UNIVERSITY OF PORTSMOUTH

Automated Assignment Scoring Via Azure OpenAI ChatGPT

By: Callum Fry

Project Unit: Your Project Unit Code Here!
Supervisor: Mani Ghahremani

Submission date:
7 May 2024

1 Introduction

1.1 Context of Research

Machine Learning (ML) is an evolving branch of Artificial Intelligence (AI) that uses computational algorithms aimed to imitate human intelligence by using data from its environment to learn and improve (El Naqa et al., 2015). In more recent years, machine learning has become more widely used; this has been driven by the development of new theories and practical applications. As well as the rapid increase in accessible online data and affordable computing power (M., 2015).

In this landscape, cloud platforms like Microsoft Azure have emerged as powerful tools for deploying AI models. Specifically, Microsoft Azure is a public cloud computing platform developed by Microsoft (Borra, 2024). This product provides a comprehensive range of services but the most relevant for this project is Azure AI Studio, this provides developers with the ability to create an AI model and deploy it in a secure environment.

In the United Kingdom, most universities currently mark dissertations by selecting two members of staff and allowing them to mark the assignment without discussing it. However, this method may be outdated, and replacing one marker for an AI could be possible, or adding a third marker in the form of AI.

Aiding lectures with this process is likely to have numerous benefits, because markers experience stress, anxiety, and other mental health challenges during periods of heavy assignment marking (Henderson-Brooks, n.d.). The benefits include saving time and resources, improved feedback, more efficient marking, and removal of human bias.

These benefits allow markers more time to improve the quality of their teaching while allowing students to review feedback to further enhance their skills for future work.

1.2 Project Aim

This paper aims to enhance both the quality and speed of assignment marking by implementing automation tools for grading assignments. By implementing AI and machine learning models using cloud solutions. Reducing the time and effort required for manual grading, the proposed solution seeks to relieve the workload on lecturers while improving feedback for students.

1.3 Project Objectives

The objectives of this project are:

1. Gain an understanding of the current research in this area, and existing tools to produce a literature review.

2. Meeting staff to gather requirements, therefore determining what the project needs to achieve. Interviewing lecturers to analyse the requirements to be able to prioritise which features are necessary and which are not.

3. Developing architecture design and sequence diagrams to be able to visualise a design that can be implemented.

4. Implement an accurate and efficient ML model that can mark assignments and produce grades. Based on an input of a marking scheme and student assignments.

5. Evaluate the effectiveness of the model by comparing the results produced to already marked assignments.

2 Literature Review

2.1 Research Background, Context, and Definitions

To fully understand this chapter, some meanings are defined here:

Artificial Intelligence (AI): computer systems that are capable of completing tasks that would require human intelligence (McCarthy, 2004).

Machine Learning (ML): a subset of AI that focuses on developing algorithms to enable computers to learn from their environment and datasets (El Naqa et al., 2015).

Dataset: a collection of data that can be either structured or unstructured, used for training or testing ML models. These datasets can be imported into Azure ML studio and used through the implementation of the model.

Hyperparameters: these are configurable parameters to be able to guide the learning of a model. An example of this could be the learning rate of a neural network, allowing for better optimization (Arnold et al., n.d.).

This literature review aims to research existing knowledge, solutions, and developments in this area. Critical analysis and investigations of already existing solutions to similar tasks will be completed to understand the most appropriate approach.

This area of research is important at this time because the trends of machine learning are starting to evolve in the education sector, a main sector of this is the grading system. By leveraging machine learning, the grading system can be reframed to encourage a more efficient and accurate system (Jalil et al., 2019).

Looking deeper into saving time, some initial research was completed on comparing an automated system against an instructor. Grading a select number of students took the instructor four hours and eleven minutes, averaging four minutes and ten seconds per student. In comparison, the automated system took fourteen seconds to mark the same number of students (Bian et al., 2020). Therefore, the educational system is starting to show promising applications for machine learning, which has increased the trend of research into this area.

There are three main sections of research for this field being, machine learning, artificial intelligence and applications for machine learning in the education sector. Research and popularity into artificial intelligence has increased significantly, since 2013 the share of research papers with titles or abstracts that mention AI or ML has increased from 10% to 27% (Van Noorden & Perkel, 2023).
Some initial research was completed into the application of using a ChatGPT model through Azure to be able to grade students' assignments. This preliminary solution was found to mark 70 students' long essay assignments in ten minutes, whereas previously it took a few hours to manually mark.

Another added benefit found was they had more time to improve the quality of their teaching instead of spending this time on marking the assignments. They spent more time planning, learning and developing their skills for their upcoming lessons.

2.2 Similar Works

- Automation of checking student assignments in IT-related subjects based on AI systems (Sharyhin & Klochko, 2024)
- Fine-tuning ChatGPT for automatic scoring - ScienceDirect
- Automated assignment grading using Azure & ChatGPT - Azure OpenAI, Chat GPT, Automated Assignment Scoring

2.2.1 Automation of checking students' assignments in IT

(Sharyhin & Klochko, 2024) uses AI systems to analyse the time complexity of code segments submitted by students. It discusses methods for submissions, reviewing and feedback to support teachers with large amounts of grading.

