Senior Project Final Presentation Fall 2017

VIRS 2.0

Team Members:

Alfredo Lopez, Milad Ebrahimi

Product Owners:

Eric Dwyer, Seyedjafar Ehsanzadehsorati

Professor:

Masoud Sadjadi

School of Computing and Information Sciences
Florida International University

Problem definition

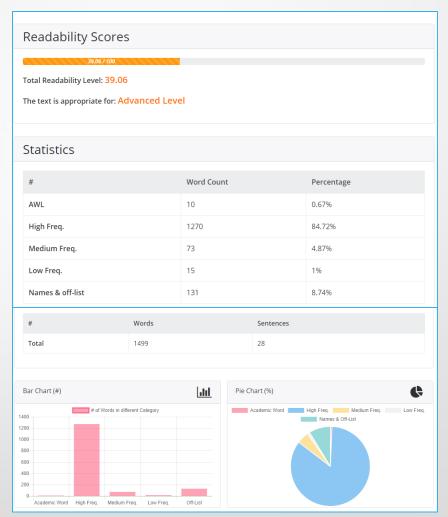
Problem:

One of the major challenges that English Language learners (ELL) as well as mainstream students face is the lack of a reliable source to improve their academic words. There is no easy way to validate the easiness of text, which allows professors to select the appropriate materials for class. A challenging text is a wonderful way to propel students forward, yet something too hard to read can cause the opposite effect.

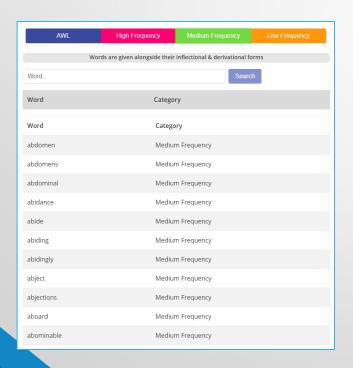
 An application that produces enhanced texts through a colorcoded system.

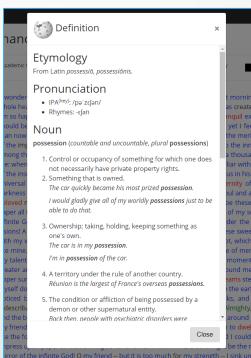


 Provides reliable readability index that would help teachers in decisions making gives a detailed analysis of texts that would help in material development.



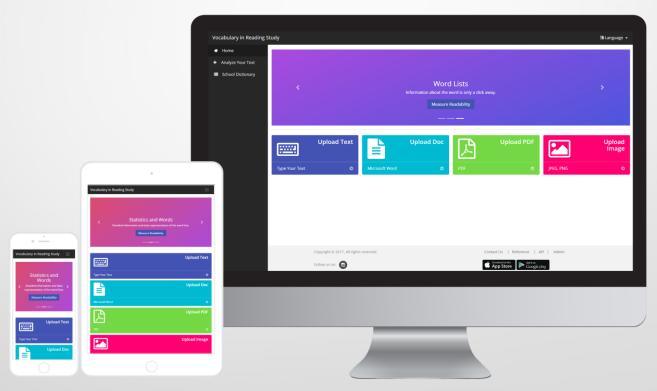
 The Wikipedia definition for the word will be displayed when clicking a word.



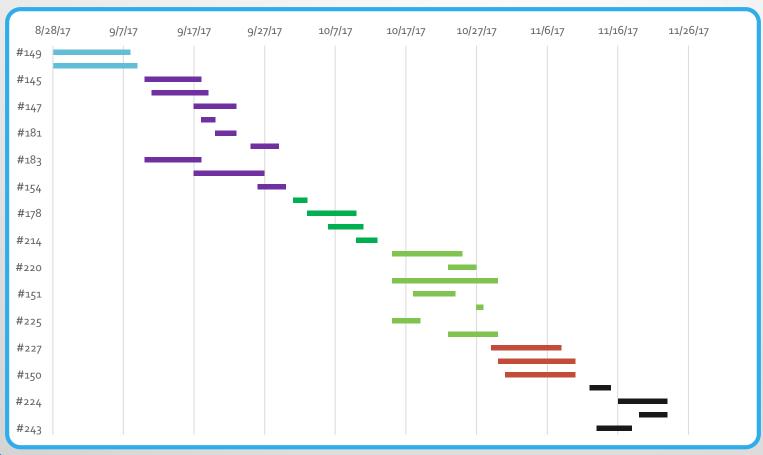


 A new frequency-based dictionary that makes it easier to read.

