers have implemented over the past three years. Some main features include:

- Streaming audio/video (flv, h.264, aac, and mp3)
- Recording client streams (flv only)
- Shared objects
- Live stream publishing
- Remoting (amf)

Some of the implementations with include the Big Blue Button project which aims to provide a video conferencing solution. Notwithstanding, Red5 is used in a variety of social applications that range from chatroullette knock-offs, to Facebook's message recording application. Red5 has also been used with projects involving IP Cameras, Gaming and Health Care. In addition, there is several other open source projects have spun off based on the server including Xuggle, an open source transcoding solution and Jedai a communications framework.

Chapter 3: Software Requirements Specification

3.1. Functional Requirements

3.1.1 Authentication

3.1.1.1. Description and Priority

This feature will ensure that user will use his/her own account to use the Lab PC application on Client Level & gives only access to his/her own account/services by using the unique User ID & password.

3.1.1.2. Stimulus/Response Sequences

User starts the application and the login screen thereby appears.

3.1.1.3. Functional Requirements

REQ 001 Login All users' Smartphone must be connected to the internet and the red5 server must be running. Users will input the username & password, Server will authenticate it from the Database and gives access to applications' services.

3.1.2. Management of voice message at Clients End

3.1.2.1. Description and Priority:

This feature will provide user to record the voice message as a media file. Once its recorded the user will be able to play and/or stop their recording via applications" embedded media player.

3.1.2.2. Stimulus/Response Sequences

User Records the voice message

3.1.2.3. Functional Requirements

REQ002 User can Record the voice message

REQ003 User will be able to play the recorded voice message on the smartphone

REQ004 User will be able to stop the recorded voice message over the play timeline in the application's embedded media player.

3.1.3. Management of Streamed Voice message

3.3.3.1. Description and Priority:

This feature will provide user to upload and stream the recorded voice message via red 5 server and get it stored on a database for further use.

3.3.3.2. Stimulus/Response Sequences

User will send the recorded voice message via android smart phones with active internet connections

3.3.3.3. Functional Requirements

REQ005 User will send the voice message via client's end for voice to text conversion of the message

REQ006 Red5 server will stream the voice message after establishing a successful connection with the appropriate client

REQ007 Red5 server will store the appropriate audio links streamed in MYSQL database

3.1.4 Processing the voice message

3.1.4.1. Description and Priority:

This feature will enable the Speech to text conversion of the audio message stored in the database. The Voice message will be converted into Roman urdu text.

3.1.4.2. Stimulus/Response Sequences

As soon as the database is loaded with a new audio link the voice message will be sent to third party (off- the shelf) Speech to text converter that will convert t the audio message into Roman Urdu Text

3.1.4.3 Functional Requirements

REQ008 The Audio message is sent to the third part Speech to text converter to convert the voice/audio message into Roman Urdu Text

REQ009 Web server will stream the text back to the Client

3.1.5. Sending the text message

3.1.5.1 Description and Priority:

This feature will enable the user to send the acquired Roman urdu text to their contact(s)

3.1.5.2 Stimulus/Response Sequences

User sends the text message received from the web server

3.1.5.3 Functional Requirements

REQ010 The user will add appropriate recipient(s) from its contact's list saved in the clients' internal database and send the text.

3.1.6. Accessing User's saved text messages and saved recordings

3.1.6.1 Description and Priority:

This feature will enable the user to resend the last sent text messages or can resend the last recorded voice messages for reprocessing.

3.1.6.2 Stimulus/Response Sequences

User logs in and gets access to last sent messages (audio or text)

Functional Requirements

REQ011 The Client will display last sent text messages making use of its internal database (SQLite in case of Android OS)

Dictation App:

3.1.7. Management of voice message at Clients End

3.1.7.1 Description and Priority:

This feature will provide user to record the voice message as a media file. Once its recorded the user will be able to play and/or stop their recording via applications" embedded media player.

3.1.7.2 Stimulus/Response Sequences

User Records the voice message

3.1.7.3 Functional Requirements

REQ012 User can Record the voice message

REQ013 User can delete the recorded messages.

REQ014 User will be able to play the recorded voice message on the Smartphone

REQ015 User will be able to stop the recorded voice message over the play timeline in the application's embedded media player.

3.1.8. Accessing User's saved recordings

3.1.8.1 Description and Priority:

This feature will enable the user to resend the last recorded voice messages.

3.1.8.2 Stimulus/Response Sequences

User logs in and gets access to last sent messages (audio)

3.1.8.3 Functional Requirements

REQ016 The Client will display last sent messages making use of its internal database (SQLite in case of Android OS or any SD Card).

3.1.9. Management of Streamed Voice message

3.1.9.1 Description and Priority:

This feature will provide user to upload and stream the recorded voice message via red5 server and get it stored on a database for further use.

3.1.9.2 Stimulus/Response Sequences

User will send the recorded voice message via android smart phones with active internet connections

3.1.9.3 Functional Requirements

REQ017 User will send the voice message via client's end.

REQ018 Red5 server will stream the voice message after establishing a successful connection with the appropriate client

REQ019 Red5 server will stream the audio messages towards the networking server.

REQ020 Networking server will store the appropriate audio links streamed in MYSQL database.

3.1.10. Authentication

3.1.10.1 Description and Priority:

This feature will ensure that only registered users can access their public and private messages received from Administrator.

3.1.10.2 Stimulus/Response Sequences

User visits the website and the login screen thereby appears.

3.1.10.3 Functional Requirements

REQ021 Public Login:

All users must be connected to the internet and the networking server must be running. Users will input the public username & password, Server will authenticate it from the Database and user will get access public audio messages.

REQ022 Private Login:

User will input his/her private username & password; Server will authenticate it from the Database and user will get access private audio messages.

3.2. Nonfunctional Requirements

3.2.1 Performance Requirements

Response Time:

Server's response time for audio streaming should be less.

Audio to text conversion should be efficient and quick.

3.2.2 Safety Requirements

Data may loss while streaming because of poor tuning of Red5 server or Client End Application (Android Based Cell)

The internet connection interruption can also cause the data loss.

3.2.3 Security Requirements

The users will get their specific user name and passwords in this way user authentication will ensure.

Only registered or allowed users will be able to receive the public and private messages.

3.2.4 Software Quality Attributes

3.2.4.1 Adaptability

System will be user friendly, manageable and easy to use.

3.2.4.2 Availability

Our system will perform its functionality whenever needed. Any registered android user having this application on his android cell can be benefited from this service.

3.2.4.3 Correctness

There will be no bugs or errors in the running system. In case of any unexpected scenario the error message will be shown by the system.

3.2.4.4 Flexibility

Our system should be flexible enough to overcome low and varying bandwidth problems by using necessary buffering and efficient compression and coding algorithms.

The user interfaces should be flexible enough for the user to pass their information without any hurdle.

It should be flexible to accommodate any additional feature required specifically on the web end.

3.2.4.5 Interoperability

The communicating modules i.e. android app, streaming server, web user interface will work together properly.

3.2.4.6 Maintainability

In case of any change in environment/requirement it will be able to upgrade /maintain or update.

3.2.4.7 Reliability

There will be less chances of system failure and in case of any failure system will be able to recover in less time and no data will be lost.

3.2.4.8 Portability

Our system will be portable in a sense that it can be operated on any android based cell phone having built in audio/video recorder and GPRS/EDGE availability on that phone.

3.2.4.9 Reusability

This application/component can be reused in any related software to upgrade its version.

3.2.4.10 Robustness

In case of any erroneous input or changes in its operational environment (where there is a compatibility issue) system will generate an error message.

3.3 Design and Implementation Constraints

The most prominent constraint for this project is the Android OS version of the smart phones; for this particular project we will require Android OS version 2.1 or above. The Clients' end internal database is SQLite therefore we need to manage the users contact on smart phone using this specific database only. The servers processing speed is another constraint that is dependent on the internets' connection speed. Programming of Android Client can be done in Eclipse framework only.

Chapter 4: Software Design Specification

4.1. INTRODUCTION

Latest technology trends show that there has been an immense growth in the smart phone mobile industry. Numerous applications have taken over the Smartphone's app markets that are capable of streaming media files effectively. Since this technology is new therefore there is immense space of innovations and creativity that can be applied to develop fascinating and useful applications.

There are quite a few streaming media servers available that provide efficient streaming of audio/video via smart phones as client. The primary objective of the project is to set up a client and server architecture, where Smart phones with Android OS and a flash media web server Red5 will be used. The architecture set up will be used as an effective engine to facilitate to develop suite of smart phone applications. The applications will be used to demonstrate the working of the client/server architecture that is a new architecture recently introduced for streaming media files.

4.2. PRODUCT SCOPE

A transmission engine will be setup that is client server architecture that incorporates Android OS smart phone as Clients and Red 5 Server as the audio/video streaming server. The transmission Engine will specifically be used for audio streaming. Two applications will be developed in order to demonstrate the viable working of the Transmission Engine.

The two applications that will be developed are

- Dictation application
- Speech to Text SMS application

The Dictation app will facilitate enterprise users to send, share and access audio messages from their Smart Phones and they will also be able to access these audio messages via a web interface also. The second app however is a Speech to Text message app (SMS application) that will allow its users to send text message in Roman Urdu of the audio message that the users will stream via their smart phones to be sent to their contact(s).

4.3. Product Functions

We are not developing any Transmission Engine. The existing Transmission Engine will be able to stream audio messages from client side to the web server and back to client side.

The two main applications developed will provide the following functionalities

- **Dictation Application:**

    - Provide Android Smart phone users to stream two way audio messages.
    - Store and manage them in a database.
    - Customize the usage of the streamed audio messages.
    - A web interface is also provided in order to access the streamed audio messages.

- **Speech to Text SMS app:**

    - Android Smart phone users will be able to stream audio SMS via Red5 Server and receive text in Roman Urdu after processing
    - Provide the facility to send the streamed text in Roman Urdu to the desired recipient.
    - On Server side off the shelf component will be used for audio to text conversion.

4.4. Design Decisions

4.4.1 Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.

4.4.2 Architecture for Speech to Text Application

To realize the overall speech to text sms functionality of the application primarily client server architecture is the most obvious architecture model for the application. But for ordering the applications components to function properly three tier Software architecture will be used for this Texting application.

The three primary tiers are:

i. Client tier
ii. Server tier
iii. Data tier

i. Client Tier:

This layer presents data to the user and permits data manipulation and data entry. The two main types of user interface for this layer in accordance to the applications are the traditional smartphone application and the Web-based application.

In case of Speech to test application the Client or the presentation layer consists of the smartphone’s installed application that is the application installed at clients end. This application is written in java and is coded such to communicate with the streaming web server. The client’s applications basically provides user with the functionality to interact with the system.

ii. Server Tier:

This tier comprises of the RED5 streaming server and the text streaming application server. This layer is responsible for all the audio messages transmission to the Speech to text conversion off the shelf component from the client (android smartphone). And for streaming the text back to the client the text streaming application server will be used.

iii. Data Tier:

This tier comprises of the built in Database SQLite of Android smartphone. The contact names and sent texts will be accessed via SQLite queries at client end.

4.4.3 Architecture for Dictation Application

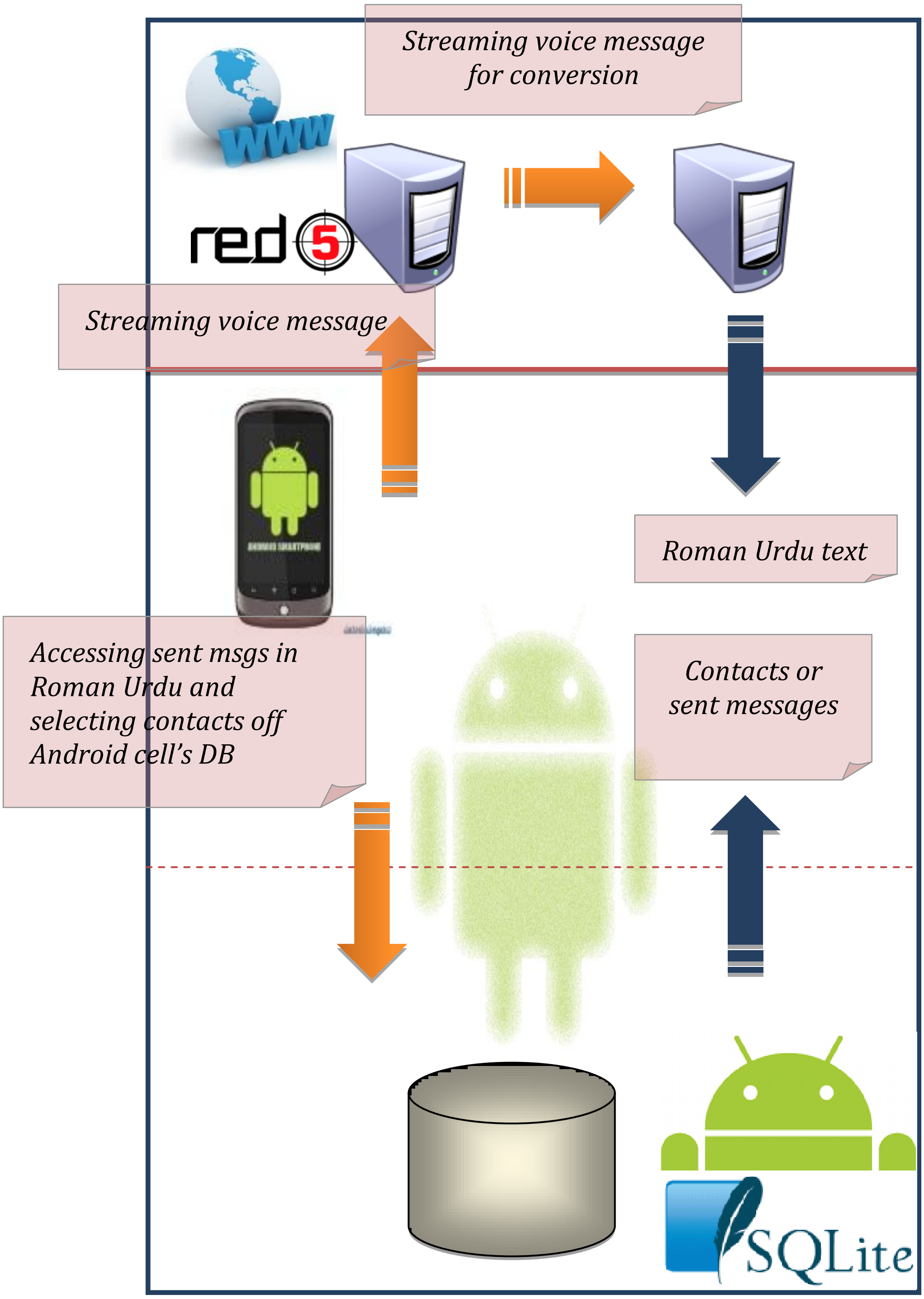
The architecture for the Dictation app is same as that of the previous application. Since this application can be accessed at web end too therefore the traditional three tier client server architecture will be modified into a web 3-tier architecture framework; considering the application access via web.

