*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

Project Title: Vocabulary in Reading Study

VIRS 5.0

**Team Members:** Julian Pineiro, Jorge Costa

**Product Owner(s)**: Eric Dweyer

**Mentor(s)**: Seyedjafar Ehsanzadehsorati

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*This document presents the information necessary to gain a good understanding of Vocabulary in Reading Study (VIRS 5.0). 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. This data is important to the user as it categorizes words in the text and gives priority to more popular words. The user then knows which words to use more and makes it easier to use these in conversation. The user can test their knowledge of the Essential Word List, a user can delete their account, update their account information, get a better reading from images uploaded, reset their password via email or username. The user can now also upload images of text which can be interpreted with high accuracy and administrators can make detailed edits to the data bank of words.*

**Table of Contents**

**Introduction** …………………………………………………………………………………………….....6

Current System ……….…………………………………………………………………………………………………..6

Purpose of New System …………………………………………………..……………………………. 6

**User Stories**

Implemented User Stories ………………………………………….…………………………………..8

Pending User Stories …………………………………………………………………...………………………………..….. 20

**Project Plan**

Hardware and Software Resources ……………………………………………………………….… 21

Sprints Plan ……………………………………………………………..………………………………………………………. 24

*Sprint 1*  …………………………………………………………………………………………………………………………... 24

*Sprint 2*  …………………………………………………………………………………………………………………………... 25

*Sprint 3*  …………………………………………………………………………………………………………………………... 26

*Sprint 4*  …………………………………………………………………………………………………………………………... 27

*Sprint 5*  …………………………………………………………………………………………………………………………... 29

*Sprint 6*  …………………………………………………………………………………………………………………………... 30

*Sprint 7*  …………………………………………………………………………………………………………………………... 31

**System Design**

Architectural Patterns ………………………………………………………………………………………………….. 33

System and Subsystem Decomposition …………………………………………………………………………….… 36

Deployment Diagram …………………………………………………………………………………………………….…... 38

Design Patterns ………………………………………………………………………………………………………….….... 39

**System Validation**  …………………………………………………………………………………………………………………….40

**Glossary**  ………………………………………………………………………………………………………………………………….47

**Appendix**  ………………………………………………………………………………………………………………………………….48

Appendix A - UML Diagrams ………………………………………………………………………………………………. 48

*Static UML Diagrams*  ……………………………………………………………………………………………………….48

*Dynamic UML Diagrams*  …………………………………………………………………………………………………..52

Appendix B - User Interface Design …………………………………………………………………………….…... 54

Appendix C - Sprint Review Reports …………………………………………………………………………...…… 59

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents …………………………………………………………………….…………… 71

**References** ……………………………………………………………………………………………………………………...………...75

# Introduction

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. Vocabulary in Reading Study(VIRS) is the solution to all these problems.

## Current System

This is the fifth iteration of the application. The current system consists of using MySQL, Spring MVC and Angular hosted in Amazon web servers. The system can currently analyze a text, word document, pdf and even images to give a readability score. The web application currently also displays a dictionary that is clickable for definitions. The application includes administrative tools for the product owners to add and remove words from the dictionary. The application is currently hosted on AWS using elastic beanstalk. The application has a cohesive theme that is present throughout the site.

## 

*Figure 1-VIRS 5.0 dashboard*

## Purpose of New System

The imperative task of the new system is to add features to make this a well rounded website, that achieves the purpose of helping those seeking to learn English as their new language. The updates must maintain the theme currently in the system and be in the vein of the purpose of this project.

Features must be accessible on both desktop and mobile platform, the features must enrich the website so as to ensure user satisfaction with not only the main feature of text analysis, but also the new features added in this version.

The features must be created to maintain the current system, while adding unto it. Therefore, maintaining the current backend API. The main features to be included are a newly improved optical character recognition system powered by Amazon’s Textract, which can now identify scanned images with near 100% accuracy, a new ‘K3’ word category, as well as improved administrative control over the database of words in the administrative panel.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the VIRS project. These user stories served as the basis for the implementation of the project’s features. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

* + **User Story 01** - install frontend dependencies and run the front end locally
    - **Description** - As a developer, I would like to install the frontend’s dependencies, so that the application can run.
    - **Acceptance Criteria**: 1. Front end dependencies are installed locally
  + **User Story 02** - investigate frontend structure
    - **Description** - As a developer, I would like to investigate and understand frontend program structure in order to facilitate the setup of the application and build on top of it in later sprints
    - **Acceptance Criteria**: 1. Developer is familiar with structure of program’s front end
  + **User Story 03** - Install the Dependencies/Research the backend of the application
    - **Description** - As a developer, I would like to install the backend dependencies and fully understand how the backend works, fixing any errors we might encounter.
    - **Acceptance Criteria**: 1. All backend dependencies are installed locally
  + **User Story 04** - Preliminary hosting research
    - **Description** - As a developer I would like to perform preliminary research on AWS hosting
    - **Acceptance Criteria**: 1. Developer is familiar with ins and outs of AWS project hosting
  + **User Story 05** - Create IAM users
    - **Description** - As a developer I would like to create IAM users on the AWS account to access the AWS service without the root user
    - Acceptance Criteria: AWS account has two IAM users with admin permissions
  + **User Story 06** - Organize meeting dates.
    - **Description** - As a developer I would like to organize meeting dates for the rest of the semester.
    - **Acceptance Criteria**: 1. Meeting dates are posted on calendar
  + **User Story 07** - Run the application with minimal features on AWS
    - **Description** - As a Product owner I would like to see some version of the application running on AWS.
    - **Acceptance Criteria**: 1. Application compiles locally 2. Application can communicate with local database 3. Basic feature (text analysis, dictionary, EWL, translation) of application are functional
  + **User Story 08** - Improve the features that are run locally while making sure integration into AWS works.
    - **Description** - As a Product owner I would like more features to be functionable when using the application
    - **Acceptance Criteria**: 1. All features of VIRS work when running the program locally
  + **User Story 09** - Get encryption working with API secrets locally and AWS
    - **Description** - As a product owner I would like to keep API keys secret when running the program.
    - **Acceptance Criteria**: 1. Existing encryption module compiles 2. Encryption module can decrypt encrypted API keys 3. Decrypted API keys are functional
  + **User Story 10** - Re-evaluate AWS billing
    - **Description** - As the product owner I would like to stop recurring payments to AWS for services that are not currently necessary
    - **Acceptance Criteria**: 1. AWS is not billing for unused services
  + **User Story 11** - CI/CD research
    - **Description** - As a developer I would like to research CI/CD to facilitate efficient development.
    - **Acceptance Criteria**: 1. The developer is familiar with the deployment requirements of AWS
  + **User Story 12** - Resolve database issues on AWS
    - **Description** - As a developer I would like to have a functioning database to run the application.
    - **Acceptance Criteria**: 1. Database is up and running on RDS
  + **User Story 13** - Resolve port/communication errors between instances
    - **Description** - As a developer I would like to resolve the communication errors between the instance running the application and the database
    - **Acceptance Criteria**: 1. Application hosted on EC2 instance is able to communicate with database hosted on RDS
  + **User Story 14** - Solve bundling issues with Elastic Beanstalk
    - **Description** - As a developer I would like to have a functioning database to run the application.
    - **Acceptance Criteria**: 1. Database is integrated into elastic beanstalk environment
  + **User Story 15** - Conduct research on options for optical character recognition
    - **Description** - As a developer I would like to research my options for implementing OCR
    - **Acceptance Criteria**: 1. Developer is familiar with multiple options for OCR implementation
  + **User Story 16** - Conduct research on options for CSV upload to DB
    - **Description** - As a developer I would like to research my options for database upload via csv file
    - **Acceptance Criteria**: 1. Developer is familiar with existing frontend code which relates to database upload
  + **User Story 17** - Implement optical character recognition with minimal inaccuracies
    - **Description** - As a user I would like to be able to upload an image of text and have the application interpret the image as text
    - **Acceptance Criteria**: 1. OCR is more accurate in low light 2. OCR is more accurate when reading skewed documents 3. OCR is more accurate when reading blurred scans 4. OCR accurately reads non-standard characters
    - **Use Case**
      * **Name**: Implement optical character recognition with minimal inaccuracies
      * **ID**: U17
      * **Actor**: Web User
      * **Preconditions**:
        + 1. System must be up and running
        + 2. The user must be on the dashboard
      * **Description**:
        + 1. Web user clicks on the ‘Upload Image’ Panel on the dashboard
        + 2. Site displays the image upload page
        + 3. Web user clicks the ‘Browse..’ button to browse for an image of text to upload
        + 4. Message displayed confirming file upload
        + 5. User clicks the enhanced text button

5.1. If file type is not an image, error message displays, return to step 3

5.2 If there is no file uploaded, display an error message, return to step 3

* + - * + 6. Application navigates to enhanced text page, displaying text from the image as plain text, highlighted by word category
      * **Postconditions**:
        + 1. The text displayed must be at least 90% accurate to the images uploaded
  + **User Story 18** - Add header to dictionary page
    - **Description** - As a product owner I would like the dictionary page to have an additional header
    - **Acceptance Criteria**: 1. Header that says ‘A Corpus Based K-12 Dictionary’ above word list on dictionary page
    - **Use Case**
      * **Name**: Add header to dictionary page
      * **ID**: U18
      * **Actor**: Web user
      * **Preconditions**:
        + 1. The system is up and running
        + 2. The user has navigated to the dashboard
      * **Description**:
        + 1. User navigates to the dictionary page
        + 2. A header that reads ‘A Corpus-based K-12 Dictionary’ is displayed
      * **Postconditions**:
  + **User Story 19** - Make adjustments to communications between application and database
    - **Description** - As a developer I would like to secure communication between the database and the application
    - **Acceptance Criteria**: 1. Database cannot be accessed by sources that are not the application
    - **Use Case**
      * **Name**: Make adjustments to communications between application and database
      * **ID**: U19
      * **Actor**: User
      * **Preconditions**:
        + 1. The system is up and running
      * **Description**:
        + 1. The user attempts to directly access the word list contained in the database
        + 2. A server error code is displayed
  + **User Story 20** - Conduct research into RDS pricing
    - **Description** - As a product owner I would like to lower the cost of hosting the server on AWS
    - **Acceptance Criteria**: 1. Costs due hosting database on AWS are lowered
  + **User Story 21** - Create a Github repository for the project and compile changes made
    - **Description** - As a developer I would like to manage changes to the project and compiling the previous changes made for the project
    - **Acceptance Criteria**: 1. Github repository is created 2. Repository is populated with project files
  + **User Story 22** - Implement bulk word upload to the DB by admin via CSV
    - **Description** - As a product owner I would like to upload words to the database in bulk
    - **Acceptance Criteria**: 1. Application accepts csv files uploaded on admin panel 2. Words contained in uploaded csv files are added to the database
      * **Use Case**
        + **Name**: Implement bulk word upload to the DB by admin via CSV
        + **ID**: U22
        + **Actor**: Admin User
        + **Preconditions**:

1. The system must be up and running

2. The user has navigated to the admin panel

3. The user has logged in as an admin

4. The user has a list of words organized as a column in a csv

* + - * + **Description**:

1. The user clicks the ‘Browse..’ button and uploads a csv file to the application

2. The user selects a category via dropdown to upload the words contained in the csv to

3. The user presses the upload button

3.1. If no file was uploaded, display an error message

3.2 If the file is not of the type csv, display an error message

4. The words from the csv are uploaded to the database under the selected category

* + - * + **Postconditions**:

1. The database contains the words from the csv, associated with their respective categories

* + **User Story 23** - Update K1 and K2 word lists
    - **Description** - As a product owner I would like to make some changes to the existing K1 and K2 word lists to resolve inconsistencies
    - **Acceptance Criteria**: 1. K1 and K2 lists are updated
    - **Use Case**
      * **Name**: Update K1 and K2 word lists
      * **ID**: U23
      * **Actor**: User
      * **Preconditions**:
        + 1. The system must be up and running
      * **Description**:
        + 1. The user navigates to the essential words list
        + 2. The words in the K1 and K2 categories display
      * **Postconditions**:
        + 1. Only the words that belong in their respective categories are displayed in the K1 and K2 lists
  + **User Story 24** - Add another category “K3” list to the project
    - **Description** - As a product owner I would like to add a new category of words for the project
    - **Acceptance Criteria**: 1. Database accepts new words with K3 category 2. K3 category displays K3 words on dictionary, EWL and test analysis pages
    - **Use Case**
      * **Name**: Add another category “K3” list to the project
      * **ID**: U24
      * **Actor**: User
      * **Preconditions**:
        + 1. The system must be up and running
      * **Description**:
        + 1. The user navigates to the essential words list
        + 2. The EWL page is displayed by the application
        + 3. The user clicks the ‘K3’ category button
        + 4. The K3 category list is displayed
    - **Postconditions**:
      * + 1. Only the words that belong in the K3 category are displayed in the K3 list
  + **User Story 25** - Get previous Tests working on the project
    - **Description** - As a developer I want to make sure previous tests work with the production build.
    - **Acceptance Criteria**: 1. All existing tests pass
    - **Use Case**
      * **Name**: Get previous Tests working on the project
      * **ID**: U25
      * **Actor**: Product owner/Developer
      * **Preconditions**:
        + 1. The user must navigate to the ‘code’ folder in the VIRS project file
      * **Description**:
        + 1. The user attempting to install the software runs ‘virs.cmd install’
        + 2. The application runs all tests and installs itself
      * **Postconditions**:
        + 1. The installer shows all tests as passed
  + **User Story 26** - Establish a global limit for how many documents can be scanned by textract
    - **Description** - As a product owner I would like to limit the amount of documents that go through textract in order to limit costs
    - **Acceptance Criteria**: 1. Application returns an error message when over 1000 images have been scanned in one month 2. The scan count resets after each month
    - **Use Case**
      * **Name**: Establish a global limit for how many documents can be scanned by textract
      * **ID**: U26
      * **Actor**: User
      * **Preconditions**:
        + 1. The system must be up and running
        + 2. The document scan counter must be greater than 1000
      * **Description**:
        + 1. The user navigates to the image upload page of the application
        + 2. The user presses the ‘Browse…’ button and uploads an image
        + 3. The user presses the enhanced text button
        + 4. The user receives an error message indicating that the monthly limit for image recognition has been exceeded
      * **Postconditions**:
        + 1. The counter remains above the limit until the next period
  + **User Story 27** - Conduct preliminary research on the best options for hosting a backup server
    - **Description** - As as a developer I would like to better understand my options for hosting a backup so that I can decide on the best implementation
    - **Acceptance Criteria**: 1. Developer is familiar with multiple options for backup hosting
  + **User Story 28** - Back up the database locally
    - **Description** - As a product owner I would like to backup my database locally in case of a data loss event
    - **Acceptance Criteria**: 1. Database is exported to an sql file 2. Database can be replicated with sql file
  + **User Story 29** - Make adjustments to word category coloring
    - **Description** - As a product owner I would like to alter the coloring that certain word categories are identified by
    - **Acceptance Criteria**: 1. Category coloring matches product owner’s specifications
  + **User Story 30** - Upgrade administrator Database editing tools
    - **Description** - As a product owner/administrator I would like to be have more options for uploading to/editing the database
    - **Acceptance Criteria**: 1. Category selected for ‘upload and replace’ are deleted 2. Category selected for ‘upload and replace’ is filled with words from uploaded csv file
    - **Use Case**
      * **Name**: Implement bulk category replacement via CSV
      * **ID**: U30
      * **Actor**: Admin User
      * **Preconditions**:
        + 1. The system must be up and running
        + 2. The user has navigated to the admin panel
        + 3. The user has logged in as an admin
        + 4. The user has a list of words organized as a column in a csv
      * **Description**:
        + 1. The user clicks the ‘Browse..’ button and uploads a csv file to the application
        + 2. The user selects a category via dropdown to upload the words contained in the csv to
        + 3. The user presses the upload and replace button

