

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: Automated Essay Grading

Team Members:

David Llerena
Timothy King
Malcolm Verdier
Jalen Williams
Ailany Rodriguez
Matias Fuenzalida
Michael Wei
Amani Hunt

Product Owner:

David Gray

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 the StudyBuddy Pro essay grader. It is an automated program that will check a student's essay based on the grammar, similarity, and the ability to stay on topic relating to the problem. This project is set to eliminate the time and resources used to manually check essay answers. This essay grader will manage that for the user. This document describes the contributions that our group has made within the Spring 2022 semester in detail, including but not limited to, UML diagrams, System Design, and Agile Ceremonies.

Table of Contents

INTRODUCTION	6
CURRENT SYSTEM	6
PURPOSE OF NEW SYSTEM	6
USER STORIES	
IMPLEMENTED USER STORIES	7
PENDING USER STORIES	12
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	13
SPRINTS PLAN	14
<i>Sprint 1</i>	14
<i>Sprint 2</i>	17
<i>Sprint 3</i>	19
<i>Sprint 4</i>	21
<i>Sprint 5</i>	23
<i>Sprint 6</i>	26
<i>Sprint 7</i>	28
SYSTEM DESIGN	
SYSTEM AND SUBSYSTEM	
DECOMPOSITION	31
DEPLOYMENT DIAGRAM	32
DESIGN PATTERNS	33
SYSTEM VALIDATION	34
GLOSSARY	35
APPENDIX	
APPENDIX A - UML DIAGRAMS	36
<i>Use Case Diagrams</i>	36
<i>Class Diagrams</i>	37
<i>Sequence Diagrams</i>	38
APPENDIX B - USER INTERFACE	
DESIGN	40
APPENDIX C - SPRINT REVIEW	
REPORTS	43

APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	52
REFERENCES	54

INTRODUCTION

StudyBuddy Pro is a law education platform that hosts a variety of study aids that can help a student succeed in law school. These study aids include features such as Case Briefs for many different cases, lessons, outlines, courses, multiple choice quizzes and exam prep workshops among other features. This project aims to create an automated essay grading system that can spot various different topics such as issues, rules, policies, inferences and conclusions in a student answer and grade them accordingly. Through this system a student should be able to receive instant feedback from their essay responses and improve their skills.

Current System

StudyBuddy Pro currently contains a myriad of law study aids, however it does not contain an automated essay grading system.

Purpose of New System

The new system is an automated essay grading system built from the ground up. It uses Natural Language Processing, some of which is provided by the spaCy library, in order to process the input answer provided by the student, which is then filtered through a rule-based system in order to classify the text under different topics. After which, this input is then filtered through a system that checks for proper grammar, similarity to a professor's answer, coherence, and structure. Finally, the system will then output a fully graded student essay with feedback on a results page.

USER STORIES

Implemented User Stories

User Story #	1
Assigned to	All team members.

User Story	As a Developer I should be able to know what NLP libraries are most relevant to address the problems faced in our project.
Acceptance Criteria	Develop clear understanding of NLP Technology

User Story #	2
Assigned to	Alexander Jimenez, Timothy King, David Llerena, Malcolm Verdier, Jalen Williams, Amani Hunt, Ailany Rodriguez, Michael Wei
User Story	As a Developer I should be able to know how to identify which subproblem of the NLP domain our work includes.
Acceptance Criteria	Develop clear understanding the problem to be tackled

User Story #	3
Assigned to	Timothy King, David Llerena, Malcolm Verdier, Jalen Williams, Amani Hunt, Ailany Rodriguez, Michael Wei
User Story	As a project member I should be able to know if the project is on track to produce our minimum deliverables for the semester, and what sections are critical.
Acceptance Criteria	Create GitHub repository for version control and establish method of communication for team cohesion

User Story #	4
Assigned to	All members

User Story	As an instructor, I want the system to be able to construct an inferred rulebook for student answers based upon the instructor's answer.
Acceptance Criteria	Create a rulebook based on labeled answer provided by professor

User Story #	5
Assigned to	Alexander Jimenez, Ailany Rodriguez, Malcolm Verdier
User Story	As a professor, I wish the program identifies sections in my answer automatically.
Acceptance Criteria	Classify student text into topics

User Story #	6
Assigned to	Matias Fuenzalida
User Story	As a developer, I want to tokenize student & professor answers into separate sentences and words so that it can later be used by the machine.
Acceptance Criteria	Parse student answer and professor answer into sentences and words for further processing

User Story #	7
Assigned to	David Llerena, Malcolm Verdier, Ailany Rodriguez, Timothy King, Matias Fuenzalida, Michael Wei, Jalen Williams
User Story	As a developer, I want to be able to use tokenized text to extract meanings

	and identify keywords plus any synonyms.
Acceptance Criteria	Process text using various NLP tasks

User Story #	8
Assigned to	David Llerena, Michael Wei, Jalen Williams
User Story	As a developer, I want to label blocks of text.
Acceptance Criteria	Classify parts of the student text into topics

User Story #	9
Assigned to	Malcolm Verdier, Timothy King, Ailany Rodriguez, Alexander Jimenez, Matias Fuenzalida, Amani Hunt
User Story	As a developer, I want to use synonyms to identify the topic in a student answer.
Acceptance Criteria	Function correctly generates a list of synonyms based on rulebook provided by professor answer

User Story #	10
Assigned to	Alexander Jimenez, Matias Fuenzalida, Amani Hunt, Ailany Rodriguez
User Story	As a developer, I want to clean up the student topic in order to more accurately compare it to the professor's topic.
Acceptance Criteria	Process student text using various NLP tasks to create a body of text that can be used to compare it to a processed professor topic

User Story #	11
Assigned to	Alexander Jimenez, Amani Hunt, Matias Fuenzalida, Ailany Rodriguez
User Story	As a developer, I want to compare the student topic and the professor topic in order to grade similarity.
Acceptance Criteria	Compare topics found in student answer to topics found in professor answer and outputs a similarity score

User Story #	12
Assigned to	David Llerena, Michael Wei, Timothy King, Jalen Williams
User Story	As a developer, I want to have a list of keywords that accurately describes a topic.
Acceptance Criteria	Function generates a list of keywords provided by the professor answer (removal of stopwords and punctuation included)

User Story #	13
Assigned to	David Llerena, Michael Wei, Timothy King, Jalen Williams
User Story	As a developer, I want to know how many times a topic is referenced in a sentence so that I can label that sentence accordingly.
Acceptance Criteria	Function correctly counts how many times a keyword related to a topic is found in a student sentence

User Story #	14
--------------	----

Assigned to	David Llerena, Michael Wei, Timothy King, Jalen Williams
User Story	As a user, I want to receive feedback on whether or not I included a topic in my answer so that I can learn and improve my skills.
Acceptance Criteria	Results page displays highlighted topics found in student answer

User Story #	15
Assigned to	Malcolm Verdier, Ailany Rodriguez, Amani Hunt, Alexander Jimenez, Matias Fuenzalida
User Story	As a user, I want to read questions so that I can answer them and practice my skills.
Acceptance Criteria	Question and answer page displays question

User Story #	17
Assigned to	Malcolm Verdier, Ailany Rodriguez, Amani Hunt, Alexander Jimenez, Matias Fuenzalida
User Story	As a student, I want to receive feedback on my answers so that I can learn and improve my skills.
Acceptance Criteria	Results page displays feedback in terms of topics spotted and scores

Pending User Stories

User Story #	16
--------------	----

Assigned to	Malcolm Verdier, Ailany Rodriguez, Amani Hunt, Alexander Jimenez, Matias Fuenzalida
User Story	As a user, I want to input answers into boxes so that I can receive feedback and my responses.
Acceptance Criteria	Input box on the question and answer page correctly receives input from student

User Story #	18
Assigned to	David Llerena, Michael Wei, Timothy King, Jalen Williams
User Story	As a student, I want to receive a score on my answer so that I can see my performance.
Acceptance Criteria	Results page displays score of cohesion, structure, similarity and score

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project involved frontend and backend development 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

- Backend was implemented using Python to create the automated essay grading system to grade and give feedback on student essay responses based on a rulebook.
- The Front-End was implemented using React, Node.js and Express.js
- We used GitHub as a DevOp tool to collaborate and share work as we were working on the project.