Tiers for the Dictation application

The first tier in the 3 tier architecture is the **client tier**, the client tier comprises of smartphone installed application as client or for a web end user the web will be the client tier. This tier comprises of web browsers for HTTP clients that efficiently interacts with the Dictation app using the most convenient and standardized protocols.

The second tier comprises of the streaming RED5 server that streams the audio messages and publish them accordingly on the web.

The third tier is the **data tier** that consists of the SQL DataBase to send audio messages to other authorized users.



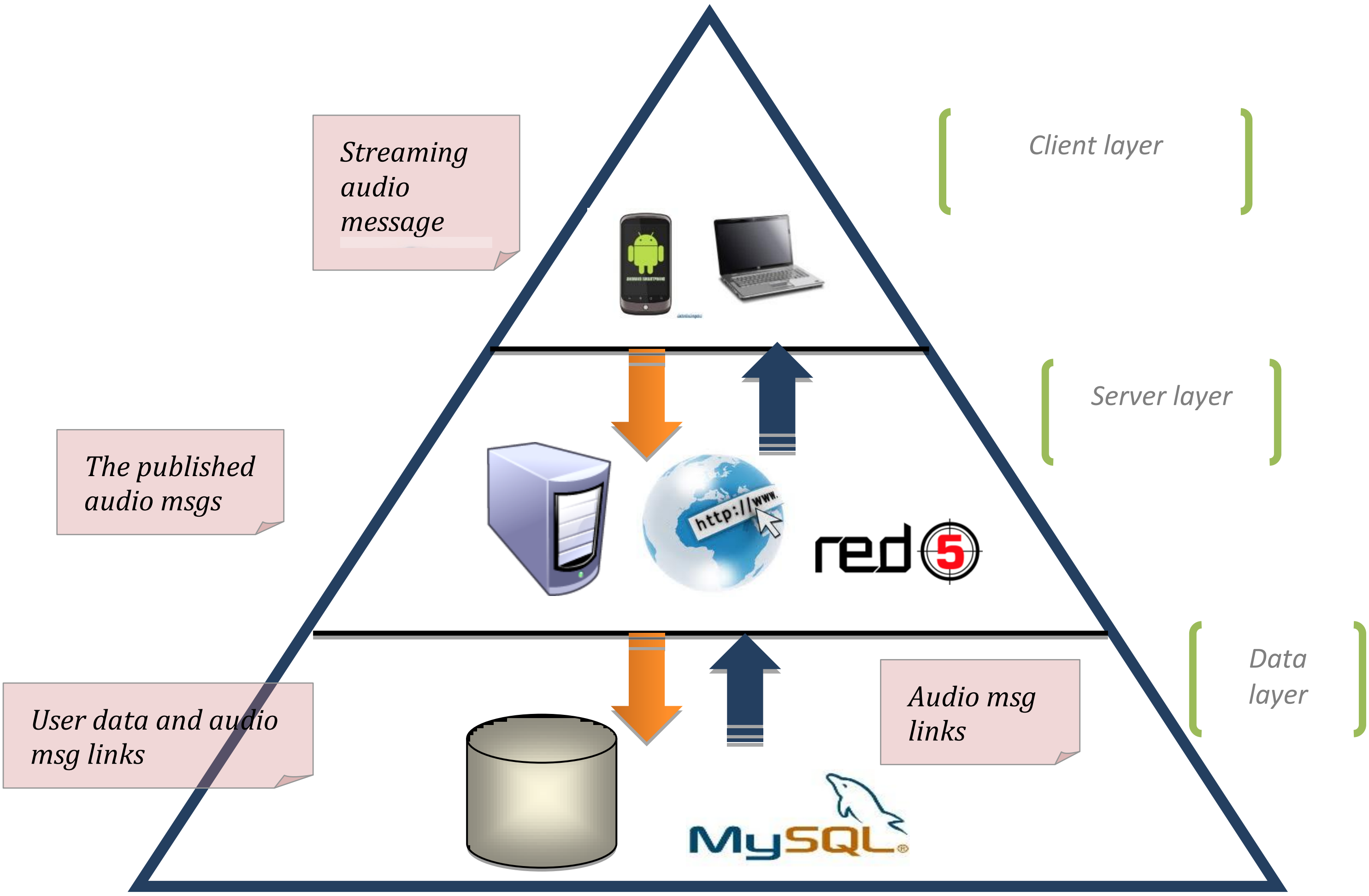
Streaming voice message for conversion

Streaming voice message

Roman Urdu text

Contacts or sent messages

Accessing sent msgs in Roman Urdu and selecting contacts off Android cell's DB


Streaming audio message
Client layer
Server layer
The published audio msgs
red5
Data layer
User data and audio msg links
Audio msg links
MySQL

4.5. DESIGN PATTERN:

4.5.1 FRONT CONTROLLER DESIGN PATTERN

The Front Controller pattern defines a single component that is responsible for processing application requests. “It provides a centralized entry point for handling requests”. Consequently, when the behavior of these functions needs to change, only a small part of the application needs to be changed that is the controller and its helper classes.

The front controller pattern is usually used because of it flexible nature for extending and controlling execution via the main controller class. In the scenario of Speech to text (Roman Urdu text) application the view class as shown in the class diagram; is the class catering for the entire requests made by the user.

In case of Dictation application the Controller class is the Audio helper that will control the functionality at smartphone client’s end.

4.6. CLASS DIAGRAM:

4.6.1 CLASS DIAGRAM OF SPEECH TO TEXT APP:

MAIN ACTIVITY

This Class is Main activity class of all Android Application. It has the basic functions necessary to initiate or stop a application running on Android phones.

PREPARE STREAM

This class is responsible for preparing the speech for streaming that is converting the stream into flv stream for the RED5 Streaming server using ffmpeg libraries.

MICROPHONE

This class extends the Android’s Activity class that provides necessary functions to control the smartphone's microphone of recording speech

SQLITE

The class that executes and prepares SQLite database queries.

VIEW

This class is the controller class in accordance to the design pattern. The user directly interacts with the functionalities of the application via this interface class.

TEXTCONVERSION

The class that caters for speech to Roman Urdu text conversion on the web server

STREAMINGSERVER

This class is responsible for creating connection and streaming audio message for transmission

4.6.2 CLASS DIAGRAM OF DICTATION APP

AUDIOHELPER

This class is responsible for initiating the voice streaming task and it records the voice.

TRANSPORT

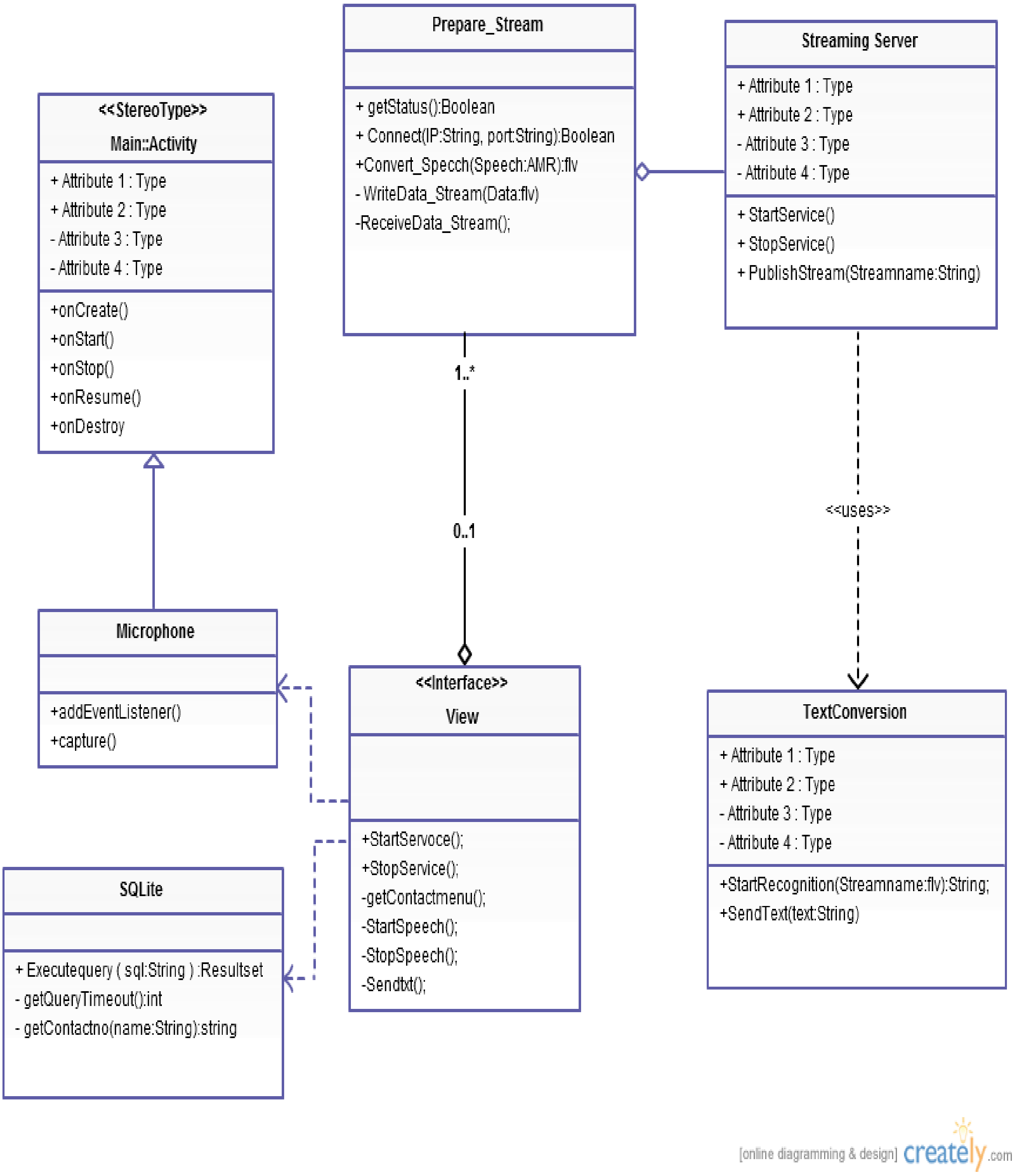
This class is responsible for creating the network packets to be further streamed via RED5 server. The conversion of voice into flv format is also done in this class

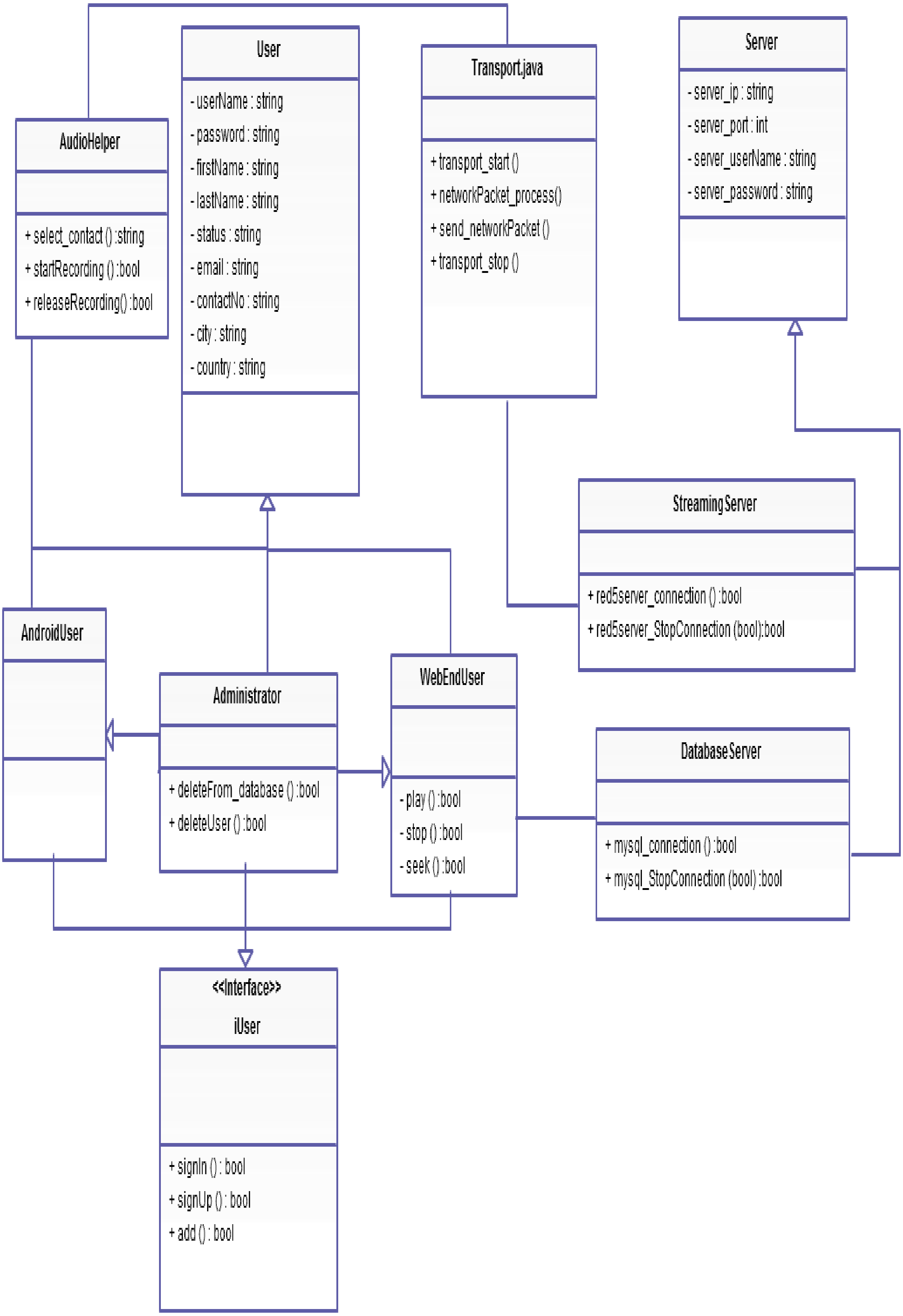
SERVER

The Sever class is responsible for connection with servers at particular port and id. The ***Streaming Server*** and ***Database Server*** extend from this server class. It inherits the functions of this sever class in addition to their nature specific functions.

