Software Engineering Principle 13016214

Project Report

Linguīs, a Phonology Teaching Application.

58090002 Damian Satya Wibowo

59090025 Punn Ouilapan

59090038 Warakorn Jetlohasiri

Software Engineering, International College KMITL

**/Introduction/**

**I. Linguīs Project Description**

Linguīs, ‘into languages’ in Latin, is a phonetic and linguistics learning software which aims to provide learners of languages and linguistics the resources and tools to make their 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 Full International Phonetic Alphabet chart with complete pronunciations, access to various linguistic topics. The users will be able to mark their most used topics, as well as IPA characters, for quick access. The user will also be able to share their favourites list.

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 two primary sections as followed:

**International Phonetic Alphabet**

The IPA Chart section will consist of two parts. The first part would be an overview containing information about International Phonetic Alphabet and its usefulness. The latter part is a complete interactive chart of all the International Phonetic Alphabet glyphs, which users can interact with to view in detail the description of each character, as well as its pronunciation and oral position. The users will also be able to save the character to their favourites list for easy access.

**Linguistics Information Hub**

The Information Hub section will consist of a database of information on linguistics topics such as Phonology, Orthographies, and more. 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.

Other in-section features in Linguīs are:

**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. The user will also be able to attach their own notes to the snippets to aid in remembering or understanding.

**Quizzes**

Each section contains a quiz that relates to the corresponding material to test and check the learning progress of the user.

**/Requirement (UF, UNF, SF, SNF)/**

**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. Users are able to train their linguistics knowledge by providing a quiz section.
3. Users are able to save their favourite topics and IPA symbols into Favourites list.
4. The text-to-speech section is able to convert IPA text into an audio output.
5. Users can share their favourite list.
6. Users can modify the application by adding customized features.
7. The application provides search topic section to enable user to easily jump to any topic within the application.
8. Users can change the appearance of the user interface, i.e. background colour, text colour.
9. Users can update the application to the newest version by installing update through the options menu.
10. Linguīs provides users to change the application’s user interface language to the given language packs or make a new language pack.
11. Users can report bug to the developer.

**User, non-functional requirements:**

1. Linguīs notifies users when it is time for them to revise.
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. The modification feature must not allow users to modify the application in a way that will cause it to malfunction.
5. The user’s favourites list will be stored locally.
6. Important user activity are logged. Useful for debugging and solving issues.
7. Linguīs is provided in English, with possible language extensions.
8. There is a help menu that provides users a guide to operate the application and navigate between menus.
9. Favourited topics are saved in a separate favourite menu, sorted by the type of topic (page, symbol, etc.)
10. Quizzes’ are equipped with scores and evaluations (strengths or weaknesses) to tell user their progress and recommended future study.

**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. When users trigger the favourite button, user data is loaded by pickle, adding/removing the corresponding favourited topic, and the updated data is dumped back to the file.
3. Update is installed by calling the updateApp(updateFile) function which will modify some specified application data which are updated.
4. The text-to-speech output file can be saved into the local storage as a .wav file, and appear in users’ favourite list.
5. When users switch the interface language, the application will restart and a new application object with a new language will be made.
6. The search function can search the keyword of each topic within the application using sequential or hashtag search.
7. Linguīs can run animations within the application by switching images through .jpg image files, sequentially.
8. The application’s user interfaces can be switched (from one menu to another menu) by clicking hyperlinks in a page, and calling screenUpdate().
9. Linguīs checks for updates automatically by scanning whether there is an update file in the update package.
10. The application can evaluate quiz results by accumulating weighted scores in various topics.

**System, non-functional requirement:**

1. The user interface of the software will be modelled or designed with PyQt5 Graphics User interface library.
2. The software requires the database called PickleDB in order to provide approximately 5 tables for different types of data collection. These different tables will be used to collect different types of data information based on variety usage. For example, database for registration system, linguistic study resources, etc.
3. The software maintenance and development will be controlled and implemented under GitHub software version control system.
4. The privacy of software will be strongly protected under supervision of software administrator. Every private information will be encrypted and being stored in the database. Also, administrators will have permission to access private information using a unique decryption method after a user’s privacy access permission.
5. All of IPA symbols and its corresponding IPA pronunciations’ audio file will be stored in PickleDB in which possess similar properties as dictionary (stored as key and values) which reduce difficulties for visualization needed for implementation and software maintenance
6. The software user interface supports the minimum screen resolution of 1366 x 768 pixels which generally supports the majority of monitoring devices in worldwide.
7. The software can automatically undergo its latest version of software update when the system connects to the internet broadband at minimum speed of 1.2 Mbps for software optimization to handle software updates.
8. The system will extract all statistics recorded in a database table and complete statistical analysis procedure required for generating questions in a quiz feature in order to enhance learner’s understandings and linguistic capability.
9. The application is written in Object Oriented Concept, using Python 3.
10. In case of a system error, an exception handler will be triggered to notify user or/and close the application.

**/Design (3 use case (basic, alt, exc)+ sequence, others: basic; UML (UI, system); State chart (3); Source code/**