ESOF 423
Software Engineering Applications
Spring 2022

By

Cam Alberghini
Chris Rosler
Rory McLean

Program:
The link to our Electronic Dissertation Validator is: https://github.com/423s22/G2.

Teamwork:
Cam Alberghini, Chris Rosler, and Rory McLean will be referenced as member 1, member 2, and member 3 respectively. Member 3 contributed the web format for the validator. Members 1 and 2 both contributed to the tokenizing and parsing of the validator.

As of 4/15/2022:
Member 3 has copied the web format from MSU's graduate dissertation website to use as the base formatting in the dissertation validator website. This has involved getting all html and css files made as well as acquiring all of the dependencies that the website needed to look professional, such as website logos. Member 3 also acquired the example dissertation file that was used in the making of the parser. Most of the backlog elements filled by member 3 involved work on the website visuals, either updating existing pages or formatting new ones. The time estimated to complete the tasks for member 3 were above what the actual turned out to be.

Members 1 and 2 have extended the parser to hold all relevant data. The data is compared to a series of rules that determine if the paper follows the formatting for the introduction. Following a comment from the subject matter expert, an issue with page number formatting prevented the parsing of the index. The body of the paper is then searched for issues which are then output to a document which is displayed for the user to identify.

Member 1 worked with expanding compatibility. Different Microsoft Word formats have different tags and xml attributes that had to be handled differently.

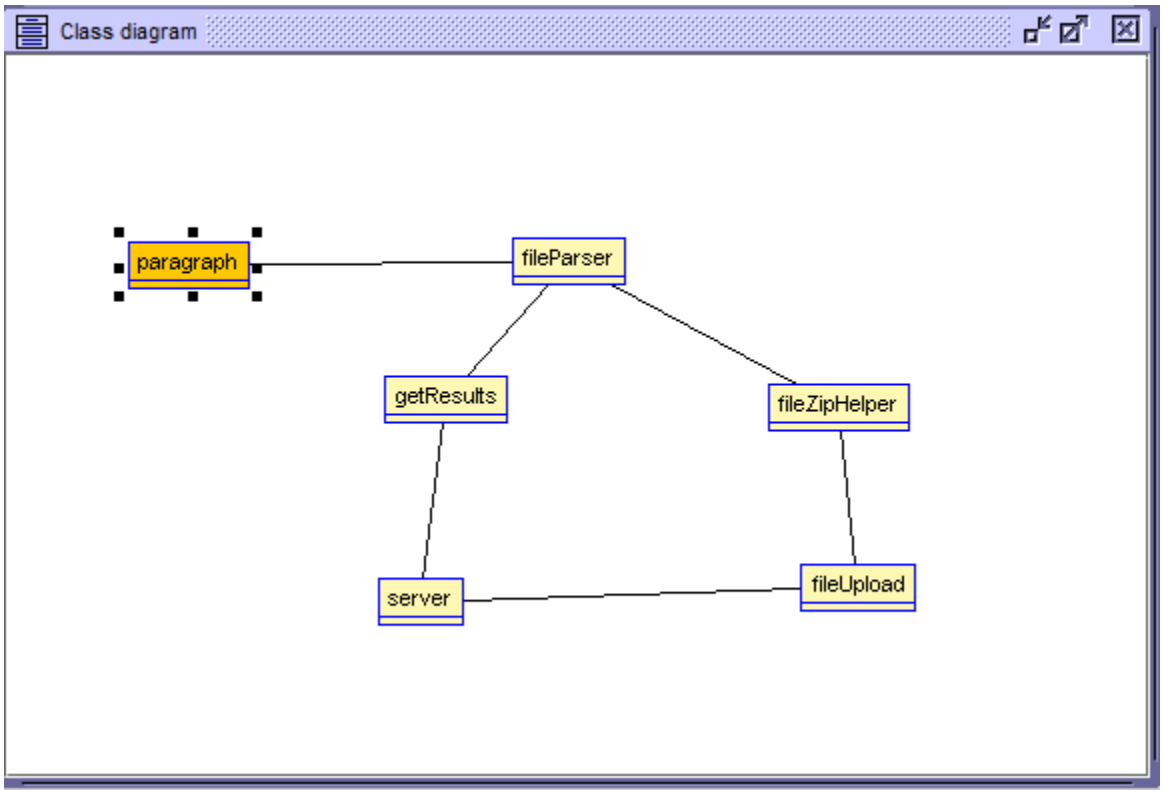
Member 2 expanded the parsing capabilities including creating a structure to hold all relevant information that can be taken from the xml documents. The text contained in the xml is then converted to a readable format for a spell checker and then returns the words that are detected as a typo.

All members contributed to the process of linking the back end parser to the front end UI and website.

Design Pattern:
We plan on using the decorator pattern for this project and it will help us when using classifying and separating xml tags during the tokenizing and parsing step of validation. The reason why we are planning this is due to the xml tags having similar base states but small classifications that separate them from each other. If we are to use a decorator pattern, we should have a much easier time when parsing the xml.

Technical Writing:
This project is a flask based web application that processes documents. The document is sent to a parser that returns a text document that is displayed with the results of comparing the paper with the guidelines. Simply clicking on the validation portal button on the index page will direct the user to the content. After that a placeholder password page exists, to be replaced by official university user validation. Then the user uploads a document and is returned a list of errors.

UML:
The simple overall structure



Design Trade-Offs:
The largest design trade-off that we had to make was to exactly follow what the clients wanted. Functionally, if the user only wanted to get a PDF that exactly followed the guidelines after providing a Microsoft Word Docx file, the best solution would have been to take in all of the user's data, then to convert the document to the latex format which could then be saved to the PDF format. The result of this is that the software was less perfect and allowed users to make mistakes. It took more time to develop this software, and it is less effective, but it does follow the clients criteria.

Software Development Life Cycle Model:

For this project, we are using an agile development life cycle for our life cycle model. This development cycle worked well and we kept on track with the bi-weekly meetings and project reviews.