3.1. If no file was uploaded, display an error message

3.2 If the file is not of the type csv, display an error message

* + - * + 4. The words from the csv are uploaded to the database under the selected category
      * **Postconditions**:
        + 1. The words that were previously in that category are wiped from the database
        + 2. The database contains the words from the csv, associated with their respective categories
  + **User Story 31** - Upgrade admin DB editing tool interface
    - **Description** - As a product owner/administrator I would like the DB editing interface to be more usable
    - **Acceptance Criteria**: 1. Drop down menu allows for selection of categories 2. Buttons for ‘csv upload’ and ‘upload and replace’ are functional
  + **User Story 32** - Conduct research into EC2 cost cutting measures
    - **Description** - As a product owner I would like to pay less for the EC2 instance the application is running on
    - **Acceptance Criteria**: 1. Monthly AWS charge due to EC2 instance is lower
  + **User Story 33** - Merging branches and conflicts
    - **Description** - As a developer I would like to merge all branches of the project so all features can be integrated
    - **Acceptance Criteria**: 1. All branches merged into one production product
  + **User Story 34** - Code refactoring image recognition
    - **Description** - As a developer I would like to refactor the code in the image and text processors to make it more readable
    - **Acceptance Criteria**: 1. Code retains functionality 2. Code does not contain unnecessary lines or comments
  + **User Story 35** - Fix admin search function
    - **Description** - As a product owner I would like the search function on the admin panel to consistently return words searched for
    - **Acceptance Criteria**: 1. Admin search returns the word typed into the search bar if the word is in the database 2. If the word is not in the database, return a message indicating so
  + **User Story 36** - Fix database category update function
    - **Description** - As a product owner I would like the replace function to replace the intended category without affecting other categories
    - **Acceptance Criteria**: 1. Database does not delete words from categories other than the intended category when replacing
  + **User Story 37** - Make changes to dictionary page
    - **Description** - As a product owner I would like to update the dictionary page in order to make it more readable
    - **Acceptance Criteria**: 1. Header moved above categories 2. Header is made larger 3. Header has an appropriate font
  + **User Story 38** - Assess necessary aesthetic changes to each page on the site
    - **Description** - As a developer I would like to assess what changes need to be made to the frontend of each page on the site
    - **Acceptance Criteria**: 1. Proposed changes are accepted by VIRS product owner

## Pending User Stories

* + **User Story** - Host a backup of the site
    - **Description**: As a developer I would like to set up a backup host to be used in the event of an aws outage
    - **Acceptance Criteria**: 1. A backup of the site is hosted somewhere outside AWS 2. The backup is ready to be deployed at any time in the case of an AWS outage

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

***Hardware:***

**● Computer running Linux, Mac OS or windows.**

**● For the deployed application make sure you stay within the AWS server constraints**

**○ 64bit Amazon Linux 2017.03 v2.5.5 running Java 8**

***Software:***

**The following list is the software used in the application. Note that it is quite extensive and includes all the development layers of the stack.**

**Front end:**

**●** Angular 4.3.1

● Ng-Bootstrap 1.0.0

● Ng-Translate 7.0.0

● Chat.js 2.7.1

● Font-Awesome 4.7.0

● Ng2-Charts 1.6.0

● Rxjs 5.1.0

● Jasmine 2.5.45

● Karma 1.7

● Typescript 2.3.3

**Back end:**

**● Maven 3.5**

**○** Commons-lang3 3.4

○ HikariCP 1.5.6.RELEASE

○ Jai-imageio-core 1.3.1

○ Jai-imageio-jpeg2000 1.3.0

○ Jasypt-spring-boot-starter 1.16

○ Levigo-jbig2-imageio 2.0

○ Mysql-connector-java 1.5.6.RELEASE

○ Opencsv 3.3

○ Opencv 3.2.0-1

○ Spring-boot-starter-data-jpa 1.5.6.RELEASE

○ Spring-boot-starter-security 1.5.6.RELEASE

○ Spring-boot-starter-test 1.5.6.RELEASE

○ Spring-boot-starter-web 1.5.6.RELEASE

○ Springfox-swagger-ui 2.7.0

○ Springfox-swagger2 2.7.0

○ Sqlite-jdbc 1.5.6.RELEASE

○ Tess4j 3.4.1

○ Thucydides-core 0.9.275

○ Tika-parsers 1.16

● Tesseract 3.05.01

● Leptonica 1.74.4

● Mysql 14.14

● Java 1.8.0

● Spring Developer Suite 3.9

**Android:**

● Android Studio 3.0

● Gradle 3.0

● Android sdk 23

**iOS:**

**●**  Xcode 9.1

● Command line tools 9.2

**Other:**

**●**  Git 2.14.1

● Bash 3.2.57

## 

## 

## Sprints Plan

### Sprint 1

**SP20 Team26 Planning 1/13/2020 4:45pm 5:30pm**

* Julian Pineiro, Jorge Costa, Eric Dwyer
* Sprint Goals:
  + Host the previous version of the project on AWS.
  + Fully establish and refine project goals.
  + Conduct preliminary research

·       **Sprint User Stories:**

* User Story 1 -  Create IAM users
  + As a developer I would like to access the AWS service via an IAM role
  + Initially assigned to Jorge.
* User Story 2 - Re-evaluate AWS billing
  + As the product owner I would like to stop recurring payments to AWS for services that are not currently necessary
  + Initially assigned to Jorge.
* User Story 3 - Preliminary hosting research
  + As a developer I would like to perform preliminary research on AWS hosting
  + Initially assigned to Jorge and Julian
* User Story 4 - Story tracking
  + As a developer I would like to set up a system to track user stories to facilitate efficient development
  + Initially assigned to Julian.
* User Story 5 - CI/CD research
  + As a developer I would like to research CI/CD to facilitate efficient development.
  + Initially assigned to Julian.
* User Story 6 - Organize meeting dates.
  + As a developer I would like to organize meeting dates for the rest of the semester.
  + Initially assigned to Julian.
* User Story 7 - Set up billing alerts
  + As a developer I would like to set up billing alerts on AWS to avoid excessive billing.
  + Initially assigned to Jorge.

·         **Summary** - The general goal of this sprint for the time being will be to set up and host the previous iteration of the project and work with the product owner to figure out the broader project goals.

### Sprint 2

SP20 Team26 Planning 1/28/2020 4pm 4:20pm

* Jorge Costa, Julian Pinerio, Ehsan Seyedjafari
* Sprint Goals:
* Get the project running on AWS
* Incorporate the old features back into the locally ran application
* Sprint User Stories:
  + User Story 1 - Run the application with minimal features on AWS
    - As a Product owner I would like to see some version of the application running on AWS.
    - Initially assigned to both Jorge and Julian
    - Large Size - Very important Priority - 8 points
  + User Story 2 - Improve the features that are run locally while making sure integration into AWS works.
    - As a Product owner I would like more features to be functionable when using the application
    - Initially assigned to both Jorge and Julian
    - Large Size - High priority - 8 points
  + User Story 3 - Get encryption working with API secrets locally and AWS
    - As a product owner I would like to keep API keys secret when running the program.
    - Initially assigned to Julian
    - Medium Size - Medium priority - 4 points
  + User Story 4 - Re-evaluate AWS billing
    - As the product owner I would like to stop recurring payments to AWS for services that are not currently necessary
    - Initially assigned to Jorge.
    - Medium Size - Low/Medium Priority - 4 points
  + User Story 5 - CI/CD research
    - As a developer I would like to research CI/CD to facilitate efficient development.
    - Initially assigned to Julian.
    - Small Size - Low Priority - 2 points
  + User Story 6 - Set up billing alerts
    - As a developer I would like to set up billing alerts on AWS to avoid excessive billing.
    - Initially assigned to Jorge.
    - Small Size - Low Priority - 2 points
* Summary - Getting the project up and running on AWS is top priority. This is what we plan to focus on and then incorporate the old features back into the application as we solve their problems. This way we can have some deliverable progress by the end of the sprint regardless of the errors that come up.

### Sprint 3

SP20 Team26 Planning 2/11/2020 5:00pm 5:30pm

* Jorge Costa, Julian Pineiro, Eshan Seyedjafari
* Sprint Goals:
  + Get running/improve optical character recognition
  + Do preliminary work on setting up bulk word uploading to the database via csv
  + Make other minor site changes
* Sprints:
  + Conduct research on options for optical character recognition
    - As a developer I would like to research my options for implementing OCR
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Conduct research on options for CSV upload to DB
    - As a developer I would like to research my options for database upload via csv file
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Implement optical character recognition with minimal inaccuracies
    - As a user I would like to be able to upload an image of text and have the application interpret the image as text
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + Add header to dictionary page
    - As a product owner I would like the dictionary page to have an additional header
    - Initially assigned to Julian
    - Small size - 2 points
  + Make adjustments to word category coloring
    - As a product owner I would like to alter the coloring that certain word categories are identified by
    - Initially assigned to Jorge
    - Small size - 2 points

### Sprint 4

SP20 Team26 Planning 3/2/2020 2:45pm 3:05pm

* Jorge Costa, Julian Pineiro, Eshan Seyedjafari
* Sprint Goals:
  + Establish a global limit on the number of documents that can be scanned with OCR
  + Set up bulk word uploading to the database via csv
  + Update K1 and K2 lists and add a K3 list
* Sprint User Stories:
  + Implement bulk word upload to the DB by admin via CSV
    - As a product owner I would like to upload words to the database in bulk
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + Make adjustments to word category coloring
    - As a product owner I would like to alter the coloring that certain word categories are identified by
    - Initially assigned to Jorge
    - Small size - 2 points
  + Establish a global limit for how many documents can be scanned by textract
    - As a product owner I would like to limit the amount of documents that go through textract in order to limit costs
    - Initially assigned to Jorge
    - Medium size - 4 points
  + Update K1 and K2 word lists
    - As a product owner I would like to make some changes to the existing K1 and K2 word lists to resolve inconsistencies
    - Initially assigned to Jorge
    - Medium size - 4 points
  + Add another category “K3” list to the project
    - As a product owner I would like to add a new category of words for the project
    - Initially assigned to Julian
    - Medium size - 4 points
  + Get previous Tests working on the project
    - As a developer I want to make sure previous tests work with the production build.
    - Initially assigned to Julian
    - Medium Size - 4 points

### Sprint 5

SP20 Team26 Planning 3/17/2020 2:45pm 3:15pm

* Jorge Costa, Julian Pineiro, Eshan Seyedjafari
* Sprint Goals:
  + Establish a backup host for the application so the site can be hosted in the event of an AWS outage
  + Backup the database and application locally
  + Make other minor site changes
* Sprints:
  + Conduct preliminary research on the best options for hosting a backup server
    - As as a developer I would like to better understand my options for hosting a backup so that I can decide on the best implementation
    - Initially assigned to Julian
    - Medium size - 4 points
  + Back up the database locally
    - As a product owner I would like to backup my database locally in case of a data loss event
    - Initially assigned to Jorge
    - Medium size - 4 points
  + Host a backup of the site
    - As a developer I would like to set up a backup host to be used in the event of an aws outage
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + Make adjustments to word category coloring
    - As a product owner I would like to alter the coloring that certain word categories are identified by
    - Initially assigned to Jorge
    - Small size - 2 points

### Sprint 6

SP20 Team26 Planning 4/1/2020 4:45pm 5:15pm

* Jorge Costa, Julian Pineiro, Eshan Seyedjafari
* Sprint Goals:
  + Establish a backup host for the application so the site can be hosted in the event of an AWS outage
  + Make aesthetic improvements to the front end
* Sprint stories:
  + Host a backup of the site
    - As a developer I would like to set up a backup host to be used in the event of an aws outage
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + Make changes to dictionary page
    - As a product owner I would like to update the dictionary page in order to make it more readable
    - Initially assigned to Julian
    - Small size - 2 points
  + Assess necessary aesthetic changes to each page on the site
    - As a developer I would like to assess what changes need to be made to the frontend of each page on the site
    - Initially assigned to Jorge and Julian
    - Medium size - 8 points

### Sprint 7

SP20 Team26 Planning 4/15/2020 3:45pm 4:00pm

* Jorge Costa, Julian Pineiro, Eshan Seyedjafari
* Sprint Goals:
  + Compile the final deliverable for the project.
  + Complete posters, documentation, presentation slides, and videos for the project.
  + Fix any bugs present that might still be in the project from previous sprints.
* Sprint stories:
  + Fix enhanced text bugs
    - As a developer I would like to fix bugs caused by backend changes made during last sprint’s bug fixes
    - Initially assigned to Jorge and Julian
    - Size - 8 points
  + Code clean up on some files changed within the project
    - As a developer I would like the updated changes made to be readable and consistent for the next group
    - Initially assigned to Julian
    - Size - 6 points
  + Poster
    - As a developer I would like to create a poster showcasing the progress made on the application
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Presentation slides
    - As a developer I would like to create presentation slides to showcase the progress made on the application
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Videos
    - As a developer I would like to create videos using the presentation slide which describe the project
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Compiling documentation
    - As a developer I would like to update some aspects of the documentation to really explain and help avoid some of the issues we encountered this semester.
    - Initially assigned to Jorge
    - Medium size - 6 points