Sprints Plan

Sprint 1

Sprint 1 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Jalen Williams, Malcolm Verdier, Timothy King

Capstone 1: [Michael Wei](#), Ailany Rodriguez, Matias Fuenzalida

Start time: 12:00 pm

End time: 1:00 pm

1/19/22

As a team we decided to add to hollowing user stories.

1. As a Developer I should be able to know what NLP libraries are most relevant to address the problems faced in our project (40 hours)
2. As a Developer I should be able to know how to identify which subproblem of the NLP domain our work includes(15 hours)
3. As a project member I should be able to know if the project is on track to produce our minimum deliverables for the semester, and what schections are critical(5 Hours)
4. As a developer I should know what subset of the project is immediately relevant for any piece code I'm working on through a breaking down of the functionality into subsystems and modules.(10 hours)

The team members indicated their willingness to work on the following user stories.

- Alexander Jimenez
 - Agreed to contribute to the project and will do everything possible to understand the challenging questions.
 - Work Item #1,#2
- Timothy King
 - Work item # 10
 - Work Item # 1
 - All user stories
- David Llerena
 - All user stories
- Malcolm Verdier
 - Work item # 10
 - Work Item # 1
 - All user stories
- Jalen Williams
 - Work item # 10
 - Work Item # 1
 - All user stories
- Matias Fuenzalida
 - Dedicating time to study topics related to the project.
 - Work Item # 1
- Amani Hunt
 - Work item # 1
 - Work Item # 1
 - All user stories

- Ailany Rodriguez
 - Agreeing to dedicate time to research topics related to the project so that I'm able to contribute in any way possible.
 - Work item # 1
 - Work Item # 3
 - All user stories

Michael Wei

- Agreed to contribute to the project and will do everything possible to understand the challenging questions.
- Work item #1
- Work Item #2
- Work item #3
- Work Item #4

Sprint 2

Sprint 2 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Jalen Williams, Malcolm Verdier, Timothy King

Capstone 1: [Michael Wei](#), Ailany Rodriguez, Matias Fuenzalida

Start time: 9:00 pm

End time: 10:00 pm

1/31/22

As a team we decided to add to following user stories.

1. **<Keyword_Identification>**As a user, I want to identify keywords in an answer based on a rulebook that is specific to the question being asked (15 points)
2. **<Rulebook_Inference>**As an instructor, I want the system to be able to construct an inferred rulebook for student answers based upon the instructor answer **Dependent on:** **<Section_Identification>** , **<Section_Flag_Identification>** (15 points)
3. **<Section_Identification>** As a professor, I wish the program identifies sections in my answer automatically.(5 points)
4. **<Section_Flag_Identification>** As a professor, I wish the program identifies section flags for grading student answers automatically (Transforming the problem classification -> clustering) (5 points)
5. **<Sentence_Tokenization>** As a developer, I want to tokenize student & professor answers into separate sentences and words so that it can later be used by the machine. (5 points)
6. **<Synonym_Identification>** As a developer, I want to be able to use tokenized text to extract meanings and identify keywords plus any synonyms.(WordNet capability?) **Dependent on:** **<Sentence_Tokenization>** (8 points)

The team members indicated their willingness to work on the following user stories.

- Alexander Jimenez
 - #3. <Section_Identification>
 - #4. <Section_Flag_Identification>
- Timothy King
 - #1. <Keyword_Identification>
- David Llerena
 - #1. <Keyword_Identification>
 - #6. <Synonym_Identification>
- Malcolm Verdier
 - #1. <Keyword_Identification>
 - <Synonym_Identification>
- Jalen Williams
 - #1. <Keyword_Identification>
- Matias Fuenzalida
 - #5. <Sentence_Tokenization>
- Amani Hunt
 - #1. <Keyword_Identification>
- Ailany Rodriguez
 - #3. <Section_Identification>
 - #6. <Synonym_Identification>
- Michael Wei
 - #1. <Keyword_Identification>

Sprint 3

Sprint 3 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Jalen Williams, Malcolm Verdier, Timothy King

Capstone 1: [Michael Wei](#), Ailany Rodriguez, Matias Fuenzalida

Start time: 9:30 pm

End time: 10:00 pm

2/14/22

As a team we decided to add to following user stories.

1. **<Keyword_Identification>** As a user, I want to identify keywords in an answer based on a rulebook that is specific to the question being asked (15 points)
2. **<Rulebook>** As an instructor, I want the system to be able to construct a rulebook for student answers based upon the instructor answer **Dependent on:**
<Section_Identification> , **<Section_Flag_Identification>** (15 points)
3. **<Section_Identification>** As a professor, I wish the program identifies sections in my answer automatically.(5 points)
4. **<Section_Flag_Identification>** As a professor, I wish the program identifies section flags for grading student answers automatically (Transforming the problem classification -> clustering) (5 points)
5. **<Synonym_Identification>** As a developer, I want to be able to use tokenized text to extract meanings and identify keywords plus any synonyms.(WordNet capability?) (8 points)

The team members indicated their willingness to work on the following user stories.

- Alexander Jimenez

- #3.<Section_Identification>
- #4.<Section_Flag_Identification>

- Timothy King
 - #1. <Keyword_Identification>
 - #5. <Synonym_Identification>

- David Llerena
 - #1. <Keyword_Identification>
 - #6. <Synonym_Identification>

- Malcolm Verdier
 - <Section_Identification>

- Jalen Williams
 - #1. <Keyword_Identification>

- Matias Fuenzalida
 - #1 <Keyword_Identification>.
 - #5 <Synonym_Identification>

- Amani Hunt
 - #1. <Keyword_Identification>

- Ailany Rodriguez
 - <Section_Identification>
 - <Section_Flag_Identification>

- Michael Wei
 - #1. <Keyword_Identification>

Sprint 4

Sprint 4 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Jalen Williams, Malcolm Verdier, Timothy King
Capstone 1: [Michael Wei](#), Ailany Rodriguez, Matias Fuenzalida, Amani Hunt

Start time: 9:30 pm

End time: 10:00 pm

3/7/22

As a team we decided to add to following user stories.

- 1. Clean up professor text to retrieve keywords**
 - a. As a developer, I want to use keywords to identify the topic in a student answer
- 2. Add topics to relate to keywords**
 - a. As a developer, I want to label blocks of text
- 3. Acquire synonyms of keywords**
 - a. As a developer, I want to use synonyms to identify the topic in a student answer
- 4. Relate synonyms of keywords to topic**
 - a. As a developer, I want to label blocks of text using synonyms
- 5. Clean up student text to prepare for student-professor topic comparison**
 - a. As a developer, I want to clean up student topic in order to more accurately compare it to the professor's topic
- 6. Compare student topic to professor topic**
 - a. As a developer, I want to compare student topic and professor topic in order to grade similarity

The groups indicated their willingness to work on the following user stories.

G1: Malcolm, Timothy, Ailany

- User story #3: Acquire synonyms of keywords
- User story #4: Relate synonyms of keywords to topic

G2: Alex, Matias, Amani

- User story #5: Clean up text to prepare for student-professor topic comparison
- User story #6: Compare student topic to professor topic

G3: David, Michael, Jalen

- User story #1: Clean up professor text to retrieve keywords
- User story #2: Add topics to relate to keywords

Sprint 5

Sprint 5 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: Jalen Williams, [Alexander Jimenez](#), David Llerena, Malcolm Verdier

Capstone 1: [Michael Wei](#), Matias Fuenzalida, Ailany Rodriguez

Start time: 9:30 pm

End time: 10:00 pm

3/21/22

As a team we decided to add the following user stories.

