COMPUTER AND ARTIFICIAL INTELLIGENCE

GRADUATION PROJECT

NeuraLearn Academy

Khaled Mohamad Fathy
Sherif Ahmed Shehata Hammam
Mahmoud Ahmad Zitony
Basem Yahia Abdelatif Ahmed
Karim abdelazim Mohamed shehtata
Mohamed Ashraf Mohamed Badwy
Beshoy Tag Makram

Supervisor
Dr. Gehad HASSAN

February 11, 2024

Abstract

NeuraLearnAcademy - Revolutionizing Distance Learning with AI-Enhanced Tools

In response to the persistent challenges faced by learners in online education platforms, NeuraLearnAcademy emerges as an innovative solution, pioneering a paradigm shift in the landscape of distance learning.

This electronic platform harnesses the power of advanced technologies, specifically Large Language Model (LLM) and Machine Learning (ML), to address and overcome the limitations inherent in existing online learning environments. It introduces a suite of tools and features meticulously designed to enhance the learning experience.

The platform tackles the issues of low engagement through interactive elements such as quizzes and discussions. It addresses the inadequacies of assessment methods by incorporating a range of evaluation techniques, ensuring a comprehensive understanding of the course material.

Acknowledgements

We would like to take this opportunity to express our sincere appreciation and best wishes to our devoted supervisor, Dr. Jehad, for her great leadership and ongoing motivation during the completion of this project. The assistance, and guidance she occasionally provided will help us get far along in the life's journey we are about to take.

We also take this occasion to thank Faculty of Computers and Artificial Intelligence Council for their support and work in getting us all the resources we need.

We would especially like to thank the FCAI members for the useful information they gave in their respective domains. Finally, we express our gratitude to Almighty, our parents, and everyone who has provided assistance, encouragement, and support.

Contents

1 Introduction **6**
- 1.1 Introduction . . . 6
  - 1.1.1 Problem Definition . . . 6
  - 1.1.2 The Role of Generative AI . . . 7
- 1.2 Introducing NeuraLearnAcademy . . . 7
  - 1.2.1 Problem Solution . . . 7
- 1.3 Generating Questions and Summarizing Video Transcripts . . . 8
- 1.4 Integration with Platform . . . 8
- 1.5 Motivation . . . 9
  - 1.5.1 Comprehensive Understanding . . . 9
  - 1.5.2 Personalized Knowledge Assessment . . . 9
  - 1.5.3 Summarization for Reinforcement . . . 9
  - 1.5.4 Iterative Learning Journey . . . 9
  - 1.5.5 Continuous Improvement Feedback Loop . . . 10
- 1.6 Project Limitations . . . 10

2 Project Planning **11**
- 2.1 Why Planning ? . . . 11
- 2.2 Scrum Overview . . . 11
- 2.3 Scrum Activities and Events . . . 12
- 2.4 Why Scrum? . . . 12
- 2.5 Main tasks of our project . . . 13
- 2.6 Risk Identification . . . 15
  - 2.6.1 Technology Dependencies . . . 15
  - 2.6.2 Data Security and Privacy . . . 15
  - 2.6.3 User Adoption . . . 15
  - 2.6.4 Scalability Challenges . . . 16
  - 2.6.5 Content Quality . . . 16
  - 2.6.6 Technological Obsolescence . . . 16
  - 2.6.7 Adaptability to Learning Styles . . . 16
  - 2.6.8 Integration with External Systems . . . 16
  - 2.6.9 Regulatory Compliance . . . 17
  - 2.6.10 User Technical Proficiency . . . 17

3 System Analysis **18**
3.1 System Requirements 18
3.1.1 Functional Requirements 18
3.1.2 Non-Functional Requirements 19
3.2 Exploring Trade-offs Architectures 20
3.2.1 Flexibility Across Diverse Tasks: 21
3.2.2 Optimizing for Different Data Distributions: 21
3.2.3 Enhanced Convergence and Training Speed: 21
3.3 Use Case Analysis 22
3.4 Use Case Diagram 23
3.5 Context Diagram 29
3.6 Data Flow Diagram 30
3.7 Sequence Diagram 31

List of Figures

2.1 Scrum Overview 11
2.2 Snapshot Of backlog 14
2.3 Snapshot Of sprint planning 14

3.1 Transformer 20
3.2 Proposed Model 21
3.3 UML Use Case Diagram 23
3.4 Context Diagram 29
3.5 Data flow Diagram 30
3.6 Sequence Diagram 31
3.7 Sequence Diagram 31

List of Tables

3.1 Use Case: Sign-Up 24
3.2 Use Case: Login 25
3.3 Use Case: Add Course to Cart 26
3.4 Use Case: Buy Course 27
3.5 Use Case: Add Course Material 28

Chapter 1

Introduction

1.1 Introduction

1.1.1 Problem Definition

Over the past decade, online learning has experienced significant growth, capitalizing on the fusion of the internet and education to offer individuals the opportunity to acquire new skills. The COVID-19 pandemic has further propelled online learning into a central position in people's lives, compelling educational institutions and businesses to embrace remote work and education. This surge in demand has given rise to a multitude of online learning platforms like Udemy, Coursera, Lynda, Skillshare, and Udacity, serving millions of users and evolving based on different user needs. Additionally, prestigious universities such as Stanford and Harvard are contributing to the democratization of education by providing accessible online courses spanning computer science, engineering, mathematics, business, art, and personal development.

Advantages of Online Learning

- **Efficiency**: Online learning enables teachers to efficiently deliver lessons using tools such as videos and PDFs.
- **Accessibility of Time and Place**: Students can attend classes from any location, breaking geographical barriers and reaching a broader network of students.
- **Affordability**: Online education proves to be more cost-effective compared to traditional learning methods.
- **Suits a Variety of Learning Styles**: Online learning caters to diverse learning styles, accommodating visual, auditory, and independent learners.

Disadvantages of Online Learning

- **Lack of Personal Interaction**: Online learning lacks face-to-face interaction between students and instructors, missing the dynamic of traditional classrooms.
- **Limited Hands-On Experience**: Certain disciplines require hands-on experience, which online learning may struggle to provide adequately.
- **Interactivity**: While online education offers interactive elements like discussion forums and virtual classrooms, the level of engagement may vary.
- **Assessment Methods**: Digital assessments in online education may be convenient, but traditional education often incorporates a mix of digital and traditional assessment methods.

