

GSOC 2012 FINAL REPORT

PRONUNCIATION EVALUATION PORTAL DESIGN AND IMPLEMENTATION

Design and implement a web portal for Pronunciation Evaluation for Language Learning using edit distance scoring with CMU Sphinx3, copious speech data collection and a game-based learning interface.

GSOC 2012 FINAL REPORT

PRONUNCIATION EVALUATION PORTAL DESIGN AND IMPLEMENTATION

***Mobile Pronunciation Evaluation for Language Learning
Using Edit Distance Scoring with CMU Sphinx3, Copious
Speech Data Collection, and Game-based Interface***

Pronunciation learning is one of the most important parts for second language acquisition. The aim of this project is to utilize the automatic speech recognition technologies to facilitate spoken language learning. This project will mainly focus on developing accurate and efficient pronunciation evaluation system using CMU Sphinx3 and maximizing the adoption population by implementing a free web portal for our evaluation system. Additionally, we also plan to design and implement game based pronunciation learning to make the learning process much more fun. Four specific sub-tasks are involved in this project, namely, automatic edit distance based grammar generation, exemplar pronunciation database building, web based pronunciation evaluation system implementation and game based learning interface development.

In this report, we mainly describe the design and implementations of the web portal for exemplary data collection and pronunciation evaluation we have done during this summer. The game based learning will be an extension for the work described in this report.

BO LI

(TROY LEE)

He is a Ph.D student at the National University of Singapore Computational Linguistics Lab. His primary research interests include acoustic modeling and speaker adaptation for automatic speech recognition. He is also interested in using Sphinx3 for language learning.

Email:

troy.lee2008@gmail.com

TABLE OF CONTENTS

CHAPTER 1 SYSTEM OVERVIEW	4
FUNCTIONALITIES	4
MODULES	4
RELATIONSHIPS	5
CHAPTER 2 DATABASE DESIGN	6
USER INFORMATION	6
TABLE USERS	6
TABLE LANGUAGES	6
TABLE REGIONS	7
PHRASE INFORMATION	7
TABLE PHONEMES	7
TABLE WORDS	7
TABLE PRONUNCIATIONS	7
TABLE PHRASES	7
TABLE PHRASEINFO	8
TABLE PHRASESTATS	8
RECORDING INFORMATION	8
TABLE RECORDINGS	8
TABLE SCORES	9
TABLE TESTS	9
TABLE TESTINFO	9
GAME DESIGN	9
TABLE NODES	10
TABLE NODEINFO	10
TABLE GAMES	10
TABLE GAMEINFO	10
CHAPTER 3 USER MANAGEMENT MODULE	11
MODULE STRUCTURE	11
MODULE IMPLEMENTATIONS	11
SCREENSHOTS	12
CHAPTER 4 AUDIO RECORDING MODULE	15
MODULE STRUCTURE	15
MODULE IMPLEMENTATION	16
SCREENSHOTS	16
CHAPTER 5 EXEMPLAR VERIFICATION MODULE	18
MODULE STRUCTURE	18
MODULE IMPLEMENTATION	19

SCREENSHOTS	19
CHAPTER 6 PRONUNCIATION ASSESSMENT MODULE	20
MODULE STRUCTURE	20
MODULE IMPLEMENTATION	21
SCREENSHOTS	21
CHAPTER 7 PHRASE LIBRARY BUILDING MODULE	22
MODULE STRUCTURE	22
MODULE IMPLEMENTATION	23
SCREENSHOTS	23
CHAPTER 8 HUMAN EVALUATION MODULE	24
MODULE STRUCTURE	24
MODULE IMPLEMENTATION	25
SCREENSHOTS	25
CHAPTER 9 CONCLUSIONS	27
DISCUSSIONS	27
FUTURE WORK	27

CHAPTER 1 SYSTEM OVERVIEW

Functionalities

Our pronunciation evaluation system will mainly have two groups of users: exemplary data recording users and students who would like to practice their pronunciation. Besides, users contributing exemplar phrases and human experts assessing the students' performances also need to be included to make our system more flexible and extensible.

There are two main purpose of building such a pronunciation evaluation system: namely, 1) Exemplar data collection for improved pronunciation evaluation performance based on exemplars; and 2) Pronunciation evaluation for foreign language learning, currently our focus is English which can be easily extended to other languages. Hence, the major functionalities the system should have are user management, exemplar recording verification, pronunciation assessment, new phrase library building and human evaluation. A detailed list of functionalities for a web based pronunciation evaluation system is shown in the table below.

General Category	Descriptions
User Information Management	1. User Registration 2. User Log in/out 3. Reset password 4. Update additional information
Exemplar Recording Verification	1. Data recording 2. Verification
Pronunciation Assessment	1. Data recording 2. Pronunciation scoring
Phrase Library Building	1. Phrase analysis 2. Pronunciation specification
Human Evaluation	1. Recording retrieval 2. Audio analysis 3. Scoring

Modules

Based on the system requirements and the analyzed functionalities, following modules are thus necessary for implementing such a pronunciation evaluation system:

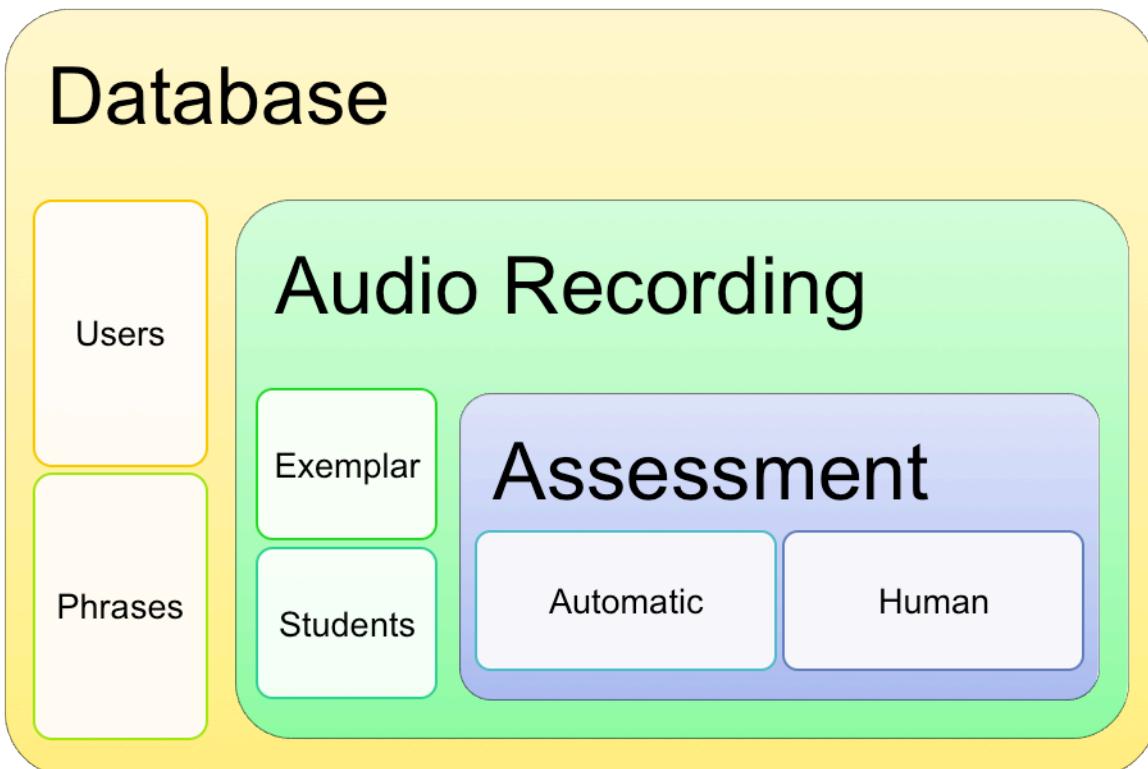
1. **Database Management Module:** Store, retrieval and update all the necessary information including both user information and various data information such as phrases, words, correct pronunciations, assessment scores and etc.
2. **User Management Module:** Create new user, reset password, and update user information for existing user.
3. **Audio Recording Module:** Recording the users' pronunciation for further processing.
4. **Exemplar Verification Module:** Justify whether the given recording is an exemplar or not.



