كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

Helwan University

HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Computer Science Department

[The Learning Picnic]

A graduation project dissertation by:

[Asmaa Arby Mohamed Abdelmaged (202000135)]

[Joman Samir Salah El-Din Mohamed (202000236)]

[Shihab Wael Abdallah Metwally (202000447)]

[Omar Ahmed Abdelrahim Mohamed (202000587)]

[Majed Mamdouh Mohamed Helmy (202000685)]

[Yara Mohamed Ahmed Abdellatif (202001050)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Computer Science Department, the Faculty of Computers & Artificial Intelligence, Helwan University.

Supervised by:

[Dr. Salwa Osama]

June 2024

جامعة حلوان

كلية الحاسبات والذكاء الاصطناعي

قسم علوم الحاسب

[The Learning Picnic]

رسالة مشروع تخرج مقدمة من :

[أسماء عربي محمد عبدالمجيد (202000135)]

[چمان سمير صلاح الدين محمد (202000236)]

[شهاب وائل عبدالله متولي (202000447)]

[عمر احمد عبدالرحيم محمد (202000587)]

[ماجد ممدوح محمد حلمي (202000686)]

[يارا محمد احمد عبداللطيف (202001050)]

رسالة مقدمة ضمن متطلبات الحصول علي درجة البكالوريوس في الحاسبات والذكاء الاصطناعي، كلية الحاسبات والذكاء الاصطناعي، جامعة حلوان، قسم علوم الحاسب.

تحت اشراف :

[د. سلوى اسامة]

يونيو 2024

Table of Contents

Abstract

The Learning Picnic Project is a comprehensive web application designed to enhance the educational experience for young students by providing a range of innovative services. This platform aims to strengthen the connection between students and teachers through the use of diverse technologies. By facilitating the observation and tracking of students' learning progress, the website supports teachers and parents in monitoring educational development effectively.

Specifically tailored for schools and classrooms, the Learning Picnic Project simplifies the process for teachers to present their subjects and follow up with students. By integrating essential features needed by both teachers and students, this project ensures a robust and interactive educational environment, promoting a seamless and efficient learning experience.

Overall, the Learning Picnic Project is an invaluable resource for schools and educators dedicated to fostering a dynamic and effective learning environment for young students. Its comprehensive range of features and services makes it an essential tool for anyone striving to enhance the educational journey and development of students.

Keywords

- Transformers
- Longformer
- T5
- Bart
- GPT
- NLP
- Finetuned
- Numpy
- Hugging Face
- Pipeline
- PyPDF2
- Child_Process
- MVC
- Responsive

Acknowledgement

We would like to express our sincere gratitude to our supervisor, **Dr. Salwa Osama**, for her invaluable guidance and unwavering support throughout our project. Her expertise, deep understanding of the subject matter, and dedication to our success were instrumental in shaping the direction and scope of our work. **Dr. Salwa**'s insightful feedback, consistent encouragement, and patience greatly enriched our learning experience and propelled us toward achieving our goals. We are immensely grateful for the time and effort **Dr. Salwa** devoted to mentoring us, fostering an environment of intellectual curiosity and collaboration.

Our deepest thanks go to our **families and friends** for their unwavering support and encouragement, especially during the challenging times. Their love and belief in our abilities have been a constant source of motivation and inspiration, and we are truly blessed to have them in our lives.

We would like to acknowledge the contributions of all the **participants** who generously gave their time and insights to this study. Without their participation, this project would not have been possible.

Finally, we express our gratitude to the **Faculty of Computer and Artificial Intelligence** at Helwan University for their support and resources that enabled us to undertake and successfully complete our graduation project. The faculty's commitment to academic excellence and their guidance have been invaluable.

Thank you all for your support and encouragement.

Chapter 1 : An Introduction

In this chapter, we provide an **overview** of the project, discussing its **objective** and **purpose**, the **problem statement**, **scope** and **motivation**.

1.1 Overview

The correct education forms the foundation of a successful society, and targeting young children is crucial for them to grow up in a nurturing learning environment. With the aim of updating educational methods, we have established "The Learning Picnic Site." This platform is designed to target fourth-grade elementary students by utilizing advanced and engaging scientific content.

It aims to encourage positive engagement with this age group in a way that captures their attention, making them excited and involved in the learning process.

Our initiative recognizes the pivotal role of early childhood education in shaping a bright future, hence the focus on fourth grade also we aim to offer our educators a flexible and professional platform for managing educational content and engaging with students. Additionally, we involve parents in our operations, providing them with insight into their child's educational journey and ensuring they are informed about every stage of their development.

By putting really cool science stuff on our platform using interesting images and videos content enriched with substantive educational value, we want to get kids super interested and excited about learning. The Learning Picnic Site aspires to revolutionize educational paradigms by seamlessly blending entertainment with learning and enlightenment. Through innovative approaches, we strive to cultivate a love for learning among children, ensuring their active participation in the educational journey.

1.2 Problem Statement

The traditional educational system often fails to engage young students effectively and students practice education as it is a duty and burden on them. This lack of engagement can lead to a lack of interest in learning, which can negatively impact their academic performance and future opportunities. The current educational content and methods may not be sufficiently engaging or relevant to capture the attention of these young minds. This is especially true for scientific content, which is often presented in a way that is not relatable or exciting for children in contemporary educational settings, educators often encounter challenges in effectively managing educational content and fostering meaningful interaction with students due to the lack of user-friendly platforms tailored to their needs. Simultaneously, parents are often distanced from their child's educational journey, lacking accessible means to actively engage with their progress and developmental milestones. This fragmented approach hampers the holistic development of students and impedes the establishment of a collaborative educational ecosystem. Consequently, there exists a pressing need for a comprehensive platform that addresses these challenges by providing educators with intuitive tools, students with dynamic learning environments, and parents with transparent insights into their child's academic journey.

1.3 Motivation

Children have increasingly become capable of using the internet, with a significant number interacting with it daily, unlike in the past. This presents a valuable opportunity for us to focus on online learning and provide their educational needs electronically rather than in traditional paper formats. By doing so, we can align with their mindset and effectively capture their attention towards studying.
This approach allows us to keep pace with the societal changes and the evolving mentality and interests of children. It helps us to engage them more effectively and make learning a more appealing and relevant experience for them.

1.4 Scope of Project

The project aims to develop a comprehensive website that serves as an integrated educational platform catering to the needs of teachers, parents, and students. Its primary focus is to enhance the teaching and learning experience through innovative features and user-friendly interfaces.

1.4.1 For Teachers :

The website facilitates seamless uploading of lesson files in PDF format, allowing educators to share educational content effortlessly.
A dynamic feature enables the generation of summary previews alongside the original lesson content.
These summaries are presented in an engaging format, incorporating animations designed to capture student interest and enhance comprehension.

Teachers have the ability to create interactive quizzes for each lesson, fostering active student participation.
The platform offers diverse question formats and interactive elements to ensure an engaging quiz-taking experience. Comprehensive reports are provided to teachers, offering insights into student progress, quiz statistics, and individual question accuracy.
These reports aid educators in assessing student performance and tailoring instructional approaches to meet diverse learning needs.

1.4.2 For Parents :

The website extends access to parents, allowing them to track their child's educational progress and engagement.
Detailed reports provide parents with valuable insights into their child's academic journey, including progress updates and quiz results.
These reports enable parents to actively monitor their child's learning outcomes and provide support where needed.

1.4.3 For Students :

Students benefit from a dynamic learning environment that fosters active engagement and comprehension. The animated summary previews accompanying lesson content serve to captivate student interest and enhance understanding. Interactive quizzes offer students an opportunity to assess their knowledge in an engaging manner, with features for saving quiz results and tracking individual performance over time.

Chapter 2 : Literature Review

In this chapter, we will examine the related works that have enhanced our understanding of the problem and are closely aligned with our research. These papers serve as crucial references and have informed our approach throughout the project. Furthermore, we will offer an overview of the technologies utilized in our work.

Recently, the primary education process has become very difficult, not only for students, but also for parents and teachers who strive to communicate information to children, especially fourth-grade children. So, our team decided to focus on education problems and try to make the learning process easier by generating summarization for lessons and generating quizzes for practice.
In this section, we review existing research and projects related to automatic text summarization and quiz generation systems. We explore the methodologies, techniques, and approaches used in previous work to generate insights and inform the development of our own system.

2.1 Text Summarization

The first goal of our system is the" **Text Summarization"** Here we will present the paper that helps us in this task.

2.2.1 COLING 2022 Shared Task: LED Finetuning and Recursive Summary Generation for Automatic Summarization of Chapters from Novels

The paper explores the challenges and solutions involved in developing systems capable of generating concise and informative summaries for literary texts.
The paper outlines the task set forth in the COLING 2022 Shared Task, which involves developing systems that can automatically summarize chapters from novels. Participants are provided with a dataset consisting of novel chapters and are tasked with generating summaries that capture the essence of each chapter.
In the paper they finetuned a pretrained transformer model for long documents called **LongformerEncoderDecoder** which supports **seq2seq**

tasks for long inputs which can be up to 16k tokens in length. They use **the Booksum dataset** for longform narrative summarization for training and validation, which maps chapters from novels, plays and stories to highly abstractive human written summaries. Additionally, the paper explores recursive summary generation, where summaries are generated recursively for individual sections or paragraphs within each chapter and then combined to create a comprehensive summary. As they use a summary of summaries approach to generate the final summaries for the blind test set, in which they recursively divide the text into paragraphs, summarize them, concatenate all resultant summaries and repeat this process until either a specified summary length is reached or there is no significant change in summary length in consecutive iterations. The paper focused on the explanation of the recursive summary by explaining that The novel chapters in the blind test dataset are divided into paragraphs not exceeding 400 words in length, with an overlap of one sentence per chunk. This means that the last sentence from the previous paragraph chunk becomes the first sentence of the new paragraph chunk. If addition of any sentence to a chunk exceeds the chunk size of 400 words, that sentence becomes a part of the next chunk.
In this paper they use two datasets for finetuning, one containing just the paragraph alignments and another containing paragraph alignments along with a subset of the chapter to summary data with maximum chapter length constrained to 500 words The maximum chapter length of 500 words is chosen because the maximum encoder and decoder length for the models is set to 512 and this ensures that a very small percentage of the total number of examples exceeds the maximum token length of 512 after tokenization. Before training, all chapters and summary text are cleaned by stripping away hyperlinks, multiple consecutive whitespaces, and non-ASCII characters. Some

statistics for the train and validation splits of both datasets after tokenization using the LED tokenizer. All lengths presented in the table are the number of words in the text.
As they mentioned before They use two datasets, so they train two models on the two Datasets 1 and 2 (referred to as Model 1 and Model 2 respectively) and choose the model with the best overall validation score for final submission for the shared task. These chunks are then summarized separately and concatenated in a recursive fashion to get the final summary. We observe that the training data has a mean summary to chapter length ratio of 0.15 and a standard deviation of 0.20 (where length is considered).
The paper also explains that the reference summaries for the novel chapters are highly abstractive with high semantic and low lexical overlap. The novel chapters have long range causal and temporal dependencies that can be effectively captured by the self-attention component in transformers, which enables the network to capture contextual information. However, the memory and computational requirements of self-attention grow quadratically with sequence length, making it very expensive for longer texts like novel chapters.is a modified transformer with a self-attention operation that scales linearly with the sequence length, making it a lucrative option for processing long sequences. they use **the led-base-163843 LongformerEncoderDecoder** model for finetuning, which is ini3 https://huggingface.co/allenai/led-base-16384 tialized from bart-base4 (Lewis et al., 2019) since both models share the exact same architecture. They finetune the pretrained **LED base** model on the two datasets for 10 epochs and evaluate the validation split after every 3000 steps. This Model outputs are decoded using beam search with 2 beams and n-gram repetition blocking for n > 3. The LED config min and max length are set to 100 and 512 respectively, with a length penalty of

2.0, early stopping set to False and a batch size of 1 due to computational constraints. The maximum encoder and decoder length is set to 512. In addition to the usual attention mask, LED can make use of an additional global attention mask defining which input tokens are attended globally and which are attended only locally, just as in the case of **Longformer.** They follow recommendations of the paper and use global attention only for the very first token and they ensure that no loss is computed on padded tokens by setting their index to -100. They also disable gradient checkpointing and the caching mechanism to save memory. They use the **ROUGE metric (Lin, 2004)** for evaluation. This paper also describes the evaluation metrics used to assess the performance of participants' systems. Metrics such as **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** may be employed to measure the overlap between the generated summaries and reference summaries provided in the dataset. And It presents the results of the shared task, including the performance of participants' systems on the evaluation metrics. It may also include an analysis of the strengths and weaknesses of different approaches and insights gained from the task. The paper also states the limitation of the LED model that has computational constraints, the full power of the LED model, in which input length up to 16k tokens can be used, could not be leveraged and the encoder decoder maximum length was limited to 512 tokens. Also, the abstractive nature of the reference summaries makes lexical overlap measured by **ROUGE (Lin, 2004)** an inadequate metric for model evaluation and can be substituted with **BERTScore** or **SummaQA** which leverage pretrained neural models. It states another problem for this model which is deciding the maximum length of the final summary for input text during the recursive summary generation method, because the current value of 35% of input text length might throw away important information or incorporate information that isn’t too

relevant. Other methods apart from plain concatenation to generate a final summary using intermediate summaries can also be considered a part of future work. But after this limitation We cannot deny the benefits of this model in the summarization process.
The paper also discusses various approaches used by participants to tackle the summarization task. This includes techniques such as **LED (Large-scale Efficient Discriminative)** model finetuning, which involves adapting pre-trained language models for the summarization task.
The best model achieves a **ROUGE-1 F-1 score** of 29.75, a **ROUGE-2 F-1** score of 7.89 and a **BERT F1** score of 54.10 on the shared task blind test dataset.
The paper also states a comparison between the predicted summaries and the reference summaries yields a few important observations. One problem that the model generated summaries frequently suffer from is repetition and it also lacks coherence due to generation of summaries of independent paragraphs.
Finally, the paper discusses the implications of the findings for the field of automatic summarization and suggests avenues for future research. This may include exploring new techniques, refining existing methodologies, or addressing challenges identified during the shared task.

2.2.2 Transformer-based Models for Long Document Summarization in Financial Domain

it presents an exploration into the application of transformer-based models for the challenging task of summarizing long financial documents. As the Summarization of long financial documents is a challenging task due to the lack of large-scale datasets and the need for domain knowledge experts to create human-written summaries which was the limitations of the Traditional summarization approaches that generate a summary based on the content cannot produce summaries comparable to human-written ones and thus are rarely used in practice. So, this paper uses **the Longformer-Encoder-Decoder (LED) model** to handle long financial reports in this paper the authors describe their experiments and participating systems in the financial narrative summarization shared task. Multi-stage fine-tuning helps the model generalize better on niche domains and avoids the problem of catastrophic forgetting. They further investigate the effect of the staged fine-tuning approach on the **FNS dataset**. Their system achieved promising results in terms of **ROUGE scores** on the validation dataset. The paper begins by highlighting the importance of summarizing long financial documents, such as reports, articles, or regulatory filings, which are often dense and contain a wealth of information. It underscores the need for automated summarization techniques to assist analysts and decision-makers in extracting key insights efficiently. This paper outlines the task of summarizing long financial documents and emphasizes the unique challenges associated with this task as the financial reports are critical to a company's financial performance and provide a snapshot of its financial situation. Long documents require models capable of understanding and retaining context over extended passages of text, making traditional summarization techniques less effective.as an example the annual reports in the financial sector are typically over 180 pages long.This overload of textual data that investors and stakeholders must read is a time-consuming and exhausting process. Furthermore, in order to maximise profits, it is critical to make financial decisions in the shortest amount of time possible. As a result, automatic summarization makes use of technology to simplify the process of concisely summarizing long financial documents. Despite recent advancements in automatic summarization approaches, summarizing long financial documents remains difficult due to the lack of large-scale datasets. Furthermore, the requirement for domain knowledge experts to create human-written summaries complicates the situation. As a result, traditional summarization approaches that generate a summary based on the content cannot produce summaries comparable to human-written summaries and are thus rarely used in practice, so they try to use a different model on which this paper talking about which is transformer-based models, **such as BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer**).

The paper proposes the use of transformer-based models, such as **BERT (Bidirectional Encoder Representations from Transformers)** or **GPT (Generative Pre-trained Transformer)**, for long document summarization in the financial domain. These models have demonstrated strong performance in various natural language processing tasks and offer the potential to capture the complex relationships and nuances present in financial texts.

The paper explained the **BERT-style transformer** models that they are typically limit the sequence length to 512 tokens as they scale quadratically due to their self-attention mechanism.

To overcome this memory and computational constraint for long sequences that they faced the paper also explained, a transformer architecture that utilizes a self-attention pattern which scales linearly with the sequence length, allowing it to process long documents.

Longformer has made it easier to process long documents for natural language tasks like question answering, long document classification, and co-reference resolution.

The original Transformer architectureuses an encoder-decoder pipeline for generative sequence-to-sequence tasks like translation and text summarization and this is the most important point that our team makes use of it. Encoder-Deccoder architectures like **BART** and **T5** have achieved good results on sequence-to-sequence tasks but are not able to scale to longer sequences. **Longformer-Encoder-Decoder (LED)** has encoder and decoder transformer stacks and utilizes their efficient local and global attention pattern that can handle the long text sequence-to-sequence tasks efficiently.

This paper helps our team to make the decision to use LED as the pretrained model for our project. It describes experiments conducted to evaluate the performance of transformer-based models on the task of long document summarization in the financial domain. It may include details about the dataset used, model architectures, training procedures, and evaluation metrics employed. Results are presented, showcasing the effectiveness of the proposed approach compared to baseline methods. in this paper, they use FNS dataset its average report length in is around 80 pages. They have mainly focused on only English language summarization and formulated the task as an extractive summarization task. In the FNS training and validation datasets, each report has 3 to 7 golden reference summaries. When they examined the reports and the golden summaries, they discovered that at least one golden summary was extracted from the report as a continuous sequence of text or section they applied the same approach as Orzhenovskii (2021) and chose the summary that had at least one continuous block of text in the report and also the most intersection with other summaries as our golden summary to train their system. It takes the first 8192 tokens from the report as input and the first 1024 tokens from the selected golden summary as the target output. The system generates 1024 tokens as output predictions. **The ROUGE F1 metrics** were very low when they used the 1025 generated tokens as predicted summary because the summary length was less than 1000 words. As a result, they identify the sequence of text in the input report

that matches this generated text and choose 1000 words as the output summary.
The paper also discusses the implications of the findings for the field of financial text summarization. It may delve into the strengths and limitations of transformer-based models in this context, as well as potential areas for improvement or further research.
Finally, the paper concludes by summarizing the key findings and contributions. It may also provide recommendations for practitioners or researchers interested in leveraging transformer-based models for long document summarization in the financial domain.
Overall, "**Transformer-based Models for Long Document Summarization in Financial Domain**" contributes to the growing body of research on automated summarization techniques, offering insights and methodologies specifically tailored for the unique challenges posed by financial texts .We as a team didn't use it directly but we can't deny this paper role in help us to find the model we need .

2.2.3 LHS712EE at BioLaySumm 2023: Using BART and LED to summarize biomedical research articles

it describes the participation of the LHS712EE team in the BioLaySumm 2023 shared task, where the objective was to generate summaries of biomedical research articles. The paper introduces the BioLaySumm 2023 shared task, emphasizing the importance of automatic summarization in the biomedical domain. Biomedical research articles contain vast amounts of information, and automated summarization can aid researchers in quickly extracting key findings and insights.
The paper outlines the task set forth in the BioLaySumm 2023 shared task, which involves generating concise summaries of biomedical research articles. Participants are provided with a dataset of articles and corresponding summaries, and they are tasked with developing

systems capable of producing accurate and informative summaries, they use two datasets containing biomedical research articles and expert620 written lay summaries for Task 1, the Lay Summarization task. The first dataset contained 24,773 articles from the **Public Library of Science (PLOS) for training** and 1,376 for validation. The median length of the articles in the **PLOS training** dataset was 6,577 words. The second dataset**, eLife**, contained 4,346 articles for training and 241 for validation.The articles in **eLife** training dataset were longer, with the median length of the articles as 9,837.5. Task 2 on readability-controlled summarization utilized only.
The decomposes summarization task as two tasks which are summarization and technical abstract So there is dataset and model for each task.
Firstly, to generate lay summaries from long articles for Task 1, we selected models that follow the **sequence-to-sequence** (seq2seq) architecture they use the **Bidirectional and Auto-Regressive Transformer (BART) model. BART** combines a bidirectional encoder and an autoregressive decoder. Theu choose BART models as they have been shown to perform exceptionally well when fine-tuned for text generation tasks such as summarization and translation, as well as language comprehension tasks like text classification and question answering. For this participation, they used facebook/bart-large-xsum, the BART model implementation variant that was pretrained by Facebook/Meta specifically targeting text summarization (Facebook, 2020). The model training and evaluation was conducted on one NVIDIA GeForce RTX 4090 GPU with a mamory capacity of 24GB. To adapt BART for the specific task, they finetuned the model using the following parameters: we set the epochs to 3, number of beams to 4, maximum encoder length to 1024, maximum decoder length to 512, minimum decoder length to 100, length penalty to 2, learning rate to

1e-4, weight decay to 0.01, evaluation strategy as "steps", per-device train batch size as 4, and per-device evaluation batch size as 4. They employed several optimization techniques to fit the model into the 24GB memory limit. First, they used mixed-precision training with fp16 (16-bit floating point) to reduce memory consumption and improve training speed. They also leveraged NVIDIA's Apex library as our fp16 backend, which simplified mixed-precision training and offered additional performance optimizations. They set the "**gradient accumulation steps**" parameter to 4, which allows us to accumulate gradients from multiple mini batches before performing a single optimization step. This technique helped reduce memory usage by reducing the frequency of weight updates. Finally, they employed "**gradient checkpointing**", which trades computation time for memory by storing only a subset of intermediate values during the forward pass and recomputing them as needed during the backward pass. This technique further reduced memory usage, allowing the model to fit within the memory size limit of 24GB. To summarize, we fine-tuned the BART model for generating lay summaries of both **PLOS** and **eLife** datasets, employing optimization techniques such as **mixed-precision training, gradient accumulation, and gradient checkpointing** to accommodate memory constraints.
However, the BART model had a significant limitation that it could only handle input sequences up to 1024 tokens. As a result, the model considered only the first 1024 tokens of each article and discarded any remaining tokens. This might lead to a loss of information needed for the optimal lay summary.
It describes the approach taken by the LHS712EE team to tackle the summarization task. They leverage two state-of-the-art transformer-based models**: BART (Bidirectional and Auto-Regressive Transformers) and LED (Large-scale Efficient Discriminative).** These models are pre-

trained on large text corpora and fine-tuned on biomedical data to generate summaries that capture the key information in research articles.

Secondly In Task 2, the goal was to generate both lay summaries and technical abstracts for the articles in the **PLOS** dataset. As Tasks 1 and 2 shared the same **PLOS** dataset, they set about to test the generality of our proposed models for Task 2 as well. First, they utilized the PLOS model trained in Task 1 as-is to generate both technical abstracts and lay summaries in Task 2. After reviewing the results of it they developed a second model by retraining the model with the abstract sections of research articles in the PLOS dataset as a reference. They only retrained the BART model for Task 2 since it is more computationally efficient compared to the LED model.