The strengths of their project are efficiency, immediate feedback, scalability, and adaptability. The automation can efficiently check solutions, providing relevant feedback to students on if they need to improve their code or if it is correct. The models used also make it scalable and adaptable, with abilities of understanding multiple programming languages and customizable by the teachers to meet certain requirements.

However, the downsides are initial setup complexity, overreliance, consistency, and correctness. To initially configure the layers and models takes a large amount of time and investment, this may put some users off due to the time commitment. Another issue that occurs with AI is overreliance, they become so focused on system-specific optimization that they become very strict and forget about real-world problem-solving. The consistency and correctness of the results are not fully accurate as shown in Figure 1 the models occasionally fail to produce the correct results which will result in correct submissions being rejected.

Results of time complexity assessment of selected Java code fragments for different AI systems.

Java code fragments	Time Complexity.ai	Chatsonic	Actual
$16 : 3Sum$	n^2	$n \cdot log(n)$	n^2
$18 : 4Sum$	n^3	$log(n) \cdot n^3$	n^3
$22 : GenerateParentheses$	$2^n \cdot n$	$2^n \cdot n$	$2^{2n} \cdot n$
$46 : Permutations$	$n!$	$n \cdot n!$	$n!$
$109 : ConvertSortedList$	n	$n \cdot log(n)$	$n \cdot log(n)$
$220 : ConvertDupli$	$n \cdot log(k)$	$n \cdot log(k)$	$n \cdot log(min(k, n))$

Figure 1. Time Complexity of Java code: Showing two models compared to the actual answer (Sharyhin & Klochko, 2024).

2.2.1.1 Features

In an automated evaluation, the project determines the time complexity of a code segment, if this meets the requirements it allows for the student to submit. If the code does not meet the optimal solution, then the student may try again until they provide the correct solution.

In feedback generation, the AI provides constructive feedback, especially if the optimal solution is not met. It provides messages when calculating the time complexity, showing students where the complexity is derived from. The AI could also provide suggestions to improve efficiency.

Customizability, the project can understand different programming languages, which allows many assignments to be checked. The learning objectives can be adapted depending on the time complexity that needs to be met for different coding segments.

2.2.1.2 Architecture

The project contains four main layers of architecture, inputs, analysis, feedback, and customizability. These four layers represent different parts of the project which can be adjusted and improved for different situations.

The input layer is where students submit their code segments, this interface simply passes the input to the analysis layer. The analysis layer uses AI models such as ChatGPT or Time Complexity.ai to evaluate the solution submitted. The feedback layer takes the results from the AI models, based on the optimal solution and detected errors, and produces responses to the student whether positive or negative. The customizability layer is a separate layer in which teachers can provide criteria and parameters for the students' programs to meet.

2.2.2 Automated assignment grading using Azure & ChatGPT

Their project researches how AI can streamline the grading process for lectures using the Azure OpenAI environment and ChatGPT. Specifically grading essays submitted by students in Microsoft Word and Adobe PDF, the models provide a grade, feedback, marks and potential plagiarism from copying or generative AI.

The strengths of their project are efficiency, scalability, customisation, and feedback. The model can grade 70 students' essays in under ten minutes, costing only $0.5 for Azure’s cloud computing services. The project is scalable because it can review multiple essays simultaneously while being customisable, so the lectures can adjust the marking scheme on which the essays are graded.

The data privacy concerns are raised because there is not enough cleaning of the data, this will be learnt to improve the upcoming projects. Due to assignments containing private information such as full names, student numbers and possibly more, they are not necessarily

needed for the AI to grade, so such should be cleaned. This takes teachers time to identify and remove these unnecessary pieces of data.

2.2.2.1 Features

This AI model produces very clear and informative feedback, using a different technique in comparison to other projects. The feedback is created by using prompt engineering techniques, which tell the AI to create a message by acting. For example, proving the prompt “Take on the role of a teacher by assigning marks and providing constructive comments for a writing assignment” to the AI. Many of these prompts are inputted to provide stricter outputs with a higher probability of success.

A main feature is using Azure's cloud services, which provide many advantages over a locally based system, such as security, integration, reliability, and customisation of models. Azure's security and reliability mean that uptime and robust support will be optimized, while the infrastructure of Azure provides more security to the data being processed.

Integrating projects with Azure is a good idea because it allows for feature improvements to be made with other Azure products that help with storage, monitoring, or analytics. The analytics would prove how effective the system is to be able to upgrade the model or for proof of product.

2.2.2.2 Architecture

The architecture flow of their project is: collecting the inputs (assignments and marking scheme), preprocessing data, model review, feedback creation, and output delivery.

The input collection is completed manually, which is a weakness of their project, as spoken about before. The data is then preprocessed, this is to ensure that the essays are in the correct file type and in the correct folder structure to be processed by the model. This will help to remove errors since many students submit essays in different ways.