5. **Pronunciation Assessment Module:** Provide users with numerical assessment for each linguistic unit in their pronunciation.
6. **Phrase Library Module:** Allow users to create new phrases for evaluation and specify the correct pronunciation for the new phrases.
7. **Human Evaluation Module:** Support human experts to evaluate the users' pronunciation which could be compared with the evaluation automatically generated.

Relationships

The general relationships between different modules are illustrated in the Figure below.



In the following chapters, each of the modules will be described in more details especially the implementation details such that the web based pronunciation evaluations system could be easily ported by others to meet their specific needs.

CHAPTER 2 DATABASE DESIGN

The database in our system implementation is using the Open Source MySQL database. The script for creating all the tables we used in the repository is **databases/s00_create.sql** and some initial data are loaded through the script **databases/s01_insertdata.sql**.

User Information

TABLE USERS

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, primary key
email	VARCHAR(200)	Compulsory, users are identified by emails
password	VARCHAR(50)	Compulsory, encrypted using SHA1, at least 8 alphanumeric characters
type	CHAR(8)	Compulsory, user type, default "10000000", currently only 5 types: (S)tudent:"10000000", (E)xemplaires:"01000000", (J)udges:"00100000", (T)eachers:"00010000", and (A)dmins:"00001000"; for now only the first 5 bits are used, others for future use.
time	DATETIME	Not compulsory, the registration time.
name	VARCHAR(100)	Not compulsory, default 'NULL'
birthdate	DATE	Not compulsory, default 'NULL', format 'YYYY-MM-DD', using birthday date instead of age
sex	SET('m', 'f')	Not compulsory, default 'NULL', accepted values {'m', 'f'}
langid	INTEGER	Foreign Key. Not compulsory, default 'NULL', Indicating the user's native language. Ref the primary key of tabel #languages.
regionid	INTEGER	Foreign Key. Not compulsory, default 'NULL'. Indicating the place when the user lived at the age between 6 and 8. Ref the primary key of the country codes table #regions.

The basic part of the user information includes: [userid, email, password, type]. In the registration step, only the [email, password] are required for each user to simply the registration step. For exemplary data recording users, an additional information page will be compulsory for their registration while for other types' users it is not.

TABLE LANGUAGES

Field	Type	Comments
id	INTEGER	Primary Key, automatically increased
code	CHAR(2)	Compulsory, accepted values the [ISO-693-1][1] two letter language codes.
description	VARCHAR(100)	Not compulsory, text explanation of the language code.

TABLE REGIONS

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, Primary Key.
country	CHAR(2)	Compulsory, Using ISO-3166-1 alpha-2 codes [2] .
subdivision	VARCHAR(3)	Not compulsory, Using the second part of ISO-3166-2 codes [3] .
description	VARCHAR(100)	Not compulsory, text description of the region codes.

Phrase Information

TABLE PHONEMES

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, Primary Key.
name	VARCHAR(3)	Compulsory, text name of the phoneme

TABLE WORDS

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, Primary Key.
name	VARCHAR(50)	Compulsory, word
numpro	INTEGER	Compulsory, Default 1, number of pronunciations

TABLE PRONUNCIATIONS

Field	Type	Comments
wordid	INTEGER	Compulsory, automatically increased, Primary Key. Ref#words
proid	INTEGER	Compulsory, Default 0, pronunciation index to distinguish multi-pronunciation words. Values in {0...#words's numpro-1}
phnid	INTEGER	Compulsory, phoneme id Ref #phonemes
phnpos	INTEGER	Compulsory, phoneme position in the word, starts from 0.

TABLE PHRASES

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, Primary Key.
name	VARCHAR(200)	Compulsory, phrase text.

TABLE PHRASEINFO

Field	Type	Comments
phraseid	INTEGER	Compulsory, Ref #phrase
wordpos	INTEGER	Compulsory, word position in the phrase, starts from 0
wordid	INTEGER	Compulsory, Ref #words
proid	INTEGER	Compulsory, Ref #pronunciations
postag	SET('q', 'n', 'v', '-', 'w', 'm', 'o', 's', 'p', 'c', 'a')	POS tag for the word in that phrase, accepted values {'q', 'n', 'v', '-', 'w', 'm', 'o', 's', 'p', 'c', 'a'}, representing {"Quantifier", "Noun", "Verb", "Negative", "Adverb", "Adjective", "Pronoun", "Possessive", "Preposition", "Conjunction", "Article"} Ref: [7] [8]

TABLE PHRASESTATS

Field	Type	Comments
phraseid	INTEGER	Compulsory, Ref #phrase
phnpos	INTEGER	Compulsory, phone position in the phrase, starts from 0
phnid	INTEGER	Compulsory, Ref #phonemes
acomean	DECIMAL	Compulsory
acostd	DECIMAL	Compulsory
acomaxz	DECIMAL	Compulsory
durmean	DECIMAL	Compulsory
durstd	DECIMAL	Compulsory
durmaxz	DECIMAL	Compulsory

Recording Information

TABLE RECORDINGS

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, Primary Key.
userid	INTEGER	Compulsory, Foreign Key, Refer #users
phraseid	INTEGER	Compulsory, Foreign Key, Refer #phrases
time	DATETIME	Compulsory, recording time.
audio	VARCHAR(300)	Compulsory, the recording file path.
exemplar	SET('y', 'n')	Compulsory, accepted values {'y', 'n'}, indicating whether this recording is an exemplar or not.