To summarize their submission and trials, we can say that they selected the models for submission based on their performance in the validation phase. To evaluate each model, they gathered metrics including the training loss, validation loss, Rogue-1, Rogue-2, and Rogue-L with a min-max normalization (they made the loss negative for this normalization because the better model had the lower loss). Then, they calculate a weighted sum of the following normalized scores:0.1*training loss, 0.3* validation loss, 0.2*Rogue-1, 0.2*Rogue-2, and 0.2*Rogue-L44We submitted two runs for Task 1. The first submission was generated from the two fine-tuned BART models. For the second (final) submission, the lay summaries for the PLOS dataset were generated from a fine-tuned LED model and those for the eLife dataset were generated from a fine-tuned BART model. For Task 2, the first submission was generated directly from the PLOS BART model used in Task 1. For the second submission, they retrained the PLOS BART model with the abstract sections of articles in the PLOS dataset as a reference.

So, the paper presents the experimental setup used by the “LHS712EE

team", including details of the datasets, model architectures, and evaluation metrics employed. Results from their experiments are provided, demonstrating the effectiveness of their approach in generating high-quality summaries compared to baseline methods.
It discusses the implications of their findings for the field of biomedical text summarization. It may delve into the strengths and limitations of using transformer-based models like BART and LED for this task, as well as potential areas for further research and improvement.
This paper not only explains the results, but also evaluates them as they explain that the submissions for the shared tasks are evaluated on three aspects – relevance, readability, and factuality. Relevance is assessed using Rouge-1, Rouge2, Rouge-L, and BERTScore metrics. Readability is evaluated using two measures: Flesch-Kincaid Grade Level (FKGL) and Dale-Chall Readability Score (DCRS). Factuality is measured using BARTScore that was fine-tuned by the organizers. Better systems have higher relevance and factuality scores and lower readability scores.
Finally, the paper concludes by summarizing the key findings and contributions of the LHS712EE team. It may also reflect on the challenges encountered during the shared task and offer insights for future work in biomedical text summarization. In the end they find that the performance of the two submissions for Task 1 was not significantly different from one another. But in the second task the performance of the model significantly improved compared to the first submission. After min-max normalization, the final scores for the second (final) submission for Task 2 in the three aspects have been improved.
Overall, "**LHS712EE at BioLaySumm 2023: Using BART and LED to summarize biomedical research articles** transformers effective approach to automatic summarization in the biomedical domain, highlighting the potential of transformer-based models to generate

informative and concise summaries of complex scientific literature but as everything it also has limitation as the team of this paper states that large language models still demand considerable time and memory resources, which remains a limitation of their work .

2.2.4 Longformer: The Long-Document Transformer

presents the Longformer model, which is specifically designed to handle long documents more effectively within the transformer architecture to overcome the limitation of the Transformer-based models that are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length.
The paper introduces the **Longformer architecture**, which extends the transformer model to handle long documents more efficiently. It achieves this by introducing a combination of global attention and local attention mechanisms, allowing it to capture dependencies over long ranges while maintaining computational efficiency.the transformers have achieved state-of-the-art results in a wide range of natural language tasks including generative language modeling and discriminative language understanding due to their self-attention which enables the network to capture contextual information from the entire sequence. While powerful, the memory and computational requirements of self-attention grow quadratically with sequence length, making it infeasible (or very expensive) to process long sequences. To address this limitation so they explain in the same paper that they present aLongformer, which is a modified Transformer architecture with a self-attention operation that scales linearly with the sequence length, making it versatile for processing long documents This is an advantage for natural language tasks such as long document classification, question answering (QA), and coreference resolution,

where existing approaches partition or shorten the long context into smaller sequences that fall within the typical 512 token limit of **BERT-style** pretrained models. Such partitioning could potentially result in loss of important cross-partition information, and to mitigate this problem, existing methods often rely on complex architectures to address such interactions. On the other hand, the proposed Longformer can build contextual representations of the entire context using multiple layers of attention, reducing the need for task-specific architectures.

Longformer's attention mechanism is a drop-in replacement for the standard self-attention and combines a local windowed attention with a task motivated global attention. Following prior work on long-sequence transformers, in this paper the team evaluate Longformer on character-level language modeling and achieve state-of-the-art results on **text8** and **enwik8**. In contrast to most prior work, they also pretrain **Longformer** and finetune it on a variety of downstream tasks. Their pretrained Longformer consistently outperforms **RoBERTa** on long document tasks and sets new state-of-the-art results on WikiHop and **TriviaQA**. They finally introduced **the Longformer-Encoder-Decoder (LED)**, a Longformer variant for supporting long document generative sequence-to-sequence tasks and demonstrate its effectiveness on the **arXiv** summarization dataset.it's attention mechanism is a combination of a windowed local-context self-attention and an end task motivated global attention that encodes inductive bias about the task. Through ablations and controlled trials, – the local attention is primarily used to build contextual representations, while the global attention allows Longformer to build full sequence representations for prediction.

the paper introduces a variant of Longformer which instead of an encoder-only Transformer architecture, it follows an encoder-decoder architecture similar to the original Transformer model, and it is intended for sequence-to-sequence (seq2seq) learning. This model is **Longformer-Encoder-Decoder (LED)**.

This paper also explained some attention pattern such as sliding window, detailed sliding window and global attention, they also explained the implementation of them and the longformr and the impact of using them on the Longformer. In regular transformers, attention scores are computed as the expensive operation is the matrix multiplication QKT because both Q and K have n (sequence length) projections. For Longformer, the dilated sliding window attention computes only a fixed number of the diagonals of QKT. this results in a linear increase in memory usage compared to quadratic increase for full self-attention. However, implementing it requires a form of banded matrix multiplication that is not supported in existing deep learning libraries like PyTorch/Tensorflow.

The team of this paper states that they tried different variants and report their controlled experiment results. To make the ablation study more manageable, they train each configuration for 150K steps4 with phase 1 configuration on a small model on text8, then report the BPC performance on the dev set. Then they explain the impact of different ways of configuring the window sizes per layer. As a result, they observe that increasing the window size from the bottom to the top layer leads to the best performance, arranging them in the reverse way leads to worse performance, and using a fixed window size (the average of window sizes of the other configuration) leads to a performance that it is in between. The bottom of Tab. 4 shows the impact of adding dilation. Adding some dilation to two heads leads to some improvement compared with no dilation at all.

They aimed to proof that the transformers, especially the Longformer overcomes all the limitations of the models before, so they mention all their experiments.

In this paper they state that The original Transformer consisted of an encoder-decoder architecture, intended for sequence-to-sequence tasks such as summarization and translation., pre-trained encoderdecoder Transformer models (e.g. BART and T5) have achieved strong results on tasks like summarization. Yet, such models can't efficiently scale to seq2seq tasks with longer inputs. To facilitate modeling long sequences for seq2seq learning, they propose a Longformer variant that has both the encoder and decoder Transformer stacks but instead of the full self-attention in the encoder, it uses the efficient local+global attention pattern of the Longformer. The decoder uses full self-attention to the entire encoded tokens and to previously decoded locations. call this model **Longformer-Encoder-Decoder (LED)**. They didn't initialize new parameters (Since pre-training LED is expensive), but they use parameters from the BART, and follow BART's exact architecture in terms of number of layers and hidden sizes. The only difference is that to process longer inputs, they extend position embedding to 16K tokens (up from BART's 1K tokens) and we initialize the new position embedding matrix by repeatedly copying BART's 1K position embeddings 16 times as in Section 5 for RoBERTa. Following BART, they release two model sizes, LED-base and LED-large, which respectively have 6 and 12 layers in both encoder and decoder stacks. They evaluate LED on the summarization task using the **arXiv** summarization dataset which focuses on long document summarization in the scientific domain. The 90th percentile of document lengths is 14.5K tokens, making it an appropriate testbed for evaluating LED. LED's encoder reads the document, and its decoder generates the output summary. The encoder uses local attention with window size 1,024 tokens and global attention on the first token. The decoder uses full attention to the entire encoder and previously decoded locations. Longformer also play a significant role in making questions and

generating answer for them (Quiz)but we as a team didn’t pay an attention for this part as it didn’t meet our need in generating quiz task. Finally, the paper addresses the challenge of processing long documents with traditional transformer models, which struggle due to their self-attention mechanism's quadratic complexity. It explains the need for a model like **Longformer**, highlighting the prevalence of long documents in various domains, such as legal documents, scientific articles, and news articles.It also describes experiments conducted to evaluate the performance of **Longformer** on various tasks, including document classification, question answering, and summarization. Results demonstrate that **Longformer** outperforms traditional transformer models on long-document tasks while maintaining comparable performance on shorter documents.The paper discusses the implications of **Longformer's architecture** and its potential applications in real-world scenarios. It also addresses limitations and future directions for research, such as optimizing the model's performance further and exploring additional applications.
Overall, **"Longformer: The Long-Document Transformer"** presents an innovative solution to the problem of handling long documents within the transformer architecture, offering promising results and paving the way for future advancements in natural language processing tasks involving lengthy texts. The paper concludes by summarizing the key contributions of **Longformer** and its significance in addressing the challenges of processing long documents with transformer-based models.
This paper helps us indirectly as we didn’t use this transformer in our project but we leveraged from it to choose the best way and the best model to make our tasks.

2.2 Generating Quizzes

Here we come to the second goal of our system is the"**generating quizzes**" here we will present the paper that helps us in this task.

2.3.1 MQAG: Multiple-choice Question Answering and Generation for Assessing Information Consistency in Summarization

it introduces a novel approach for assessing the consistency of information in text summarization through multiple-choice question answering (MCQA). It introduces the concept of MCQA to evaluate information consistency. This paper proposes a Multiple-choice Question Answering and Generation framework, MQAG, which approximates the information consistency by computing the expected statistical distance between summary and source answer distributions over automatically generated multiple-choice questions. This approach exploits multiple-choice answer probabilities, as predicted answer distributions can be compared. This paper conduct experiments on four summary evaluation datasets: **QAG-CNNDM/XSum, XSum-Hallucination, Podcast Assessment,** and **SummEva**l. Experiments show that MQAG, using models trained on **SQuAD** or **RACE**, outperforms existing evaluation methods on the majority of tasks.In this work, a measure of consistency between the source and summary is defined from an information-theoretic perspective. They propose a Multiple-choice Question Answering and Generation framework, MQAG, where instead of comparing text-based answer spans, multiple-choice questions are generated and the resulting answer distributions from the source and summary are compared. They represent the answers to the generated questions via probability distributions instead of lexical

or embeddings.
The paper presents the MQAG (Multiple-choice Question Answering and Generation) architecture, which consists of a question generation module and a question answering module. The question generation module generates multiple-choice questions based on the content of the summary, while the question answering module evaluates the consistency of the summary by answering these questions.A question-answering approach consists of a question-generation model and an answering model. Given automatically generated questions, the first answer is derived from the source and the second answer is derived from the evaluated summary, and then the two answers are compared, and it gives many examples about this.
The paper also explains the motivation behind using MCQA for assessing information consistency, highlighting the advantages of this approach over traditional evaluation methods. MCQA provides a more objective and structured way to assess consistency by formulating questions based on the content of the summary.
It describes experiments conducted to evaluate the effectiveness of MQAG in assessing information consistency in text summarization. Results demonstrate that MQAG outperforms baseline methods in detecting inconsistencies and provides valuable insights into the quality of summaries.
The paper illustrates how to generate the quiz and its answers, it decomposes it into First model G1 generates the question q and answer a, then model G2 generates the distractors (the wrong answers). Both G1 and G2 are sequence-to-sequence T5-large models (Raffel et al., 2020). The question-answer generation system G1 is fine-tuned to either RACE or SQuAD, and the distractor generation system G2 is fine-tuned to RACE. The answering stage contains one model A, which is Longformer-large with a multiple-choice setup. The input to the model

is a concatenation of context, question, and option. The answering model A is also fine-tuned to RACE they also pay attention for the quality of the generated questions, so they consider filtering out low-quality questions through question-context answerability measures consider a simple answerability measure based on the entropy of the probability distribution over the options.
It also discusses the implications of using MCQA for assessing information consistency and its potential applications in text summarization and related tasks. It also addresses limitations and future directions for research, such as refining the question generation process and exploring additional evaluation metrics.
Finally, the paper concludes by summarizing the key contributions of MQAG and its significance in improving the evaluation of text summarization systems. It highlights the importance of assessing information consistency and the role of MCQA in achieving this goal.
Overall, "**MQAG: Multiple-choice Question Answering and Generation for Assessing Information Consistency in Summarization**" offers a novel approach to evaluating the quality of summaries by leveraging multiple-choice questions and answers, providing a valuable tool for researchers and practitioners in the field of natural language processing. We as the projects team leveraged from this paper as we use the T5 model and the same datasets that have been explained in this paper in our generating quiz task of our project.

Chapter 3 : The Proposed Solution

In this chapter, we embark on a journey through the intricate design process of our Learning Picnic System, addressing the challenges highlighted in the preceding chapters.

Our tasks primarily involve Natural Language Processing (NLP), as detailed in Chapter 2 where we discussed the research papers guiding our work. Our focus predominantly revolves around Transformer-based models, which form the backbone of our methodologies. We meticulously select the most appropriate model for each task, alongside identifying the most suitable dataset. While not every task employs the Transformer model architecture, it serves as the foundation of our approach.

3.1 Transformers

The Transformer architecture has significantly impacted the field of natural language processing (NLP) since its introduction by Vaswani et al. in the paper "Attention is All You Need" in 2017. It revolutionized the way neural networks handle sequential data, particularly in tasks like language translation, text generation, and sentiment analysis. Here's a breakdown of its key components:

1- Self-Attention Mechanism: The core innovation of the Transformer is the self-attention mechanism. It allows the model to weigh the importance of different words in a sequence when encoding or decoding information. Self-attention calculates the attention scores between all pairs of words in a sequence and generates context vectors that capture the dependencies between words.

2- Encoder-Decoder Architecture: The Transformer model consists of two main parts: an encoder and a decoder. The encoder processes the input sequence, while the decoder generates the output sequence. Each of these parts comprises multiple layers of self-attention mechanisms and feed-forward neural networks

3- Multi-Head Attention: To enhance the representational capacity of the model, self-attention is applied multiple times in parallel, with different learned linear projections of the input. These parallel self-attention mechanisms are called "heads." Multi-head attention allows the model to focus on different parts of the input simultaneously and learn more complex patterns.

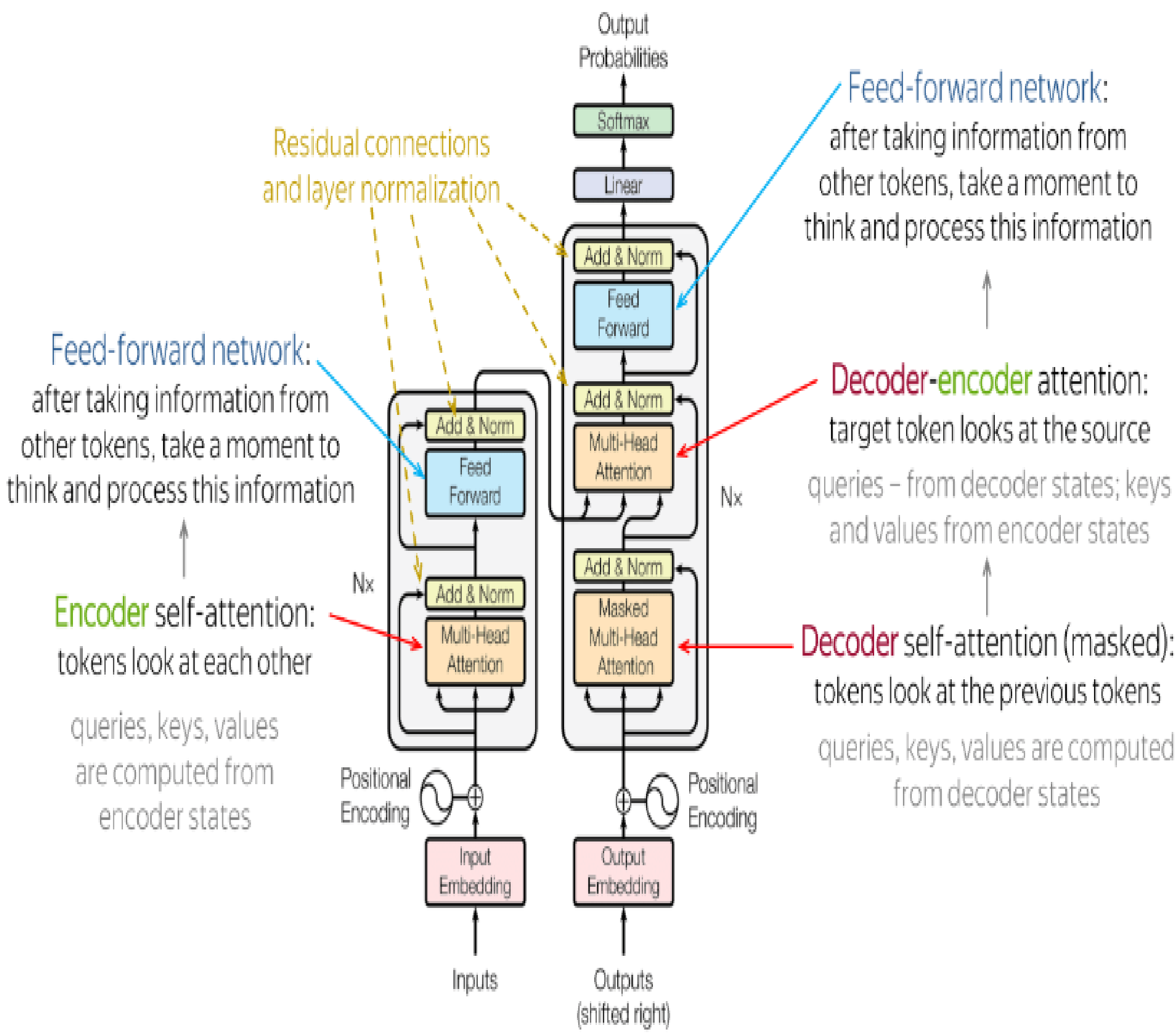
4- Positional Encoding: Since the Transformer model lacks recurrence or convolution mechanisms, it needs a way to incorporate information about the order of words in a sequence. Positional encoding is added to the input embeddings to provide the model with positional information, enabling it to understand the sequential nature of the data.

5- Feed-Forward Neural Networks: In addition to self-attention mechanisms, each layer of the Transformer model contains fully connected feed-forward neural networks. These networks apply a non-linear transformation to the output of the self-attention layer, helping the model learn complex patterns in the data.

6- Layer Normalization and Residual Connections: To facilitate training and improve gradient flow, layer normalization and residual connections are applied after each sub-layer in the Transformer model. Layer normalization normalizes the activations of each layer, while residual connections pass the input directly to the output of the sub-layer, aiding in the flow of gradients during training.

7- Decoder Masking: During training: the decoder is provided with the entire target sequence, but during inference, it generates the output sequence one token at a time. To prevent the model from attending to future tokens during training, a masking mechanism is applied to the decoder's self-attention layer.

Overall, the Transformer architecture has demonstrated state-of-the-art performance in various NLP tasks and has become the foundation for many subsequent advancements in the field, including models like BERT, GPT, and T5.



3.2 Text Summarization Model

3.2.1 BART Model (Experiment 1)

Recognizing the significance of Transformer models, we initially adopted the Bart model coupled with the CNN/DailyMail dataset as our primary choice. While it demonstrated satisfactory performance, it exhibited limitations such as truncating lessons and producing less efficient summaries that may overlook key information. Despite these drawbacks, in the absence of an alternative, it would have remained our preferred option. Additionally, the structure of the fine-tuned dataset was found to be unsuitable for our specific task requirements.

3.2.1.1 Results

Adaptations are characteristics that help living organisms survive and reproduce in the ecosystem in which they live .
For example, thick, white fur is an adaptation in polar bears, which helps them stay warm in their cold, Arctic home .
The path of blood vessels in a penguin's feet help its feet stay warm.
Blood vessels bring cold blood up from the feet.
Other blood vessels bring warm blood down to the feet from the feather-coated body.
Where they touch, the warm blood vessels can then heat up the cold blood vessels.
This means the blood traveling up into the body is not cold, and blood flowing down the toes is warm enough to keep their toes from freezing.
How do penguins' feet help them survive in cold climates? and give other example of animals that live in different cold environment.
The big ears on a fennec fox help it stay cool.
Then, write two questions you have.
Read the text about another type of adaptation

3.2.2 Longformer-Encoder-Decoder (LED)

The **Longformer-Encoder-Decoder (LED)** model represents a significant advancement in the realm of natural language processing, particularly in the domain of summarization tasks. It was developed to address the inherent limitations of traditional Transformer architectures when dealing with long sequences of text.

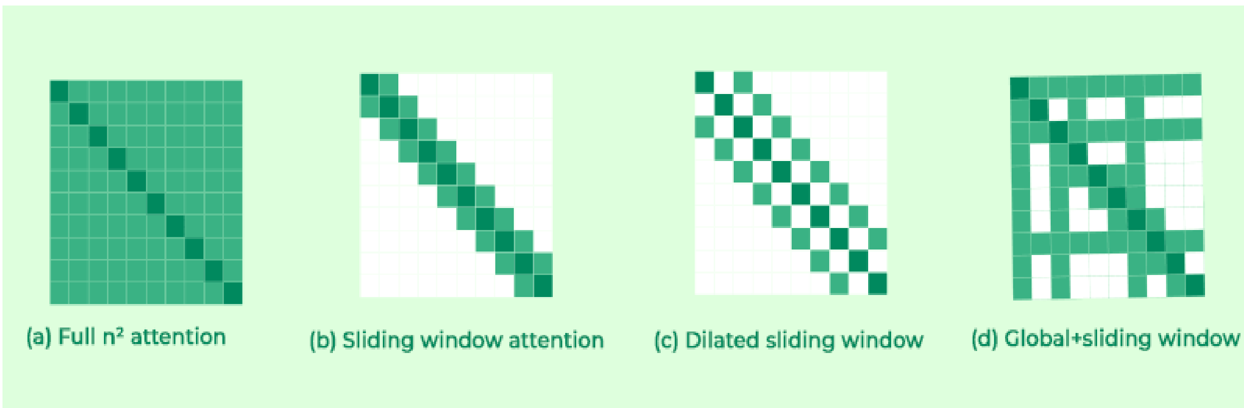
One of the primary challenges in processing long documents with Transformer-based models is the **quadratic complexity** of the attention mechanism, which results in high memory and computational requirements. To mitigate this issue, LED introduces a novel attention mechanism that **combines global attention with sparse local attention**.

Global attention allows the model to consider the entire input sequence, enabling it to capture long-range dependencies and understand the overall context of the document. However, applying global attention to every token in the sequence is computationally expensive, especially for lengthy documents.

To maintain efficiency, LED incorporates **sparse local attention**, which focuses on a subset of tokens within a fixed window around each position. This local attention mechanism enables the model to capture fine-grained details and context from nearby words while significantly reducing the computational cost associated with processing long sequences.

By striking a balance between global and local attention, LED achieves superior performance in summarization tasks compared to traditional Transformer models. It can effectively handle documents of varying lengths, generating accurate and coherent summaries that capture both the essence of the content and the nuances of the language used.

Overall, the attention mechanism employed in **the Longformer-Encoder-Decoder** model represents a breakthrough in the field of natural language processing, paving the way for more efficient and effective processing of long documents in a wide range of applications.