- 1. Clean up professor text to retrieve keywords**
 - a. As a developer, I want to use keywords to identify the topic in a student answer
- 2. Add topics to relate to keywords**
 - a. As a developer, I want to label blocks of text
- 3. Acquire synonyms of keywords**
 - a. As a developer, I want to use synonyms to identify the topic in a student answer
- 4. Clean up student text to prepare for student-professor topic comparison**
 - a. As a developer, I want to clean up student topic in order to more accurately compare it to the professor's topic
- 5. Compare student topic to professor topic**
 - a. As a developer, I want to compare student topic and professor topic in order to grade similarity
- 6. Remove keywords that don't make topics unique**

- a. As a developer, I want to have a list of keywords that accurately describes a topic
- 7. Count number of references to a topic per sentence in student answer**
 - a. As a developer, I want to know how many times a topic is referenced in a sentence so that I can label that sentence accordingly.
- 8. Classify parts of student text**
 - a. As a user, I want to receive feedback on whether or not I included a topic in my answer so that I can learn and improve my skills
- 9. Display questions on webpage**
 - a. As a user, I want to read questions so that I can answer them and practice my skills
- 10. Create input boxes for student and professor answers**
 - a. As a user, I want to input in answers into boxes so that I can receive feedback and my responses
- 11. Display results on webpage**
 - a. As a student, I want to receive feedback on my answers so that I can learn and improve my skills

The groups indicated their willingness to work on the following user stories:

G1: David, Michael, Jalen, Timothy

- User Story 1: Clean up professor text to retrieve keywords
- User Story 2: Add topics to relate to keywords
- User Story 6: Remove keywords that don't make topics unique
- User Story 7: Count number of references to a topic per sentence in student answer
- User Story 8: Classify parts of student text

G2: Alex, Matias, Amani

- User Story 3: Acquire synonyms of keywords
- User Story 4: Clean up student text to prepare for student-professor topic comparison
- User Story 5: Compare student topic to professor topic

G3: Malcolm, Ailany

- User Story 9: Display questions on webpage
- User Story 10: Create input boxes for student and professor answers
- User Story 11: Display results on webpage

Sprint 6

Sprint 6 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: David Llerena, Timothy King, Jalen Williams

Capstone 1: [Michael Wei](#), Amani Hunt, Ailany Rodriguez, Matias Fuenzalida

Start time: 9:30 pm

End time: 10:20 pm

4/4/22

As a team we decided to add the following user stories.

1. Acquire synonyms of keywords

- a. As a developer, I want to use synonyms to identify the topic in a student answer

2. Compare student topic to professor topic

- a. As a developer, I want to compare student topic and professor topic in order to grade similarity

3. Remove keywords that don't make topics unique

- a. As a developer, I want to have a list of keywords that accurately describes a topic

4. Count number of references to a topic per sentence in student answer

- a. As a developer, I want to know how many times a topic is referenced in a sentence so that I can label that sentence accordingly.

5. Classify parts of student text

- a. As a user, I want to receive feedback on whether or not I included a topic in my answer so that I can learn and improve my skills

6. Display questions on webpage

- a. As a user, I want to read questions so that I can answer them and practice my skills
- 7. Create input boxes for student and professor answers**
 - a. As a user, I want to input in answers into boxes so that I can receive feedback and my responses
 - 8. Display results on webpage**
 - a. As a student, I want to receive feedback on my answers so that I can learn and improve my skills

The groups indicated their willingness to work on the following user stories:

G1: David, Michael, Jalen, Timothy

- User Story 3: Remove keywords that don't make topics unique
- User Story 4: Count number of references to a topic per sentence in student answer
- User Story 5: Classify parts of student text

G2: Alex, Matias, Amani, Ailany

- User Story 1: Acquire synonyms of keywords
- User Story 2: Compare student topic to professor topic

G3: Malcolm

- User Story 6: Display questions on webpage
- User Story 7: Create input boxes for student and professor answers
- User Story 8: Display results on webpage

Sprint 7

Sprint 7 Planning Meeting Minutes:

P.O. David Gray

Attendees:

Capstone 2: David Llerena, Timothy King, Jalen Williams

Capstone 1: [Michael Wei](#), Amani Hunt, Ailany Rodriguez

Start time: 9:30 pm

End time: 10:20 pm

4/18/22

As a team we decided to add the following user stories.

1. Acquire synonyms of keywords

- a. As a developer, I want to use synonyms to identify the topic in a student answer

2. Compare student topic to professor topic

- a. As a developer, I want to compare student topic and professor topic in order to grade similarity

3. Display questions on webpage

- a. As a user, I want to read questions so that I can answer them and practice my skills

4. Create input boxes for student and professor answers

- a. As a user, I want to input in answers into boxes so that I can receive feedback and my responses

5. Display results on webpage

- a. As a student, I want to receive feedback on my answers so that I can learn and improve my skills

6. Scoring System

- a. As a student, I want to receive a score on my answer so that I can see my performance.

The groups indicated their willingness to work on the following user stories:

G1: David, Michael, Jalen, Timothy

- User Story 6: Scoring System
- User Story 2: Compare student topic to professor topic

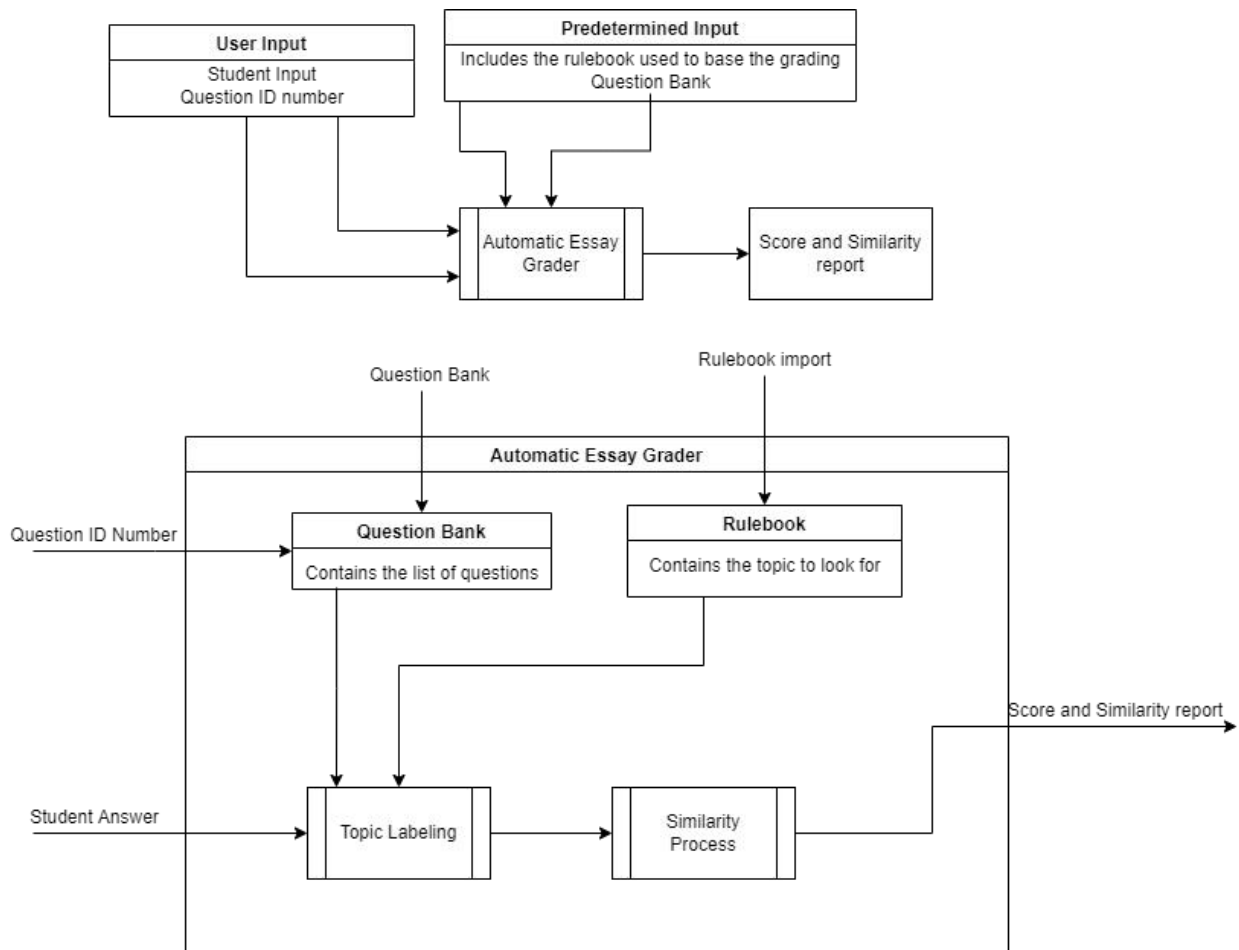
G2: Alex, Matias, Amani, Ailany, Malcolm