 The web application that is accessible from desktops, tablets and mobile devices.



Project Management



Requirements: User Stories

- 132 [Backend] Perform OCR
- 145 [Backend] Analyze Text
- 146 [Backend] Analyze PDF
- 147 [Backend] Analyze Doc
- 148 Implement Responsive Design
- 149 Merge Applications
- 150 [Frontend] Create Admin Panel
- 151 [Backend] Implement Word Definition

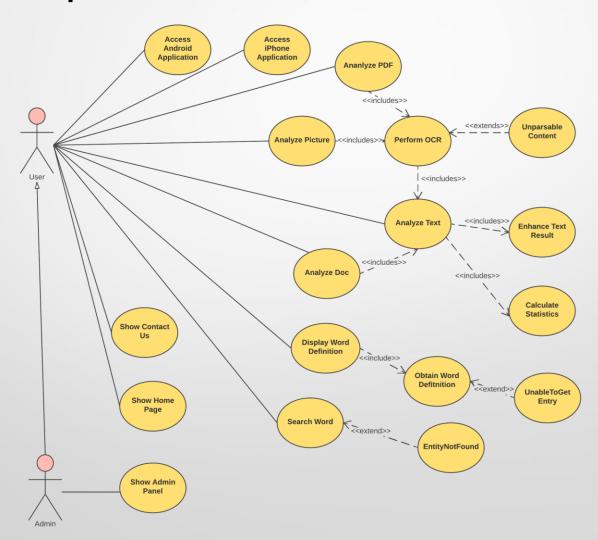
- 154 [Frontend] Create Sidebar Menu
- 155 [Frontend] Apply Category Color to Words
- 157 Implement Google Analytics
- 158 [Backend] Create Admin Login
- 176 [Frontend] Analyze Text
- 178 [Frontend] Analyze PDF
- 179 [Frontend] Analyze Doc

Requirements: User Stories

- 181 [Backend] Manage words list
- 182 Create cloud application
- 183 [Frontend] Create page for enhanced
- 184 [Frontend] Create page for statistics
- 185 [Frontend] Create instructions
- 214 [Frontend] Analyze Image
- 220 [Frontend] Create Word Lists page

- 222 [Frontend] Implement Word definition
- 223 [Frontend] Add Credits Page
- 224 [Frontend] Add Contact Us Page
- 225 [Backend] Calculate Statistics
- 227 [iPhone] Create iPhone Application
- 228 [Android] Create Android Application
- 243 Create Search API

Requirements: Use Cases



User Story: #132 Perform OCR

As a user, I want to see the words and their different categories from an image file so that I can know more about the words in the image.

Deliverance

Moonlight, clear as water, I pace the ground under a tree In deep, deep thought. Deep in thought, I pick up a fallen twig To tap, with a sigh, my own shadow On the moonlit ground.

Life-

Everybody treats it as a dream, A blurred dream. My friend, As you try to find clear lines in the blurred world. Your life's suffering Thus begins!

You may treasure life's snow-white robe, Yet life has to cross

The immense sea of darkness.

My friend,

The world does not abandon you, Why should you abandon the world?

Let life stand alone and noble like a stork,

Free as a cloud,

And pure and calm as water, Even if life were a dream,

Let it be a clear dream.

Deliverance

Moonlight, clear as water, I pace the ground under a tree In deep, deep thought. Deep in thought, I pick up a fallen twig To tap, with a sigh, my own shadow On the moonlit ground.

Life-

Everybody treats it as a dream, A blurred dream. My friend, As you try to find clear lines in the blurred world, Your life's suffering Thus begins!

