FIT3161: Advanced Computer Science Project 1

Project Proposal including Literature Review

A tool for streamlining of information/document processing.

Group - CS_11

Word Count: 8250 (excluding tables and diagrams)

Raunak Koirala

32509987

rkoi0003@student.monash.edu

Chen Zhang

30702429

cha0103@student.monash.edu

Lau Zi Fu

32685092

Zlau0010@student.monash.edu

Table of Contents

1. Introduction

In our modern digital era, processing vast amounts of documents and information has become a staple in a multitude of sectors, from academia to healthcare. From reviewing job applications to analysing medical reports, individuals and organisations grapple with various different data formats and criteria to draw conclusive responses. The current project seeks to introduce a tool that streamlines the examination, assessment, and response formulation processes related to various documents, ensuring efficiency, and minimising human error. The project team brings a diverse array of skills and experiences to the development of this project, aiming to harness technological advancements to address the challenges.

The expected outcome of this project is to develop a versatile software tool capable of seamlessly integrating document examination, assessment against predefined criteria, and the formulation of appropriate responses. This tool is envisioned to be adaptable across a variety of sectors, and will be designed to handle multiple document formats, including text, graphics, audio, and video. By digitising and automating these processes, we aim to significantly reduce processing times, minimise human errors, and improve efficiency overall. Additionally, with a strong emphasis on user experience, the software will prioritise intuitive navigation and rapid document transition, making sure that users can swiftly and effortlessly handle large volumes of documents. Furthermore, the inclusion of a comprehensive system administration interface will allow for easy customization and adaptability to diverse requirements. In essence, our anticipated outcome is an all-inclusive tool that is capable of reforming document processing and meeting a standard for efficiency, accuracy, and user-friendliness.

This proposal will look into a comprehensive background of the problem, drawing from existing research and technologies to establish the need and rationale for our software solution. Following the literature review, we will provide an in-depth project management plan, outlining the scope, objectives, and detailed strategies for successful project completion. The report will then transition into the design aspects, presenting a clear external view of the software's functionalities, interfaces, and interactions. Our chosen methodology, including technical toolsets and approaches, will be outlined to ensure a systematic approach to the software's development. Finally, we will present initial thoughts on how we aim to test our software to ensure its robustness and efficiency.

2. Literature Review

2.1. Introduction

For the literature review, we will be going through historical, current and potential future approaches to natural language processing, particularly in the realm of multimodal processing. We will be highlighting the pros, cons and potential alternatives to the researched approaches.

The literature is not purely restricted to a document processing tool but is broader in scope. This allows us to have a deeper, and more nuanced understanding of Natural Language Processing as a field of computer science.

In addition to this background literature research, throughout this proposal document, we've incorporated further comprehensive literature reviews addressing various key concepts, to ensure that we show a well-informed approach throughout the entirety of the project.

2.2. Approaches to building the document processing tool

Currently, one of the most successful approaches to Natural Language Processing has been "**BERT**", short for **Bidirectional Encoder Representations from Transformers**. It is a Machine Learning (ML) model for natural language processing. It was developed in 2018 by researchers at Google AI Language and serves as a Swiss army knife solution to 11+ of the most common language tasks, such as sentiment analysis and named entity recognition.

It is designed to pretrain deep bidirectional representations from unlabelled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications. It is built with the framework of ***Fine-Tuning*** and ***Pre-Training.*** During pre-training, the model is trained on unlabelled data over different pre-training tasks. For finetuning, the BERT model is first initialized with the pre-trained parameters, and all the parameters are fine-tuned using labelled data from the downstream tasks (Devlin, J., et al, 2018).

It is also immensely empirically successful, it obtained new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7%-point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (Devlin, J., et al, 2018).

As it stands, it's a preferred way of handling natural language processing compared to conventional methods such as Concurrent Neural Networks and Deep Neural Networks, due to its pre-trained model and higher accuracy results. However, it comes at a cost of flexibility. BERT is primarily used to handle natural language and is not able to process not text documents without significant modification. To handle truly universal document processing, a multi modal approach would be more appropriate.

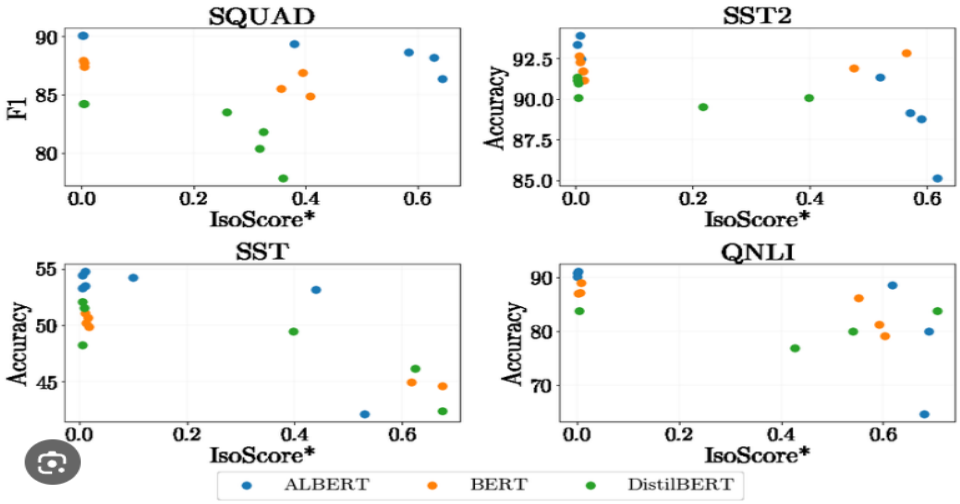


Figure 2.1 Accuracy Results of BERT

2.3. Current document processing tool workings

Automated document processing systems have become essential in various domains, including the medical field, for efficiently extracting information from complex documents (Adamo et al., 2015). This section looks at ideas and designs that are already used and highlights crucial components such as pre-processing, layout analysis, data extraction, and XML data exportation, essential for document processing tools. Additionally, Good's (1981) evaluation of the Etude system looks at the importance of user-centric design principles and user-friendly interfaces. Integrating insights from both sources can help optimize our tool for effective information management, particularly for handling intricate data such as medical records.

The first study emphasises the following components essential for document processing:

Pre-processing: The paper discusses image enhancement techniques, including equalization, binarization, and the removal of long lines, to improve document readability and facilitate subsequent processing steps (Adamo et al., 2015). The application of similar pre-processing techniques can significantly enhance the efficiency of our proposed tool, ensuring accurate data extraction from various document formats.

Layout Analysis: The research emphasizes the significance of layout analysis in identifying table structures and relevant data components within the document (Adamo et al., 2015). By employing techniques such as image segmentation and row analysis, the proposed tool can effectively identify and extract pertinent information from different document layouts, ensuring accurate data retrieval and classification.

Data Extraction and Classification: The paper highlights the importance of accurate data extraction from table cells, emphasizing the role of Optical Character Recognition (OCR) and knowledge base utilization in identifying and classifying medical test results (Adamo et al., 2015). Leveraging similar data extraction and classification techniques in our tool can streamline the processing of diverse document types, ensuring precise data retrieval and interpretation.

Exportation in XML Format: The study proposes the use of XML format for exporting extracted data, facilitating easy integration with databases and enabling seamless data analysis (Adamo et al., 2015). Implementing a similar export feature in our tool would allow for standardized data storage and efficient data sharing across different platforms, enhancing the overall usability and accessibility of the tool.

By incorporating insights from this research, our document processing tool can optimize data extraction, improve document layout analysis, and streamline information management, catering to the specific requirements of diverse document processing tasks in various domains. An example of an implementation could be the adoption of XML format for exporting data, as proposed in Adamo et al.'s (2015) study, which can enable continuous integration with databases and facilitate efficient data analysis. By implementing a similar export feature in our tool, we can ensure standardised data storage and enable effortless data sharing across different platforms. This integration will increase the overall usability and accessibility of our tool, providing healthcare professionals with a robust platform for managing and analysing medical data effectively and efficiently.

Good (1981) also presents a comprehensive evaluation of the Etude document processing system, emphasizing its ease of use and user-centric design principles. This evaluation serves as a vital reference for understanding the significance of user-friendly interfaces and efficient learning curves in the development of document processing tools.

The Etude system, developed by the MIT Laboratory for Computer Science, emphasizes ease of learning, and use through its English-like commands, reversible operations, and integrated online assistance feature (Good, 1981). Good highlights the system's tailored design, catering to the needs of both novices and experts, thereby minimizing any potential user apprehension and ensuring a seamless user experience.

The assessment of ease of use encompassed four fundamental aspects, namely ease of learning, ease of use once learned, anxiety factor, and user attitudes, each contributing to the overall user experience (Good, 1981). Specific measurable variables, including training time, task ease, anxiety levels, and user attitudes, were meticulously analysed through the application of psychological questionnaires such as the State-Trait Anxiety Inventory (STAI) and the Semantic Differential (SD).