TABLE SCORES

Field	Type	Comments
recordingid	INTEGER	Foreign Key, Compulsory, Ref #recordings's id.
type	SET('p', 'b', 'w', 's')	Compulsory, indicating the type of the score: 'p':phonemes, 'b':biphones, 'w':words, 's':phrases(sentences).
wordid	INTEGER	Foreign Key, Ref #words
wordpos	INTEGER	word position in the phrase, starts from 0
proid	INTEGER	Ref #pronunciations
phoneid	INTEGER	Foreign Key, Ref #phonemes
phonepos	INTEGER	phone position in the current word, starts from 0
acousticscore	DECIMAL	Compulsory,
judge	INTEGER	Compulsory, Foreign Key, Ref #users the one who give the current score.
durationscore	DECIMAL	Compulsory, Time duration score.

TABLE TESTS

Field	Type	Comments
id	INTEGER	Compulsory, automatically increased, Primary Key.
title	VARCHAR(50)	Compulsory.
type	CHAR(4)	Compulsory, expected values {'1000':phonemes, '0100':biphones, '0010':words, '0001':phrases} With multiple 1s indicates a mixed test.
length	INTEGER	Compulsory, number of prompts in this test.
ordered	SET('y', 'n')	Compulsory, expected values {'y', 'n'}, indicating whether the test should be done in the specified order.

TABLE TESTINFO

Field	Type	Comments
testid	INTEGER	Foreign Key, Compulsory, Ref #tests's id.
phraseid	INTEGER	Foreign Key, Compulsory, Ref #phrases' id.
orderid	INTEGER	Compulsory, The order index of the current prompt in the specified test.

Game Design

The game development for the pronunciation evaluation system will be the extension of the GSoC project. As our pronunciation evaluation system is exemplar based, we have to collect enough exemplars for all the possible phrases/sentences before we can make the game work. However, this component has always been there since the start of this GSoC project, we have design and implemented the database structure for the game, which are thus also included in this report.

TABLE NODES

Field	Type	Comments
id	INTEGER	Primary Key, Compulsory.
phraseid	INTEGER	Foreign Key, Compulsory, Ref #phrases id.
description	VARCHAR(200)	Text description of the current game node.
graph	VARCHAR(200)	Graphic file path.
sound	VARCHAR(200)	Sound file path.

TABLE NODEINFO

Field	Type	Comments
nodeid	INTEGER	Foreign Key, Compulsory, Ref #nodes id.
oldwid	INTEGER	Foreign Key, Compulsory, Ref #words id.
oldpos	INTEGER	Word position in the original phrase.
newwid	INTEGER	Foreign Key, Compulsory, Ref #words id.

TABLE GAMES

Field	Type	Comments
id	INTEGER	Primary Key, Compulsory.
nodeid	INTEGER	Foreign Key, Compulsory, Ref #nodes id.
description	VARCHAR(200)	Text description of the current game node.
level	INTEGER	Compulsory.

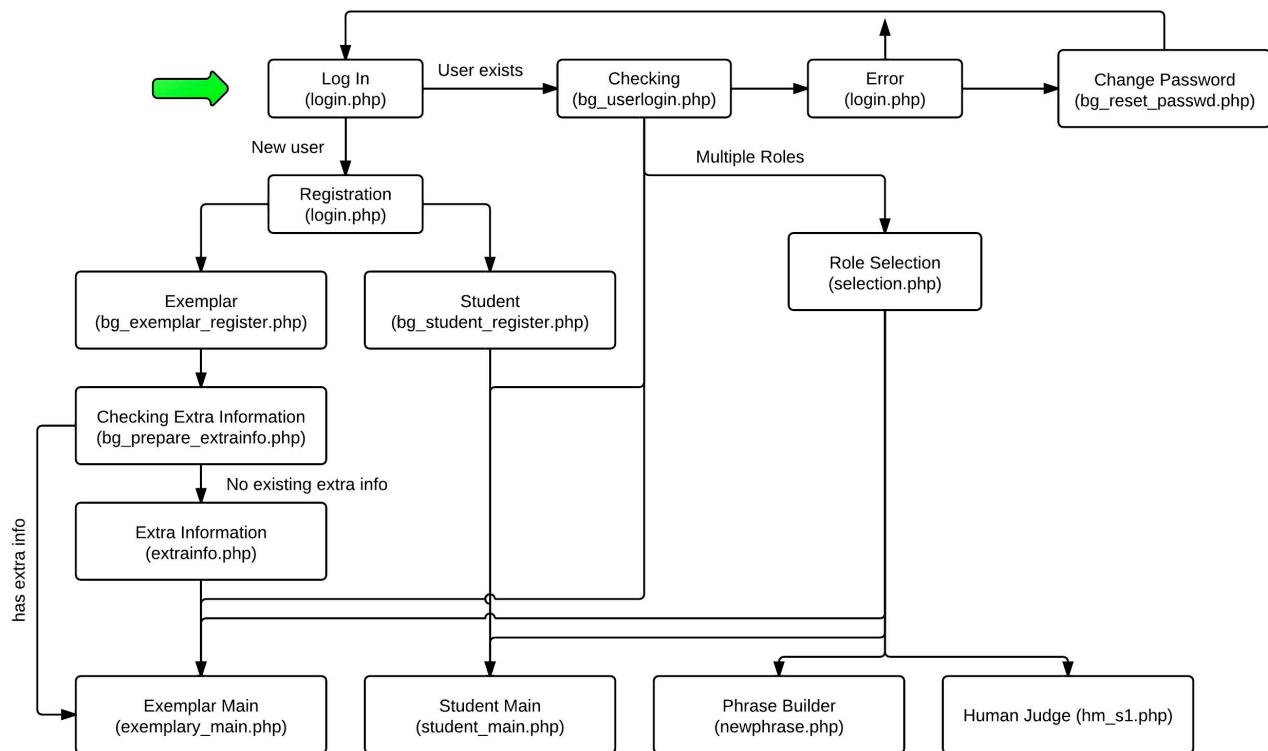
TABLE GAMEINFO

Field	Type	Comments
gameid	INTEGER	Foreign Key, Compulsory, Ref #games id.
nodeid	INTEGER	Foreign Key, Compulsory, Ref #nodes id.
nodepos	INTEGER	The order of the current node in the current game, starts from 0.

CHAPTER 3 USER MANAGEMENT MODULE

Module Structure

The User Management module and the corresponding page navigation information are illustrated in the following figure. The file names in the parenthesis are the actual implementations, which are all under the folder **datacollection** in the repository.



This module is implemented mainly using MySQL database and PHP and HTML5.