3.2.2.1 Results

In this first lesson, the narrator explains how different types of animals and plants adapt to different climates and make certain adaptations in order to survive.
He uses examples such as a desert lizard that finds shade during a hot sunny day and discusses how lizards have colorful scales that make them hard to see among the rocks.
He also uses the example of the penguin to demonstrate how certain adaptations help an animal survive in a cold climate.
Next, he demonstrates how penguins feet keep warm in extremely cold temperatures by using blood vessels in their feet as an example of an adaptation.
The goal of this lesson is to learn about how various forms of adaptation are related to each other so that one part of an organism can be used to benefit another part of the body.
For example, if you could stand on an icy sheet of ice in bare feet, you would lose feeling in your feet after only two minutes; for penguins, however, they have no feathers on their feet and can stand on the ice all day without losing feeling in their toes.
This is very important because penguins cannot fly.
Later in the same lesson, we discuss how the big ears on a fox help it stay cool and how the path of blood vessel in a penguin helps its feet stay warm.
Each activity in this lesson focuses on trying to apply lessons from this lesson to real-world applications in real-life scenarios.
In this case, talking with a scientist might be the best way to find out about penguins' ability to survive in extreme climates.
Here, for example, we look at how Arctic polar bears use their thick white fur to blend in with the snow as they go hunting and also dark fur helps brown bears live in temperate environments and light fur helps desert animals like brown bears blend in to their natural landscape.
Finally, we turn our attention to pale lizards, which have colorful colored scales on their hideously colorful lizards.
What kind of camouflage do these lizards need to camouflage themselves? And why do bighorn sheep need to wear different colors of fur depending on the season? These activities end with a discussion of how different kinds of animals cope with different seasons and lengths of captivity.
Following this lesson, students should always try to understand how other people deal with difficult or extreme situations and come to the same conclusion: "Adaptation and Survival" is a general term for dealing with problems in the natural world.
It's not just about being able to talk about things you're not sure about; it can also refer to specific life skill areas.
An example of this type of learning is listening to a scientist who knows a lot of questions.
When scientists learn something new, they may start asking more questions until they get to the point of answering those questions.
A penguin, for instance, doesn't have any feathers on its feet, but it can stick around on an ice patch all day even though it lives in Antarctica, which is one of the coldest places on the Earth.
Why? Because there are blood vessels moving through its feet.

3.2.3 BookSum Dataset

The BookSum dataset is a curated collection of book summaries, designed to facilitate natural language processing tasks such as text summarization, sentiment analysis, and topic modeling. Each entry in the dataset typically includes key information about a book, such as its title, author, summary, and potentially additional metadata.

Offering a rich source of book summaries and metadata, empowering researchers and developers to explore and analyze textual content in diverse ways. Whether for academic research, algorithm development, or industry applications, the dataset provides valuable insights into the world of literature and language.

3.3 Generating Quizzes

3.3.1 T5 Model

T5 (Text-to-Text Transfer Transformer) excels in generating quizzes by converting various NLP tasks into a text-to-text format. It leverages transfer learning, allowing it to understand and generate human-like text. T5's flexibility and pre-training on diverse datasets enable it to create relevant, context-aware quiz questions. Its performance in generating quizzes is enhanced by fine-tuning on specific educational data, ensuring accuracy and coherence, we used the t5 model twice to generate the MCQ quizzes.

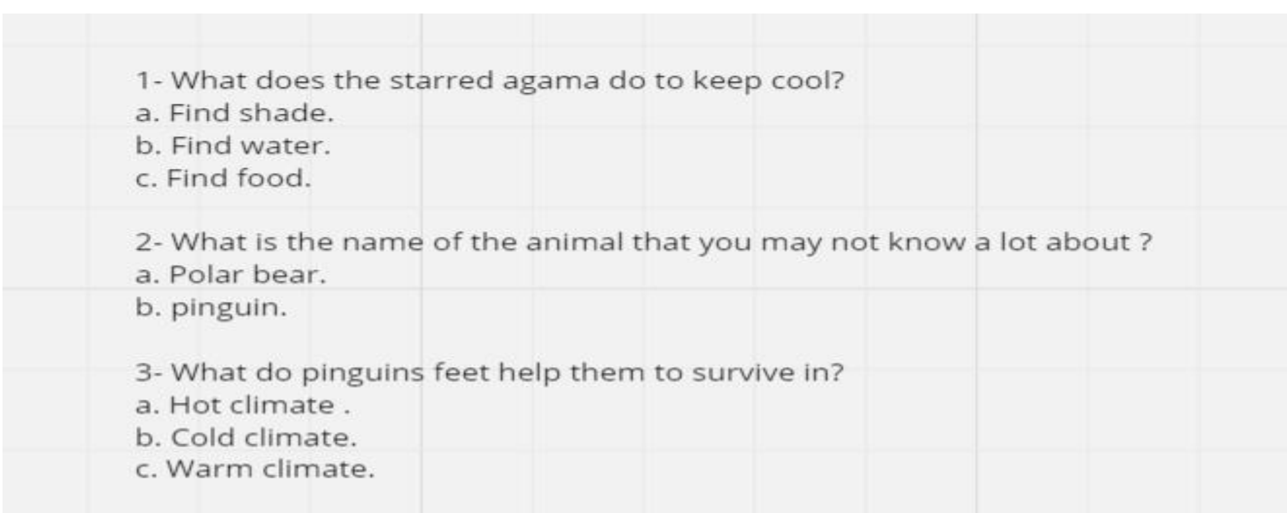
3.3.1.1 Questions Answering

Firstly, the T5 model is fine-tuned on a specific dataset to generate and answer questions from given paragraphs. In this phase, the model creates a list of questions along with model answers containing the correct responses to these questions, using SQuAD dataset.

3.3.1.2 Distractors

Secondly, using the generated questions, answers, and the original paragraphs, we can create distractors (incorrect choices) to form a complete multiple-choice quiz, using RACE dataset.

3.3.2 Results


1- What does the starred agama do to keep cool?
a. Find shade.
b. Find water.
c. Find food.

2- What is the name of the animal that you may not know a lot about ?
a. Polar bear.
b. pinguin.

3- What do pinguins feet help them to survive in?
a. Hot climate .
b. Cold climate.
c. Warm climate.

3.3.3 Datasets

3.3.3.1 SQuAD Dataset

The Stanford Question Answering Dataset (SQuAD) is a widely used benchmark dataset in the field of natural language processing, specifically designed for the task of machine reading comprehension. It consists of a large collection of question-answer pairs, created from Wikipedia articles, where each question is paired with a corresponding answer span within the text.
The SQuAD dataset plays a pivotal role in advancing the field of natural language understanding, serving as a standard benchmark for evaluating and comparing machine reading comprehension models. Its rich collection of question-answer pairs derived from real-world text enables researchers to push the boundaries of language understanding and build more intelligent systems capable of comprehending and responding to human queries.

3.3.3.2 RACE

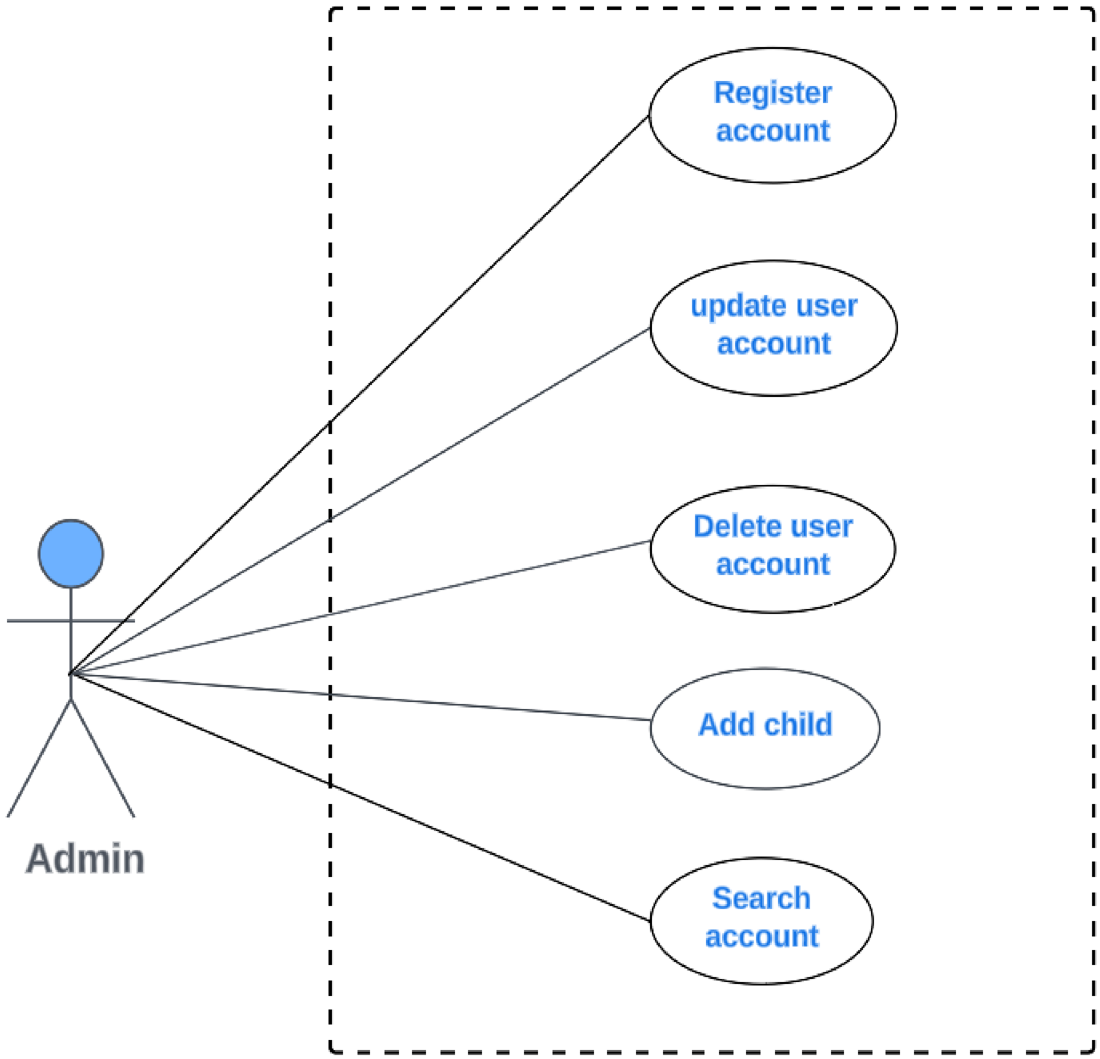
The RACE (Reading Comprehension from Examinations) dataset is a benchmark dataset designed to evaluate the reading comprehension abilities of machine learning models. It consists of passages followed by multiple-choice questions, derived from English language exams for middle and high school students. The dataset is specifically tailored to assess a model's ability to understand and answer questions based on written text.

Chapter 4 : System Analysis and Design

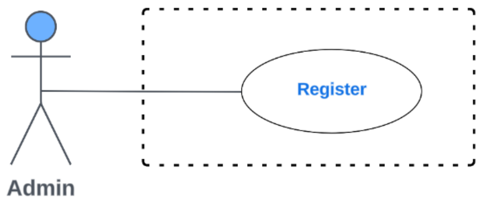
This chapter presents a comprehensive exploration of System Analysis and Design through a series of detailed diagrams. Each visual representation provides an intuitive understanding of complex concepts, methodologies, and processes involved in analyzing and designing effective information systems. The chapter is designed to facilitate quick comprehension and retention of key ideas through clear and concise graphical depictions.

4.1 Usecase Diagram

4.1.1 Admin

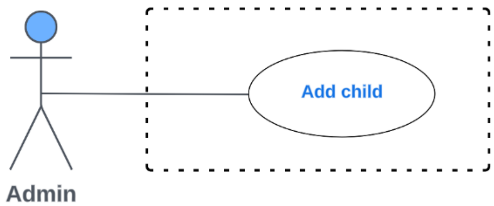


4.1.1.1 Registration:



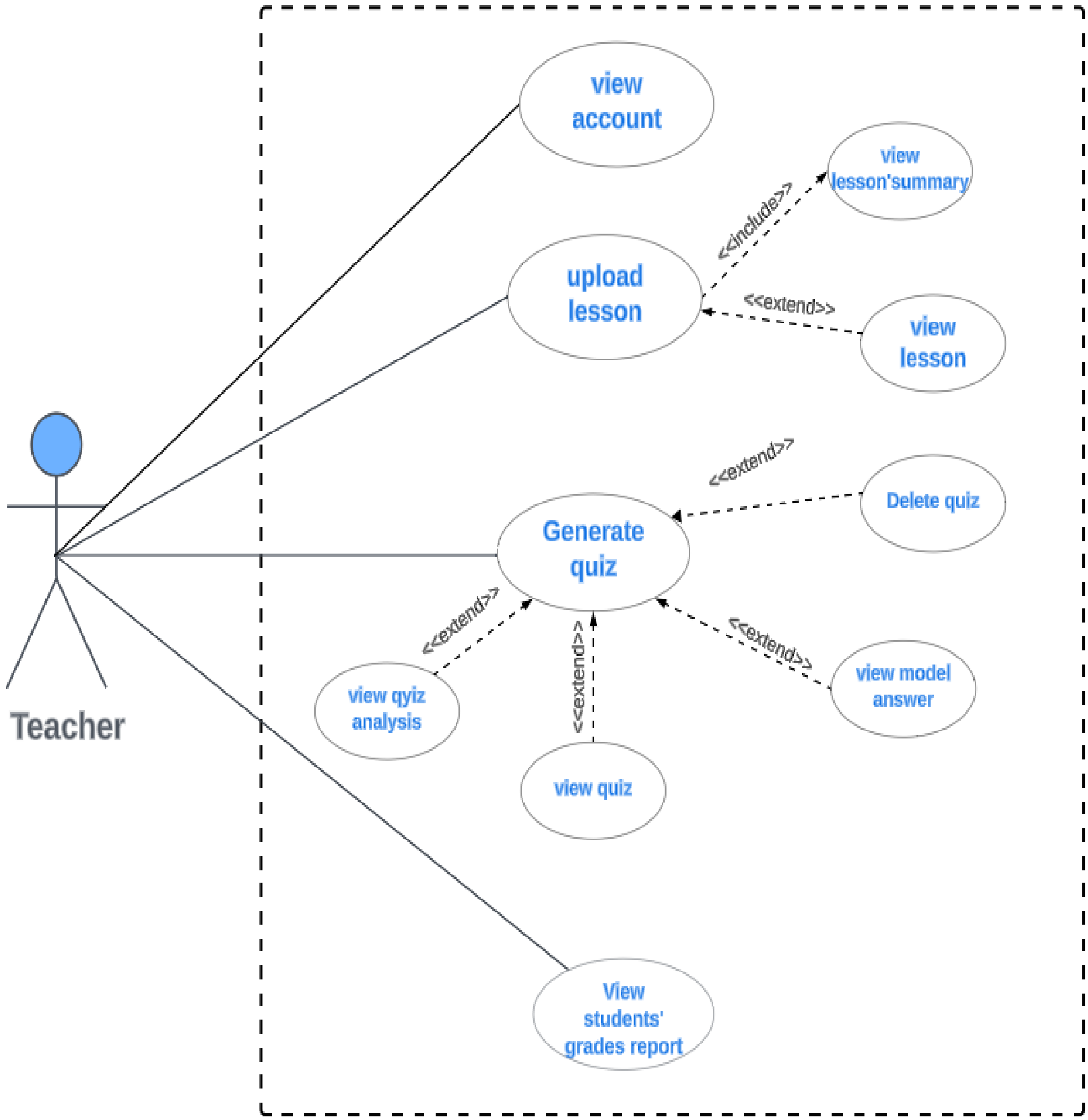
Actor	Admin
Description	New admins create an account which give him the privilege to do special functions our system.
Precondition	None -
Postcondition	Admin account is created and added to the system, now the admin can update and delete a user, add a child and search for any account.
Main successful scenario	1.Admin selects the "Registration" option from the admin dashboard. 2. System prompts the admin to enter the details of the new user. 3. Admin fills in the required user information, such as username, email, password, etc. 4. Admin submits the registration form. 5. System validates the entered information. 6. If the information is valid, system creates a new admin account with the provided details.
Unsuccessful scenario	The admin may fill fields with wrong data so a message of invalidation is appear and the admin cannot use the system until he enter the correct data .
Actor	Admin
Description	New admins create an account which give him the privilege to do special functions our system.
Precondition	None -

4.1.1.2 Add child:

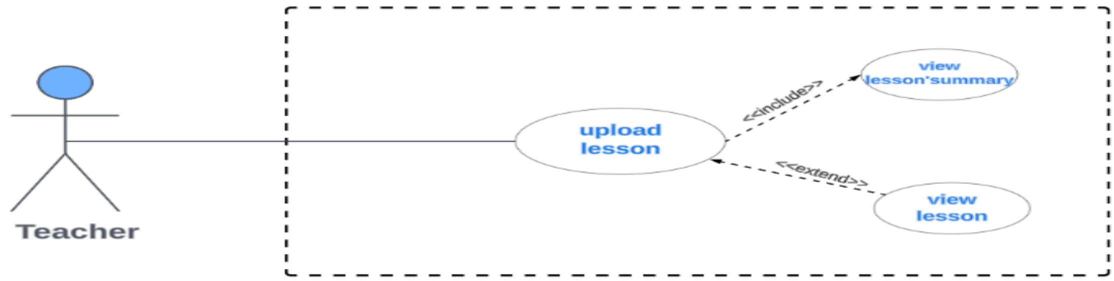


Actor	Admin
Description	This use case describes the process of an admin adding a new child user to the system.
Precondition	The user must sign-up as an admin.
Postcondition	A new child account is created add to the system .
Main successful scenario	1.Admin selects the "Add Child" option from the admin dashboard. 2. Admin fills in the required user information, such as child's name. 3. Admin submits the form to add the child user. 4. System validates the entered information. 5. If the information is valid, the system creates a new child user account with the provided details. 6. System notifies the admin that the child user has been successfully added 7. Admin can search for an existing account if he wants to determine if the child user already exists in the system. 8. If the account is found, the admin selects the existing account. 9. If the account is not found, the admin proceeds to create a new child user account by following the steps from no. 2 to no. 6.
Unsuccessful scenario	The teacher may fill fields with invalid data so the new child account cannot be created, so the system will inform the teacher to enter valid information. Or the account is already existed, so the admin won't need to add one.

4.1.2 Teacher

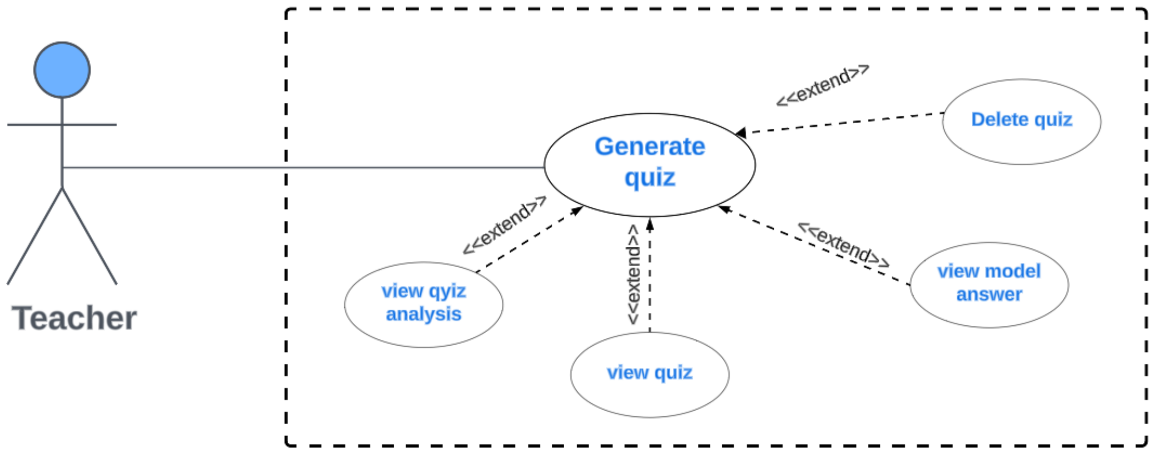

view account
view lesson'summary
<<include>>
upload lesson
<<extend>>
view lesson
<<extend>>
Delete quiz
Generate quiz
<<extend>>
<<extend>>
<<extend>>
view qyiz analysis
view model answer
view quiz
Teacher
View students' grades report

4.1.2.1 Upload Lessons



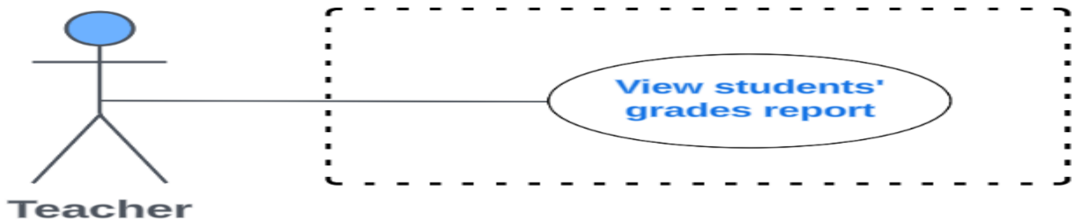
Actor	Teacher
Description	Teacher upload educational materials (lessons), to a lesson in the system, which includes viewing the lesson where the materials will be uploaded.
Precondition	He must sign-up as a teacher.
Postcondition	The educational materials are successfully uploaded to the lesson.
Main successful scenario	1. Teacher selects the "Upload" option from the lesson management menu. 2. Teacher selects the option to upload materials. 3. System prompts the teacher to select the files to upload. 4. Teacher selects the files from their local system and uploads them. 5. System validates the uploaded files. 6. If the files are valid, the system stores the materials in the lesson's repository and generates the lesson's summary automatically. 7. System notifies the teacher that the materials have been successfully uploaded and the summary generated. 8. after uploading teacher can select the "View Lesson" sub-process to select the lesson where materials will be uploaded. 9. Teacher can view the lesson selects the desired lesson from the list of available lessons. 10. system retrieves the uploaded lesson and display it to the teacher.
Unsuccessful scenario	The user didn't sign up as a teacher, so the system redirects the teacher to the login page. Or invalidate type of the document that the teacher tries to upload.

