

SOFTENG 206 Software Engineering Design 1

Department of Electrical, Computer, and Software Engineering

Project

The project will be in pairs.

Deliverables

- **14th October 2019, 9am (Week 11)**
Beta version of the project which will include (i) **source code**, (ii) easy-to-run Linux **executable**, and (iii) **readme file** with installation and basic usage instructions.
- **25th October 2019, 5pm (Week 12)**
Final version of the project which will include (i) **source code**, (ii) easy-to-run Linux **executable**, and (iii) **report + user manual** discussing the functionality, development and evaluation of the project.

The class will be involved in finalizing the project requirements and assessment.

VARpedia: Visual Aural Read encyclopedia

The project will involve developing an encyclopedia tool that provides **visual**, **aural** and **reading** capabilities for its users. The information content in the VARpedia will be sourced from Wikipedia. When a user searches a term, it is presented in all three VAR formats. You will be using Java to create the user interface, invoking Linux system calls from the Java application.

Whilst the actual coding and design of the tool will take place in the latter part of the semester, preparation for the project has actually been from the beginning of the course. The following table outlines the development process of the project:

Week	Activity
2-6	Lectures on Linux commands, graphics, begin implementation
4	Assignment 1 Linux command line interface
6	Assignment 2 Linux with Java GUI
9	Assignment 3 on the prototype of VARpedia
9	Assignment 3 Presentation
11	BETA version of project due in (no submission → automatically lose 20%)
11	Peer Evaluation of aids
12	Report and final code for project due

Project

The project involves creating a **visual**, **aural** and **read** encyclopedia. It will be called VARpedia. Some schools of thought believe that different learners have different learning preferences or styles (e.g. the VARK model: visual, aural, read/write and kinaesthetic). Other schools of thought simply recognise that presenting information in multiple formats is best for enhancing learning (i.e. its not about the learning preferences, but rather showing the same information in multiple modes).

The content of the searched term will be presented to the user in writing (so the user **reads** it much like any other encyclopedia). The user will then be able to listen to selected parts of the content (i.e. **aural**). To support the above, photos will also be included to the user (i.e. **visual**). All these components come together to form what is known as a “creation”, authored by learner themselves (i.e. the user of the application). The text will be sourced from Wikipedia, audio will be created using a speech synthesizer, while the images will be sourced from Flickr. Finally, there should be a **quiz**-component to the application that supports active learning. This is a section of the application where users engage with the creations they have made in a pedagogically-meaningful manner. This is elaborated below. ~~can guess the term by looking at the photos and/or listening to the audio. They input their guesses by writing.~~

In addition to the above core features, you will be providing other functionality to support active learning. Users will be able to save their favourite searched terms (known as *creations*) to be reviewed later on without requiring them to regenerate the content from scratch. They should be able to ~~tweak~~ define the content when they make the creations (for example, what synthesised text and images are included in the creation), specify which subset of content should be included in the creation, and so on. They should also have the option to add background music to the creation, in addition to the synthesised text. The ~~quiz~~ active learning component will be ~~constructed~~ composed from ~~terms/creations already existing.~~ downloaded.

The features of the application will include:

- A well designed graphical user interface.
- Help on how to use the different features of the application.
- Term searching abilities.
- Sound playback in the creations, both speech and music.
- Video playback (created from images).
- Creation and saving of terms (creations) for faster future playback.
- An active learning component, incorporating the creations that the user has already created. You get to decide which active learning component to implement using the examples provided below.
- ~~A quiz component where questions (creations) are presented in a visual and aural manner (i.e. just photos/audio), and the user inputs answers by typing.~~
- ~~A scoring mechanism for the quiz, which might also include a timing function, including basic statistics of scores and difficult terms that need more study.~~
- It must run on Linux, in particular it will be tested either in the UG4 Linux image (beta or main), or the provided VirtualBox image. You must provide a README file that clearly specifies which system/image to test it on.

Examples of active learning component

You should implement one of the following features:

- **Simple matching game:** The game session involves presenting the user with presented with pre-created creations to the user. The difference though, is that the creations are presented with at least the (search) term missing. For example, the video is presented with the pictures (and possibly the audio). The user is then tasked with specifying what the term is. They can either type in the term, or select from a combination of terms. You can decide whether you want to show “one at a time” (and they type the term), or you can show multiple creations at the same time and they match them with terms you provide.