1.1.2 The Role of Generative AI

Generative Artificial Intelligence (AI) has become a revolutionary force in education, particularly in online learning. Harnessing the advancements in technology, educators are leveraging generative AI to create personalized, engaging experiences for students. In this article, we will explore how generative AI transforms student interaction with online materials, fostering a more dynamic and effective learning environment.

1.2 Introducing NeuraLearnAcademy

The project aims to develop a new generation of learning platforms that address the challenges of online learning using Generative AI. NeuraLearnAcademy merges the power of Generative AI with an online platform, offering a solution to the drawbacks associated with traditional online education. This innovative approach aims to enhance interactivity, overcome the lack of personal interaction, and provide a more immersive and effective learning experience for students.

1.2.1 Problem Solution

Generative Artificial Intelligence (Generative AI) refers to a subset of artificial intelligence that focuses on creating and generating new content, such as images, text, or other data types. Unlike traditional AI models that rely on pre-existing data for classification or prediction tasks, generative models have the capability to generate novel and coherent output. These models learn patterns and structures from training data and use that knowledge to create new, similar content.

A language model is a type of artificial intelligence that is specifically designed to understand and generate human-like language. Language models learn the structure, grammar, and context of language from large datasets and can be utilized for various natural language processing tasks, such as machine translation, text summarization, and text completion. One notable example of a language model is OpenAI's GPT (Generative Pre-trained Transformer), which has demonstrated remarkable proficiency in understanding and generating coherent text.

1.3 Generating Questions and Summarizing Video Transcripts

Transcribing videos and extracting valuable information from them can be a powerful method for knowledge extraction. This process can be further enhanced by utilizing language model techniques for generating questions and summarizing the content.

Once you have identified key information from the transcripts, you can use language models to generate questions automatically. These questions can be crafted based on the content, aiming to cover essential topics, clarify ambiguities, or prompt further exploration. Question generation models can be trained on existing datasets or fine-tuned for specific domains. There are various approaches to summarizing textual content, and they can be adapted to video transcripts as well. Extractive summarization involves selecting the most important sentences or phrases from the transcript, while abstractive summarization generates a concise summary in the model's own words.

1.4 Integration with Platform

Integrate the trained question-answering model seamlessly into your educational platform. Design a user-friendly interface that allows users to submit their questions or feedback related to the video content. Enable the question-answer model to generate dynamic responses based on the context of the questions and feedback. The model should consider the entire transcript, relevant sections, and any additional information available to provide accurate and context-aware responses.

Implement interactive platforms or chatbots that can engage with users based on the generated questions. These platforms can provide a more dynamic and personalized learning experience.

1.5 Motivation

In today's education scene, Machine Learning isn't just a passing trend; it is a powerful tool that can fundamentally transform the way we teach and learn. By integrating Machine Learning into our project, we want to use its capabilities to revolutionize traditional educational methods and improve the overall learning experience.

1.5.1 Comprehensive Understanding

- **Goal**: Ensure every student comprehensively understands each chapter of the course.
- **How**: Implement a post-chapter feedback mechanism where students interact with ChatGPT, expressing what they've grasped and clarifying any uncertainties. This iterative process ensures a solid foundation before progressing to subsequent chapters.

1.5.2 Personalized Knowledge Assessment

- **Goal**: Tailor assessments to each student's understanding and learning pace.
- **How**: Develop a dynamic quiz model that adapts to individual progress, assessing not only factual knowledge but also the ability to connect concepts across chapters. This personalized approach enhances the learning experience and identifies areas that may need additional focus.

1.5.3 Summarization for Reinforcement

- **Goal**: Reinforce learning through concise summarization.
- **How**: Implement a summarization model that extracts key points from each chapter, generating a PDF file containing both a textual summary and a visual representation of important concepts discussed in the video. This resource aids in reinforcement and serves as a quick reference for students.

1.5.4 Iterative Learning Journey

- **Goal**: Facilitate a structured and iterative learning journey.

– **How**: Restructure the course format to unlock subsequent chapters only after the student submits an understanding summary for the previous chapter. This ensures a stepwise progression, solidifying knowledge before advancing.

1.5.5 Continuous Improvement Feedback Loop

– **Goal**: Establish a continuous improvement feedback loop.

– **How**: Gather feedback from students on the effectiveness of the ChatGPT interactions, summarization model, and quizzes. Utilize this feedback to enhance the platform, ensuring its responsiveness to the diverse needs and learning styles of users.

By aligning these goals and objectives, NeuraLearnAcademy aims to create a dynamic and adaptive learning environment that not only imparts knowledge but also actively engages students in the learning process, fostering a deeper and more sustained understanding of course material.

1.6 Project Limitations

The primary limitation lies in the accuracy of assessing students to gauge their comprehension of specific course sections. Ensuring precise evaluation of understanding within the platform is a critical focus, recognizing potential challenges in achieving absolute accuracy due to inherent complexities in subjective assessments.

Chapter 2

Project Planning

2.1 Why Planning ?

In project management, planning is crucial for successful project implementation. so we should carefully planning and choose the methodology that fit our project and specify, prioritize, manage the main project tasks, as well as identify and manage potential risks. Furthermore, effective planning can significantly enhance project performance and success rates, leading to cost and time savings. Additionally, it improves team communication, ensuring the best utilization of resources making it easier to track project goals and outcomes.

2.2 Scrum Overview

What is Scrum? Scrum is an agile way to manage work. Ever few weeks (typically two to four), teams deliver a fully functional chunk of work (an increment). Teams and the business use the feedback from each delivery to determine what to build next, or how to adapt what they've already built. Scrum works through a series of events that happen over a defined period of time: that time period is called a sprint. Sprints are short timeboxes during which the team turns ideas into working product. The events then repeat every sprint.

