The University of Hong Kong

Faculty of Engineering

Department of Computer Science

COMP7704
Dissertation Title
Mobile Application For Chinese Learning

Submitted in partial fulfillment of the requirements for the admission to the degree of
Master of Science in Computer Science

By
Lin Shaobi and Wang Xin
3035149958 and 3035149300

Dr. Vincent M.K. Lau
Date of submission: 01/08/2015

Abstract

This is a project focusing on how to help foreigners learn oral Chinese in their small piece of time by using the mobile phones. The project is down by a group of two people, in which one is to accomplish the implementation of Android part, and another one is to accomplish the iOS part.

We are meant to build up a powerful and practical mobile application by using speech synthesizer and speech evaluator in iFly services. Another technical problem is solved by using Parse server to build up a could database for the data management and user management.

As a result, we successfully develop two complete mobile applications in Android and iOS that can provide the educational value for learners to speak Chinese fluently, tiny game idea to bring some entertainments to them, together with attractive and interactive interfaces to make users fall in love with it to raise the efficiency of learning.

Contents

1. Introduction 1
1.1. Project Description 1
1.2. Project Objectives 2
1.3. Summary of Chapters 4
2. Requirement Analysis 5
2.1. Background Research 5
2.2. User Requirement Analysis 10
2.2.1.Description 10
2.2.2.Specific User Requirement 11
2.2.3.Use case diagram 14
2.2.4.Class diagram 14
3. System Design 16
3.1. System Structure 16
3.1.1.Interaction Model 16
3.1.2.System Flow Chart 16
3.1.3.Prototype Design 18
3.2. Technical Adoption 21
3.2.1.Speech Synthesis 21
3.2.2.Speech Evaluation 23
3.3.Development Tools 26
3.3.1.Development Environment 26
3.3.2.iFly Speech Engine 26
3.3.3.Parse Server 27

1. Introduction

In the recent society, communication is one of most important skills as bilingual, trilingual or multilingual people have great advantages. As a Chinese student living in HK, we need to have a good command of English, including written and spoken English. On the contrary, many foreigners come to China working or studying, and find out really hard to learn Chinese Pinyin or Chinese characters on their own. They usually have not enough time to take regular courses to learn Chinese step by step as we do in our childhood, and what they need most is to understand some daily expressions and learn some oral expressions which are the most useful. Having a good command of regular usage of spoken Chinese could help foreigners adapt to our environment and get to know our culture.
Hence, we'd like to design an useful mobile application for foreigners to learn to speak Chinese effectively and attractively in a series of organized courses.

1.1. Project Description

Our project mainly focuses on how to make foreigners to learn spoken Chinese faster and more efficiently in a powerful and interesting tool. So we want to combine the Chinese learning tool and the mobile game to help them to learn Chinese at any leisure time in their lives. The first step to learn one kind of language is to learn the phonograms, like Pinyin. For this reason, we'd like to design certain mobile game of rating based on the spoken Chinese to help those people want to speak Chinese fluently with perfect pronunciation.

As a result, our project is about the application in mobile phones that provides the educational value for users to learn to speak Chinese with structured courses, attractive interface and entertainment at the same time.

1.2. Project Objectives

To fulfill foreigners' need and attract more users, we are meant to follow the guideline of user interface design and take some of them into consideration, which is Shneiderman's Golden Rules of Interface Design[1], Jun Gong's Guideline for Mobile Application[2] and Nurul's Threes Layers Design Guideline for Mobile Application[3]. As a starting point, Shneiderman's guideline has been existed for some time while the last two guidelines are developed based on it, in particular about the mobile interfaces design.

Apart from that, we still deeply analyze a few similar published applications and extract the advantages and disadvantages about more than just interface design. Absolutely, we need to take more aspects into consideration, not only the interaction model, but also the practicality is another vital motivation.

Based on the background research of existing related applications, therefore, we could draw a list of objectives of designing a really practical Chinese speech learning tool without those obvious drawbacks in previous applications, and then find out the difficulties when we need to reach our objective.

Our application's final goal is to help foreigners to learn to speak Chinese in correct pronunciation and practice those conversations which possibly happen in people's daily life. Hence, our objectives can be concluded in five aspects:

1) **Attractive:** To make our application more attractive, the application should have friendly, beautiful and well-organized interface. It's the first factor to attract users.
2) **Structured:** Many of existing applications only consists the lectures by plaintext and images, and few of them uses the audio lecture to help users to get deeper understanding. In order to make the lecture more efficient and effective, the plaintext, images and recordings should be well-organized in each course.
3) **Interactive:** If users could interact with the application with diverse methods, users may be aspired more enthusiasms when using this application. Using interactive design in teaching could make users feel more like talking with a real person more than a cold machine.
4) **Educational:** An effective learning process must consist of one part of evaluation, just like the exams in our normal studies. After providing all course information and combining those information in some ways to help to get deeper understanding, examination is the necessary part to check whether users do good job in this lecture. If users perform not good, the suitable feedbacks and advises should be given and ask users to practice it again.
5) **Innovative:** We suppose to combine the general education app's advantages and game's entertainment together because users will feel the sense of achievements when passing each level. Moreover, we'd like to create a story of a personalized character and associate the course's content with those

daily events. This definitely could arouse users' interest to learn each conversation because they may be curious to apply this conversation in their life with their Chinese friends.

To satisfy previous objectives, we firstly will meet those difficulties: 1) how to organize the lecture in reasonable way; 2) how to add the voice interaction; 3) how to add the evaluation in the application and let users know how to improve. Hence, we need to figure out corresponding methods to solve previous problems, and arouse users' interests in short period of time.

1.3. Summary of Chapters

In Chapter 1, we mentions the project contents including the project description and objectives, together with the whole structure of this report.

In Chapter 2, the background research, the user requirements and analysis are described and discussed in detail, and the use case diagram and the class diagram are used to further illustrate this system.

In Chapter 3, we are going to do the system design of this Chinese learning application and then analyze the technical problems about speech synthesizer and speech evaluation.

In Chapter 4, the implementation of this application is independent in two platforms, so the specific method to implement the project may be different. This chapter is done separately.

In Chapter 5, the conclusion of the whole report.

2. Requirement Analysis

2.1. Background Research

In the topic of our project, we found that there are several similar Chinese learning applications in different platforms, such as Android and iOS. There are many types of applications or games in this area, but are mainly suitable for babies to learn Chinese or there are some obvious drawbacks. After doing some researches about this type of educational applications in App Store, we could roughly conclude that they have something in common:

- ***Language learning application is few and not very popular.***

There are just few language learning applications on market and even less in top list. We did the research and found that there were only 3 educational apps among 180 top apps in App Store as the pie chart shown below in Figure 2-1.

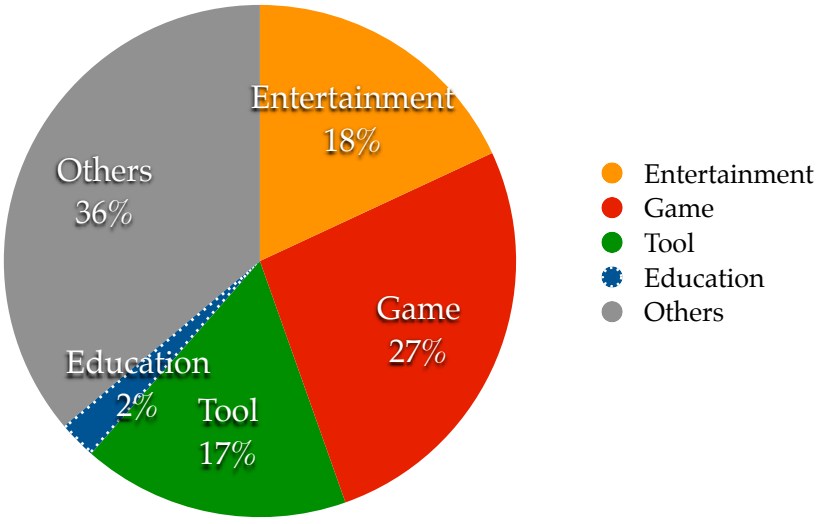

Figure 2-1: Top apps in App Store

- ***Existing language learning applications are hard to attract users with poor UI design.***

Many learning applications don't pay much attention on the user interface, but it is no more than UI attract and arouse user's interest firstly.

- ***The interaction method with users is limited and is not helpful for users to learn to speak one kind of language.***

The existing language learning application did bad in the interaction with users. Majority of them just show simple text on the screen related to corresponding images to help users memorize, but it is not enough. Without the help of the examinations or tests, users are hard to know how well they do and make some progress based on the feedback.

In order to design an structured, attractive and practical application to help foreigners or people cannot speak Mandarin well. To be specific, we further do some research for these similar applications and analyze some of those representative educational applications to better illustrate those drawbacks and get a better understanding for the project.

A. Fun Chinese

https://itunes.apple.com/app/fun-chinese-er-tong-zhong/id547571511?l=zh&mt=8

It is a popular educational applications for iOS, and there are diverse tiny games of different themes. The game layout is great with attractive graphics and user-friendly interface as Figure 2-2 shows, and the visual and audio effects are helpful

for users to memorize the vocabularies. Moreover, in every game, there are different

Figure 2-2: Screen captures of Fun Chinese

levels from easy to hard, and players can practice more when they come to hard level.

Take on game of "color course" for example, as Figure 2-3 shows, the user need to click the card to turn it over and see some pictures with corresponding colors and corresponding audio hints. This is a game of memory to help child recognize different colors and corresponding pronunciations.

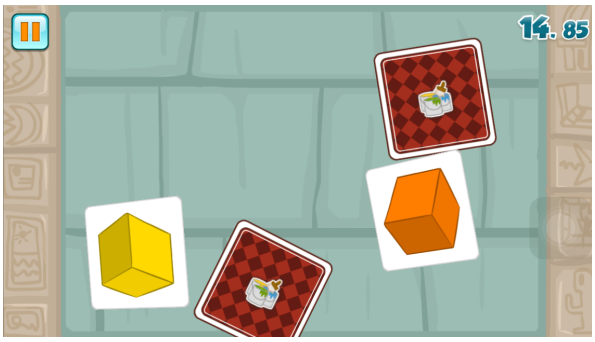

Figure 2-3: The interface of "color course" in Fun Chinese

The short comes of the application are obvious:

- The application is mainly designed for kids aged 3-10, and those games may be useful for children but may be a little stupid for adults.

- The courses are limited within some aspects and only contain the Chinese vocabulary learning. In real life, people not only see or hear a simple word like this. So this might not be practical for people in daily life to learn to speak Chinese fluently with correct pronunciation.
- No any other choice for different level. Some experienced players may want to skip some easy part and learn more.

B. Penyo PAL

https://itunes.apple.com/app/lian-xi-zhong-wen-pu-tong/id531843574?mt=8

This is an application for learning basic Mandarin Chinese vocabulary with fun flashcard game. Each level of the game helps the player to practice new Chinese words with flashcards that feature pictures, pinyin, characters and audio.

For example in Figure 2-4, in a given class, the player need to hear the standard pronunciation of one word and choose the right one, see the Pinyin of one word and choose the right one or see the Chinese character of one word and match the right one. This process repeats several times to ensure the player remember it.

Figure 2-4: The interface of class in Penyo PAL

However, as far as we can see, there still may be some possibilities to improve:

- There is no level choice for the player, which means those simple courses may make some experienced learners to feel boring.

- For the diversity of the application, the database of Chinese vocabularies may need to be enlarged.
- This application also doesn't cover the the part of phrases or sentences which can help users have real communication skills in their daily life.

C. Learning of Chinese Pinyin

http://apk.gfan.com/Product/App917515.html

Learning of Chinese Pinyin is one of the representative app that teaches users from the fundamental part — Pinyin, but actually, users usually use mobile phones no more than 5 minutes each time which is totally not enough for a complete lecture to show all points to remember.

Figure 2-5: The interface of Learning of Chinese Pinyin

The application consists of three part: the learning of initial consonant, the learning of compound vowel and the learning of whole syllables. Figure 2-5 shows the main interface for users to do the recognition job. When the user choose one of the part, the user interface will show the corresponding pinyin. Besides, if user choose one of the pinyin, the app will display a word made up of the pinyin and the way of writing the character.

The short comes of the application are obvious that:

- It's a little simple no matter the functions or UI design, and it lacks the entertainment. Users may feel boring after a short period.
- The app just display the pinyin characters, without teaching the pronunciation. It may cause the inefficiency of learning.
- People only do recognition job during learning. It may lead to the problem of memorizing because of the lack of evaluation of users' performance.

2.2. User Requirement Analysis

2.2.1.Description

Our aim it to develop a Chinese speech learning system (CSLS). Users firstly can register an account of this system, then login to the main screen of this system and start learning. A new user is set to only unlock the first unit in first course within first level by default. Every time user passes the last unit, the next unit will be unlocked and appear there. Within each unit in this learning system, there are two main functions, one is called CLASSROOM, and another one is called CHALLENGE.

In the CLASSROOM part, users can read the sentence by sentence with corresponding recording in one conversation. Recording can be played for several times and users can also go back when they want to review previous one. In the final segment after learning each sentence in detail, people will see a recognition test based on the previous conversation.

In the CHALLENGE part, users will firstly see the preview of whole conversation and prepare on their own. After pressing start to start

challenging, user need to play one role in the conversation first and play another role in the conversation again. After that, a result screen will show them how they perform in this test.

2.2.2.Specific User Requirement

The detailed user requirements for the CSLS system are as follows:

I. A new user can register an account to enter this system. They need to input username and password, if those inputs are invalid or existed, they need to input again. After successfully registration, the system will store the information in the database. Then the new user can be accessible to the first unit in Course 1 in Level 1 by default.

II. When a user want to learn to speak Chinese, they need to use the existing username and password to login. If the user enters the correct password, the system will allow him to login and access those units that are unlocked for him. After logging into the system, the user will see the main level screen to choose one unlocked level(vertical). If it is the first time, the system will ask the user to download all related coursed and units before entering. After choosing one level, user will enter the main course screen and see a vertical list of courses within this level. Then user need to choose one course to enter main unit screen in a horizontal scrollview. Within each unit, there are two functions, CLASSROOM and CHALLENGE.

III. Each lecture consists of several sentences within one conversation, together with several pictures for multiple choice. Manager could upload

and organize these lecture information in the database which users are learning.