Module Implementations

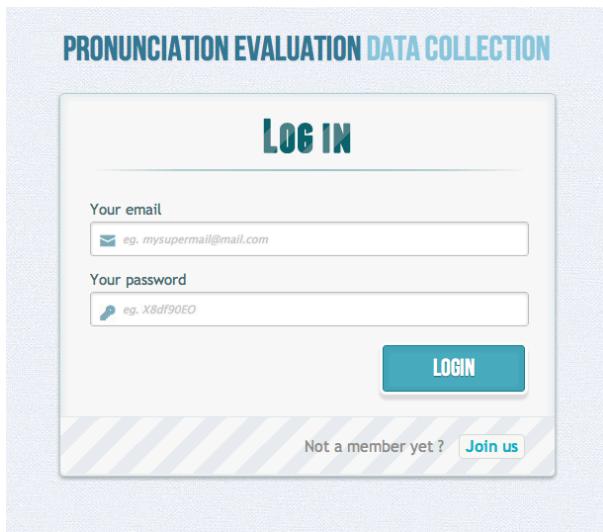
The list of files to implement this module and brief descriptions are tabulated in the following table.

File	Description
login.php	HTML5 page for user login, registration and error messages.
bg_userlogin.php	Check with the database whether the user login information is correct.
bg_reset_passwd.php	Change password.
selection.php	For multiple role users to select the task they would like to do.
bg_exemplar_register.php	Exemplar registration with basic information saved in database.

bg_prepare_extrainfo.php	Checking in database, whether the current user has already had extra information.
extrainfo.php	For user to type in extra information such as native language, birth place etc.
bg_student_register.php	Student registration with basic information.
exemplary_main.php	Exemplar recording main page.
student_main.php	Student pronunciation practice main page.
newphrase.php	Phrase builder page for input new phrases.
hm_s1.php	Human judge evaluation page.

Screenshots

The login page:



The login page features a light blue header with the text "PRONUNCIATION EVALUATION DATA COLLECTION". Below it is a white rectangular form with a teal header "Log in". It contains two input fields: "Your email" with placeholder "eg. mysupermail@mail.com" and "Your password" with placeholder "eg. X8df90EO". A teal "LOGIN" button is centered below the inputs. At the bottom of the form, there's a link "Not a member yet ?" followed by a "Join us" button.

The registration page:



The registration page features a light blue header with the text "PRONUNCIATION EVALUATION DATA COLLECTION". Below it is a white rectangular form with a teal header "SIGN UP". It contains three input fields: "Your email" with placeholder "eg. mysupermail@mail.com", "Your password" with placeholder "At lease 8 alphanumeric characters", and "Please confirm your password" with placeholder "At lease 8 alphanumeric characters". Below the inputs are two teal buttons: "STUDENT" and "EXEMPLARY". At the bottom of the form, there's a link "Already a member ?" followed by a "Go and log in" button.

The additional information page:

PRONUNCIATION EVALUATION DATA COLLECTION

Current User: kate@talknicer.net [Home](#) [Change Password](#) [Update Info](#) [Log Out](#)

ADDITIONAL INFORMATION

Name:

Sex: Male Female

Your birthdate:

Native Language:

Place you lived when you were between 6 and 8:

UPDATE



The password change page:

CHANGE PASSWORD

Old password

New password

Please confirm your new password

UPDATE



The Error page:

PRONUNCIATION EVALUATION DATA COLLECTION

WHOOPS!

You have entered a incorrect password!

Try to [Log In](#) again ; or [Find Password](#)



The password reset page:

The screenshot shows a web page titled "PRONUNCIATION EVALUATION DATA COLLECTION". Below the title, a large button labeled "RESET PASSWORD" is centered. Underneath this button, there is a text input field labeled "Your email" with the placeholder "eg. mysupermail@mail.com". Below the input field is a blue "RESET" button.

The selection page:

The screenshot shows a web page titled "PRONUNCIATION EVALUATION DATA COLLECTION". At the top, it displays "Current User: jack@talknicer.net" and navigation links: "Home", "Change Password", "Update Info", and "Log Out". Below this, a large button labeled "SELECT A MODE" is centered. The page then lists four options, each with a corresponding teal button:

- "For student, please go with" followed by a teal button labeled "PRACTICING".
- "For exemplar recording, please go with" followed by a teal button labeled "RECORDING".
- "For phrase building, please go with" followed by a teal button labeled "PHRASE BUILDING".
- "For evaluating students, please go with" followed by a teal button labeled "JUDGE".

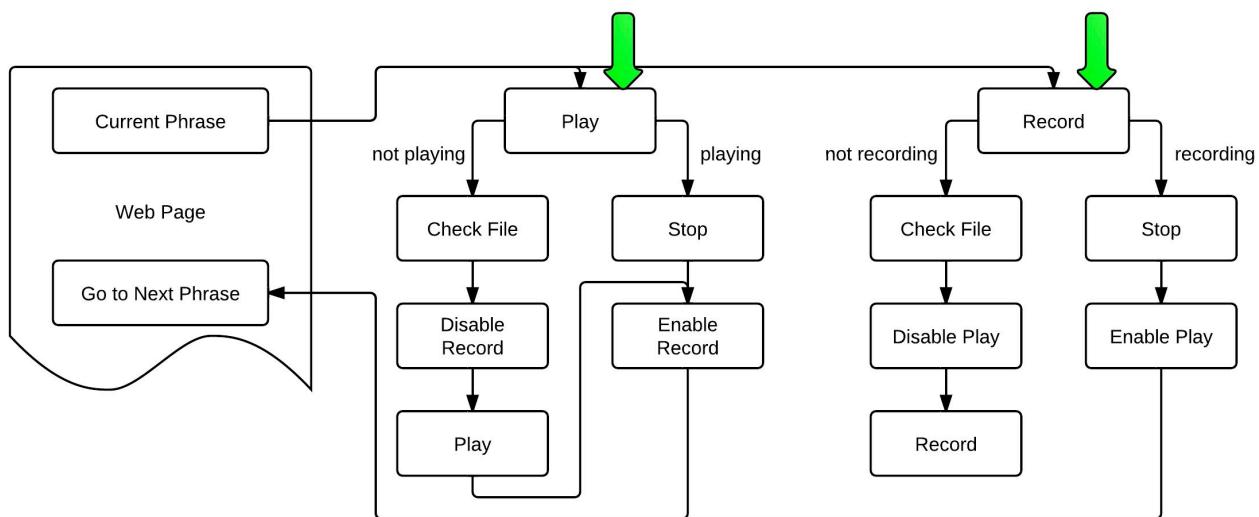
At the bottom of the main content area is a teal "Log Out" button.