Good's study really highlights the importance of user-centric design principles and user-friendly features in shaping the usability and effectiveness of document processing systems. Etude's success in meeting the criteria for ease of use reflects the positive impact of user-oriented design on overall user satisfaction and experience, emphasizing the significance of a well-designed user interface in promoting efficient and effective document processing.

Integrating the insights from Good's (1981) evaluation of the Etude document processing system into our project holds significant potential to enhance the usability and performance of our proposed tool, particularly in the context of handling medical laboratory data. For instance, our project can incorporate user-friendly interfaces and intuitive command structures inspired by Etude's design. This approach would enable users, regardless of their expertise levels, to navigate the system seamlessly, minimizing the learning curve and fostering a more efficient processing workflow.

Moreover, the implementation of reversible operations and integrated online assistance, similar to those in the Etude system, can provide real-time guidance and support to users, thereby reducing any anxieties associated with processing complex medical data. By integrating these user-centric features, our tool can ensure a smooth and hassle-free experience for individuals tasked with managing and analysing intricate medical laboratory results.

Furthermore, the assessment framework highlighted by Good, focusing on factors such as ease of learning, task execution, and user attitudes, can serve as a guiding principle in evaluating the effectiveness of our tool. By considering measurable variables like training time, task ease, and user anxiety levels, we can continuously refine our tool's interface and functionality to meet the evolving needs of users.

2.5. Conclusion

By combining the ideas from Adamo et al. (2015) and Good's (1981) work, we can create a user-friendly tool for managing medical data effectively. Using their methods for preparing documents, analysing layouts, and designing for users, we can ensure that our tool is easy to use and helpful for medical professionals. This will make it easier to handle complex medical data and improve the overall experience for users.

3. Project Management Plan

3.1. Project Overview

As there's an increasing need for streamlined systems that can manage the procedures of processing, examining, and responding to diverse documents better, our project, at its core, aims to fill this gap, bridging the divide between more traditional methods that are currently used and the modern demands of information processing.

The primary business goal is to offer organisations, be it educational institutions assessing assignments, medical professionals reviewing patient data, or HR departments sifting through job applications, an integrated platform that can help improve their efficiency, reduces errors, and drives rapid decision-making. By optimising these processes, organizations can potentially save time, resources, and improve overall productivity.

Our team believes that beyond the obvious operational benefits, our system serves a larger social and human objective. In education, quicker and more efficient feedback can lead to improved learning outcomes. In healthcare, faster response times can literally save lives. And in business, fast decision-making can lead to better talent acquisition and resource allocation. By focusing on creating a tool that is both technically robust and centred around users, we aim to create a positive impact on these important areas and make a difference.

The general objective of the project is to design and develop a comprehensive software tool that streamlines the examination, assessment, and response formulation processes for various documents and information across diverse sectors. In terms of the more technical objectives, our tool is planned to incorporate a range of technical functionalities to meet the diverse needs of users.

1. It will enable the viewing of any digitised material, ensuring users can seamlessly engage with content irrespective of its original medium.
2. The software will provide the capability for simultaneous viewing of the document and its respective assessment criteria, streamlining the assessment process. An integral feature will be the responsive formulation module, which allows users to relay appropriate feedback or responses based on the document's assessment.
3. The tool will ensure users can start processing within 30 seconds of accessing the system and switch assessments in 10 seconds or less.
4. To further improve user experience, the software will require minimal navigation inputs, presenting users only with essential information and ensuring a clean interface.
5. A dedicated administration and configuration interface will be added in, catering to system updates and back-end processes.

The project's major activities include the design and development of the software, the integration of features to ensure quick user interaction and timely feedback creation, and a lot of testing phases to ensure functionality and user experience meet the highest standards. Key milestones involve the completion of the prototype, the successful integration of core functionalities, and the final deployment of the software after thorough testing and user feedback integration. Each milestone is geared towards delivering a product that innovates traditional document processing methods, offering efficiency and accuracy.

3.2. Project Scope

3.2.1. Project Scope

In-Scope:

- Development of a software interface that allows for quick document processing.

- Features enabling simultaneous viewing of material in a digitised format alongside assessment criteria.

- A responsive method for users to generate appropriate feedback based on the provided document or information.

- Integration of user performance criteria, ensuring quick access, smooth transitions, minimal navigation, and focused information presentation.

- Creation of a separate system administration and configuration interface.

Out-of-Scope:

- Conversion of physical documents into a digitised format. The system assumes documents are already in digital form.

- Training users on how to use basic computer functionalities.

- Integration with third-party systems or APIs not mentioned in the project brief.

Constraints:

- Time: The project has a fixed timeline aligned with semester schedules. (Needs to be completed by the end of semester 1 2024)

- Budget: As a student project, there's a limited budget for any potential external resources or tools.

3.2.2. Product characteristics and requirements

REQUIREMENTS TRACEABILITY MATRIX

ID	Priority	Requirements	Assumption(s) and/or Customer Need(s)	Category	Functional or Non-Functional	Source	Status
R1	**High**	Document Upload	Users need to upload documents for assessment	User Interface	Functional	Project Supervisor/Project Document	Pending
R2	**High**	View Mode	Users need simultaneous view of document and criteria	User Interface	Functional	Project Supervisor/Project Document	Pending
R3	**Medium**	Response Formulation	Users should provide feedback based on viewed documents	User Interface	Functional	User Feedback	Pending
R4	**High**	User Performance Metrics	System responsiveness is essential for user satisfaction	Performance	Non-Functional	Project Supervisor/Project Document	Pending
R5	**Medium**	Administration Interface	Admins need to manage the system and monitor interactions	Administration	Functional	Project Supervisor/Project Document	Pending
R6	**High**	Quick Document Load times	System should be able to load documents swiftly	Performance	Non-Functional	User Feedback	Pending
R7	**High**	System Uptime	The system should be available for use most of the time	Reliability	Non-Functional	Project Supervisor/Project Document	Pending
R8	**Medium**	Scalability	The system should handle more users without losing performance	System	Non-Functional	Project Supervisor/Project Document	Pending
R9	**High**	Data Encryption	Data protection is essential for user trust	Security	Non-Functional	Project Supervisor/Project Document	Pending

R10	**Medium**	Web Accessibility	The system should be usable by people with disabilities	Accessibility	Non-Functional	User Feedback	Pending
R11	**Low**	User Guide Documentation	New users might need assistance in navigating the platform	Documentation	Functional	Project Supervisor/Project Document	Pending
R12	**Low**	Feedback Form Integration	Collecting user feedback could be beneficial for improvements	User Interface	Functional	User Survey	Pending

Table 2.1. Requirement Traceability Matrix

3.2.3. Product User Acceptance Criteria

1. Document Upload and Viewing

- Users can successfully upload documents of varying formats without errors.
- Uploaded documents are instantly available for viewing.
- The system displays the document and its corresponding assessment criteria side by side.

2. Response Formulation

- Users can provide feedback or formulate responses based on the viewed document.
- The response interface is intuitive and user-friendly.
- Responses, once submitted, are recorded and accessible for future reference.

3. Performance Metrics

- Upon initial access, the system loads and displays its interface in 30 seconds or less.
- Users can transition from one assessment to another within 10 seconds.
- The system provides navigation with a minimum number of mouse or keyboard clicks.

4. Administrative Interface

- System administrators can monitor and manage user interactions without encountering system errors.
- They have access to critical system settings, user management, and feedback collected.

5. Reliability and Performance

- The system maintains an uptime of 99% or higher.
- Document loading times are consistently fast, regardless of the number of users accessing the system.

6. Security Measures

- Uploaded documents and user data are encrypted.
- Unauthorised access attempts are detected and prevented.

7. User Guide and Feedback

- A comprehensive user guide is available, covering all functionalities of the system.
- Users can provide feedback through the integrated form, and feedback is stored for administrative review.

3.3. Project Organisation

3.3.1. Process Model

Selecting the right life cycle model for a project is quite important in determining its success. After a lot of deliberation and discussion as well as analysis of our project's unique characteristics and the requirements, we have chosen the Agile life cycle model. The complexity and evolving nature of our project needs a process model that supports adaptability and continuous improvement. Agile, born from the fervour to address the limitations of traditional software development methods, supports adaptability, collaboration, and customer feedback (Beck, K., et al., 2001). Instead of rigid phases, it divides development into short, iterative cycles, or sprints, allowing for continuous improvement. Therefore, our decision is backed by several relevant reasons make sense for both the nature of our project and the advantages the Agile model presents.

Reasons for Selection:

Given the specifics of our tool such as simultaneous viewing of assessment criteria and digitised material it's probable that our initial requirements might evolve or expand as we go deeper into its development. For our project, we're not just coding but also navigating the interaction between digitised content and user-friendly assessment criteria. The iterative nature of Agile, especially its sprint-based strategy, is ideal for such dynamic requirements (Serrador, P., et al., 2015). As we decode the synergy between digitized material and user-friendly assessment interfaces, the constant feedback after every sprint will further refine our tool.