# 

# 

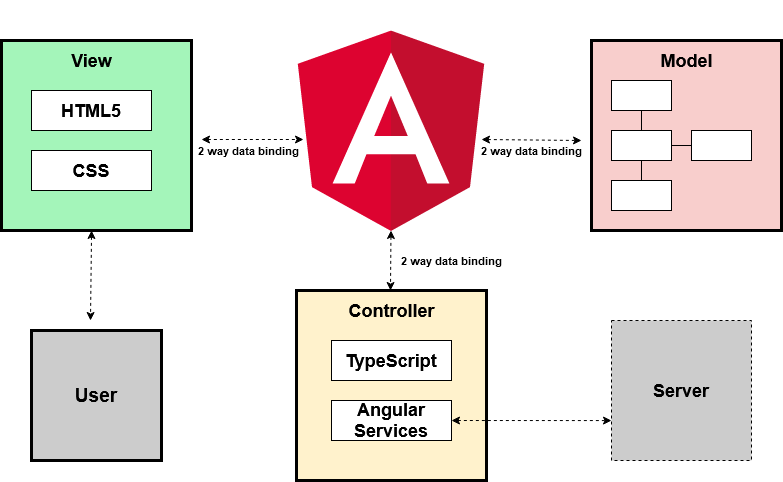
# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

## Architectural Patterns

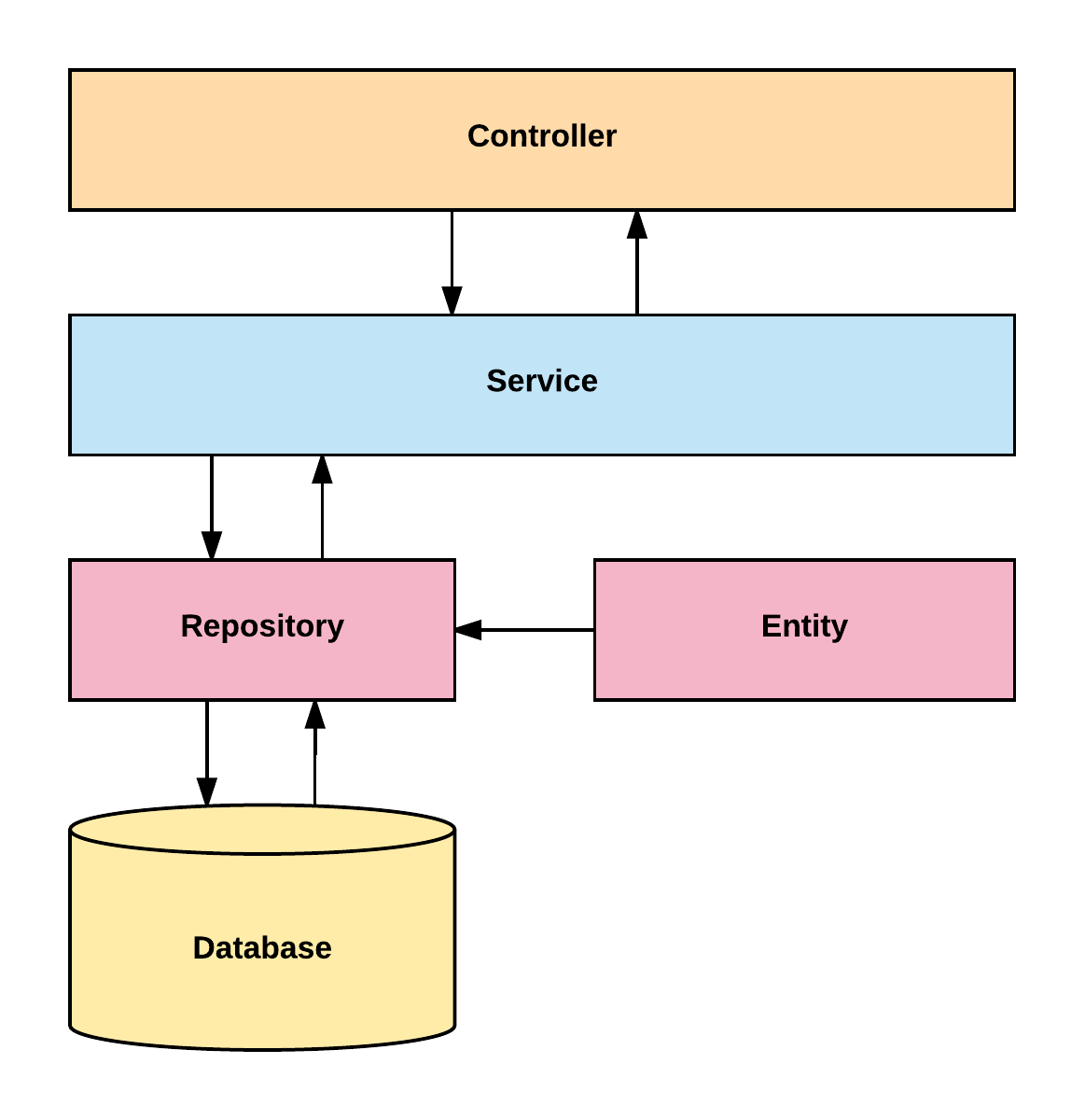
Model View Controller is the main design for the architecture. We are separating the main three parts of the application: user interaction, processing of information and storage. The segregation of these concerns favors production since each of the developers work in a separate section. This separation of components makes the system easily modifiable in the future. With this design, we ensure that we have multiple views for a controller; the system produces APIs that can be consumed by any other application.

Client-Server is used in the system to deliver the application. A highly available system that can be consumed from several parties called for this design. It allows the centralization of the code logic and database. We had to access the application from the web, iOS and Android devices so this was a very good choice

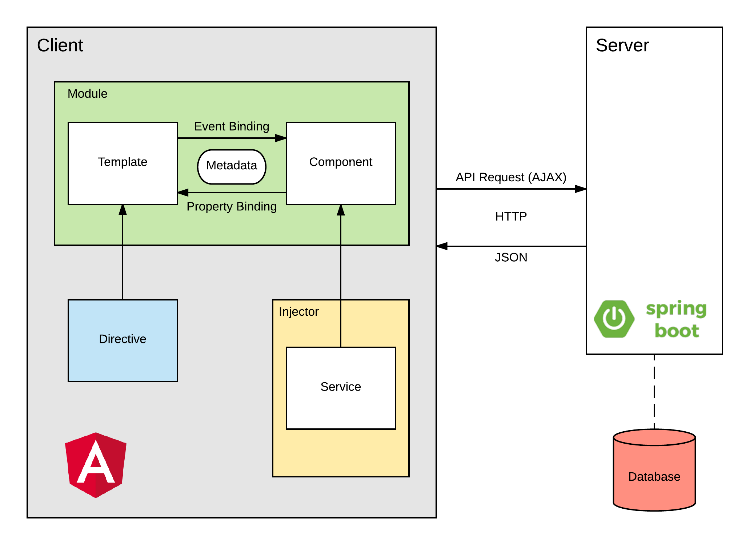


*Figure 2-Angular MVC*

Repository pattern is another design used in our system. It minimizes the amount of duplicate code in the system by abstracting the basic CRUD operations. It also ties the data entities to the domain model which favors development. The code would have to comply with the entity restrictions in order to even comply. It also helps maintain data integrity.



*Figure 3-Server System Design*



*Figure 4-Client System Design*

## System and Subsystem Decomposition

The system is made out of two major subsystems and two minor ones. They are broken down as follows according to its tasks and interactions.

Server Subsystem:

● Interacts with the database.

● Serves all the requests from the web

● Handles server side security.

● Analyzes the statistics of the text.

● Performs OCR.

● Optimizes images.

Client Subsystem:

● Displays the application.

● Routes server-side API calls.

● Contains the main boundaries for user interaction.

● Client side data validation.

Android Subsystem:

● Interacts with the web application to display in an Android device.

● Handles device data storage needed for the web.

iPhone Subsystem:

● Interacts with the web application to display it in an iOS device.

## Deployment Diagram

Deployment of the application consists in several steps run with a bash script. It starts in the front end and it propagates to the backend and eventually to the final product. We are building with ng for Angular products and Maven for the backend.

This is a detailed explanation for the development pipeline.

● Angular

○ Clean and build the application

○ Run karma tests

○ Run e2e

● Copy resources to the backend

● Spring

○ Clean and build

○ Run Unit tests

○ Create jar executable

This is a detailed explanation for the production pipeline.

● Angular

○ Clean and build the application

○ Run karma tests

○ Run e2e

● Copy resources to the backend

● Spring

○ Increase version number

○ Clean and build

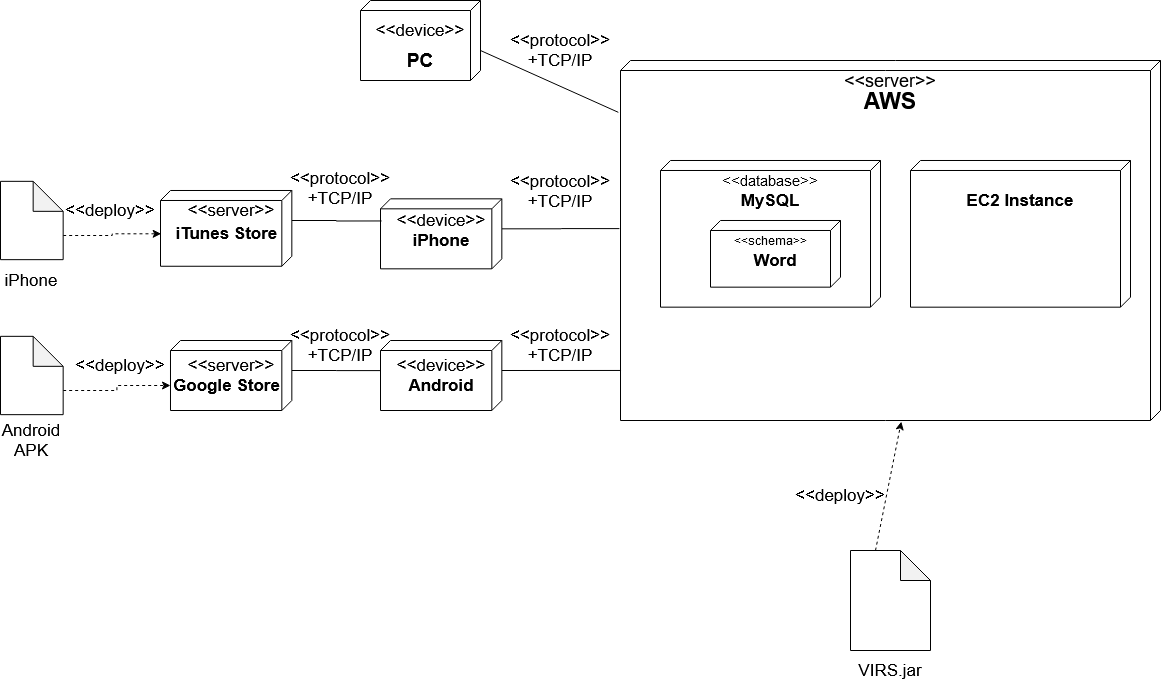
○ Run Unit tests

○ Run Integration tests

○ Create jar executable

● Bundle the application for AWS

● Upload to AWS servers.



*Figure 5-Deployment Diagram*

## Design Patterns

The following design patterns were used in the application.

**Dependency Injection:** This is a core design pattern for Spring and Angular 4. It allows the objects to be readily available and injected when needed rather than having to create them.

**Bridge:** This design pattern is used by the application to interact with libraries that are not native. Ex: tesseract and OpenCV

**Front controller:** Controllers are one of the main Components in the MVC design.

**Marker:** All components in Spring are annotated with markers to further specify the correct stereotype. This allows the initialization of the correct optimized components when the application runs.

**Module:** The application is divided in several modules according to functionality. This separation promotes organization and grouping of features.