CHAPTER 4 AUDIO RECORDING MODULE

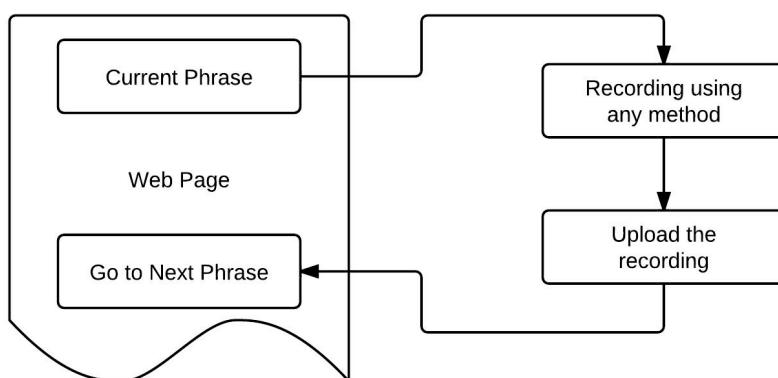
Module Structure

As one of the key component in the pronunciation evaluations system, the audio recording module is implemented using Flex. The player and recorder are Flash based and work well with most of the desktop computers. For mobile devices, which have a large user population nowadays, we have planned to develop platform dependent player and recorder. Currently, we provide the mobile users with the functionality of uploading the recordings for data collection/assessment.

The Flash based audio recording module is illustrated in the following figure.



The fallback mode using HTTP/multipart mechanism to upload the recordings user generated with his/her favorite apps. The fallback model is simply illustrated in the following figure.



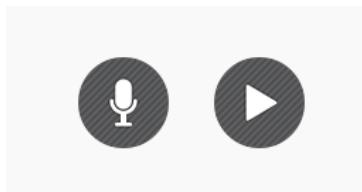
Module Implementation

The list of files implementing this audio recording module and brief descriptions are listed in the following table. The Flex source files are under the folder **audioRecorder/src** and the PHP and JavaScript files are under the **datacollection** folder.

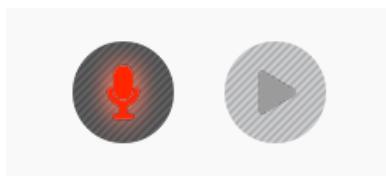
File	Description
audioRecorder.mxml	Implementation of the audio player and recorder.
bg_checkfile.php	Checking the given file name derived from phrase information exists or not.
bg_saverec.php	Save the recording information in the database.
bg_flv2wav.php	Using FFMPEG to convert the FLV recordings saved by the Flash recorder to WAV format.
bg_uploadfile_exemplar.php	Upload a user generated audio file for exemplary data collection.
bg_uploadfile_student.php	Upload a user generated audio file for student pronunciation evaluation.

Screenshots

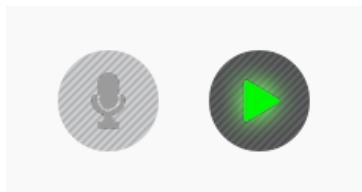
The audio recorder (normal state):



The audio recorder (recording state):



The audio recorder (playing state):



The fallback file upload interface:

UPLOAD THE RECORDING

Seek a solution in building their confidence through a holistic approach.

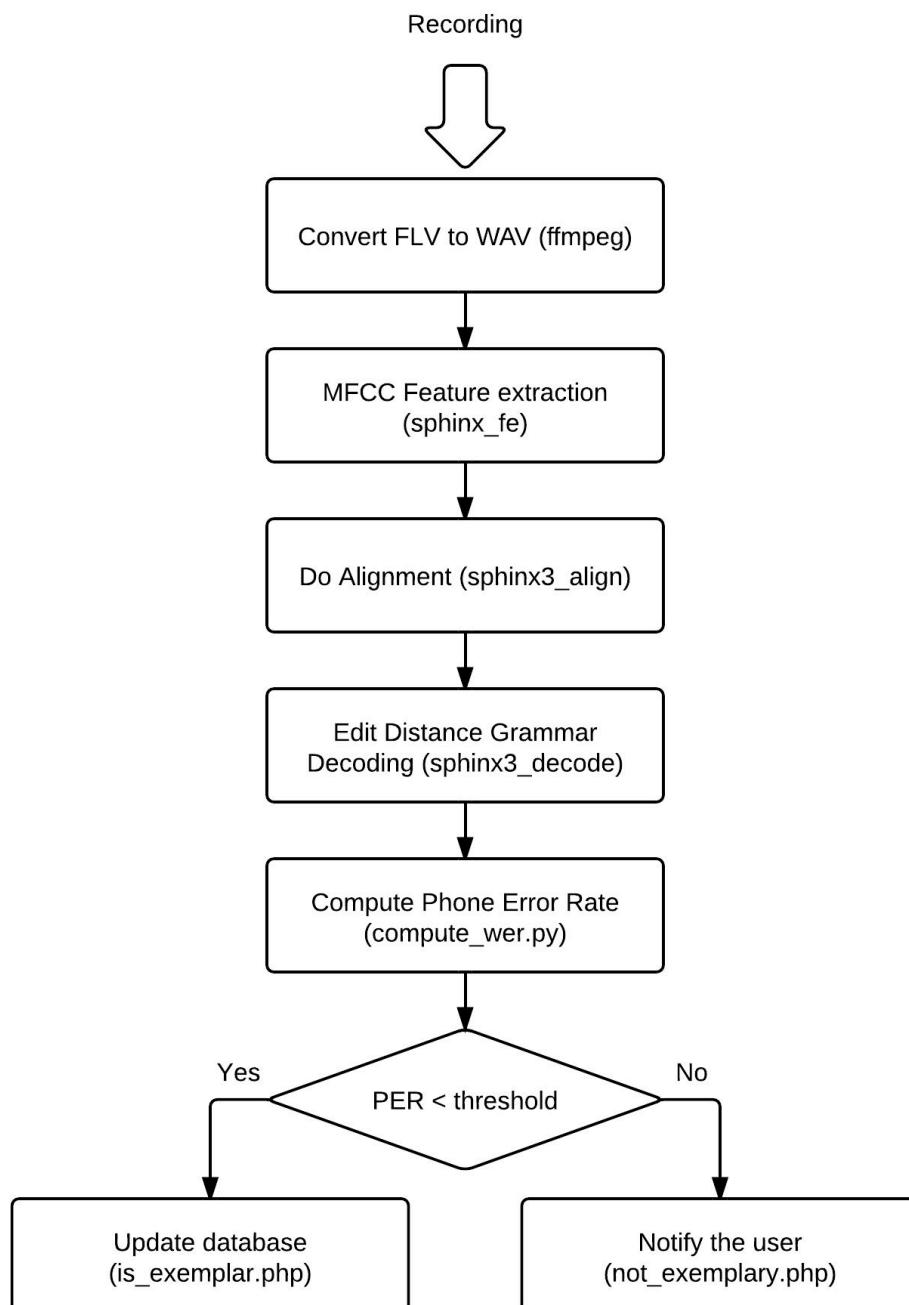
File Name:

CHAPTER 5 EXEMPLAR VERIFICATION MODULE

Module Structure

The exemplar verification is based on the edit distance grammar decoding using CMU-Sphinx tools. If the decoding Phone Error Rate (PER) is smaller than a given threshold, which is usually estimated from the existing exemplars, the recording will be accepted as an exemplar.