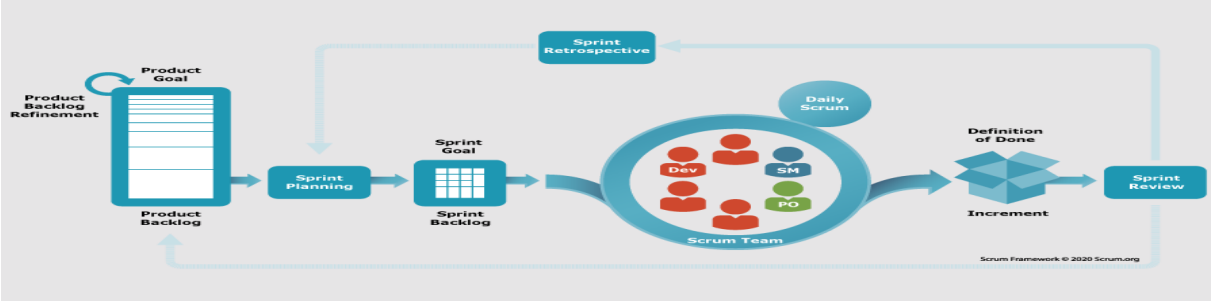

Figure 2.1: Scrum Overview

2.3 Scrum Activities and Events

- **Sprint Planning.**

  Each sprint begins with a sprint planning meeting, where the team leader presents the top items on the product backlog to the team, and team members figure out how much work they can commit to during the coming sprint.

- **The Sprint.**

  During each sprint, the team takes a small set of features from idea to fully implemented and tested functionality. At the end, these features are done and could potentially be released.

- **Daily Scrum.**

  On each day of the sprint, all team members attend a daily scrum meeting. Daily scrums are a way for team members to synchronize their work and collaborate to move that work to done. The daily meetings last no more than 15 minutes and are intended to give the team a time to share what they worked on the prior day, will work on that day, and identify any impediments to progress.

- **Sprint Review.**

  At the end of a sprint, the team conducts a sprint review during which the team demonstrates the new functionality to the stakeholder who wishes to provide feedback that could influence the next sprint. It's critical that the sprint review remains informal and doesn't become its own task, distracting from the work itself.

2.4 Why Scrum?

1. **Responsive team:** with scrum, our teams will be more responsive in their productivity, especially as changes and pivots are required. The scrum discipline requires frequent reviewing of progress, which often demands changes to prevent a project from failing.

2. **More accurate planning:** by using Scrum, our plans will be less apt to fail. Why? Because our teams are constantly putting in the effort to keep them on track by shifting and changing as needed. And because of the way scrum is designed, our teams will constantly be reflecting how things are going and can make small or large adjustments to the plans, according to the winds of change. By adhering to scrum artifacts and events, our plans are far less likely to fail.

3. **Everyone in sync:** when using scrum, a project's stakeholders are always in sync. And because the scrum methodology prioritizes individuals and interactions over all else, keeping everyone involved in sync is actually built into the process.

4. **Flexible priorities:** with scrum, it's very easy to prioritize and re-prioritize as the project moves through the process. With this ability, our developer team become more flexible and our project becomes more agile. This also makes it possible to easily (and quickly) adjust short-term goals while still adhering to the overall strategy of the project.

5. **More control:** finally, we will have more control over the entire project. That's not to say we will be able to better control our staff. No. Instead, we have more control over the direction and flow of the development process. And when we have consistent input from developers and other stakeholders, it lends a level of cohesion to the process you wouldn't otherwise have.

2.5 Main tasks of our project

- Design UI and UX
- Design database diagrams
- Implement the frontend
- Implement the backend
- Test and debug the project
- Writing the documentation