Considering the uniqueness of our tool's functionalities, especially the need for quick transition between assessments and minimalistic UI design for focused user experience, we anticipate frequent tweaks and changes. Traditional models can be pretty inefficient for mid-project shifts. But with Agile, every sprint is an opportunity to adjust and deploy new requirements (Dingsøyr, T, et al., 2012). For example, if we discover that users prefer a different layout for digitised content in the middle of development. Agile would allow smooth integration of this change without major disruptions to the project as a whole.

Our tool's functionalities like transitioning from one assessment to another in under 10 seconds, while ambitious, come with inherent risks. Agile's in-built mechanism of retrospectives post each sprint allows us to identify potential bottlenecks or challenges early on (Dybå, T., et al., 2008). For instance, if there's a risk of increased load times due to simultaneous content viewing and assessment, Agile allows us to identify and resolve it proactively.

Our key user acceptance criteria, from quick system access to intuitive navigation, are milestones. With Agile's sprint reviews, we can measure our progress against these milestones consistently, ensuring that our tool's development remains on track and aligns with our vision (Conforto, E. C., et al., 2014).

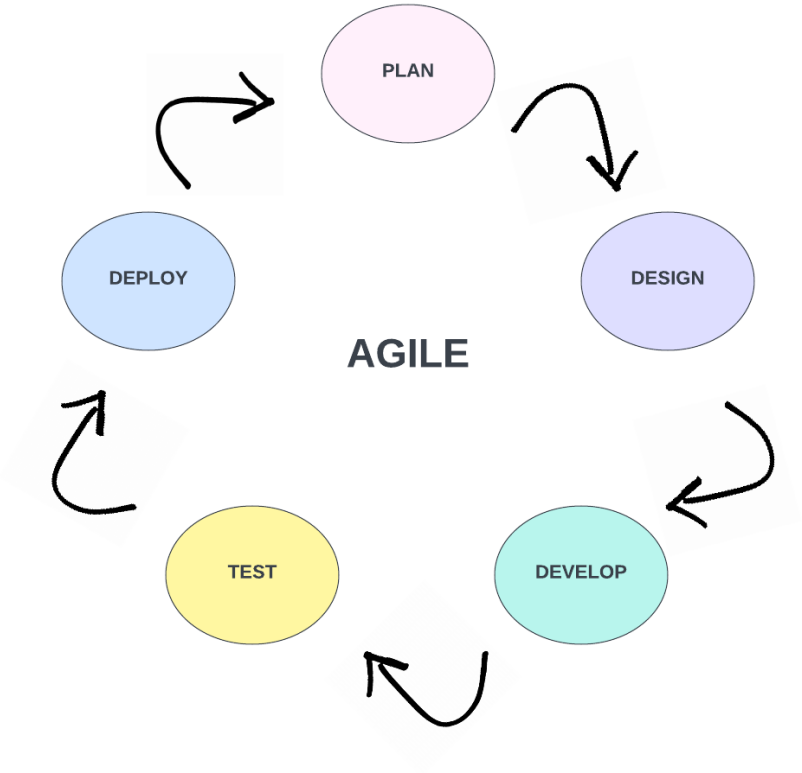


Figure 1. Agile Development Cycle

The Agile method we've adopted revolves around a continuous loop of planning, designing, developing, testing, and deploying. This cyclic process ensures that our tool remains adaptable and aligns with user needs. Each iteration refines our program and gives a dynamic approach to efficiently accommodate to any changing or additional requirements and/or stakeholder feedback.

3.3.2. Project Responsibilities

Team Member	Role	Key Responsibilities
Raunak Koirala	Project Manager & Technical Lead	• Setting and tracking milestones. • Ensuring proper resource allocation. • Leading development sprints and ensuring code quality. • Stakeholder communication and feedback integration. • Risk assessment and mitigation.
Lau Zi Fu	Systems Architect Lead & Quality Assurance Lead	• Creating and maintaining the system blueprint. • Ensuring the system design aligns with the project requirements. • Designing test cases and overseeing the testing process. • Identifying bugs and system inefficiencies.
Chen Zhang	UX/UI Lead & Evaluation Lead	• Designing wireframes and mock-ups for the software tool. • Collecting user feedback on design elements. • Ensuring the design aligns with the project's user acceptance criteria. • Creating and updating project documentation. • Collaborating with the development team to ensure design faithfulness during the development phase.

Table 2.2. Project roles and responsibilities

3.4. Management Process

3.4.1. Risk Management

Risk management is important in our project to ensure potential threats and challenges are foreseen, assessed, and a plan is in place to navigate them. For this project, our approach to identifying, analysing, and managing risk factors involves a systematic, iterative process, combining both structured and collaborative techniques. For risk identification we have chosen the following:

SWOT Analysis

We will use the SWOT (Strengths, Weaknesses, Opportunities, Threats) methodology to recognize both internal and external factors that might influence the project.

The process would look like so:

Identify Strengths: Assess internal advantages the project has in its current environment.

Pinpoint Weaknesses: Recognise limitations or areas of improvement.

Spot Opportunities: Examine external factors or scenarios that can be beneficial if used.

Determine Threats: Acknowledge external challenges or potential difficulties the project may face.

Brainstorming Sessions

Our team will partake in regular brainstorming sessions. These collaborative engagements are there to increase creative thinking, allowing us to really think deep and uncover risks from different perspectives, combining our diverse technical knowledge and the expectation of the project (Osborn, A. F., 1963).

Risks are all documented in a risk register, which is attached in **Appendix A**.

Once risks are identified, we categorise and prioritise them based on their potential impact and likelihood. Each risk is rated on a scale, allowing the team to focus on the most significant risks first. For each identified risk, we'll determine a response strategy which will be chosen based on the severity of the risk and the potential benefits or drawbacks of each response.

We know that risks can change as the project progresses and therefore our team will continually monitor and reassess risks. Any new threats or challenges that come up will be given back into our risk management process, which ensures our approach remains relevant and effective.

3.4.2. Stakeholder Analysis and Communication plan

Understanding the roles and influence of our stakeholders is crucial for ensuring the smooth execution of our project. Stakeholders, in this context, include our project team members, project supervisors, and the end users of our tool. Identifying their impact, influence, and potential risks, as well as developing strategies for effective engagement, will contribute significantly to the project's success. Below is an overview of our key stakeholders and strategies to mitigate potential risks associated with them.

Stakeholder	Impact	Influence	Tasks	Risks	Mitigation Strategy
Raunak Koirala	High	High	Project management, technical leadership, risk mitigation	May become a bottleneck if unavailable, risk of miscommunication	Cross-training, clear documentation, regular check-ins
Lau Zi Fu	High	High	System architecture, quality assurance	Potential delays if system design is not well-aligned with requirements	Regular design reviews, clear requirement documentation
Chen Zhang	High	Medium	UX/UI design, project documentation	Design may not align with user expectations or acceptance criteria	Regular user feedback sessions, iterative design process
Project Supervisors	Medium	High	Oversight, guidance	May have different expectations or change requirements	Regular status updates, clear communication of project goals and progress
End Users	High	Medium	Provide feedback, use the final tool	May not find the tool user-friendly or useful	Regular user acceptance testing, user feedback sessions

Table 2.3. Stakeholder Analysis

To ensure all stakeholders are well-informed and actively engaged, we have established a comprehensive communication plan. This plan outlines the reporting and communication strategies to keep the project on track and stakeholders aligned.

Item	Frequency	Audience	Purpose	Medium	Responsible Party
Weekly Check-ins	Weekly	Project Team	Update on individual tasks, identify any roadblocks or support needed	Direct Messages, Video Calls	Project Manager
User Acceptance Testing Results	As Needed	Project Team, End Users, Supervisors	Validate system meets user needs, gather feedback for improvements	Document Sharing, Email	Quality Assurance
Direct Messages	Daily/As Needed	Project Team	Quick updates, clarification on tasks, instant communication	Messaging Apps, Email	All Team Members
Team Meetings	Weekly	Project Team	Review project progress, strategize for upcoming tasks, team building	Video Calls, In-Person Meetings	Project Manager
Supervisor Meetings	Bi-weekly/Monthly	Project Supervisors, Project Manager	Provide updates to supervisors, seek guidance and approvals	Video Calls, In-Person Meetings	Project Manager
End User Feedback Sessions	As Needed	End Users, Quality Assurance	Gather user feedback, identify areas for improvement, ensure user satisfaction	Meetings, Surveys	Quality Assurance, Project Manager

Table 2.4. Communication Plan

3.4.3. Monitoring and Controlling Mechanisms

To make sure that we're making steady progress while also adhering to planned milestones in our project, we're using various monitoring mechanisms. Our Gantt Chart and Kanban Board will provide visual insights, helping identify any delays early on, while weekly check-ins and status update

meetings will allow for transparent communication and quick issue resolution. The Project Manager will monitor efforts, examining tools, and sessions and make sure any changes from the plan are quickly addressed. This will assure that we remain on track and maintain high standards across all project components.