The verification procedure is illustrated in the figure below.



Module Implementation

The related files implementing this module are listed in the table below.

File	Description
<code>s0_outlier_analysis.sh</code>	The bash script that implements most of the verification steps including file format conversion, feature extraction, alignment, edit distance grammar decoding and PER computation.
<code>wsj_all_cd30.mllt_cd_cont_4000/</code>	The folder contains the acoustic model we used for pronunciation evaluation
<code>fsgs/</code>	Edit distance grammar for each phrase.
<code>phone.dic</code>	Phoneme list.
<code>phone.filler</code>	Non-speech words.
<code>is_exemplar.php</code>	Update the information for the current recording to reflect in the database that it is an exemplar recording.
<code>not_exemplar.php</code>	Notify the user, the current recording is not an exemplary recording, asking them to redo the recording.

Screenshots

The exemplar-recording page:

The screenshot shows a web-based application interface for recording audio. At the top center, the word "RECORDING" is displayed in a large, bold, teal font. Below it, a instruction message reads: "Seek a solution in building their confidence through a holistic approach." In the center, there are two circular icons: one with a microphone symbol and another with a play button symbol. At the bottom, there are two buttons: "PREVIOUS" on the left and "NEXT" on the right. A note at the bottom indicates that users can upload a recording file if they prefer. The overall design is clean and modern.

Select an item to start: Sentence 2

RECORDING

Select an item to start: Sentence 2

Seek a solution in building their confidence through a holistic approach.

PREVIOUS

NEXT

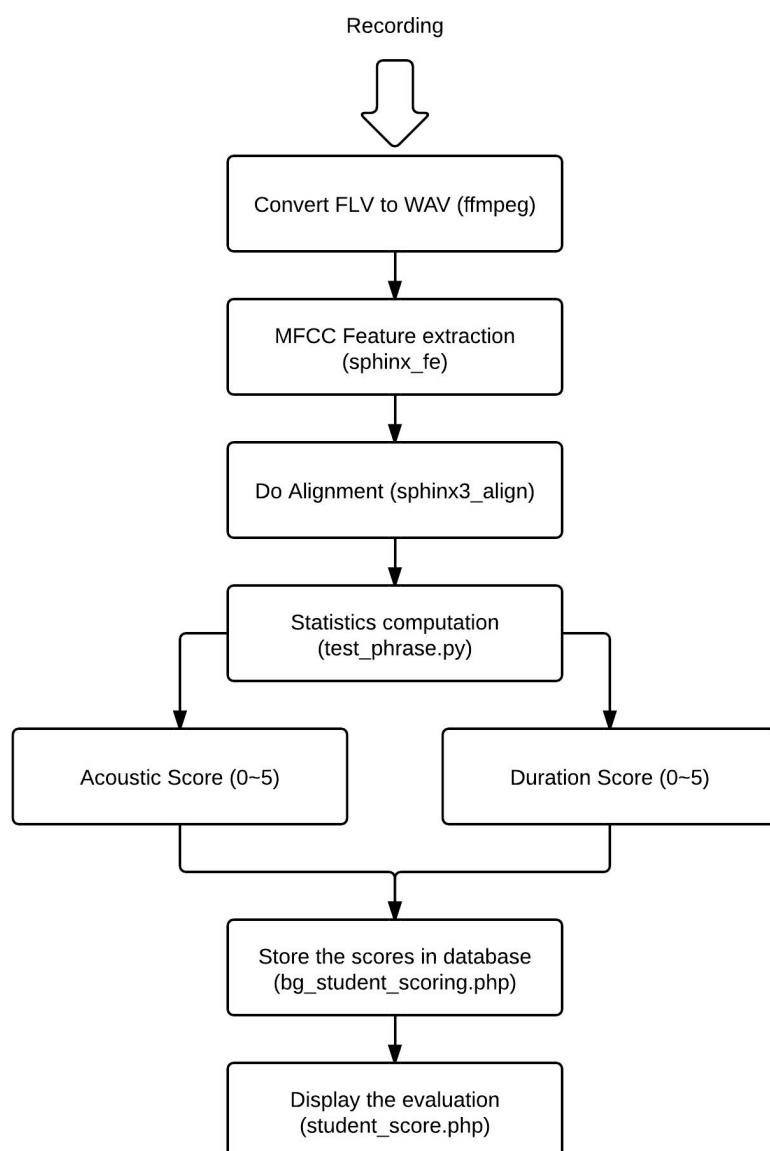
Alternatively, you can [Upload](#) a recording file.

CHAPTER 6 PRONUNCIATION ASSESSMENT MODULE

Module Structure

The student pronunciation assessment module is similar to the exemplar verification module. Instead of decoding with the edit distance grammar, the recording is aligned to the correct transcription and the acoustic and duration statistics are computed for each phoneme and then compared with the corresponding statistics from exemplary recordings. This will give us both acoustic likelihood and duration likelihood of the current user's pronunciation to the exemplary populations. These scores are finally converted into 0~5 scales so that the summation of these two scores would be 10 in maxima.

The scoring procedure is illustrated in the figure below.



Module Implementation

The files implementing this module are listed in the following table.

File	Description
<code>s1_student_scoring.sh</code>	The bash script that implements the major steps for scoring students' pronunciation, including file format conversion, feature extraction, alignment and score generation.
<code>wsj_all_cd30.mllt_cd_cont_4000/</code>	The folder contains the acoustic model we used for pronunciation evaluation
<code>phone.dic</code>	Phoneme list.
<code>phone.filler</code>	Non-speech words.
<code>test_phrase.py</code>	Compute the acoustic and duration score based on the student statistics and the exemplar statistics for each phoneme.
<code>bg_student_scoring.php</code>	Save the acoustic and duration scores for each phoneme into the database.
<code>student_score.php</code>	Webpage for illustrating the assessment results.

Screenshots

The student pronunciation evaluation page:

RECORDING

Select an item to start: Sentence 1

Examine why learners fail to achieve good pronunciation.

PREVIOUS NEXT

EXEMPLARS

Maximum 5 exemplars for the phrase 1 from the database are listed below for your reference.