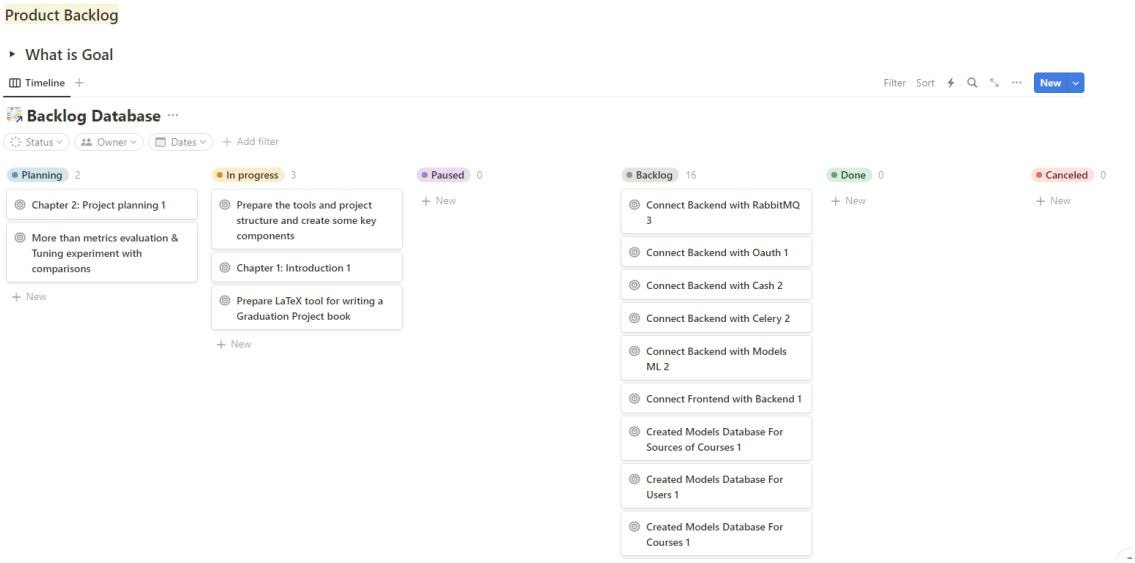

Figure 2.2: Snapshot Of backlog

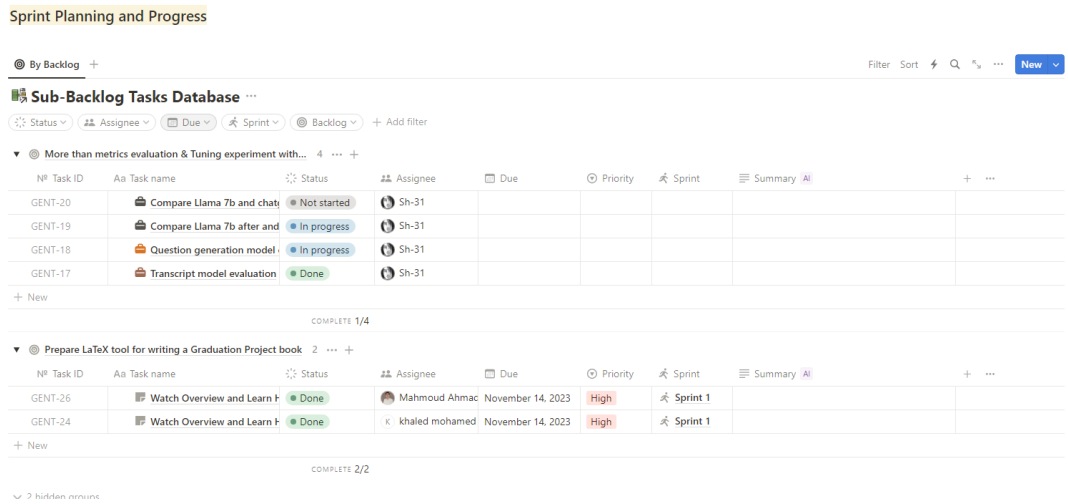

Figure 2.3: Snapshot Of sprint planning

2.6 Risk Identification

Project Risk Management includes the processes of conducting risk management planning, identification, analysis, response planning, and controlling risk on a project. The objectives of project risk management are to increase the likelihood and impact of positive events and decrease the likelihood and impact of negative events in the project. The key step in risk management planning is project risk identification. Risk identification identifies the risks that could have an impact on the project and lists their characteristics. However, as recommended, we should avoid devoting too much time to risk identification.

2.6.1 Technology Dependencies

Risk: Dependencies on third-party technologies or frameworks may undergo updates or discontinuation, impacting the platform's functionality.
Mitigation: Regularly update and maintain dependencies, conduct thorough compatibility checks, and have contingency plans for potential disruptions.

2.6.2 Data Security and Privacy

Risk: Security breaches or data privacy issues could compromise user information and erode trust.
Mitigation: Implement robust security measures, adhere to data protection regulations, and conduct regular security audits to identify and address vulnerabilities.

2.6.3 User Adoption

Risk: Users may find the platform challenging to navigate, leading to low adoption rates.
Mitigation: Implement a user-friendly onboarding process, provide clear documentation, and offer responsive customer support to ensure users can easily navigate and utilize the platform.

2.6.4 Scalability Challenges

Risk: An unexpected surge in user traffic may lead to server overload, causing system slowdowns or crashes.
Mitigation: Utilize scalable cloud services, conduct load testing, and have a scalable infrastructure in place to accommodate a growing user base.

2.6.5 Content Quality

Risk: Inaccuracies or insufficient quality in course content may impact the effectiveness of the learning experience.
Mitigation: Establish a thorough content review process, actively seek user feedback on content quality, and iterate based on user input.

2.6.6 Technological Obsolescence

Risk: Rapid advancements in technology may render certain components of the platform obsolete.
Mitigation: Regularly update the technological stack, monitor emerging technologies, and plan for phased upgrades to avoid obsolescence.

2.6.7 Adaptability to Learning Styles

Risk: The platform may not fully cater to the diverse learning styles and preferences of users.
Mitigation: Gather user feedback on learning experiences, conduct usability testing with a diverse user base, and iterate the platform design to enhance adaptability.

2.6.8 Integration with External Systems

Risk: Challenges in integrating seamlessly with external platforms or tools may hinder collaboration opportunities.
Mitigation: Conduct thorough compatibility testing with external systems, establish robust APIs, and foster collaborations through ongoing communication with potential partners.

2.6.9 Regulatory Compliance

Risk: Changes in educational or data protection regulations may necessitate adjustments to the platform.
Mitigation: Stay informed about regulatory changes, conduct regular compliance checks, and adapt the platform accordingly to ensure alignment with current standards.

2.6.10 User Technical Proficiency

Risk: Users with limited technological skills may struggle to navigate and utilize the platform effectively.
Mitigation: Provide comprehensive user guides, tutorials, and responsive customer support to assist users with varying levels of technical proficiency.

Chapter 3

System Analysis

3.1 System Requirements

A requirement is simply a statement of what the system must do or what characteristics it needs to have. During a systems development project, requirements will be created that describe what the business needs (business requirements), what the users need to do (user requirements), what the software should do (functional requirements), characteristics the system should have (nonfunctional requirements), and how the system should be built (system requirements).

3.1.1 Functional Requirements

Functional requirements are the specifications of the product's functions (features). In another words, functional requirements define what precisely a software must do and how the system must respond to inputs. Functional requirements define the software's goals, meaning that the software will not work if these requirements are not met.

1. **Authentication**

    1.1. The system will allow users to create an account.

    1.2. The system must validate users credentials to login.

    1.3. Users should have the ability to reset their passwords in case of forgotten credentials.

2. **Enroll Courses**

    2.1. Students can enroll courses and see course content We will extend this to enroll paid course wit credit card, debit card etc. in the future.

3. **Course Managment**

    3.1. Instructors can create a course and upload to our website.

    3.2. Instructors can upload various types of course content, including text, multimedia, and documents.

    3.3. Only authorized instructors should have the ability to modify or update course content.

3.1.2 Non-Functional Requirements

Non-functional Requirements define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, non-functional requirements are just as critical as functional requirements. They ensure the usability and effectiveness of the entire system. They specify criteria that judge the operation of a system, rather than specific behaviors.

- **Performance**

  The application should be able to handle large numbers of concurrent users.

- **Security**

  The application should protect user data from unauthorized access or theft.

- **Scalability**

  The application should be able to handle an increasing number of users and services.

- **User-friendliness**

  The application should be easy to use and navigate, with clear instructions and explanations of the analysis process.

- **Privacy**

  The application should have a clear privacy policy and should not retain user data

3.2 Exploring Trade-offs Architectures

- Sequence to sequence models (LSTM-GRU-RNN):
  - Difficulty in Incorporating Domain Knowledge
  - Limited Interpretability
  - Handling Varied Output Lengths
- Encoder-Decoder Transformer:
  - Transformer architecture, with self-attention mechanisms, has been widely used in Seq2Seq tasks.
  - It does come with some potential disadvantages, including high training costs.
  - Alternative way Fine-Tuning and Transfer Learning.