- User Story 3: Display questions on webpage
- User Story 4: Create input boxes for student and professor answers
- User Story 5: Display results on webpage

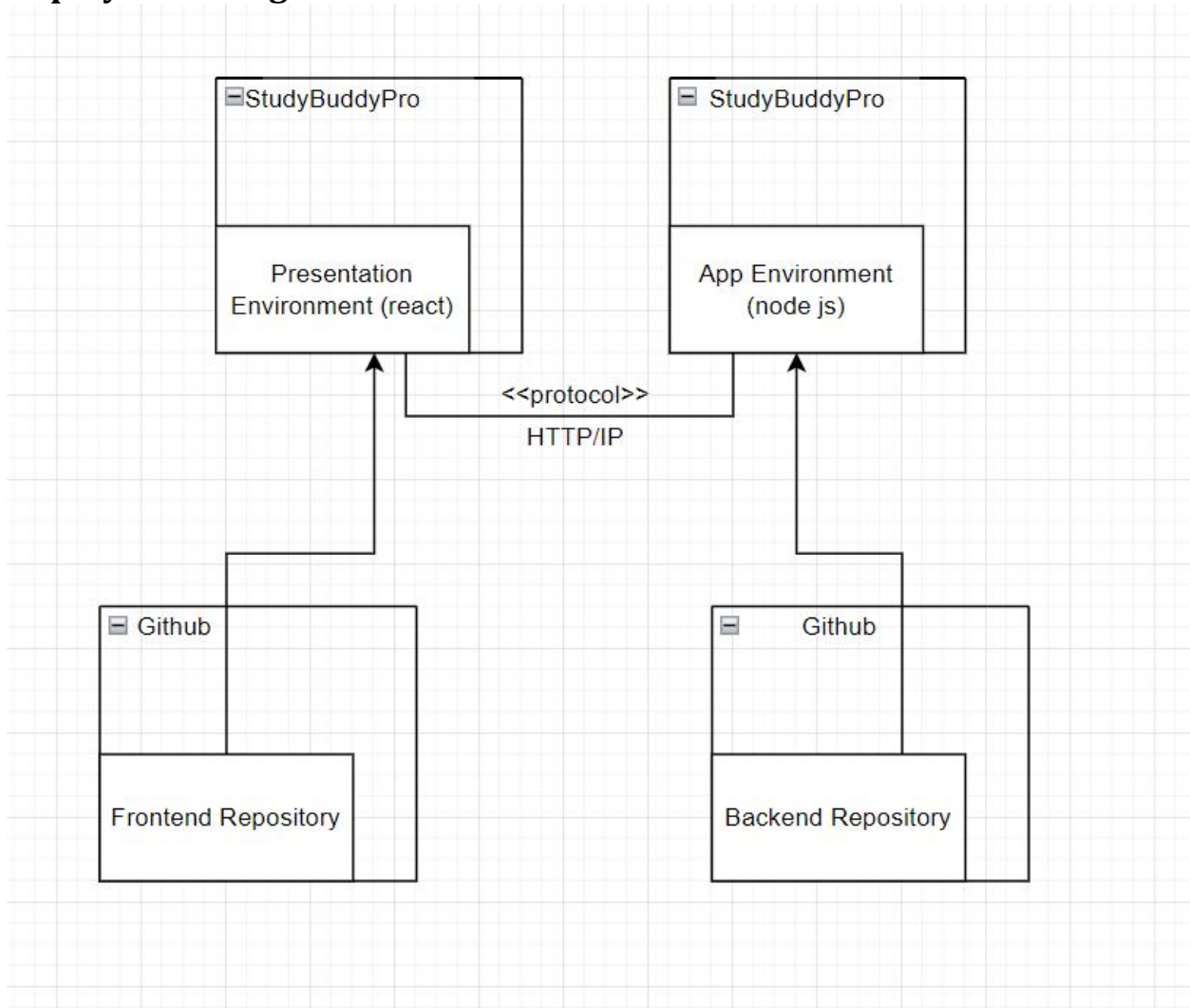
SYSTEM DESIGN

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.