IV. After selecting CLASSROOM function, the system will show another screen of several segments in one unit. Each segment consists of one sentence in one conversation. Segments are settled in order as a top progress bar shows where it is. In the last segment view, the user need to pass a simple multiple choice after finishing the listening question. In this multiple choice, the user will see the question on top of the screen and listen to short conversation or just sentences' recording, and on the below

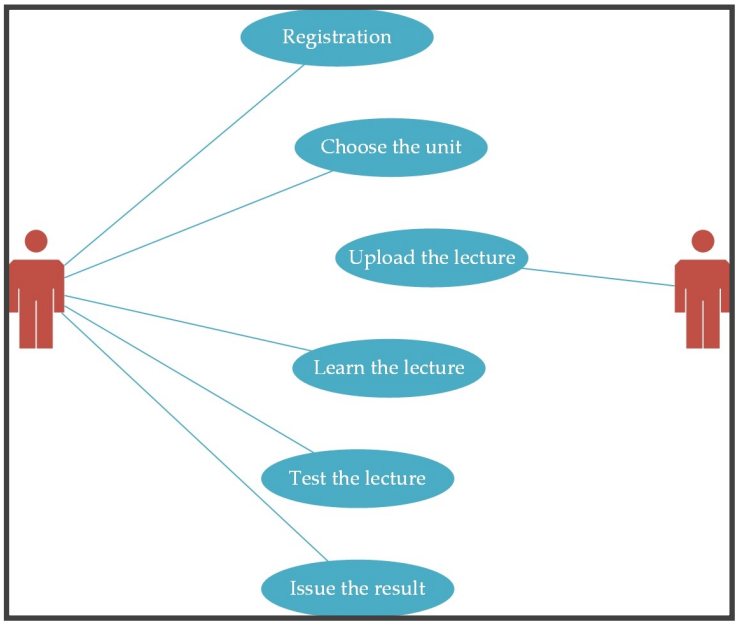

Figure 2-6: Use case diagram of Chinese speech learning system

of the question, there are four pictures to choose. The user possibly chooses the wrong answer before finally choosing the correct pictures related to the question. Only when the user passes this simple test, there will be one text showing the script corresponding to the recording and the user is able to finish current learning and enter to the CHALLENGE part.

V. After selecting CHALLENGE function, the system will show the preview of whole conversation to the user first. Then the user can click “start” to start challenging just like have a conversation with the system. The system will show count down view as the cue before starting. After that, the system will start and play the role on the left, then say the first sentence. Once the system finishes talking, it is turn for users to complete their part and say what the screen shows. So there is another is cue called “switch role” which tells the user to play the role on the opposite in the first time conversation. If the user complete the challenging part twice that means the user has played two of the roles in this conversation, the system will automatically show the result screen to the user.

VI. On the result screen, the system need to parse the results in the previous challenging part. Then, the result of pass or failure, the final score of the whole test, each score for each sentence and the highest score in this level(may be from other user’s) will be well-organized in this view. Moreover, the system parses the detail in each sentence to mark the word red or green and show the corresponding score on the right. Also, the user can listen to their recording by clicking that sentence.

2.2.3.Use case diagram

After analyzing the specific user requirements in the previous part, we could identify the actors and use cases in our system as Table 2-1 shows:

Actor	Use Case Name	Use Case Description
User	Registration	Anyone who wants to use this application should register as a user.
User	Choose one unit	Users choose the unit they want to learn.
Manager	Upload the lecture	The manager uploads the lecture information in every unit.
User	Learn the lecture	Users learn the lecture in the learning system.
User	Test the lecture	A test is used as the evaluation of user's performance.
User	Issue testing result	The system will parse and issue the testing result to user.

Table 2-1: Identifying actors and use cases

On the basis of the identifying of actors and use cases, we could easily construct the use case diagram as Figure 2-6 shows. There are six use cases in total.

2.2.4.Class diagram

After documenting typical and alternative course of event in those use cases, we find out those potential classes by discovering nouns from use case documents. And then select proposed classes from potential classes to draw

the class diagram. Hence, Figure 2-7 represents the class diagram of our Chinese speech learning system.

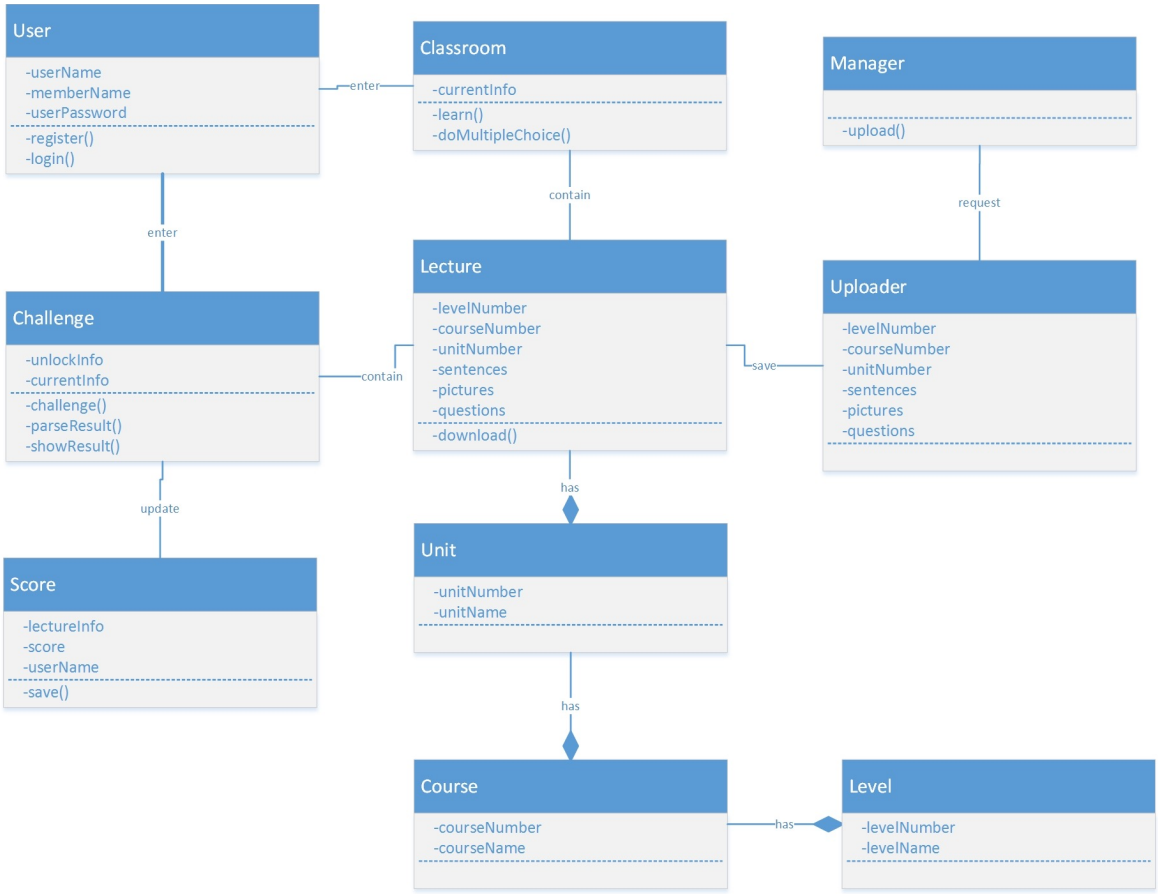

Figure 2-7: Class diagram of Chinese speech learning system

3. System Design

3.1. System Structure

3.1.1.Interaction Model

We implement our application both on Android and iOS platform, with Parse as the server and iFly as the speech engine. The courses information store in the Parse, but the course text will be deal with iFly to do speech synthesis and user's voice will be evaluated by iFly speech evaluation, then the result will be returned to the user.

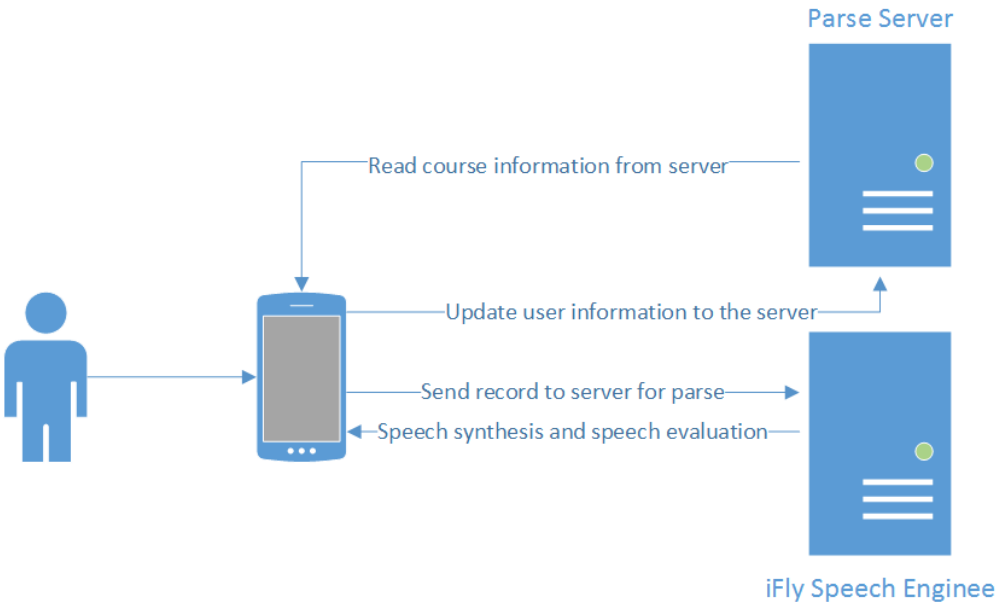

Figure 3-1: The user-system interaction graph

3.1.2.System Flow Chart

The system is designed as the "Level - Course - Unit - Segment" scheme. Each course is designed as a scene and each unit contains two parts: Classroom and Challenge. In "Classroom", there are five segments. The first four segments is the detail contents for the unit and the fifth is a multiple choice. The multiple

choice has four pictures, the system will play an record, user should choose the right answer according to the listening material. If user choose the right answer, system will let user enter to the “Challenge” part. In “Challenge” part, system provide a conversation for user to practice. At first, the user will play the “Men” role and read the 1st and 3rd sentences. After that, there will be a “Switch Role” model, then user will change the role as a “Woman” and read the 2nd and 4th sentences. When user finish the “Challenge”, speech engine will evaluate user’s records and give the user mask of his/her pronunciation.

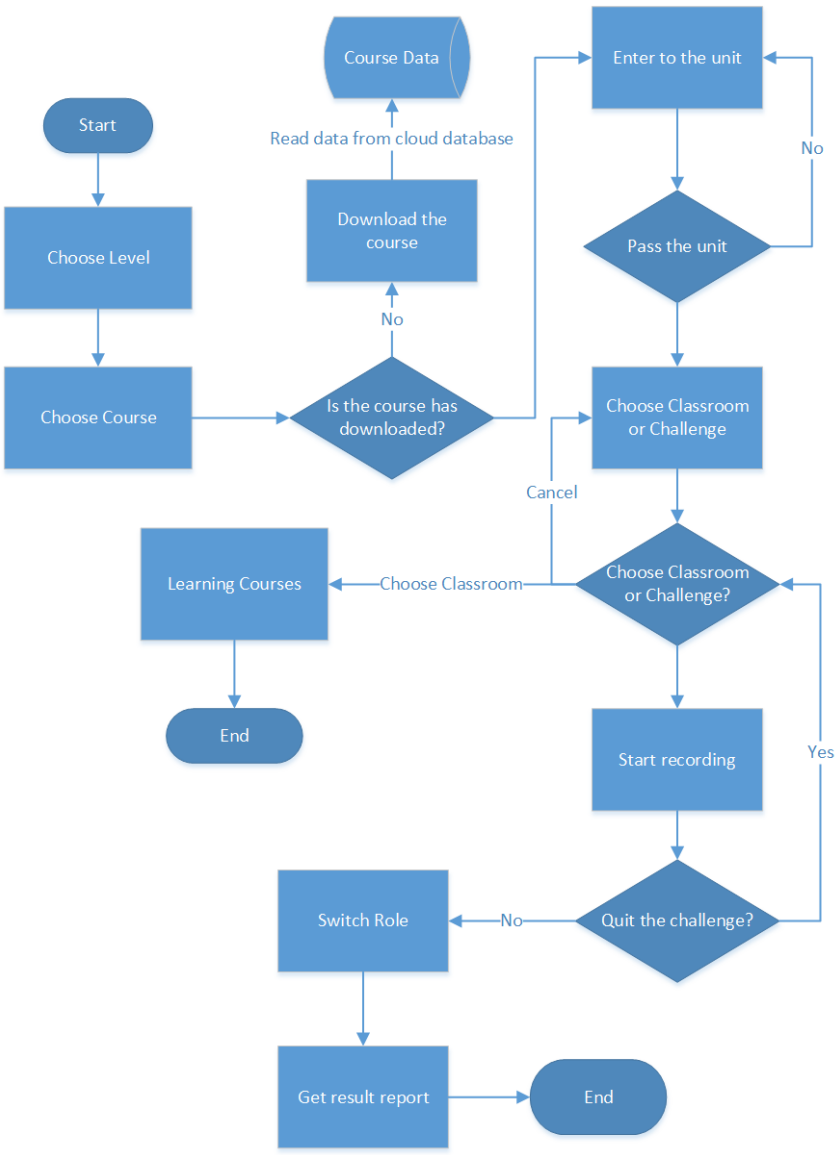

Figure 3-1: The system flowchart graph

3.1.3.Prototype Design

In this stage, we mainly use the photoshop to design the prototype. And in this part, we mainly show the prototype design of Unit page, Classroom page and Challenge page.

- **Unit Page**

Figure 3-2: The unit page user interface

Then let us take the "Hello" scenario for an example. Each scenario contains various courses. The upper dots display the courses number while the green dot meaning the current course the user is in. Same as the above explanation, only user passed the previous course then he can go to the next.

The title "Hello 你好" represents the name of the course, and the stars are the outcome of "CHALLENGE" (we will do more detail explanation

in the following description). User can enter to the course class and challenge with clicking the "CLASSROOM" button and "CHALLENGE" button respectively.