You may treasure life's snow-white robe, Yet life has to cross

The immense sea of darkness.

My friend,

The world does not abandon you, Why should you abandon the world?

Let life stand alone and noble like a stork, Free as a cloud.

And pure and calm as water,

Even if life were a dream, Let it be a clear dream.

Deliverance

Moonlight, clear as water,

I pace the ground under a tree In deep, deep thought. Deep in thought, I pick up a fallen twig To tap, with a sigh, my own shadow On the moonlit ground.

Everybody treats it as a dream, A blurred dream.

My friend,

As you try to find clear lines in the blurred world, Your life's suffering Thus begins!

You may treasure life's snow-white robe,

Yet life has to cross

The immense sea of darkness.

My friend.

The world does not abandon you, Why should you abandon the world?

Let life stand alone and noble like a stork,

Free as a cloud.

And pure and calm as water, Even if life were a dream,

Let it be a clear dream.

Use Case: #132 Perform OCR

Name: Perform OCR

Actor: User

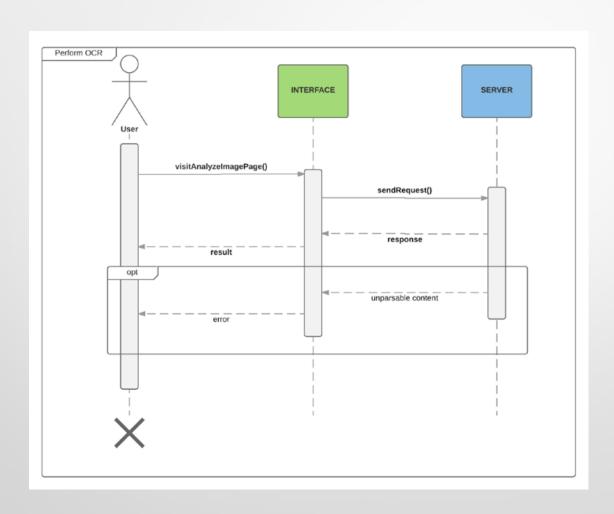
Preconditions: The user has the desire to know more information

about words in a picture.

Description <Flow of events>:

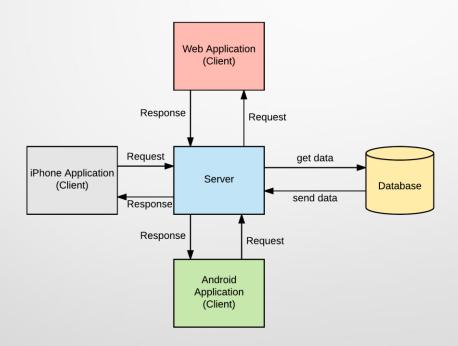
- -The user interacts with systems and chooses a picture
- -The user sends the information
 - -The system receives the information and process it
- -The system displays the information.
- -The use case ends
- -Alternative: The user receives an unparsable exception.