# 

# System Validation

* **Test Case ID**: TC01\_Frontend\_Dependencies
  + **Description**: Frontend dependencies should be installed
  + **Test steps**:
    - 1. Navigate to virs code directory
    - 2. Run ‘virs.cmd install-only’ command
  + **Expected result**: Frontend installation success message
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC02\_Backend\_dependencies
  + **Description**: Backend dependencies should be installed
  + **Test steps**:
    - 1. Navigate to virs code directory
    - 2. Run ‘virs.cmd install-only’ command
  + **Expected result**: Backend installation success message
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC03\_IAM\_Users
  + **Description**: IAM users created
  + **Test steps**:
    - 1. Navigate to AWS IAM user console
    - 2. Verify existence of two separate IAM users
  + **Expected result**: Two new IAM users ‘Jorge’ and ‘Julian’
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC04\_AWS\_Host
  + **Description**: Testing whether application running on AWS
  + **Test steps**:
    - 1. Open browser and navigate to www.myvirs.com
  + **Expected result**: VIRS dashboard is displayed
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC05\_Decryption
  + **Description**: Testing whether the encrypted API keys are being decrypted
  + **Test steps**:
    - 1. Ensure encrypted versions of API keys are in xml file
    - 2. Navigate to the VIRS image upload page
    - 3. Upload an image with 100 words of text
    - 4. Click enhanced text button
    - 5. Click a word on the enhanced text panel
  + **Expected results**: Words are properly transcribed(Textract API) and clicking them brings up their definitions (Wikipedia API)
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC06\_Database\_Running
  + **Description**: Test whether database is up on AWS RDS
  + **Test steps**:
    - 1. Navigate to AWS console
    - 2. Open RDS control panel
    - 3. View databases
    - 4. Check status of database with identifier ‘aayrca52p4x1gp’
  + **Expected result**: Status of database is available
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC07\_Server\_communication
  + **Description**: Test whether application running on EC2 instance can communicate with RDS server
  + **Test steps**:
    - 1. Navigate to myvirs.com dashboard
    - 2. Navigate to dictionary page
  + **Expected result**: Words contained in the RDS database are listed on the dictionary page
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC08\_Beanstalk\_Integration
  + **Description**: Test that database is integrated into elastic beanstalk
  + **Test steps**:
    - 1. Navigate to AWS console
    - 2. Open elastic beanstalk control panel
    - 3. Navigate to virs-prod environment
    - 4. Open virs-prod environment configuration
  + **Expected result**: RDS database ‘aayrca52p4x1gp’ is listed as an associated database
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC09\_OCR\_Sunny
  + **Description**: Testing optical character recognition with minimal inaccuracies
  + **Test steps**:
    - 1. Navigate to the VIRS image upload page
    - 2. Click the browse button and select a test image to upload
    - 3. Click the enhanced text button
  + **Expected result**: Transcribed text displays within 90% accuracy of the test image
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC09\_OCR\_Rainy
  + **Description**: Optical character recognition failure
  + **Test steps**:
    - 1. Navigate to the VIRS image upload page
    - 2. Click the browse button and select an unreadable file to upload
    - 3. Click the enhanced text button
  + **Expected result**: Image is unreadable, display error message
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**:TC10\_Dictionary\_Header
  + **Description**: Test if header displays on dictionary page
  + **Test steps**:
    - 1. Navigate to dictionary page of VIRS
  + **Expected result**: Header reading ‘A Corpus-based K-12 School Dictionary’ is displayed above word lists
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**:TC11\_Bulk\_Upload\_Sunny
  + **Description**: Test if bulk word upload to database via csv is functional
  + **Test steps**:
    - 1. Login to the admin section of the site
    - 2. Click the ‘Browse..’ button
    - 3. Upload a csv containing words to be added to category K3
    - 4. Choose category K3 from the dropdown
    - 5. Click the add button
  + **Expected result**: Words added to K3 are shown in session history and the words are contained in the database after the operation is complete
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**:TC11\_Bulk\_Upload\_Rainy
  + **Description**: Test error message when bulk adding with no csv uploaded
  + **Test steps**:
    - 1. Login to admin section of the VIRS site
    - 2. Choose a category from the bulk add dropdown
    - 3. Click the add button
  + **Expected result**: Error message indicating that no csv has been uploaded
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC12\_Global\_Limit\_Sunny
  + **Description**: Test the global limit of the OCR image update when below threshold
  + **Test steps**:
    - 1. Artificially set the global counter below 1000
    - 2. Click the browse button and select a test image to upload
    - 3. Click the enhanced text button
  + **Expected result**:
    - Enhanced text displays accurately
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC12\_Global\_Limit\_Rainy
  + **Description**: Test the global limit of the OCR image update when above threshold
  + **Test steps**:
    - 1. Artificially set the global counter to over 1000
    - 2. Click the browse button and select a test image to upload
    - 3. Click the enhanced text button
  + **Expected result**:
    - Error message indicating the global OCR limit has been exceeded for this month
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC13\_Database\_Local\_Backup
  + **Description**: Test that the locally download virs schema can be restored from MySql dump file
  + **Test steps**:
    - 1. Navigate to file location via command line
    - 2. Run the command ‘mysql -u root -p vir < data.sql
  + **Expected result**: Database is populated with contents of the sql dump
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC14\_Upload\_and\_Replace\_Sunny
  + **Description**: Test that the upload and replace category is functioning correctly
  + **Test steps**:
    - 1. Navigate to the admin section of the site and login
    - 2. Click the browse button and upload a csv populated with words to be aded
    - 3. Choose a category to be replaced
    - 4. Press the upload and replace button
  + **Expected result**:
    - 1. The words that were previously in that category are wiped from the database
    - 2. The database contains the words from the csv, associated with their respective categories
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC14\_Upload\_and\_Replace\_Rainy
  + **Description**: Test that the upload and replace category is functioning correctly
  + **Test steps**:
    - 1. Navigate to the admin section of the site and login
    - 2. Choose a category to replace
    - 3. Press the upload and replace button
  + **Expected result**:
    - 1. An error message is displayed indicating that a csv has not been uploaded
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC15\_Admin\_Search\_Sunny
  + **Description**: Test that the search function on the admin panel is returning search results
  + **Test steps**:
    - 1. Navigate to the admin section of the site and login
    - 2. Search for a word that exists in the database
  + **Expected result**:
    - 1. The word searched for is displayed underneath the search bar with its category
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass
* **Test Case ID**: TC15\_Admin\_Search\_Rainy
  + **Description**: Test that the search function on the admin panel is functional
  + **Test steps**:
    - 1. Navigate to the admin section of the site and login
    - 2. Search for a word that does not exist in the database
  + **Expected result**:
    - 1. An message deisplays indicating the word does not exist in the database
  + **Actual result**: As expected
  + **Status(Pass/Fail)**: Pass

# Glossary

**Academic Word List (AWL):** List of words used in the natural English language with frequency high enough but that does not make it to the high frequency list.

**High Frequency List:** List of words used in natural English language with a high frequency.

**Medium Frequency List:** List of words used in natural English language with a medium frequency.

**Low Frequency List:** List of words used in natural English language with a low frequency.

**Flesch Reading Ease Score:** A test designed to calculate how hard a text is to understand in English.

**Word Definition:** Meaning of a word as per Wiki Dictionary. It contains etymology, meaning, and usage information.

**Category:** The assigned value to a word from one of the above lists. A word category can be: AWL, High Frequency, Medium Frequency, and Low Frequency.

**Inflection:** A modification of a word to express additional meanings: plural and conjugations.

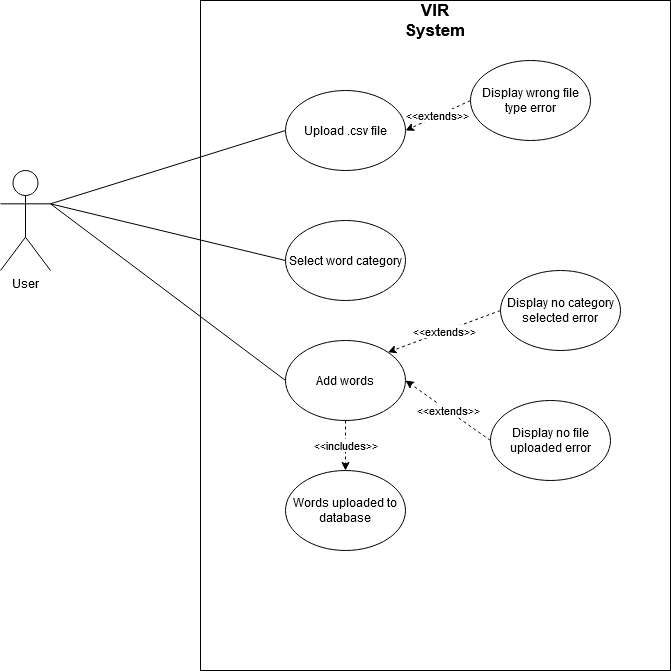
**School Dictionary:** A collection of all the lists. It can be used as a reference for the word categories.