Ensuring the quality, consistency, and security of our digital material assessment tool needs a comprehensive approach to review and audit mechanisms. This includes various practices ranging from version control and quality assurance to thorough documentation and ongoing training. These mechanisms are not only crucial for maintaining the integrity of our product but also for encouraging a culture of continuous improvement within our team.

1. Version Control:

Tool Utilisation: We will employ Git for version control, ensuring that every piece of code, every document, and every asset related to the project is tracked and versioned. This allows us to maintain a comprehensive history of changes, making it easy to roll back to previous versions if needed.

Branching Strategy: Our team will follow a feature-branch workflow. New features or bug fixes will be developed in separate branches before being merged into the main branch post-review. This makes sure that the main branch always contains stable and tested code.

Code Review: Before any code is merged into the main branch, it will undergo a rigorous review process. This involves checking for code quality, adherence to coding standards, and ensuring that new code does not introduce bugs or vulnerabilities.

2. Quality Assurance:

Automated Testing: We will implement a suite of automated tests, including unit tests, integration tests, and end-to-end tests. These tests will be run automatically on every code commit, ensuring that new changes do not break existing functionality.

Manual Testing: In addition to automated tests, our Quality Assurance Lead, Lau Zi Fu, will oversee a process of manual testing. This will include user acceptance testing, performance testing, and security testing, ensuring that the tool meets all user and performance requirements.

Continuous Integration/Continuous Deployment (CI/CD): Implementing CI/CD pipelines will ensure that our code is always in a deployable state. This facilitates faster release cycles, quicker feedback loops, and more stable releases.

3. Documentation:

Comprehensive Documentation: Every aspect of the project, from the codebase to the system architecture and user interfaces, will be thoroughly documented. This ensures that any team member can quickly understand and contribute to any part of the project.

Regular reviews and updates of the Project Scope, Requirements Traceability Matrix, and Risk Register are essential to keep the project aligned with its objectives and proactively address any potential risks. Tools like the Work Breakdown Structure, Gantt Chart, and Kanban Board provide visual aids and task management support, ensuring all team members are aware of their responsibilities and the project's timeline. Documentation such as the Version Control Log, Code Documentation, and Meeting Minutes will be regularly audited and reviewed to maintain transparency, uphold coding standards, and ensure that all decisions and changes are well-documented and justified.

4. Training:

Our team will take part in an ongoing training program to boost our skills and learn new things using online resources. We'll share what we learn with each other, creating a team environment where everyone helps each other grow. We'll have a collection of helpful articles, tutorials, and courses that everyone can use to improve their skills. This plan is not just for our current project's success, but also to make sure our team stays sharp for future challenges and to foster a culture of continuous improvement and adaptability.

3.5 Schedule and Resource Requirements

3.5.1. Schedule

For the schedule we use three main methods to manage our project and ensure that our team is on track to meet deadlines and milestones. We use the **work breakdown structure (see Appendix B)**,

which is a divide and conquer strategy which will help us break down our larger tasks into smaller, more manageable ones. We also utilise a **Gantt chart (see Appendix C)** to keep track of the duration of our entire project and figure out which tasks are dependent on the others and thus in which order they must be completed. Finally, we make use of a **Kanban board (see Appendix D)** for keeping track for what tasks have been completed already, what tasks are currently being done by team members and what is left to do. This not only provides a good visualisation on the amount of work completed and yet to finish, but it also ensures that team members are not redoing certain tasks.

3.5.2. Resource Requirements

This project will be completed by a team of three with advice and guidance from project supervisors (lecturers/tutors). Below are all the software and hardware requirements that were decided on earlier however may be subject to change during implementation according to feasibility.

Software Components:

- HTML, CSS, and JavaScript for UI development
- Python and Flask for backend development
- SQL Developer for robust database management
- Figma for collaborative UI design
- Git for version control and collaboration
- Jenkins and pytest for continuous integration and deployment

Hardware Components:

- High-performance laptops
- High-speed Wi-Fi connection for reliable networking
- Cloud-based Git repository for collaborative code management
- Development Platform:
- Visual Studio Code (VS Code) as the primary integrated development environment (IDE) for its familiarity, excellent support for web development, and a wide range of useful extensions and plugins.

4. External design

Our software is a web-based application, and we take Moodle as an example. Moodle is a mature business software that provides a customizable eLearning solution and is used by a lot of universities around the world.

4.1. Accessibility

One of the goals is to let anyone who is first accessing the software can start using it in 30 seconds. This requires the UI to be simple and navigate easily. Simple means the UI should be easy to read. Moodle uses colors to distinguish work zones and the menu. It uses dark colors to show the menu clearly in a list. The bright colors show the user general information. We will be concerned about when we design the UI. The menu bar listed on the left is a good option. It allows users to create many indexes and navigate easily. The user can only see the relative information with a specific index unless they click other indexes. In this case, users only need 4 steps to submit a file.

