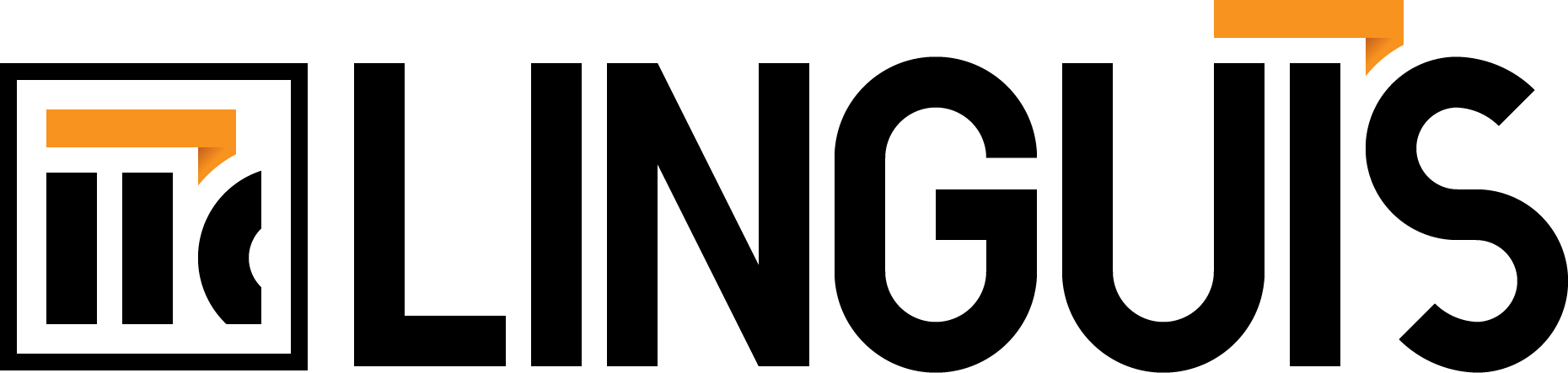
**Software Engineering Principle 13016214**

Project Report



Phonology Teaching Application

**58090002** Damian Satya Wibowo

**59090025** Punn Ouilapan

**59090038** Warakorn Jetlohasiri

Software Engineering, International College KMITL

**I. Linguīs Project Description**

Linguīs, meaning ‘into languages’ in Latin, is a phonetic learning software which aims to provide learners of languages and linguistics the resources and tools to make their phonology learning process more effective and convenient. It also aims to serve as a resource hub for easy access of language and linguistics knowledge. The app will achieve said goals by offering features such as the International Phonetic Alphabet chart completed with pronunciations, as well as access to various phonology topics. The users will be able to mark their most used topics, as well as IPA characters, for quick access.

The application will be written in Python 3 and will utilise PyQt5 to render its Graphical User Interface. This is to illustrate the powerful capabilities of the Python programming language, as well as the flexibility of the Qt GUI framework.

The application will be split into four primary sections as followed:

**International Phonetic Alphabet**

The IPA Chart section will consist of two parts. The latter part is a complete interactive chart of all the International Phonetic Alphabet glyphs, which users can interact with to hear the pronunciation in its initial, medial and final position (consonants) and independent position (vowels)*.* The users will also be able to save the character to their favourites list for easy access.

**Phonology Information Hub**

The Information Hub section will consist of a database of information on phonology topics, including an overview containing information about International Phonetic Alphabet and its usefulness. Each topic in the information hub will also be able to be favourited for easy access. Users can expand the information by installing additional databases or revisions into the application.

**Quizzes**

There are sets of quizzes which can be used to test and check the learning progress of the user. Three types of questions are: 1. Multiple choices, where a user should choose one correct answer out of five options; 2. True/false, deciding whether a statement is true or not; 3. Fill-in-the-blank, where a user should fill the correct answer.

**Favourites**

The Favourites section is will contain all the glyphs and information topics that the user has marked to allow the user to access them easily. The favourites will be organised systematically according to their type: phonemes and content topics.