Sample 0: [play] 0:04 [volume] [delete]
Sample 1: [play] 0:03 [volume] [delete]
Sample 2: [play] 0:06 [volume] [delete]
Sample 3: [play] 0:08 [volume] [delete]

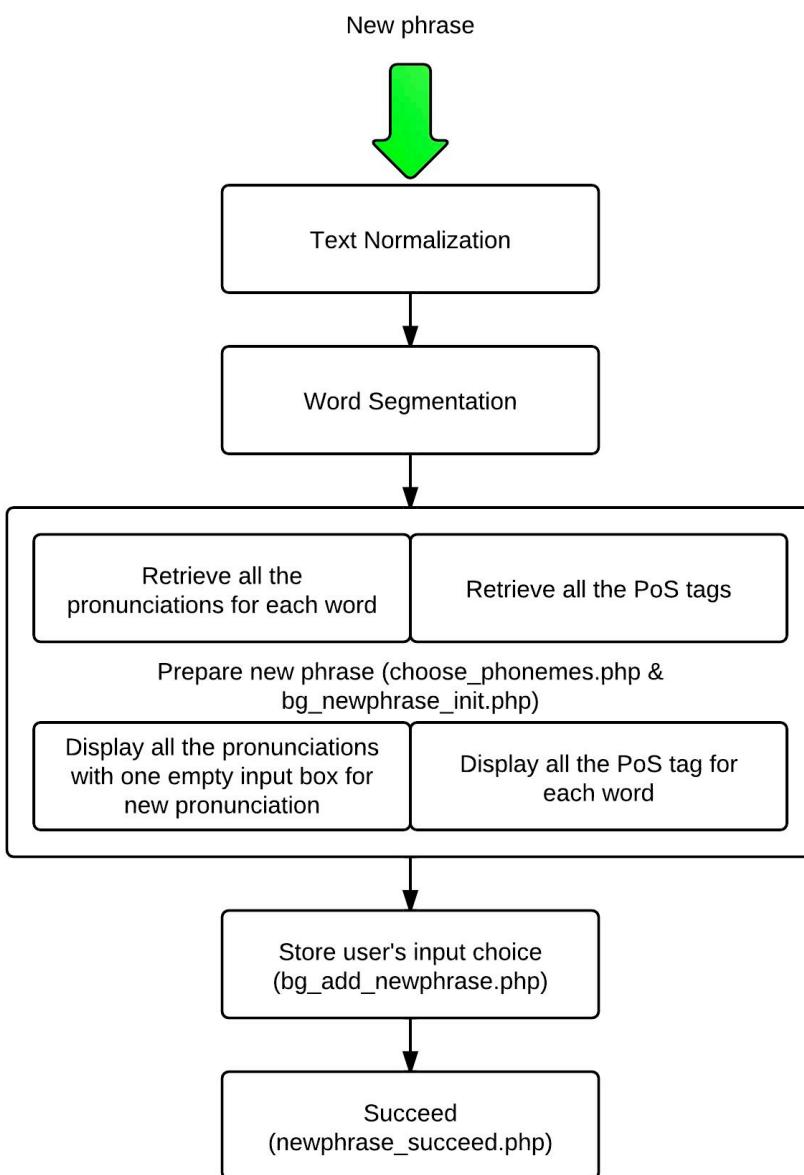
Alternatively, you can [Upload](#) a recording file.

CHAPTER 7 PHRASE LIBRARY BUILDING MODULE

Module Structure

The phrase library building is a relative simple module. The main processing is the retrieval of existing pronunciations for each word in the phrase the user input and displays them for user to choose the expected one. Moreover, if a pronunciation does not appear in the current database, the system should allow the user to type in the pronunciation and update the database accordingly. Additionally, the system should also ask the user to select the Part of Speech tag for each word in the phrase as some words' pronunciation are dependent on the role they play in the phrase.

The processing procedure for the creating a new phrase is displayed in the figure below.



Module Implementation

The list of files together with brief descriptions is tabulated below.

File	Description
<code>newphrase.php</code>	Webpage for user to type in a new phrase.
<code>choosephonemes.php</code>	Retrieve all the possible pronunciations for each word.
<code>bg_newphrase_init.php</code>	Retrieve all the PoS tags for each word and display both the word and PoS tags as options for user to select.
<code>bg_add_newphrase.php</code>	Store the new phrase together with corresponding correct pronunciations into database.
<code>newphrase_succeed.php</code>	Notify the user the creation of the new phrase is successfully done.

Screenshots

The phrase input page:

The screenshot shows a light gray rectangular form with a title 'TYPE IN A NEW PHRASE' at the top in bold blue capital letters. Below the title is a section labeled 'New Phrase' with a placeholder 'eg. new phrase' containing a small blue envelope icon. A large blue button labeled 'CONTINUE' is centered below the input field. The bottom of the form has a light gray diagonal-striped pattern.

The pronunciation selection page:

The screenshot shows a light gray rectangular form with a title 'CHOOSE THE PRONUNCIATION' at the top in bold blue capital letters. Below the title is the text 'hello world' in a large, stylized blue font. A table titled 'Word' is displayed, showing two rows for the words 'hello' and 'world'. Each row contains three columns: 'Word', 'Pronunciation', and 'PoS Tag'. The 'Pronunciation' column lists multiple options for each word, with the correct one selected. The 'PoS Tag' column lists various parts of speech with radio buttons for selection. A blue 'SUBMIT' button is located at the bottom of the table. At the very bottom of the page, there are links for 'Add another phrase' and 'Log Out'.

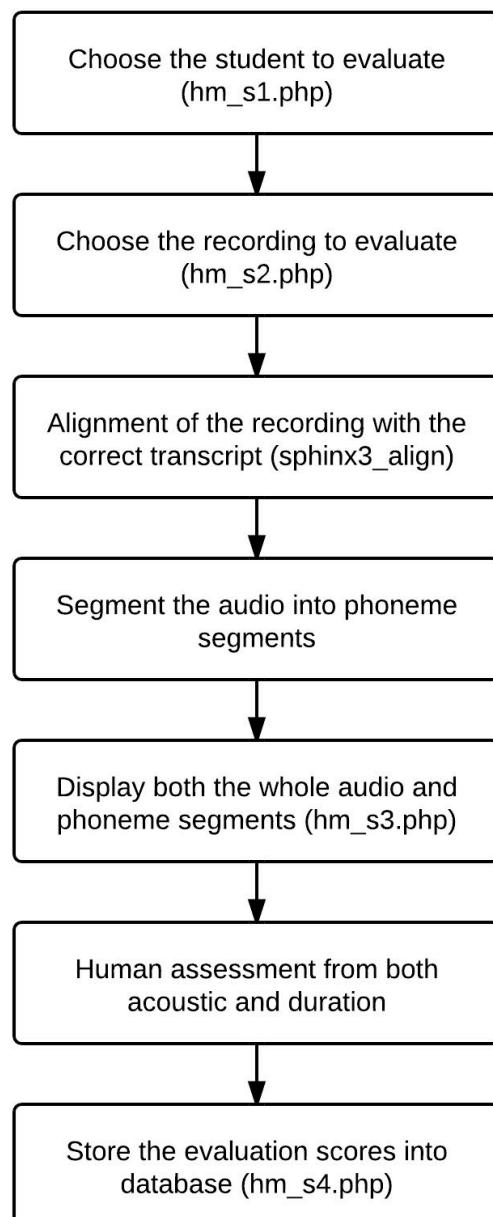
Word	Pronunciation	PoS Tag
hello	HH AH L OW HH AH L OW HH EH L OW HH AA L OW	<input checked="" type="radio"/> Quantifier <input type="radio"/> Noun <input type="radio"/> Verb <input type="radio"/> Negative <input type="radio"/> Adverb <input type="radio"/> Adjective <input type="radio"/> Pronoun <input type="radio"/> Possessive <input type="radio"/> Preposition <input type="radio"/> Conjunction <input type="radio"/> Article
world	W ER LD W ER L D	<input checked="" type="radio"/> Quantifier <input type="radio"/> Noun <input type="radio"/> Verb <input type="radio"/> Negative <input type="radio"/> Adverb <input type="radio"/> Adjective <input type="radio"/> Pronoun <input type="radio"/> Possessive <input type="radio"/> Preposition <input type="radio"/> Conjunction <input type="radio"/> Article