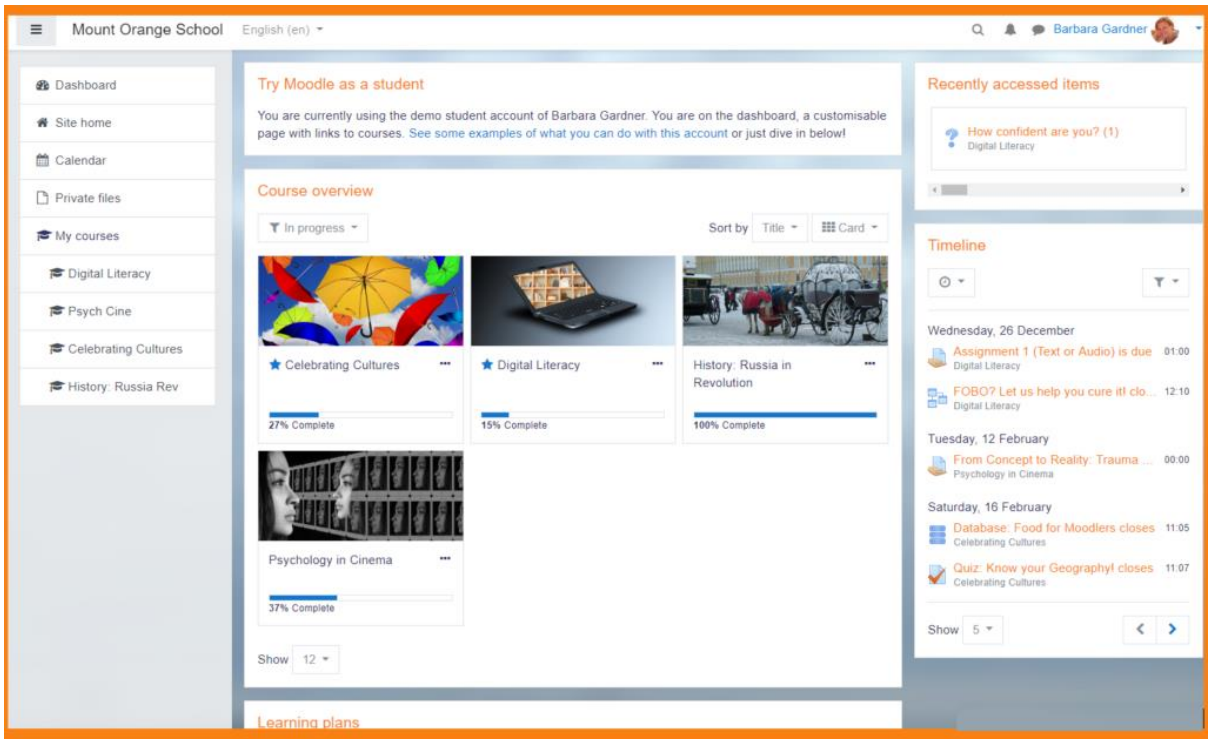


Image 4.1. Example UI

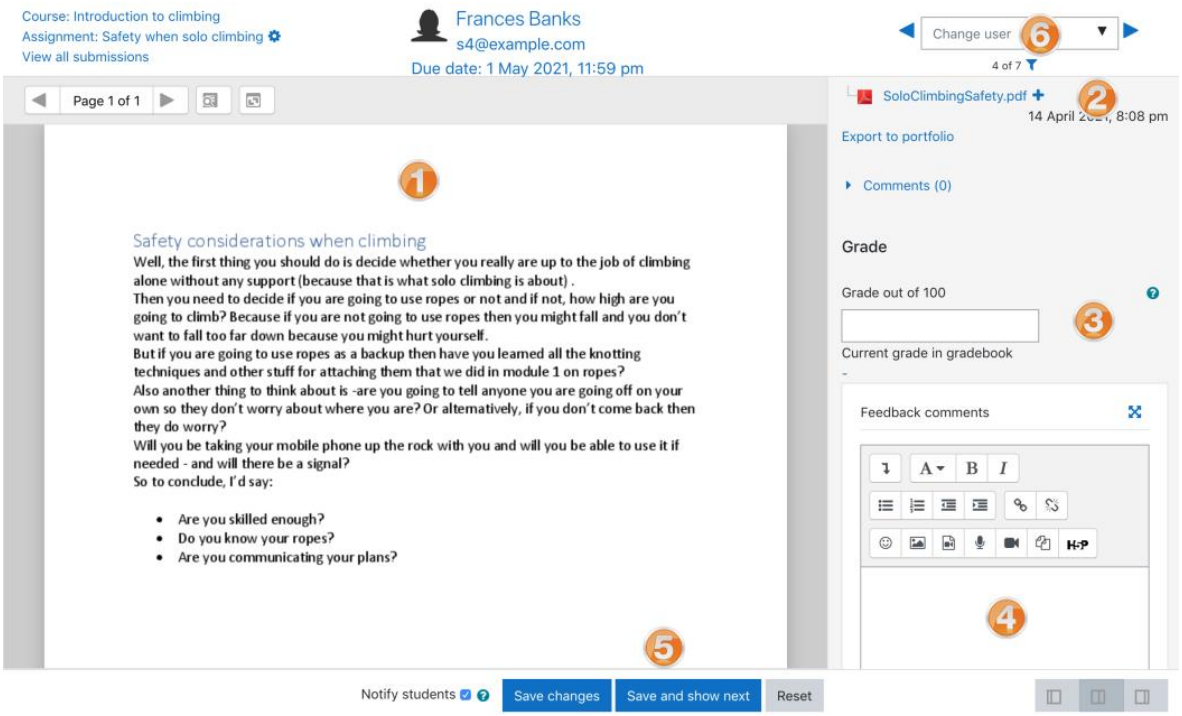


Image 4.2. Mockup

4.2. Time

Time means the software should be accessed quickly. We have let users quickly submit a file, but we also want users to switch between files quickly. We want users to browse the file through the software and sometimes the file size can be huge. So, we decide when user browse the file, the web will load a compressed file to reduce the loading time. Users can also ask to load the original file but will require more time to download.

4.3. Compatibility

The software allows users to use computers to process the documents. We are going to support it run in Edge, Chrome, Firefox. We are also going to make a smartphone version website tool in the future. JavaScript can provide large compatibility and it has powerful package managers like Npm that allow us to develop features easily. It is a lightweight language that does not require heavy software installations or hardware infrastructure to run. This makes it a great choice for web applications and mobile apps that require fast and efficient performance.

5. Methodology

The methodology of the project is extremely important, and represents the entire framework of how to approach, design, and implement our project. Our team spent a considerable amount of time contemplating on the toolset and approaches, version control system, data management solutions, including but not limited to libraries, programming language, data storage and processing. This section will provide a detailed justification, and explanation of what tools, software, and algorithms are used, and how they interact with each other to form the backbone of our project.

5.1 Back End, Web Server, Database

The **backend** is responsible for processing data, implementing business logic, and interacting with the database. After much discussion, we have come up with a comprehensive software stack that will be used to create the logic and data management side of our web application. Below Is a list of the full technology stack for our backend.

Category	Software
Programming Language	Python
Framework	Flask
Database Management	SQL developer, MySQL
Libraries	Natural Language Tool Kit (NLTK), scikit-learn
Algorithms	Concurrent Neural Networks, Deep Neural Networks
Web server proxy	Apache
Open-Source Software	Hugging Face Transformers, BERT

Table 4.1. Backend Stack

5.1.1 Programming Language

Python is an ideal choice for the backend of our document processing tool. Python's versatility and extensive libraries enable us to efficiently handle the logic required for document examination, assessment against criteria, and response formulation which falls under natural language processing. Python's built in frameworks like flask allows for us to build efficient web-based applications. The community support is also very strong, it's easy to use for beginners, many of the frameworks provide an API for python and optimization of the NLP based system is less complex compared to other programming paradigms (Thanaki, J., 2017).

5.1.2 Framework

When it comes to web-based frameworks, the two most popular for Python are Django and Flask specifically, and for good reason. However, we will be using **Flask** for our project, since it is more suitable for our needs. Let's compare both frameworks to see why.

The most significant advantages of Flask were that it provides simplicity, flexibility, fine-grained control and quick and easy to learn. On the other hand, Django was easy to work with because of its extensive features and support for libraries. Another main advantage of Django is its scalability. It is

best fit for a large-scale application. Each framework has its limitations and radiates a fair share of disadvantages. For example, Django is a bit cumbersome for smaller sized applications. However, Flask is too simple to not have the necessary features within the framework (Ghimire, D., 2020).

As it stands, our project is smaller and simpler compared to other projects out there. Therefore, the steep learning curve, suitability, and disadvantages for smaller applications compared to Flask's simplicity, flexibility, and ease of learning presents a more favourable choice for our project.

5.1.3 Database Management

SQL Developer is selected as our database management tool as all members are familiar with it and due to its robust capabilities. In our project, we must store and retrieve diverse documents, assessment criteria, and user data. SQL Developer offers advanced features for creating, editing, and querying databases, which aligns perfectly with our need for efficient data storage and retrieval. For instance, when assessing job applications against selection criteria or responding to government agency submissions, SQL Developer helps maintain data integrity and supports complex queries.

For the actual Database system itself, we were torn between choosing either SQLite, or **MySQL**. Both present significant advantages and disadvantages. Ultimately, our team chose MySQL because it is highly scalable, supports multiple user environments, handle multiple connections simultaneously, and is open source. We expect our project to have a significant user base, therefore based on these factors, MySQL is more appropriate (Vishwash, V., 2020).

S.NO.	MySQL	SQLite
1.	MySQL requires a database server for its functioning. Hence, it follows client/server architecture.	SQLite does not require a server to run. Hence, it is serverless.
2.	It can handle multiple connections simultaneously.	It can handle only one connection at a time.
3.	It is highly scalable and can handle a large volume of data very efficiently.	It can handle only small set of data if the volume of data increased its performance degrades.
4.	It requires large space in the memory for its functioning (approx 600 Mb).	It requires only some KBs of space as it is very lightweight approx (250Kb-300Kb).
5.	MySQL supports multiple user environment.	SQLite does not support multiple user environment.
6.	It also supports XML format.	It does not support XML format.
7.	MySQL is licensed under the GNU General Public License which means it is open source and free to use	SQLite is in the public domain and can be used without any licensing restrictions.
8.	MySQL is a client-server database management system, which means it requires a separate server process to be running on the machine to handle database requests	SQLite is a serverless database management system, which means it does not require a separate server process to be running and can be embedded directly into an application.

Table 4.2. Difference between MySQL and SQLite

5.1.4 Libraries

Since our backend is using Python, and involves heavy usage of natural language processing, the **Natural Language Tool Kit (NLTK)** library is a natural choice for data pre-processing, since it provides a vast toolkit for text transformation, such as tokenization, stop word expulsion, stemming, POS tagging, chunking and NER. (Yogish, D., et.al, 2019)

To explain further, NLTK is a prominent platform for Python programs in natural language data processing, offering interfaces to 50+ corpora and lexical resources, text processing libraries, NLP wrappers, and an active forum. It's valuable for linguists, engineers, students, educators, and researchers. NLTK is free and open source, available on Windows, Mac OS X, and Linux. It's known as an excellent tool for computational linguistics and offers a book, "Natural Language Processing with Python," providing a practical introduction to Python programming for language processing, authored by NLTK creators (NLTK, 2009).

5.1.5 Web Server Proxy

We will be using **Apache** as a reverse proxy for our application. The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial grade, featureful, and freely available source code implementation of an HTTP (Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. This project is part of the Apache Software Foundation. In addition, hundreds of users have contributed ideas, code, and documentation to the project. (Apache, 2019)

The question is, why use a reverse proxy at all? Indeed, one can establish a connection between the client and the back end with flask only, but it is not as efficient as a dedicated webserver since it can only handle one application at a time. On the other hand, A reverse proxy serves to protect sensitive data on isolated networks by routing Internet requests through a firewall. It can reduce network traffic by caching data and distribute requests across multiple content servers for load balancing. One key advantage is that it operates discreetly, redirecting or rejecting requests without revealing the actual content servers. It validates and caches requests for efficiency. (IBM, 2023)

Hence, the advantages of security and efficiency of using a reverse proxy is suitable for a project like ours which expects a sizable user base.

5.1.6 Open-source software

Open-source software is extremely important, since it reduces costs and is always updated constantly if the user base is large enough. It reduces time spent on reinventing, allowing us to focus on the user-experience of our web app. Since our project is largely based on Natural Language Processing, it makes sense to utilize pre-trained models that are time-tested and proven to be effective.

BERT

In this case, we will be using the family of language models “**BERT**”, short for **Bidirectional Encoder Representations from Transformers**. It is a Machine Learning (ML) model for natural language processing. It was developed in 2018 by researchers at Google AI Language and serves as a Swiss

army knife solution to 11+ of the most common language tasks, such as sentiment analysis and named entity recognition. This technology has been explored in section 2 literature review.

Because of how versatile BERT is, and how we can fine tune it to perform most Natural Language Processing, we will be fine-tuning it to perform Natural Language Understanding to assess documents based on the given criteria.