ADMINISTRATOR

Administrator class provides necessary functions required to manipulate privileges of the registered users. Since it extends from ***Android User*** and ***Web end User*** class therefore it can not only use the Dictation application as a normal user but it has additional functions such as deleting a user or setting up privileges for registered users respectively.

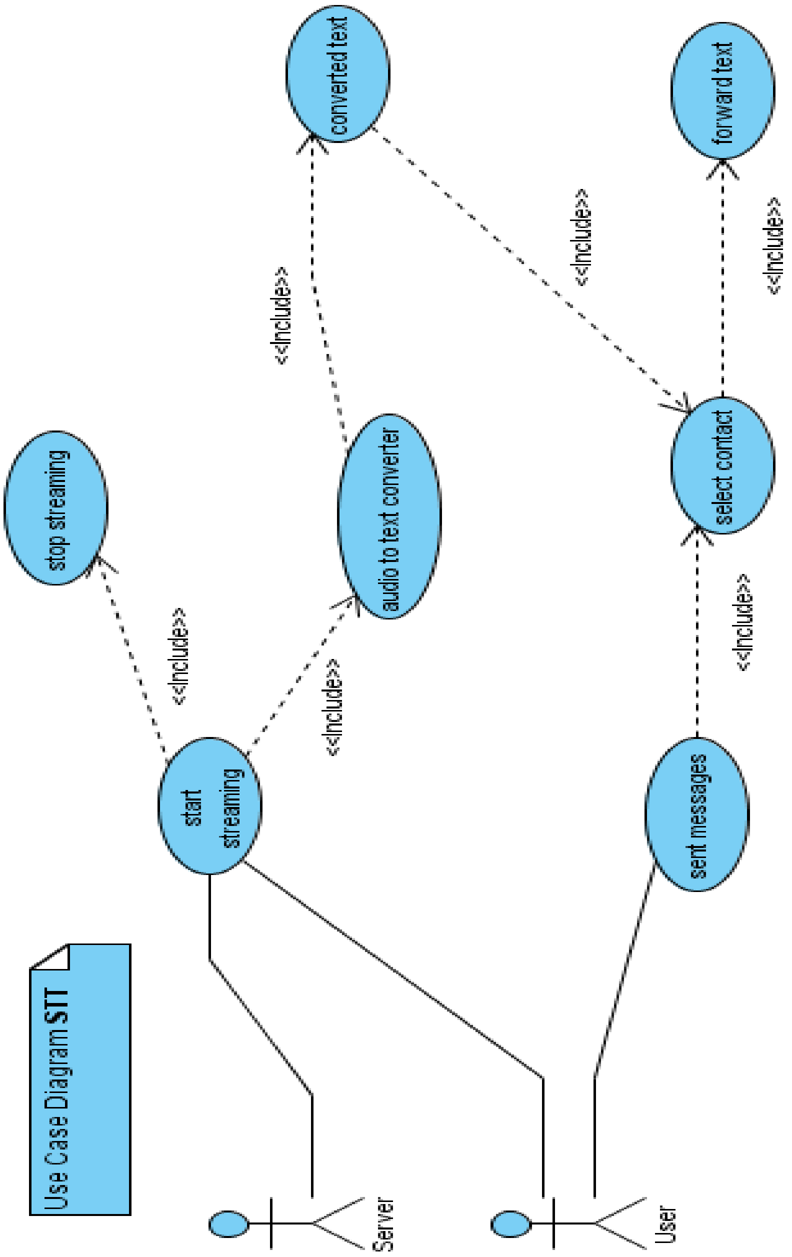



User
- userName : string
- password : string
- firstName : string
- lastName : string
- status : string
- email : string
- contactNo : string
- city : string
- country : string
Transport.java
+ transport_start ()
+ networkPacket_process()
+ send_networkPacket ()
+ transport_stop ()
Server
- server_ip : string
- server_port : int
- server_userName : string
- server_password : string
AudioHelper
+ select_contact () :string
+ startRecording () :bool
+ releaseRecording() :bool
StreamingServer
+ red5server_connection () :bool
+ red5server_StopConnection (bool):bool
AndroidUser
Administrator
+ deleteFrom_database () :bool
+ deleteUser () :bool
WebEndUser
- play () :bool
- stop () :bool
- seek () :bool
DatabaseServer
+ mysql_connection () :bool
+ mysql_StopConnection (bool) :bool
<<Interface>>
iUser
+ signIn () : bool
+ signUp () : bool
+ add () : bool

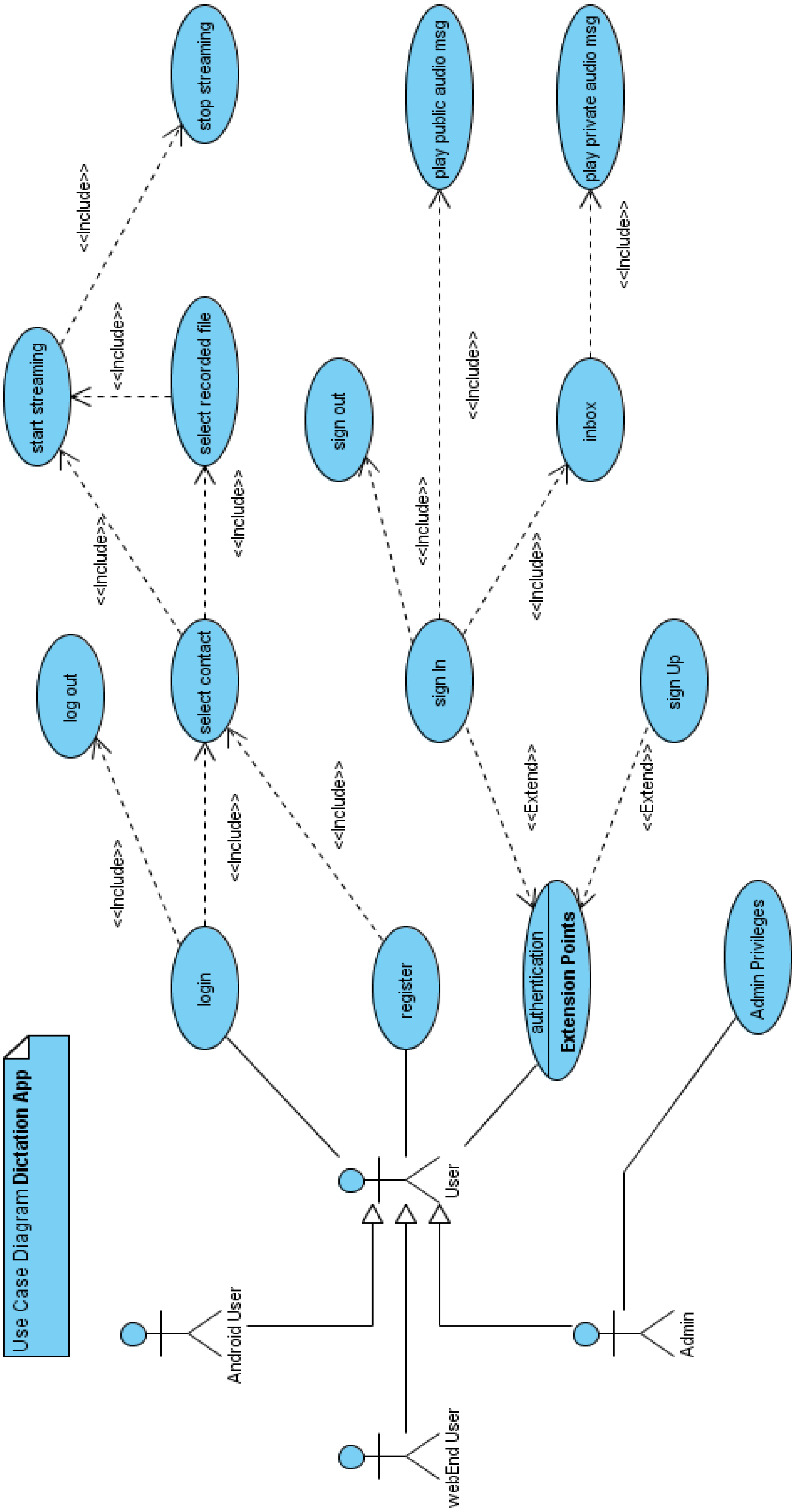
4.7. Use Case Diagram:

Use cases are actions that the user can undertake to interact with the application. From the application's point of view, these actions are functions that the application provides for the user. The following set of use cases defines the functions that the two applications provide:

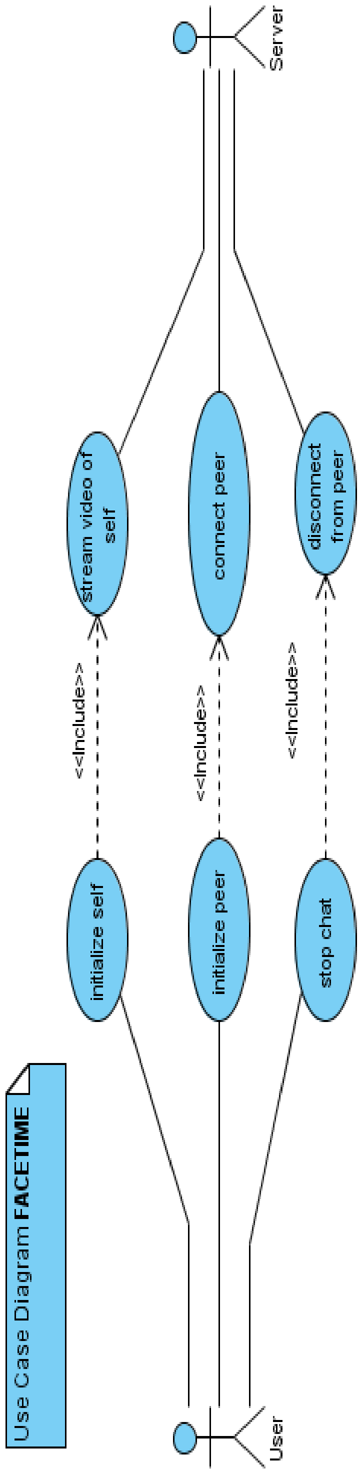
4.7.1. Use Case Diagram for Speech to Text Sms App



