**A System with an …**

**Essay grader with AI feedback**

**Description**

The objective of this project is to take advantage of emerging and existing artificail intelligence and machine learning technologies to develop a system that is targeted towards university essay

To achieve these aims, this project will be implemented on or with either a mobile application or webiste system.

The system will provide students and possibly lecturers with an AI tool to do a rough draft of an essay grade, with feedback on areas to improve.

The system will grade the essay with the AI providing interactivity and social aspects to enhance their experience and offer a more digestable answer.

The project will choose an appropriate architecture . . .

Local Grading System with a cloud AI feedback api

Overview: In this approach, the system is built as a integrated local deep learning model trained for grading and a cloud feedback AI. This would involve having the local ML system grade the essay based on the marking scheme, with the AI processing the results and generating the feedback for the user, i.e. the student or lecturer.

Components:

Document Processor: Embedded within the application to convert various document formats into text.

Integrated AI Grading Model: A deep learning model trained on educational datasets and essay samples to perform grading. The api would receive the grading matrix, essay, and results from the ML system to generate feedback. The model can use attention mechanisms to pinpoint areas needing improvement (e.g., based on scoring rubrics).

User Interface: A mobile application or web interface where users upload documents and receive grading and feedback.

Local Database: Stores grading rubric, past submissions, and relevant feedback data for model updates and record-keeping.

Workflow:

1. User uploads a document.
2. Document Processor converts it to plain text.
3. The integrated Model grades the text and generates improvement suggestions.
4. The application displays the grade and feedback to the user.

**Requirements Engineering**

**Feasibility Study**

1. Are there similar systems to this in the marketplace (in Desk top, Cloud, Android, iOS, Windows …)?

Identify at least 4 examples of Similar Systems.

Describe each of them, by gathering information from their websites, “*put directly copied material in quotes and italics*”. The more information gathered here will reduce the time required to go back to the websites. Reference the URL for each.

1. Bongo: [Make Your Credentialing Programs Stand Out | Bongo Learn](https://bongolearn.com/)

Bongo has a smart scoring system that generates a grade based on predetermined metrics, e.g. time of video, words per min, and clarity of speech.

1. ChatGPT: [ChatGPT](https://chatgpt.com/)

ChatGPT is an AI chatbot that can generate feedback to users, e.g. ask a question and it will answer it.

1. Gemini: [Gemini](https://gemini.google.com/app)

Gemini is an AI assistant and chatbot that is used for business and personal use.

4.CareerSet: [CareerSet - My Account](https://careerset.com/my-home)

This is the TUD CV checker. This is an example of a system, because it uses a ML system to grade CVs, similar to my system grading essays, and gives feedback to improve.

1. Identify the main system features and services provided in the reviewed systems, above. Consider the existing systems and the services they provide.

* Interface submission portal to enter documents, i.e. essays.
* Document processing system. Used to extract the text from docs into a processable format.
* ML system to grade, based on the marking scheme.
* Api, to send the essays, marking scheme, and results to an AI to generate feedback.
* AI to generate feedback and advise.
* Interface to send feedback and results to user.

[Reference and copy the URLs]

1. In what ways would users have accomplished the activity (get/use information), when not using a system, an app or online services?

Get a professor to grade their essay before the deadline and then rewrite it before the final submission.

1. Describe a new type of system, The proposed system

Explain in detail how it might operate for different end users.

Consider the existing systems that provide similar services to different end users and system adminstrators.

Take inspiration from the systems identified in section 1 and key features identified in section 2.

The proposed system is an AI-driven, automated essay grading and feedback platform designed to support students. It provides a grade, detailed feedback, and personalized recommendations for improvement, helping students refine their writing skills and allowing educators to efficiently assess work.

The platform supports document uploads in various formats, generates analytics on grading trends and areas for improvement. This system aims to serve as a comprehensive educational tool for continuous assessment and improvement of academic writing.

1. Who are the stakeholders? How would this new system affect them positively or negatively?

* University – This would be the group required to approve the app and give funding. This app would affect them by giving their students another tool to improve. Positive effect.
* Lecturers- This would be the group required to fill in the ML databases and confirm the accuracy of results. This would affect them by giving their students another tool to improve and more work to do. Possibly Negative effect.
* Students – This is the group that would benefit the most from the app and the main users. They would be the ones that would be able to test their essays on something better than a general chatbot. Positive effect.

1. What other research would be necessary to ascertain feasibility, Market Research information for marktet size e.g., ownership of smartphones …? (Gartner Research etc.)

Questionnaires to students and lecturers.