- **Classroom page**

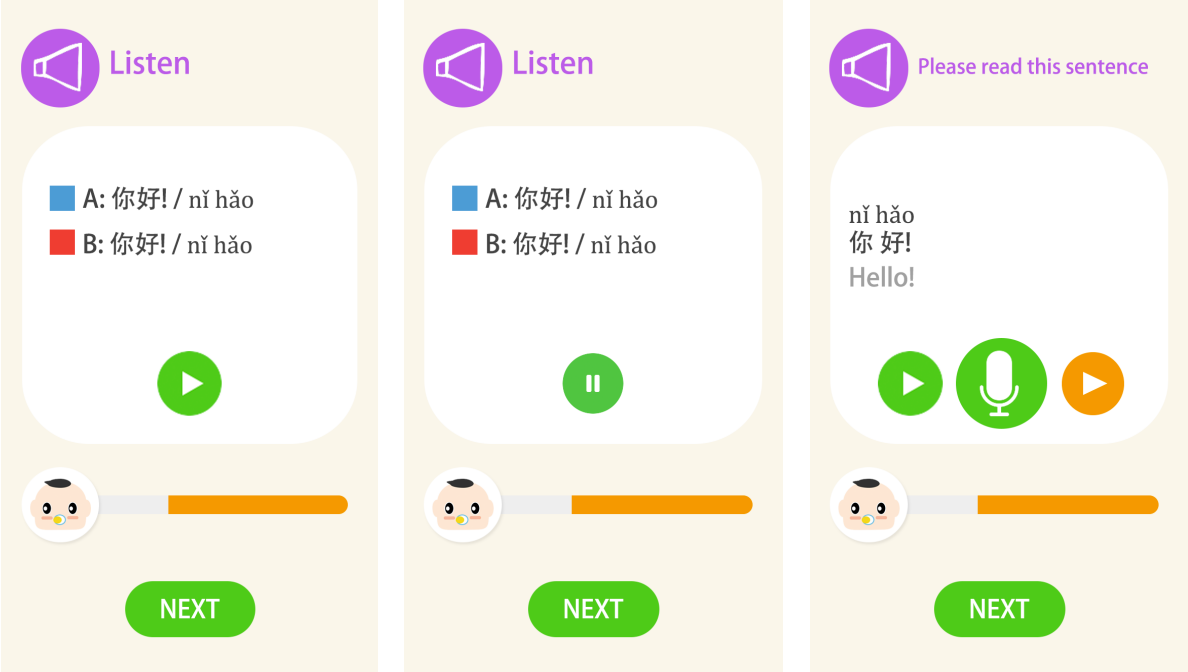

Figure 3-3: The main user interfaces in classroom

Each course contains various topics. In the beginning training topics, the system plays the sentences which will be used in the challenge stage and these topics are displayed with conversation pattern. Moreover, each sentence is followed by Chinese pinyin to help user pronounce. When user click the play button in the first graph, system will pronounce this sentence and users can stop it whenever they like (the play button will be the pause stage in second graph). The "NEXT" button can lead user to the next topic. After training topics, user will go to the exercise topics. The interface (third graph) will display the sentences occurred in the training topics. The green play button plays system's pronunciation, user can record their own voice by

click microphone button and click again to end the record. Once users end their record, the orange play button will show on the interface, users can play the orange button to listen their voice. Users can do repeat records to train their pronunciation. The progress bar shows the progress of topics, the passed topics are light grey while the rests are orange.

- **Challenge page**

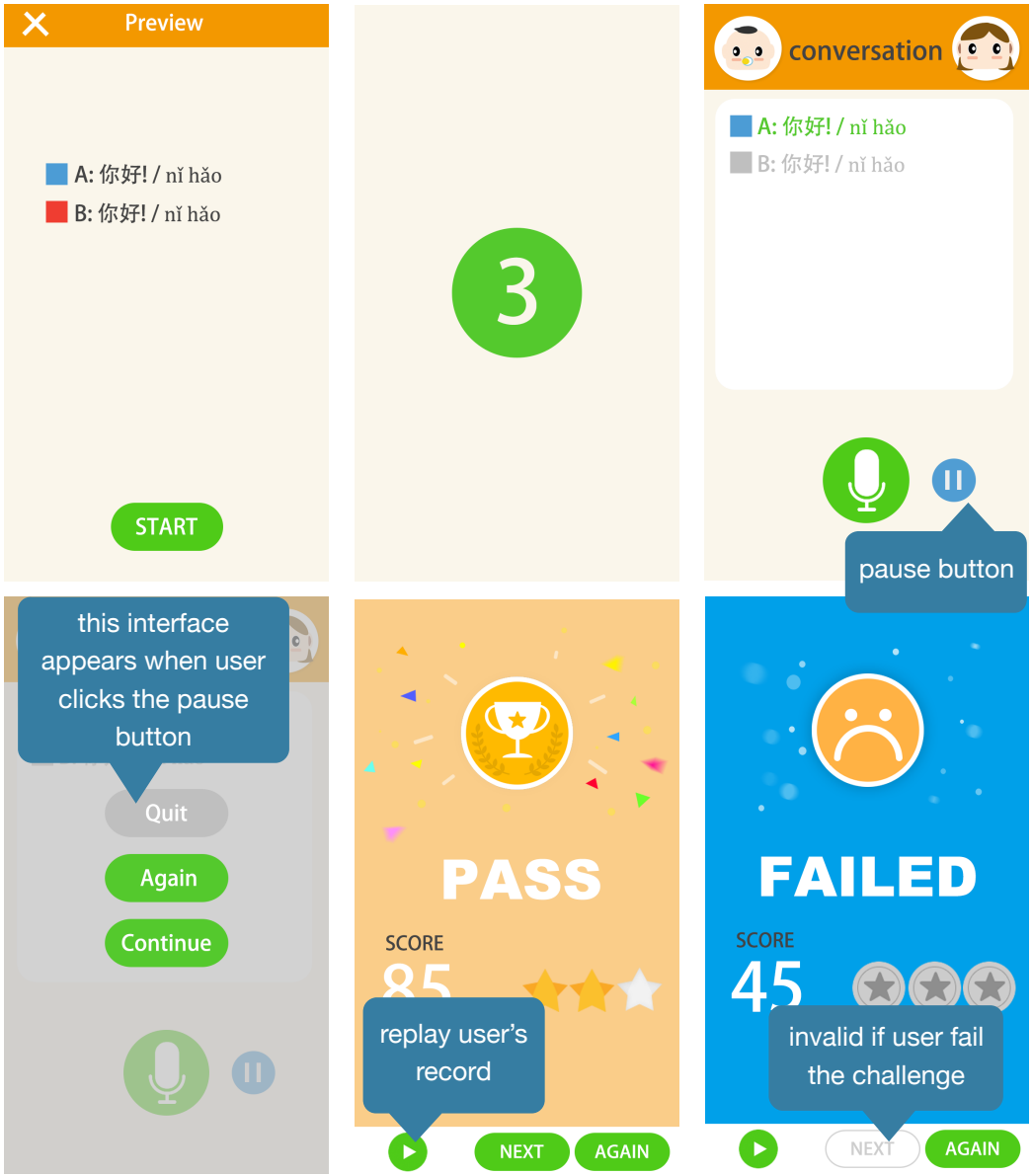

Figure 3-4: The main user interfaces in challenge

When users finish the classroom, they can enter to the challenge. At first, user will see a preview of following conversation contents, if users choose to

start challenge, the interface will display a count down dialogue with number sequence “3 2 1” (the 2nd graph of this part). The conversation is more like a role play, the system will play the role as “A” at first and the sentence which is being read will highlight and on the top. When it’s user’s turn, the microphone will appear and users can click it to record their pronunciation. There is a pattern called “switch role”, which means the conversation will repeat again but user should playing the role of “A” instead of “B”. This method can promise user to do a complete conversation. If users click the pause button on the right bottom of third graph, they can choose to quit, again the conversation from the start or continue the current conversation. When users finish the whole challenge, they will get their mark. Users can not access into the next level unless they pass the challenge.

3.2. Technical Adoption

3.2.1.Speech Synthesis

iFly speech synthesis mainly adopts the HMM speech synthesis technology. Compare with the traditional speech synthesis method, which is by large-corpus based concatenation method, the TTS system depends on HMM model is more easier on system structure. HMM model doesn’t need any linguistics guidance and practice and the construction time is shorter without manual intervention[4]. Since the system use parametric synthesis method, the results of the speech synthesis are changeable and can be adopted in different voicer

and voice style. It's the primary choice of embedded speech synthesis technology.

HMM speech synthesis mainly has two phases: practice phase and synthesis phase. Firstly, we use parameter extraction on the corpus (including spectrum parameter and frequency parameter)[5]. HMM's observation vector can be divided into two parts: spectrum parameter and frequency parameter. Spectrum parameter use the continuous probability distribution HMM to do build. Frequency parameter use MSD-HMM[6] to build model. In synthesis phase, do contextual analysis and transfer the word text to the model unit sequence. Then produce algorithm according to the HMM speech synthesis method's parameters, and get objective speech parameters sequence with considering static parameters and dynamic parameters. Finally, the synthesis speech can be produced by speech synthesis machine.

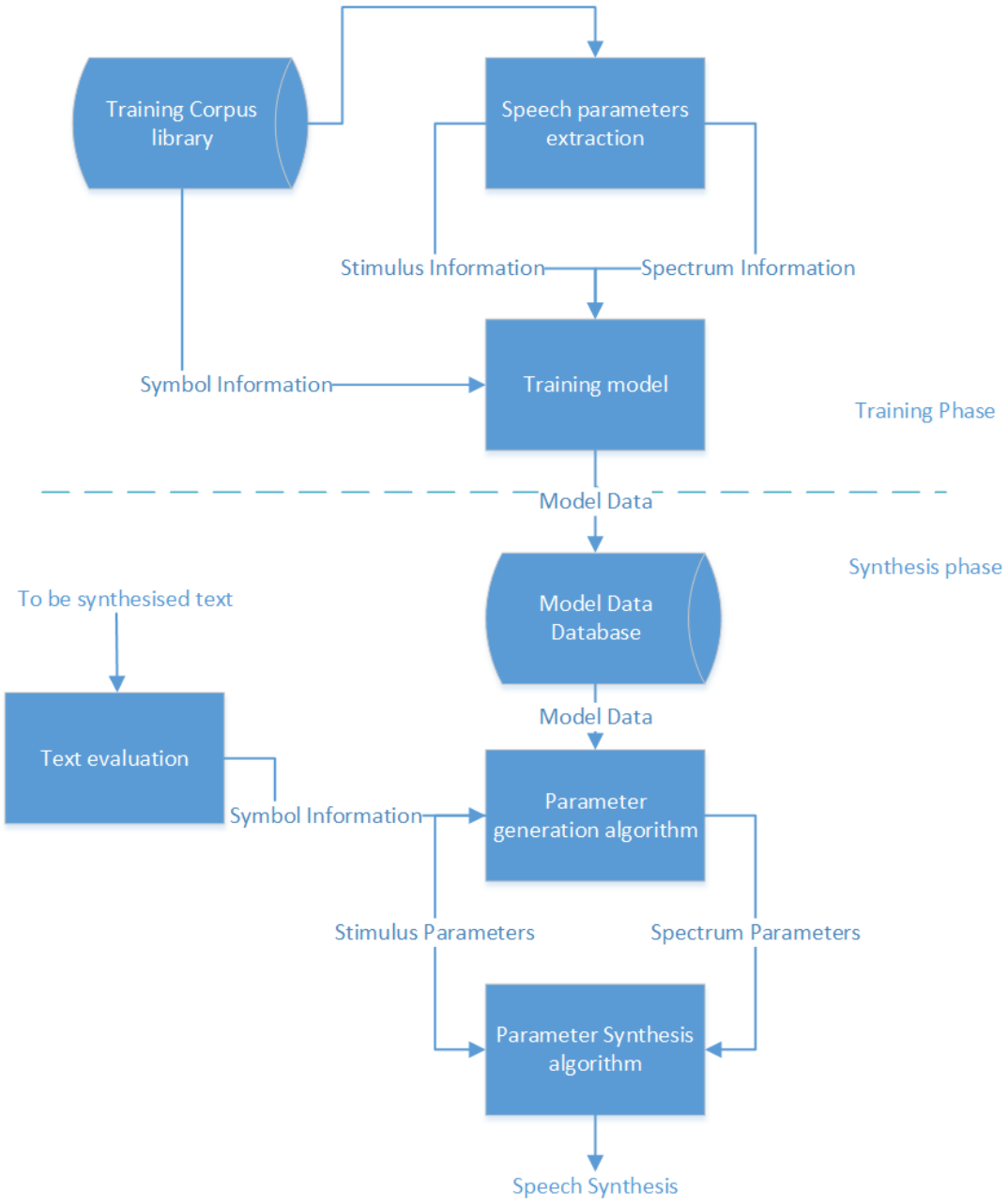

Figure 3-5: The speech synthesis flowchart

3.2.2.Speech Evaluation

iFly speech evaluation mainly uses GPU cluster and DNN algorithm. DNN is multiple layers of Neural Network[7]. Even though neural network appeared long time ago, but since the computation, the layers of neural network can not be improved. With the development of neural network theory and the

improvement of computation ability, especially the appear of GPU, DNN has shown its remarkable influence.

DNN has replaced GMM to be used in acoustics model. In acoustics model, the upper part is HMM structure, HMM's structure and state transition probabilities are the training results of HMM-GMM[8] model. In middle part of acoustics model is DNN, DNN model decide output probabilities of HMM. The layer numbers of DNN model will not be lesser than 5 and each layer is made up of thousands nerve cell. The half bottom of DNN model is the inputing, and the inputing is multi-frame, not a frame of GMM model. During the process of speech recognition, every part of the record will be extracted as an observation and the observation will be computed output probabilities (which means compare the similarity between different pronunciation) according to the HMM state. The largest output possibility path will be as the final result.