4.7.2. USE CASE DIAGRAM FOR DICTATION APP

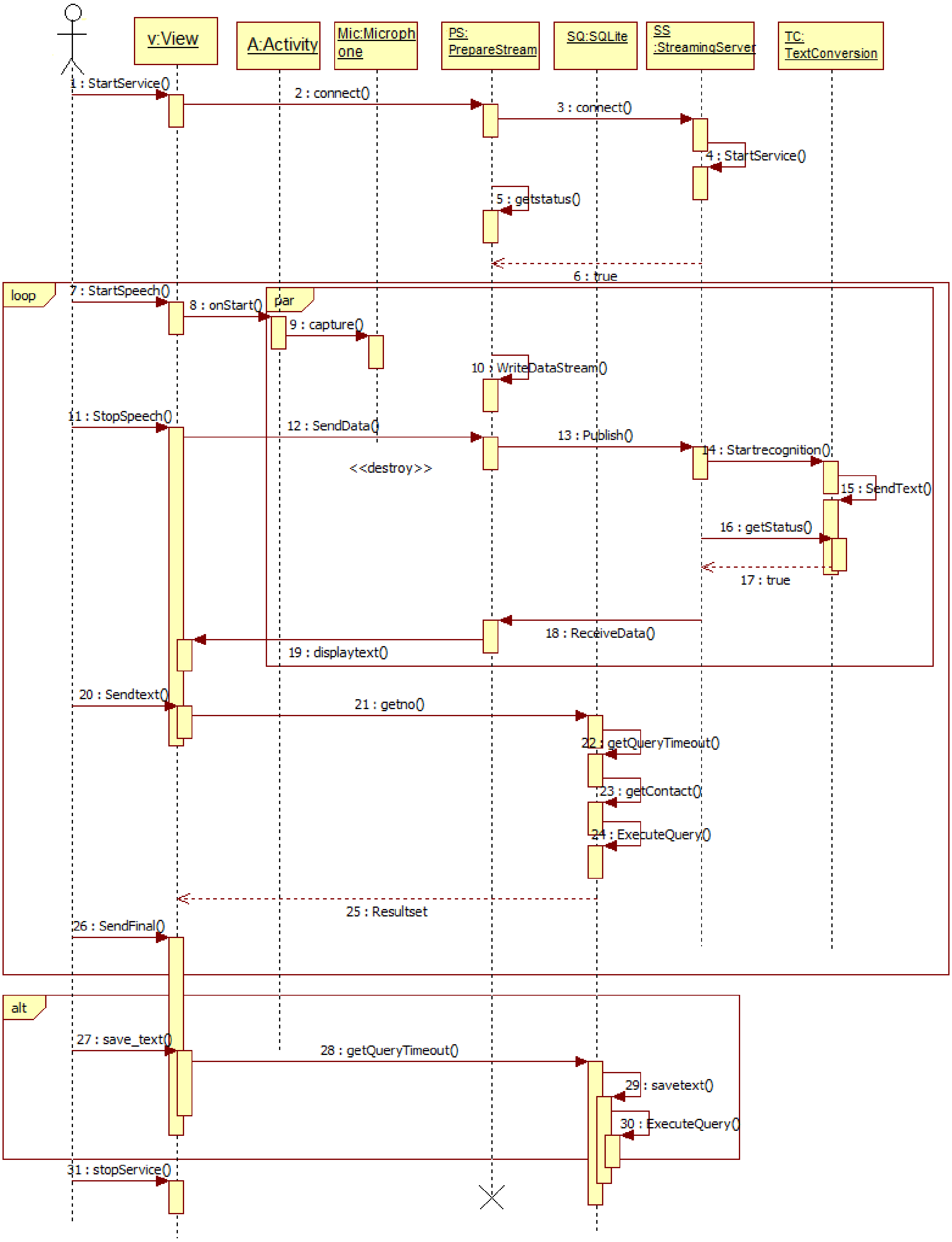


4.7.3. Use Case Diagram for FaceTime app



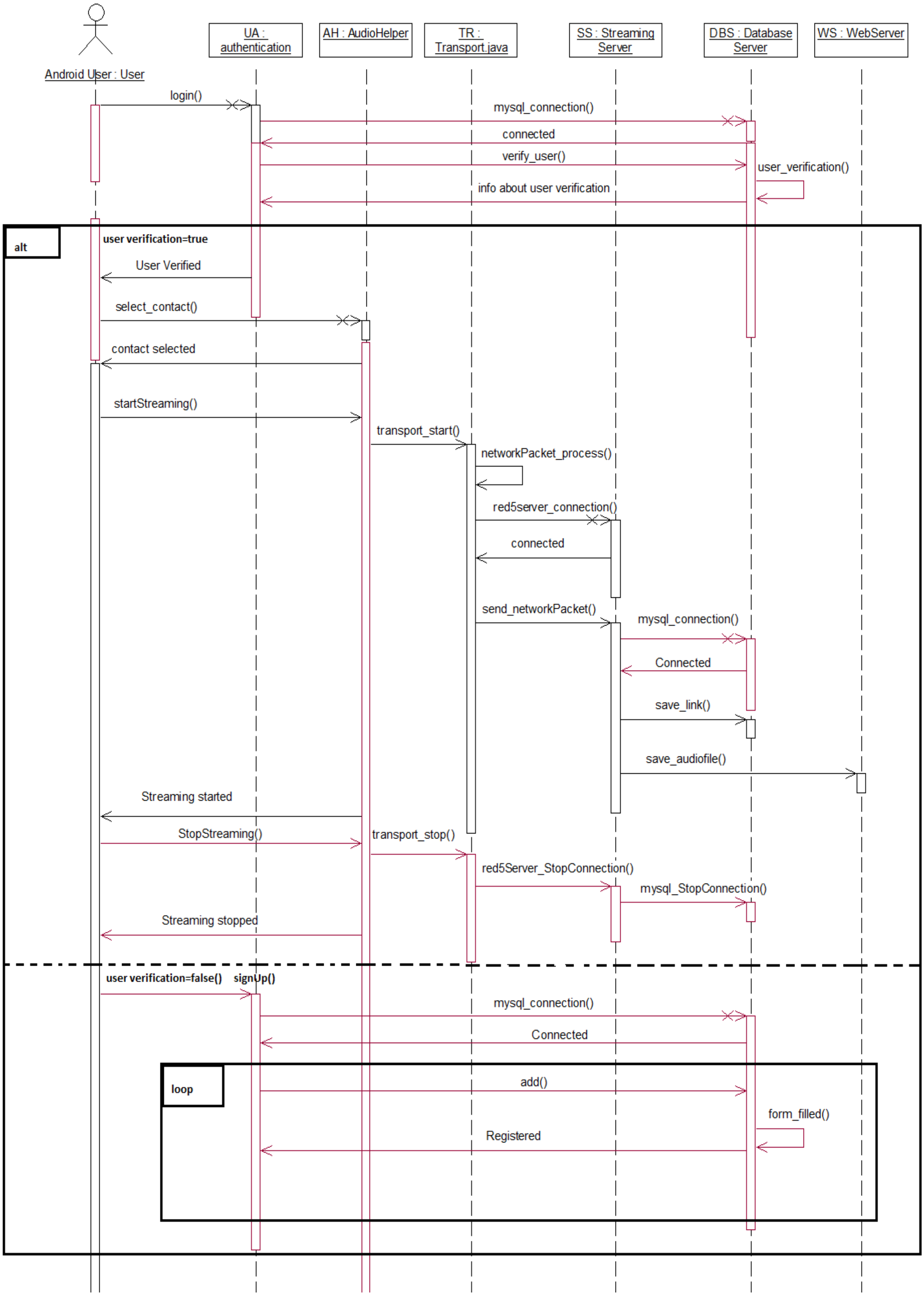
4.8. Sequence Diagrams

4.8.1. Sequence Diagrams Diagram for STT app

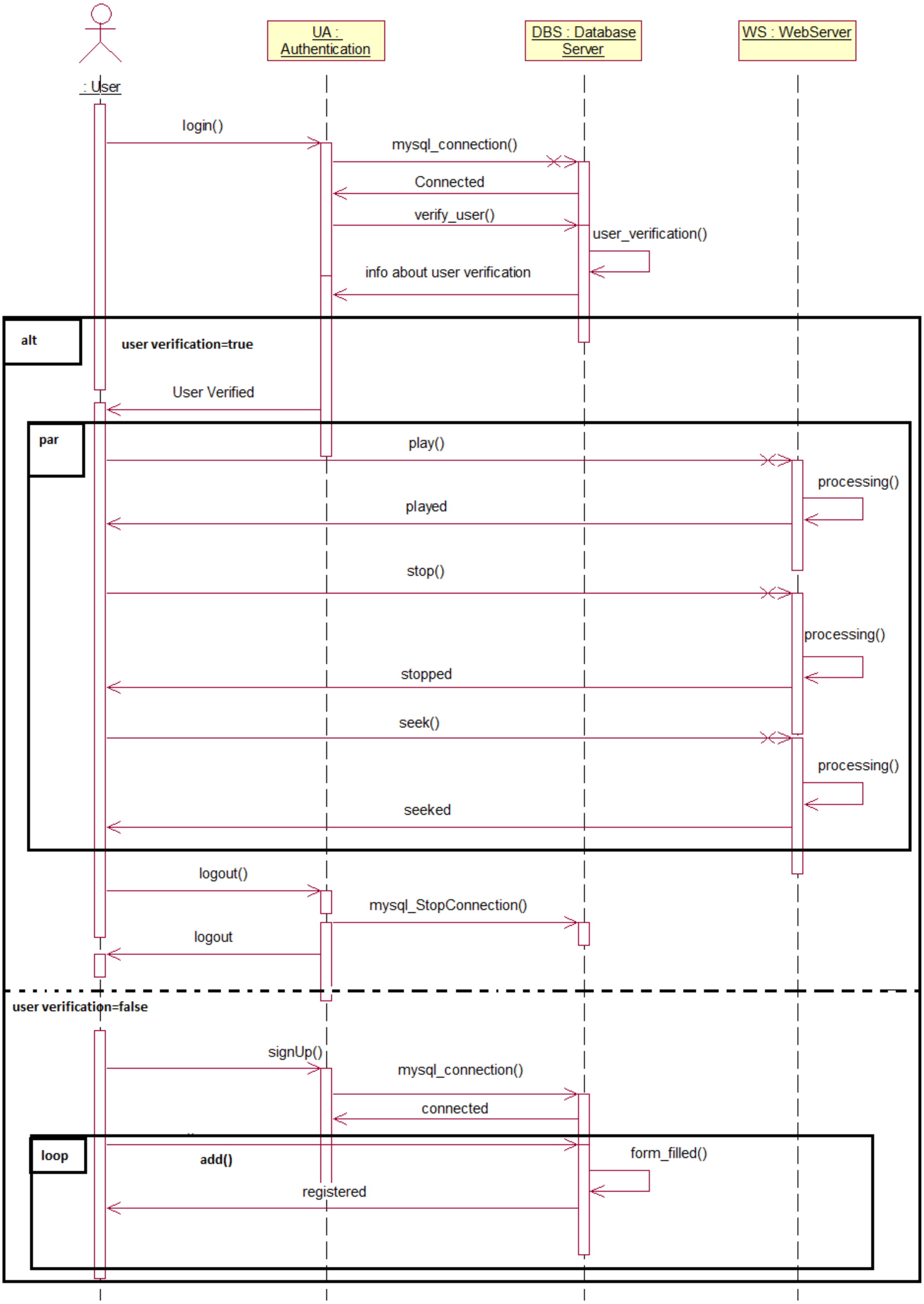


4.8.2. Sequence Diagrams Diagram for Dictation app

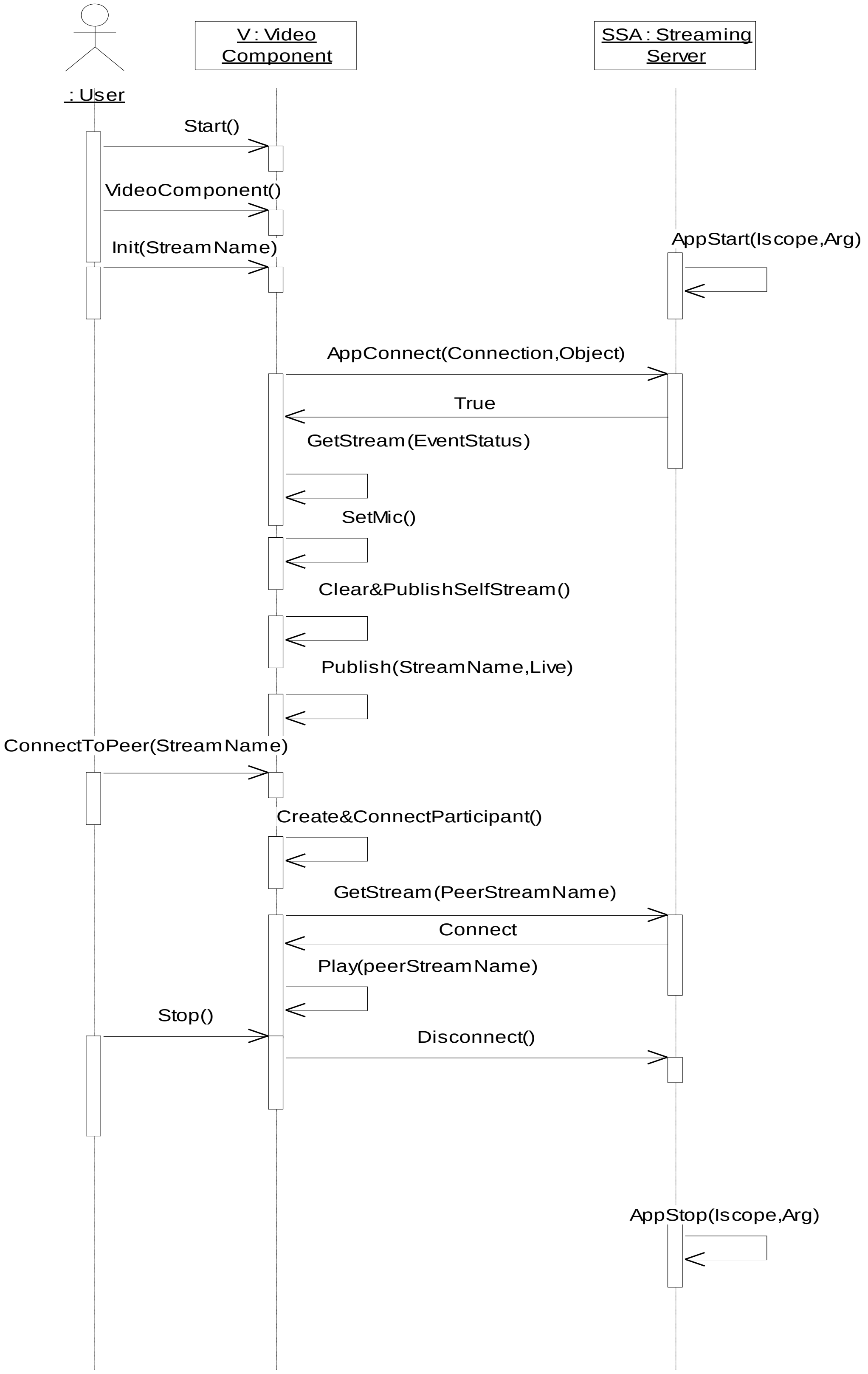
4.8.2.1. Mobile Application (Client-End)