CHAPTER 8 HUMAN EVALUATION MODULE

Module Structure

The last module implemented during this GSoC project is the human experts evaluation module, which allows human teachers to judge the students' pronunciation performance. The main purpose of implementing this module is to allow deeper understanding of the performance of our automatic pronunciation evaluation system and to further improve our system performance towards human performances.

This module is implemented as shown in the following process.



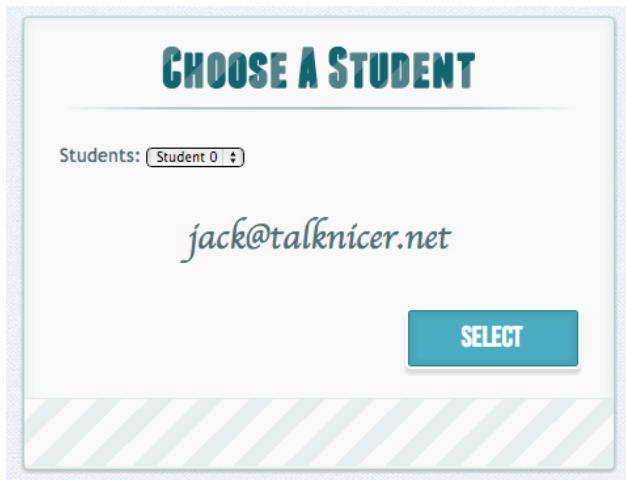
Module Implementation

The files implementing this module are listed below.

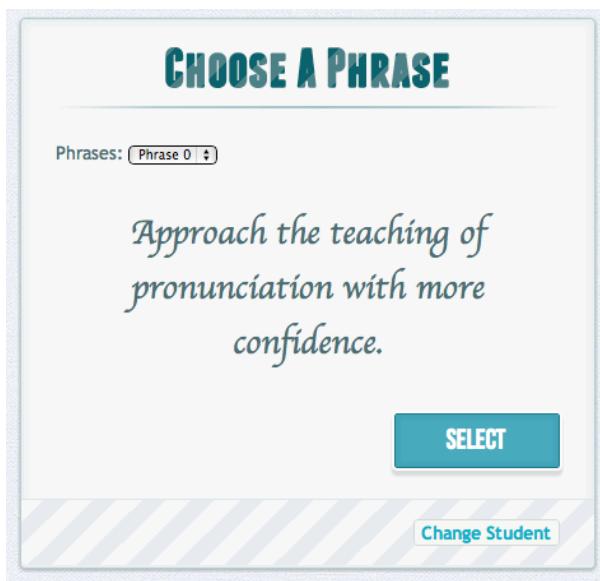
File	Description
hm_s1.php	For the teacher to choose which student to evaluate.
hm_s2.php	For the teacher to choose which phrase to evaluate.
hm_s3.php	Retrieve the word and phoneme information for the recoding and do alignment to get the phoneme segmentation. Finally display both the whole audio and phoneme segments. Each segment will be accompanied with an acoustic and duration scoring slider.
hm_s4.php	Store the teacher's evaluation.

Screenshots

The student selection page:



The phrase selection page:



The human evaluation page:

PRONUNCIATION EVALUATION JUDGE PAGE

Current User: kate@talknicer.net [Home](#) [Change Password](#) [Update Info](#) [Log Out](#)

ASSESS THE PRONUNCIATION

[jack@talknicer.net]

Approach the teaching of pronunciation with more confidence.

0.00

Phoneme	Acoustic Score	Duration Score
AH	2.5	2.5
P	2.5	2.5
R	2.5	2.5
OW	2.5	2.5
CH	2.5	2.5
DH	2.5	2.5
AH	2.5	2.5
T	2.5	2.5
IY	2.5	2.5
CH	2.5	2.5
IH	2.5	2.5
NG	2.5	2.5
AH	2.5	2.5
V	2.5	2.5
P	2.5	2.5
R	2.5	2.5
AH	2.5	2.5
N	2.5	2.5
AH	2.5	2.5
N	2.5	2.5
S	2.5	2.5
IY	2.5	2.5
EY	2.5	2.5
SH	2.5	2.5
AH	2.5	2.5
N	2.5	2.5
W	2.5	2.5
IH	2.5	2.5
TH	2.5	2.5
M	2.5	2.5
AO	2.5	2.5
R	2.5	2.5
K	2.5	2.5
AA	2.5	2.5
N	2.5	2.5
F	2.5	2.5
AH	2.5	2.5
D	2.5	2.5
AH	2.5	2.5
N	2.5	2.5
S	2.5	2.5

SUBMIT

CHAPTER 9 CONCLUSIONS

Discussions

During these two months' participation in the GSoC project with my mentor James Salsman and Nickolay V. Shmyrev's guidance, and my team member Srikanth Ronanki's help, we have managed to put up this pronunciation evaluation web portal online. With these seven modules implemented the system now could play its function now. However, due to the underlying exemplar based evaluation, we need to collect more exemplar data for each phrase to be evaluated before the system could give users more contents for evaluation. Moreover, for a online system to be robustly used, lots of testing are needed. I also would like to thank Robert Butler's help in testing the website and providing us bug reports.

Future work

Although the GSoC is approaching the end, our pronunciation evaluation project is still in its beginning stages, a promising start. The web portal has already been equipped with all necessary functionalities. Mobile clients are also under development. Currently, my mentor James and I are investigating ways of utilizing Amazon Mechanical Turk to boost the exemplar data collection, which will lay a solid foundation for the automatic pronunciation evaluation system to be launched officially. Meanwhile, we have already finished the game based pronunciation evaluation database design and implementation. In future, we will settle down the game contents including the game scenes, background music and especially the pronunciation evaluation phrases. With the Mechanical Turk collected exemplar recordings, the implementation of the game module will also become feasible. I have strong confidence that this project will become a successful story of improving people's lives with open source technologies.