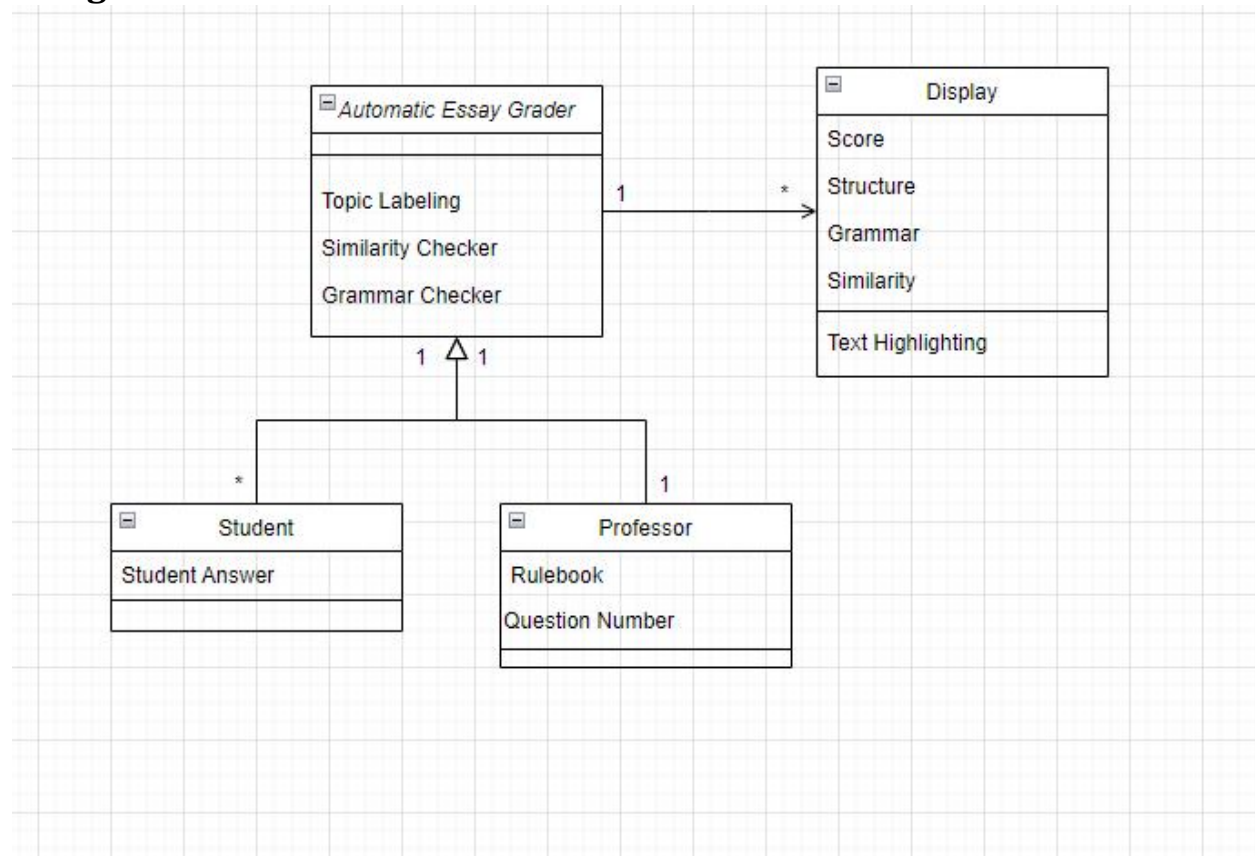
System and Subsystem Decomposition



Deployment Diagram



Design Patterns



SYSTEM VALIDATION

Testing was done frequently throughout the entire development process. On the back end, unit testing was done almost immediately after any function had been implemented to determine whether the functions were behaving as we had intended. These tests helped verify the correctness of the program, and set boundaries/parameters for each of the functions. For specific subsystems, like the similarity functions, dynamic analysis was done in the form of white-box and data structure-based testing to optimize the function to meet quality standards. For imported subsystems, like the grammatical analysis system, black box testing was done to check the effectiveness of the function's interactions with the other subsystems. During the week before the showcase, integration and system testing was conducted on the entire system that was created during the 6 prior sprints. These tests helped assure that our subsystems coalesced into a working product that could be demoed at the showcase.

GLOSSARY

Python: Programming language used to implement the backend of the project

SpaCy: A Python library for NLP

Language Tool: Library used for grammar checking

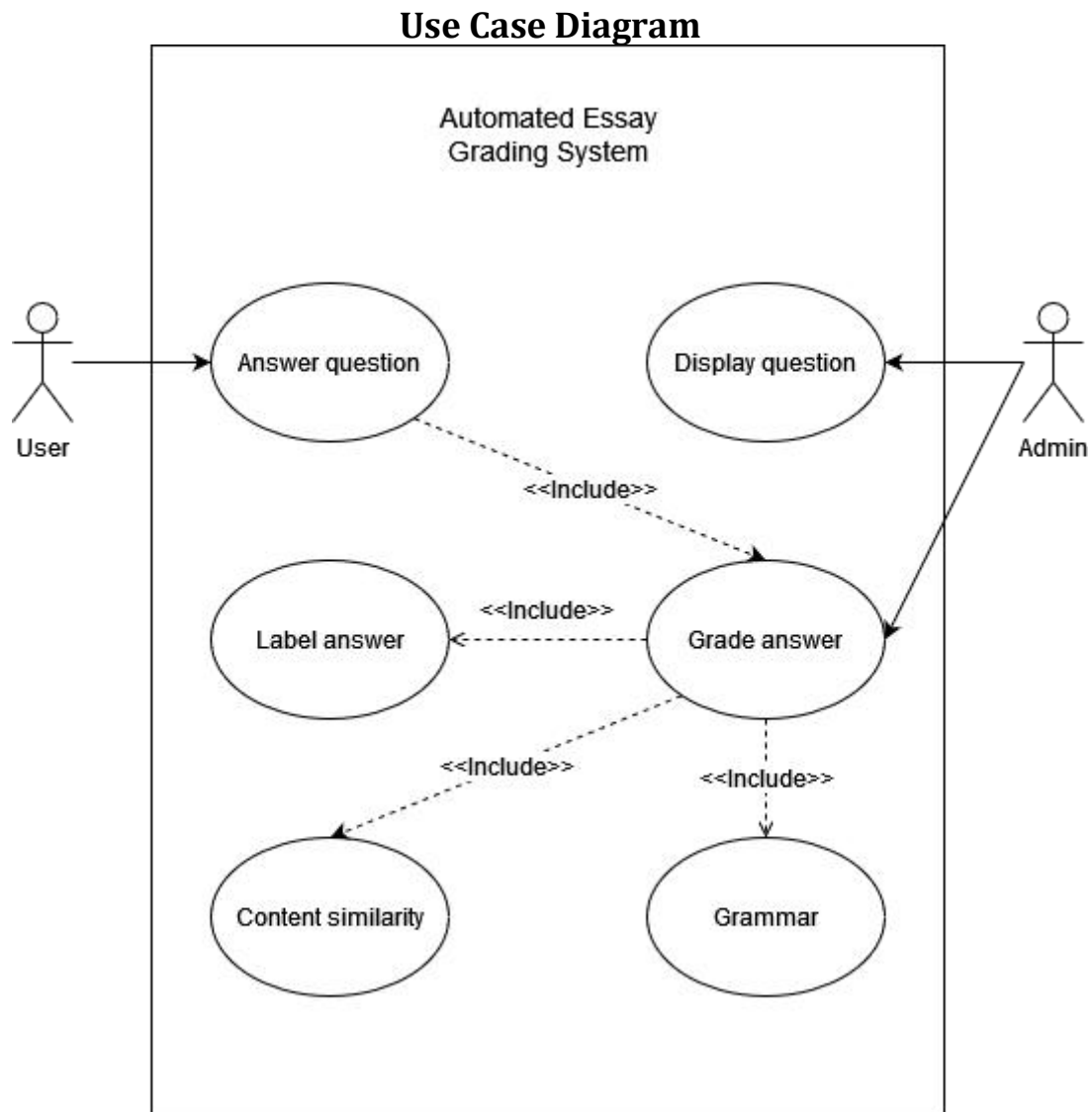
Node.js: Frontend tech used for developing web pages

GitHub: Version Control and Collaboration tool used for improving efficiency of teamwork