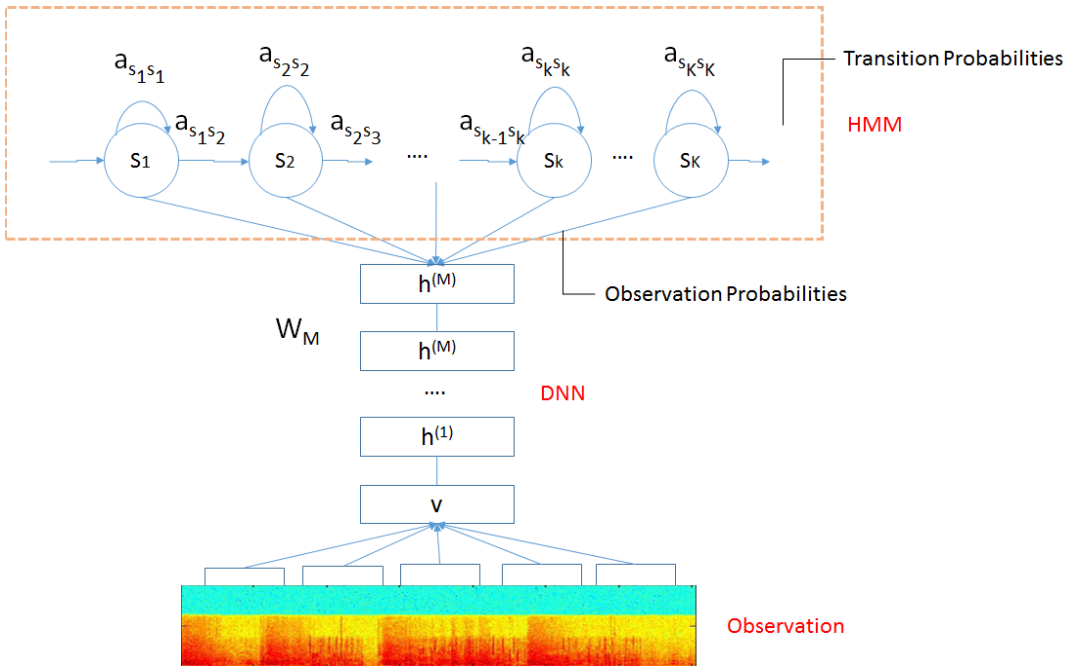

Figure 3-6: use DNN as the acoustics model

The following diagram show the training process of DNN acoustics model. Before the training of DNN model, the HMM-GMM model will be designed in order to do forced alignment. The result of forced alignment will be the training samples of DNN to be submit. The process contains two parts: the first part is pre-train of neutral network which based on GPU, the second part is use BP algorithm to do fine tuning of neutral network and then get the final model.

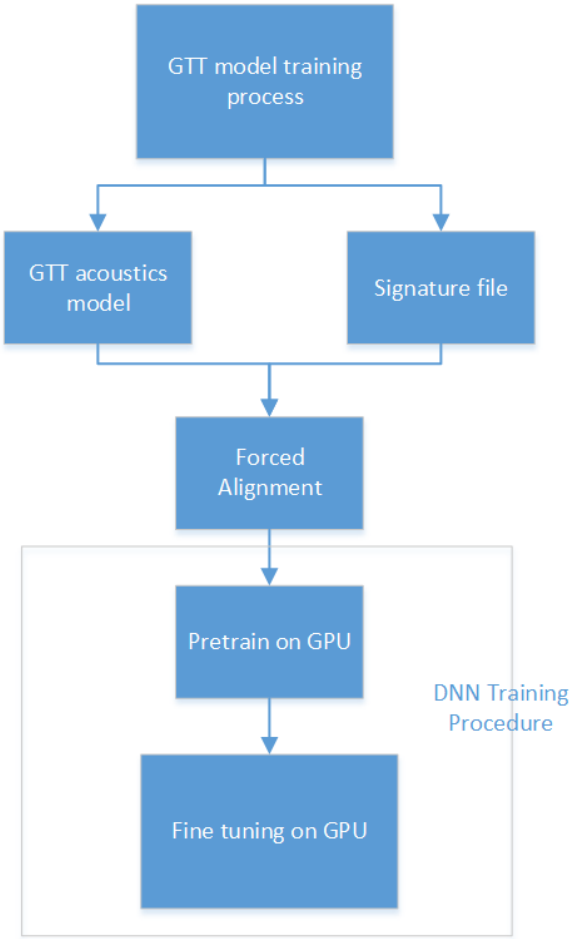

Figure 3-7: DNN acoustics model training procedure

DNN network has big data of parameters to learn, each network has millions of parameters and input of next network is the output from previous network. Generally, train a acoustics model needs nearly 2000 CPU to run about a

month. Besides, since acoustics model use special DNN: each layer's nerve cell depends on all the nerve cells on previous layer. However, if distribute different layers to separate servers to do training will cost huge network overhead, so the DNN training procedure should use GPU to do optimize. The training speed will increase nearly 2000 times than on the single machine, which make the training of DNN model to be practice.

3.3.Development Tools

3.3.1.Development Environment

We development on Android and iOS platform.

Platform	IDE	Language
Andoird	Eclipse	JAVA
iOS	Xcode	Objective-C

Table 3-1: the information of the development environment

3.3.2.iFly Speech Engine

Now there are so many speech engine that can provide TTS (Text To Speech) function to synthesize normal recording like a real person, such as WXVoice Engine, Microsoft Speech Engine, Google Speech API, and iFly Speech Engine. However, in these speech engine, iFly's precision is higher than other speech engines and has been applied into a few prevalent apps like Sina Weibo, Gaode Map, cTrip, etc.

iFly provides a series of speech services and corresponding technique supports. iFly supports development platforms including Android, iOS, Windows Phone 8, Windows, Linux, Html5….especially Android and iOS, the

thorough open API allow the developer to develop a complete speech application. iFly's speech synthesis engine can provide continuous speech synthesis service which is capable of converting any text information into highly natural and smooth speech at a quality. Beside, the most important factor is the fact that iFly is the only engine that provides speech evaluation function, the engine can analyze speaker's record and return the result with a XML file, including the result of each word, each syllable and each phoneme.

3.3.3.Parse Server

Parse is a perfect tool that provides cloud database, push notification, services and analytics tracking for mobile applications, it focus on creating a great user experience and stop worrying about server maintenance and complex infrastructure.

The "core" service Parse provides the cloud database service. Parse handles everything the user need to store data securely and efficiently in the cloud. Store basic data types, including locations and photos, and query across them without spinning up a single server. Besides. Parse Local Datastore provides the offline database, the user should get the data from network then save the data on local. When the user need to query information, the data can be captured from local database replacing send a network request. This can save a lot of time and decrease the network overload.

So, we can use Parse Cloud as our server and share the same database which could handle the entire backend automatically.

The University of Hong Kong

Faculty of Engineering

Department of Computer Science

COMP7704
Dissertation Title
Mobile Application For Chinese Learning

Submitted in partial fulfillment of the requirements for the admission to the degree of
Master of Science in Computer Science

By
Wang Xin
3035149300

Dr. Vincent M.K. Lau
Date of submission: 01/08/2015

Acknowledgement

My supervisor Doctor Vincent M.K. Lau is the one I really want to appreciate, because he inspired me to dig out the user's need in Chinese Learning, and then encouraged me to develop an application for Chinese speech learning. I appreciate all the useful resources and valuable advices that he gave to me, without which I could not accomplish this project smoothly and successfully.

In addition, I also want to thank for the help from my partner Lin Shaobi who assists me in many aspects about programming and inspire me when I feel frustrated. Those days we keep working together, find difficulties together and solve the problem together, are the impressive memory when I am at the university of Hong Kong.

What's more, also want to convey my appreciation for Dr. Beta C.L. Yip and my friends who gave me valuable suggestions to make me do better job in this project.

Contents

4. Implementation.....1
4.1. Overview 1
4.2. Tools.....1
4.3. Deployment.....2
4.4. Implementation of Application 4
4.4.1.The Structure of Application 4
4.4.2.Functionality Implementation.....5
4.4.2.1.User Management 6
4.4.2.2.Lecture Downloading 8
4.4.2.3.Lecture learning in Classroom.....10
4.4.2.4.Lecture quiz in Challenge 13
4.4.2.5.Quiz result display 17
4.4.3.The Database of the Application.....22
5. Conclusion 27
5.1. Project outcomes.....27
5.2. Future work.....28
Reference 30

4. Implementation

4.1. Overview

After analyzing the user requirements and designing the system in flow chart, we are going to use mentioned techniques to implement those indispensable functions to satisfy user's needs. In the first place, the system consists of client part and server part, and the server is deployed on Parse server conveniently. The Parse Cloud handles my entire backend so we don't need to worry about server maintenance, databases management, or performance. Also, the client part is implemented in two platforms (Android and iOS) separately and similarly, and I accomplish the iOS part. In our program, Parse takes over the data storage of user information, lecture information and score information about the challenge function in this program, at the same time, we mainly focus on implement the applications by using iFly Speech Engine on smart phones that are easy for users to use and connect it to the server. Hence, the following section will then introduce those thins in order: all the tools that I use, the deployment before programming, the structure of the whole program, the functionality implementation in detail. What's more, some pseudo-codes and charts will help me to further illustrate my logical thinkings and relationships.

4.2. Tools

All the tools that I user can be divided into two parts as follows: hardware and software.

- ***Hardware:***

MacBook Pro (Retina, 13-inch, Mid 2014) + OS X Yosemite 10.10.4

iPhone 5S + iOS 8.4

- ***Software:***

  Xcode + iPhone SDK + iOS simulator

  Parse SDK for iOS

  iFly Speech Engine SDK for iOS

4.3. Deployment

Since it is the first time to use Parse and iFly Speech Engine, there are few steps of deployment before programming.

Thus, the procedure of deploying Parse is like this in Table 4-1:

1. Download & unzip the SDK （Xcode 5.0+ and iOS 6.0+）
2. Add the SDKs to my app (Parse.framework and Bolts.framework)
3. Add the dependencies
Click on Targets -> “CnMate” -> 'Build Phases' tab -> expand 'Link Binary With Libraries' item.
Click the + button in the bottom left and add the following libraries:
AudioToolbox.framework/CFNetwork.framework/CoreGraphics.framework/
CoreLocation.framework/QuartzCore.framework/Security.framework/StoreKit.framework/
SystemConfiguration.framework/libz.dylib/libsqlite3.dylib/Accounts.framework/
Social.framework
4. Connect the project to Parse
in AppDelegate.m, add this line to didFinishLaunchingWithOptions:
[Parse enableLocalDatastore];
[Parse setApplicationId:@“xxxxx” clientKey:@“xxxxx”];

Table 4-1: The procedure of deployment of Parse

Meanwhile, iFly Speech Engine is a powerful tool that provides many functions, including Speech Recognizer, Speech Synthesizer, Data Uploader, Data Downloader, Voice Wakeuper, ISV Recognizer, Face Recognizer and Speech Evaluation. The whole structure is shown in Figure 4-1.

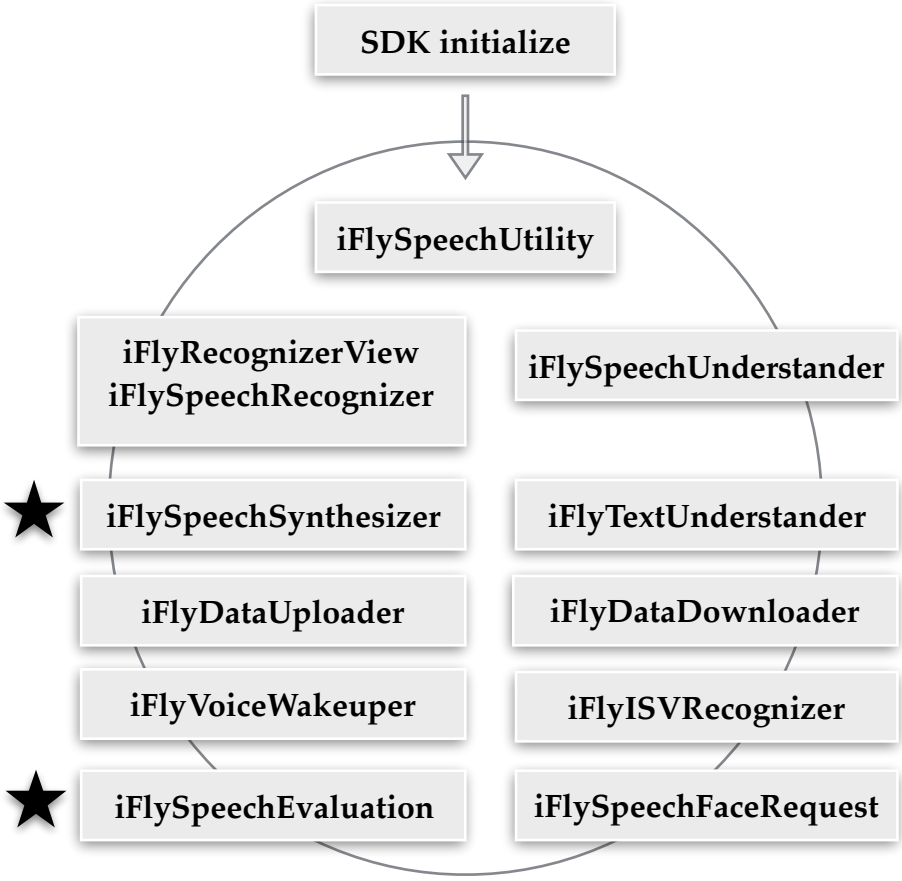

Figure 4-1: Structure of iFly Speech Engine

Thus, we only use the iFlySpeech Synthesizer and iFlySpeechEvaluation within this engine, and the procedure of deploying iFly Speech Engine and importing of this two functions shows in Table 4-2 in below, which is similar to the Parse's deployment.