Hugging Face Transformers

Hugging face Transformers, is an open-source catalogue of NLP models provided by the LLM community. Transformers provides APIs and tools to easily download and train state-of-the-art pretrained models. These models support common tasks in different modalities, such as:

Natural Language Processing: text classification, named entity recognition, question answering, language modelling, summarization, translation, multiple choice, and text generation.
Computer Vision: image classification, object detection, and segmentation.
Audio: automatic speech recognition and audio classification.
Multimodal: table question answering, optical character recognition, information extraction from scanned documents, video classification, and visual question answering

5.2 Front End

In this section, we will be detailing the software needed to construct our front-end. It is extremely important to consider the correct tool kit, and software to ensure our website is built with the latest and most suitable stack.

Programming Languages: JavaScript/ TypeScript, CSS, HTML

Framework: Model View Controller (MVC)

5.2.1 Programming Languages

For our document processing tool, using HTML, CSS, JavaScript or TypeScript is suitable because these technologies allow us to create a responsive and accessible web-based UI. This is crucial because our tool needs to handle documents in various formats, including text, graphics, audio, and video. HTML, CSS, and JavaScript provide the flexibility and compatibility required for rendering these diverse content types within the user interface. For instance, when viewing medical examination results with graphical data or watching videos for rating, HTML, CSS, and JavaScript ensure seamless user interactions and display capabilities.

5.2.2 Framework

The chose architecture for our web application is the **Model View Controller**, otherwise known as "MVC". It is a pattern in software design commonly used to implement user interfaces, data, and controlling logic. It emphasizes a separation between the software's business logic and display and is used in most Front-end frameworks in modern industry. A concrete Framework for web development have not been confirmed yet, but our team has decided on either **Angular** or **React**. One is an actual framework, while the other is more of a library.

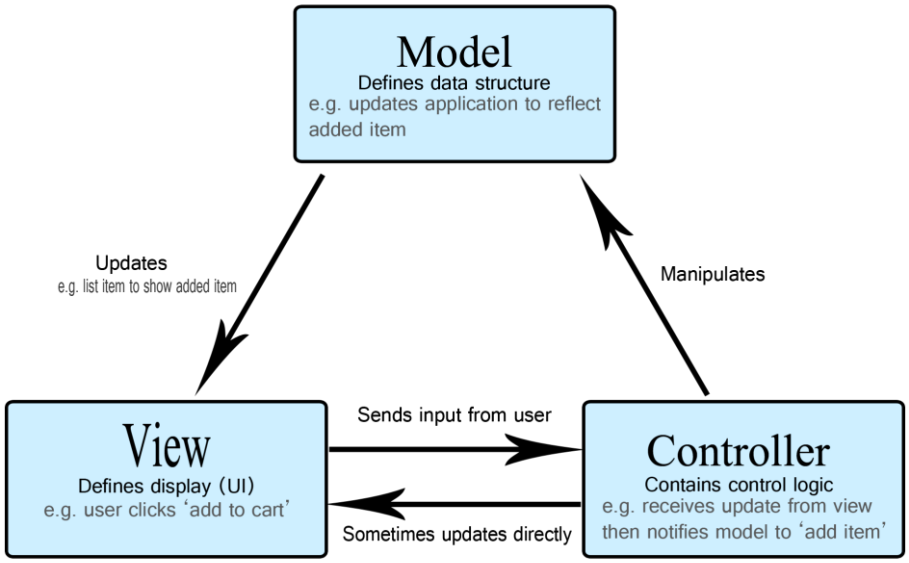


Figure 4.1 Model View Controller

5.3 Approach

The approach of our document processing tool is ambitious, but rather simple and straight forward approach wise. In essence, our tool's algorithm is modular in nature, separating responsibilities to a chunk of the workflow. This will ensure scalability, ease of debugging, and reduced dependencies between modules. The program is separated into three distinct phases:

Document Conversion -> **Assessment** -> **Response Formulation**

5.3.1 Document Conversion

There are 4 types of documents, which all types of documents are grouped into: Text, Audio, Image, Video. The idea, is to transform each type of document into natural language, using specific NLP techniques for each. Phase 1 will ensure all documents are usable for phase 2, which can only accept natural language documents because of the specific approach we took.

Index	Document Type	Algorithm
1.	Text	Summarization
2.	Audio	Sentiment Analysis, Text to Speech
3.	Image	Image captioning
4.	Video	Multimodal Transformers

Table 4.3. Algorithms for each type of Document

5.3.1.1. Text

Almost all forms of text files are accepted, such as .txt, .pdf, .doc, etc. Basically, anything that can be considered as natural language. These files themselves are very varied in length and size, ranging from potentially 5 pages to hundreds. To address this issue, summarization is used to shorten the documents to workable sizes. This will be achieved with a **fine-tuned BERT model**.

Using a BERT model for summarization has numerous advantages over traditional methods like recurrent neural networks. In a study conducted by Yang Liu, they found that a fine-tuned BERT model – BERTSUM, outperformed the previous best-performed system by 1.65 on ROUGE-L, when trained on a dataset of CNN/Daily mail articles (Yang Liu, 2019). Our approach will be heavily based on their implementation.

To fine tune BERT for summarization, multiple changes are made. BERT is trained as a masked-language model, the output vectors are grounded to tokens instead of sentences. Meanwhile, although BERT has segmentation embeddings for indicating different sentences, it only has two labels (sentence A or sentence B), instead of multiple sentences as in extractive summarization. To fix this issue, we will be using two techniques: **Encoding Multiple Sentences** and **Interval Segment Embeddings**. A diagram of the approach is provided in Figure 4.3 (Yang Liu, 2019).

We will then fine tune it with **Summarization Layers.** we will build several summarization-specific layers stacked on top of the BERT outputs, to capture document-level features for extracting summaries, using two main techniques: **Simple Classifier** and **Inter-sentence Transformers** (Yang Liu, 2019).

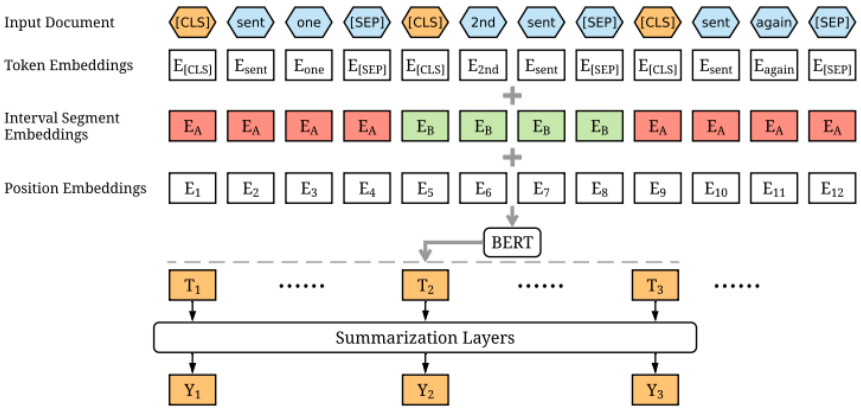


Figure 4.2 The overview architecture of the BERTSUM model

5.3.1.2. Audio, Image, Video

For Audio, Image, and Video, no specific model has been chosen as of now. Our team has a clear understanding of what needs to be done, but because of the vast area of research and scope of the project, we have decided to focus on getting text files implemented correctly first. Our team will go through an iterative process of testing and documenting many models to choose the most suitable ones for our project.

We will be using artificial intelligence techniques to perform multimodal summarization. Primarily, the open-source catalogue of "**hugging face**" will be used extensively throughout the process, as it provides a vast array of usable models that can tackle audio, video and image summarization. Custom made Deep Neural Networks are also considered. API's and outsourced models are options as well, such as OpenAI's whisper, HiveAI for image captioning and so forth. As always, this stage of the process is still under development and will change according to needs and requirements.

5.3.2 Pre-Processing

Before analysing the converted documents, pre-processing is needed to ensure everything goes smoothly. For this task, the library of NLTK will be used, with techniques including but not limited to tokenization, stop word expulsion, stemming, POS tagging, chunking and NER.

Data pre-processing is vital because data can come in various forms, such as structured tables, unstructured tables, images, audio files, and videos. Machines can't directly understand this data; it needs to be converted into 1s and 0s. Raw data can't be fed to a machine learning model and expected to train it (Maharana, K., et al, 2022).

Data pre-processing is the initial step in machine learning, transforming and encoding data so that machines can quickly analyse its features. It's crucial for the generalization performance of supervised machine learning algorithms. The time spent on pre-processing can be a significant portion of the entire classification process, possibly taking up to 50% to 80%. This highlights the importance of data pre-processing in building effective models and improving data quality (Maharana, K., et al, 2022).

5.3.3 Assessment

To perform such an analysis, two systems will be used. Primarily, the bulk of the assessment will be done with a **fine-tuned BERT model** like before, while simpler criterion can be handled by **rules-based systems** such as **REGEX** and **parsing**. The Bert-model will be configured in a similar way compared to BERTSUM, so we will instead focus on rules-based systems.