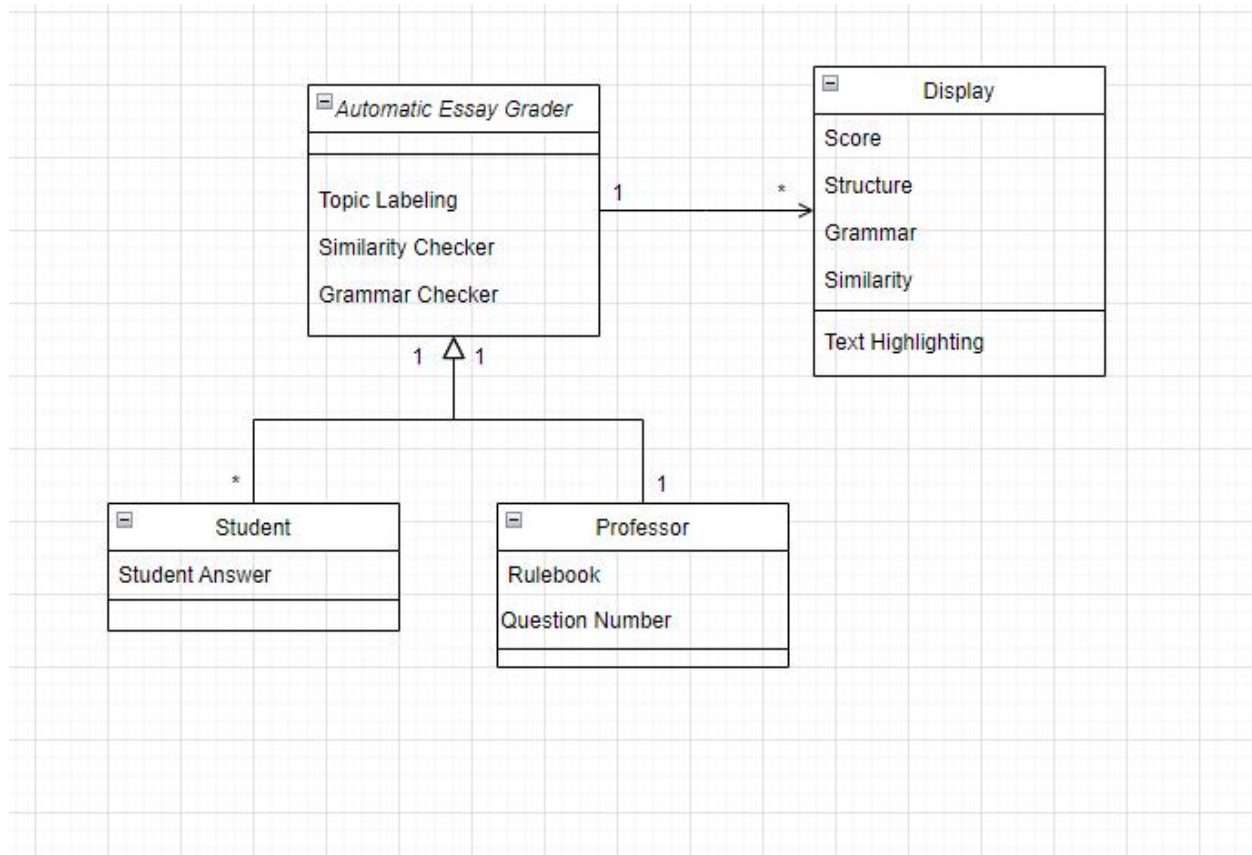
TF-IDF: Term Frequency-Inverse Document Frequency can determine the relevancy of a word in a document

Stopwords: Commonly used in a language that do not add any significant meaning to the text

Natural Language Processing: A study of computer science that that is concerned with the machines ability to understand the human language