**Extra: Updater**

Linguīs can be updated! An updater file is a separate program to update the firmware of Linguīs. *[][][][][][][][][][][][][[][][][[][][][][][[][[][][][]][][][*

**II. Software Requirements**

**User, functional requirements:**

1. Linguīs provides user to listen to each IPA symbols’ pronunciation at initial, medial and final position (consonants) and independent position (vowels).
2. Linguīs provides arrays of information about various topics about phonology.
3. By doing the questions in the quiz section, user can train their phonology knowledge.
4. Users are able to save their favourite topics and IPA symbols into Favourites list.
5. The text-to-speech section is able to convert IPA text into an audio output.
6. Outputted audio from the text-to-speech section can be saved to user-specified location.
7. Users can update the application to the newest version by installing update through the options menu.
8. Users can report bug to the developer.
9. Users can preserve their login information (‘remember me’ section) in the login page.
10. Users can pronounce their favourite phonemes and/or jump to their favourite articles.

**User, non-functional requirements:**

1. Linguīs notifies users through e-mail when an update is delivered.
2. The user interface will be easy to navigate, with information categorised into sections.
3. Information about the development of Linguīs as well as its Github page will be available within the application.
4. Linguīs is provided in English.
5. There is a help menu that provides users a guide to operate the application and navigate between menus.
6. Favourited topics are saved in a separate favourite menu.
7. Quizzes’ are equipped with scores and evaluations to tell user their progress and recommended future study.
8. User statistics and biodata can be seen from the Options menu.
9. ,

**System, functional requirements:**

1. The text-to-speech converts IPA symbols and match the sequence with its corresponding IPA pronunciation sound file, and play the sequence.
2. Update is installed by calling the updateApp() function inside the Updater object in a separate Updater application which will modify some specified application data which are updated.
3. The text-to-speech output file can be saved into the local storage as a .wav file.
4. The application can evaluate quiz results by accumulating weighted scores in various topics.
5. When a quiz is open, list of questions is load to the QListWidget, then selected question and its answer field are shown in a QWidgetStack object besides the QListWidgetList.
6. Quizzes are stored in a text file named quizzes.txt which contain a formatted text. Loading quiz involves parsing the text file into QuizList objects, and ultimately, broken down into quizzes.
7. Favourites jump/pronounce
8. When the user quits the main menu/logging out, user’s newly user obtained/modified data is stored to userlist pickle file.
9. ,
10. ,

**System, non-functional requirements:**

1. The user interface of the software will be modelled or designed with PyQt5 Graphics User interface library.
2. The software requires PickleDB in order to provide approximately 4 files for different types of data collection. These different files will be used to collect different types of data information based on variety usage. For example, database for registration system, user list, etc.
3. The software maintenance and development will be controlled and implemented under GitHub software version control system.
4. All of IPA symbols and its corresponding IPA pronunciations’ audio file will be stored in local storage and mapped in a dictionary (stored as key and values) which reduce difficulties for visualization needed for implementation and software maintenance
5. The software user interface supports the minimum screen resolution of 900 x 600 pixels which generally supports the majority of monitoring devices in worldwide.
6. The application is written in Object Oriented Concept, using Python 3.
7. In case of a system error, an exception handler will be triggered to notify user or/and close the application.
8. Linguīs involves the MVC (Model, View, Controller) concept in its software design.
9. User IDs cannot be duplicate in the same device, and passwords should comply with certain format: contains at least one lowercase letter, at least one capital letter, at least one numeral and at least one symbol, and the length should not less than 6 or more than 12 character.
10. The user’s favourites list is stored locally, inside lists in the User class.

**/Design**

**use cases (3 basic, 3 alt, 3 exc), others: basic**

**3 sequence diagrams (basic, alt, ext)**

**other sequence diagrams: basic only**

**3 State charts**

**Full UI-system UML**