# Appendix

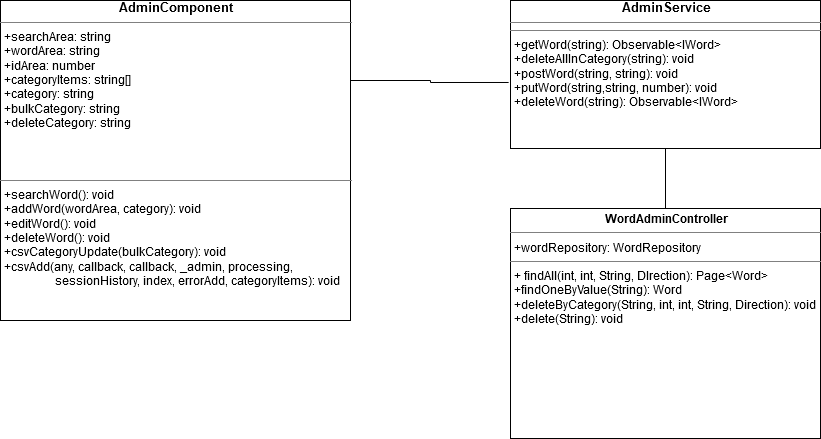
## Appendix A - UML Diagrams

## Static UML Diagrams

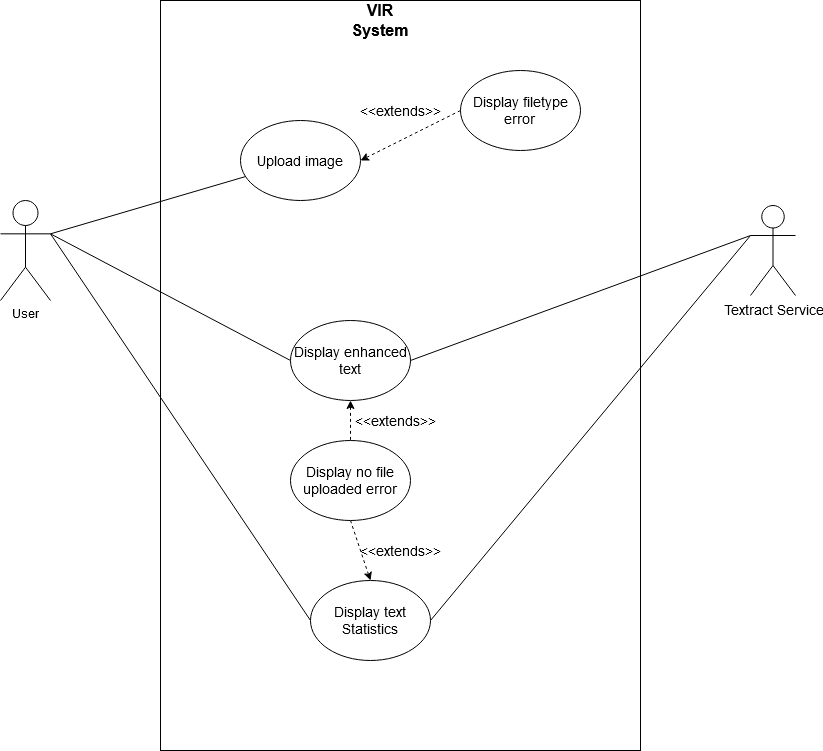
**Bulk Word Upload Use Case Diagram**

****

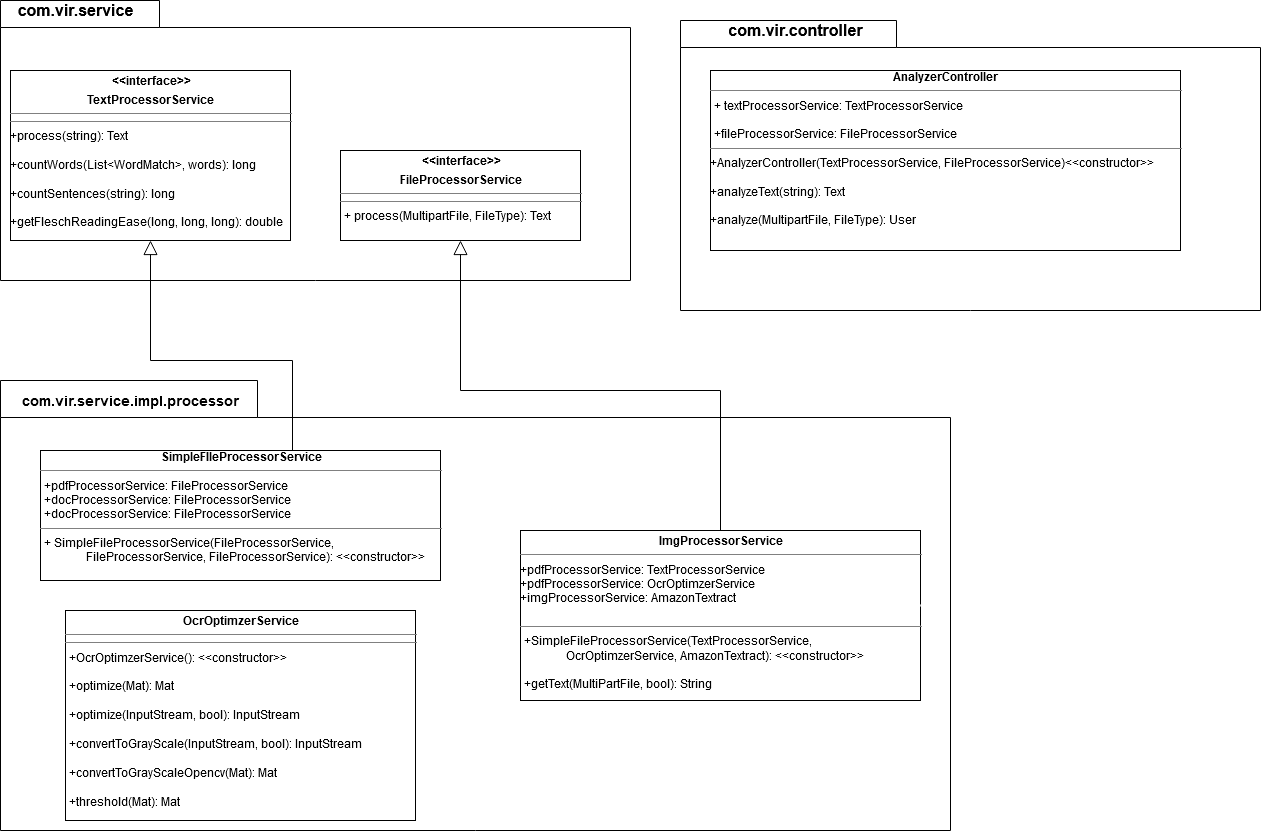
## Bulk word upload Class Diagram



**Image upload to enhanced text Use Case Diagram**

****

**Image upload to enhanced text Class Diagram**

****

## Dynamic UML Diagrams

## Bulk word upload Sequence Diagram

## 

## 

## 

## 

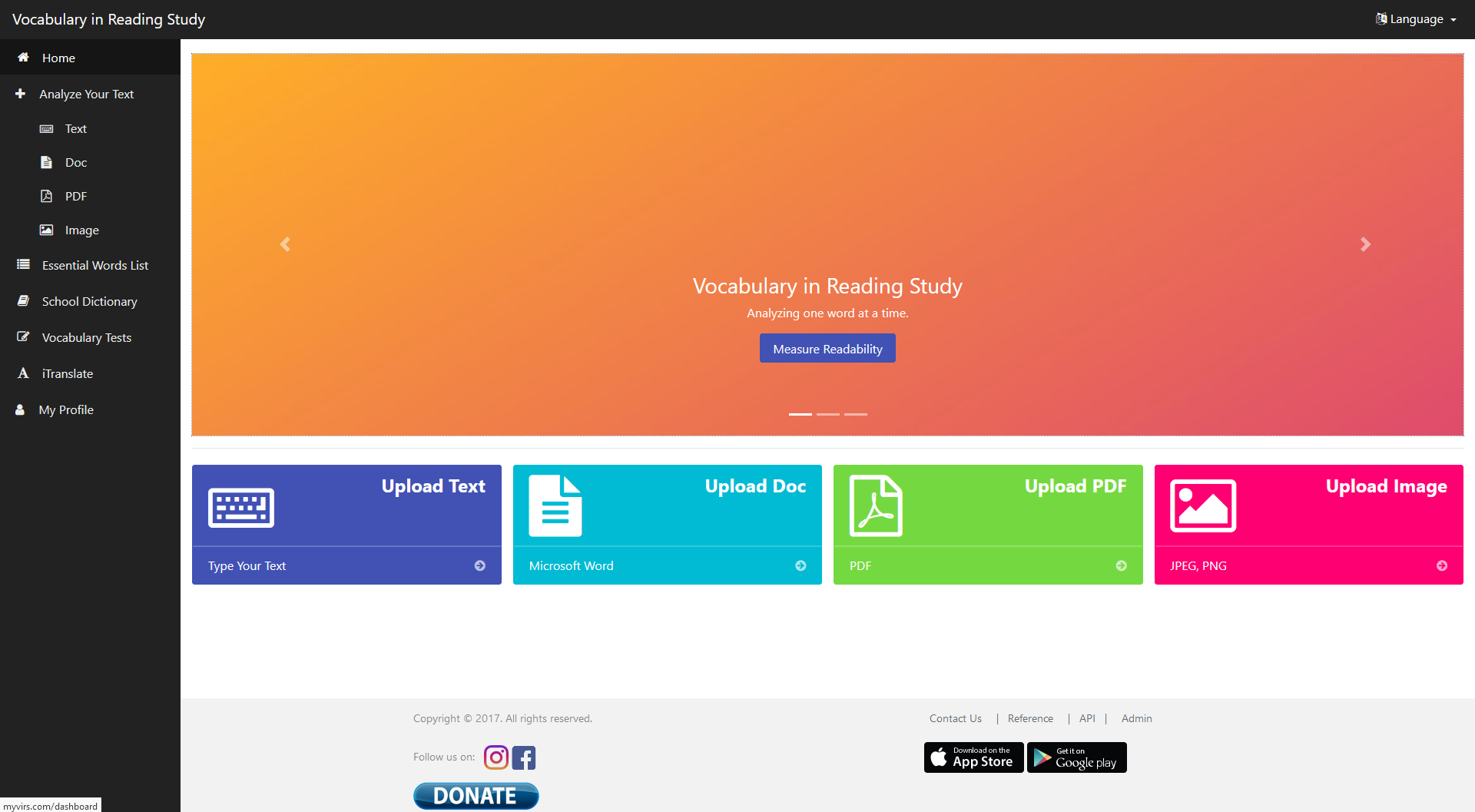
## Image upload to enhanced text Sequence Diagram