4.8.2.2. Application (Web-End)

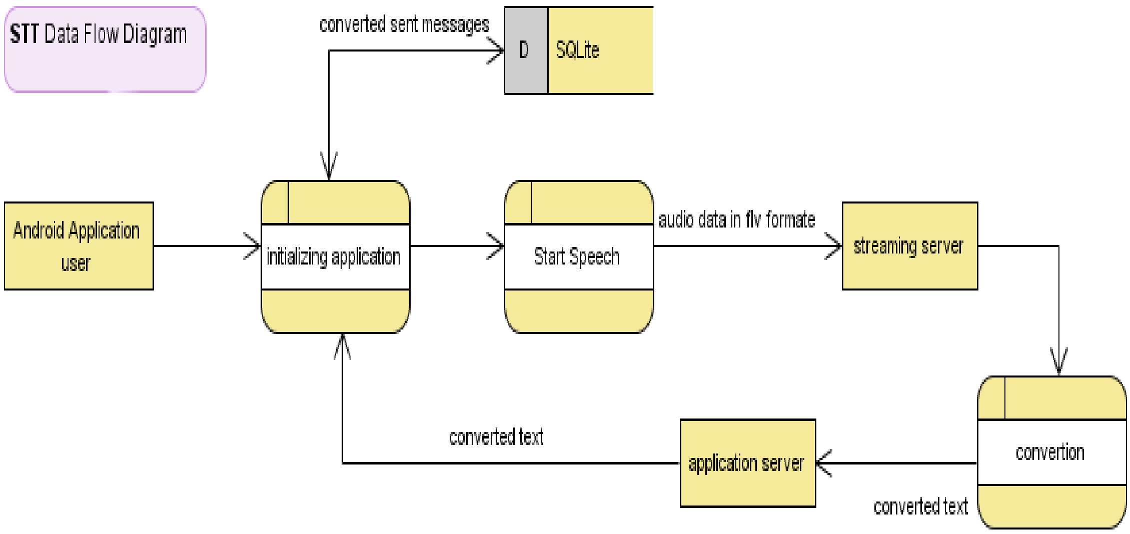


4.8.3. SEQUENCE DIAGRAMS DIAGRAM FOR FACETIME APP

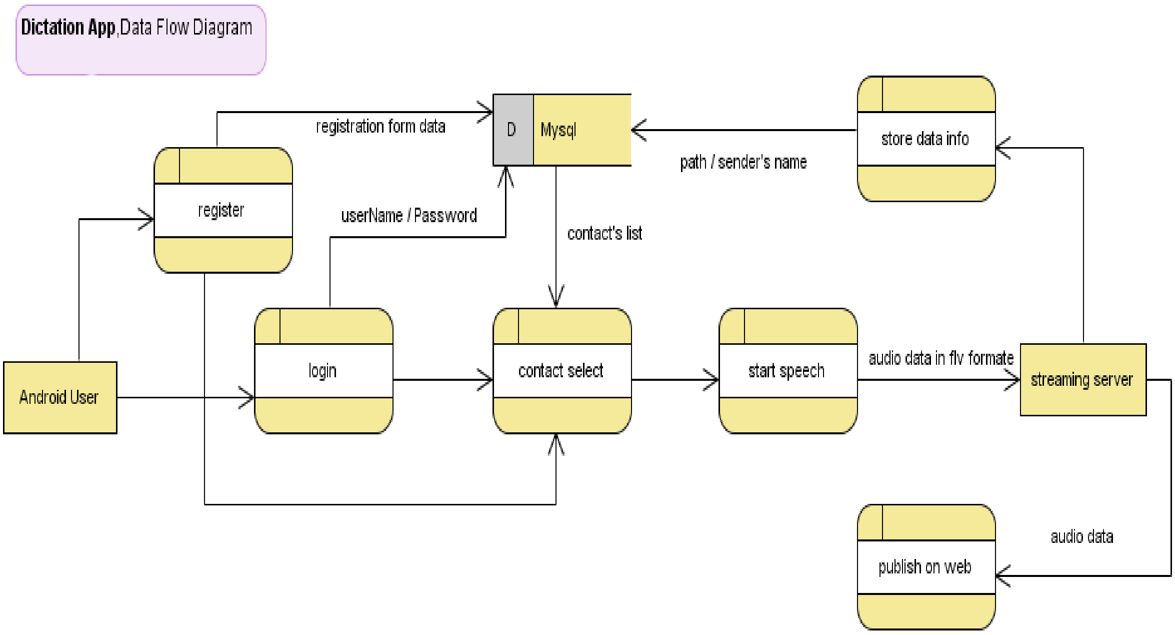


4.9. Data Flow Diagrams

4.9.1. Data Flow diagram for Speech to text App

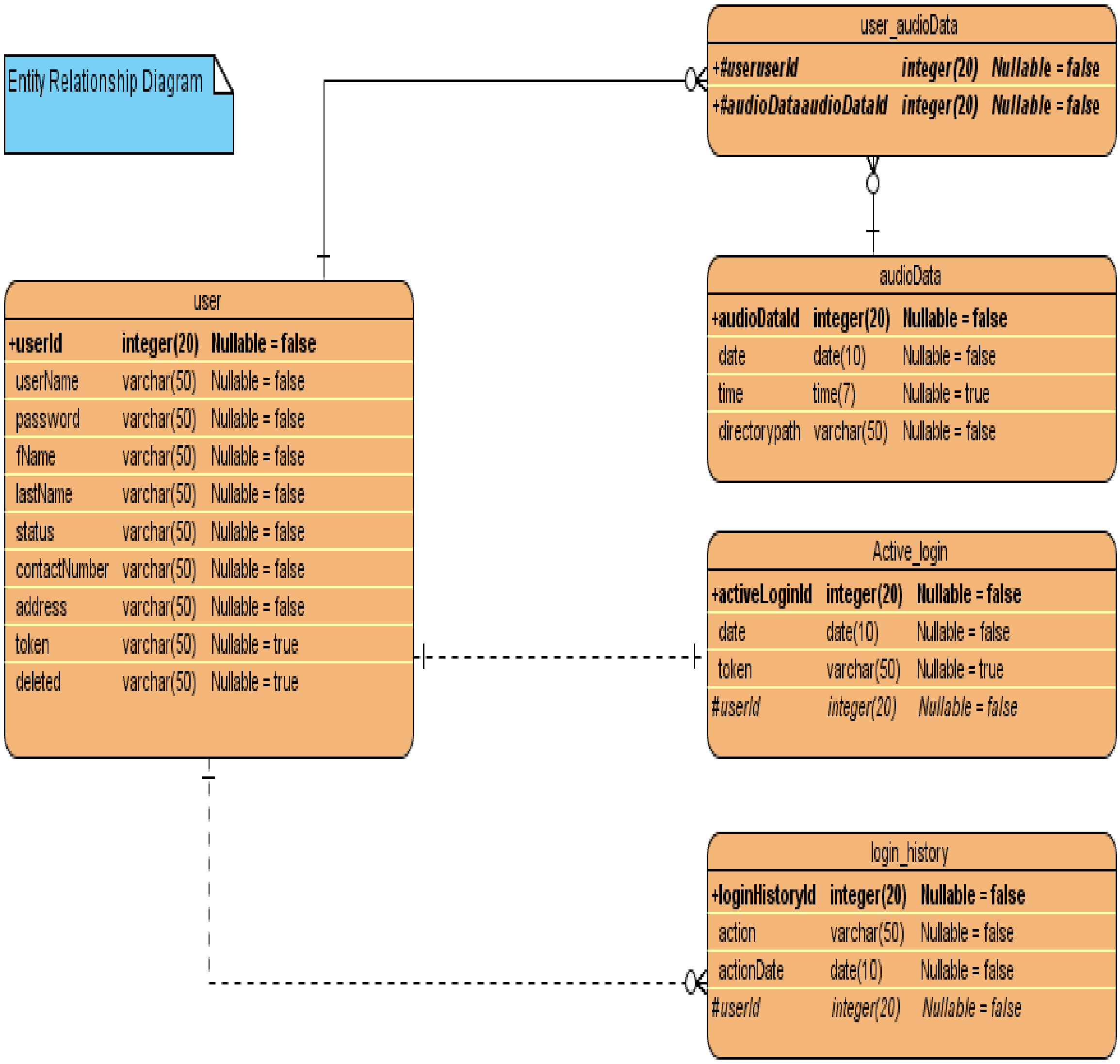


4.9.2. Data Flow diagram for Dictation app



4.10. Data Storage

4.10.1. Entity Relationship Diagram



user

+userId	integer(20)	Nullable = false
userName	varchar(50)	Nullable = false
password	varchar(50)	Nullable = false
fName	varchar(50)	Nullable = false
lastName	varchar(50)	Nullable = false
status	varchar(50)	Nullable = false
contactNumber	varchar(50)	Nullable = false
address	varchar(50)	Nullable = false
token	varchar(50)	Nullable = true
deleted	varchar(50)	Nullable = true

user_audioData

+#useruserId	integer(20)	Nullable = false
+#audioDataaudioDataId	integer(20)	Nullable = false

audioData

+audioDataId	integer(20)	Nullable = false
date	date(10)	Nullable = false
time	time(7)	Nullable = true
directorypath	varchar(50)	Nullable = false

Active_login

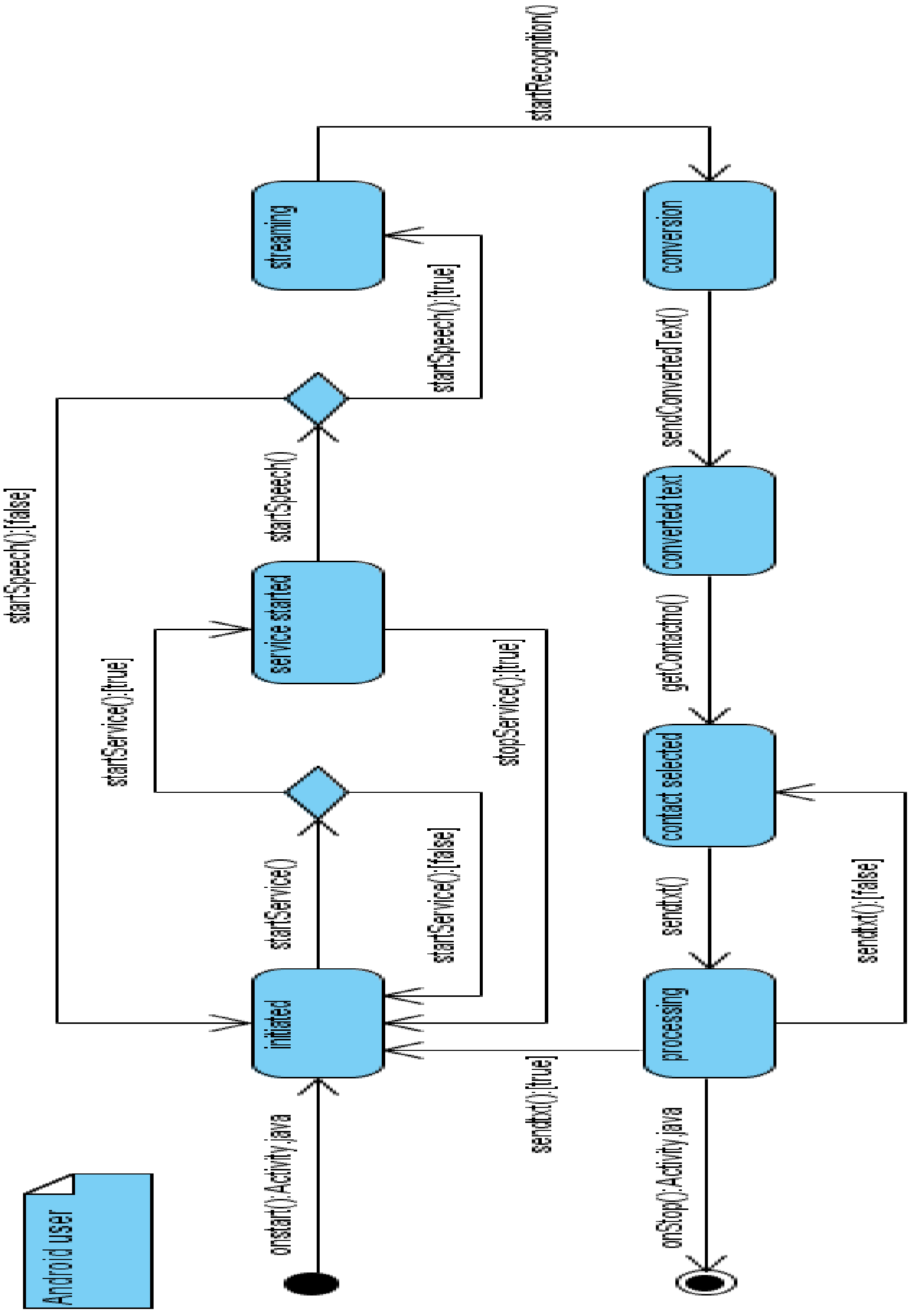
+activeLoginId	integer(20)	Nullable = false
date	date(10)	Nullable = false
token	varchar(50)	Nullable = true
#userId	integer(20)	Nullable = false

login_history

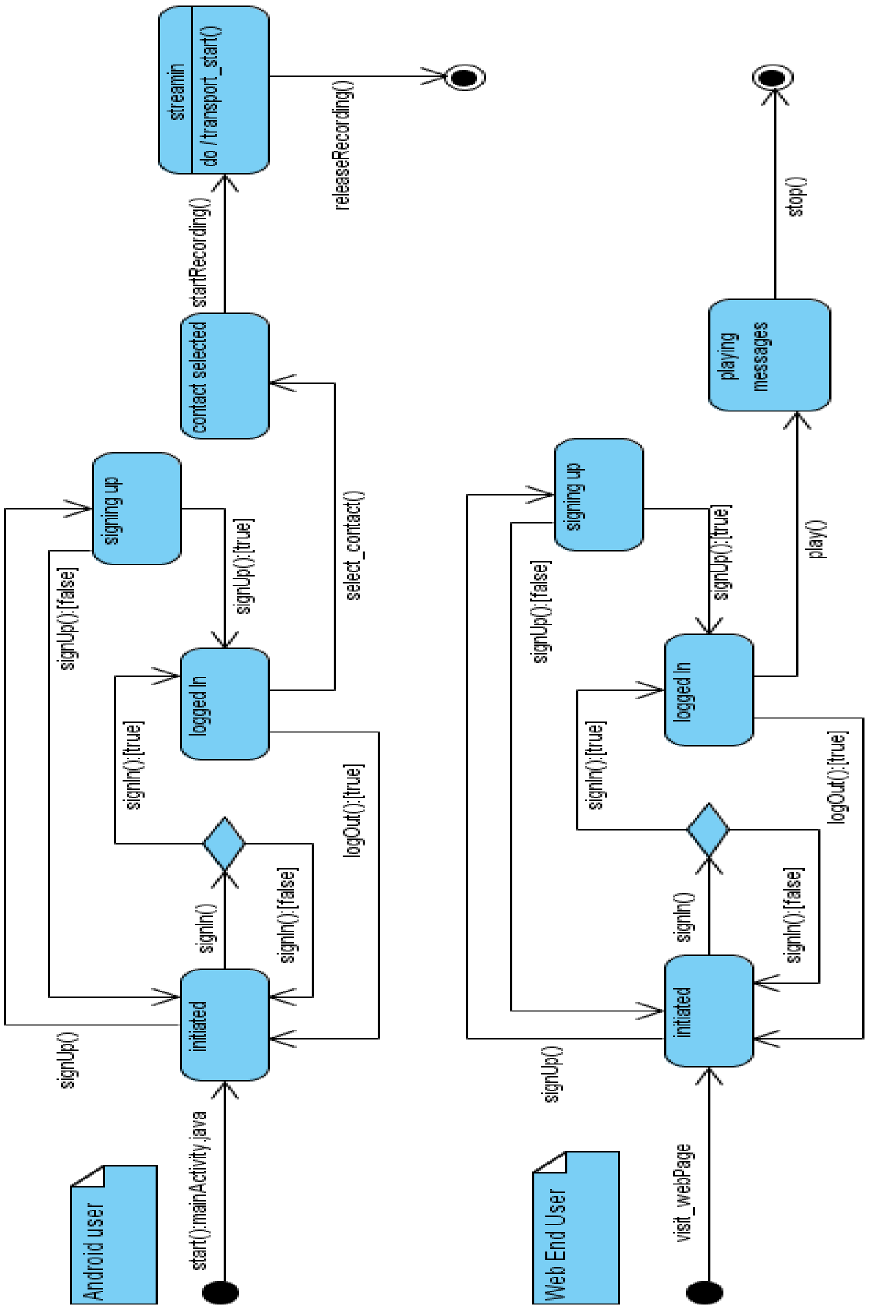
+loginHistoryId	integer(20)	Nullable = false
action	varchar(50)	Nullable = false
actionDate	date(10)	Nullable = false
#userId	integer(20)	Nullable = false