1. Make an initial list of **functional** and **non-functional** requirements.

Functional:

* Document Upload and Processing
* Automated Essay Grading
* AI API
* Feedback and Improvement Suggestions
* Local Database Management
* Error Handling and Notifications

Non – functional

* Reliability
* Performance
* Usability
* Scalability – i.e. able to handle adjustments of databases and changes to marking scheme
* Security and Privacy
* Maintainability
* Data Integrity
* Internet

**Requirements Elicitation**

1. Could observation of exisitng processes or behaviours (Ethnography) be used in this case study? If so, in what way?

No. There would be no ability to study students.

1. Identify a significant stakeholder(s), which will be **interview**ed. Do up an interview plan and pre-prepare approximately 10 questions.

Educators:

1. Current Grading Practices:

* Can you walk us through your current grading process for essays or written assignments?

1. Feedback Preferences

* What type of feedback do you typically provide to students? Is it more focused on content, grammar, structure, or something else?
* How do you ensure feedback is personalized and constructive?

1. Challenges and Pain Points

* What are the biggest challenges you face in grading essays?
* Are there common issues you observe in students’ work that you would like an AI system to help address?

1. Student Improvement and Feedback Reception

* How do students generally respond to feedback? Do they understand it, and do they usually make improvements?
* What improvements would you like to see in how feedback is delivered to students?

1. Expectations from the AI System

* What specific features or functions do you think would be most valuable in an AI grading system?
* Are there any grading aspects (e.g., argument quality, logical flow) that you think would be challenging for an AI system to assess?

1. Rubric Customization and Flexibility

* Would you prefer a system where grading rubrics are customizable for different assignments?
* How much control would you like to have over the grading criteria and feedback generation?
* Classroom Analytics and Insights

1. System Usability

* What are some usability aspects you find important for a grading tool?
* Are there any specific user interface or design elements that would make the system easier for you to use?
* Integration with Existing Tools

1. What other grading or educational tools do you currently use? Would you want the AI system to integrate with these tools?
2. Identify a significant group of stakeholders, which will receive **questionnaires**.

Justify your choice of stakeholders.

The questionnaire should have approximately 10 questions.

1. How often do you submit essays or written assignments for grading?
2. How helpful do you find the feedback you currently receive on your essays?
3. How quickly do you typically receive feedback on assignments?
4. What type of feedback do you typically receive?
5. What features would you like an AI grading system to provide?
6. What features would make you use this system?
7. Any other comments or concerns about using AI for grading?

# Requirements Analysis

1. Use the use case template to analyse the proposed system

Draw an initial *use-case diagram* with supporting scenario description for this app (possibly using *StarUML* for the diagram).

The first iteration of the use-case diagram can consist of a single overall use case with supporting main flow and 2 or 3 alternative flows.

System Name

(Automated Library System)

Actor

(Student)

The use case description is developed from analysing the description of the use case. This is the statement of the goal of the use case.

For the first iteration this will be a description of the how the system operates.

Use Cases focus on functional requirements and specific system behaviour.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE** | | <number> | <Name of Use Case>  <the name is the goal as a short active verb phrase> | |
| **Description of Goal in Context** | | <a brief statement of the goal for this Use Case in context >  <Where, when, by Whom> | | |
| **Preconditions** | | <what we expect is already the state of the system>  <list> | | |
| **Post Conditions, Success End Condition** | | <the state of the system upon successful completion> | | |
| **DESCRIPTION of Scenario** | | < The long description of the scenario to be analysed.  The use case description is a more complete description identifying behaviour, it comes from the requirements gathering> | | |
| **Main Flow** | | | | | |
| **Step** | **Action** | | | **Alternate** | |
| n.1 | <put here the steps of the scenario from trigger to goal delivery, and any clean-up after> | | |  | |
| n.2 | **<…>** | | |  | |
| n.3 | **<…>** | | |  | |
| n.4 | **<…>** | | |  | |
| n.5 | **<…>** | | |  | |
| n.6 | **<…>** | | |  | |
|  | | | | | |
| **EXCEPTIONS or ERROR Flow**  **Description** | | | | | |
| **Step** | **Branching Action**  < Exception number m of Use Case n> | | | **Alternate** | |
| n.m.1 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  | |
|  | | | | | |
| **ALTERNATIVE or VARIATION Flow**  **Description**  <condition causing alternative>  <list of variation> | | | | | |
| **Step** | **Branching Action** | | | **Alternate** | |
| n.m.1 | <Action, steps of scenario to goal delivery> | | |  | |
| n.m.2 | <Action, steps of scenario to goal delivery> | | |  | |

Non-functional Requirements for each use Case can be added in the Table below. Only consider 2 or 3 non functional requiremnts from the list in the table.

