

CS 242 Final Project Proposal

GRE Word Memorizing and Categorizing Application

Ke Lu (kelu2), Jing Zhao
Yuanyuan Zhao (zhao102), Peiyuan Zhao

1. Abstract

1.1. Project Purpose

The purpose of this project is to help students who need to take GRE test in order to attend Graduate schools.

The problem we want to solve is to alleviate the difficulty students feel when they try to memorize tons of unfamiliar words that are hard to understand.

1.2. Background/Motivation

Because one of the members in experiencing the painful process of memorizing lots of words in order to take GRE.

We had incentives of doing something like this, but this is the first time we put it into practice.

2. Technical Specifications

2.1. Platform: iOS Application

2.2. Programming Languages: Swift, MySQL

2.3. Stylistic Conventions: (i.e. commenting, naming conventions, camelCase)

Multi-line commenting

camelCase naming for functions

Lower case naming for files

First letter capital for naming for classes

Same-line curly braces

2.4. SDK: iOS 10 SDK

2.5. IDE: Xcode

2.6. Tools/Interfaces: iPhone6 Plus

2.7. Target Audience: GRE Test takers who use iPhone

3. Functional Specifications

3.1. Features

Several bullet points of what kind of functionality your project will feature. What should the user be able to do?

Memorizing GRE words more easily.

Be able to keep track of the progress.

Be able to rate the familiarity of each word.

Be able to share progress with others.

Vocabulary level test.

User can customize the lexicon or switch lexicon.

Users can communicate and share experience on a forum.

Users can switch between daytime/nighttime mode and color themes.

Offline Availability--User will still be able to access the words even without network.

Sentence of the day.

Practice test functions.

Word list with different levels of familiarities.

3.2. **Scope of project** Mobile Application

May not be able to receive authorization from the most popular lexicon

May not be able to apply the most efficient memorizing pattern and algorithm

Timeline:

Blue lines are works allocated to Ke Lu;

Black lines are works allocated to Yuanyuan Zhao.

3.3. **Week 1**

Set up building environment

User Interface flow chart design

Build the basic structure of application on Xcode with a simple User Interface (UI)

Find a viable lexicon that we can use and retrieve the data

3.4. **Week 2**

Implement the User Interface (UI)

Practice test questions

Word list basing on level of familiarity

Design an icon for the application

[Parse the acquired data into processable file and save it online](#)

Design algorithms for word selection

[Design algorithms for sequence arrangement](#)

3.5. Week 3

Build vocabulary level test module

[Build progress tracking module](#)

Build lexicon modifying and switching module

[Implement the online forum based on online database](#)

Build module for theme switching

[Build module day-night-mode switching](#)

3.6. Week 4

Sentence of the day

[Integrate the algorithms with the application structure](#)

Testing on overall functionalities

[Find actual users to try the application and improve minor features basing on feedbacks](#)

4. Future Enhancements

Ability for fusion of multiple lexicons

Inclusion of practice questions of GRE and other tests

We do plan to work on them in the future and publish it to App Store