4.11. State Machine Diagrams

4.11.1. Speech to Text App

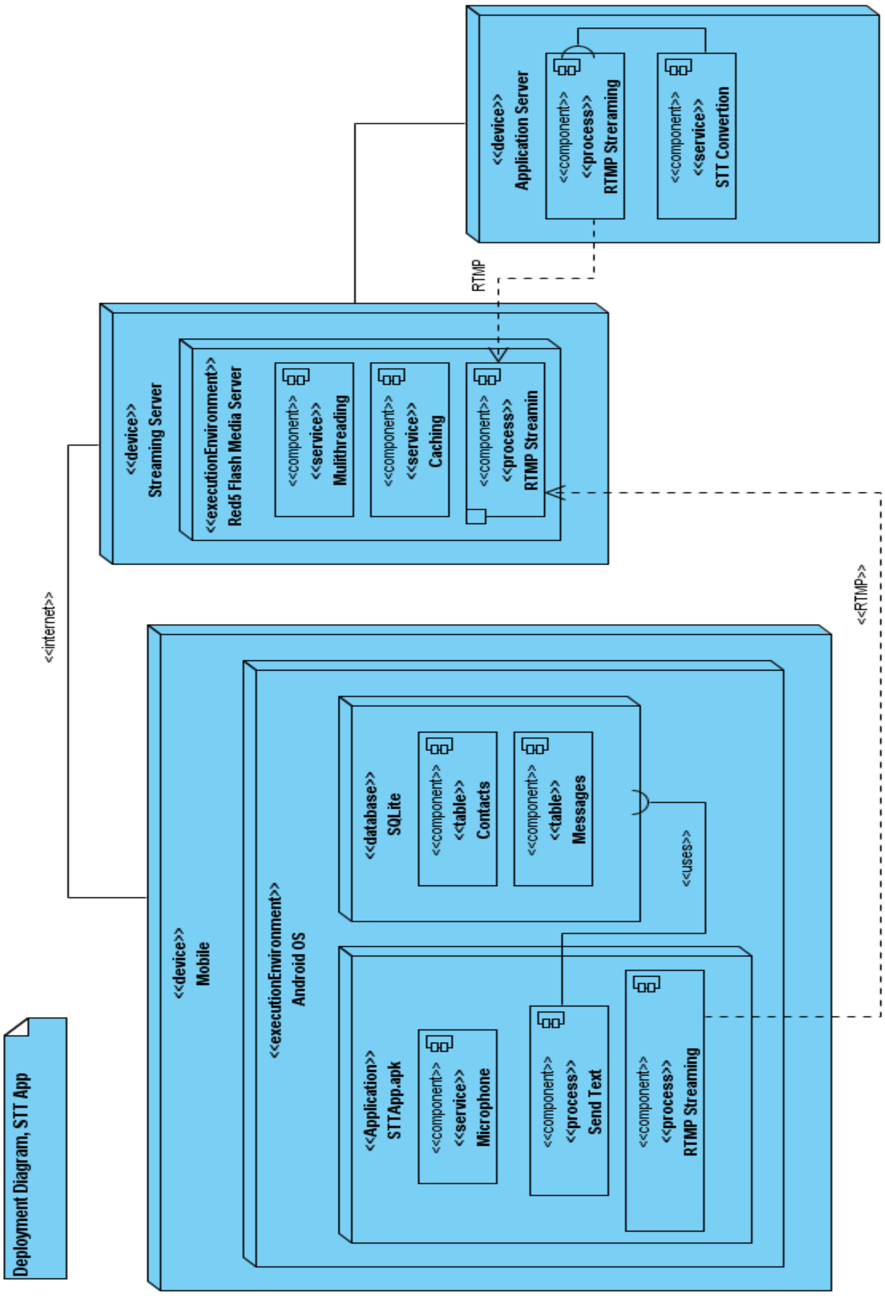


4.11.2. Dictation Application

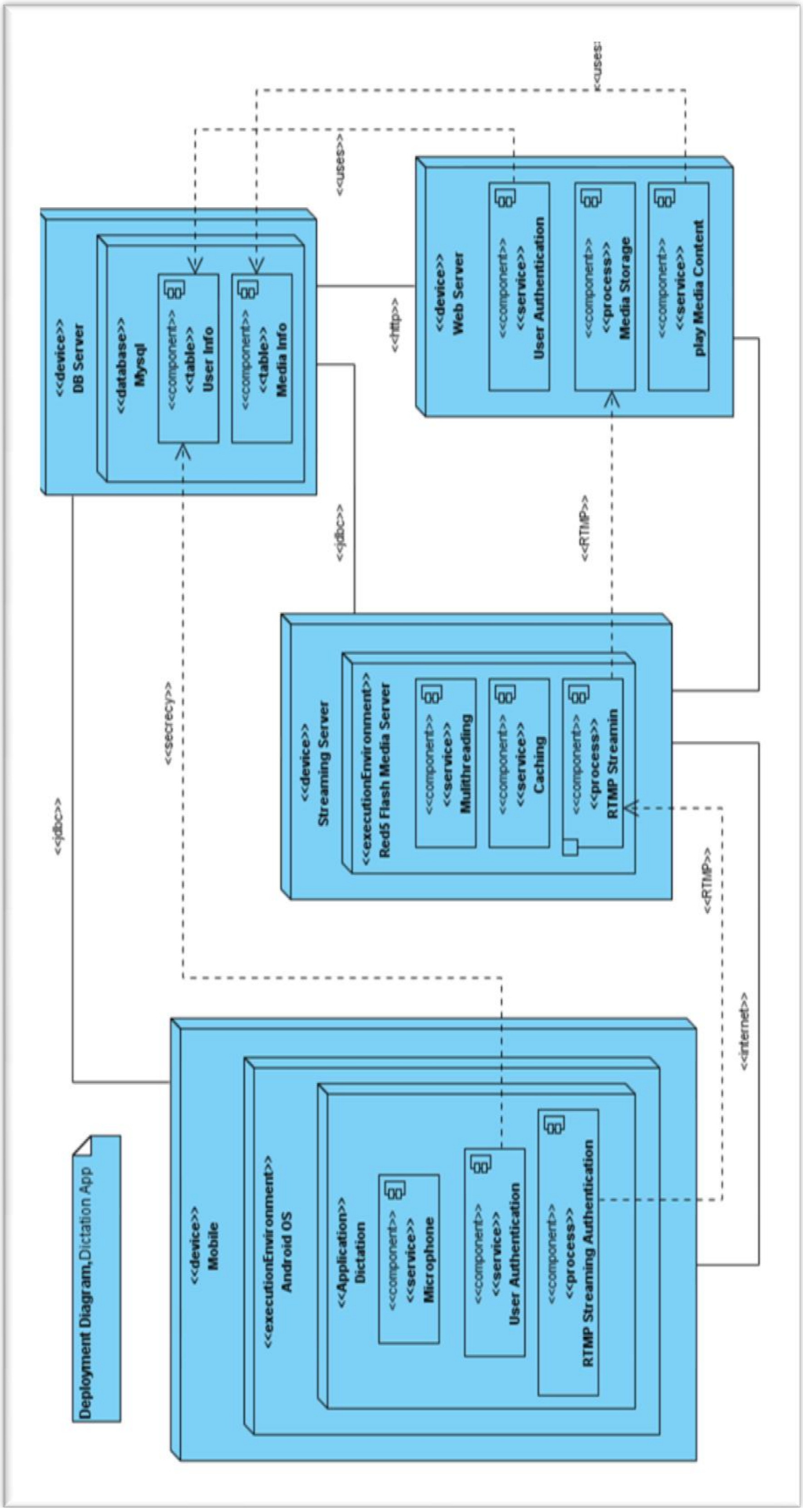


4.12. Deployment Diagram

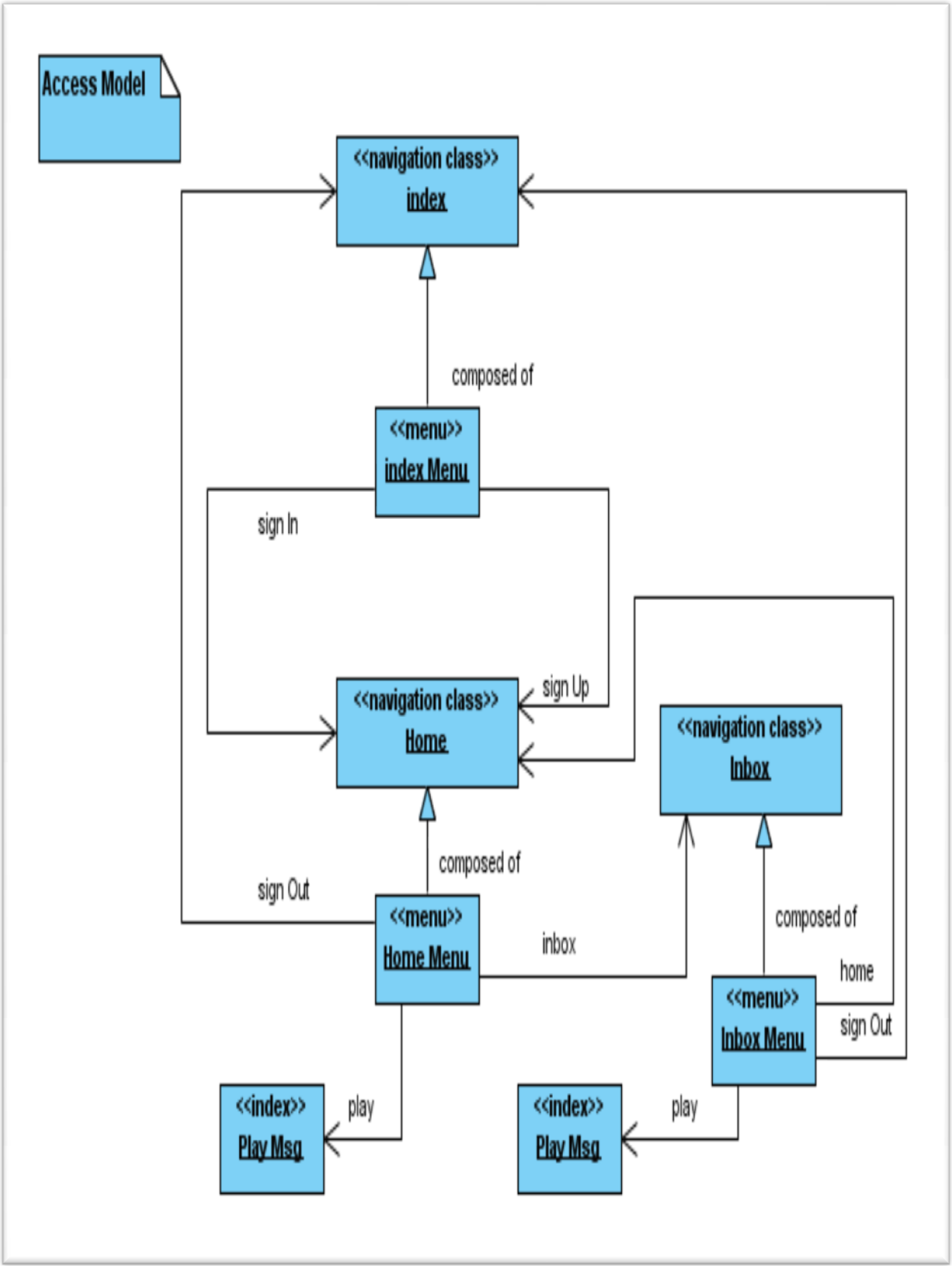
4.12.1. Speech to Text App



4.12.2. Dictation Application