4.1.2.2 Generate Quiz



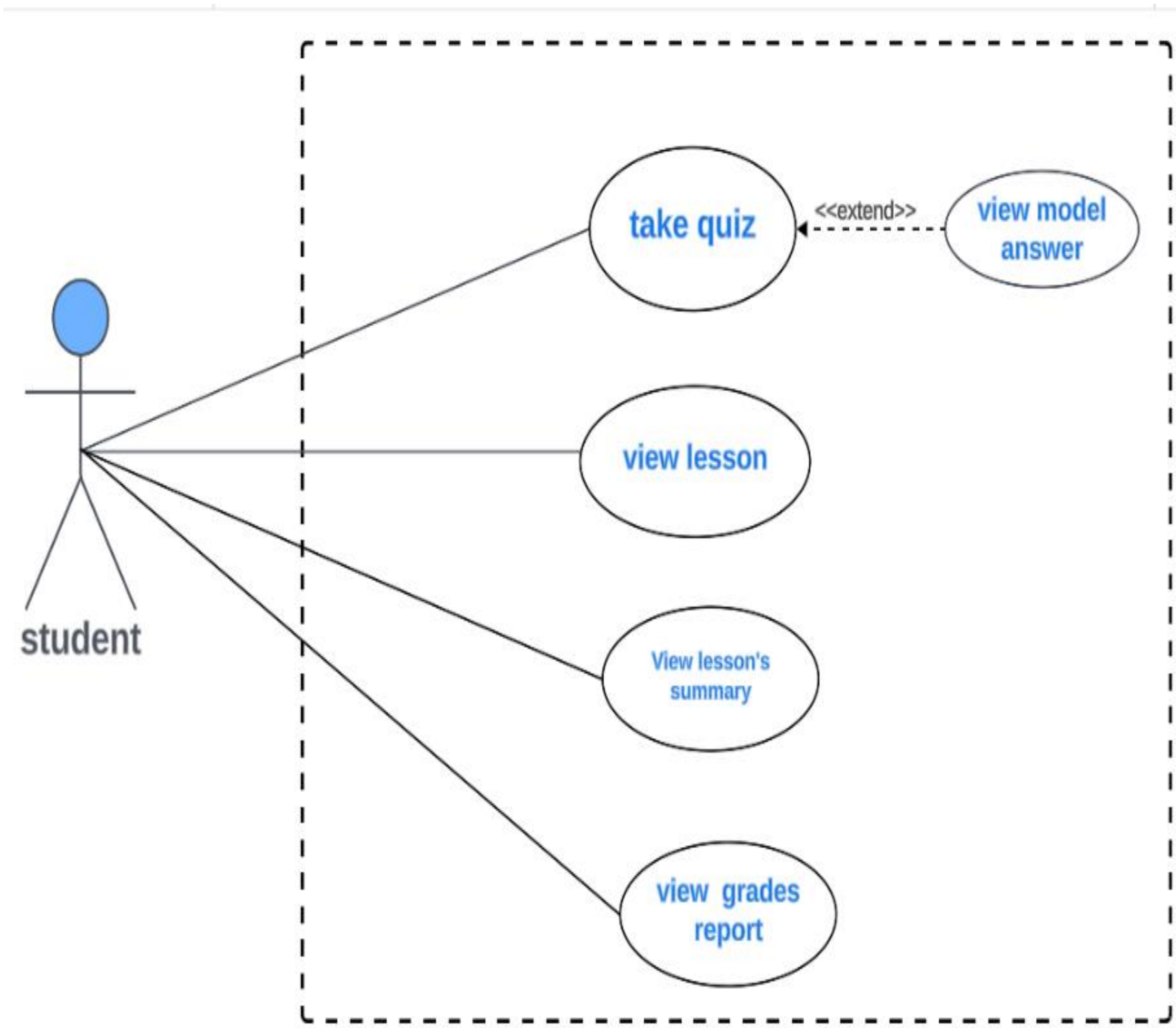
Actor	Teacher
Description	Teacher can generate a quiz for students. which includes viewing the model answer for reference.
Precondition	He must sign-up as a teacher.
Postcondition	The quiz is successfully generated and made available to students, and he can view the quiz or delete it, view the model answer, or view the quiz analysis.
Main successful scenario	1. Teacher selects the "Generate Quiz" option from the teacher dashboard. 2. System notifies the teacher that the quiz has been successfully generated. 3. Teacher can select the option to view the model answer, view quiz, delete quiz or view quiz analysis. 4. if he selects "view quiz" System retrieves quiz and displays it to the teacher 5. if he selects "view model answer" System retrieves the model answer and displays it to the teacher. 6. if he selects "delete quiz" System will delete the quiz and notify that the quiz deleted successfully. 7. if he selects "view quiz analysis " System retrieves the quiz analysis and displays it to the teacher.
Unsuccessful scenario	the user didn’t sign up as a teacher, so the system redirects the teacher to the login page.

4.1.2.3 View students' grades report

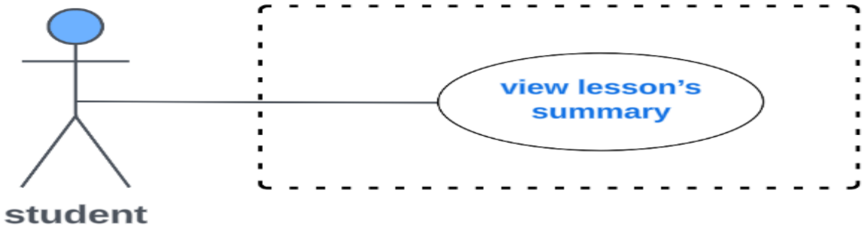


Actor	Teacher
Description	Teacher can view a report containing the grades of his students.
Precondition	He must sign-up as a teacher.
Postcondition	The teacher is successfully viewing the grades report.
Main successful scenario	1. Teacher selects the "View Grades Report" option from the teacher dashboard. 2. System prompts the teacher to select the student for whom the teacher wants to view his grades. 3. System retrieves the grades data for the selected student. 4. System displays the grades report to the teacher, which includes information such as student names, grades, and any additional relevant details. 5. Teacher reviews the grades report to assess student performance.
Unsuccessful scenario	May be because the user didn't sign up as a teacher, so the system redirects the teacher to the login page.

4.1.3 Student

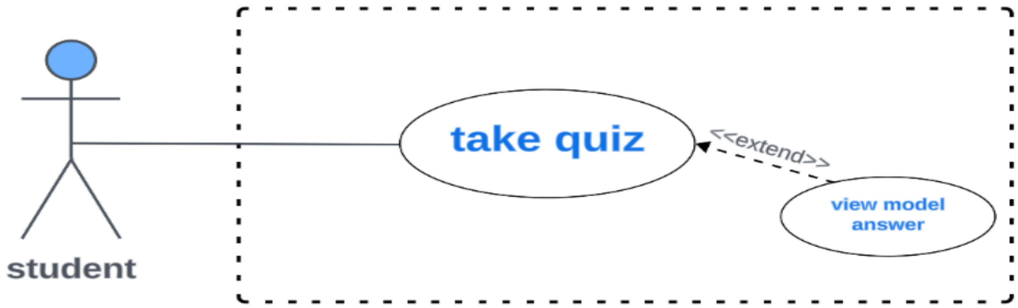

take quiz
<<extend>>
view model answer
view lesson
View lesson's summary
view grades report
student

4.1.3.1 View lesson's summary



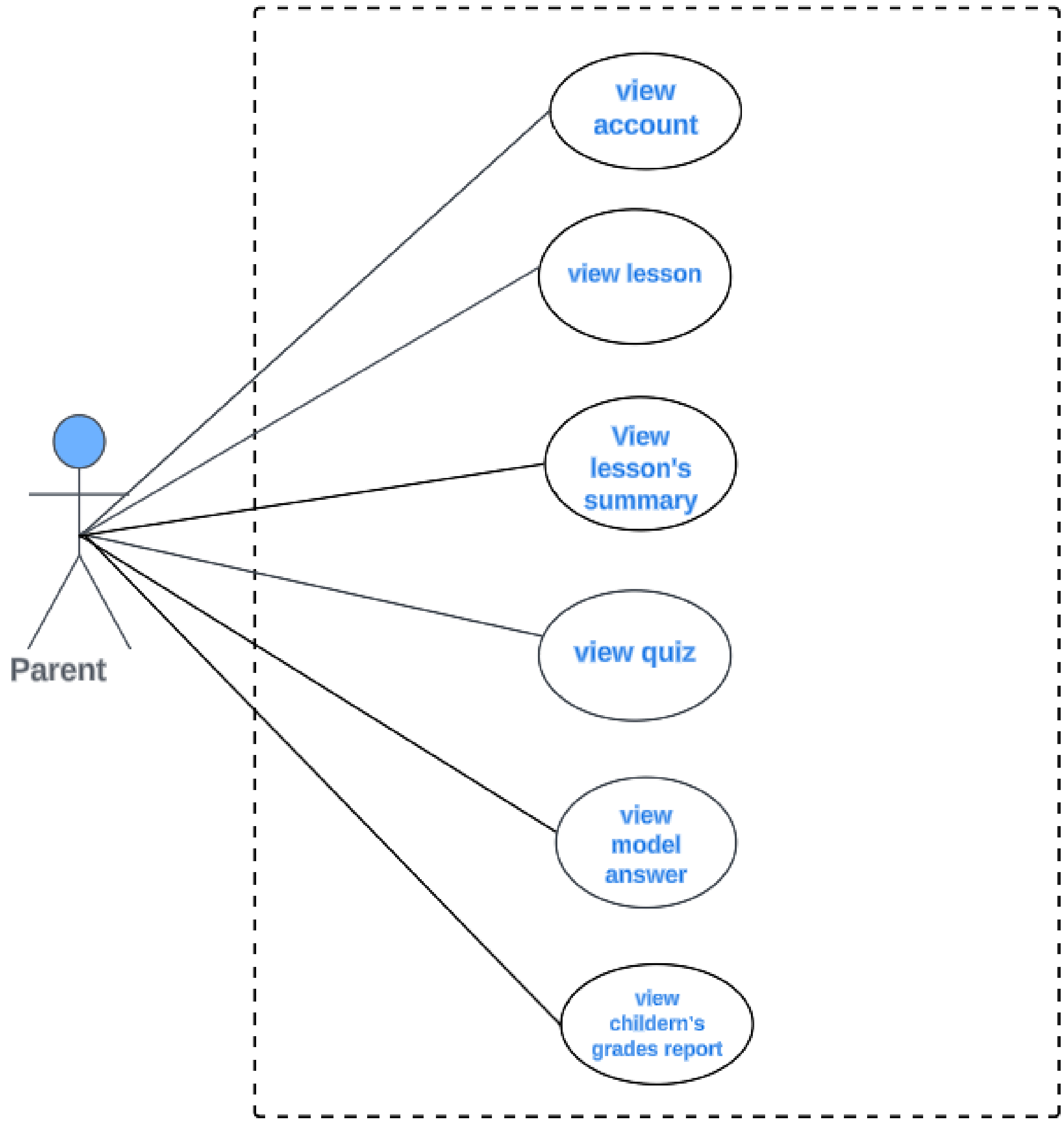
Actor	Student
Description	The student can view the summary of the selected lesson within the learning process
Precondition	The user must sign up as a student, and the desired lesson must have been added.
Postcondition	The student successfully views the lesson's summary.
Main successful scenario	1. Student selects the " View Lesson Summary " option from the student dashboard. 3. Student selects the desired lesson from the list of available lessons. 4.System retrieves the summary of the lesson and displays it to the student.
Unsuccessful scenario	The user didn't sign up as a student, so the system redirects the student to the login page. Or the desired lesson doesn't exist or have a problem in it.

4.1.3.2 Take Quiz

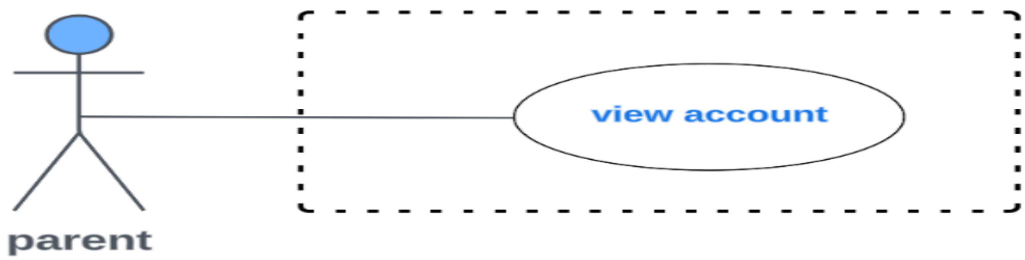


Actor	Student
Description	The student can take a quiz and view the model answer if he wants.
Precondition	The user must sign up as a student, and the quiz must be available and accessible to the student (the teacher must add a quiz to be taken).
Postcondition	The student successfully takes his quiz and view the model answer of the quiz (optional).
Main successful scenario	1. Students select the quiz they want to take from the list of available quizzes. 2. System presents the quiz questions to the student. 3. Student answers the quiz question. 4. After completing all the questions, the student submits the quiz for evaluation. 5. System evaluates the student's responses and provides immediate feedback if enabled. 6. If he wants to view the correct answer, the student has the option to extend the quiz-taking process to view the model answers. 7. Student selects the option to view the model answers. 8. System retrieves the model answers and displays them to the student. 9. Student reviews the model answers for comparison with their own responses.
Unsuccessful scenario	May be because the user didn't sign up as a student, so the system redirects the student to the login page. Or there isn't a quiz to be taken.

4.1.4 Parent

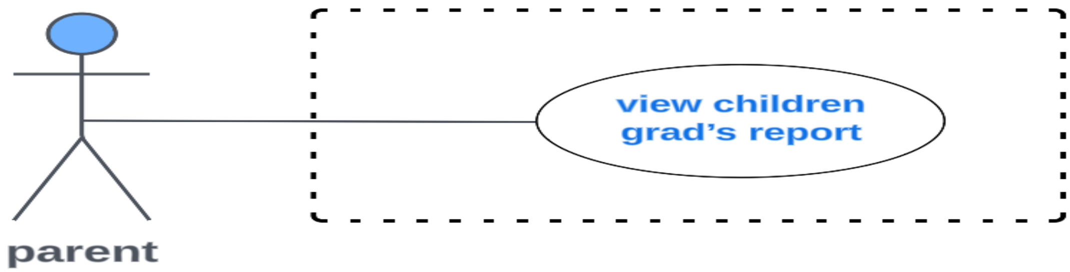

view account
view lesson
View lesson's summary
view quiz
view model answer
view childern's grades report
Parent

4.1.4.1 View Account



Actor	parent
Description	The parent can view his own account details activities within the system.
Precondition	The user must sign up as a parent. And the parent must have an active account.
Postcondition	The parent successfully views their account details and activities.
Main successful scenario	1. Parent navigates to the "Account" or "Profile" section within the parent dashboard. 2. System retrieves the account details of the logged-in parent. 3. System presents the account details to the parent, including personal information, contact details, and any other relevant information. 4. Parent reviews the account information.
Unsuccessful scenario	May be because the user didn't sign up as a parent, so the system redirects the parent to the login page. Or because the parent account isn't active.

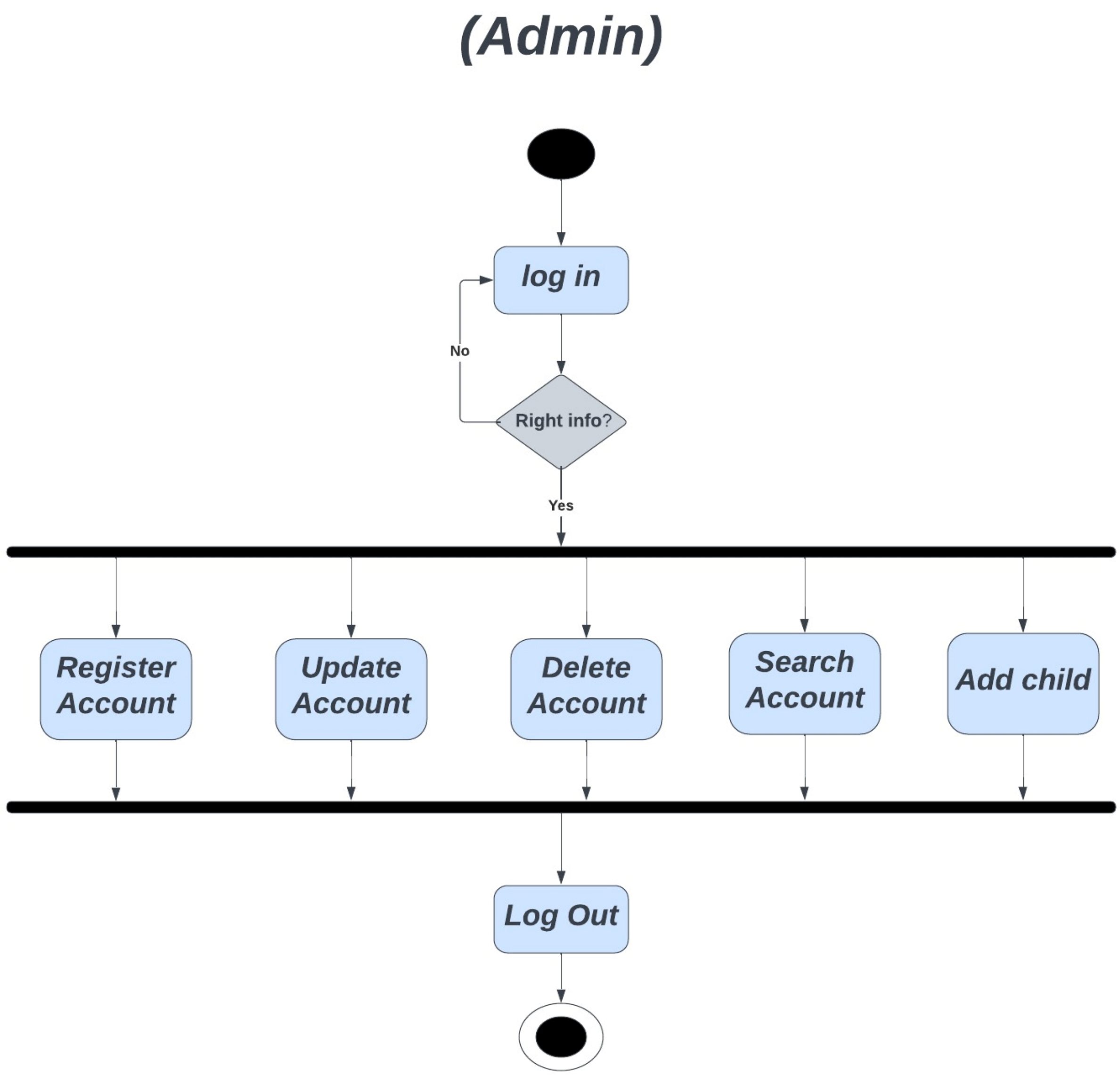
4.1.4.2 Children Report



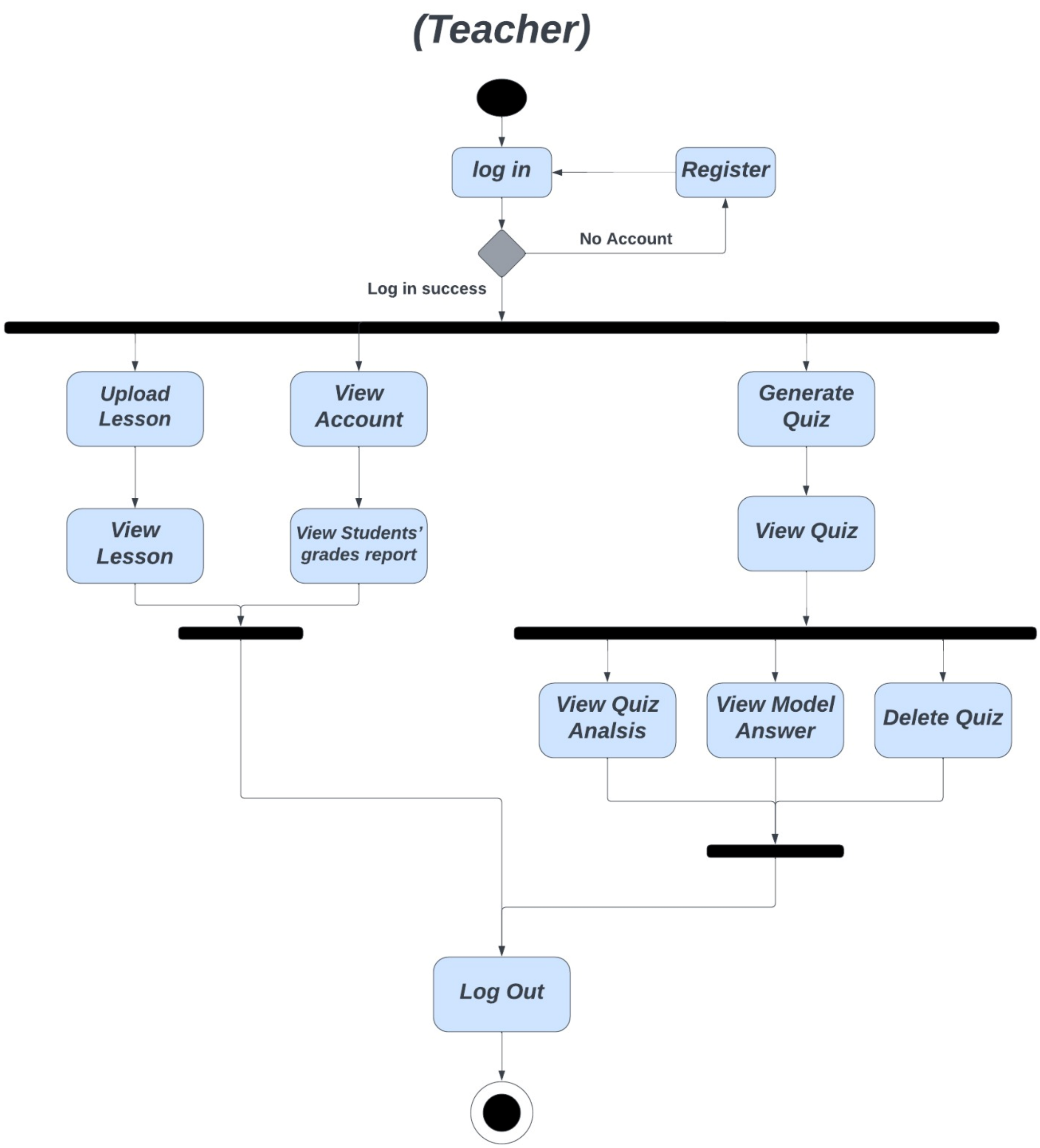
Actor	Parent
Description	The parent can view the grades report of their children within the learning management system.
Precondition	The user must sign up as a student. The parent must have one or more children linked to their account. The parent's children should have taken a quiz before.
Postcondition	The parent successfully views the grades report of their children.
Main successful scenario	1. Parent navigates to the "Grades" or "Reports" section within the parent dashboard. 2. System retrieves the list of linked children associated with the parent's account. 3. Parent selects the desired child or children whose grades report he/she wants to view. 4. System presents the grades report of the selected children to the parent. 5. Parent reviews the grades report, which may include scores, percentages, and any additional relevant details for each child.
Unsuccessful scenario	May be because the user didn't sign up as a parent, so the system redirects the student to the login page. Or his account didn't link with any children account. Or the parents children didn't take any quiz yet .