The model then reviews the essays submitted, produces feedback from them and creates an output for the lecture to be checked by a human and sent to students. Azure AI Studio provides a playground for users to check the performance of a model before deploying it. Using different evaluation tools to review the model and apply a rubric-based scoring.

The feedback is then produced by using prompt engineering to provide personalised constructive feedback. Just like natural human feedback, the tailored responses are designed to help students see where they did well and areas of improvement, to highlight gaps of knowledge. The evaluation grade and feedback are then to be reviewed by a human before sending back out to the class.

2.2.3 Fine-tuning ChatGPT for automatic scoring

(Latif & Zhai, 2024) explores the potential of fine-tuning ChatGPT-3.5 for automated assignment grading and compares its results with BERT (Bidirectional Encoder Representations from Transformers). This is to better understand how automated grading systems can be implemented and which models may be superior. The study highlights the potential benefits while digging into some concerns.

The strengths and weaknesses of using ChatGPT for automated grading have been highlighted, offering valuable insights into its potential applications and limitations. The strength their project shows is how ChatGPT outperforms BERT, which is very important knowledge for selecting the correct models to base the upcoming project on.

ChatGPT shows an average of accuracy 9.1% higher than BERT, this is a significant difference and can be further shown in Figure 2. A range of different questions were given to the models, represented by the names on the X axis of Figure 2. In some questions, such as Gas-filled balloons, no difference was noted, which showed the models were equally capable of identifying this question.

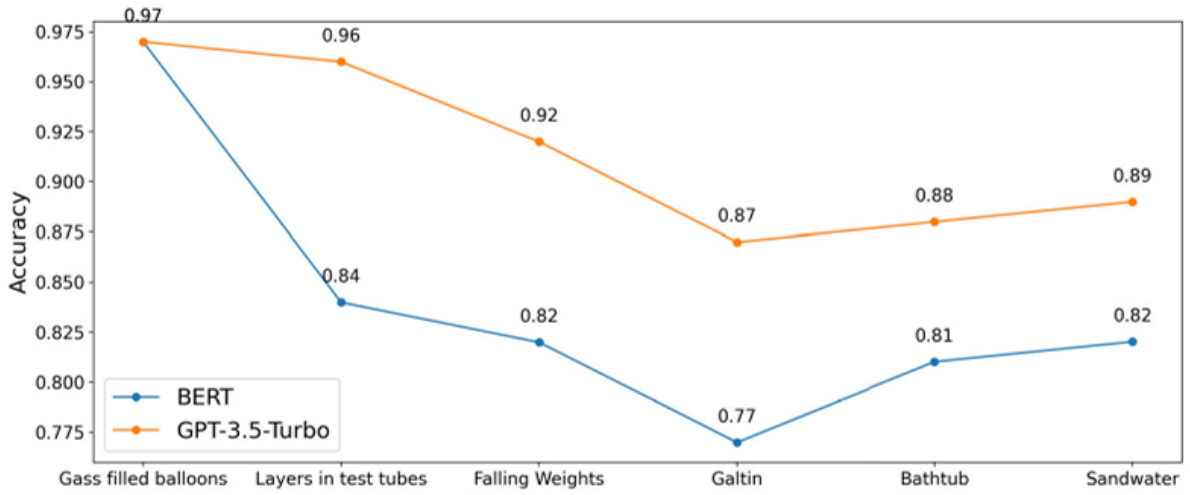


Figure 2. Accuracy comparison of fine-tuned ChatGPT and BERT for different assessment tasks (Latif & Zhai, 2024).

However, their project does not account for bias, fairness, and ethical implications. While also only comparing ChatGPT to BERT when some newer AI models such as Google's Gemini or Bard could be used, these models could provide better accuracy for automated grading.

2.2.3.1 Features

The features their project presents are data collection and preprocessing, fine-tuning and validations. These three main features are to ensure that the models are ready for grading assessment questions correctly.

The data is cleaned during preprocessing, which removes any irrelevant data or personally identifiable information. This is a necessity which should be carried out in the upcoming project to ensure student privacy. The data is then tokenized, so the AI model understands the data in a suitable format. However, the volume of data has not been taken into account which could cause issues when tokenizing large amounts of data.

The fine-tuning adjusts the learning rate, epochs, and batch size, which are all equally important in optimizing the accuracy of the AI model. A loss function is used, which measures how well the model is learning, quantifying this statistic allows for further fine-tuning of the model. Once fine-tuned, the model is validated using a separate evaluation set to access the performance. These are some interesting processes which will highly advance the performance of the AI model.

2.2.3.2 Architecture

The architecture of their project is collecting the data, preprocessing, model initialization, fine-tuning procedures, evaluation and validation, and baselines. An influential architectural step which has been included in their project is the fine-tuning procedures. This phase takes an already intelligent model in ChatGPT and turns it into a more functional model that can more accurately grade assessments.