1. Download & unzip the SDK

2. Add static lib (iFlyMSC.framework)

3. Add the dependencies

In 'Link Binary With Libraries', add following frameworks:

libz.dylib/CoreLocation.framework/CoreTelephony.framework/AVFoundation.framework/

AddressBook.framework/AudioToolbox.framework/SystemConfiguration.framework/

QuartzCore.framework/UIKit.framework/Foundation.framework/CoreGraphics.framework

4. import necessary headers

when I need to use Speech Synthesis service:

```
#import “iflyMSC/IFlySpeechSynthesizerDelegate.h”
#import “iflyMSC/IFlySpeechSynthesizer.h”
```

when I need to use Speech Evaluation service:

```
#import "IFlyMSC/IFlyMSC.h"
```

5. Connect the project to Parse

in AppDelegate.m, add this line to didFinishLaunchingWithOptions:

```
NSString *initString = [[NSString alloc] initWithFormat:@"appid=%@",@” 12345678”];
[IFlySpeechUtility createUtility:initString];
```

Table 4-2: The procedure of deployment of iFly Speech Engine

4.4. Implementation of Application

4.4.1.The Structure of Application

Folder Name	Brief introduction of functions
Global	It contains several classes that are globally used for application settings and parameter settings.
Logins	The View Controllers are the control classes that can interact with views and models when user logs in or signs up.
Levels	The View Controller controls all the views and data communications when choosing one level.
Courses	The View Controller controls all the views and data communications when choosing one course. If necessary, it also needs to call download function in data controller.
Units	The View Controller controls all the views and data communications when choosing one unit.
Classroom	The View Controller organizes related segmented views in one screen and responds to user's operations.
Challenge	The View Controller controls the user's interaction with the complex views's transformation, and call Speech Synthesis and Speech Evaluation function in correct order.
Result	The View Controller for the result view to show productive information by calling my own Parser to parse the XML results first.
Utility	Those are the my own views that inherent the base view of UIView to satisfy different needs to show users the interface in more intuitive and attractive way.
ISERusult	These set of classes are mainly used for parse the XML file in regular way by calling the build-in NSXMLParser, and then encapsulate it into structured classes.
Data	These classes are built for the control of data, including download the data from server, update the score, delete unnecessary data, and also the query function, etc. Moreover, related information are encapsulated into related classes here.

Table 4-3: The whole structure of the program

4.4.2.Functionality Implementation

The main functions that my application is able to provide are those: 1) User Management, 2) Lecture Downloading, 3) Lecture learning in the classroom part, 4) Lecture quiz by using the challenge part, 5) Quiz result display. Hence, the specific functionality implementations about previous five aspects will be described and illustrated in below.

4.4.2.1.User Management

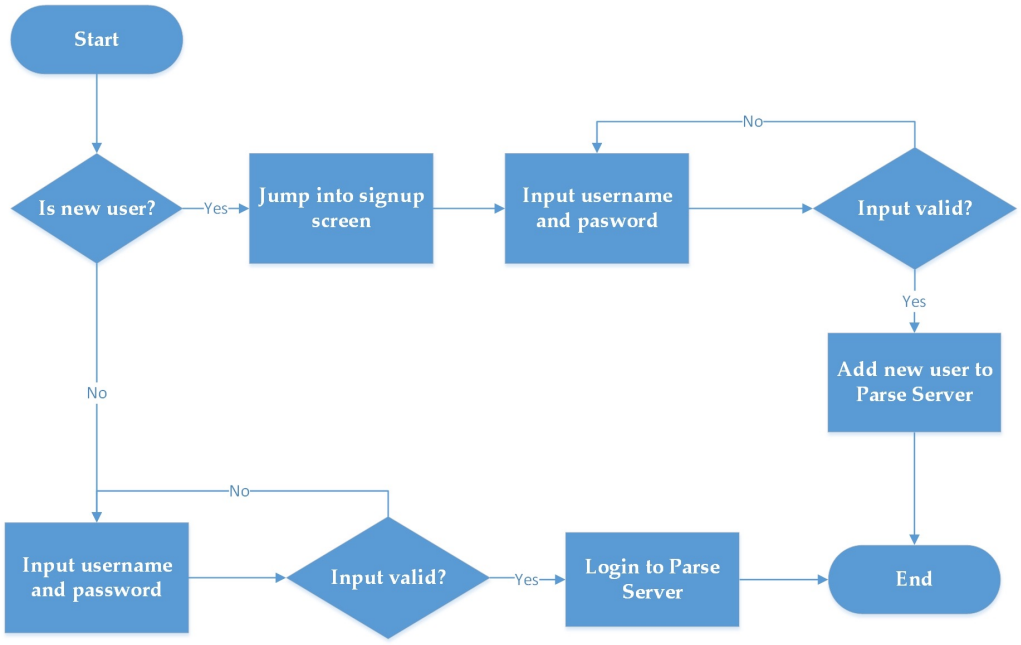

Figure 4-2: Flow chart of user management

Figure 4-2 shows the user management's structure in this Chinese speech learning system, and there are two features: one is login, another one is signup. To complete this function, I need to connect to the Parse server to check the user's information, obtain the user's information and also create a new user to save in the cloud database.

Hence, the connection of the Parse server for user management is related to "PFUser" class. The pseudo code is like this in Figure 4-3:

```
[PFUser
logInWithUsernameInBackground:
 if (success) {
        //jump to main screen of level

} else {
        //show error information
}
];
```

```
[newuser
signUpInBackgroundWithBlock:
 if (success) {
        //save new user
        //jump to main screen of level

} else {
        //show error information
                                }];
```

Figure 4-3: The pseudo code of user management

As the Figure 4-4 shows the main interface of Login and Signup, user can easily login by inputing valid username and password, or sign up a new user

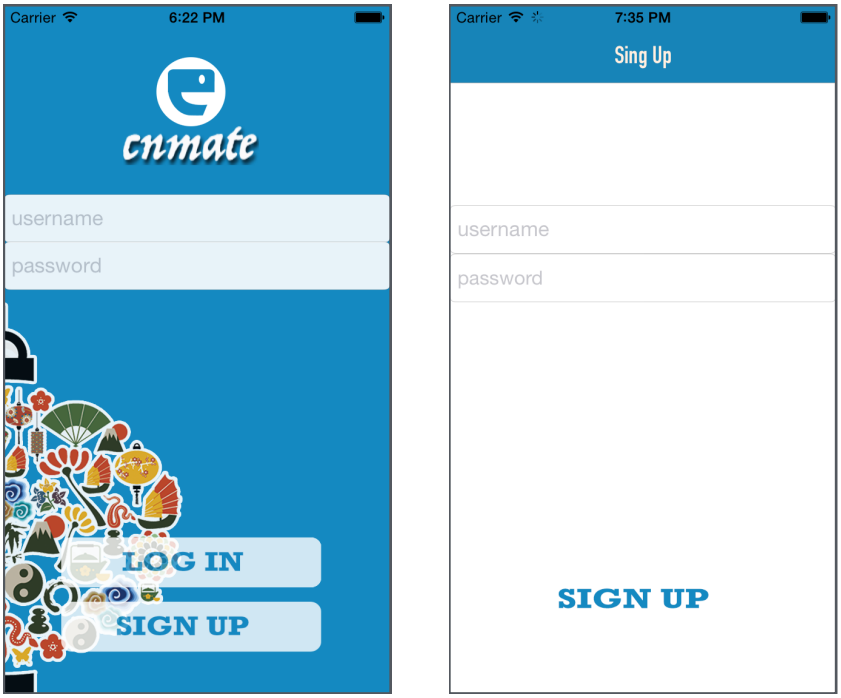

Figure 4-4: The interfaces of user management

with valid input and then directly enter the main screen of our Chinese speech learning system in current new user.

4.4.2.2.Lecture Downloading

Our lectures are uploaded by a simple uploader program (not included in CnMate project), and we can also modify those tables in databases using the dashboards on Parse conveniently. Since each lecture contains not only the plaintext resources but also some pictures for multiple choice part, the efficiency and performance of our program may be limited by the network connection. It is also unnecessary to query the same lecture for more than once, and it wastes user's cellular data. So, the downloading function is indispensable.

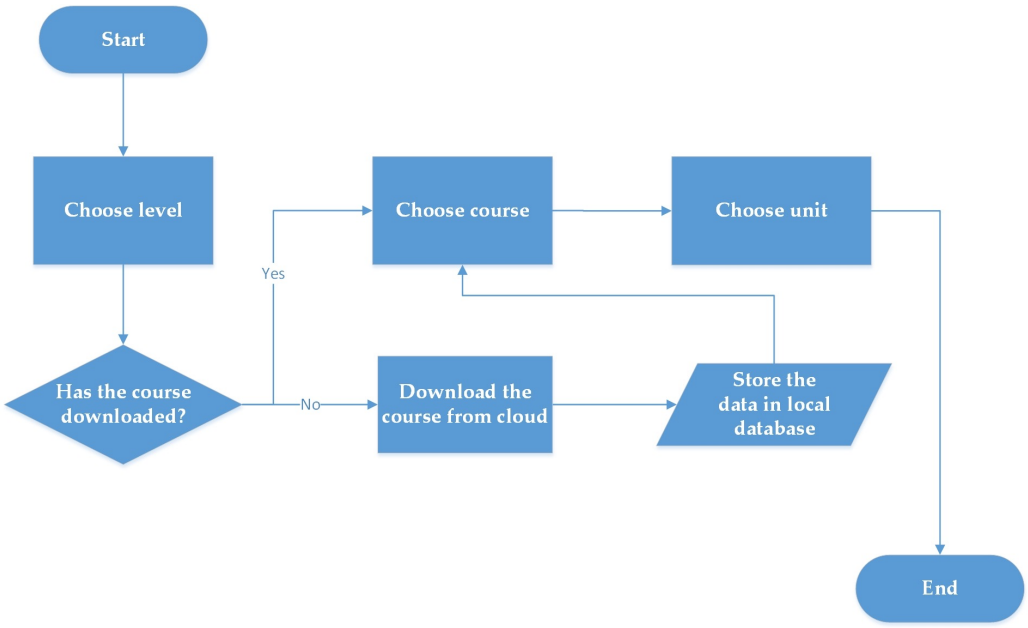

Figure 4-5: Flow chart of lecture downloading

As Figure 4-5 shows the system procedure of the lecture downloading, it only happens when the user first time unlock one course, because the system needs

to download all related units' information and lectures' information within this course and store them into the local database for quickly access.

To satisfy this function, the online queries of all units' information, all lectures' information and all pictures information should be done, and all the query results then stored to local database that is provided by Parse Service. This process may take a lot of time with the limitation of the network connection (different locations may perform differently, like Mainland China and HongKong) and system performance. Therefore, I design a rotation progress view to present the loading hint to the user in case that user thinks the system is blocked or crashed. In Figure 4-6, the pseudo code shows how I complete this job by using "pinAllInBackground" to save objects into local database, and perform a call back function to tell the view controller to let the progress view stop and disappear after finishing downloading.

```
for (unitName in database)
{
[query queryWithClassName: unitName];
objects = [query findObjects];
[PFObject pinAllInBackground: objects];
}
[self.delegate resultCallBackFunction];
```

Figure 4-6: The pseudo code of lecture downloading

If the user unlock the course “Hospital” at the first time or the user logins to the system in another devices (There is no local data in new device), as Figure 4-7 shows, there will be a “Download ALL” button on the bottom, and a rotation progress view is used here to show the progress of loading to let user know what is doing and wait.

Figure 4-7: The interfaces of lecture downloading

4.4.2.3.Lecture learning in Classroom

We focus on designing a well-structured and reasonable lectures for users to learn to speak Chinese correctly, so we build up the “Classroom” part for learning the speech mainly by listening. Figure 4-8 implys the procedure when the user enters in the classroom. For the lecture, we design one conversation for one lecture, including four sentences within the conversation. And the classroom could be divided into five segments, which means four of them corresponds to the sentences and the last segment is a simple multiple

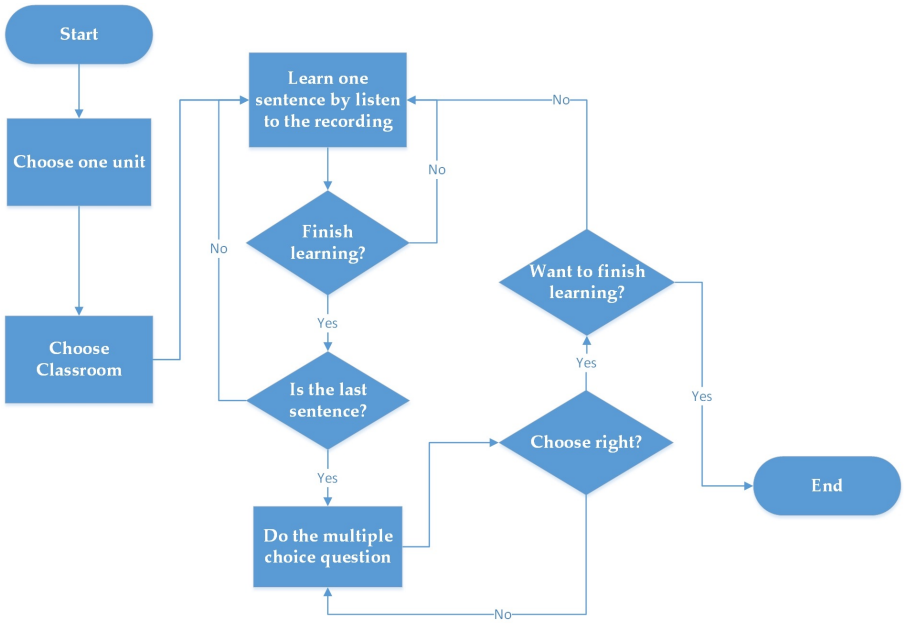

Figure 4-8: Flow chart of lecture learning

choice question after user learns the previous lecture, then listen to the recording, answer the question by selecting the only one right pictures. Only when the user reaches the last segment, the user could see the finish button to go back to up layer. In the procedure of lecture learning, the user could listen to each sentence's recording in each segment, but not necessary. The reason is that the lecture is mainly designed for those who is learning to speak Chinese in the first step, but not experienced users. Some users may want to just have a look of the whole conversation and they could pronounce correctly without practicing. It is possible in reality.

Moreover, the last question about the multiple choice is a tiny test that we design for users to get deeper understanding by combining the pictures with the scene, sentences, recordings, etc.