## 

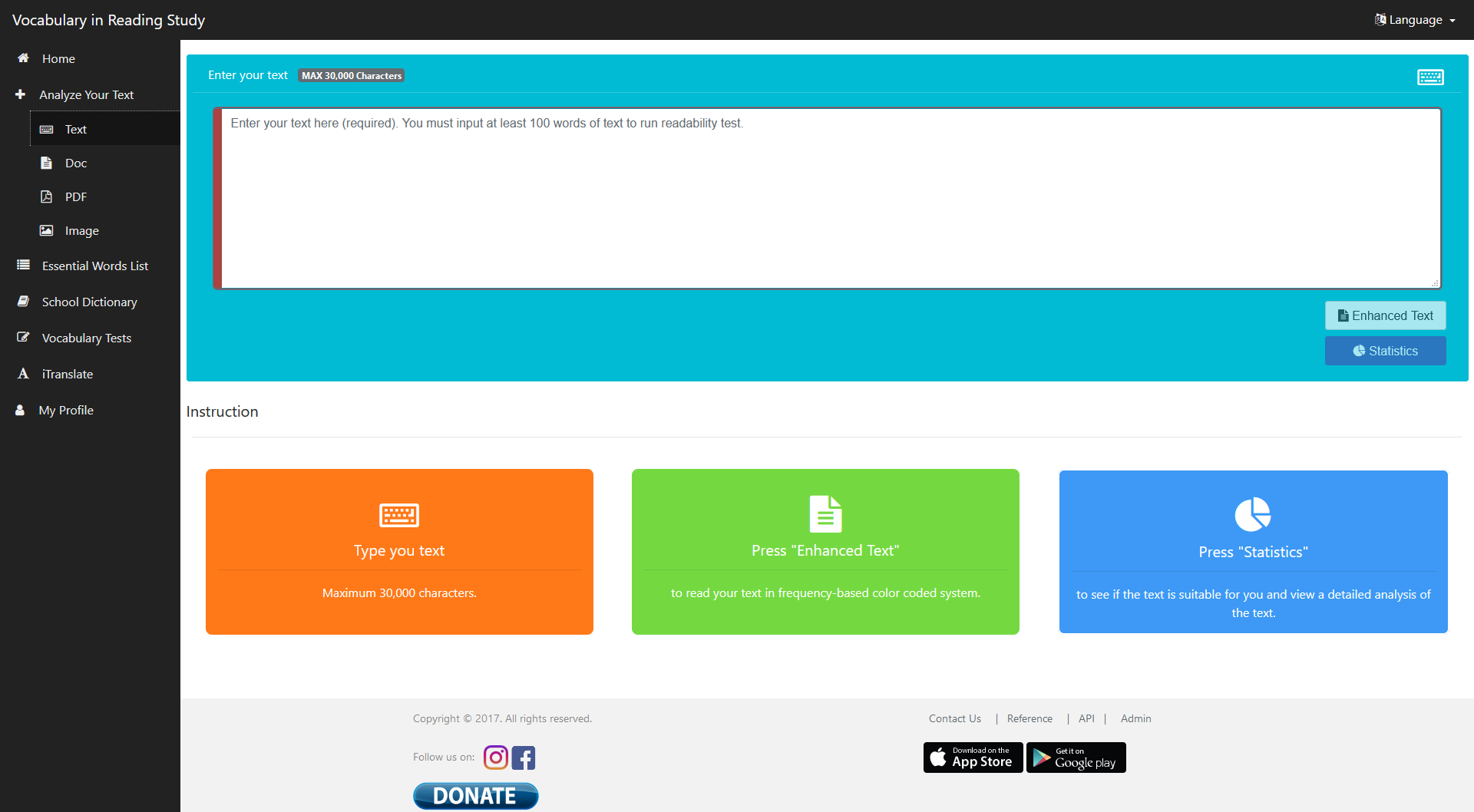
## 

## Appendix B - User Interface Design

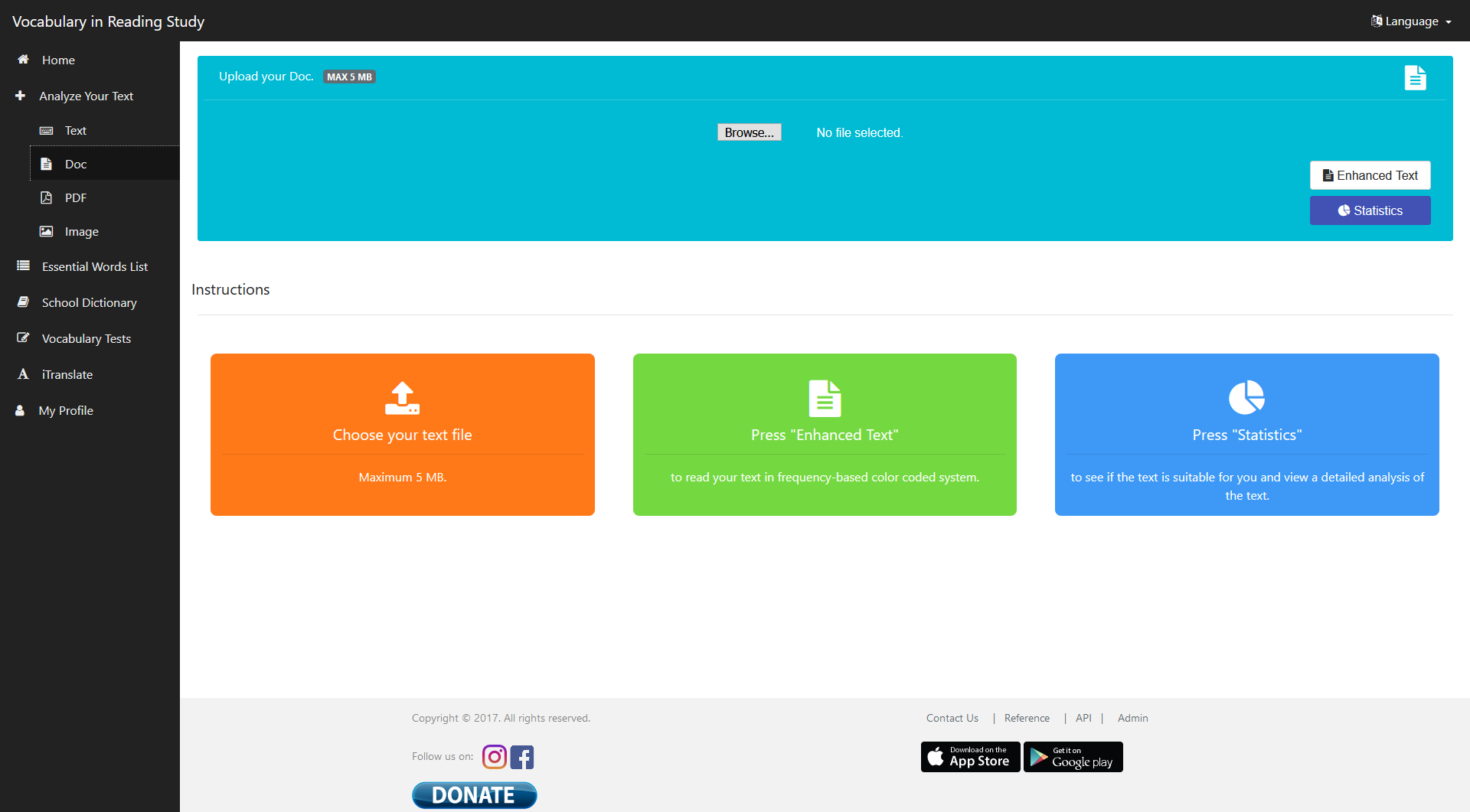
**Dashboard**



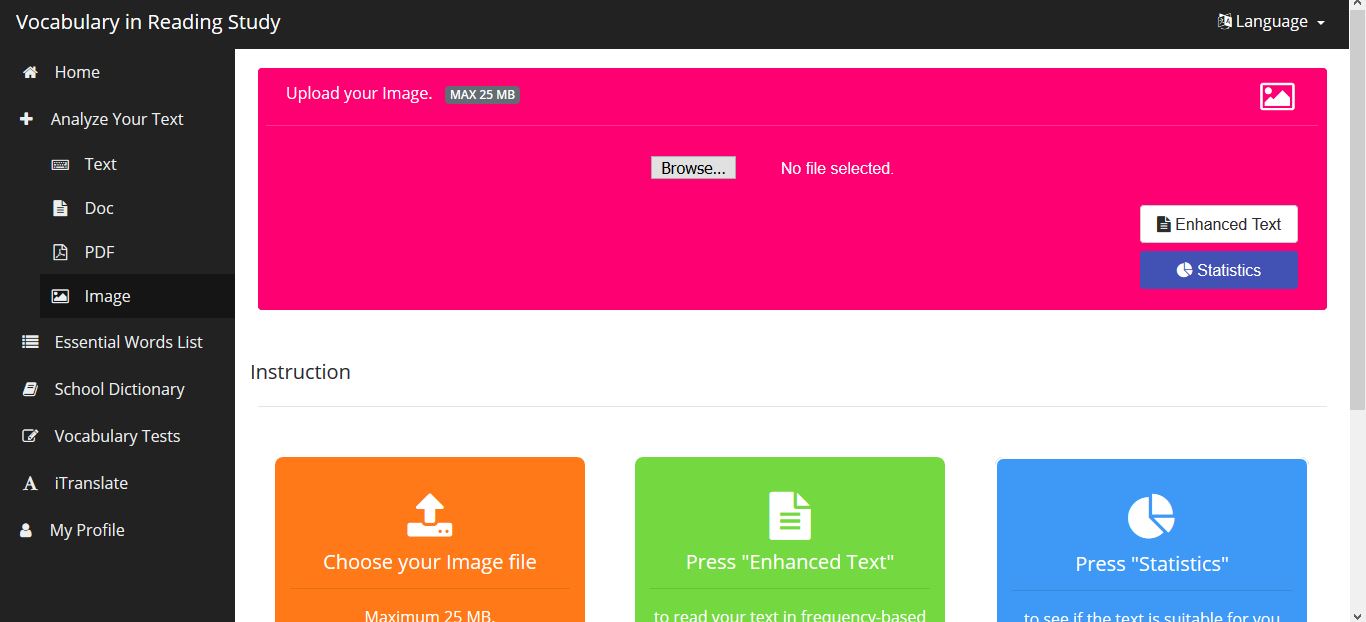
**Text upload screen**



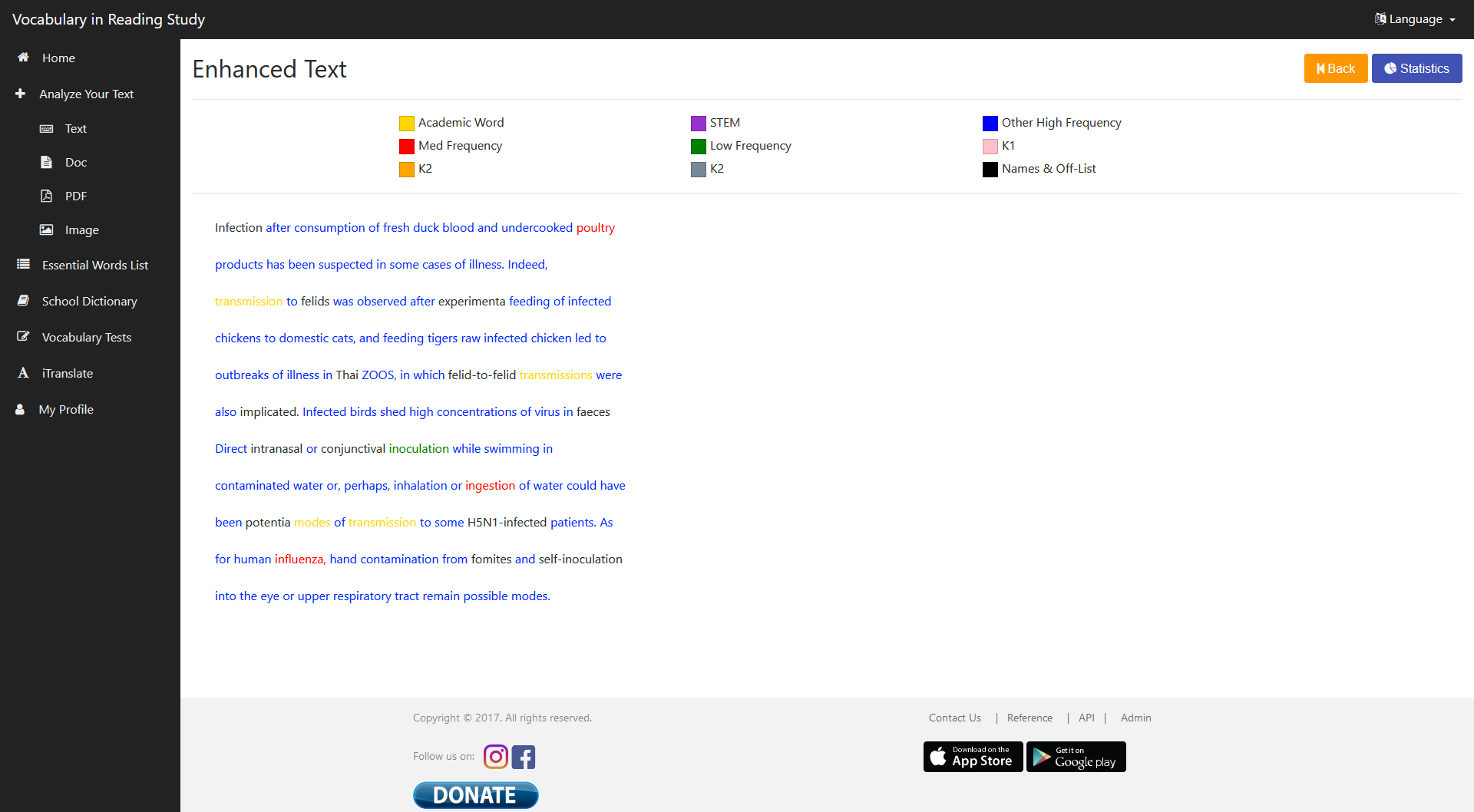
**Document upload screen**



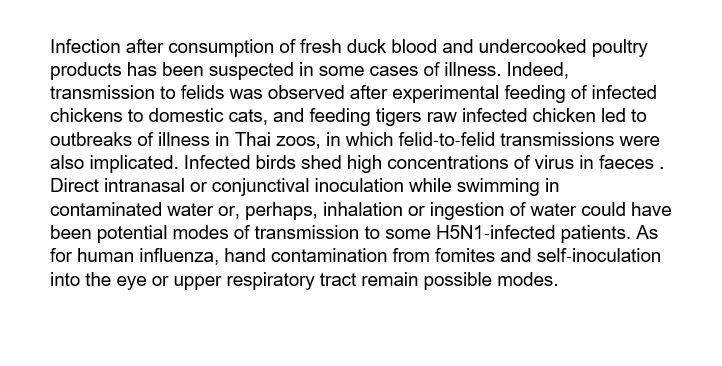
**Image upload screen**

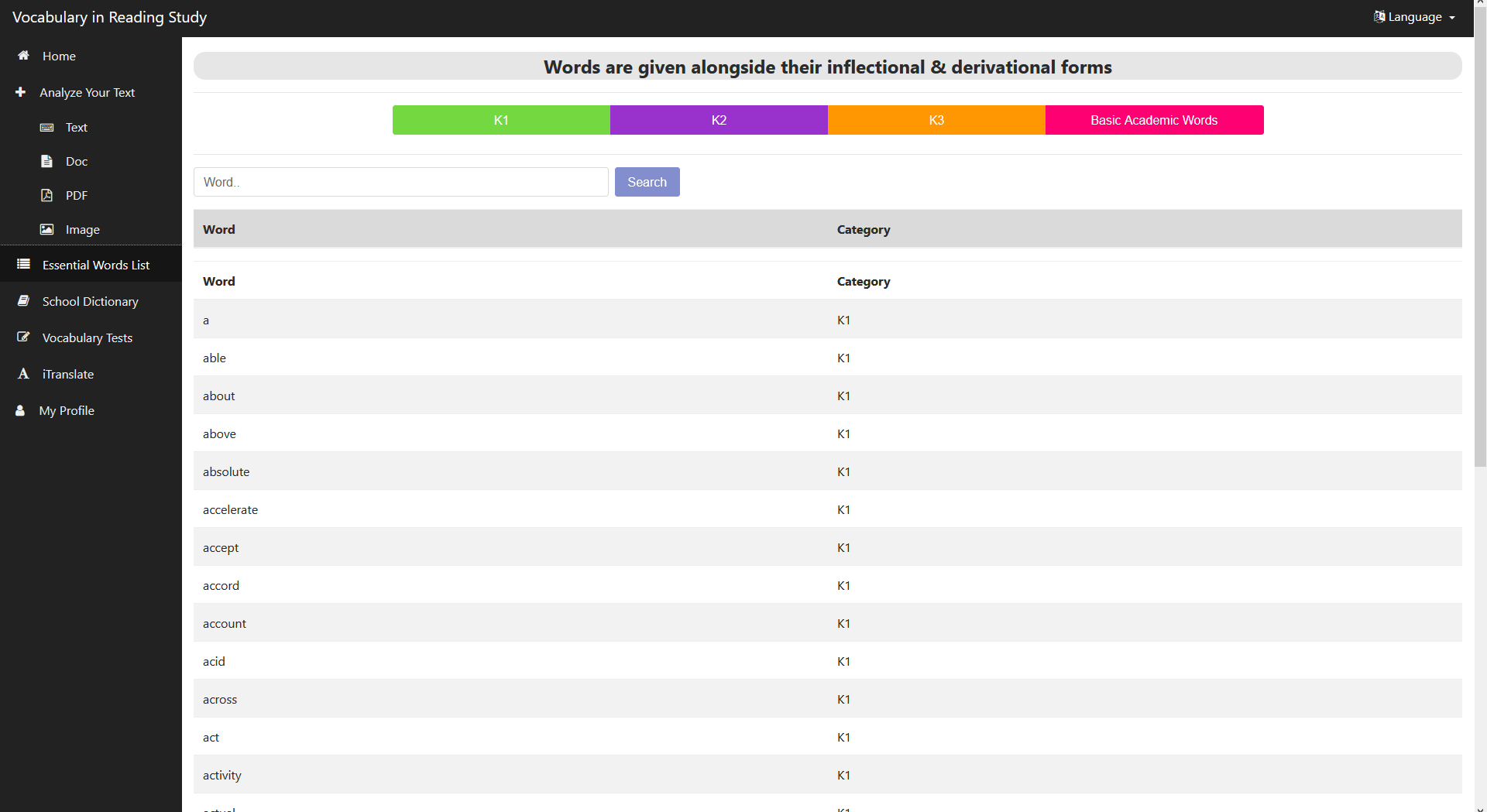


**Enhanced Text display**

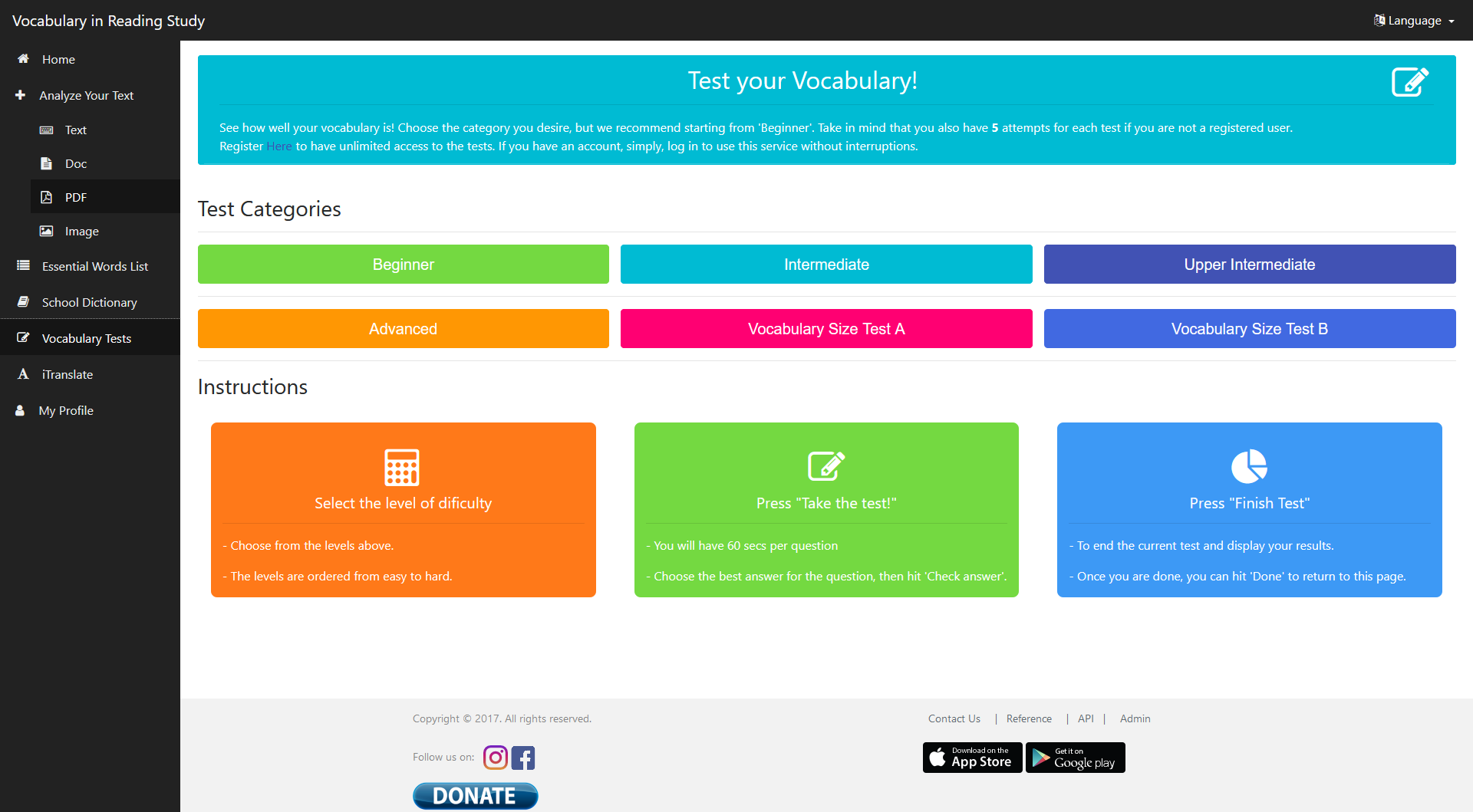


**Enhanced text submission**

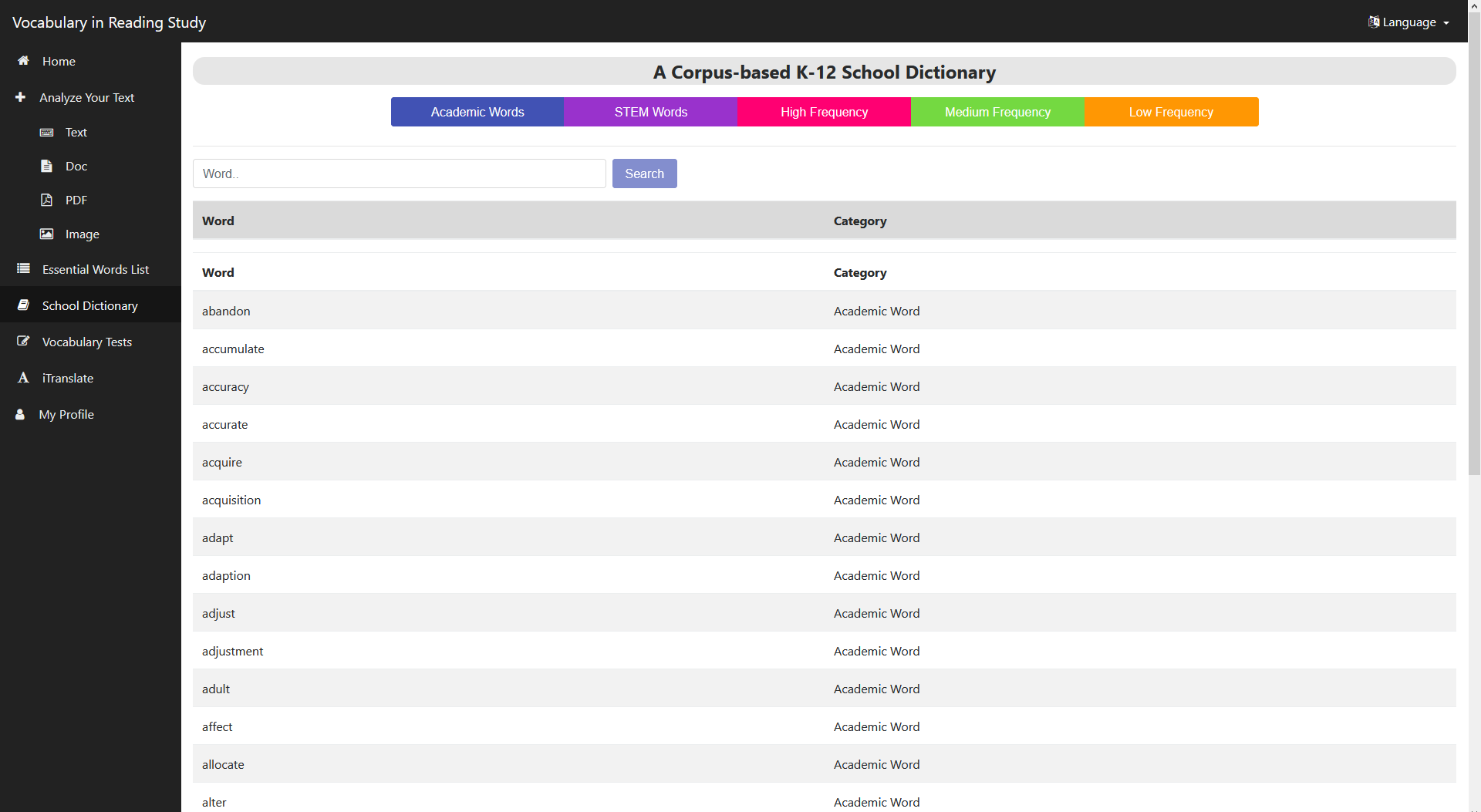


**Essential Words List**

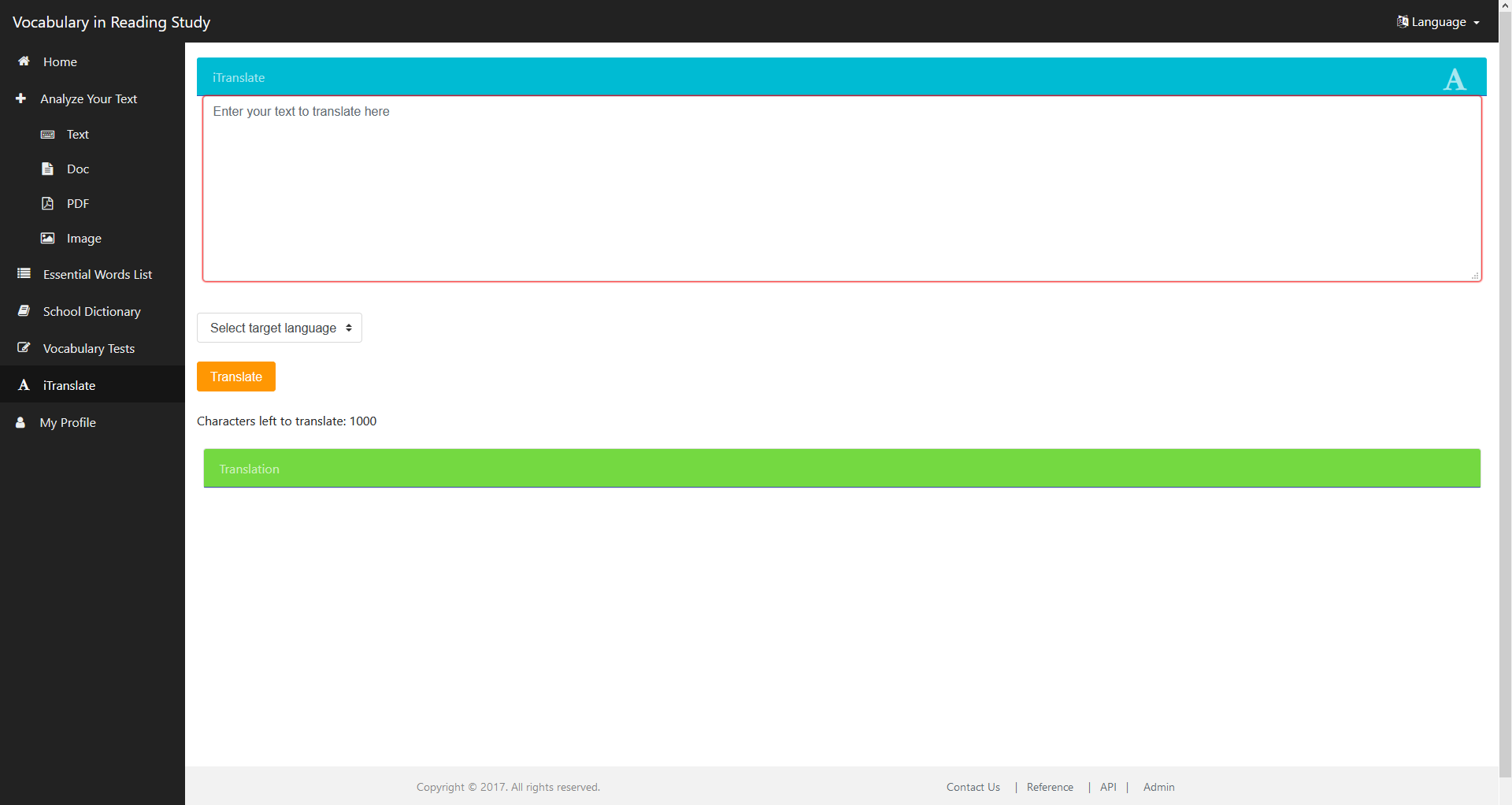
**Vocab test**



**Dictionary**



**Translator**



## Appendix C - Sprint Review Reports

### Sprint 1

SP20 Team26 Review 1/28/2020 3pm 3:45pm

* Jorge Costa, Julian Pineiro, Ehsan Seyedjafari
* List of Approved User Stories:
  + User Story - install frontend dependencies and run the front end locally
    - As a developer, I would like to install the frontend’s dependencies, so that the application can run.
    - Initially assigned to Julian and Jorge
    - Medium size - 4 points
  + User Story - investigate frontend structure
    - As a developer, I would like to investigate and understand frontend program structure in order to facilitate the setup of the application and build on top of it in later sprints
    - Initially assigned to Julian and Jorge
    - Small size - 2 points
  + User Story - Install the Dependencies/Research the backend of the application
    - As a developer, I would like to install the backend dependencies and fully understand how the backend works fixing any errors we might encounter.
    - Initially assigned to Julian and Jorge
    - Large size - 8 points
  + User Story - Preliminary hosting research
    - As a developer I would like to perform preliminary research on AWS hosting
    - Initially assigned to Julian and Jorge
    - Small size - 2 points
  + User Story - Create IAM users
    - As a developer I would like to create IAM users on the AWS account to access the AWS service without the root user
    - Initially assigned to Jorge
    - Small size - 1 points
  + User Story - Organize meeting dates.
    - As a developer I would like to organize meeting dates for the rest of the semester.
    - Initially assigned to Julian.
    - Small size - 1 point
* List of Rejected User Stories:
* None

### Sprint 2

SP20 Team26 Review 2/10/2020 5:00pm 5:20pm

* Jorge Costa, Julian Pineiro, Ehsan Seyedjafari
* List of Approved User Stories:
  + User Story 1 - Run the application with minimal features on AWS
    - As a Product owner I would like to see some version of the application running on AWS.
    - Initially assigned to both Jorge and Julian
    - Large Size - Very important Priority - 8 points
  + User Story 2 - Improve the features that are run locally while making sure integration into AWS works.
    - As a Product owner I would like more features to be functionable when using the application
    - Initially assigned to both Jorge and Julian
    - Large Size - High priority - 8 points
  + User Story 3 - Get encryption working with API secrets locally and AWS
    - As a product owner I would like to keep API keys secret when running the program.
    - Initially assigned to Julian
    - Medium Size - Medium priority - 4 points
  + User Story 4 - Re-evaluate AWS billing
    - As the product owner I would like to stop recurring payments to AWS for services that are not currently necessary
    - Initially assigned to Jorge.
    - Medium Size - Low/Medium Priority - 4 points
  + User Story 5 - CI/CD research
    - As a developer I would like to research CI/CD to facilitate efficient development.
    - Initially assigned to Julian.
    - Small Size - Low Priority - 2 points
  + User Story 7 - Resolve database issues on AWS
    - As a developer I would like to have a functioning database to run the application.
    - Initially assigned to Julian
    - Small Size - Medium priority - 2 points
  + User story 8 - Resolve port/communication errors between instances
    - As a developer I would like to resolve the communication errors between the instance running the application and the database
    - Initially assigned to Jorge
    - Small Size - Medium priority - 4 points
  + User Story 9 - Solve bundling issues with Elastic Beanstalk
    - As a developer I would like to have a functioning database to run the application.
    - Initially assigned to Julian
    - Small Size - Med priority - 1 points
* List of Rejected User Stories:
  + User Story 5 - CI/CD research
    - As a developer I would like to research CI/CD to facilitate efficient development.
    - Initially assigned to Julian.
    - Small Size - Low Priority - 2 points
    - Reason for rejection - Was not important for the sprint so it was not done, but instead moved over the the next sprint as a low priority task.

### Sprint 3

SP20 Team26 Review 3/2/2020 2:00pm 2:30pm

* Jorge Costa, Julian Pineiro, Ehsan Seyedjafari
* List of Approved User Stories:
  + User Story 1 - Conduct research on options for optical character recognition
    - As a developer I would like to research my options for implementing OCR
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + User Story 2 - Conduct research on options for CSV upload to DB
    - As a developer I would like to research my options for database upload via csv file
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + User Story 3 - Implement optical character recognition with minimal inaccuracies
    - As a user I would like to be able to upload an image of text and have the application interpret the image as text
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + User Story 4 - Add header to dictionary page
    - As a product owner I would like the dictionary page to have an additional header
    - Initially assigned to Julian
    - Small size - 2 points
  + User Story 6 - Make adjustments to communications between application and database
    - As a developer I would like to secure communication between the database and the application
    - Initially assigned to Jorge
    - Small size - 2 points
  + User Story 7 - Conduct research into RDS pricing
    - As a product owner I would like to lower the cost of hosting the server on AWS
    - Initially assigned to Jorge
    - Small size - 2 points
  + User story 8 - Create a Github repository for the project and compile changes made
    - As a developer I would like to manage changes to the project and compiling the previous changes made for the project
    - Initially assigned to Julian
    - Small size - 2 points
* List of Rejected User Stories:
  + User Story 5 - Make adjustments to word category coloring
    - As a product owner I would like to alter the coloring that certain word categories are identified by
    - Initially assigned to Jorge
    - Small size - 2 points
    - Reason for rejection - Rolling it over to the next sprint specified requirements kind of lacking and will be given at a later date.