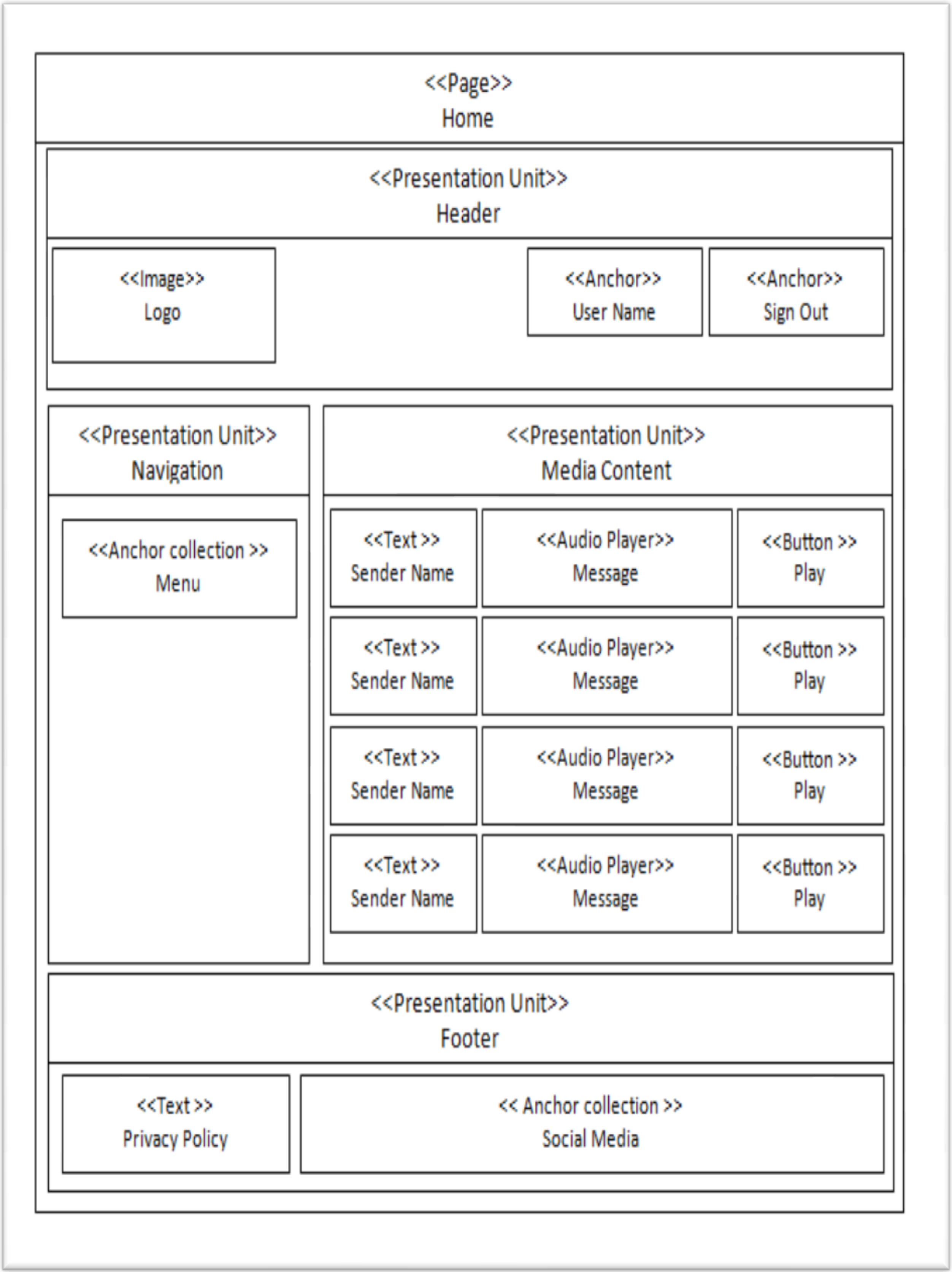


4.13. ACCESS MODEL

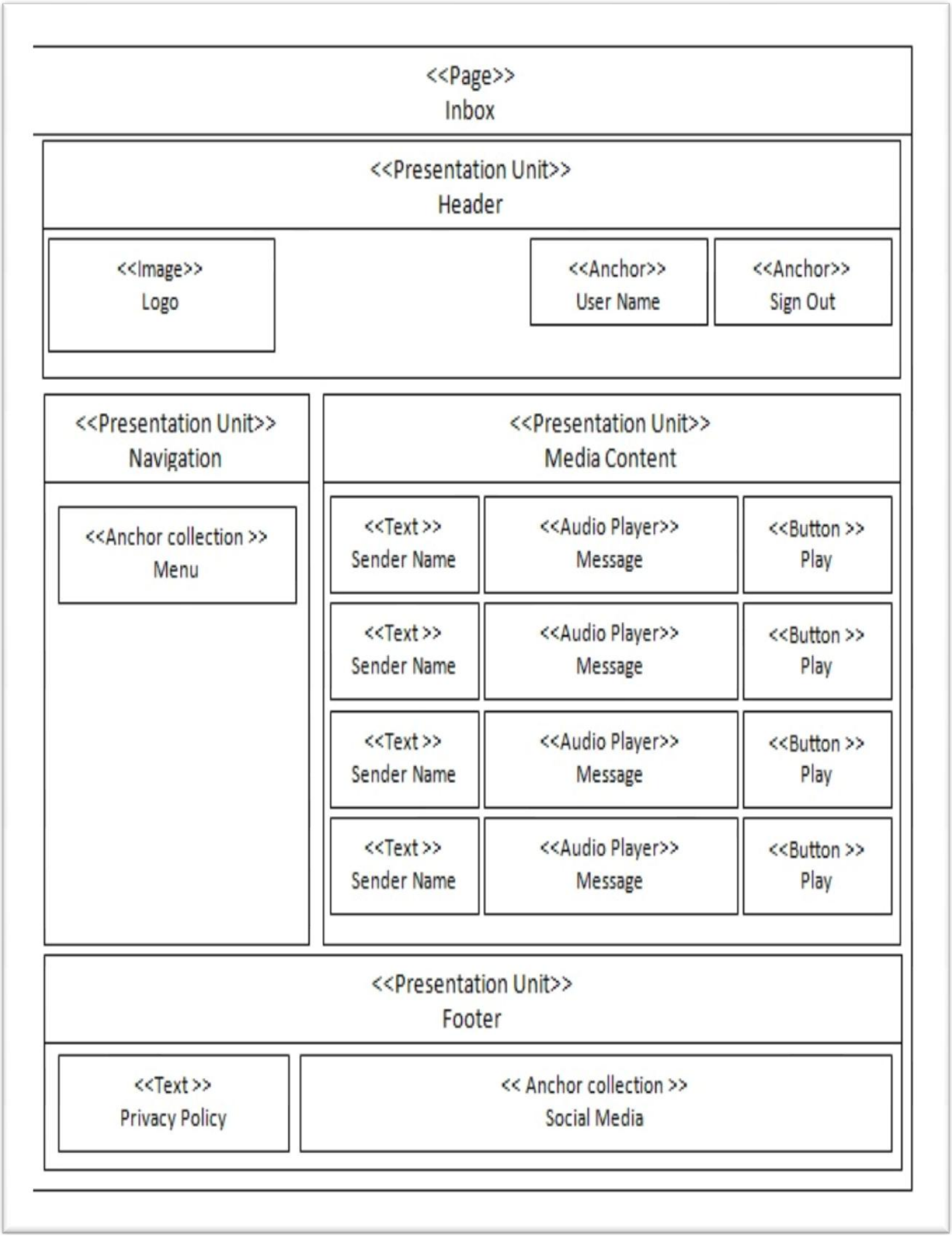


4.14. Presentation models

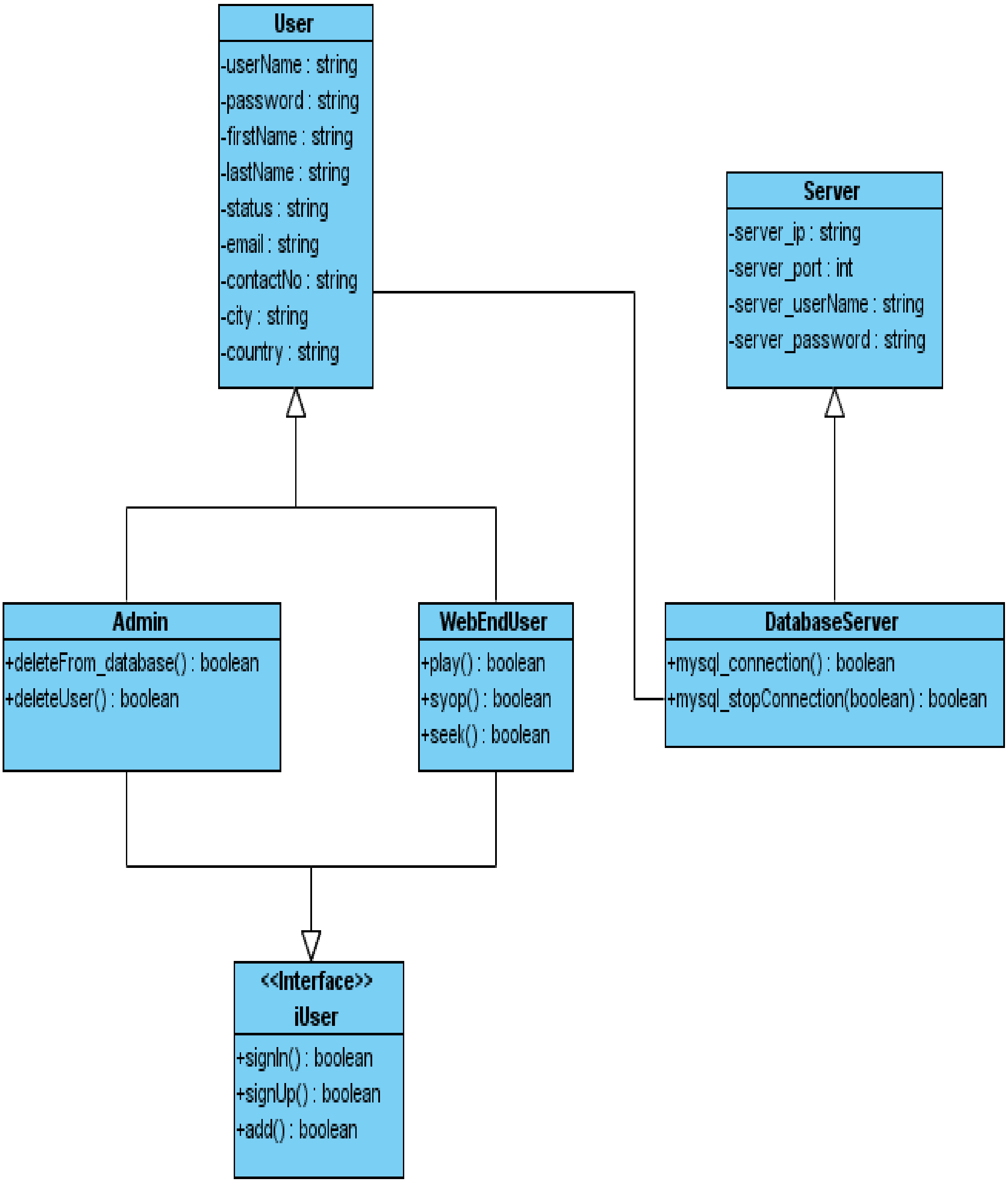
4.14.1. Presentation Model of Home Page



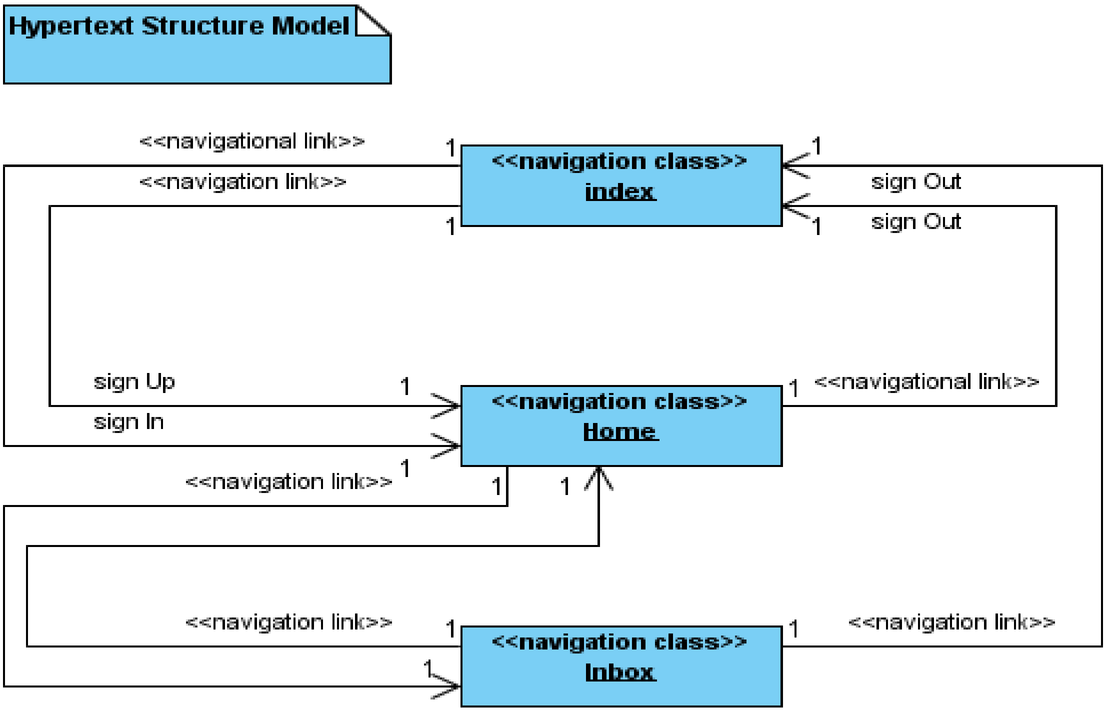
4.14.2. PRESENTATION MODEL OF INDEX PAGE