In order to fulfill our design about lecture learning, we definitely need to use iFly Speech Synthesizer to generate the recording automatically. Since iFly only provides the on-line version of speech synthesizer free, we will use it and

suppose the network connection will be always good. To start the speech synthesizer, I need to set parameters before starting speaking as Figure 4-9 describes. When the speaking is over, the call back function “onCompleted:” will be executed automatically, and it is also the required method for iFly Speech Synthesizer Delegate. There are several call back functions like “onSpeakProgress:”, “onSpeakPaused:”, “onSpeakResumed:”, “onSpeakCanceled:”…If there is an error during the speaking, the error code can be called in “onCompleted:” function. Hence, it is easy to call the speaking method in speech synthesizer, and the input of it is a normal NSString.

```
[_iss setParameter:@"50" forKey:[IFlySpeechConstant SPEED]];
[_iss setParameter:@"50" forKey: [IFlySpeechConstant VOLUME]];
if (girl) {
[_iss setParameter:@" xiaoyan " forKey: [IFlySpeechConstant VOICE_NAME]];}
else {
[_iss setParameter:@" xiaoyu " forKey: [IFlySpeechConstant VOICE_NAME]];}
[_iss startSpeaking:lectureSentenceInString];
```

Figure 4-9: The pseudo code of lecture learning

Once the user enters in the classroom, there will be a progress bar on the top to show the progress of learning. As the Figure 4-10 shows, the left interface is the one teaching user the sentence’s Chinese, Chinese Pinyin and English translation together with Chinese pronunciation of standard speaker by pressing the red “play” button in the middle. The user can also swipe to left to

go back and swipe to right to go to next sentence, and touch top-left "close" button to close current lecture learning process.

The last segment is the special one as a tiny test for the learning. Once user sees this segment, the recording will be played automatically (press the blue speaker button can replay), and then the user need to answer the question and choose the right pictures to get the better understander. Moreover, the script of the recording will be shown below those four pictures if the user clicks the right answer. This part is just a practice test, and not related to the final score.

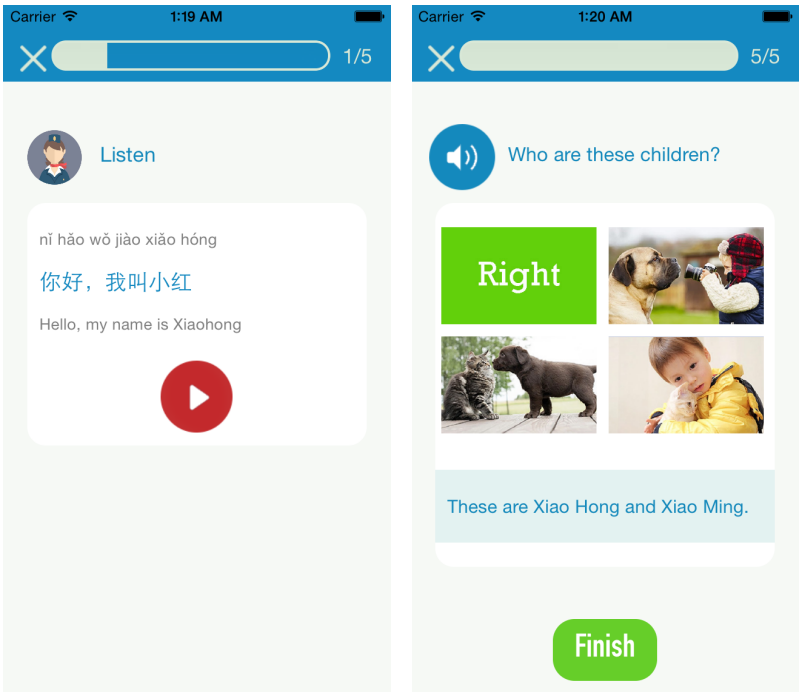

Figure 4-10: The interfaces of lecture learning

4.4.2.4.Lecture quiz in Challenge

According to our system design, we also need to implement one part called "Challenge" to test the learning effect by evaluating the user's pronunciation via the professional iFly Speech Evaluator. As Figure 4-11 shows, the whole procedure of the lecture quiz it to accomplish the conversation in two roles

separately, which means the conversation need to play twice. For example, in the first time of the conversation you play Role B, the system plays Role A, and then in the second time, you need to play Role A and the system will play Role B. In case that user may feel confused to start conversation, we also design the a count down part before starting, a switch role part for interval of two conversations. The lecture quiz is mainly about the pronunciation of the user, and let the user to apply what he learns in the classroom, and read the sentence as he remembered.

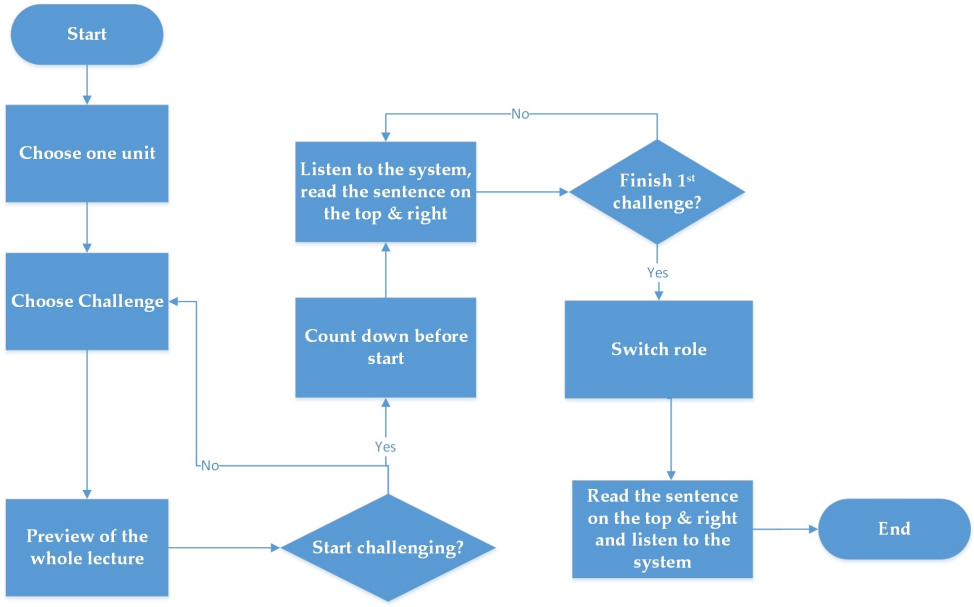

Figure 4-11: Flow chart of lecture quiz

To implement the design in this part, we surely should use iFly Speech Synthesizer together with iFly Speech Evaluator to accomplish those process we design as the flow chart shows. The iFly Speech Evaluator only supports the on-line evaluation, so the good network connection is very important and indispensable. In the section, Speech Synthesizer is used to generate recording to pretend someone behind the system who is talking to you, and Speech

Evaluator is used to evaluate how the user speaks and give out the

```
- startEvaluation() {
[_ise setParameter:  forKey:   ];
[_ise startListening:];
}
- onResults: {
        //store the evaluation result
        startSynthesis();
}
- onError: {
        //handle errors }
```

```
- startSynthesis() {
[_iss setParameter:  forKey:   ];
[_iss startSpeaking:];
}
- onComplete: {
        //handle errors if it has
        startEvaluation();
}
```

Figure 4-12: The pseudo code of lecture quiz

convincible result. Since the lecture learning part has used the Speech Synthesizer successfully, I'd like to focus on the usage of Speech Evaluator here.

Similarly, we should set some required parameters before starting the evaluator to listen to user's speaking. Since the process of this conversation depends on the network's performance to finish the speech task, there are some difficulties in the consistence of UI and the backend's service. To avoid the inconsistence of certain cases, we design that next sentence goes up only when the previous one is finished no matter what kind of sentence it is, to listen or to speak. This might be a good way to ensure that no disorder or

error will happen, but it might be to some extent limited by the network connection. If the network is not very great, the user may need to wait the response from iFly server for a few seconds under certain circumstance. The pseudo code in Figure 4-12 describing the logical relations.

The procedure of the lecture quiz part may look a little complicated but reasonable, and the views and animations appear strictly follow the system design's flow. As Figure 4-13 is showing, when the user clicks "Challenge" button to enter this section, there will be a preview of the whole lecture in the first place, and the user can choose start to start challenging or touch the "close" button on top left to go back. After starting challenging, a count-down from 3 to 1 is prepared for the user to be ready to fall into the conversation and have an overview of the whole interface.

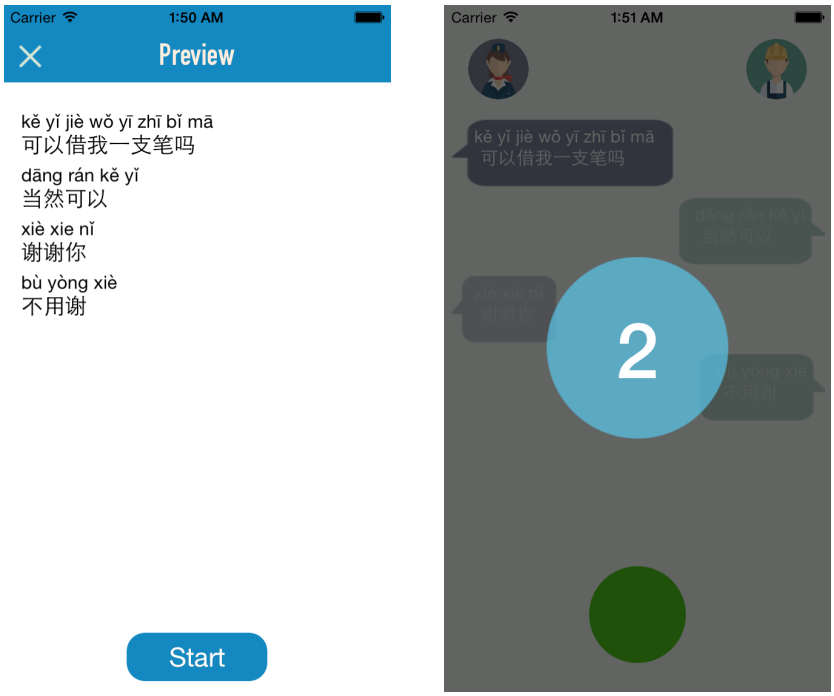

Figure 4-13: The interfaces of lecture quiz before start

In a very short time, the system will start the process in first time conversation. The user always plays the role on the right hand side as Figure 4-14 is presenting, therefore the system will act as another person to talk to you on the left hand side. Every time the user need to talk, the microphone button on the bottom will display the change of the volume, in order to help the user to adjust in appropriate way to speak. The interface in the middle of Figure 4-14 is used for a small interval between two conversations, which is similar as the count-down part, in order to give users a small piece of time to be ready.

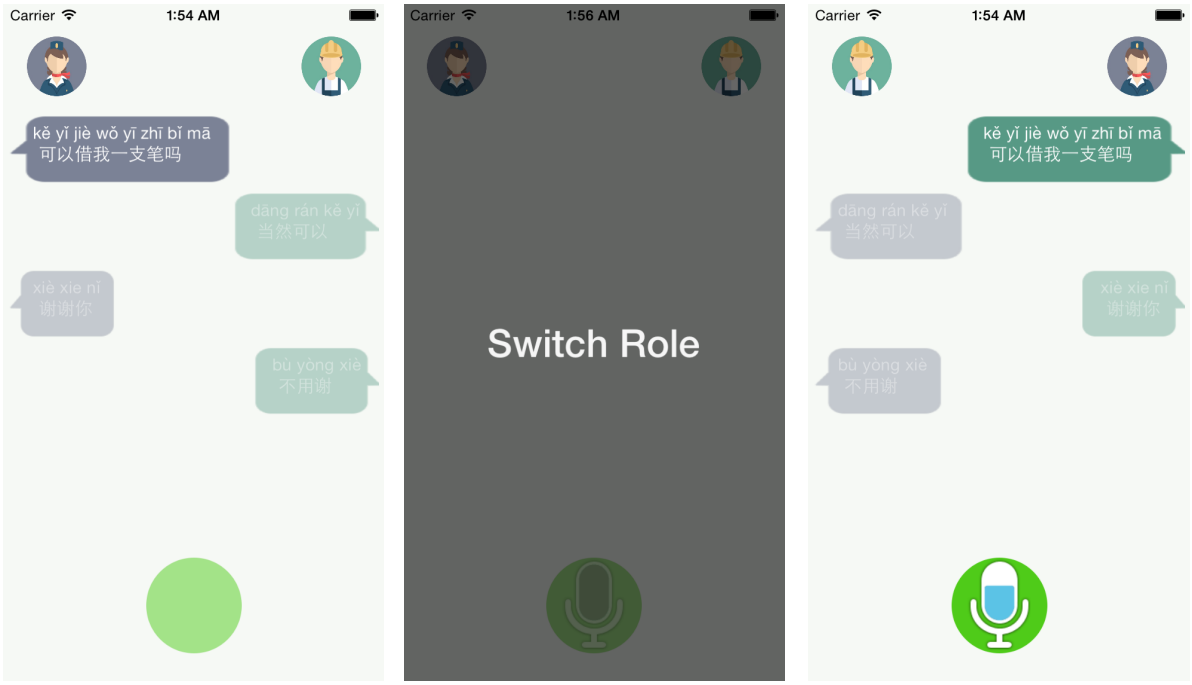

Figure 4-14: The interfaces of on-going lecture quiz

4.4.2.5.Quiz result display

If the user finishes the quiz in challenge part successfully, the user must need to gain proper feedback from the Speech Evaluator, and the results are only provided in XML format which means I need to implement a simple XML parser to parse the results into accessible classes in the program and then

display them in well-organized order. As the Figure 4-15 illustrate the system flow of result displaying, there are two more cases that need to take care: 1) when the user passes, the score information displayed in unit selection interface should be updated, and the score record also is needed to store in database (for the purpose of highest score displaying); 2) when the user passes and it is the last unit, the system should update the current user's unlock information in the server to guarantee the user could access the permitted units.