Model	Parameters	Jump Factor	Chinchilla Tokens (B)	Jump Factor	CS-2 Config	Days To Train	Jump Factor	Price To Train	Jump Factor	Cost Per 1M Parameters
GPT-3XL	1.3		26		4 * CS-2	0.4		$2,500		$1.92
GPT-J	6	4.6 X	120	4.6 X	4 * CS-2	8	20.0 X	$45,000	18.0 X	$7.50
GPT-3 6.7B	6.7	1.1 X	134	1.1 X	4 * CS-2	11	1.4 X	$40,000	0.9 X	$5.97
T-5 11B	11	1.6 X	*34*	0.3 X	4 * CS-2	9	0.8 X	$60,000	1.5 X	$5.45
GPT-3 13B	13	1.2 X	260	7.6 X	4 * CS-2	39	4.3 X	$150,000	2.5 X	$11.54
GPT NeoX	20	1.5 X	400	1.5 X	4 * CS-2	47	1.2 X	$525,000	3.5 X	$26.25
GPT NeoX	*20*	*1.5 X*	***400***	*1.5 X*	***16 * CS-2***	***11.1***	*0.3 X*	***$656,250***	*4.4 X*	***$32.81***
GPT 70B	70	3.5 X	1,400	3.5 X	4 * CS-2	85	1.8 X	$2,500,000	4.8 X	$35.71
GPT 70B	*70*	*3.5 X*	***1,400***	*3.5 X*	***16 * CS-2***	***21.3***	*0.3 X*	***$3,125,000***	*6.0 X*	***$44.64***
GPT 175B	175	2.5 X	3,500	2.5 X	4 * CS-2	***110.5***	1.3 X	***$8,750,000***	3.5 X	***$50.00***
GPT 175B	*175*	*2.5 X*	***3,500***	*2.5 X*	***16 * CS-2***	***27.6***	*0.3 X*	***$10,937,500***	*4.4 X*	***$62.50***

Figure 3.1: Transformer

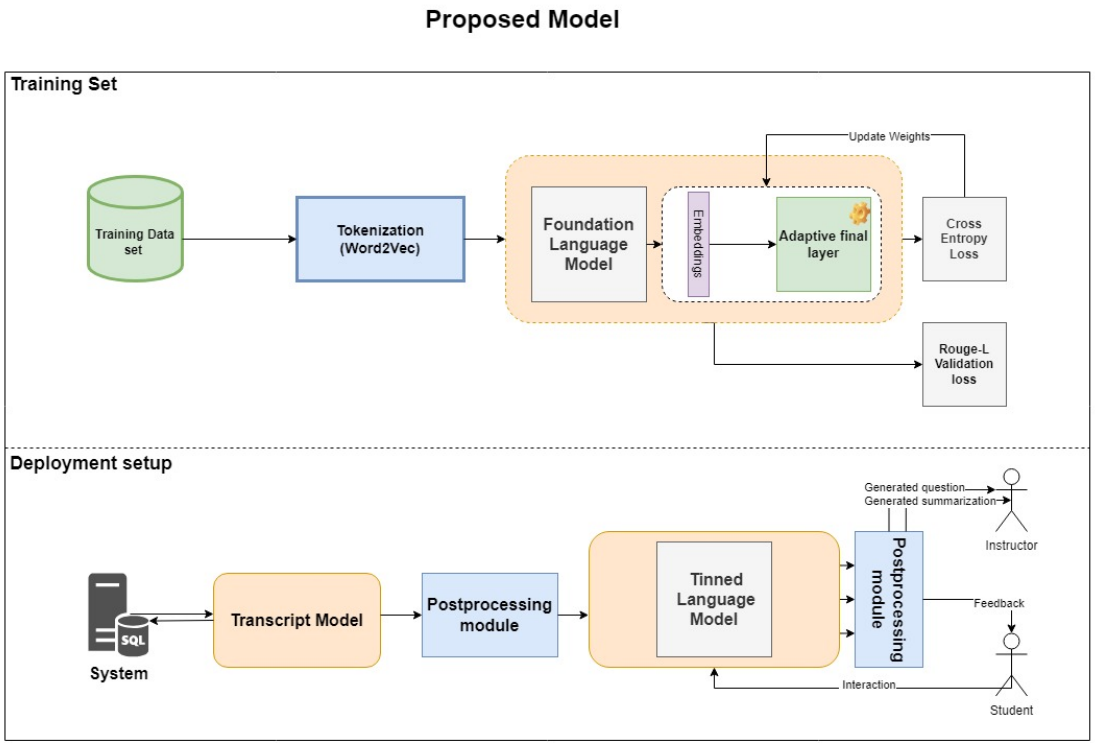

Figure 3.2: Proposed Model

The motivation behind designing a large language model with adaptive layers lies in the pursuit of creating a model that can efficiently and effectively adapt to the complexities and nuances of different datasets and tasks

3.2.1 Flexibility Across Diverse Tasks:

Adaptive layers enhance the model's flexibility to handle various natural language processing tasks. Whether it's text classification, language translation, sentiment analysis, or any other task, the adaptive layers allow the model to dynamically adjust its parameters based on the specific requirements of each task.

3.2.2 Optimizing for Different Data Distributions:

Different datasets may exhibit varying data distributions and characteristics. Adaptive layers provide a mechanism for the model to adapt its learning rates or normalization parameters dynamically, optimizing its performance for the specific patterns present in the data.

3.2.3 Enhanced Convergence and Training Speed:

Adaptive learning rate methods contribute to faster convergence during training. By adapting learning rates for individual parameters, the model can converge

more quickly, leading to reduced training times and improved overall efficiency.

3.3 Use Case Analysis

A Use Case is a type of behavioral representation that shows the interactions between a system and its users, or actors. It is used to represent and model the functionality of a system. The main purpose of a use case diagram is to capture the functional requirements of a system.