Requirements: Sequence Diagrams



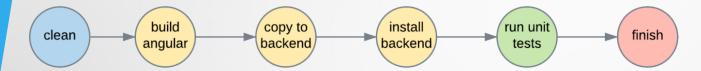
System Design: Architecture

Model-View-Controller

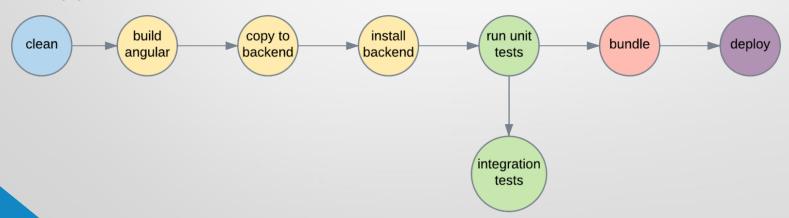


System Design: Deployment

Dev pipeline

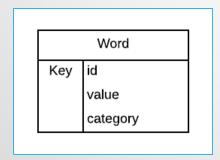


Prod pipeline



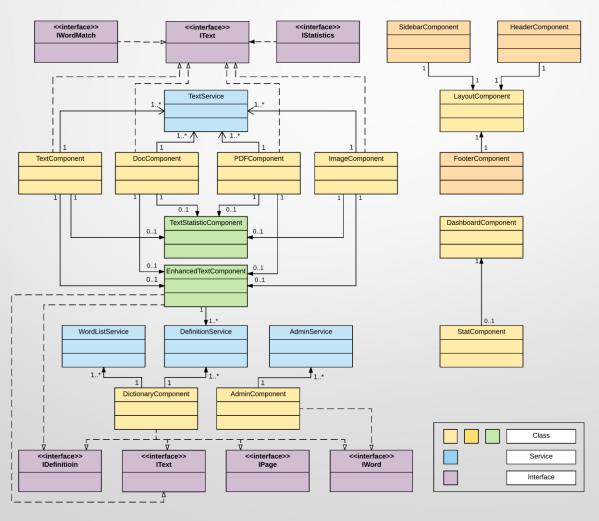
System Design

- Persistent data design
 - The data layer consists in a single table with all the data.

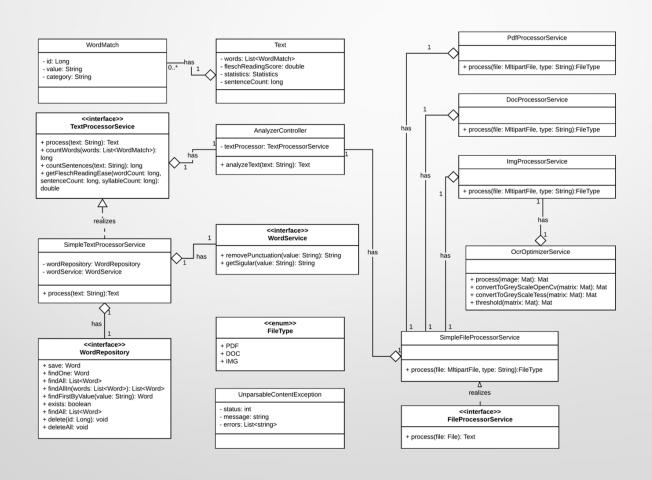


- Security/Privacy
 - All keys are encrypted using Jasypt.
 - The Administrator keys are all secured.

Minimal Class Diagram: Frontend



Minimal Class Diagram: Backend



Main algorithm

```
List<Word> words;
Map<String, Word> resultMap;
FOR EACH word IN text.words
  Word cleanWord = clean(word);
  IF (resultMap.contains(lemma)) THEN
    words.add(word)
  ELSE
    word found = wordRepository.find(word);
    resultMap.put(word, found);
    words.add(found)
  END IF
NEXT
```

Test Suites and Test Cases

- Junit was the main tool utilized to verify the system.
- Implemented several integration tests that are run before deploying to the web.
- Karma and Jasmine were used to perform unit testing for frontend.
- Compodoc and Istanbul were used to generate a code coverage report.

Test Suites and Test Cases

- Test case ID: getFleschReadingEase_WithDataFromWebSite_Returns6o_2o
- Description/Summary of Test: Getting the Flesch Reading Score with word count of 300, a sentence count of 12 and a syllable count of 430 returns 60.20.
- Pre-condition: None
- Expected Results: A reading score of 60.20
- Actual Result: A Reading score of 60.20
- Status (Fail/Pass): Pass

Test Suites and Test Cases

- Test case ID: TextComponent_ShouldCreate_oo1
- Description/Summary of Test: The Text component should be created.
- Pre-condition: The server must be running.
- Expected Results: Test Component should be created
- Actual Result: Test Component was created
- Status (Fail/Pass): Pass

Summary

Summary:

VIRS is a web app which facilitates learning of new languages through analysis of text and gathering of data which is then displayed to the user. Data given to the user includes category of word as far as popularity and definition of the words in the text with respect to the information in the database. It also provides the user with a complete statistical text analysis report and readability score.



- Alfredo Lopez: alfredo.lopezoo2@gmail.com
- Milad Ebrahimi: milad19@gmail.com
- Questions?
- Thank You!



