- **Review reminder system:** The idea here is that after the learner creates a range of creations, they get archived. Ultimately, they should be encouraged to review them in order to reinforce their learning. The application will scaffold this review process by keeping track of which items the user should prioritise in reviewing. The priority order is based on a combination of (i) user confidence ratings (the user gives a confidence rating for each creation, where lower-rated confidence creations get higher priority for reviewing), and (ii) least-reviewed creations (oldest-reviewed creations get highest priority). The user might decide to rank priority by confidence first (and then oldest-reviewed), or rank priority by oldest-reviewed first (and then confidence).

When you design the aid you will need to keep the target user in mind, this will have a direct impact on the GUI which is the actual spelling interface. You need to **choose one** of the following three target user options:

- a. A child user (7-10 years).
- b. A second language learner, young adult (18-25 years).
- c. An elderly user.

Design Decisions

There are many choices you can make in regards to the graphical user interface, and also in terms of how the information is displayed. Think carefully about these as you will be asked to justify them in the accompanying report. For instance you will need to consider the window design, the choice of colours for both the background and the plots. What kinds of menus are you going to offer the user, and what about button design? Will you have key shortcuts? These are only a few of the decisions you will be making. Whatever your decisions, it is important to discuss these fully and justify them in your report.

VARpedia – User Manual details (group submission)

As part of the group submission, you should supply a user manual for VARpedia. This will help introduce users to the application and explain how to use the core functionality. The user manual must take your target user into account (i.e. it should not be technical). Make sure the user manual is submitted as a PDF.

VARpedia – The report details (individual submission)

As an individual component, you must also include a report in the IEEE ~~double~~single columned style (about 2000-3000 words which is 4-6 pages without pictures, but more once pictures are added). The intended reader of this report will be the chief technical officer, who will be assessing whether to fund you to complete the VARpedia product. They are particularly interested in the design and developmental skills you have used in building VARpedia, and the features of the final product that you have created. The chief technical officer strongly encourages you to include relevant figures in your report. The intended reader of this report will be the chief technical officer, who will be assessing whether your program, the VARpedia, is potential candidate for a marketable product. There is no need for you to address any marketing issues. The report will need to contain:

A user manual for VARpedia:

1. Information about the features of VARpedia, highlighting the design decisions made, for example the design of the learning modules, what is used for the graphics, what was the reason behind the choice of colours, etc. Be sure to justify all the approaches taken.
2. Details about who the intended users of VARpedia might be.

3. Details about the software development process, for example: how was version control dealt with? How was the work load distributed? What would you do better next time? What worked well?
4. Details about how VARpedia was evaluated; both self and peer, include here such things as: comparisons to other systems (in your class and in general), what changes you would make to your tool as a consequence of the peer evaluations, and justify your reasons. You may decide no changes would be necessary, but you need to also justify these, also you should discuss what exactly has been evaluated and what has not.
5. Details about how your team worked: how did you work together to complete the project? Did you split tasks, or did you work on everything together. How did you reach a consensus on a design decision, how did you resolve disagreements. Looking forward to design in Part 3, where you work in even bigger teams, what experiences from this year would you like to recreate.
6. Table of contributions in your group. Mention what each member contributed towards in the collaboration, based on the major tasks. For each task, associate a percentage for each member. Include the overall contribution percentage also. Obviously you cannot give exact/accurate percentages, but it should be illustrative how how YOU felt the work was split between you and your partner. You should NOT discuss this with your project partner, it should remain confidential and be your own personal view on the workload balance.

Use of external code or information

You may occasionally use snippets of code or information from other sources if you wish, but you **MUST** indicate where you have done so and reference the original work. The majority of both the program code must be your own. If you have included someone else's work without clearly indicating each and every usage, we will have to assume the intent was to pass the work off as your own, and take appropriate action.

Marking Schedule

Marks will be allocated for the project on:

- Functionality of the application
- Code quality
- Code documentation
- Development documentation (written in the journal)
- Report
- User manual

We will release the actual marking schedule of the project in week 10, after a discussion with the class.