There are two formats to represent use cases:

1. Use case diagram.
2. Use case specifications or scenario in textual format.

3.4 Use Case Diagram

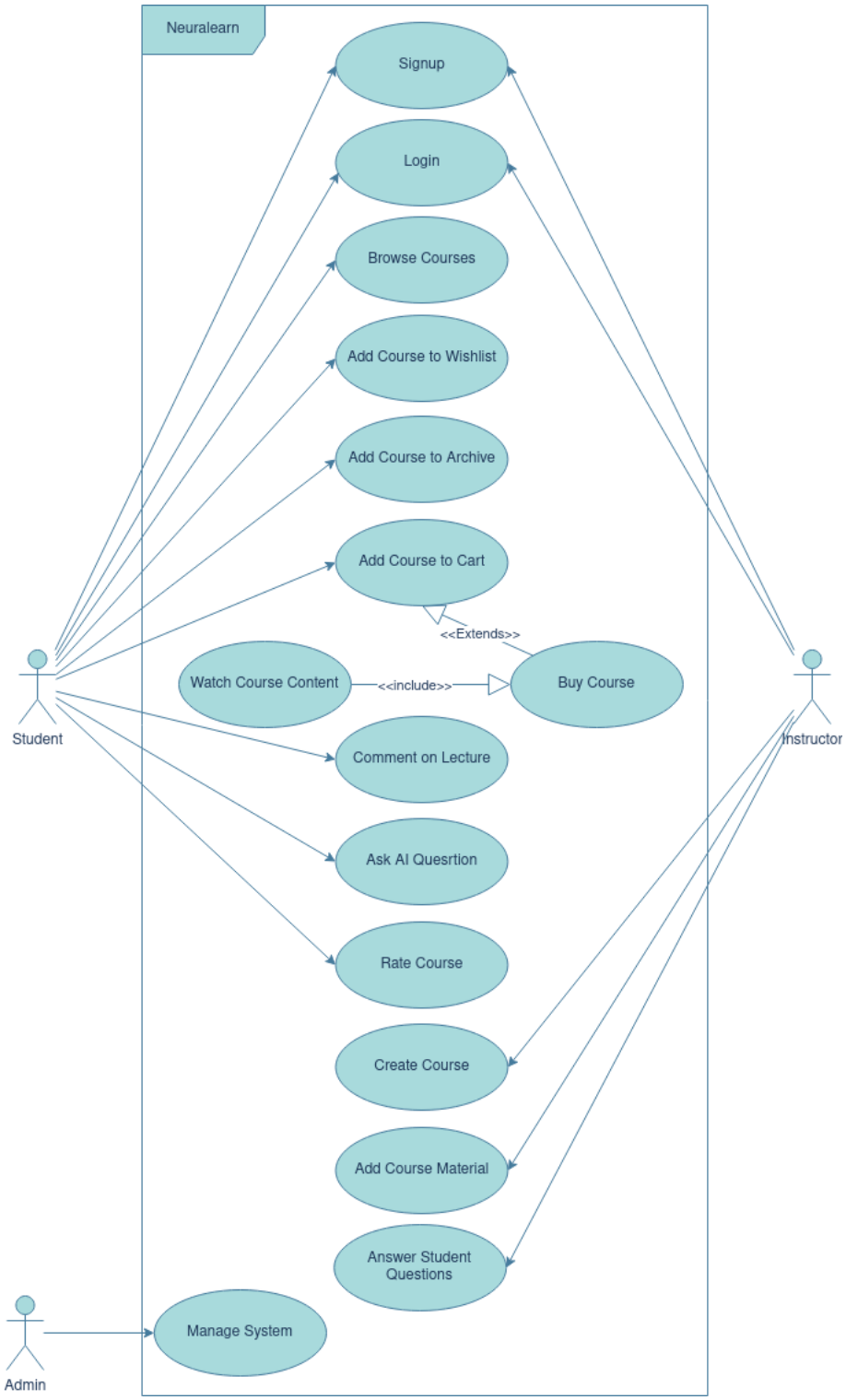

Figure 3.3: UML Use Case Diagram

Table 3.1: Use Case: Sign-Up

Use Case Name	Sign-Up
Number	1
Actor	Student, Instructor, and Admin
Preconditions	• Full name • Email address • Date of birth • Username • Password (and password confirmation)
Flow of Events	• The student clicks on the sign-up button, leading to the registration form page. • The form prompts the student to enter their full name, email address, date of birth, username, and password. • The student fills in the required information and clicks the ”Submit” button.
Exception	If the verification email is not received, the student may have the option to request a new verification.

Table 3.2: Use Case: Login

Use Case Name	Login
Number	2
Actor	Student, Instructor, and Admin
Preconditions	• The student has successfully registered an account on the platform. • The student has a valid username/email and password.
Flow of Events	• The student clicks on the sign-up button, leading to the registration form page. • The form prompts the student to enter their full name, email address, date of birth, username, and password. • The student fills in the required information and clicks the ”Submit” button.
Postcondition	• The student is successfully logged into the platform. • The platform displays the student's personalized dashboard with relevant information. • Can Browse Courses • Add Course to wishlist • Add Course to Archive • Add Course to Cart • Comment on lecture • Rate Courses • Ask Question
Exception	Invalid Credentials: If the entered credentials are invalid (e.g., incorrect password or username), the platform displays an error message and prompts the student to re-enter the information.

Table 3.3: Use Case: Add Course to Cart

Use Case Name	Add Course to Cart
Number	3
Actor	Student
Preconditions	• The student is logged into their account. • The student has navigated to the course catalogue or details page.
Flow of Events	• The student browses the available courses in the platform's catalogue. • The student selects a course they are interested in by clicking on its title or a designated "Add to Cart" button. • The platform adds the selected course to the student's shopping cart. • Optionally, the student may choose to view their cart to review the selected course and its details. • The student may choose to continue browsing and adding more courses to the cart.
Postcondition	The selected course is successfully added to the student's shopping cart.
Exception	• If the student tries to add a course to the cart that is already present, the platform recognizes this as a duplicate request. • In case of technical issues, such as server downtime or network problems, during the course addition to the cart, the platform informs the student about the problem.

Table 3.4: Use Case: Buy Course

Use Case Name	Buy Course
Number	4
Actor	Student
Preconditions	• The student has added at least one course to their shopping cart.
Flow of Events	• The student decides to proceed with the purchase. • The student views the contents of their shopping cart, confirming the courses they want to purchase. • The student clicks on the ”Proceed to Checkout” button. • The platform prompts the student to provide billing information such as credit card details or select a saved payment method. • The student reviews the order summary, including the selected course(s) and the total cost. • The student confirms the purchase by clicking on the ”Confirm Purchase” or ”Place Order” button. • The platform processes the payment using the provided billing information. • Successful payment, the platform displays a purchase confirmation message.
Postcondition	• The purchased course is now accessible in the student’s account. • The student may receive a confirmation email with details of the purchase.
Exception	• If the student attempts to purchase a course but does not have sufficient funds in their account, the platform notifies the student about the insufficient balance. • The student may be prompted to try the purchase again or choose an alternative payment method. • In the event of a technical issue or a failure in processing the payment transaction, the platform informs the student about the problem.