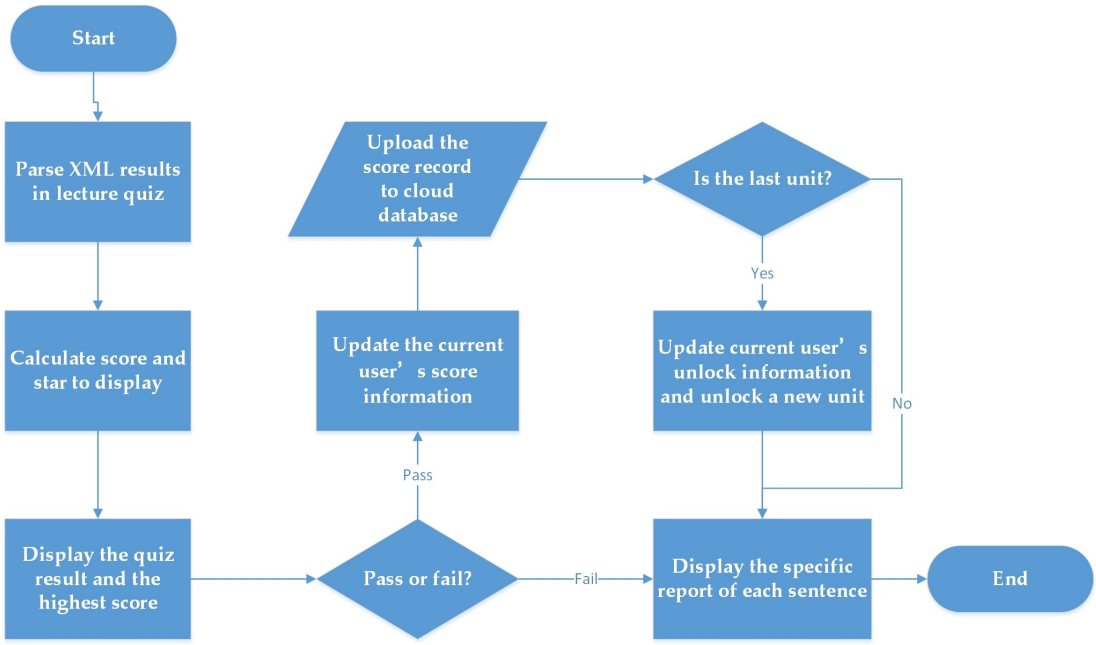

Figure 4-15: Flow chart of quiz result display

This section mainly concerns about the own particular XML Parser implementation and the update of the interface about the score and unlock information. For the XML Parser, I refer to the demo of the MSC's XML Parser and override the NSXMLParser delegate to parse the result and encapsulate it into ISEResult class for the purpose of easy access from the program. In deed,

I still need to add a delegate to set the call back function when all the parse jobs are down successfully.

```
- parser:didStart() {
        //handles the start elements
        if (@"phone"==element) {
        createPhone();
        } else if…
}
- parser:didEnd: {
        //handles the end elements
        if (@"phone"==element) {
        addObject(phone);
        } else if…
        //perform the success call
back
}
- parser: error{
        //handle errors
        //perform error call back
}
```

```
parseResults();

calculateScroeAndStart();

checkPassOrNot();

if (pass) {

        //upload the score the server

        if (lastunit) {

                unlockNewUnit();

                }
}
updateUI();
```

Figure 4-16: The pseudo code of quiz result display

In deed, I only override 3 delegate functions: “parser: didStartElement: namespaceURI: qualifiedName: attributes:”, “parser: didEndElement: namespaceURI: qualifiedName:” and “parser: parseErrorOccurred:”, and in last two functions to perform my own delegate to announce the completion of the parse process and store the parsed results. The left part in Figure 4-16 describes the process of XML parser’s job in a rough view.

In addition, the right part in Figure 4-16 simply indicates the main function in the result displaying section. Since iFlySpeechEvaluator does not provide the audio recording path in SDK, the part of playing own recording is unable in iOS platform, which isa pity. If it does, I will connect the recording to the corresponding sentences in the main function here too.

In consequence, the user will see the result like the Figure 4-17 is showing , either right one or left one. Similarly, the displaying items will show the total score, the highest score from other user’s, the stars and also the specific report

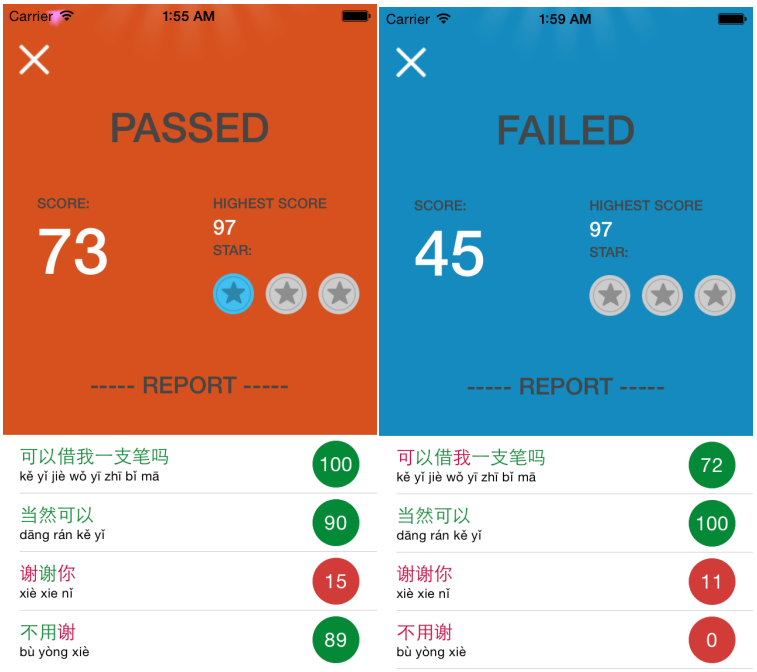

Figure 4-17: The interfaces of quiz result display

with specific scores for sentences. As the example of the right one in Figure 4-17, the first sentence in the conversation is “可以借我一支笔吗”, in which some syllables are marked green, and some are marked red. It is the displaying of the parsed result of each sentence. The structure of one sentence is made up of several words, and one word is made up of several syllables, and one syllable is made up of several phones. There is a label for each phone’s performance, in which 1 stands for wrong speaking and 0 stands for normal speaking. Our rules is that once there is one 1 within one syllables, and the syllable will be marked red, otherwise it will be marked green.

Apart from the result displaying views, the unit selection view also will be updated if the user passes the quiz. Once the user finishes the quiz and knows the quiz results, the user will touch the top left “close” button to go back to the unit selection screen as Figure 4-18 illustrates. We could easily find the stars’

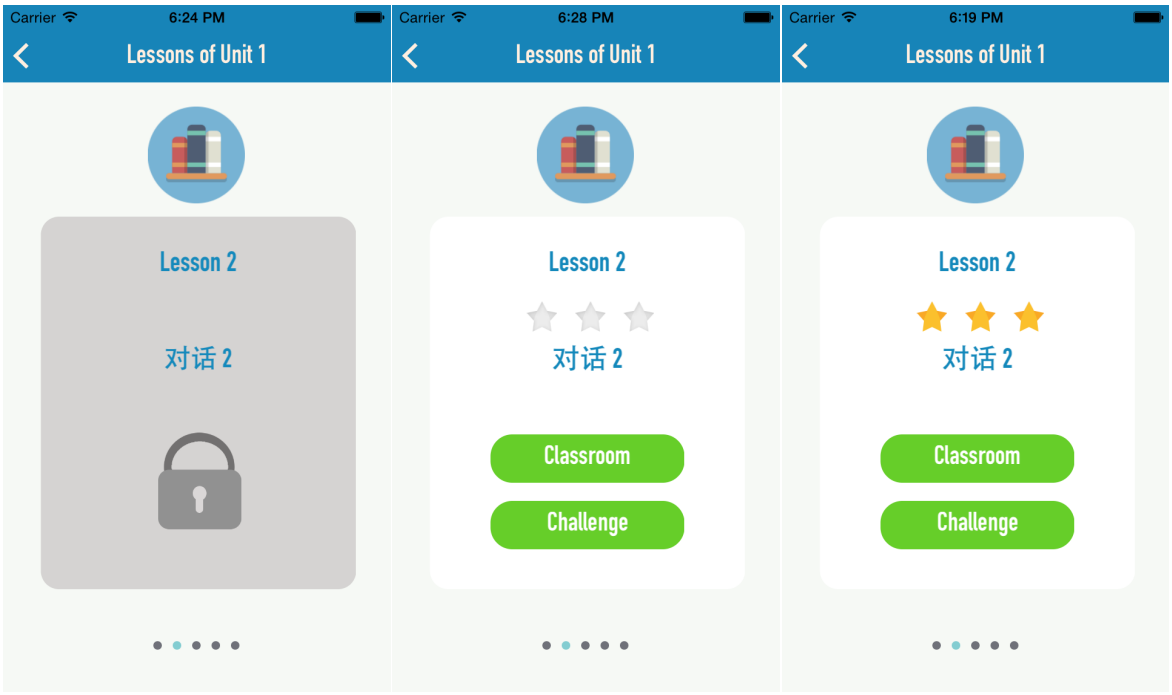

Figure 4-18: The interfaces of unit selection

view will be updated and the stars are consistent with the score: 0 star for score less than 60; 1 star for score more than 60 but less than 75; 2 star for score more than 75 but less than 85; 3star for score more than 85.

4.4.3.The Database of the Application

The database of our application is mainly used to store the information of the lecture information and the specific resources in each lecture, together with the user and score management. In order to support two platforms(Android & iOS) at the same time, we determine to build the cloud database on the server so that we could share the same resources. Hence, Parse is the best choice for us, because it is the particular server especially designed for mobile applications, and we are going to use the core service of data there.

By following the general process of designing database, we analyze the user requirements in the Chapter 2, and then build up the concept model, logical model and implement the design in Parse's cloud database. Since Parse uses the non-relational database similar to MangoDB, the performance of the query efficiency is great.

In the procedure of implementation, we firstly design the lectures in the local machine, and do the data preprocessing before organizing the data into the cloud database. The task to pick related pictures and preprocess the images manually cost a lot of time.

In general, we designed 6 tables in total for storage of different information: 1)CourseInfo, 2)UnitInfo, 3)UnitDetail,4)UnitSelectIofo, 5)UnitSelectImg, 6)GameScore. In addition, a special table called User which is auto generated

by Parse for user management. Since we decide to develop level 1 temporarily, we do not put level's information in this database.

The detail of each table will be shown in below figures:

I. CourseInfo(Table 4-4): This table is used for the storage of the courses' information, including couseEnName, courseZhName, courseIcon and level. There are 6 courses here.

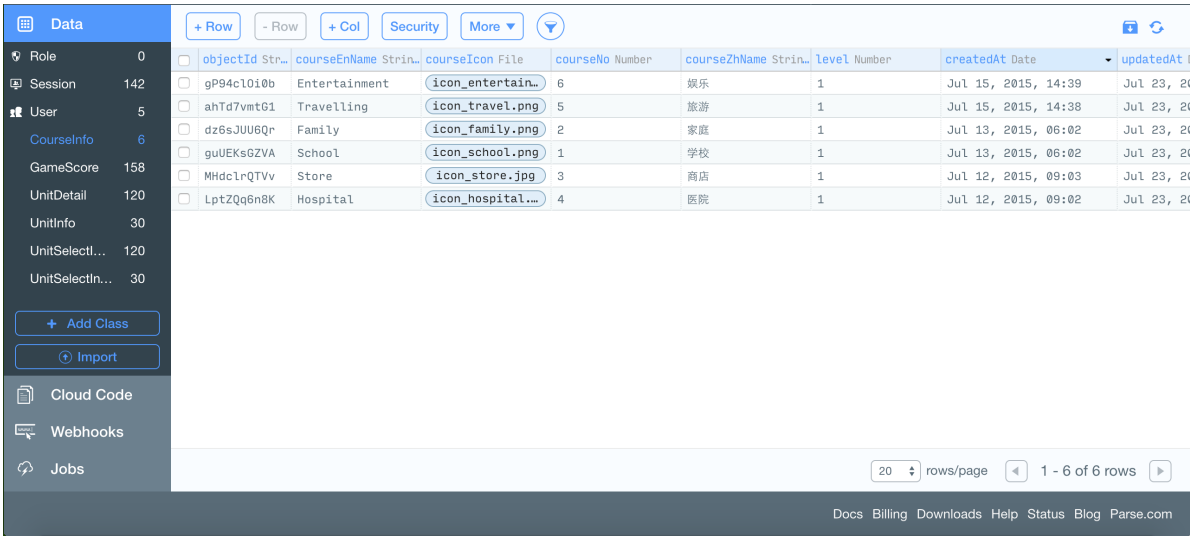

Table 4-4: The table of CourseInfo

II. UnitInfo(Table 4-5): This table is used for the storage of the units' normal information, including level, courseNo, unitNo, segmentSize,

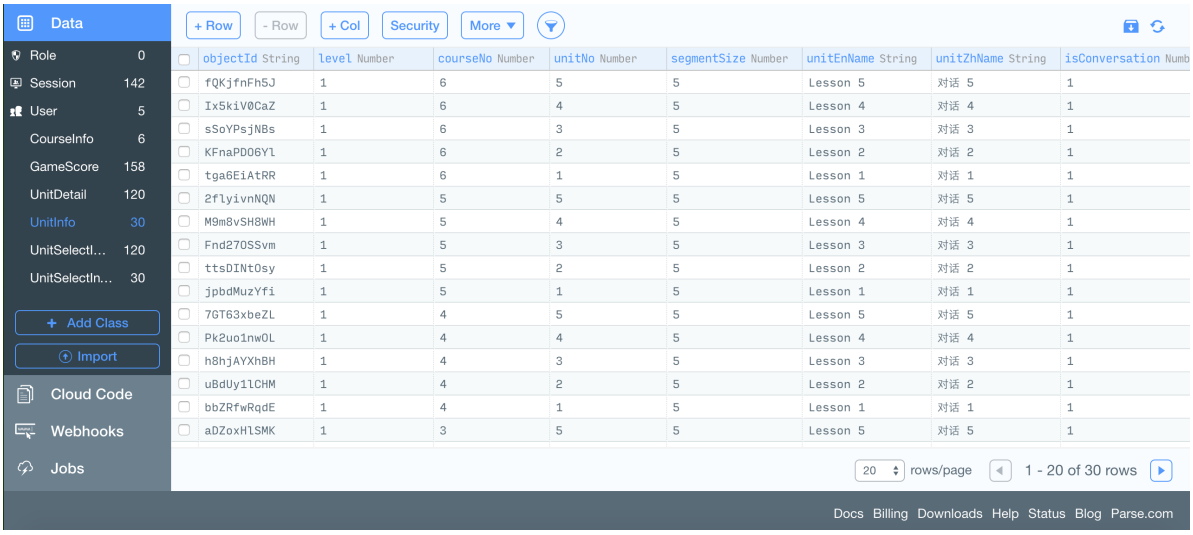

Table 4-5: The table of UnitInfo

unitEnName, unitZhName and isConversation (may be used in the future). There are 30 units in total, and 5 units in each course.