2.3 Summary

As part of the reviews conducted in this chapter, several points arose that we would address in our requirements.

An automated assignment grading system needs to be backed by a large cloud provider such as Azure,as discussed in 2.2.2. The reliability that Azure provides means that the project is more stable, it also allows for many extra improvements and services to be added such as monitoring methods.

Another issue identified in this literature review was the need to clean/preprocess assignments to remove unnecessary data such as student ID's. Removing such data is a necessity due to it having no impact on the AI's result but could carry some heavy privacy and security issues.

In 2.2.2 we will now be implementing prompt engineering into the project to allow the feedback layer to be more accurate. This will help to provide better feedback to students that feels more realistic and gives insights into mistakes.

The design of the project has been improved since reviewing these papers due to having two new layers brought to my attention. These are the fine-tuning layer and validation layer, the fine-tuning layer is very important to improve the accuracy of the AI as shown in 2.2.3. The validation layer is critical because it allows us to check that the fine-tuning is making the model more accurate, when trying to improve a model some issues may occur such as overfitting. These can be resolved easily if the problem is identified by a validation layer, this layer can also help to prove the effectiveness of the AI, making it more appealing to users.

2.3.1 Comparison

All of these sources reviewed provided different learnings, with distinctive strengths and weaknesses. Some significant insights gained were for privacy, architecture, models, and feedback.

They all specialise in different areas of automation, however, they all highlight the need for data privacy. When collecting assignments from students, the data must be cleaned. This is because assignments can contain data such as student numbers, full names, and sensitive data. Properly cleaning the data will remove these points, since they are unnecessary for the AI to learn and grade papers.

These sources also express the need for customisation in different cases. The AI models should be able to learn different programming languages for example, this is needed because different assignments may contain different programming languages. The customisation should also allow markers to input different mark schemes, different assignments will need different marking schemes for the AI to process.

3 Methodology

3.1 Software Development Methodology

3.1.1 Suitable Software Development Models

3.1.1.1 Agile Model

This model involves breaking the project down into different phases to be able to highlight continuous development and enhancement. This is done by using sprints/iterations that follow a set amount of time, such as two weeks. Agile allows for the response to challenges, changes or improvements quickly and easily, making it a popular model in software engineering teams.

Agile models can be implemented with different frameworks such as Kanban boards to further deepen the model. This framework drives efficiency through the visualization of tasks, allowing teams to see where each part of a project is. Reviewing these tasks can help to quickly identify bottlenecks and remove them for more efficient development

Since change is accepted easily, it is a realistic approach to software development because of the tendency to have many moving parts. However, due to being so flexible, there is a lack of documentation which may confuse developers if not managed properly.

This model is suitable for my project because of the amount of change it allows for. When developing artificial intelligence, many pieces may need to be revised, therefore choosing a model with an acceptance for this is necessary.

3.1.1.2 Scrum Model

Scrum is an effective method because it allows developers to self-manage, and learn from previous experiences while being room for change. A main feature of Scrum is that it encourages teams to improve their skills continuously. Within Scrum, there tends to be three main roles: a product owner, a Scrum master and the development team (IEEE International Conference on Computing, Communication and Automation, 2017).

The product owner is responsible for the product, they are focused on the business, customers, and requirements. They use this knowledge to correctly prioritise the development team.

The Scrum master coaches teams and the product owner, looking for ways to fine-tune the methodology. They understand the work that is happening within the team to be able to help with the flow of the product.

The development team creates the product using the methods laid out by the product owner and Scrum master. They work closely together to be able to quickly develop products using different skill sets, organising themselves based on their current sprint plan.

3.1.1.3 Prototype Model

The prototype model uses a method where a prototype is made quickly. Requirements and design are done very fast which creates the prototype, this prototype is then made into the final product. However, if the final product is reviewed and not what was expected then a new prototype is made, and the process is repeated until the final product is as expected (Arnowitz et al., 2007).

A benefit of this model is that if the developer does not know of the abilities of the algorithm they are creating, this allows them to better understand the final results of their algorithms to reach the desired outcome quicker. A main issue of this model is that quality can be overlooked when attempting to develop prototypes too quickly, which means the final product is not up to standard.

This model is suitable because it allows AI algorithms to be completed and tested quickly, this would mean that many machine learning models could be made over a short period and tested with the users.

3.1.2 Software Development Model Used

The software development model used will be Agile with the Kanban framework. This is because agile allows developers to be able to quickly respond to changes in the software or requirements. This is highly beneficial because while training AI, testing is completed after each iteration of training, which allows for change based on this testing.

The iterative sprints are effective because they can align with the meetings with the project supervisor. This makes tracking the progress of requirements more manageable, leading to higher-quality development.