Table 3.5: Use Case: Add Course Material

Use Case Name	Add Course Material
Number	5
Actor	Instructor
Preconditions	• The instructive actor is logged into the platform. • The instructive actor has the necessary permissions to add course materials. • A course has been created, and the instructive actor has access to it.
Flow of Events	• The instructive actor navigates to the course dashboard or management section. • The instructive actor selects the specific course for which they want to add course materials. • Within the course management interface, the instructive actor finds and clicks on the ”Materials” or ”Content” section. • The instructive actor selects the type of material they want to add and uploads the file or provides the necessary information. • Can use AI to generate Questions or summarization for a section on the course.
Postcondition	• The course material is successfully added and available within the specified course. • Students enrolled in the course can access the added material based on the visibility settings.
Exception	• If there is an issue with uploading the material (e.g., file format not supported, network error), the system provides an error message and prompts the instructive actor to try again.

3.5 Context Diagram

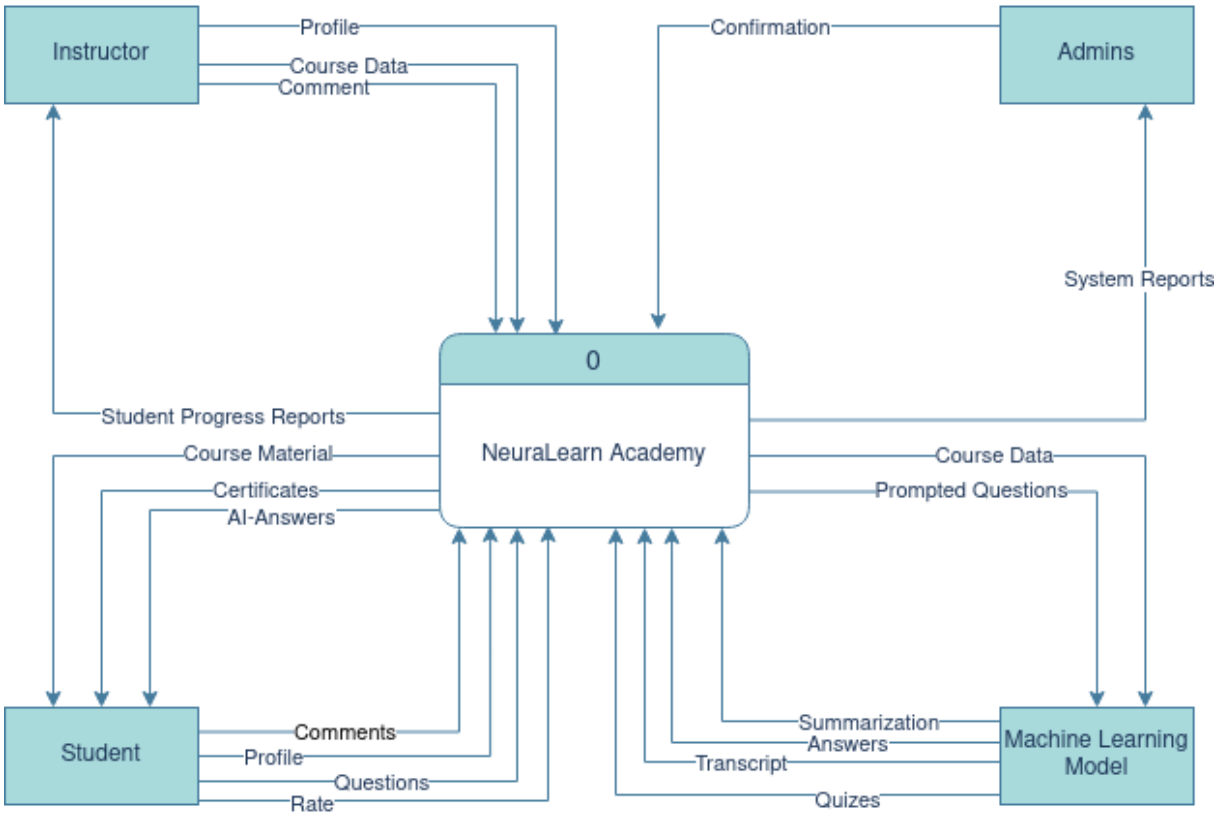

Figure 3.4: Context Diagram

3.6 Data Flow Diagram

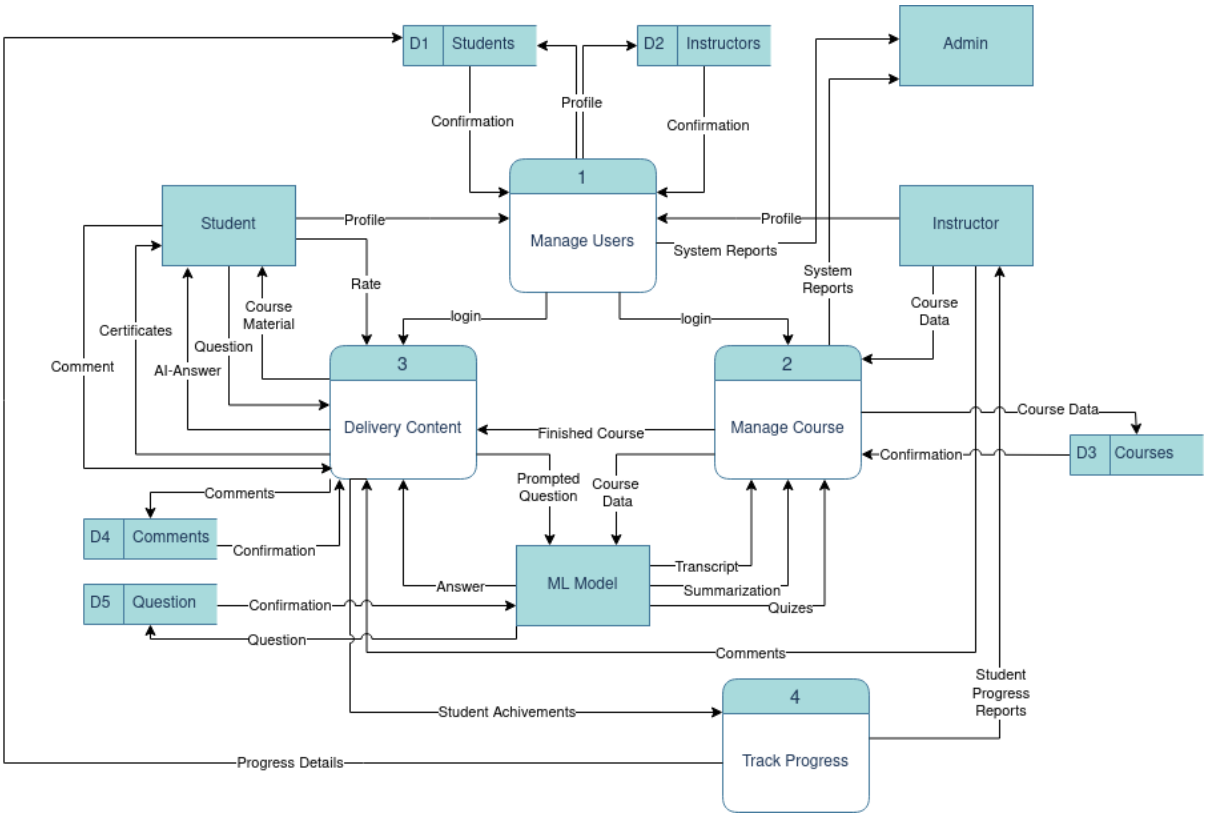

Figure 3.5: Data flow Diagram

3.7 Sequence Diagram

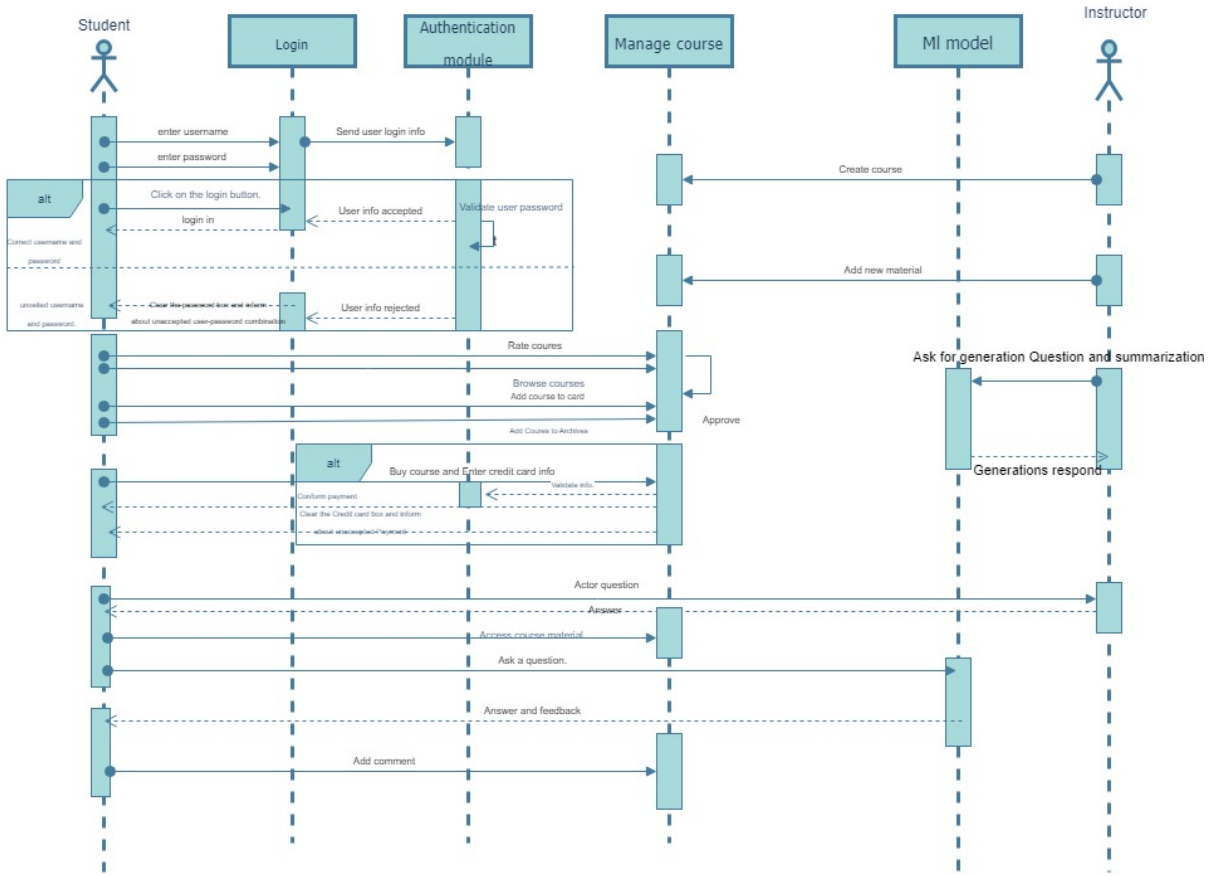

Figure 3.6: Sequence Diagram

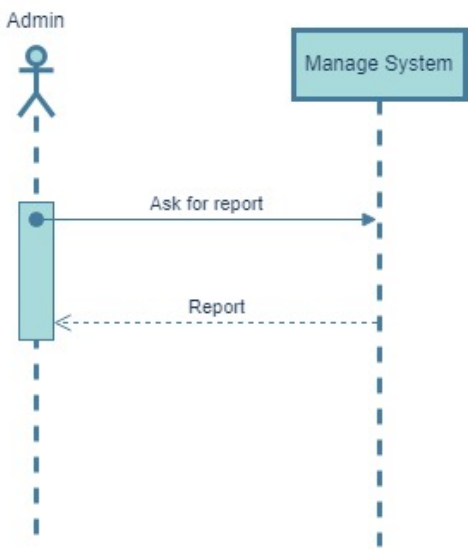

Figure 3.7: Sequence Diagram