III. UnitDetail(Table 4-6): It stores the detail in each unit which has the same meaning of “lecture” that I used in the previous part. There are columns identifying level, courseNo, unitNo, segmentNo, content(Chinese), contentPinyin and contentEn(English Translation). This table consists of 120 records because we have 30 units and 4 records for 1 unit.

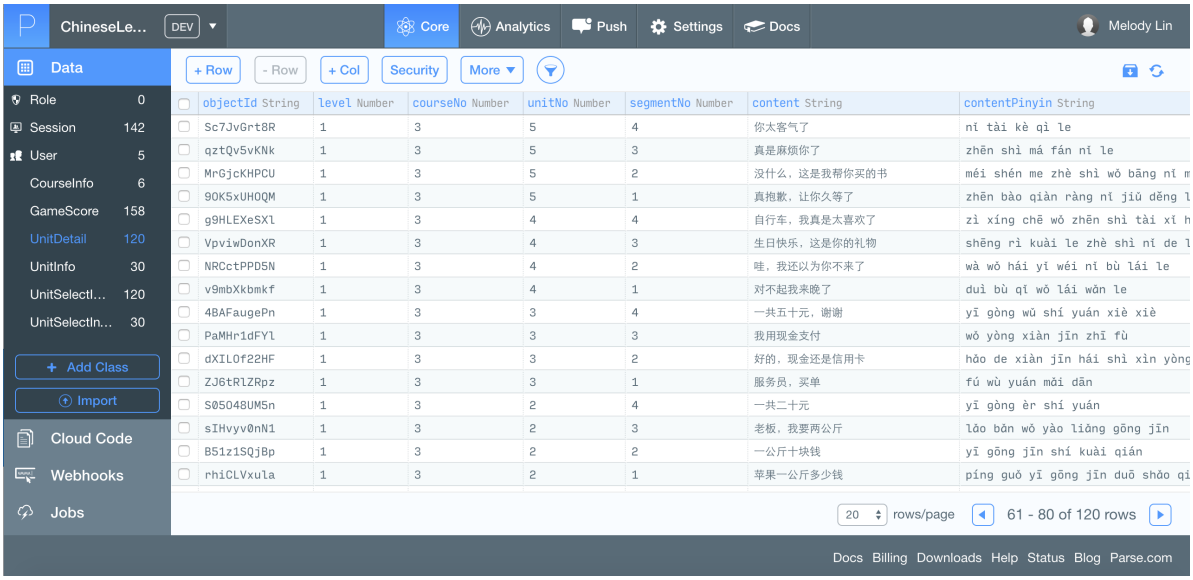

Table 4-6: The table of UnitDetail

IV. UnitSelectInfo(Table 4-7): It is mainly designed for the multiple choice within the classroom part, so there are 30 records for 30 units in total. Each record contains level, courseNo, unitNo, segmentNo, selectTitle, and testContents.

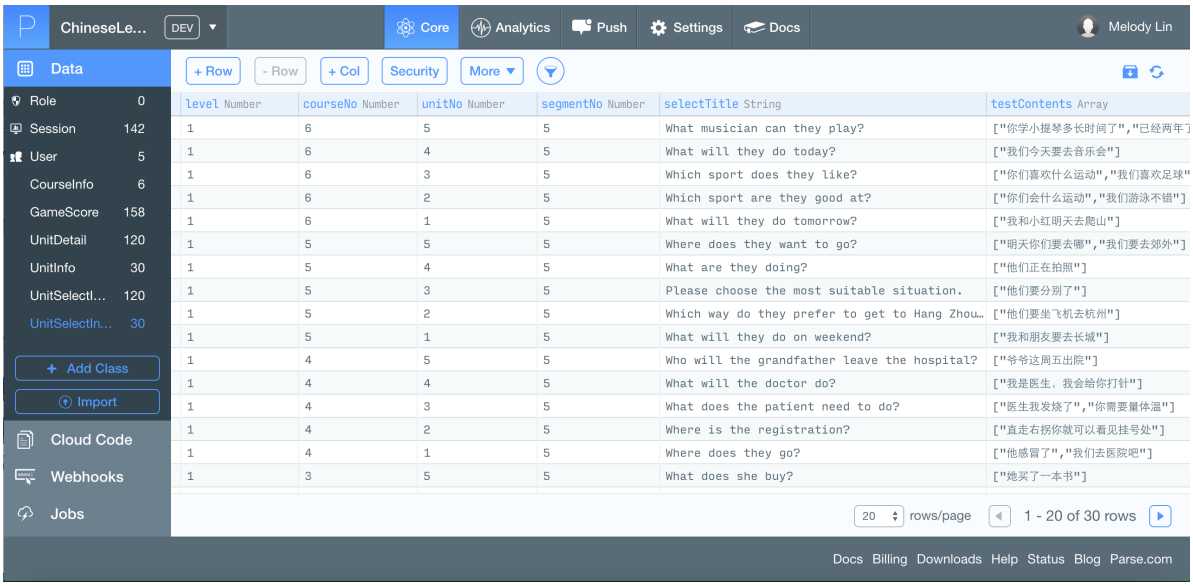

Table 4-7: The table of UnitSelectInfo

V. UnitSelectImg(Table 4-8): This table is also designed for the multiple choice question, but four records for one unit. Therefore, there are still 120 records in this table, and each record contains the column: level, courseNo, unitNo, segmentNo, tag(1 means corresponding record is the correct one) and selectImg.

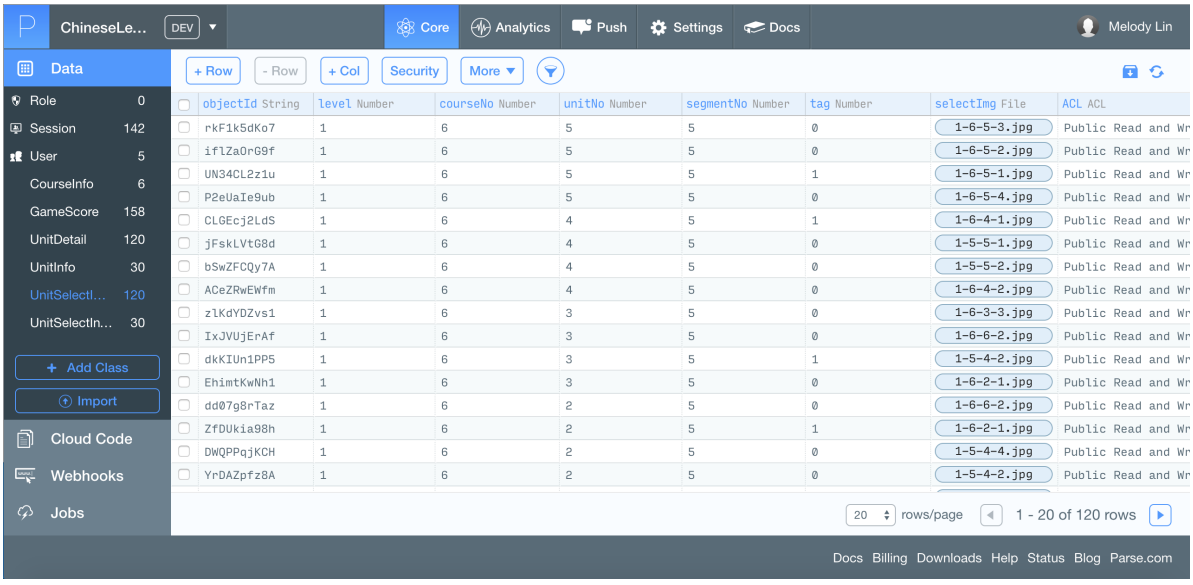

Table 4-8: The table of UnitSelectImg

VI. GameScore(Table 4-9): Since our applications need to store the user's performance in each unit, the score table is necessary. Each score record corresponds to one user's successful challenge result. There are level, courseNo, unitNo, mark together with username.

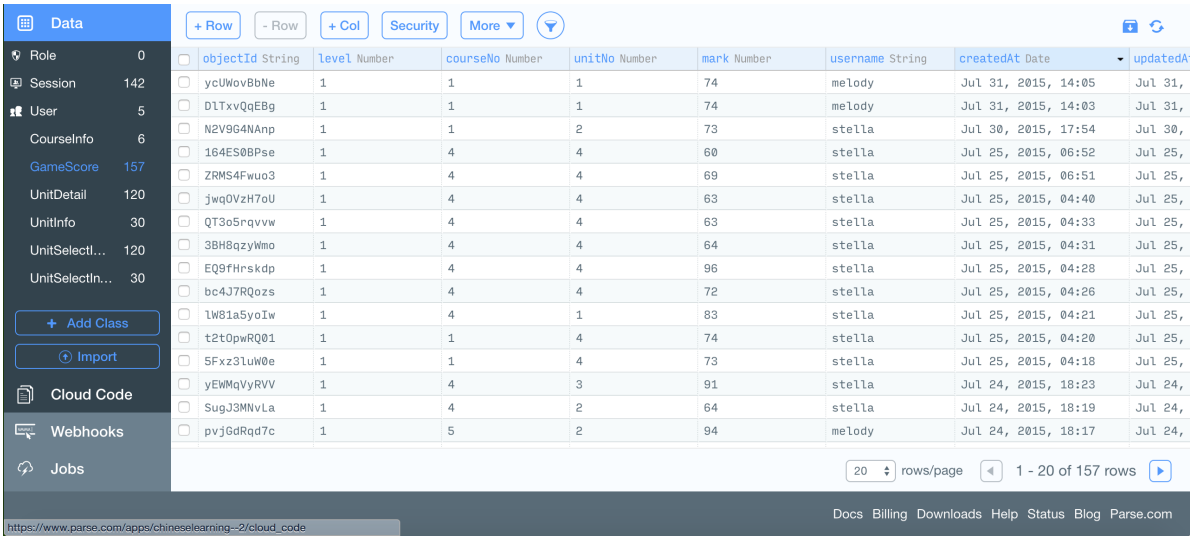

Table 4-9: The table of GameScore

5. Conclusion

The final chapter in this report primarily discusses the outcomes in this project and the some obvious shortcomings to fix or adjust in the future, then look forward to the future if we have time to keep this project on and publish it in one day.

The major outcome of this project is to particularly design a Chinese speech learning application (called CnMate) which may be considerably attractive and meaningful for foreigners who wants to learn spoken Chinese. And more details will be discussed in the following section.

On the other hand, I also objectively point out some aspects that I need to improve, such as the consistency of UI design in two platforms. Looking forward to the future, I will take those shortcomings into consideration and find the possible ways to figure it out.

5.1. Project outcomes

On the basis of the problem analysis, user requirement analysis, system design and the solution implementation, I completed an iOS application called CnMate for Chinese speech learning especially for foreigners with little basis of Chinese.

To conclude my effort in this project, I would like to classify all the outcomes into two major aspects — one is the practicality of an mobile Chinese learning application, and another one is the attraction in user interaction design.

A. Practicality of the application

As the background research describes, there are few applications applied for Chinese learning which are in the lack of practicality in real life because in people's

daily life, the oral Chinese must be used most frequently. Hence, we feel that we do good job in the system design to make it more practical and close to real life, and to some extent, it is more than part of existing applications in app store.

In this application, foreigners can learn to speak Chinese by circular process of the listening, speaking, practicing and evaluating from time to time to build up those knowledge into their long-term memory. We strictly and possibly combine all related information to help users memorize in different method efficiently.

B. Attractive user interaction

In the early stage of the design and development of this project, we always take user interaction into account, and always find the proper way to communicate with users like friends. Thanks to our attention on the interaction design, it brings worthy feedback from some classmates who have tested our application. The main reason that educational application is not prevalent may be the dull and boring process when they interact with the system. If the user interface design or the interaction method is not innovative and attractive, the user will feel frustrated very soon and not open this application again. Successfully, we receive some positive commands to appreciate the user interaction design.

5.2. Future work

Unfortunately, the project still perform not so good in some aspects as follows:

A. The data preprocessing work in our project takes a lot of time, including manually input the unit detail and cut the images of selection part into suitable size.

B. The dependency of the network connection. Since iFly only provides the on-line version of speech synthesizer and speech evaluator free, the system's performance may be affected under some circumstances. The robustness of the whole system still need to improve, because the different situations in the network connection may lead to some ridiculous problems in the application.
C. The lack of large amount of user testing. We only complete the basic function testing manually on our own, and bring it to a number of classmates to test the functions at the same time. However, the poor testing result may not be very convincible, and a series of formal automated testings are necessary.

In the future, I extremely expect to publish my own application in app store, but there are more works that I must do before that. First, I need to write another small program to take over the data preprocessing instead of manual operations, which can raise the efficiency dramatically. Second, I should improve the robustness of this application and if possible, use the off-line iFly speech engine to ensure the system's performance. Last but not least, a large amount of automated testing is indispensable before publishing. What's more, I plan to user the build-in UIAutomation tool in Xcode and some professional automated testing tool like FoneMonkey for iOS.

Reference

[1] Shneiderman, Ben, et al. "Designing the User Interface: Strategies for Effective Human-Computer Interaction." (2009).

[2] Gong J, Tarasewich P. Guidelines for handheld mobile device interface design[C]//Proceedings of DSI 2004 Annual Meeting. 2004: 3751-3756.

[3] Ayob N, Hussin A R C, Dahlan H M. Three layers design guideline for mobile application[C]//Information Management and Engineering, 2009. ICIME'09. International Conference on. IEEE, 2009: 427-431.

[4] T. Masuko, K. Tokuda, T. Kobayashi and S. Imai, “Speech synthesis from HMMs using dynamic features,” Proc. of ICASSP, pp.389–392, 1996.

[5]K. Tokuda, T. Masuko, N. Miyazaki and T. Kobayashi, “Hidden Markov Models Based on Multi-Space Probability Distribution for Pitch Pattern Modeling,” Proc. of ICASSP, 1999.

[6]M. Russell and R. Moore, “Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition,” in Proc. of ICASSP, 1985, pp. 5-8.

[7]H. Zen, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” Proc. of ICASSP, 2013.

[8] Rabiner, L.R., “A tutorial on hidden Markov models and selected applications in speech recognition”, Proceedings of the IEEE, 77(2):257–286, 1989.

[9] https://parse.com/

[10] http://www.xfyun.cn/