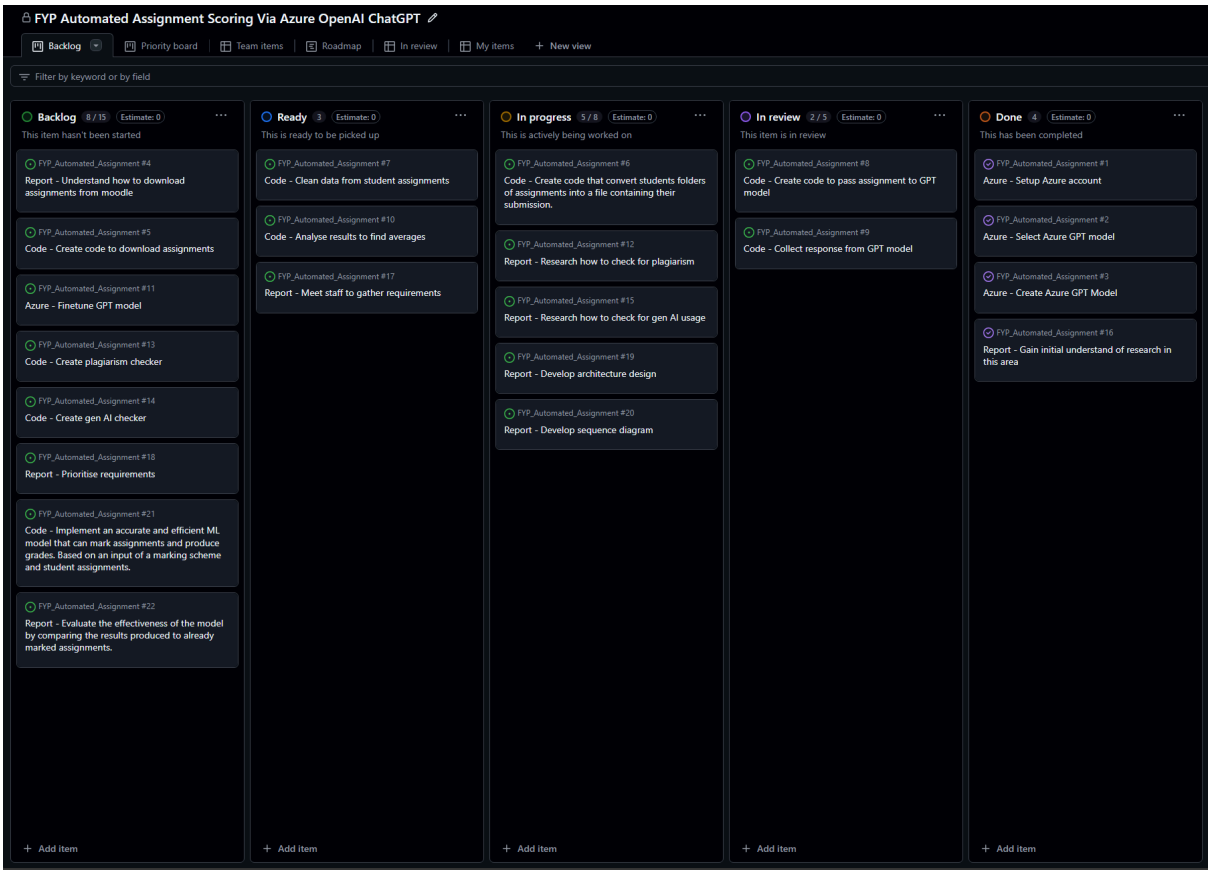
The Kanban board is effective since the simplicity of visualisation allows for easy management, which reduces bottlenecks. This tool also makes it easy to see the stage of each task from the requirements, which is useful when discussing with clients.

3.2 Project management

To ensure the development of the project and the deliverables are met, techniques have been put in place such as bi-weekly meetings with the supervisor, a planned timeline and source code management in GitHub.

The requirements will be gathered by using an interview to understand how the project can help to improve the current manual process. We will request access to past assignments and their feedback, this can be used as training data for a prototype. Once developed, a meeting with the client can be arranged to get feedback.