### Sprint 4

SP20 Team26 Review 3/17/2020 2:00pm 2:30pm

* Jorge Costa, Julian Pineiro, Ehsan Seyedjafari
* List of Approved User Stories:
  + Implement bulk word upload to the DB by admin via CSV
    - As a product owner I would like to upload words to the database in bulk
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + Update K1 and K2 word lists
    - As a product owner I would like to make some changes to the existing K1 and K2 word lists to resolve inconsistencies
    - Initially assigned to Jorge
    - Medium size - 4 points
  + Add another category “K3” list to the project
    - As a product owner I would like to add a new category of words for the project
    - Initially assigned to Julian
    - Medium size - 4 points
  + Get previous Tests working on the project
    - As a developer I want to make sure previous tests work with the production build.
    - Initially assigned to Julian
    - Medium Size - 4 points
  + Establish a global limit for how many documents can be scanned by textract
    - As a product owner I would like to limit the amount of documents that go through textract in order to limit costs
    - Initially assigned to Jorge
    - Medium size - 4 points
* List of Rejected User Stories:
  + Make adjustments to word category coloring
    - As a product owner I would like to alter the coloring that certain word categories are identified by
    - Initially assigned to Jorge
    - Small size - 2 points

### Sprint 5

SP20 Team26 Review 4/1/2020 3:00pm 3:30pm

* Jorge Costa, Julian Pineiro, Ehsan Seyedjafari
* List of Approved User Stories:
  + Conduct preliminary research on the best options for hosting a backup server
    - As as a developer I would like to better understand my options for hosting a backup so that I can decide on the best implementation
    - Initially assigned to Julian
    - Medium size - 4 points
  + Back up the database locally
    - As a product owner I would like to backup my database locally in case of a data loss event
    - Initially assigned to Jorge
    - Medium size - 4 points
  + Make adjustments to word category coloring
    - As a product owner I would like to alter the coloring that certain word categories are identified by
    - Initially assigned to Jorge
    - Small size - 2 points
  + Upgrade administrator Database editing tools
    - As a product owner/administrator I would like to be have more options for uploading to/editing the database
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
  + Upgrade admin DB editing tool interface
    - As a product owner/administrator I would like the DB editing interface to be more usable
    - Initially assigned to Julian
    - Medium size - 4 points
  + Conduct research into EC2 cost cutting measures
    - As a product owner I would like to pay less for the EC2 instance the application is running on
    - Initially assigned to Jorge
    - Medium size - 4 points
* List of Rejected User Stories:
  + Host a backup of the site
    - As a developer I would like to set up a backup host to be used in the event of an aws outage
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
    - Reason for rejection - Unsure about the full details and what the product owner specifically wants so we pushed this off to the next sprint.

### Sprint 6

SP20 Team26 Review 4/15/2020 3:00pm 3:15pm

* Jorge Costa, Julian Pineiro, Ehsan Seyedjafari
* List of Approved User Stories:
  + Merging branches and conflicts
    - As a developer I would like to merge all branches of the project so all features can be integrated
    - Initially assigned to Julian
    - Small size - 2 points
  + Code refactoring image recognition
    - As a developer I would like to refactor the code in the image and text processors to make it more readable
    - Initially assigned to Jorge
    - Small size - 3 points
  + Fix admin search function
    - As a product owner I would like the search function on the admin panel to consistently return words searched for
    - Initially assigned to Julian
    - Medium size - 8 points
  + Fix database category update function
    - As a product owner I would like the replace function to replace the intended category without affecting other categories
    - Initially assigned to Jorge
    - Medium size - 8 points
  + Make changes to dictionary page
    - As a product owner I would like to update the dictionary page in order to make it more readable
    - Initially assigned to Julian
    - Small size - 2 points
  + Assess necessary aesthetic changes to each page on the site
    - As a developer I would like to assess what changes need to be made to the frontend of each page on the site
    - Initially assigned to Jorge and Julian
    - Medium size - 8 points
* List of Rejected User Stories:
  + Host a backup of the site
    - As a developer I would like to set up a backup host to be used in the event of an aws outage
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
      * Reason for rejection - Unable to register site with product owner’s desired host due to coronavirus related complications

### Sprint 7

SP20 Team26 Review 4/27/2020 3:00pm 3:15pm

* Jorge Costa, Julian Pineiro
* List of Approved User Stories:
  + Fix enhanced text bugs
    - As a developer I would like to fix bugs caused by backend changes made during last sprint’s bug fixes
    - Initially assigned to Jorge and Julian
    - Size - 8 points
  + Code clean up on some files changed within the project
    - As a developer I would like the updated changes made to be readable and consistent for the next group
    - Initially assigned to Julian
    - Size - 6 points
  + Poster
    - As a developer I would like to create a poster showcasing the progress made on the application
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Presentation slides
    - As a developer I would like to create presentation slides to showcase the progress made on the application
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Videos
    - As a developer I would like to create videos using the presentation slide which describe the project
    - Initially assigned to Jorge and Julian
    - Medium size - 4 points
  + Compiling documentation
    - As a developer I would like to update some aspects of the documentation to really explain and help avoid some of the issues we encountered this semester.
    - Initially assigned to Jorge
    - Medium size - 6 points
  + Complete final document
    - As a developer I would like to create a comprehensive document that describes the project
    - Initially assigned to Jorge
    - Medium size - 6 points
* List of Rejected User Stories:
  + Host a backup of the site
    - As a developer I would like to set up a backup host to be used in the event of an aws outage
    - Initially assigned to Jorge and Julian
    - Large size - 16 points
      * Reason for rejection - Unable to register site with product owner’s desired host due to coronavirus related complications