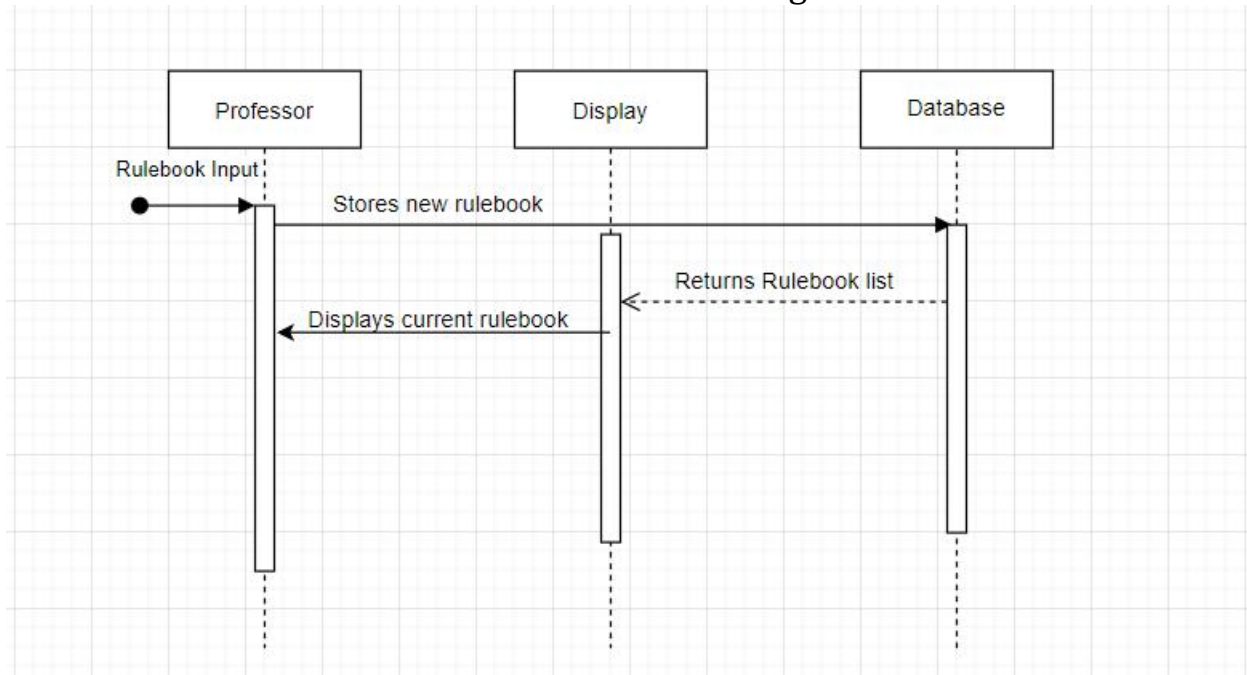
APPENDIX**Appendix A - UML Diagrams**

Class Diagrams

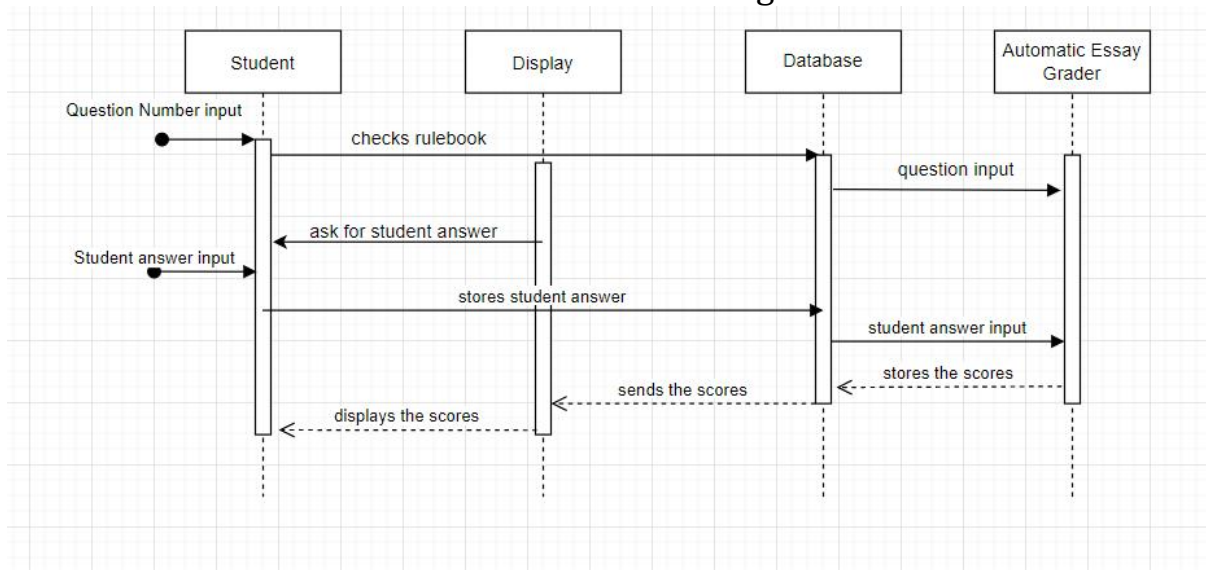


Sequence Diagrams

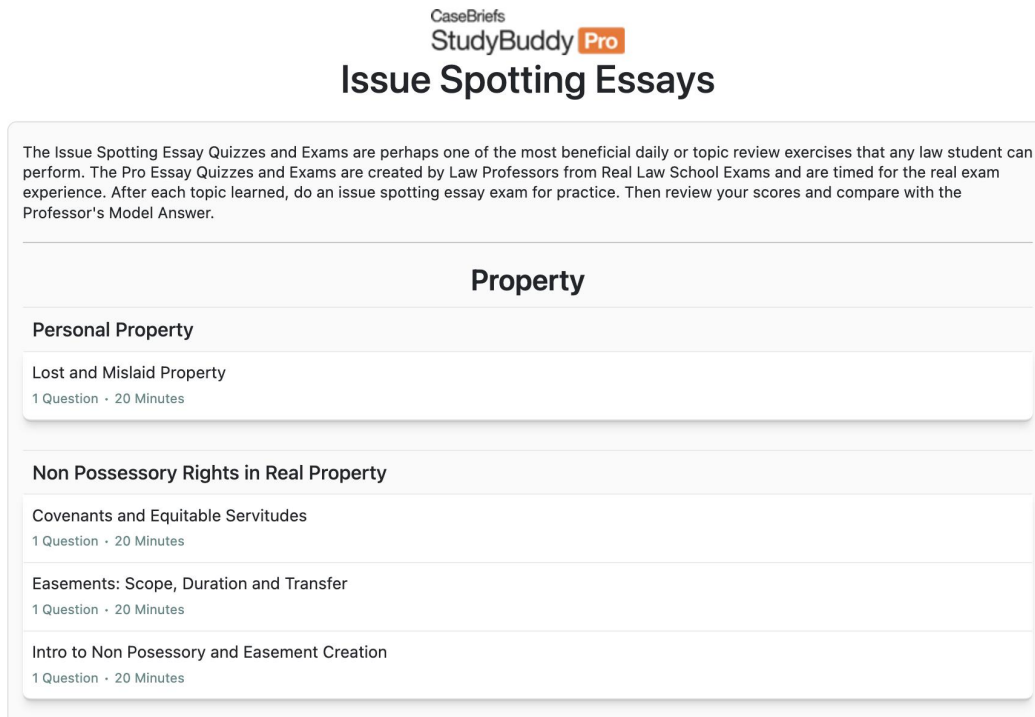
Professor Interaction Diagram



Student Interaction Diagram



Appendix B - User Interface Design



Question Selection Screen

Question One •

Daniel and Mike owned adjacent residences, each on its own lot, in Mowtown. Both lots bordered side and back on other lots, and on the front, to the north, by Jazz Road. Both men had recently retired. Daniel rented his house to a young family and moved away to live with his son in Citytown. Mike continued to live in his home. After about six months, Mike decided to convert his garage to a den. He built a carport at the rear of his house and laid an asphalt driveway down the side of the house to the carport. The driveway encroached five feet onto Daniel's lot all along its length. The renters didn't appear to notice anything, and Daniel never visited his old home. Seven years later, Daniel was in Mowtown for a company reunion and drove by his old house. He noticed the asphalt driveway and was sure it encroached on his property. A survey documented the encroachment, and Daniel brought an appropriate action to cause Mike to remove the encroaching portion of the driveway. In a jurisdiction whose statutory period for adverse possession is five years, what will be the result of the action brought by Daniel?

19m 38s

Scan My Answer

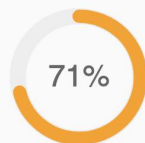
Quiz Screen

Results

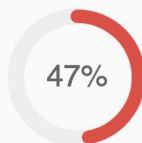
Your Answer

Mike will likely prevail, be allowed to keep his driveway, and be awarded title of the five feet strip of land in which it encroached. His five foot encroachment along Daniel's lot meets the elements to prevail on a adverse possession claim. Namely, his use of the land was open and notorious; and in his continuous and exclusive exclusive use for the statutory period. Further, it was hostile, as he did not have Daniel's permission and used the land as if it were his own. While a majority of jurisdictions do not care about Daniel's mindset, it is not of importance as the fact pattern suggests Mike acted in good faith, and treated the five feet plot as his own.

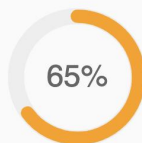
Issue Rule Policy Inference Conclusion



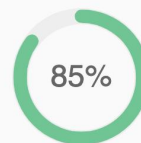
Similarity



Coherence



Structure



Grammar

Professor Answer

At issue is whether Mike obtained rights over the five feet of driveway that encroached onto Daniel's property. As Daniel did not grant Mike any right to use this stretch of property, Mike can obtain rights to this tract of property by prescriptive easement. A prescriptive easement is created when a person adversely and continuously uses a tract of the true owner's land for the requisite statutory period. Paving the five-foot tract of Daniel's land was an open and notorious use of that land which could have put Daniel on notice had he inspected his property. Mike's use of the five-foot tract was also hostile, as he did not have permission from Daniel, and actual and continuous as he paved the land and used it as part of his driveway for the statutory period of time required to acquire rights to the five-foot tract of land. Although Daniel had tenants living on his land when Mike paved it, it was up to Daniel to tend to his land and see that no one encroached upon it. As Daniel did not bring an action until after seven years had passed since Mike paved the tract of land, Daniel will lose and Mike will acquire an easement by prescription.

Issue Rule Policy Inference Conclusion

Results Page

Appendix C - Sprint Review Reports

Sprint 1 Review Meeting

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Timothy King, Malcolm Verdier, Jalen Williams

Capstone 1: Matias Fuenzalida, Ailany Rodriguez, [Michael Wei](#), Amani Hunt

Date: Sunday, January 30, 2022

Start time: 9:00 PM

End time: 9:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: As a developer, I want to be able to understand the project proposal and wishlist, as well as what has already been done by past teams so that I understand the project I am working on. (6 hours)
- User Story 2: As a developer, I want access to the Github repositories and documentation needed to communicate with my team members so that my team and I are informed on the progress of each team member. (1 hour)
- User Story 4: As a developer, I want to begin to view/review the each libraries needed to complete the project and learn how to use them in our project. (6 hours)
- User Story 6: As a developer, I want to share the previous code of the last team with our current team through either giving them access to the zip or (0.5 hours)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting for next semester:

- None

Sprint 2 Review Meeting

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Timothy King, Malcolm Verdier, Jalen Williams

Capstone 1: Matias Fuenzalida, Ailany Rodriguez, Michael Wei, Amani Hunt

Date: February 11, 2022

Start time: 9:30 PM

End time: 10:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- 1. <**Rulebook_Inference**>As an instructor, I want the system to be able to construct an inferred rulebook for student answers based upon the instructor answer
- 2. <**Keyword_Identification**>As a user, I want to identify keywords in an answer based on a rulebook that is specific to the question being asked
- 1. <**Feedback**>As a student, I should be able to receive feedback on any sections that are not answered correctly or are missing.
- 2. <**Grammar_Check**>As a student, I should be able to receive feedback on any grammatical errors and possible solutions to the error.
- 3. <**LearnDash_Usability**>As an instructor, I should be able to upload questions to LearnDash for students to practice.
- 4. <**LearnDash_Integration**>As a developer, I should be able to use the LearnDash documentation to implement our model into the working system.
- 5. <**Synonym_Identification**> As a developer, I want to be able to use tokenized text to extract meanings and identify keywords plus any synonyms.