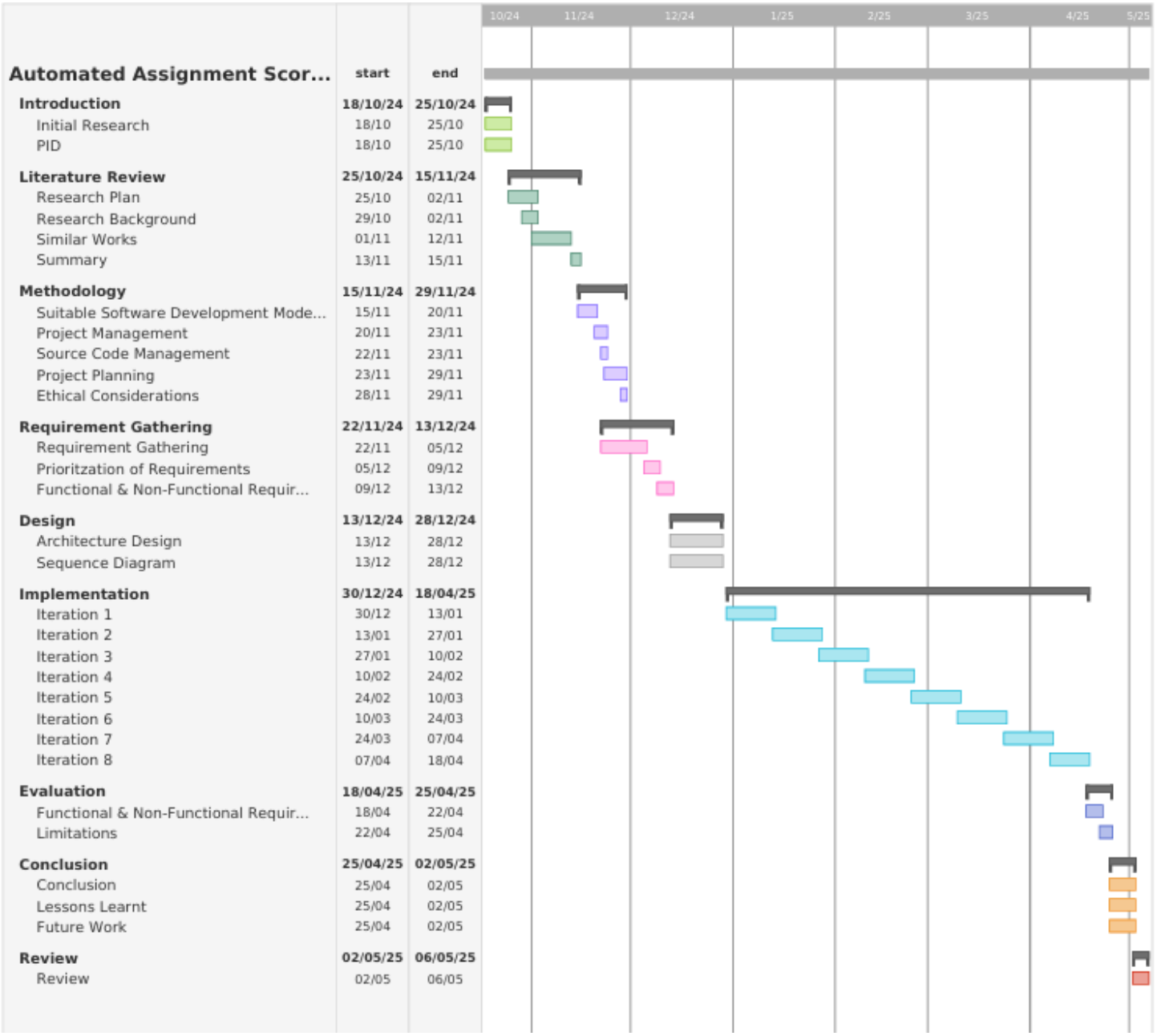
3.2.1 Source code management



3.3 Project Planning

3.3.1 Initial Plan

	Start	Duration	Task
	18/10	1 week	Introduction
	25/10	3 weeks	Literature Review
	15/11	2 weeks	Methodology
	22/11	3 weeks	Requirement Gathering
	13/12	2 weeks	Design
	30/12	3 1/2 months	Implementation
	18/04	1 week	Evaluation
	25/05	1 week	Conclusion and Future Work
	02/05	4 days	Review



Automated Assignment Scor...	start	end
Introduction	**18/10/24**	**25/10/24**
Initial Research	18/10	25/10
PID	18/10	25/10
Literature Review	**25/10/24**	**15/11/24**
Research Plan	25/10	02/11
Research Background	29/10	02/11
Similar Works	01/11	12/11
Summary	13/11	15/11
Methodology	**15/11/24**	**29/11/24**
Suitable Software Development Mode...	15/11	20/11
Project Management	20/11	23/11
Source Code Management	22/11	23/11
Project Planning	23/11	29/11
Ethical Considerations	28/11	29/11
Requirement Gathering	**22/11/24**	**13/12/24**
Requirement Gathering	22/11	05/12
Prioritzation of Requirements	05/12	09/12
Functional & Non-Functional Requir...	09/12	13/12
Design	**13/12/24**	**28/12/24**
Architecture Design	13/12	28/12
Sequence Diagram	13/12	28/12
Implementation	**30/12/24**	**18/04/25**
Iteration 1	30/12	13/01
Iteration 2	13/01	27/01
Iteration 3	27/01	10/02
Iteration 4	10/02	24/02
Iteration 5	24/02	10/03
Iteration 6	10/03	24/03
Iteration 7	24/03	07/04
Iteration 8	07/04	18/04
Evaluation	**18/04/25**	**25/04/25**
Functional & Non-Functional Requir...	18/04	22/04
Limitations	22/04	25/04
Conclusion	**25/04/25**	**02/05/25**
Conclusion	25/04	02/05
Lessons Learnt	25/04	02/05
Future Work	25/04	02/05
Review	**02/05/25**	**06/05/25**
Review	02/05	06/05

3.3.2 Actual Plan

	Start	Duration	Task
	18/10	1 week	Introduction
	25/10	3 weeks	Literature Review
	15/11	2 weeks	Methodology
	20/01	3 weeks	Requirement Gathering
	01/01	3 weeks	Design
	30/12	3 1/2 months	Implementation
	18/04		Evaluation
	25/05		Conclusion and Future Work
	02/05		Review

Some parts of the plan have not gone to play, and have either taken more time to complete or have started late. This is because of many factors such as other work, exams, misinterpretations of timescale for other objectives, and unexpected delays.