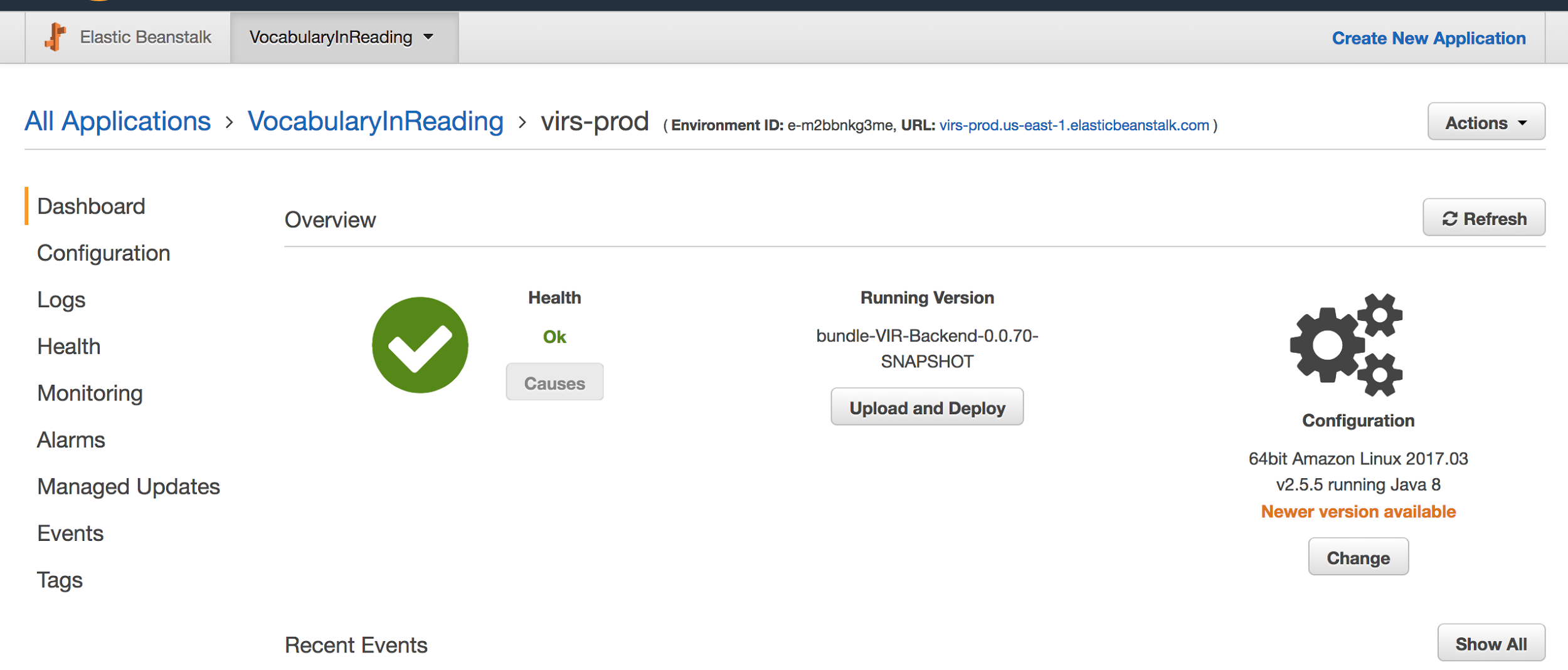
## Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

**Manuals**

These are all the manuals for the system

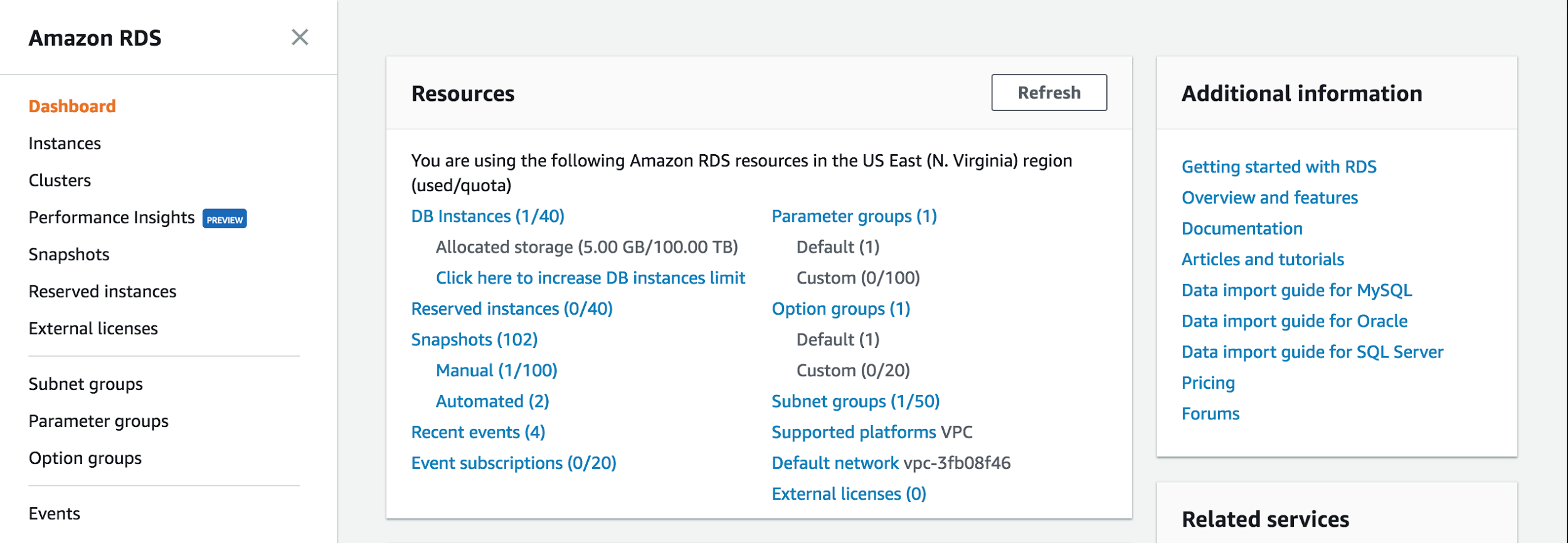
*Creating AWS Environment from scratch*

**Elastic Beanstalk Dasboard**



* If needed create a new Application.
* Create a new environment with all the default configurations and the sample application.
* Under Config got ahead and create a database with the "username" and "password" (Note the password has to be long, keep in mind that this password will be encrypted later on)
* Under Config/Software Configuration add a new Property name with the name used in the java EncryptorConfig.java for the name and the value.

**RDS Dashboard**



* Go to the database associated with the beanstalk above.
* Select it and click Instance Details.
* Click on the security groups: the one that starts with "rds-aws...."
* There go to the "InBound" tab at the bottom and add a new TCP protocol with port 3306 and pick your ip address. (This will allow to connect using workbench)

**Workbench**

* Login with the credentials and the connection string listed on the RDS dashboard.
* Create a new database.
* Populate any data needed.

**Java IDE**

* Add properties for the database, encryptor place holder, and port 5000 (needed for production)
* Compile and generate JAR.

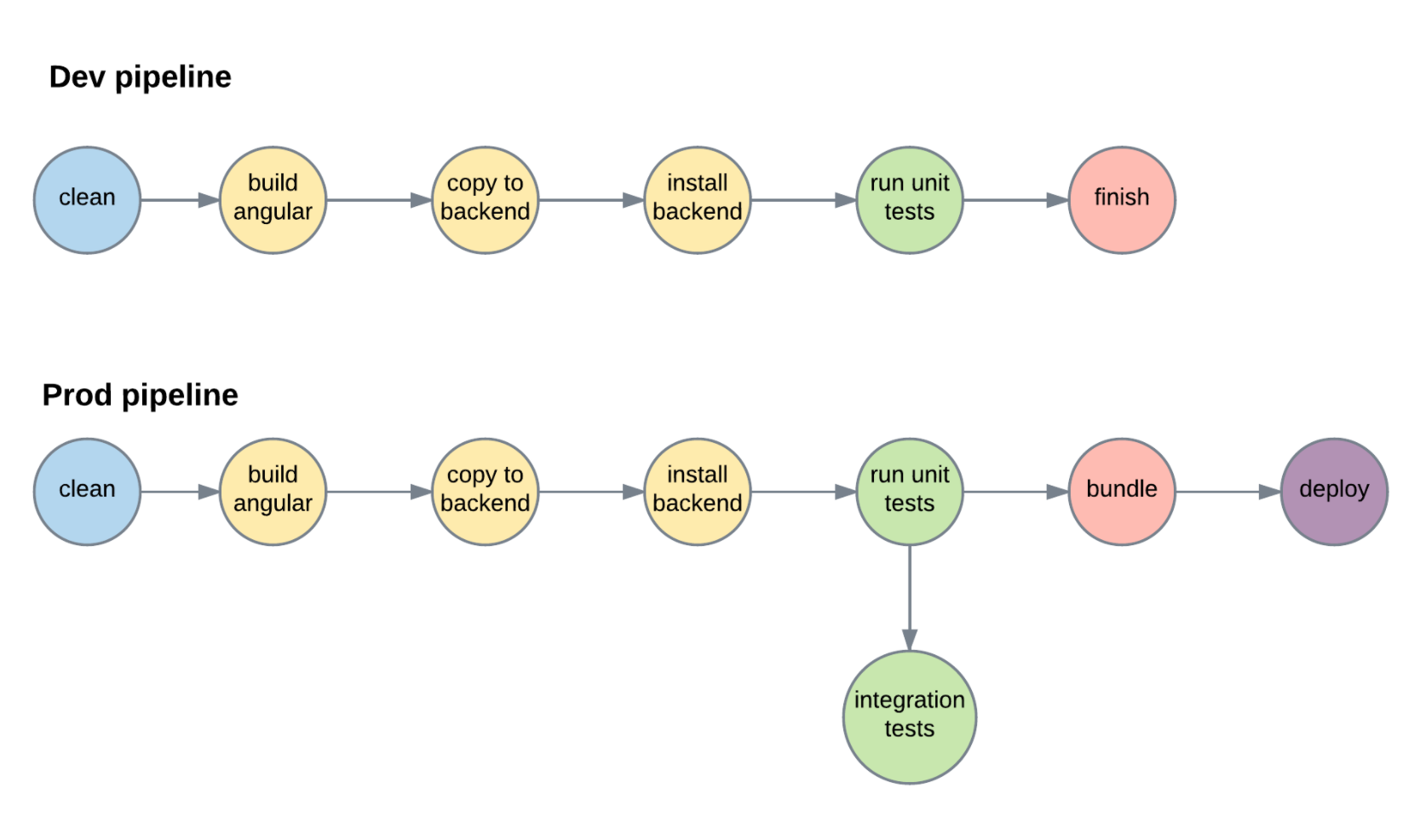
**Elastic Beanstalk Dashboard**

* Upload the new version of the application

**EC2 Dashboard**

* If you do not have a .pem file. Create a key to login via ssh.
* Click 'Key pairs' to go to the section of the keys.
* Create a copy of the current VM instance
* Locate the EC2 instance and right click on it to get the ssh connection information. (for the key to work it has to be associate with the BeanStalk instance, this should be already done, but you can do this in the 'Instance settings' of the Beanstalk)

*Building and Deploying the Application*



* Open a command line and navigate to the Code folder
* Run ‘virs.cmd prod-build’ (‘./build.sh prod’ on linux)
* Upload the generated bundle located in /<project root>/Code/VIR-Backend/release to the elastic beanstalk environment

*Local installation*

* Create a local MySQL account with the following credentials
  + Username: root
  + Password: root
* Restore the vir schema using the sql files contained in <project root>\Code\VIR-Backend\src\main\resources
* Run virs.cmd install-only to install the project
* Run virs.cmd run to run the project
* Access the application at localhost:8080

**Shortcomings/Wishlist**

**Price optimization**

* With this project running on AWS there are a couple services including EC2 instances and the RDS database.
* We talked about adding some optimization to the pricing but never got around to it.
* Currently these services are running On-Demand instances, but if these instances were running on reserved there should be a cost reduction of around 30%. Look at options and do think about possible scaling options for growth if choosing a long term plan.

**Back up Website**

* Before our time working on the project the website was down for around 5 months due to some issues with the previous group.
* While we did not have time to add it one of the product owners has expressed some interest in possibly having a back-up site. This in any case prevents the product owner's from running into any issues that will render the website useless.

**Enhanced Text (Multiple Categories)**

* Some words may be in multiple categories and enhanced text displays a color coated copy of the text for users to read.
* Frequency and other categories can have overlapping words.
* Easy way to see words that belong to multiple categories while viewing enhanced text (Possibly shown with the dictionary popup).

**OCR Improvements**

* The OCR image analyzer reads text from images in a very accurate way.
* Currently this reads from left to right but can cause issues in certain text where there is separation down the middle.

# References

• Begeny, J. C. and Greene, D. J. (2014), CAN READABILITY FORMULAS BE USED TO SUCCESSFULLY GAUGE DIFFICULTY OF READING MATERIALS?. Psychol. Schs., 51: 198–215. doi:10.1002/pits.21740

• Flesch, Rudolf. “How to Write Plain English.” Guide to Academic Writing Article - Management - University of Canterbury - New Zealand, [www.mang.canterbury.ac.nz/writing\_guide/writing/flesch.shtml](http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml).

• “Tesseract-Ocr/Tesseract.” GitHub, 10 Nov. 2017, github.com/tesseract-ocr/tesseract.

• “OpenCV library.” OpenCV library, opencv.org/.

• R. Guo, Q. Dai and D. Hoiem, "Single-image shadow detection and removal using paired regions," CVPR 2011, Providence, RI, 2011, pp. 2033-2040. doi: 10.1109/CVPR.2011.5995725 URL: 'http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5995725&isnumber=5995307'

• Zucker, Matt. “Page dewarping.” Needlessly complex, 14 Aug. 2016, mzucker.github.io/2016/08/15/page-dewarping.html.

• “Wiktionary” Wiktionary, the free dictionary, en.wiktionary.org/wiki/Wiktionary:Main\_Page.

•”Detecting and Analyzing Text in Single-Page Documents” AWS Documentation, https://docs.aws.amazon.com/textract/latest/dg/sync.html