To handle simple grading criterion like grammar, punctuation and sentence structure, rule-based systems are used. To get a better understanding, a brief history of the method is mentioned.

Rule-based systems, initially developed by AI researchers, were often synonymous with expert systems, though there's a distinction. "Expert system" pertains to tasks requiring expert knowledge, while "Rule-based system" relates to representing knowledge explicitly instead of procedural code. While early knowledge-based systems were mainly expert systems, the same architecture extended to various systems. Rule-based systems played a crucial role in expert systems like Mycin for medical diagnosis, offering advantages in knowledge acquisition, maintenance by domain experts, explanation of reasoning, and enabling general-purpose inference engines to derive conclusions not initially apparent. (Masri, N., et al, 2019)

The core components of the architecture of a rule-based system consists of a knowledge base, inference engine, knowledge acquisition, explanation facility, user interface. The knowledge base holds domain-specific knowledge, including factual and heuristic knowledge. Factual knowledge is widely shared, while heuristic knowledge is experiential and individualistic. The inference engine, the system's brain, performs reasoning by processing rules and can use forward chaining or backward chaining. Knowledge acquisition involves transferring expertise to the system and is achieved through techniques like protocol analysis and interviews. The explanation facility justifies the system's actions, and the user interface facilitates communication. Building a rule-based system is known as Knowledge Engineering, involving domain experts, knowledge engineers, and users. Domain experts provide specialized knowledge, knowledge engineers develop the system components, and user needs should be considered during system design. (Masri, N., et al, 2019)

Knowing this, our approach will most probably use an already existing model that is constructed with a knowledge base of essays, movie reviews, podcast reviews and artistic evaluations. If such a model does not exist, we will construct our own. However, sourcing data for this specific task presents a huge challenge, therefore open-source datasets will be used extensively.

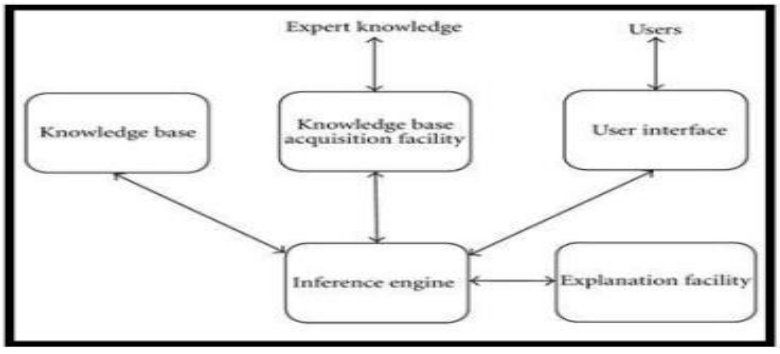


Figure 4.3 The architecture of a Rule-Based system

5.3.4 Response Formulation

After the data is assessed and processed, it will be fed into the 3rd and final phase of the process. Like before, we will use a custom BERT model to construct a response in natural language. The model will be fine-tuned to generate a response in a set format. It will either be in text as seen on the UI, or downloadable as PDF. An example is given below.

Introduction:
In this assessment, we evaluated the provided text against the following criteria:
1. Rich Vocabulary
2. Correct Grammar
3. Professional Style
4. Adherence to Topic
5. Demonstrates Deep Understanding of Researched Materials

Summary of Assessment:
The text met the criteria for Correct Grammar and Adherence to Topic. However, it did not fully meet the criteria for Rich Vocabulary, Professional Style, and Demonstrates Deep Understanding of Researched Materials.

Specific Feedback:
- ***Rich Vocabulary***: The text contains a good variety of vocabulary, but it could benefit from the inclusion of more domain-specific terms to enhance its richness.
- ***Professional Style***: The writing style is generally formal and professional, but there is room for improvement in terms of maintaining consistent tone throughout the text.
- ***Demonstrates Deep Understanding***: While the text shows a good understanding of the topic, it would be strengthened by providing more detailed references to the researched materials.

Overall Evaluation:
The text demonstrates a solid foundation but has room for improvement in vocabulary richness, style consistency, and referencing the researched materials.

Recommendations:
To enhance the text, consider incorporating more domain-specific terminology, ensuring a consistent professional tone, and providing detailed references to demonstrate a deeper understanding of the subject matter.

Closing Remarks:
If you have any questions or need further guidance on improving your text, please don't hesitate to reach out. We're here to help you achieve your writing goals.

6. Test Planning

6.1. Test scope:

In-scope:

1. The software's basic functions can work correctly. The users can upload and download required files. These files should be able to be browsed and marked through the software.
2. Users can use the software easily. They can start using the software without a very specific tutorial.
3. The UI should align with the project's user acceptance criteria.
4. The data storage is safe and there is no information leak.
5. The software can run on most computers and platforms.

Out of scope:

1. Users require more features on this stage.

6.2. Schedule

Start dates: The test will start when the software is completed.

Duration: 3 weeks

Deadline: before the semester ends.

Testers: The university students and tutors and testers from the Internet.

Technical resources:

Pytest: for code assurance

Git: for version control

Jenkins: for server testing

Communication software: Twitter, Facebook and Discord etc.

The first week we will work on debugging. The testers are all the group members. We will find out if any serious code issue exists. The basic requirement is that the software can achieve its basic functionality. While development teams are debugging, other team members will find testers for further testing. The testers can be students and tutors. We can also create social accounts to advertise the software and collect testers from the Internet.

When we ensure the software is ready for use, we will release a public test version via git and let testers test them and give feedback. This stage of testing is focused on improving user experience. The testers can give them feedback via git, discord, emails and private messages. During the public test, all code issues will be solved by hotfixes. Other features like UI design or user tutorials will be dependent on their importance, solved later or fixed by hotfixes.

7. Conclusion

In summary, our project aims to address the inefficiencies and challenges associated with manual document processing by developing a comprehensive software tool. Through an extensive literature review, we have identified the need for a more efficient and user-friendly solution that can cater to diverse document processing requirements. Our well-structured project management plan emphasizes the importance of a streamlined organizational framework, effective risk management, and clear communication channels to ensure the successful execution of the project.

Our design considerations prioritize accessibility, time efficiency, and compatibility, reflecting our commitment to creating a user-friendly and adaptable system. The outlined methodology, covering both back end and front-end aspects, demonstrates a strong understanding of the technical requirements and the necessity for a streamlined data processing approach. The comprehensive test planning approach guarantees the reliability and robustness of the software tool, promising a seamless and error-free processing experience.

Overall, our goal is to develop a software tool that not only simplifies the document processing process but also sets a new standard for efficiency, usability, and accountability. By improving the document processing experience across different domains, our project aims to significantly contribute to improved productivity and informed decision-making throughout all industries.