4.2 Activity Diagram

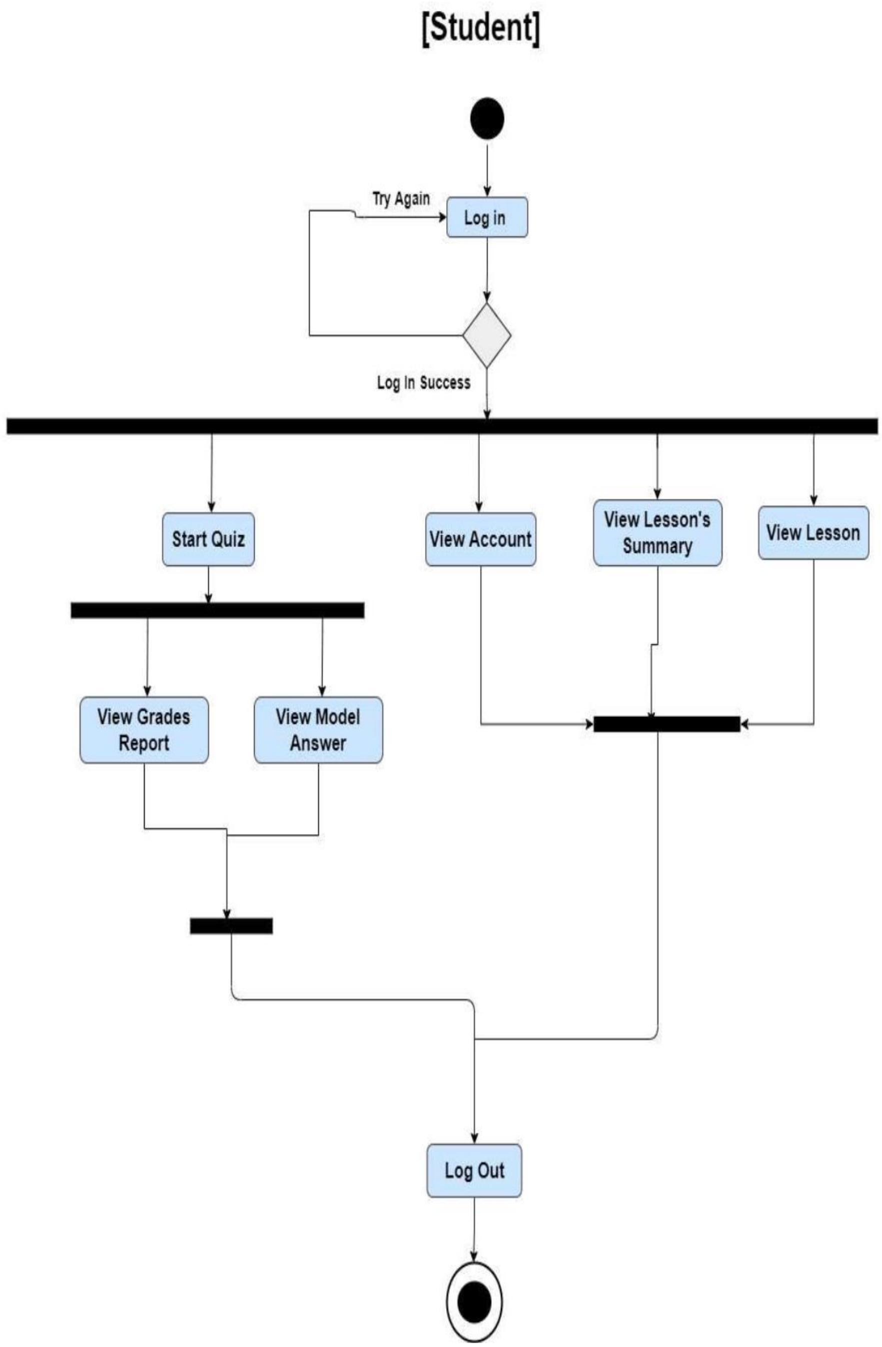
4.2.1 Admin



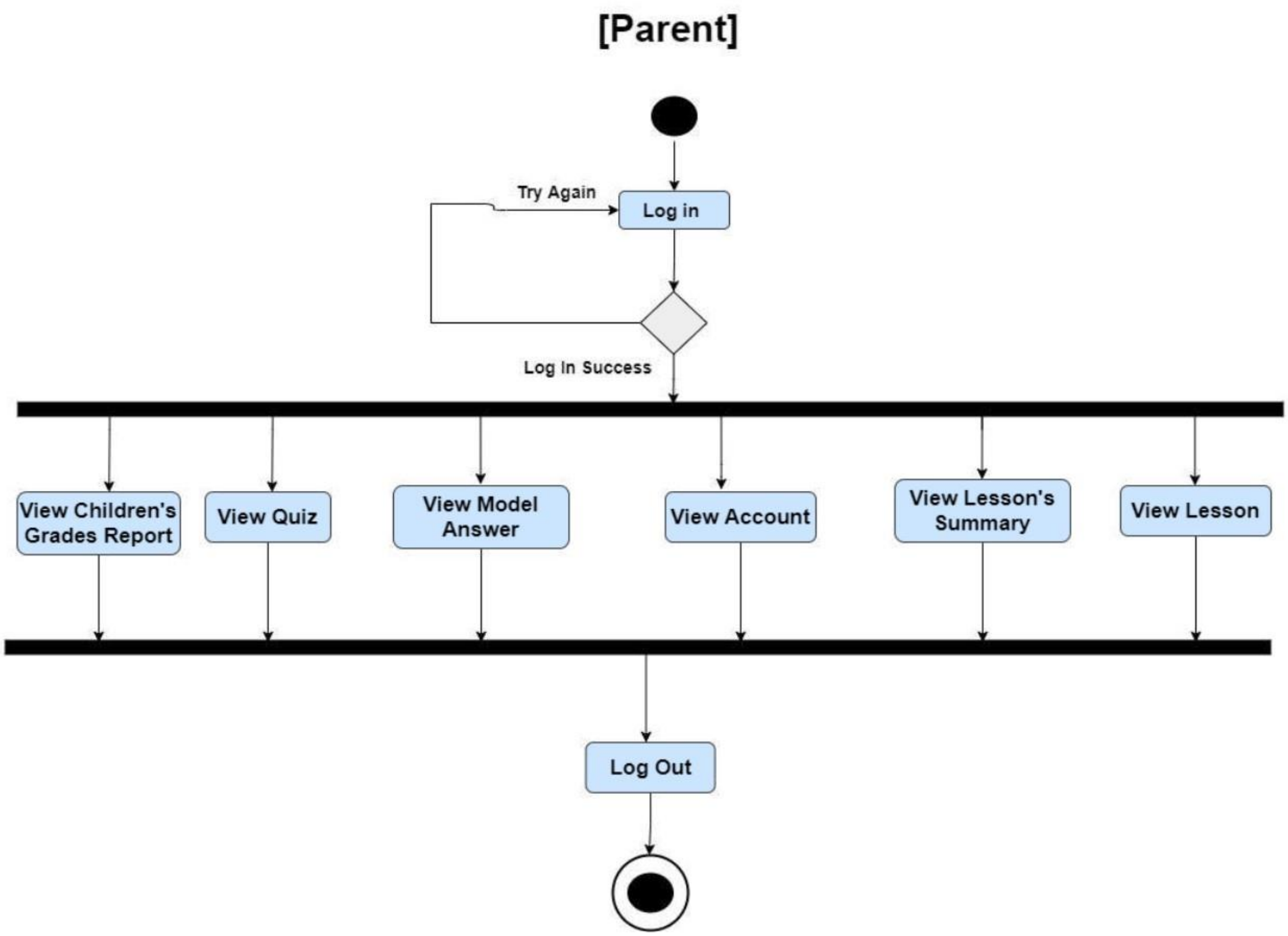
4.2.2 Teacher


(Teacher)
log in
Register
No Account
Log in success
Upload Lesson
View Account
Generate Quiz
View Lesson
View Students' grades report
View Quiz
View Quiz Analsis
View Model Answer
Delete Quiz
Log Out

4.2.3 Student

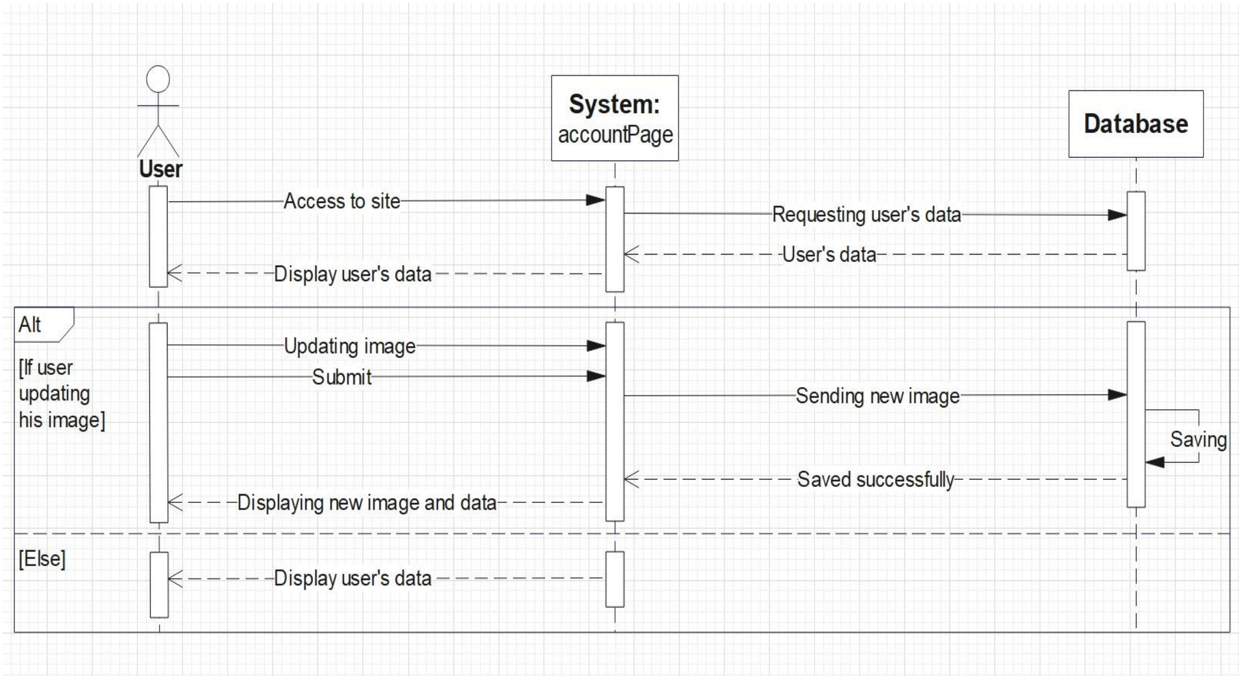

[Student]
Try Again
Log in
Log In Success
Start Quiz
View Account
View Lesson's Summary
View Lesson
View Grades Report
View Model Answer
Log Out

4.2.4 Parent

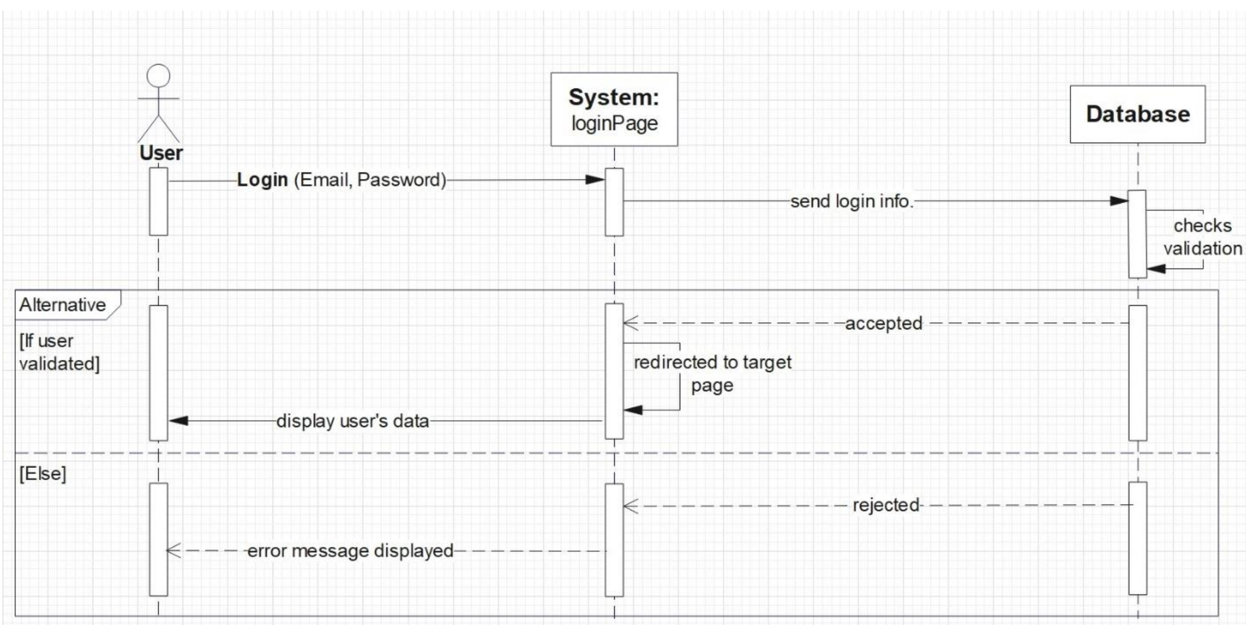

[Parent]
Try Again
Log in
Log In Success
View Children's Grades Report
View Quiz
View Model Answer
View Account
View Lesson's Summary
View Lesson
Log Out