Non-functional requirements, management issues and decisions required to be made, can be identified in the following table.

From the table below, **choose a limited number of appropriate non-functional requirements** relevant to the Use Case.

Non Functional Requirements can be categorised as

* Product related
* Organisation related, process and approaches set by the company
* External, imposed by outside bodies

|  |  |  |
| --- | --- | --- |
| **RELATED INFORMATION** | Use Case: 1 | Using the system to borrow Book |
| **Priority:** | <how critical to your system/organization> | |
| **Product: Performance** | <Process and memory capacity, throughput, response time> | |
| **Product: Efficiency** | <level of memory usage, processor usage etc.> | |
| **Product: Reliability** | <likelihood of correct operation over a period of time> | |
| **Organisation: Standards** | <company standards for development, documentation etc.> | |
| **Organisation: Delivery** | <approach to deploying systems to users> | |
| **External: Legislation** | <Privacy, data protection, data retention rules, safety etc.> | |
| **External: Ethical** | <appropriate usage: is it appropriate for the target user> | |
| **Frequency** | <how often it is expected to happen> | |
| **Channels to actors** | <e.g. interactive, static files, database, timeouts> | |
| **OPEN ISSUES** | <list of issues awaiting decision affecting this use case> | |
| **Due Date** | <date or release needed> | |
| **…any other management information…** | <as…needed> | |

**More Systems Analysis**

1. Develop a second iteration in a separate word report consisting of 4 or 5 use cases.

Each use case requires a use case narrative describing the scenario analysis.

Each use case should have 2 or 3 exception or alternative flows

Student Database

Automated Library System

Student

Library Database

**Requirements Specification Matrix**

1. From the requirements analysis identified with the use Case analysis identify key functional requirements.

There should be 6 to 10 easily identifiable feature or requirements that can be listed in this matrix.

All the features (that will become use cases) identified previously need to be included, review section 7 for the additional features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Name of Req** | **Description** | **Priority** | **User Contact** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**System Modeling**

From the systems analysis and the requirements table, identify additional features and actors.

Normally this would be done through additonal Use Case models (diagrams and narratives).

At this stage the aim is just to list what would be needed to complete the model in a list by reviewing the requirements table and the systems analysis models.

1. List all the potential *actors* in this system.
2. List the potential *use-cases* in this system.

**Validation and Verification of Requirements**

1. Test Case Planning.

Develop test cases for the main use cases, the abstracted **high priority** functional requirements, identified in Iteration 2 of the Use Case Analysis.

Develop at test case for the **most important** non-functional requirements for the highest priority Use Cases.

Use the test case template to create initial Use Acceptance Test plans that will permit users and developers to agree the system will have been developed as specified by the requirements

Consider the test plan as a user guide or user manual for non-technical novice users of the system

|  |
| --- |
| **Test Case Number: 1** |
| **Test Case Name: Verify Borrow Book** |
| Related Use Case  Name: Borrow Book  Number: **1** |
| **Purpose:**  **Confirm the main flow for Borrow book** |
| **Procedure Steps:** (Guided by Main flow or other flows of Use case)   1. Tester will present valid identification 2. System indicates borrower is authenticated 3. Tester enters a valid book name, that is available 4. System indicates book is available 5. Tester asks for book 6. . . . 7. . . . |
| **Expected Results:**  **All steps worked as expected for the main flow** |

**Completing the Feasibility Study**

**Review Previous Versions**

1. Before you complete the final submission of a feasibility report, review and update the non-functional requirements, if necessary.
2. Review and, if necessary, Update the requirements Specification Matrix, section 13, and identify the high level core system features

**Update Requirement Specification (RS) & set of Use-Case Diagrams (UCD) with narratives**

1. Consider the Use case Model to ensure that core/key functionality has been addressed in the analysis and modelling process.

Add comments here on what might need to be done to address any omissions or corrections.

* 1. Do any of your use-cases need to be broken down further i.e., is there is too much functionality in one use-case?
  2. Update the potential Use Case list in section 15 as necessary.
  3. Update the Requirement Specificationtable with additional requirements as necessary.

**Prototype**

1. Create an initial prototype of the proposed system.

Such as: sketch the home page/starting page of the system. Then take a photo of it and insert the photo into the document.

**Functional & Non-Functional Test-Cases**

1. Write additional test-cases (using the test-case template) for each of three abstracted **high priority** *functional* requirements (one test-case per requirement/use case).
2. Write additional test-cases (using the test-case template) for each of the two **most important** *non-functional* requirements (one test-case per requirement).