Requirement Gathering: This has had a delayed start due to not being completed before Christmas. Since there is a long Christmas break no interviews were able to be conducted, the return was on the 27th of January.

Design: This took slightly longer than expected, due to spending more time researching the possible methods of implementing the project. This meant more time was spent designing and therefore caused a delay.

3.3 Ethical Considerations

3.3.1

An ethical consideration that has been taken into account is student privacy. When grading the assignments if no preprocessing has been completed then the private data of students such as full names or student IDs could be exposed. This will mean cleaning of the data will be completed before passing the assignments to the AI model.

4 Requirements

4.1 Requirement Gathering

4.1.1 Interview

Questions to ask:

1) How should the AI assess the work?
    a) How does the interface/input look and work? Submitting folder of PDFs, GUI?
    b) How should the output look? Separate text responses for each submission? The table, where each row uses a marking scheme.
    c) Would you like a summary of the results? Analytical graphs (to show averages or students etc)? GUI?
2) Should different criteria be used for each assignment?
    a) Should the user specify the type of project when uploading?
    b) In what format should the criteria be uploaded?
3) What identifying data should be removed?
    a) How should the chat memory be handled? Assignments can be retained within chats.
4) What other functionalities be provided?
    a) Such as plagiarism checking in reports? Similarity in reports?

Justification for each question:

5) To provide an understanding of which areas should be assessed, such as structure, technical accuracy, clarity and such.
    a) To be able to provide better usability for assessors, such as bulk uploading PDFs, to streamline the process.
    b) The output needs to be clear and understandable for all teachers, depending on how they would normally look at results.
    c) Summarisation could help to see averages, trends, and strengths/weaknesses in the class. However, clarification on whether they would be helpful to teachers is needed.
6) Every final year project is on different topics for different reasons. However, there could be a general mark scheme for the structure.
    a) This could help to enable different mark schemes based on which topic is selected. The topic could be provided to the AI for the more specific parts of marking. Or should the markers define this for the AI?
    b) Understanding whether a mark scheme should be used or not, how would it need to be uploaded to the AI? In PDF format or something else.
7) Understand which data is normally contained within FYP's, such as first names or student names. Which ones typically would be handled carefully.
    a) The memory of the AI may also allow it to remember details about projects. To understand whether this may be an ethical data concern.

8) Extra features could be added. Understand what this may be.
    a) Typically, plagiarism is checked in assignments, so understanding if this sort of feature would be helpful.

Additional items to request:
Previous assignments
Previous feedback
Previous criteria

4.2 Analysis/prioritization of requirements

4.3 Functional Requirements

ID	Description	Arrived from	Priority	Justification
1	Store data on each student storing data on …, …	Client interview, question 2	Must have	This is needed to assign the roles for every student on the server
2.0	Create an admin web page	Got the idea from an existing bot, Carl Bot as discussed in 2.4	Would h…	This would be a good addition to the project, but the client did not specifically ask for it
2.1	The admin page needs to have a …			
3				

4.3 Non-functional Requirements

ID	Description	Arrived from	Priority	Justification
1	Do not store any private data (e.g., UP numbers on the cloud)	Data protection acts	Must have	This is needed to assign the roles for every student on the server
2	Create an admin web page	Got the idea from an existing bot, Carl Bot as discussed in 2.4	Would h…	This would be a good addition to the project, but the client did not specifically ask for it