4.3 Sequence Diagarm

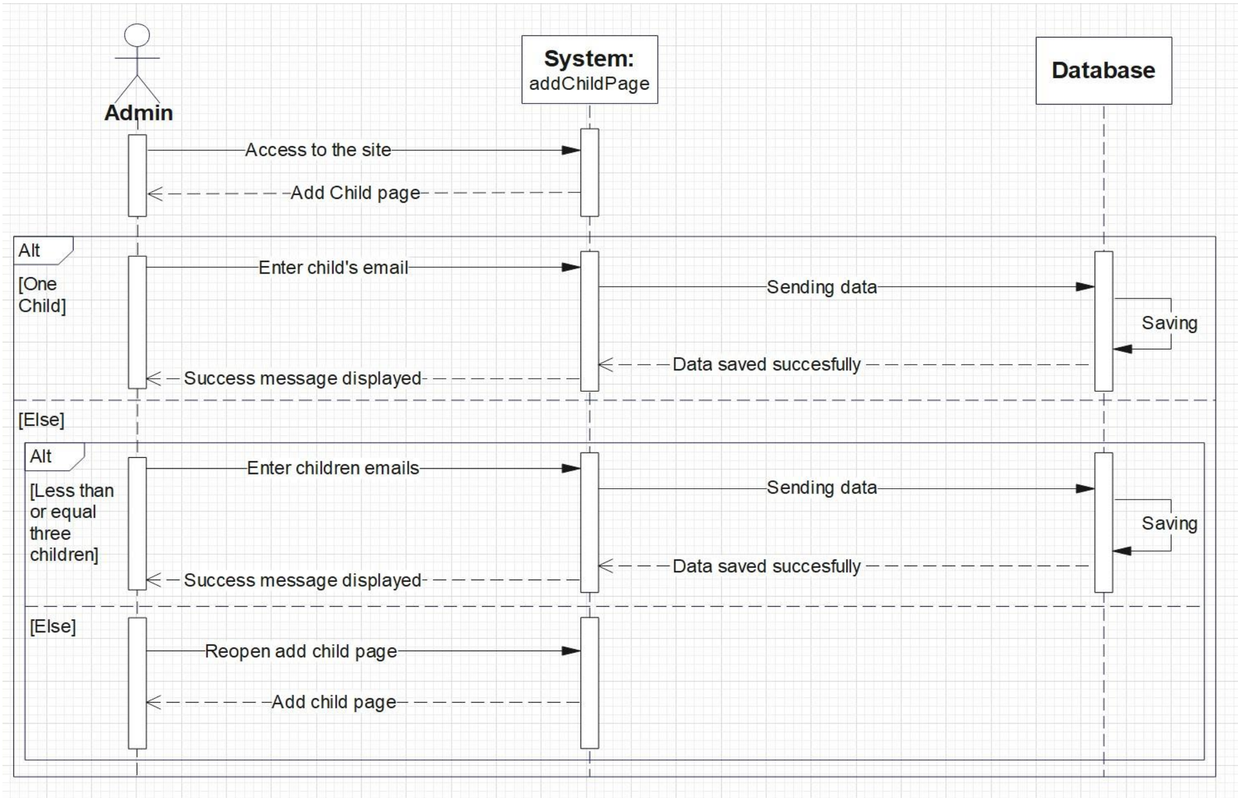
4.3.1 Account Page



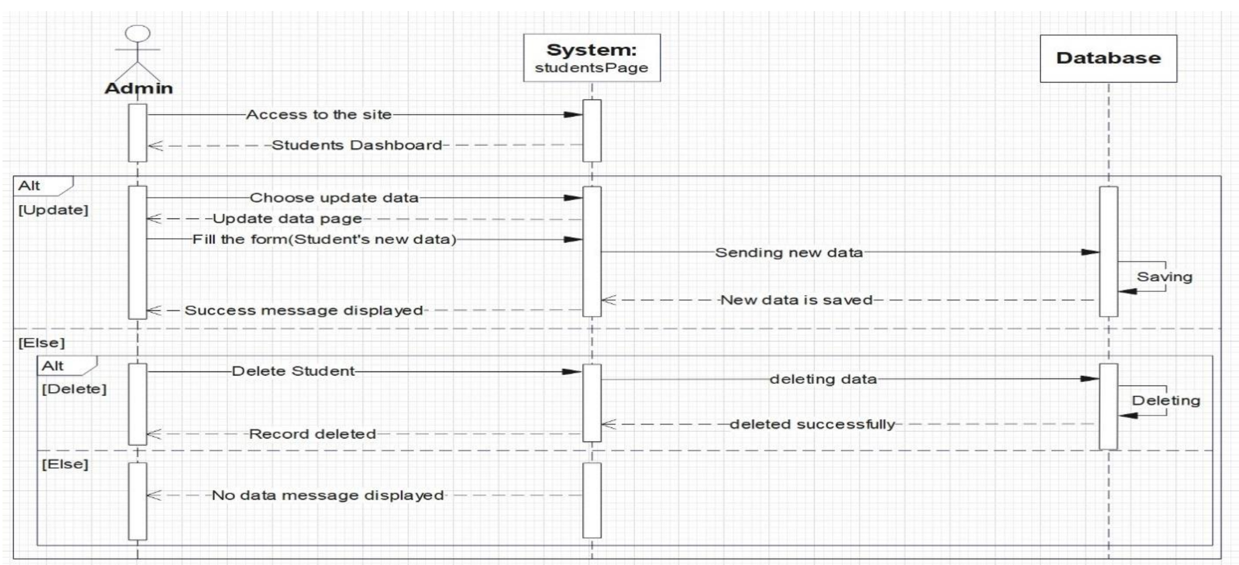
4.3.2 Login Page



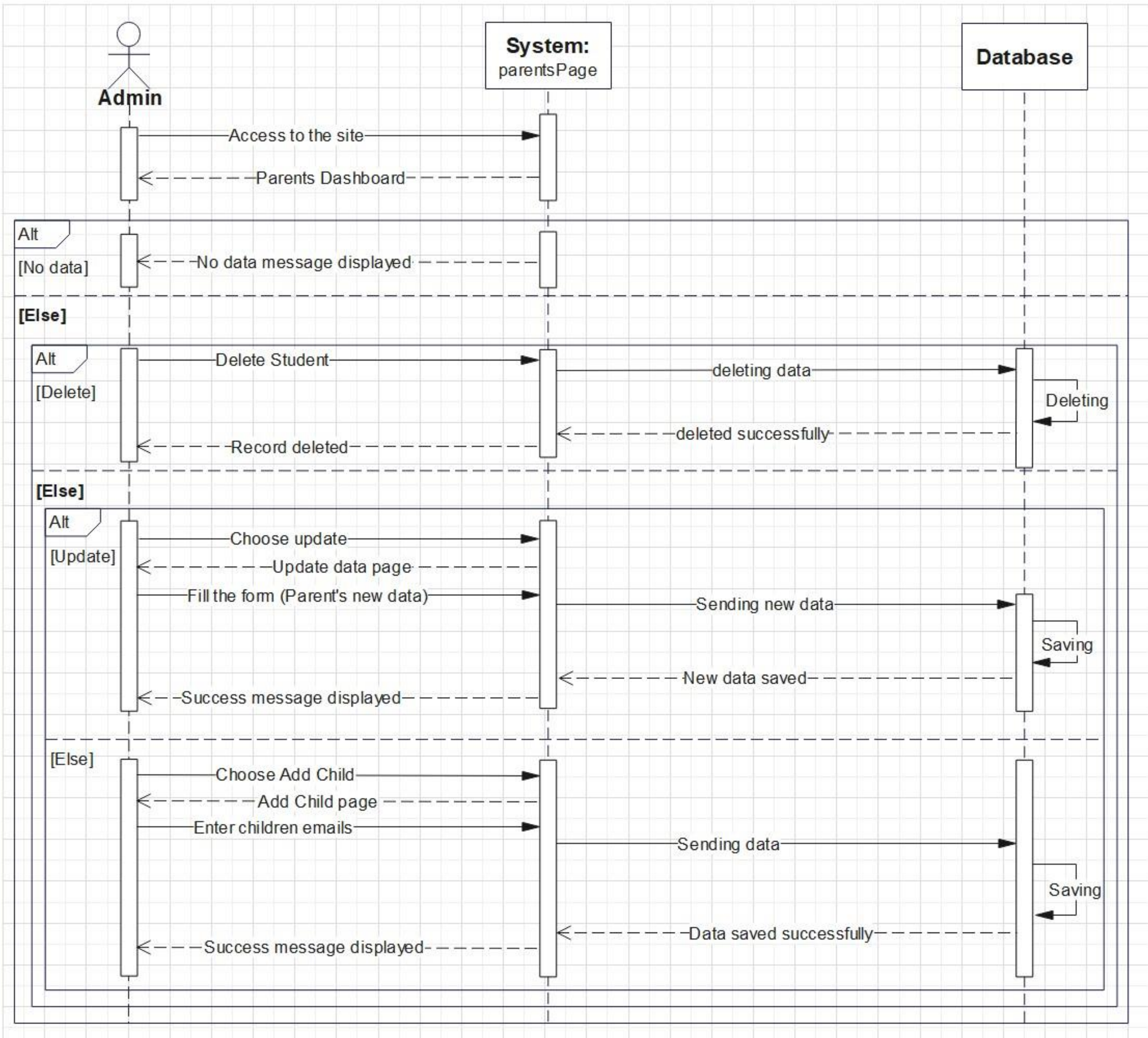
4.3.3 Add Child Page



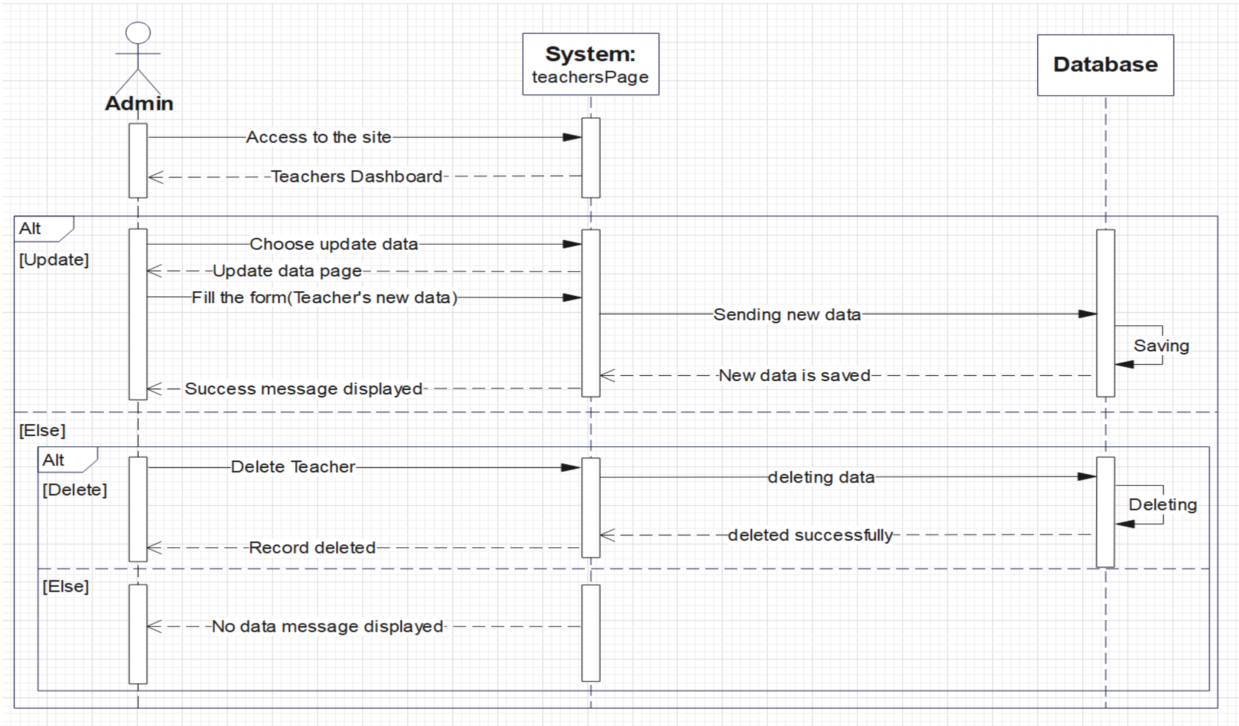
4.3.4 Managing Student Page



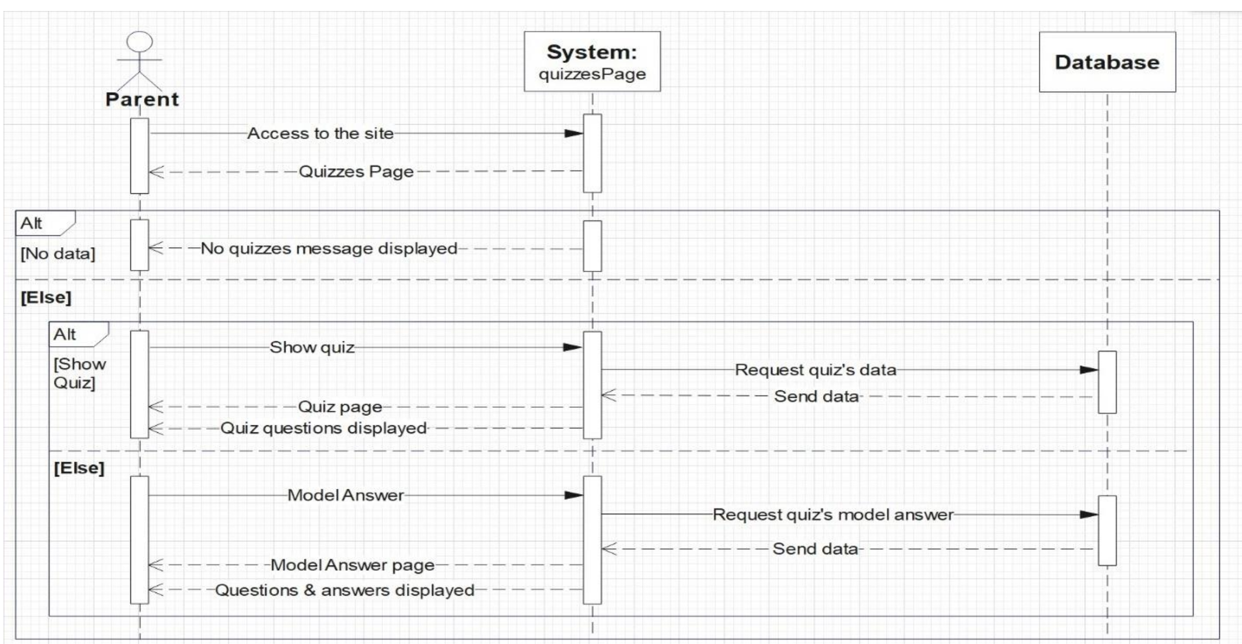
4.3.5 Managing Parent Page



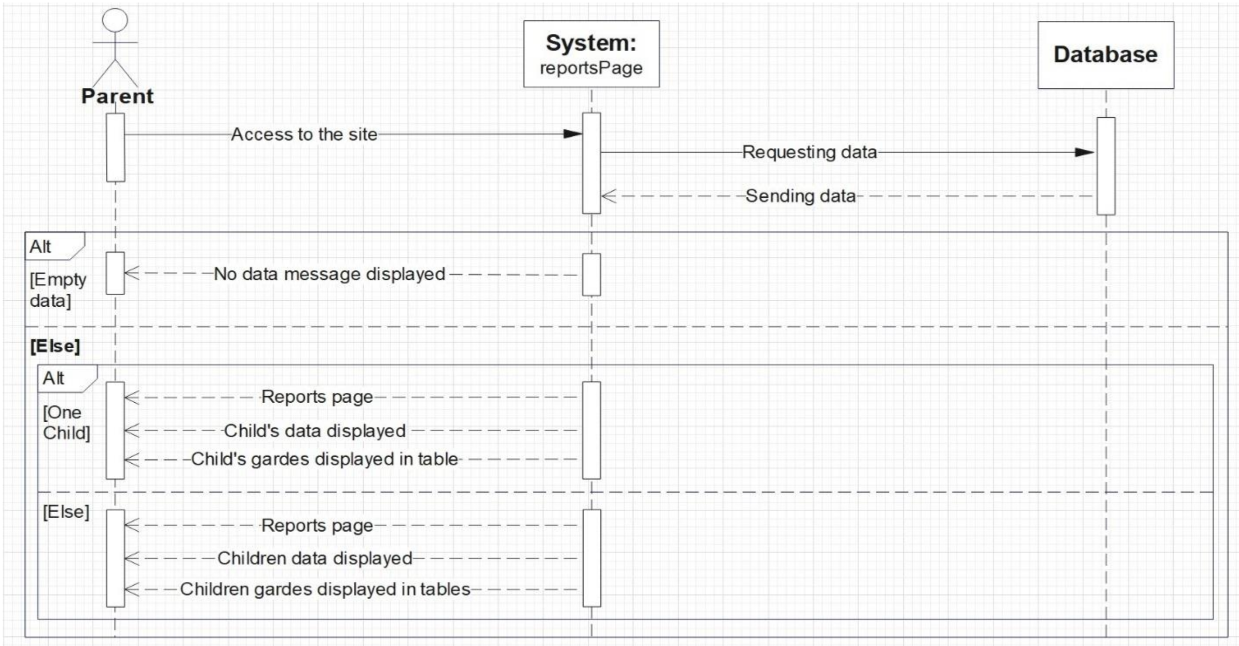
4.3.6 Managing Teacher Page



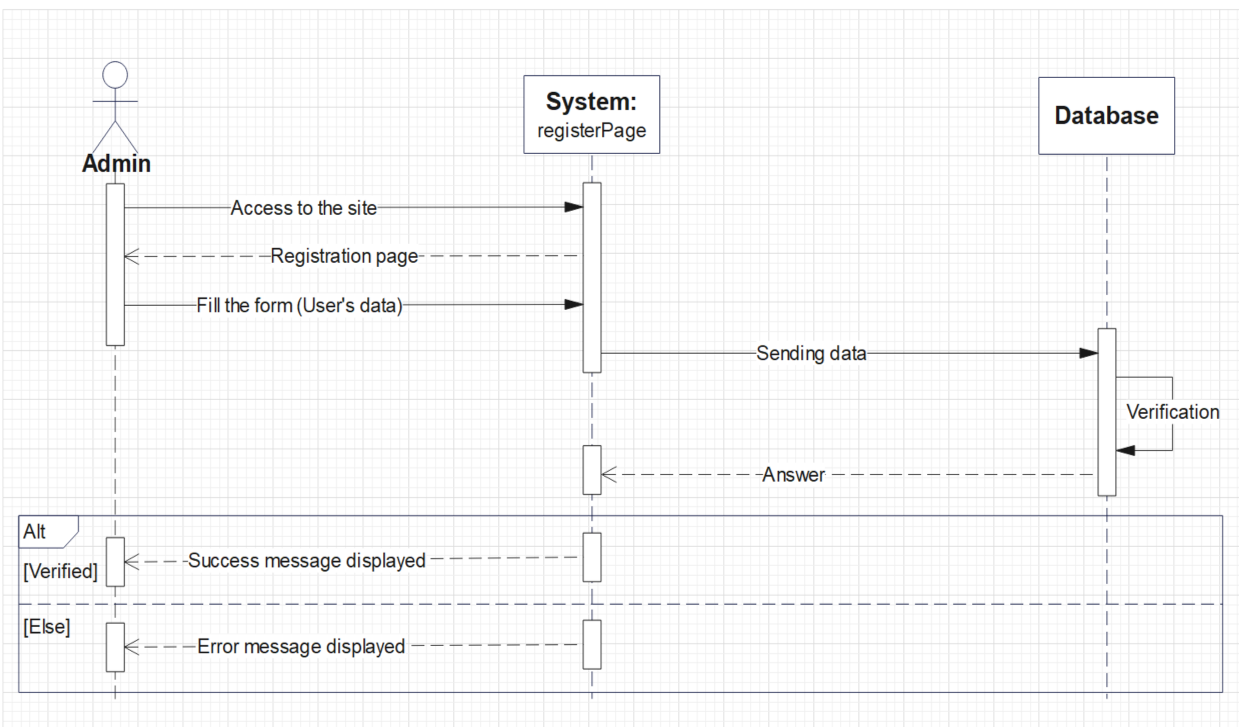
4.3.7 Quizzes Page For Parent



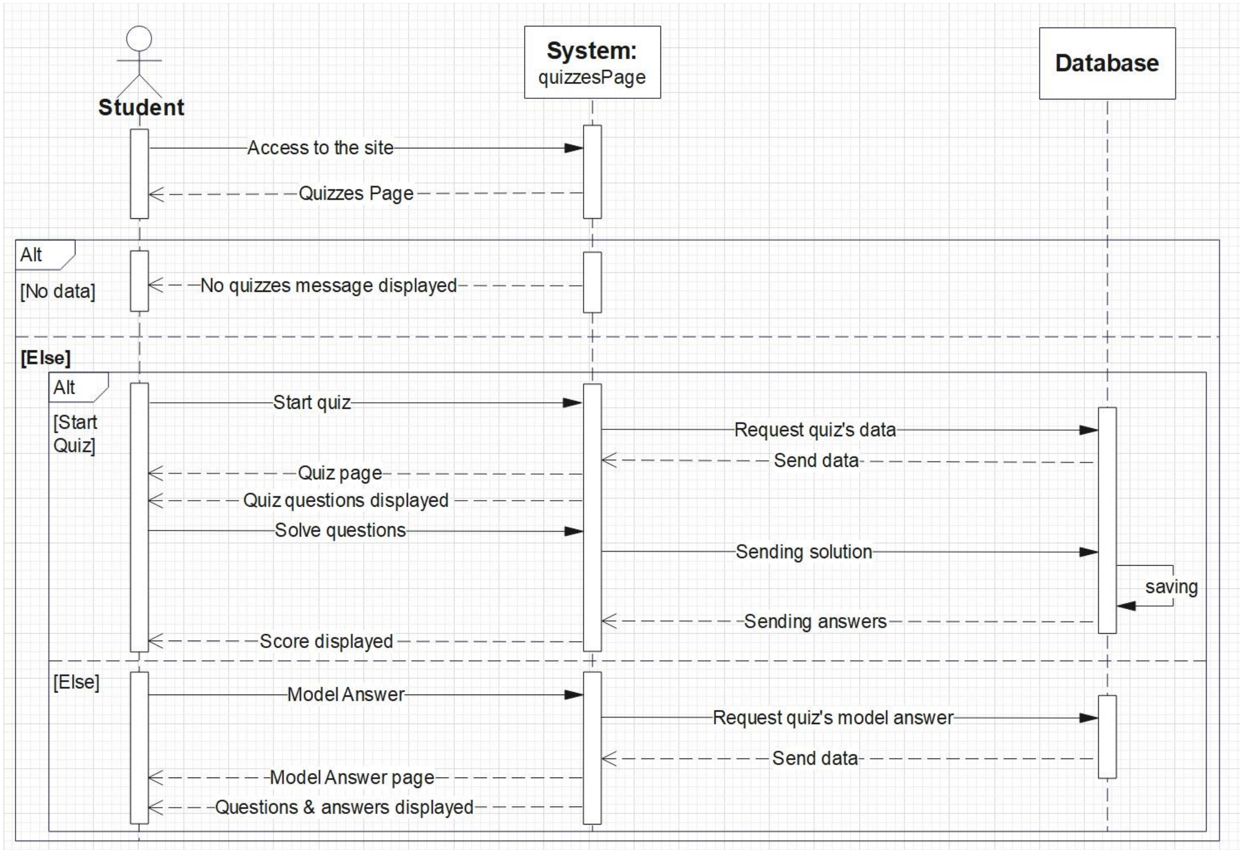
4.3.8 Reports Page for Parent



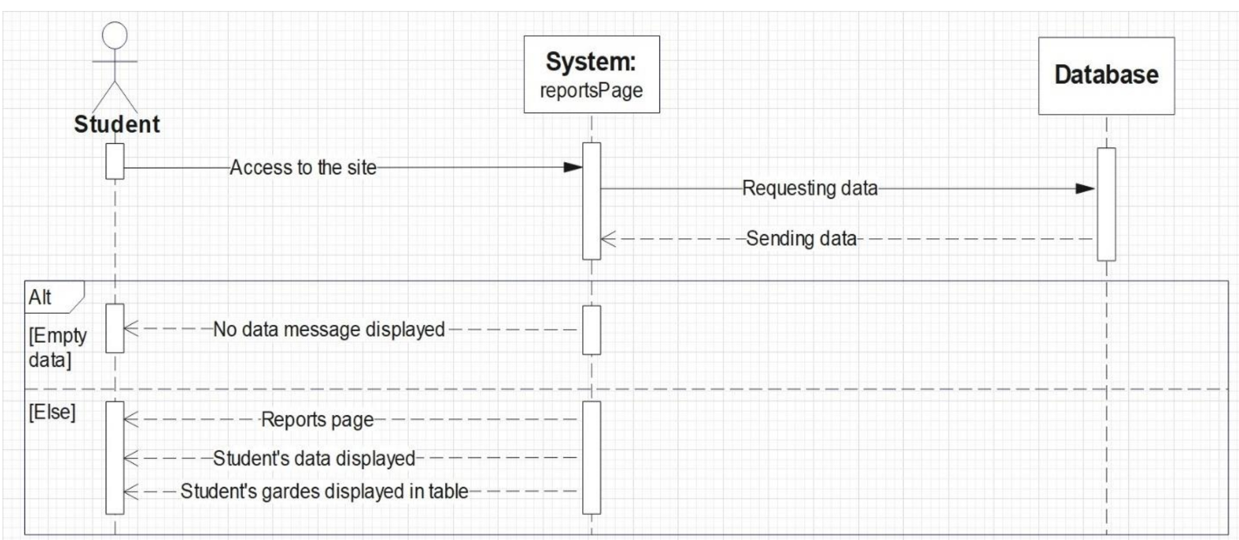
4.3.9 Register



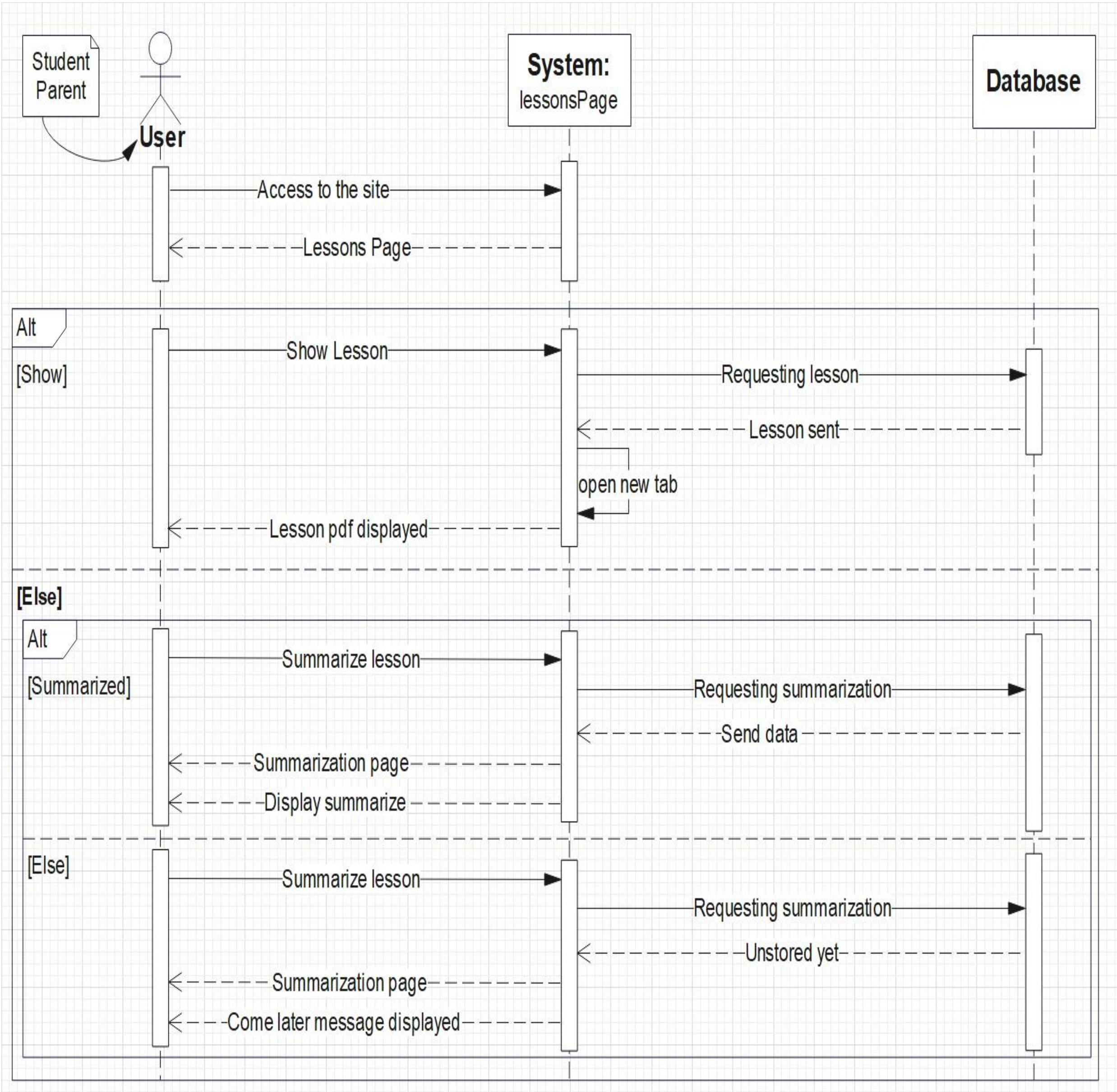
4.3.10 Quizzes Page For Student



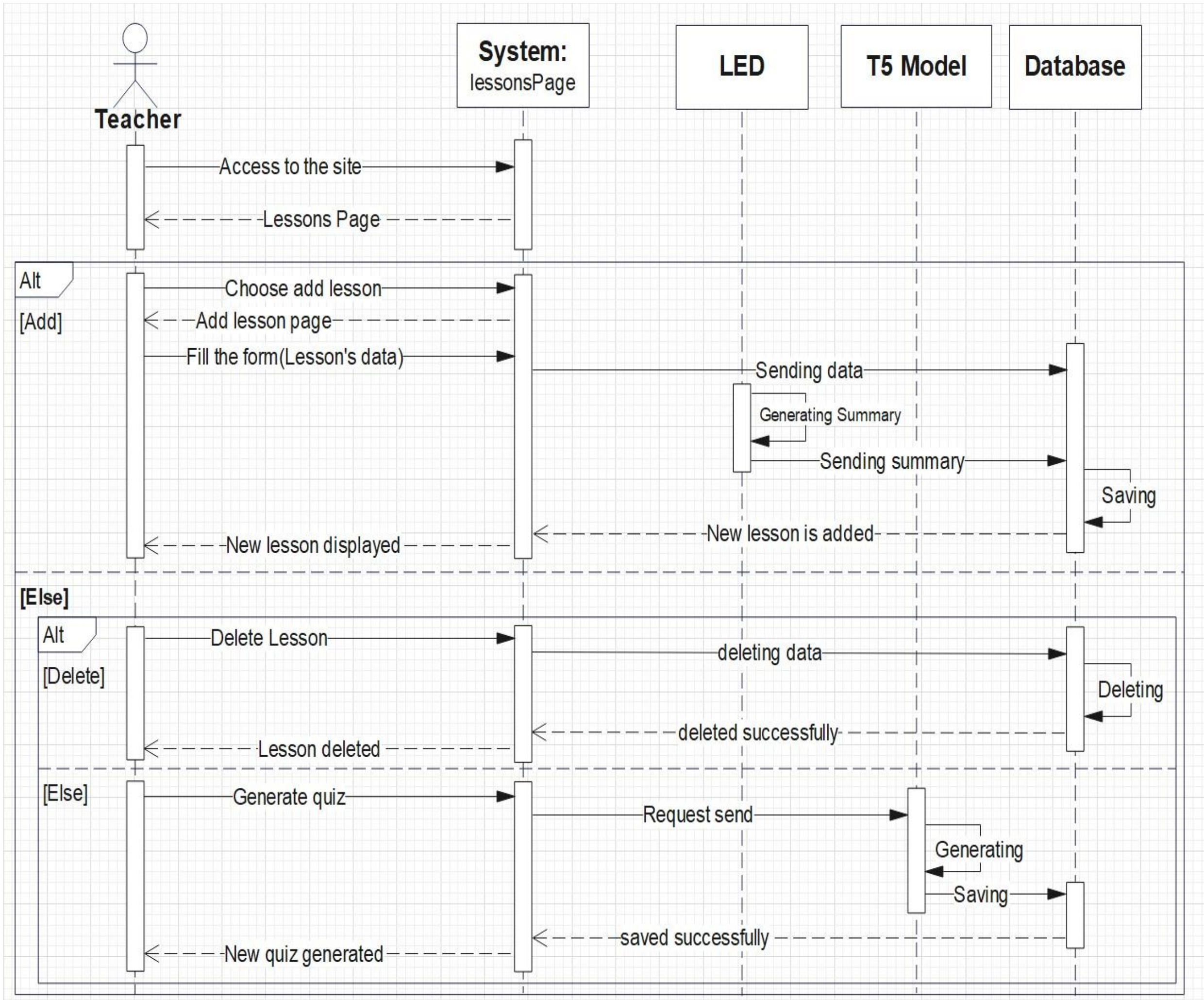
4.3.11 Reports Page For Student



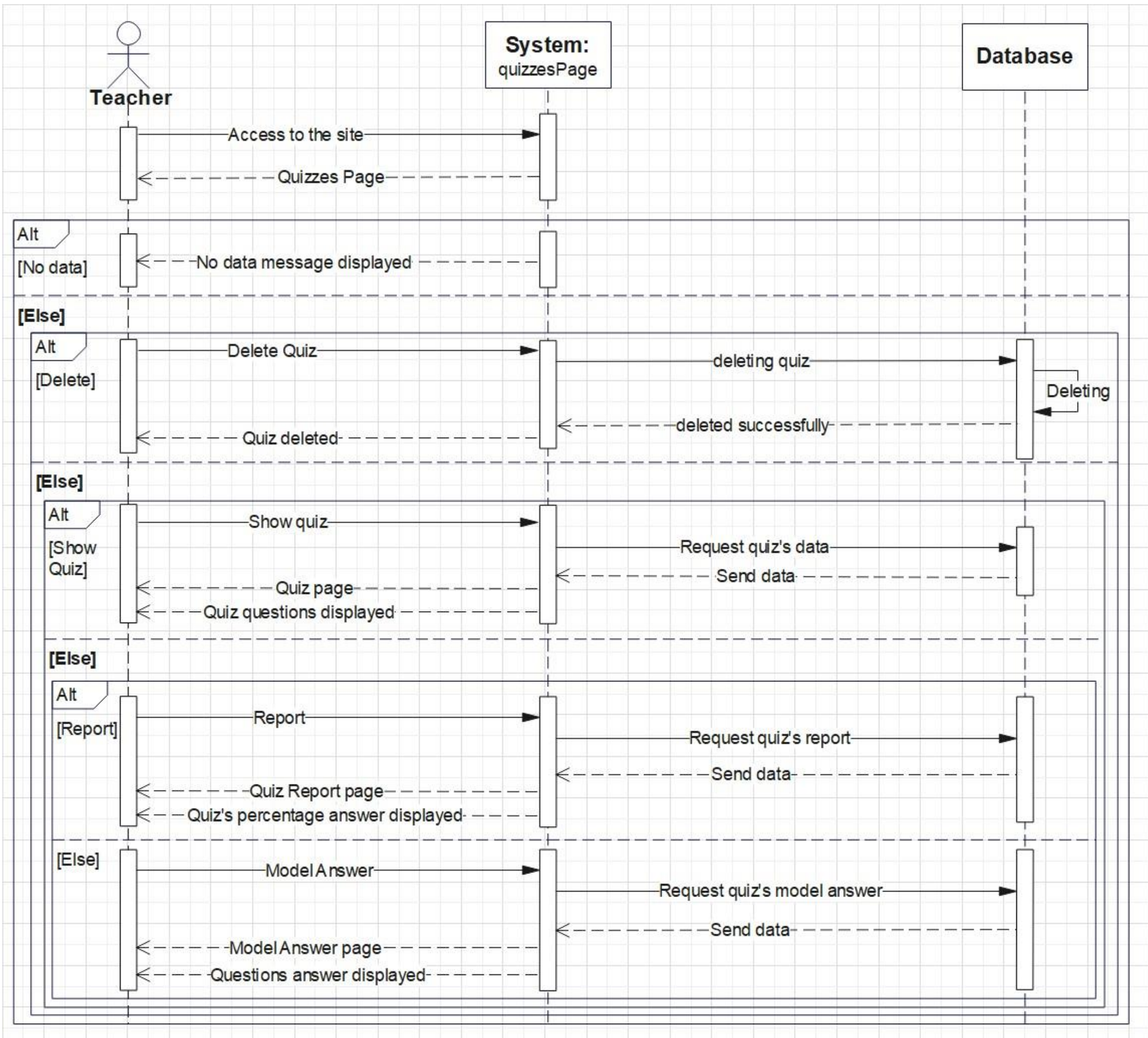
4.3.12 Lessons Page For Student and Parent