Appendix

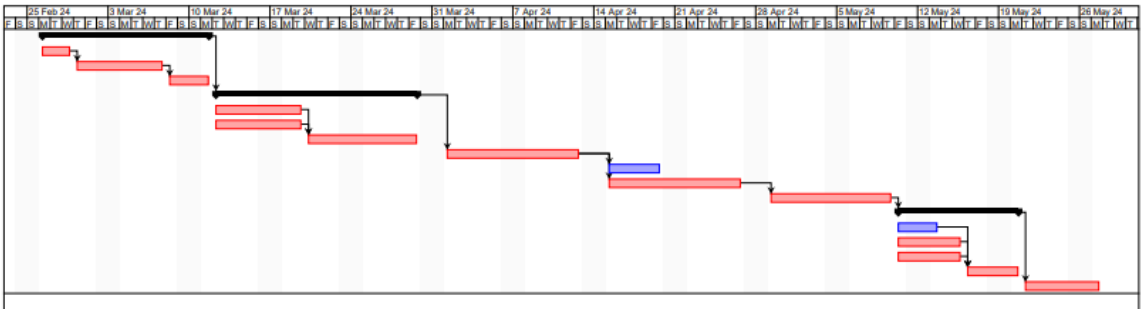
A. Risk Register

Rank	Risk ID	Description	Root Cause	Trigger	Risk Response	Incident Response	Owner	Probability	Impact (Score/10)	Overall Score	Risk Status
1	R11	Security vulnerabilities in the system	Coding errors, third-party tools, outdated libraries	Discovery of a security loophole or actual breach attempt	Regular security audits, keeping libraries up-to-date, secure coding practices	Immediate patching, inform stakeholders, and possibly notify users	Chen Zhang	50	9	450	High Priority
2	R10	Performance bottlenecks in the developed system	Suboptimal code, hardware limitations	Slow processing times during testing or real-world usage	Regular performance testing, efficient code practices	Optimize code, upgrade hardware, or adjust system architecture	Lau Zi Fu	50	7	350	Monitoring
3	R4	Scope creep due to changing requirements	External stakeholder interference, or unclear initial objectives	Increase in project tasks beyond initial plan	Clear project scope document, frequent stakeholder communication	Re-evaluate the scope, resources, and timelines	Raunak Koirala	40	8	320	Monitoring
4	R9	Software integration failures	Incompatibilities between modules or external APIs	Errors during integration testing	Pre-evaluation of tools/APIs, modular design for easy replacements	Debug and identify faulty modules, consider alternative tools	Lau Zi Fu	40	8	320	Monitoring
5	R2	Miscommunication among team members	Ambiguous task descriptions, or lack of regular meetings	Conflicting project outcomes	Regular meetings, clear documentation	Re-discuss, align, and document the decision	Raunak Koirala	50	6	300	Monitoring
6	R1	Insufficient skills in a specific area	Gap in team skill set	Project task requires a skill no team member possesses	Training team members and getting assistance	Seek external help or online resources	Raunak Koirala	40	7	280	Monitoring
7	R3	Loss of crucial project data	Hardware failure, human error, or cybersecurity breach	Missing data files	Regular data backups, secure storage solutions	Restore from backup, assess data loss impact	Lau Zi Fu	30	9	270	At Watch
8	R6	Team burnout due to overwork	Tight deadlines, or lack of task distribution	Decline in team productivity or morale	Balanced workload, regular check-ins, breaks	Reallocate tasks, team well-being sessions	Raunak Koirala	40	6	240	Monitoring
9	R5	Overlooked technical requirements	Initial assessment errors, or evolving project needs	Identification of new technical needs during development	Comprehensive initial assessment, regular reviews	Adjust project plan, inform stakeholders	Lau Zi Fu	30	7	210	At Watch
10	R7	Conflicts within the team	Personal differences, or disagreements on project directions	Noticeable tension or disputes among team members	Team-building exercises, conflict resolution strategies	Immediate conflict resolution meeting	Raunak Koirala	20	6	120	Low Priority

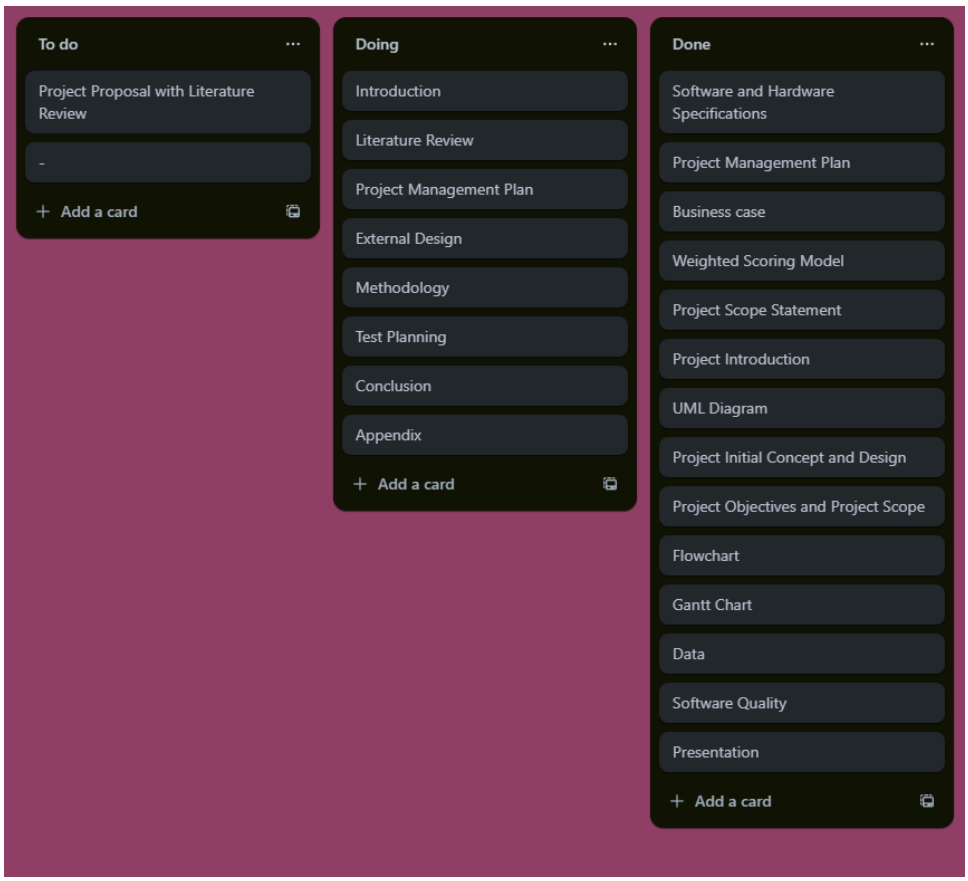
B. Work Breakdown Structure

C. Gantt Chart

		Name	Duration	Start	Finish	Predecessors
1		Research and planning	11 days?	2/26/24 8:00 AM	3/11/24 5:00 PM	
2		Communication plan Finalized	3 days?	2/26/24 8:00 AM	2/28/24 5:00 PM	
3		Market research	6 days?	2/29/24 8:00 AM	3/7/24 5:00 PM	2
4		Research Complete and gather ...	2 days?	3/8/24 8:00 AM	3/11/24 5:00 PM	3
5		Code creation	14 days?	3/12/24 8:00 AM	3/29/24 5:00 PM	1
6		Basic feature implementation	6 days?	3/12/24 8:00 AM	3/19/24 5:00 PM	
7		UI design	6 days?	3/12/24 8:00 AM	3/19/24 5:00 PM	
8		Complete prototype	8 days?	3/20/24 8:00 AM	3/29/24 5:00 PM	6;7
9		Functional testing	10 days?	4/1/24 8:00 AM	4/12/24 5:00 PM	5
10		Alpha user test and feedback	5 days?	4/15/24 8:00 AM	4/19/24 5:00 PM	9
11		Debugging	10 days?	4/15/24 8:00 AM	4/26/24 5:00 PM	9
12		Beta user test and feedback	9 days?	4/29/24 8:00 AM	5/9/24 5:00 PM	11
13		Polishing	7 days?	5/10/24 8:00 AM	5/20/24 5:00 PM	12
14		Research deployment method	2 days?	5/10/24 8:00 AM	5/13/24 5:00 PM	
15		Final polish based on beta u...	4 days?	5/10/24 8:00 AM	5/15/24 5:00 PM	
16		Finalize any legality issues	4 days?	5/10/24 8:00 AM	5/15/24 5:00 PM	
17		Product ready for deployment	3 days?	5/16/24 8:00 AM	5/20/24 5:00 PM	14;15;16
18		Publish project	5 days?	5/21/24 8:00 AM	5/27/24 5:00 PM	13



D. Kanban Board



References

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.

Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. International Journal of Project Management, 33(5), 1040-1051.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. Journal of Systems and Software, 85(6), 1213-1221.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and software technology, 50(9-10), 833-859.

Conforto, E. C., Salum, F., Amaral, D. C., da Silva, S. L., & de Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development? Project Management Journal, 45(3), 21-34.

Osborn, A. F. (1963). Applied imagination: Principles and procedures of creative problem-solving. New York: Charles Scribner's Sons.

Thanaki, J. (2017). *Python natural language processing*. Packt Publishing Ltd, 16-17.

Ghimire, D. (2020). Comparative study on Python web frameworks: Flask and Django, 13-31.

NLTK. (2009). Natural Language Toolkit - NTLK 3.4.4 documentation. Nltk.org. https://www.nltk.org/

Yogish, D., Manjunath, T. N., & Hegadi, R. S. (2019). Review on natural language processing trends and techniques using NLTK. In *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III 2* (pp. 589-606). Springer Singapore.

IBM. (2023, October 10). Proxy server types and uses for HTTP Server Www.ibm.com. https://www.ibm.com/docs/en/i/7.5?topic=concepts-proxy-server-types

Documentation Group. (2019). About the Apache HTTP Server Project – The Apache HTTP Server Project. Apache.org. https://httpd.apache.org/ABOUT_APACHE.html

BERT 101 - State Of the Art NLP Model Explained. (n.d.). Huggingface.co. https://huggingface.co/blog/bert-101

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Vishwash, V. (2020). *Difference between MySQL and SQLite*. GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-mysql-and-sqlite/

Mozilla. (2023). *MVC*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Glossary/MVC

Liu, Y. (2019). Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Liu, X., Xu, Q., & Wang, N. (2019). A survey on deep neural network-based image captioning. *The Visual Computer*, *35*(3), 445-470.

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91-99.

Masri, N., Sultan, Y. A., Akkila, A. N., Almasri, A., Ahmed, A., Mahmoud, A. Y., ... & Abu-Naser, S. S. (2019). Survey of Rule-Based Systems. *International Journal of Academic Information Systems Research (IJAISR)*, *3*(7), 1-23.

Moodle. (2023). Online Learning With The World's Most Popular LMS. https://moodle.com

Adamo, F., et al. (2015). Extracting information from printed medical laboratory results. Measurement, 61, 88-99.

Good, M. (1981). An Ease of Use Evaluation of an Integrated Document Processing System.