- 6. **<Section_Identification>** As a professor, I wish the program identifies sections in my answer automatically.(5 points)
- 7. **<Section_Flag_Identification>** As a professor, I wish the program identifies section flags for grading student answers automatically (Transforming the problem classification -> clustering) (5 points)

Sprint 3 Review Meeting

Attendees:

Capstone 2: Alexander Jimenez, David Llerena, Timothy King, Malcolm Verdier, Jalen Williams

Capstone 1: Matias Fuenzalida, Ailany Rodriguez, Michael Wei, Amani Hunt

Date: February 27, 2022

Start time: 9:30 PM

End time: 10:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

1. Produce NLP pipeline (spaCy pipeline)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

1. Clean up professor text to retrieve keywords

- a. As a developer, I want to use keywords to identify the topic in a student answer

2. Add topics to relate to keywords

- a. As a developer, I want to label blocks of text

3. Acquire synonyms of keywords

- a. As a developer, I want to use synonyms to identify the topic in a student answer

4. Relate synonyms of keywords to topic

- a. As a developer, I want to label blocks of text using synonyms

5. Clean up student text to prepare for student-professor topic comparison

- a. As a developer, I want to clean up student topic in order to more accurately compare it to the professor's topic

6. Compare student topic to professor topic

- a. As a developer, I want to compare student topic and professor topic in order to grade similarity

Sprint 4 Review Meeting

Attendees:

Capstone 2: David Llerena, Timothy King, Jalen Williams, Malcolm Verdier

Capstone 1: Michael Wei, Matias Fuenzalidaalida, Ailany Rodriguez

Date: March 20, 2022

Start time: 9:30 PM

End time: 10:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

1. **Clean up professor text to retrieve keywords**
 - a. As a developer, I want to use keywords to identify the topic in a student answer
2. **Add topics to relate to keywords**
 - a. As a developer, I want to label blocks of text
3. **Acquire synonyms of keywords**
 - a. As a developer, I want to use synonyms to identify the topic in a student answer
4. **Clean up student text to prepare for student-professor topic comparison**
 - a. As a developer, I want to clean up student topic in order to more accurately compare it to the professor's topic
5. **Compare student topic to professor topic**
 - a. As a developer, I want to compare student topic and professor topic in order to grade similarity
6. **Remove keywords that don't make topics unique**
 - a. As a developer, I want to have a list of keywords that accurately describes a topic
7. **Count number of references to a topic per sentence in student answer**

- a. As a developer, I want to know how many times a topic is referenced in a sentence so that I can label that sentence accordingly.

Sprint 5 Review Meeting

Attendees:

Capstone 2: David Llerena, Jalen Williams, Timothy King

Capstone 1: Michael Wei, Matias Fuenzalida, Ailany Rodriguez, Amani Hunt

Date: April 3, 2022

Start time: 9:30 PM

End time: 10:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- 1. Clean up professor text to retrieve keywords**
 - a. As a developer, I want to use keywords to identify the topic in a student answer
- 2. Add topics to relate to keywords**
 - a. As a developer, I want to label blocks of text
- 3. Acquire synonyms of keywords**
 - a. As a developer, I want to use synonyms to identify the topic in a student answer
- 4. Clean up student text to prepare for student-professor topic comparison**
 - a. As a developer, I want to clean up student topic in order to more accurately compare it to the professor's topic
- 5. Compare student topic to professor topic**
 - a. As a developer, I want to compare student topic and professor topic in order to grade similarity
- 6. Remove keywords that don't make topics unique**
 - a. As a developer, I want to have a list of keywords that accurately describes a topic
- 7. Count number of references to a topic per sentence in student answer**

- a. As a developer, I want to know how many times a topic is referenced in a sentence so that I can label that sentence accordingly.
- 8. Classify parts of student text**
 - a. As a user, I want to receive feedback on whether or not I included a topic in my answer so that I can learn and improve my skills
- 9. Display questions on webpage**
 - a. As a user, I want to read questions so that I can answer them and practice my skills
- 10. Create input boxes for student and professor answers**
 - a. As a user, I want to input in answers into boxes so that I can receive feedback and my responses
- 11. Display results on webpage**
 - a. As a student, I want to receive feedback on my answers so that I can learn and improve my skills

Sprint 6 Review Meeting

Attendees:

Capstone 2: David Llerena, Jalen Williams, Timothy King

Capstone 1: Michael Wei, Matias Fuenzalida, Ailany Rodriguez

Date: April 17, 2022

Start time: 9:30 PM

End time: 10:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

1. **Classify parts of student text**
 - a. As a user, I want to receive feedback on whether or not I included a topic in my answer so that I can learn and improve my skills

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

1. **Acquire synonyms of keywords:** As a developer, I want to use synonyms to identify the topic in a student answer
2. **Compare student topic to professor topic:** As a developer, I want to compare student topic and professor topic in order to grade similarity
3. **Display questions on webpage:** As a user, I want to read questions so that I can answer them and practice my skills
4. **Create input boxes for student and professor answers:** As a user, I want to input in answers into boxes so that I can receive feedback and my responses
5. **Display results on webpage:** As a student, I want to receive feedback on my answers so that I can learn and improve my skills
6. **Scoring System:** As a student, I want to receive a score on my answer so that I can see my performance.

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

Installation/Maintenance guide:

1. Development Environment

Visual Studio Code: (<https://code.visualstudio.com/>)

Github Account: (<https://github.com/>)

Git/Git bash: for pushing changes to github repositories

Node.js: for npm commands (<https://nodejs.org/>)

2. Access to Project Source Code:

GitHub: (<https://github.com/alejim/AutomatedEssayGrading>)

3. Getting Project Dependencies:

Backend:

Download and install Python.

If you have created a PATH System variable for your python interpreter, create a virtual environment using the following command:

python -m venv [name_of_virtual_environment]

This is highly recommended in order to avoid potential conflicts when working on multiple projects. After you have created your virtual environment, activate or deactivate your virtual environment with the following commands:

[Path to activate file]\activate

[Path to deactivate file]\deactivate

Once you have activated your virtual environments, all the following modules will be installed in the python library of your virtual environment. Now you are finally ready to install the dependencies required for this project.

First, you will install the NLP module spaCy with the large model used for various NLP tasks by typing in the following commands:

pip install -U pip setuptools wheel

```
pip install -U spacy  
python-m spacy download en_core_web_lg
```

After you have installed spaCy, you will now be able to handle a variety of NLP related tasks. Next, you will install a grammar checking module called Language Tool with the following command:

```
pip install language-tool-python
```

You should now be able to grammar check all kinds of input text. Next, if your Python installation does not already come packaged with NumPy, install it with the following command:

```
pip install numpy
```

This will allow you to run the synonyms function that retrieves a list of synonyms for a given word. Finally, install scikit-learn. If you are currently using the Anaconda distribution of Python, scikit-learn comes included in the package, otherwise execute the following command:

```
pip install -U scikit-learn
```

This should allow you to run the similarity function that can determine how similar one document is from another document. Now you have installed all the backend dependencies required and you are ready to begin contributing to the project.

To View the Website:

Download and install Node.js from the official website.

Once this is done, go to the root directory and run the following command to install all of the dependencies:

```
npm install && cd frontend && npm install && cd .. && cd backend && npm install
```

After this is done, type in the following command so that you can run a local server that will host your website:

```
node server.js
```

After this is completed you will finally be able to view the website by going to your web browser and typing in the following:

```
localhost:3001
```

WISHLIST FOR AUTOMATED ESSAY GRADING

- 1. Improve accuracy of rule-based text classification system
- 2. Improve accuracy of similarity scoring system
- 3. Implement cohesion and structure scoring system
- 4. Utilize statistical or machine learning techniques to better extrapolate semantic context from shorter texts.

REFERENCES

Ramesh et al., An automated essay scoring systems: a systematic literature review, Artif Intell Rev. 2021 Sep 23 : 1–33.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8460059/>

Harshanthi Ghanta, Automated Essay Evaluation Using Natural Language Processing and Machine Learning, 2019, Theses and Dissertations, Columbus State University.

https://csuepress.columbusstate.edu/cgi/viewcontent.cgi?article=1330&context=theses_dissertations

Zhang et al., eRevise: Using Natural Language Processing to Provide Formative Feedback on Text Evidence Usage in Student Writing, 2019, The Thirty-First AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI-19).

<https://files.eric.ed.gov/fulltext/ED598633.pdf>

<https://monkeylearn.com/guides/>

<https://www.youtube.com/watch?v=dIUTsFT2MeQ>