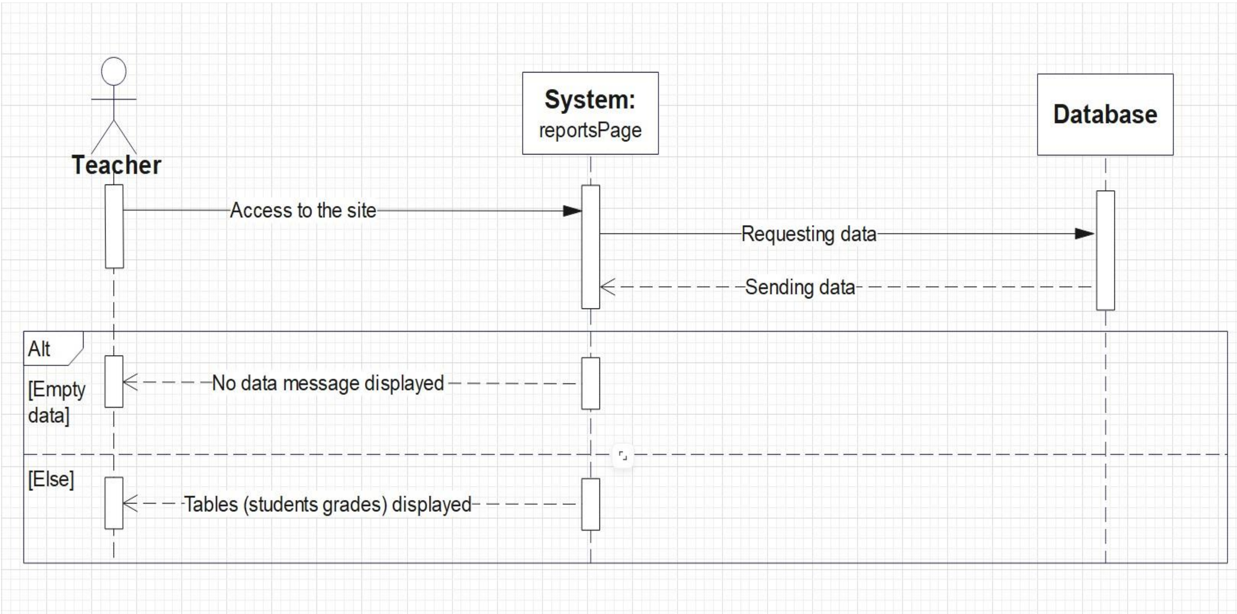
4.3.13 Lessons Page For Teacher



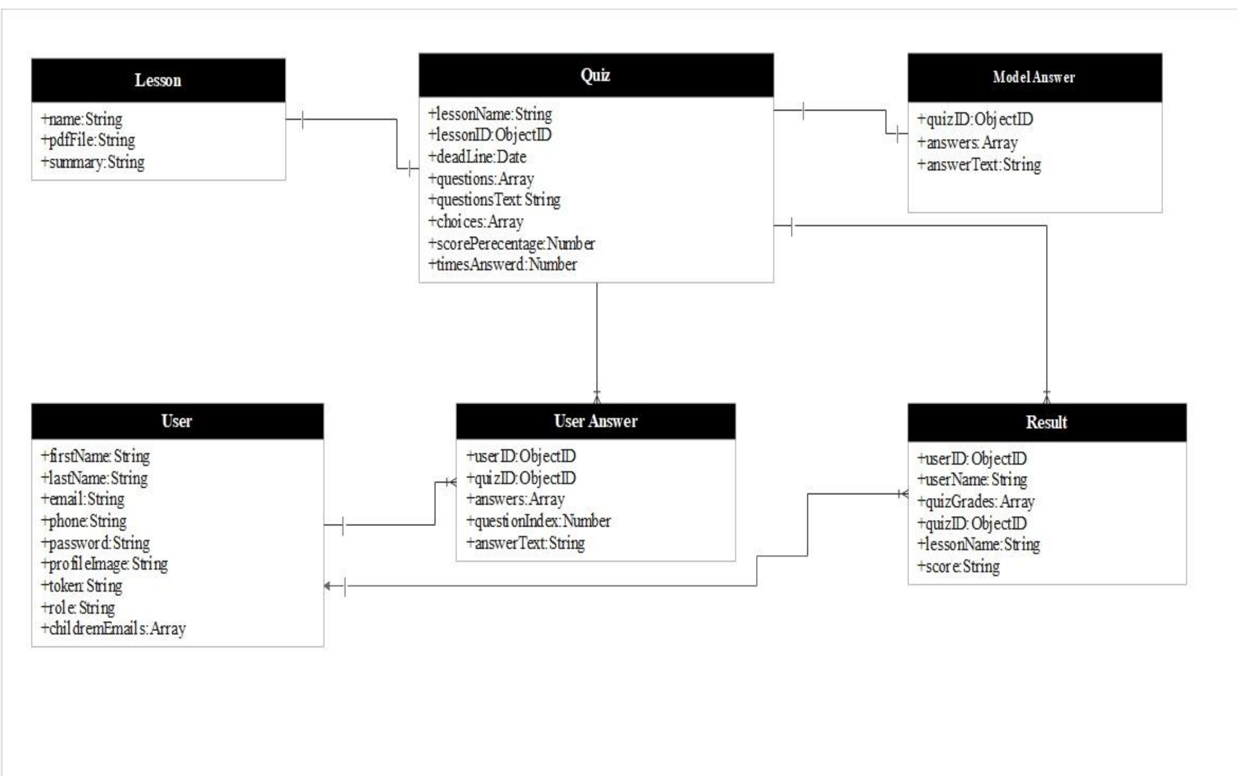
4.3.14 Quizzes Page For Teacher



4.3.15 Reports Page For Teacher

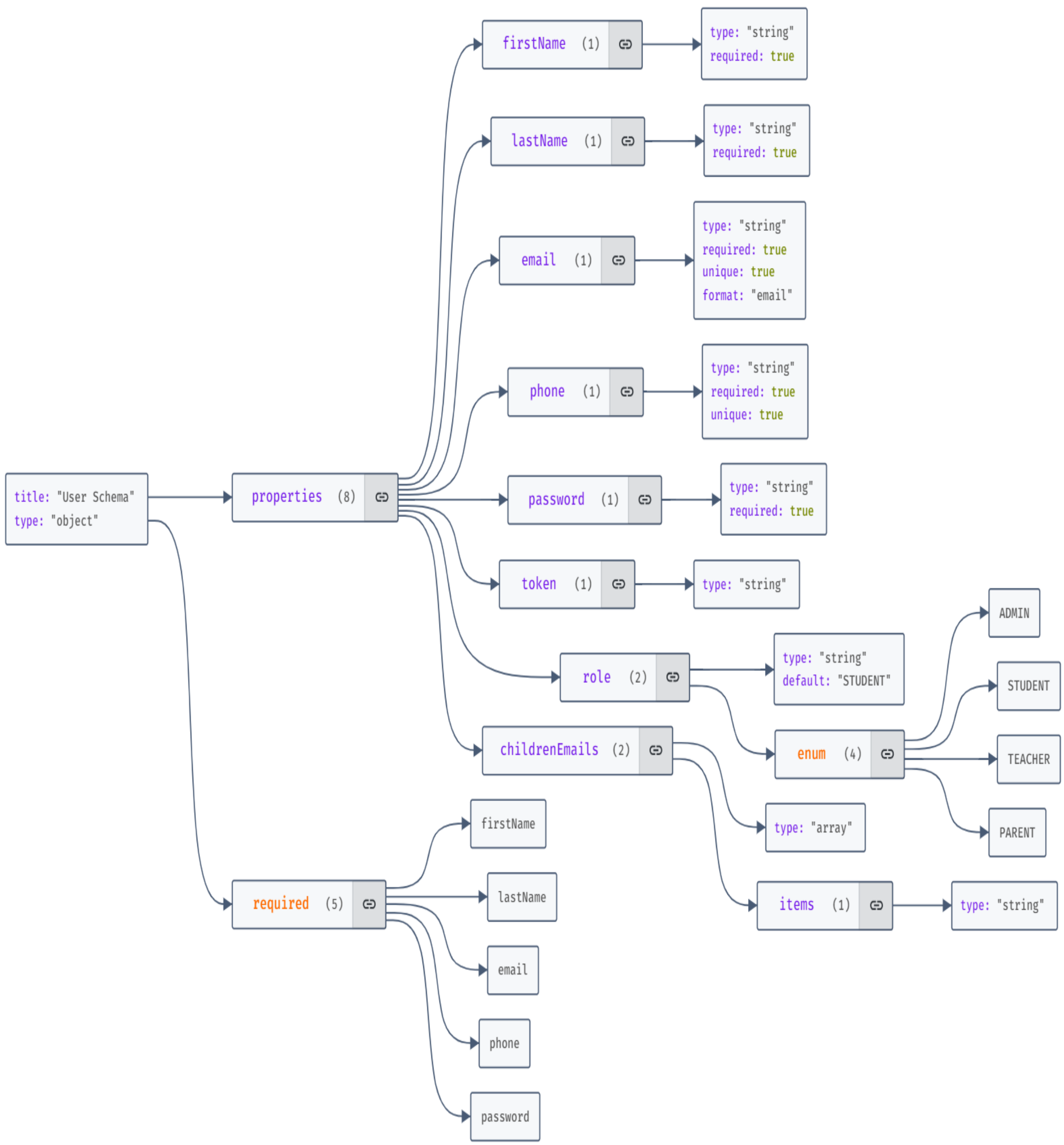


4.4 Document Diagram

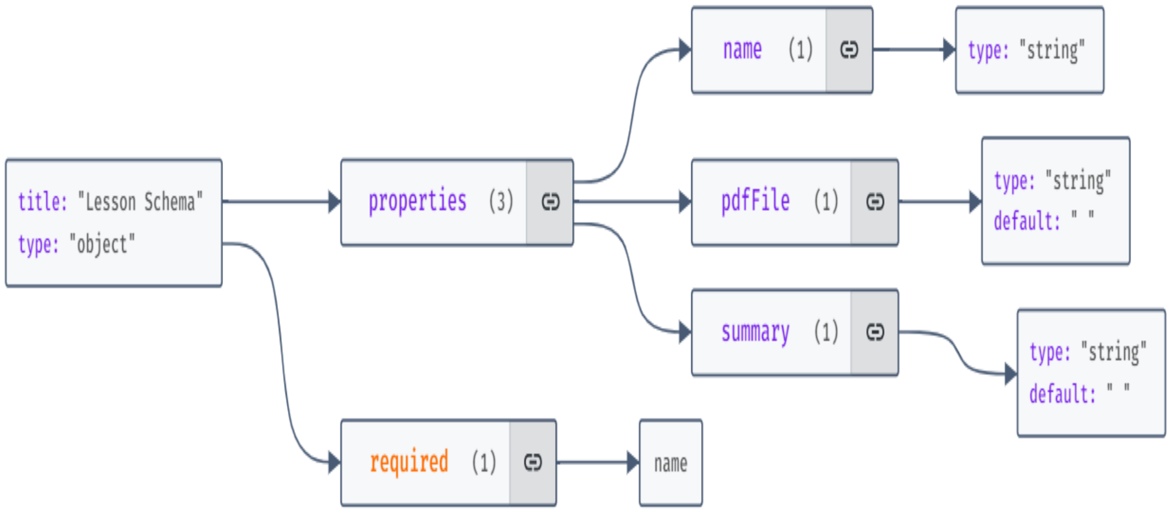


4.5 Json Schema Diagram

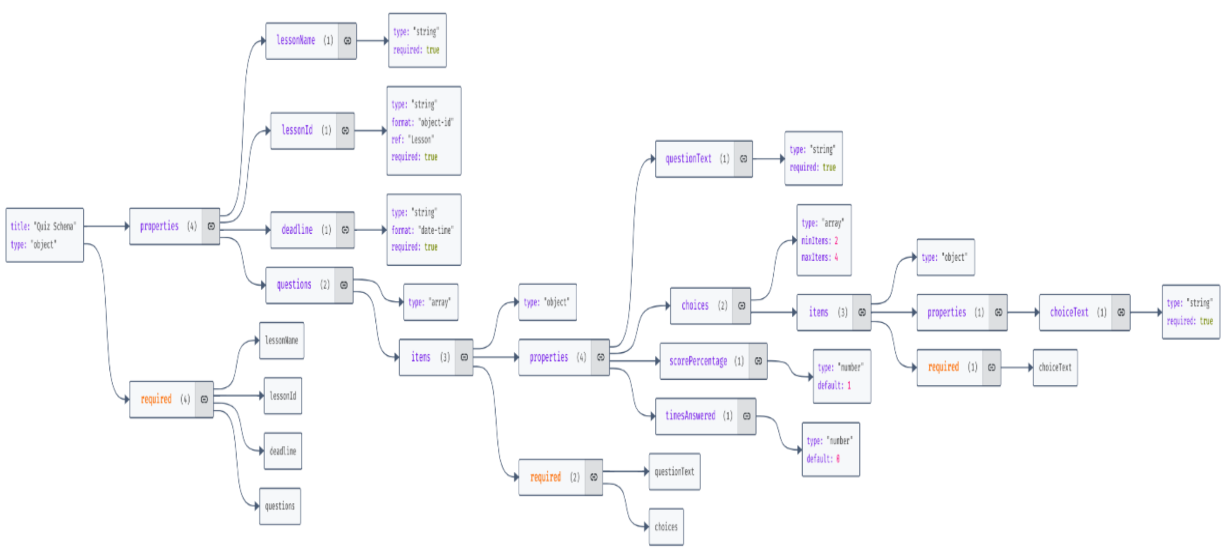
4.5.1 User Schema



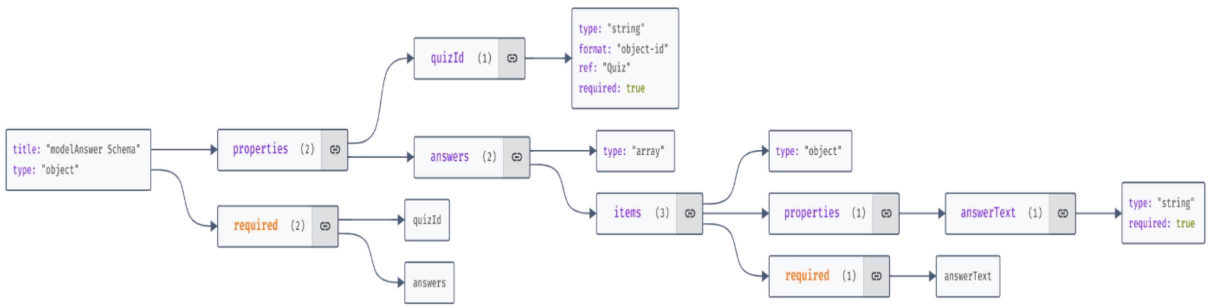
4.5.2 Lesson Schema



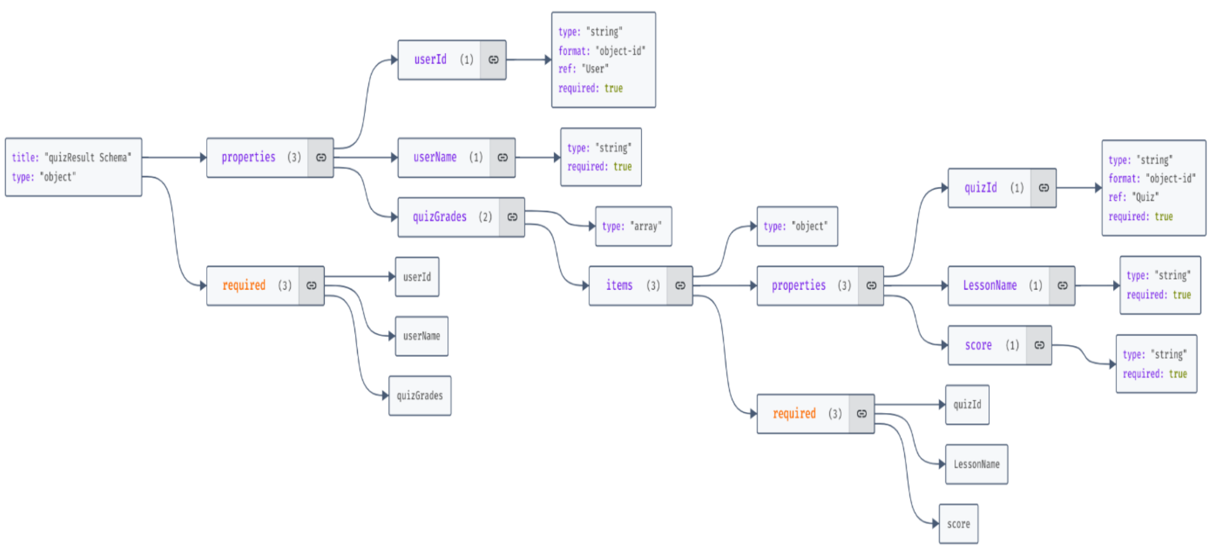
4.5.3 Quiz Schema



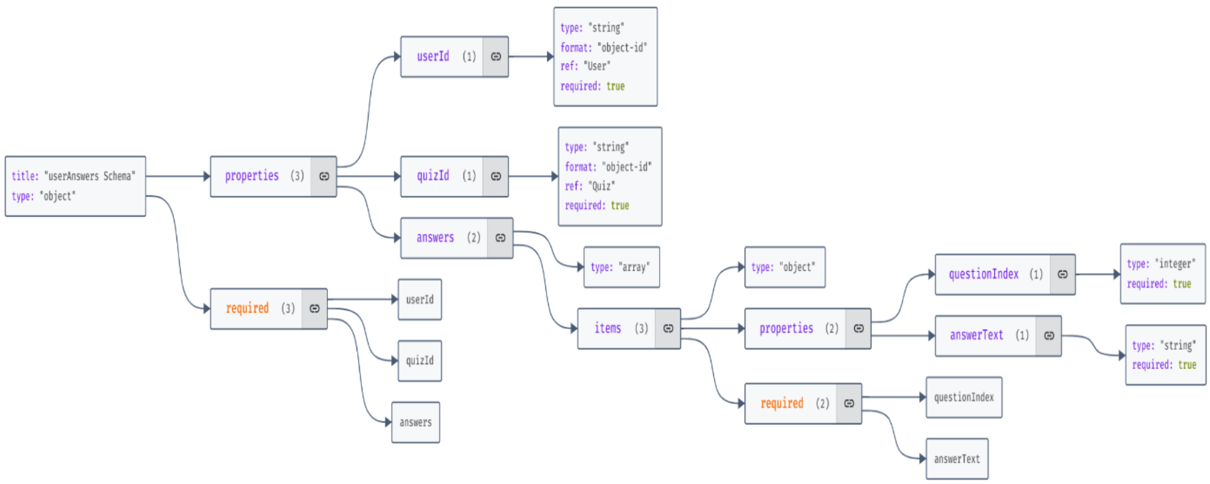
4.5.4 Model Answer Schema



4.5.6 Grades Schema



4.5.7 User's Answers Schema



Chapter 5: Implementation Details

This chapter delves into the implementation details of our system design, highlighting the key factors that influenced our approach. With a focus on precision and clarity, we offer a comprehensive overview of the main components of our system: the ML models, backend web application, and frontend web application.

5.1 Tools

5.2 Front-End

5.2.1 What is React?

React is a popular open-source JavaScript library used for building user interfaces, particularly single page applications where you need a fast, responsive user experience. **It allows developers to create large web applications that can change data, without reloading the page.** React primary feature is the ability to create components—self-contained modules that define their own structure, behavior, and state, which can be composed together to build complex UIs.

5.2.2 What is React used for?

React can be used to build web, mobile, and desktop applications, making it a versatile framework for cross-platform development, especially single-page applications (SPAs) where you need a dynamic and highly responsive user experience. React Native which is a framework based on React is specifically designed for mobile app development, while React Desktop allows you to create desktop applications using web technologies.

5.2.3 Why did we choose React?

- React **uses a virtual DOM, a lightweight copy of the actual DOM**. When the state of an object changes, React updates the virtual DOM, and then efficiently updates the real DOM to match this makes updates faster and more efficient.
- React uses a declarative paradigm, making it easier to understand and predict the behavior of the application. Instead of describing how the UI should change over time, developers declare what the UI should look like at any given point.

- Many large companies (e.g., Facebook, Instagram, Airbnb) and numerous startups use React, which demonstrates its reliability and scalability. Due to its popularity, there is a high demand for React developers, making it a valuable skill in the job market.
- It is easy to integrate with various back-end technologies makes it a popular choice for modern web and mobile application development.
- Using React can significantly enhance the development process and the performance of web applications. Its modular approach, performance optimizations, rich ecosystem, and strong community support make it a preferred choice for many developers and organizations.

5.2.4 What is Tailwind CSS?

Tailwind CSS is a **utility-first CSS framework** that enables developers to rapidly build custom user interfaces. Instead of using pre-defined components or class names, Tailwind CSS provides low-level utility classes that can be combined to create unique designs**. makes it easy to create responsive designs with its mobile-first responsive utilities.** Tailwind CSS works by scanning all your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file.