4.15. Hypertext Structure Model Class Diagram

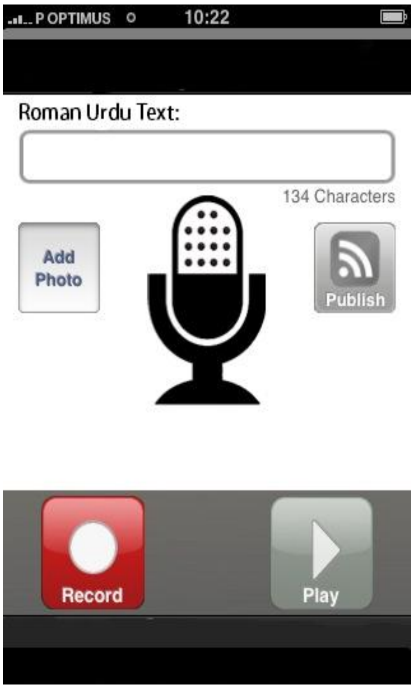


4.16. HYPERTEXT STRUCTURE MODEL



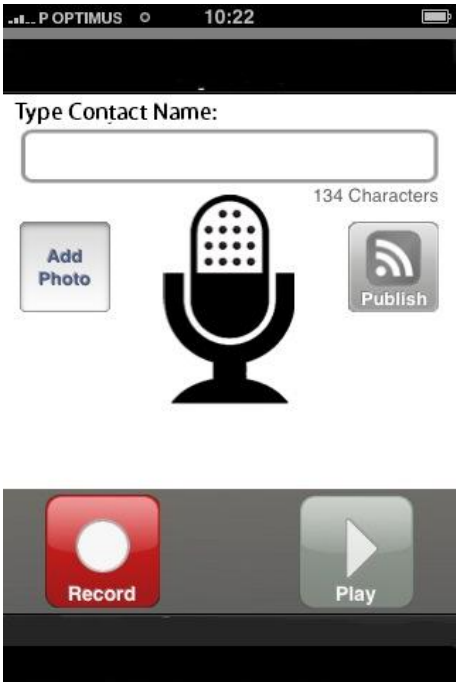
4.17. ANDROID END USER'S INTERFACES

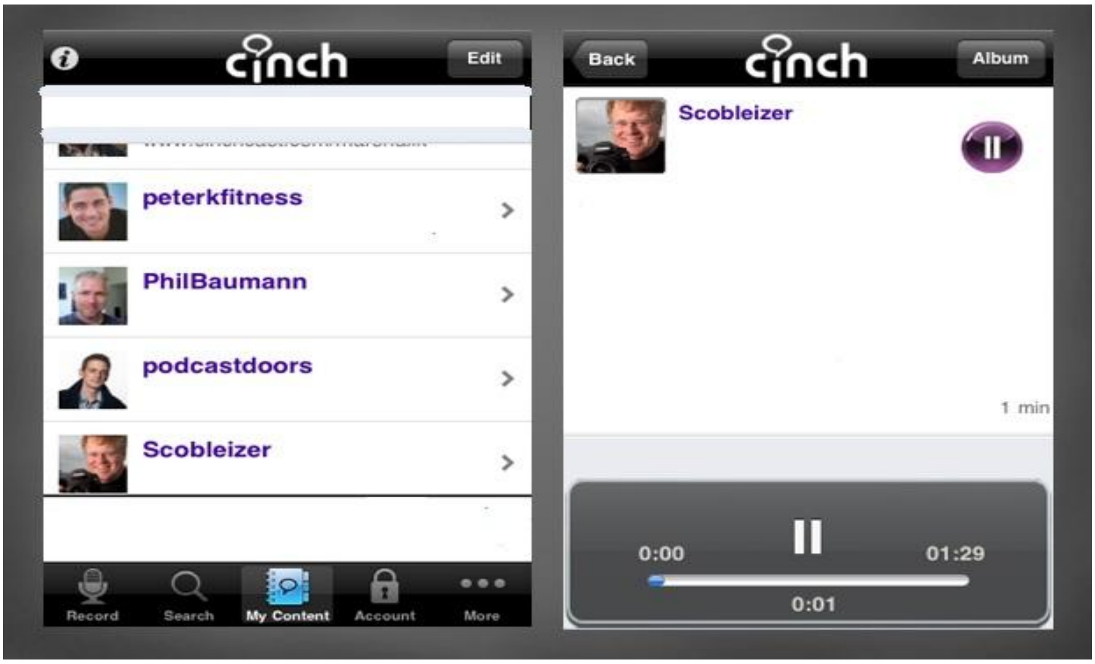
4.17.1. SPEECH TO TEXT APPLICATION



4.17.2. Dictation Application

Figure:
Message Streaming Page for the Dictation App





Chapter 5: System Implementation

5.1. Introduction

5.1.1. Purpose

The Implementation Plan describes how the Audio video Transmission engine would be setup using RED5 Server; how will it be deployed. The plan contains an overview of the applications developed on top of this transmission engine to show its functionality, a brief description of the major tasks involved in the implementation, and the overall resources needed to support the implementation effort (e.g., hardware, software, facilities, materials, and personnel).

5.1.2. System Overview

The project is based on the functionality of RED5 Server to be used as a transmission engine in collaboration with flex clients. Total of three applications have been developed to show how the transmission engine could be used.

Following are the three applications

- FaceTime
- Dictation app (Enterprise app)
- Speech to Text application

5.1.3. System Description

The Server used for the applications described below is RED5 Server and the client end created for the applications in web and smartphone are designed in Flex.

FaceTime
FaceTime is a peer to peer video chat application that allows users to chat via web application or via smart phones. This application makes use of the live streaming capability of RED5 Server.

Dictation (Enterprise) Application:
This application allows the employees to publish their video messages either via flex web application or via smartphone. This application uses the record capability of the server and uses MySQl Databases to store and retrieve information from the server and provide to users at web end.

Speech to Text Application:
This application is based on Android OS Clients and it uses the services of Google Server via Google API.

5.1.4. Assumptions and Constraints

Budget

Due to the budget constraints we couldn’t make the Server IP global, otherwise the Server would have been available anywhere on the go. However it does not effect the main scope of the project.

5.1.5. Technologies Involved:

We have used Flex clients as they are more compatible with Flash media server that RED5 is. These clients have been designed for web and smartphones.

RTMP Protocol:

The protocol used for communication between client and server is rtmp protocol. RTMP (except RTMFP) is a TCP-based protocol which maintains persistent connections and allows low-latency communication. To deliver streams smoothly and transmit as much information as possible, it splits streams into fragments and their size is negotiated dynamically between the client and server while sometimes it is kept unchanged: the default fragment sizes are 64-bytes for audio data, and 128 bytes for video data and most other data types.

```
public function init(streamName:String):void
        {
            var rtmpNow:String = "rtmp://192.168.1.12/VideoServer";

            selfStreamName = streamName;

            netConnection = new NetConnection;
            netConnection.connect(rtmpNow,"sampleParameter");
        netConnection.addEventListener(NetStatusEvent.NET_STATUS,getStream);
        netConnection.client = this;
        }
```

Snippet showing the connection establishing function at client end

```
private function setCam():void
        {
            cam = Camera.getCamera();
```

Audio/Video Streaming:

In order to stream audio video content via smartphone and laptops and other devices the control over microphone and camera at client end is necessary. Suitable parameters for noise reduction in audio and video were selected.

```
            if(cam)
            {
                cam.setMode(160 * 2, 120 * 2, 30, false);
                cam.setQuality(0, 88);
            }
        }

        private function setMic():void
        {
            mic = Microphone.getMicrophone();
            if(mic)
```

```
                        {
                                mic.encodeQuality = 10;
                                mic.setUseEchoSuppression(true);
                                mic.setSilenceLevel(0,20000);
                                mic.rate = 16;
                                mic.gain = 70;
                        }
                }
```

Snippets showing microphone and camera control at client end

Live and Recorded Streams:

For FaceTime Live streaming has been used and for the Enterprise application Streams are recorded that are further loaded into database. To decide this functionality at client end the live or record parameters are used.

For live streaming

```
        netStreamSelf.publish(selfStreamName, "live");
```

For storing streams at Server end

```
        netStreamSelf.publish(selfStreamName, "record");
```

Red5 Server:

At Server end for both the applications FaceTime and Dictation Application the class used is MultiThreaded Adapter class. The main functions of this class were overridden; following is a snippet of overridden function of Server class. This function is called whenever the client applications tries to launch itself and connects to the server.

```
@Override
        public boolean appConnect(IConnection conn, Object[] params) {
                System.out.println("Number of parameters passed: "+
                 params.length+" Value: "+params[0]);
                System.out.println("vidStreams peek: " + vidStreams.peek());

                                if (vidStreams.empty()) {
                        System.out.println("Quota full, rejected");
                        rejectClient("Quota full, rejected");
                }
                                return true;


        }
```

Databases:

```
// database server's connection
<?php

   define('DB_HOST', 'localhost');
   define('DB_USER', 'root');
   define('DB_PASSWORD','');
   define('DB_DATABASE', 'validation');
?>
```

```
// user login query
<?php
$qry="SELECT * FROM signup_tbl WHERE uname='$username' AND
password='".md5($_POST['password'])."'";
$result=mysql_query($qry);
?>
```

```
// user registration query
<?php
$qry = "INSERT INTO signup_tbl( fullname, uname, email, password, cnumber, country,
dob)
VALUES('$fullname','$username','$email','".md5($_POST['password'])."','$cnumber','$cou
ntry','$dob')";
          $result = @mysql_query($qry);
?>
```

Google API:

For Speech to Text application the api used to convert speech to text is hosted by Google. This functionality can only be accessed via Android Smartphones. The following code snippet shows the client accessing the google api and selecting English Language code as the language for text conversion.

```
private void startVoiceRecognitionActivity()
{
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.LANGU
AGE_MODEL_FREE_FORM);
intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speech recognition demo");
startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);
}
```

This function is called as soon as the application in Android smartphone triggers the activity.

Chapter 6: Testing and Result Analysis

6.1. DICTATION APPLICATION (WEB-END)

Test Case ID	1
Unit to test	**Login**
Assumptions	1. User not logged in. 2. User exists. 3. Login page is opened.
Test data	Username Password
Steps to be executed:	1. Visit Login Page. 2. Enter username. 3. Enter password. 4. Click login button.
Expected result:	Redirect user to the home page
Actual result:	User is redirected to the home page
Pass/Fail:	Pass

Test Case ID	2
Unit to test	**Login Out**
Assumptions	1. User logged in.
Test data	None
Steps to be executed:	1. Click on logout button.
Expected result:	Redirect user to the home page.
Actual result:	User is redirected to the home page.
Pass/Fail:	Pass.

Test Case ID	3
Unit to test	**Play public message**
Assumptions	1. User logged in. 2. User exists. 3. Homepage is opened.
Test data	None
Steps to be executed:	1. Play public message.
Expected result:	Play public message
Actual result:	Played public message
Pass/Fail:	Pass

Test Case ID	4
Unit to test	**Register User**
Assumptions	1. Clicked on Register Button. 2. User not already registered. 3. User name available.
Test data	1. First name. 2. Last Name. 3. User name 4. password 5. Email. 6. Phone number 7. City
Steps to be executed:	Click on register button. Fill registration form. Click register user button
Expected result:	Redirect user to the home page
Actual result:	User is redirected to the home page
Pass/Fail:	Pass

Test Case ID	5
Unit to test	**Play private message**
Assumptions	1. User logged in. 2. User exists. 3. Inbox page is opened.
Test data	None
Steps to be executed:	1. Play private message.
Expected result:	Play private message
Actual result:	Played private message
Pass/Fail:	Pass

Test Case ID	6
Unit to test	**Change password**
Assumptions	1. User logged in. 2. User exists. 3. Account settings page is opened.
Test data	Current password
Steps to be executed:	1. Enter current password. 2. Enter new password
Expected result:	Password changed
Actual result:	Password changed
Pass/Fail:	Pass

6.2. Dictation Application (Mobile-End)

Test Case ID	1
Unit to test	**Post message on wall**
Assumptions	1. Application is running 2. Connected with server
Test data	Message posting
Steps to be executed:	1. Connection with server 2. Start streaming 3. Stop streaming
Expected result:	Message posted
Actual result:	Message posted
Pass/Fail:	Pass

6.3. FACE TIME TEST CASES

Test Case ID	1
Unit to test	**Initialize yourself**
Assumptions	1. User is not initialized. 2. Application is running.
Test data	Connection with server
Steps to be executed:	1. Run application. 2. Initialize yourself. 3. Click initialize button.
Expected result:	Initialized and connected with server
Actual result:	Initialized and connected with server
Pass/Fail:	Pass

Test Case ID	2
Unit to test	**Connection with peer**
Assumptions	1. User is initialized. 2. Application is running. 3. Peer is initialized
Test data	Connection with peer
Steps to be executed:	4. Run application. 5. Initialize yourself. 6. Click connect with peer button.
Expected result:	connected with peer
Actual result:	connected with peer
Pass/Fail:	Pass

Chapter 7: Conclusion and Future Work

7.1. Conclusion

The Project covers two main applications that depicts the functionality of RED5 Media Streaming Sever. The projects main objective was to develop a transmission Engine to stream live or record streams. The Two applications FaceTime is developed to use the engine's live stream ability. The dictation application, Enterprise application shows the stream storing capability of Server and it's a complete packaged application that can be deployed at any Enterprise and customized accordingly with its web apps and a customizable we page to be maintained at the Enterprise level.The third application has been developed specifically for Android Smartphones that provides users an edge to use text on the move.

7.2. FUTURE ENHANCEMENTS

We plan to follow the following enhancements for our project.

7.2.1. For Diverse Smartphone OS platforms

Our primary choice for the development of applications at smartphone end was Android but since we have designed the clients in flash therefore it covers diverse OS platforms in Smartphone's but a different IDE or Flash builder experience will be required to deploy it. In the future we would also like to cover Iphones with respect to the trends in Smartphone market.

7.2.2. Noise Reduction

To improve the quality of audio video content streamed we would like to apply different algorithms to improve the media content quality and increase the current frame rate.

7.2.3. Video Format Support

There are a number of video formats available and there quality varies with bandwidth and processing power. We have supported specifically flv format we would like to broaden its scope to support other video format s too.

7.2.4. Speech to Roman Urdu Text

For Our Speech to Text application we would further want to use it to convert to Roma Urdu but for that we would have to use Sphinx libraries and dictionary but that itself is a project to make it work with the streamed sound via RED5 server.