5 Design

5.1 Architecture design/Component diagram

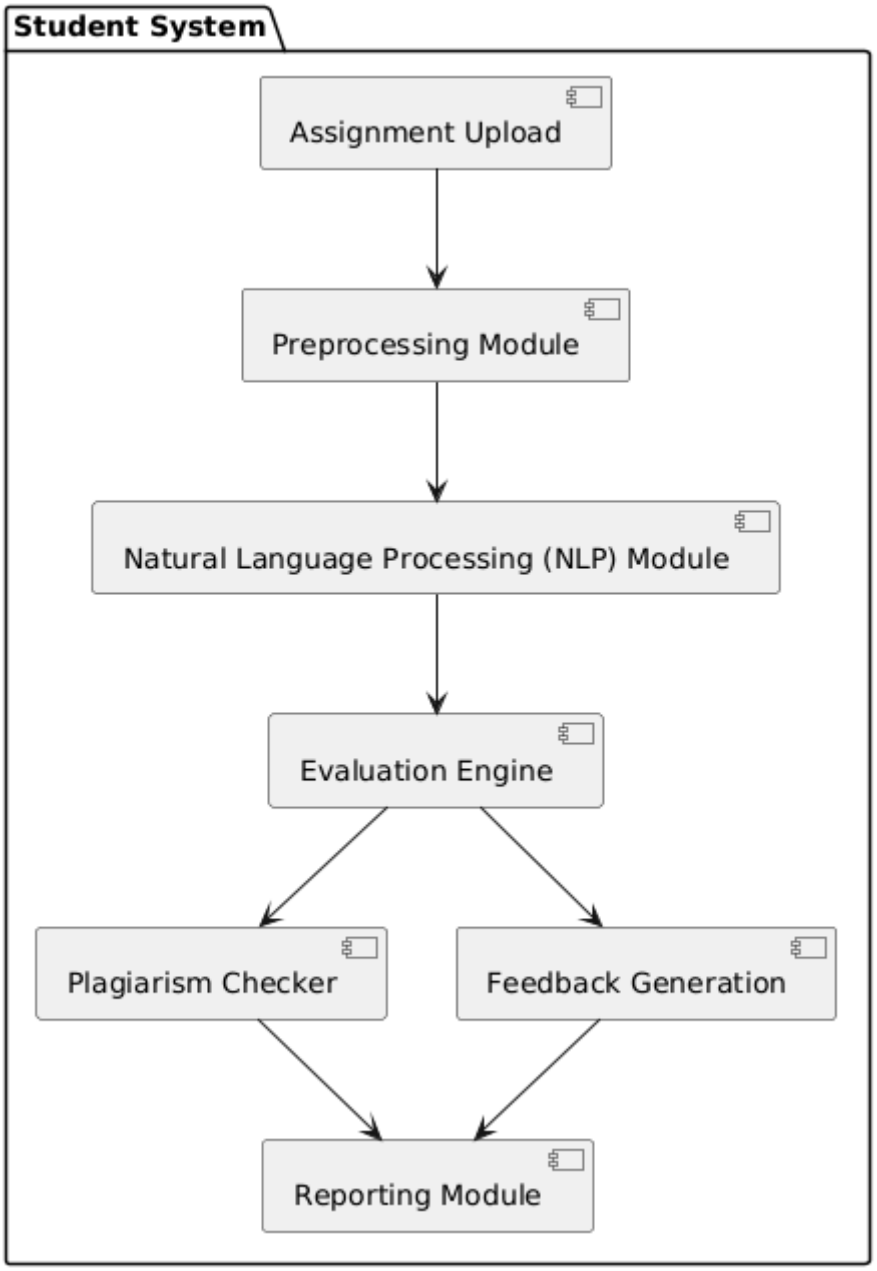


Figure x - Component diagram of the the system

5.2 Sequence Diagram

Assignment upload -> Criteria upload -> Preprocessing module (text extraction) -> NLP module (semantic analysis, grammar and syntax) -> Evaluation (grades the assignments based on criteria) -> Plagiarism checker (checks for plagiarism and maybe AI generation?)

-> feedback generation (feedback is produced to help the students improve) -> reporting module

6 Implementation

6.1 Iteration 1/Sprint 1/ …

6.1.3 Implementation

6.1.4 Testing

6.1.5 Review

6.2 Iteration 2

7 Evaluation

7.1 Functional Requirements

Title or ID	Description	Priority	Satisfied	Explanation
1			Not satisfied	
2			Partially sati…	
			Satisfied	

7.2 Non-functional Requirements

Title or ID	Description	Priority	Satisfied	Explanation
			Not satisfied	
			Partially sati…	
			Satisfied	

7.3 Limitations

8 Conclusion and Future Work

8.1 Summary

8.2 Lessons Learnt

6.3 Future Work

References

drstrangegithub. (2021). *drstrangegithub/discord.js-ModerationBot: A simple Moderation bot which can manage your server easily!! New updates come every month!!* GitHub. Retrieved October 23, 2023, from https://github.com/drstrangegithub/discord.js-ModerationBot

Marchand, B., Ponty, Y., & Bulteau, L. (2022). tree diet: reducing the treewidth to unlock FPT algorithms in RNA bioinformatics. *Algorithms Mol Biol*, *17*(8). https://doi.org/10.1186/s13015-022-00213-z

Appendices

Appendix I: Signed ethics form

Certificate of Ethics Review

Project title: Automated Assignment Scoring VIA Azure OpenAI ChatGPT

Name:	Callum Fry	**User ID**:	2061187 **Application date**: 18/11/2024 23:29:05	TETHIC-2024-109798 **ER Number**:

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Elisavet Andrikopoulou, Kirsten Smith**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Mani Ghahremani** Is the study likely to involve human subjects (observation) or participants?: No
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No
Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No
Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy. Supervisor comments:

Supervisor's Digital Signature**: mani.ghahremani@port.ac.uk** Date**: 26/11/2024**

Appendix II: Interview transcript

Appendix III: Questionnaire results