5.2.5 Why did we choose Tailwind CSS?

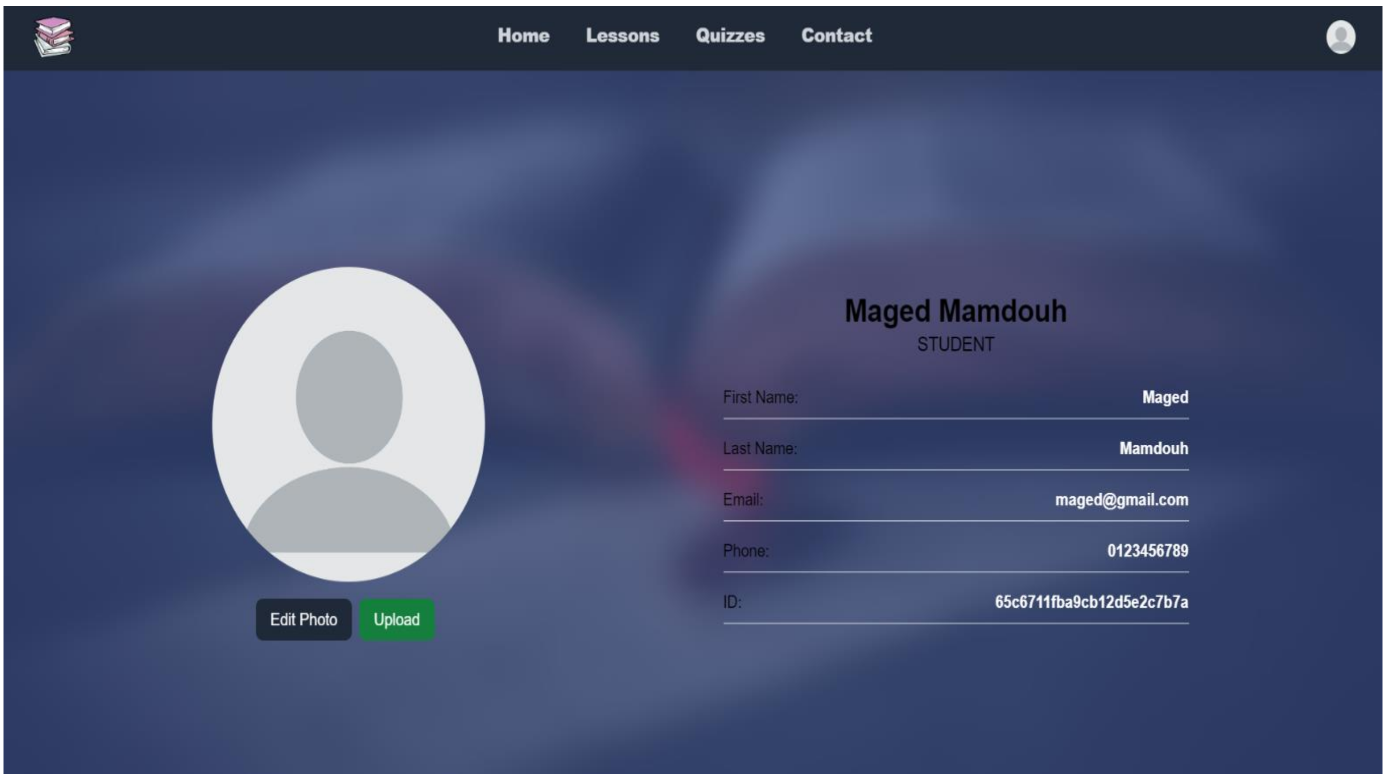
- Choosing Tailwind CSS offers several advantages, particularly for looking for a more streamlined, efficient, and customizable approach to styling. Here are some reasons why we choose Tailwind CSS:
- Tailwind's utility-first approach allows you to build designs directly

in your markup without writing custom CSS. This can speed up the development process as you don't need to constantly switch between your HTML and CSS files.

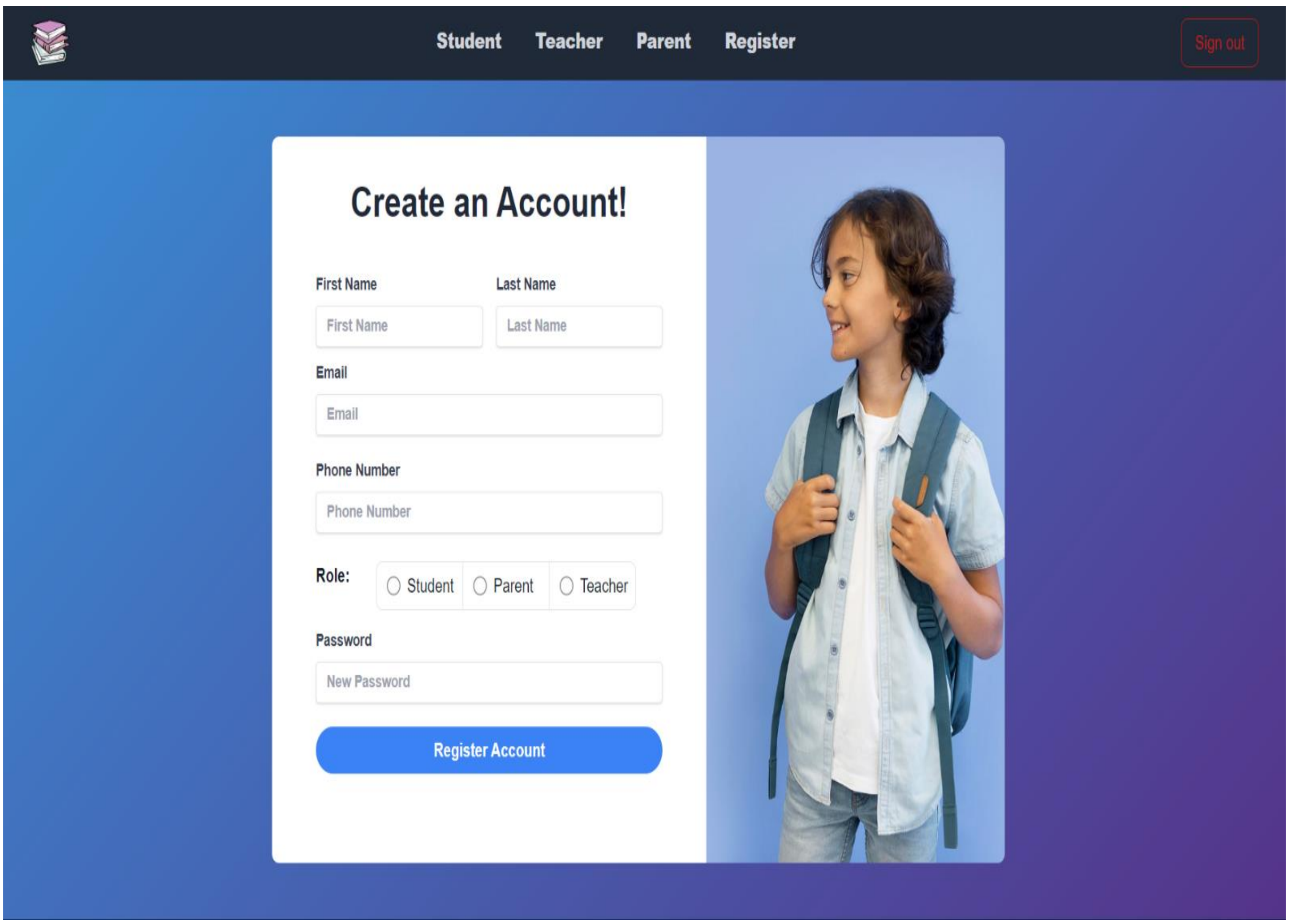
- The framework is designed to be efficient and performant, ensuring that only the necessary CSS is included in your production build.
- Tailwind responsive utilities are mobile-first, which means you start with styles for small screens and add styles for larger screens as needed makes it easy to create responsive designs.
- Tailwind CSS is highly customizable. You can define your own design system in the tailwind.config.js file, tailoring the framework to your project's specific needs makes it easy to implement custom themes and modify the default styles to fit your brand or design requirements.
- Tailwind CSS integrates seamlessly with modern JavaScript frameworks like React. This makes it a great choice for our project that uses component-based architecture.

5.2.6 Snapshots

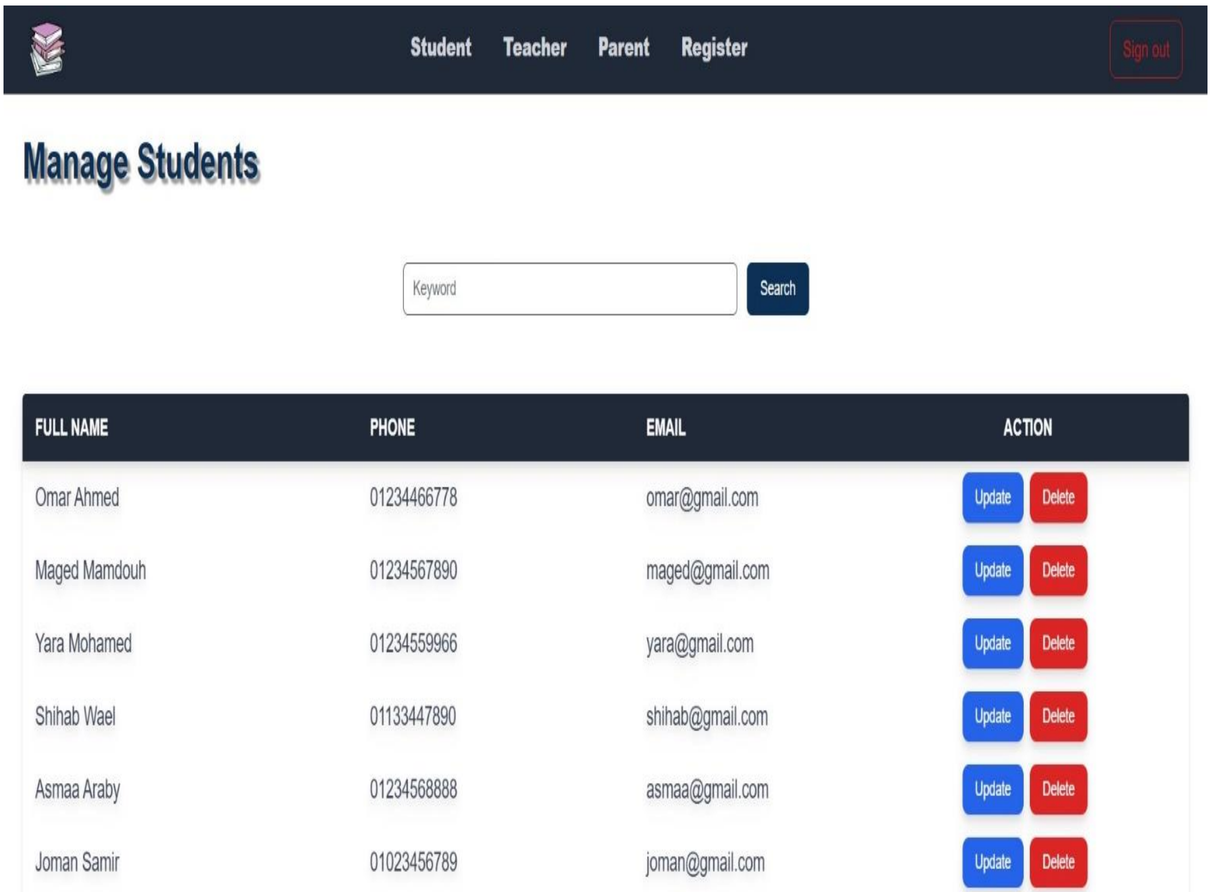
- This is profile page which enables the user to observe his data in addition making him/her to upload his photo and edit it in the future, the profile photo will appear in the navbar each time he/she enters the website.



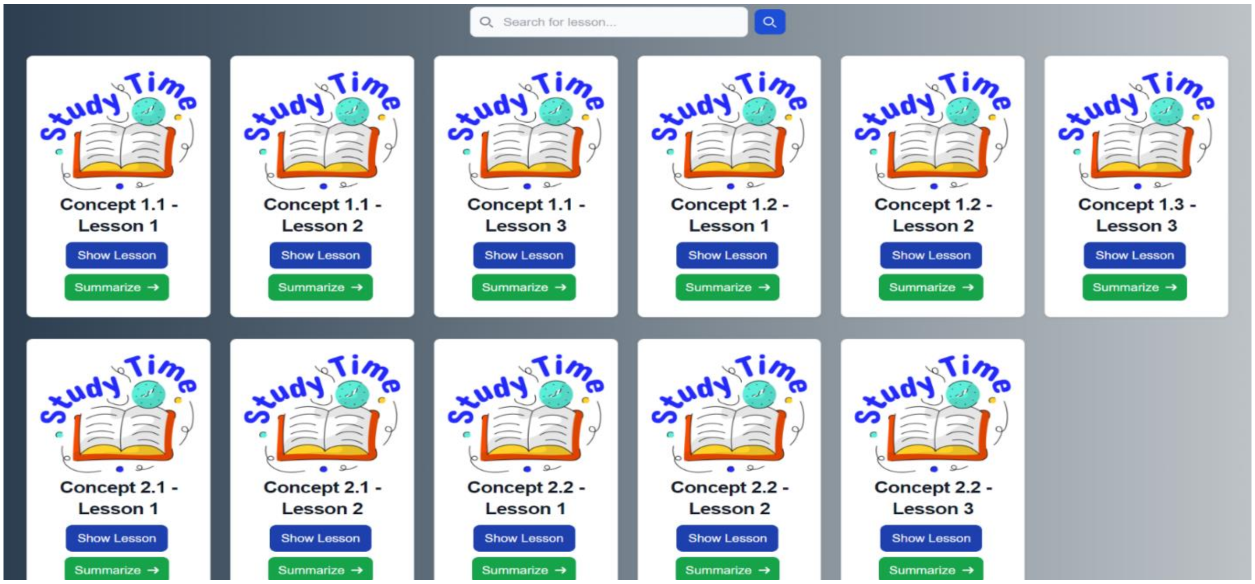
- This is the registration page which is managed by the administrator, the data of the users are entered by the admin of the website, also in the registration the admin specifies the user's role, and also the data if it is need to be updated the admin handled it.



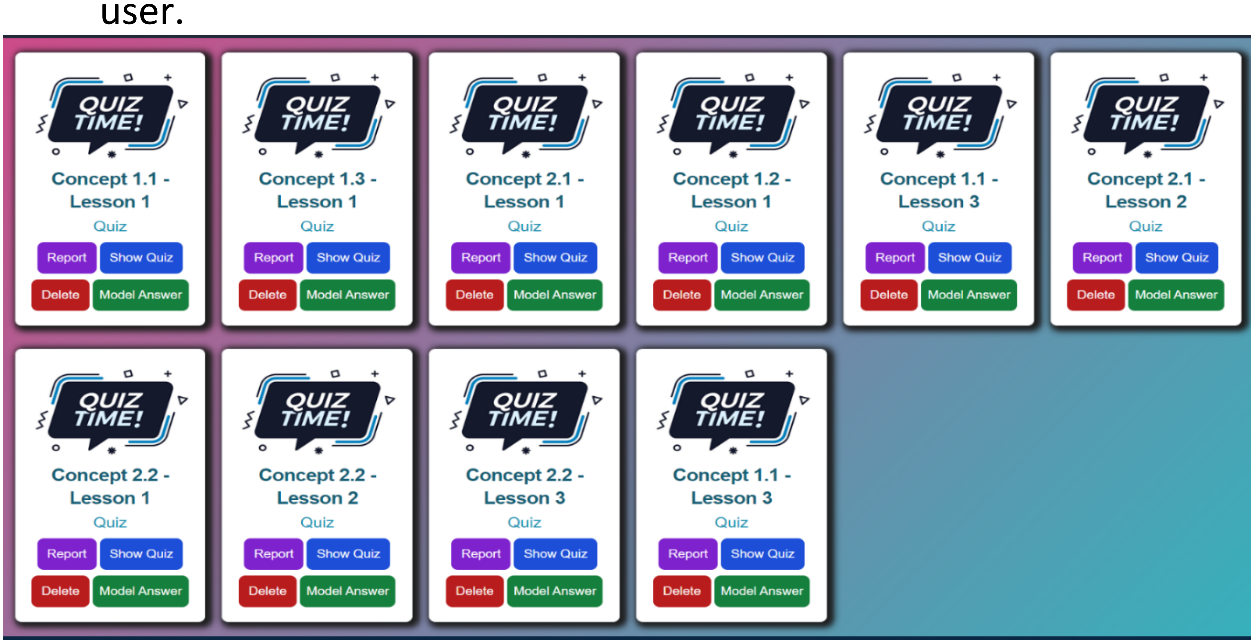
- This is the manage student page, which is handled by the admin, in which he/she can update the data of the user or delete the whole data of the user, also the admin can manage the teacher and parent from their manage pages.



- This is the interface of lessons page which enables the student to show lesson of his subject, also he/she can summarize this lesson.



- This is the quizzes interface for the teacher which provides some functions that will help him/her in managing the quizzes for the user.



5.3 Back-End

5.3.1 What is Node js ?

Node.js is a runtime environment that allows you to run JavaScript on the server side. It was developed by Ryan Dahl in 2009 and has since become a popular choice for building server-side applications. Here are some key features and aspects of Node.js , Node.js uses the V8 JavaScript engine, which is the same engine that powers Google Chrome. This allows developers to use JavaScript, a language traditionally confined to the client side (in browsers), on the server side as well.

5.3.2 Why we use node js with MVC model ?

Using Node.js with the MVC model offers clear separation of concerns, making the codebase more organized and maintainable. It enhances scalability by modularizing the application into models, views, and controllers, facilitating easier updates and debugging. This structure also improves code reusability and collaboration among developers. Additionally, the combination of Node.js's non-blocking I/O model and MVC's organized architecture results in efficient and scalable web applications.

5.3.3 Why we use mongo database ?

MongoDB offers schema flexibility, allowing storage of documents with different structures, which is ideal for evolving data models. It supports horizontal scalability through sharding, making it easier to handle large, distributed data sets. MongoDB can provide better performance for large volumes of data with quick read/write operations. Its document-oriented approach simplifies development by aligning closely with modern programming languages. It's well-suited for handling unstructured data in applications like content management and real-time analytics. Built-in replication and high availability features enhance reliability and fault tolerance

5.3.4 Why we use child process in our project ?

Using child processes to handle tasks such as storing questions and model answers, as well as summarizing uploaded lessons, can be an efficient way to manage resource-intensive operations in a Node.js backend.

```
const pythonProcess = spawn('python', ['machine/Quiz_data.py', pdfPath]);

pythonProcess.stdout.on('data', (data) => {
    const output = data.toString();
    const prints = output.split('\n');
    const questions_data = prints[0];
    const answers_data = prints[1];
    const questions = JSON.parse(questions_data);
    const answers = JSON.parse(answers_data);

    const newQuiz = new Quiz({
        lessonName: lessonName,
        lessonId: lessonId,
        deadline: deadline,
        questions: questions
    });
```

Using child processes in Node.js for integrating machine learning with a backend involves running resource-intensive tasks, such as machine learning inference or training, in separate processes. This allows the main server to handle requests efficiently without being blocked by these heavy tasks. Here's an explanation of the usage and benefits:

1. **Offloading Heavy Computing :**
   Machine learning tasks, like model inference or training, can be computationally expensive. Running these tasks in a separate child process ensures that the main server remains responsive to incoming client requests.
2. **Parallel Processing :**
   Child processes run independently of the main process. You can spawn multiple child processes to handle concurrent machine learning tasks, enabling parallel processing and better resource utilization.
3. **Crash Isolation :**
   If a child process crashes due to errors in the machine learning code, the main server process remains unaffected. This isolation increases the stability and reliability of your backend.
4. **Language Interoperability :**
   Child processes allow you to run scripts in different languages. For example, you can use a Python script for machine learning tasks while the main server is written in Node.js.
5. **Asynchronous Communication :**
   Child processes communicate with the parent process using message passing, which is non-blocking. This fits well with the asynchronous nature of Node.js.

5.3.5 What is middleware?

Middleware in the context of web development, particularly in frameworks like Express.js for Node.js, refers to functions that have access to the request and response objects in the application's request-response cycle. These functions can modify the request and response objects, terminate the request-response cycle, or pass control to the next middleware function in the stack.
Benefits of Using Middleware:

1. **Modularity and Reusability :**
   Middleware functions are modular and can be reused across different routes or applications. This promotes code organization and reduces duplication.
2. **Request Processing :**
   Middleware functions can preprocess or postprocess incoming requests and outgoing responses. For example, parsing request bodies, logging requests, or setting response headers.
3. **Error Handling :**
   Middleware functions can handle errors by catching exceptions or rejecting promises. They can also pass errors to the next error-handling middleware in the stack.
4. **Authentication and Authorization :**
   Middleware can enforce authentication and authorization rules by validating user credentials, checking permissions, and controlling access to routes.

5. **Route Handling :**
   Middleware functions can handle specific routes or groups of routes. This allows for fine-grained control over the behavior of different parts of the application.
6. **Request Filtering and Validation :**
   Middleware can filter or validate incoming requests to ensure they meet certain criteria or adhere to specific standards. For example, checking for CSRF tokens, validating input data, or sanitizing input.
7. **Chaining and Composition :**
   Middleware functions can be chained together to perform multiple operations on a request or response. This allows for complex request processing pipelines to be built with ease.
8. **Extension Points :**
   Middleware provides extension points for adding functionality to an application without modifying its core logic. This makes applications more flexible and easier to extend.

5.3.6 How we achieve security in our project ?

By implementing user roles and token generation functions, users gain access to personalized routes for performing specific tasks tailored to their roles, ensuring a secure and efficient user experience.. Here's how it works:

1. **Authentication :**
   Users authenticate themselves with the application using their credentials (e.g., username and password).

2. **Authorization :**
   Upon successful authentication, the application generates a token containing information about the user's role.

3. **Route Protection :**
   Each route in the application is protected by middleware that verifies the validity and role of the token attached to incoming requests.

4. **Access Control :**
   Based on the user's role encoded in the token, the middleware either allows or denies access to specific routes and functionalities.

5. **Specific Jobs :**
   Users are directed to routes and functionalities tailored to their roles, enabling them to perform their specific tasks while restricting access to unauthorized functionalities.

```
const verifyToken = (req, res, next) => {
    const authHeader = req.headers['Authorization'] || req.headers['authorization']
    if (!authHeader) {
        const error = appError.create("token is required", 400, httpStatusText.FAIL)
        return next(error)
    }
    const token = authHeader.split(' ')[1]

    try {
        const currentUser = jwt.verify(token, process.env.JWT_SECRET_KEY)
        req.currentUser = currentUser
        next()
    } catch (err) {
        const error = appError.create("invalid token", 401, httpStatusText.ERROR)
        return next(error)
```

5.3.7 Why we use multer package in our project ?

We use the Multer package in our project to simplify file upload handling in Node.js applications. Multer efficiently parses multipart/form-data, allowing us to easily manage file uploads. It provides configurable storage options and supports file filtering and validation for enhanced security. Additionally, it integrates seamlessly with Express.js, making it straightforward to add file upload capabilities to our routes.

```
const diskStorage = multer.diskStorage(
    {
        destination: function (req, file, cb) {
        cb(null, 'uploads')
            },
        filename: function (req, file, cb) {
            const fileName = file.originalname
            cb(null, fileName)},
    }
)
const fileFilter = (req, file, cb) => {
    const imageFile = file.mimetype.split("/")[1];
    if (imageFile === 'jpeg' || imageFile === 'png') {
        return cb(null, true)
    }
    else
        return cb(appError.create('file must be image', 404, false))
  }
const upload = multer({ storage: diskStorage, fileFilter })
```

5.4 ML

5.4.1 Main Libraries

- **PyPDF2 :**
  PyPDF2 is a versatile Python library for manipulating PDF files. It allows users to read, extract information, merge, split, and modify PDFs. PyPDF2 supports extracting text, metadata, and images from PDFs, and can also handle encrypted documents. It provides functionalities to rotate pages, add watermarks, and create new PDF files from scratch. The library is easy to use and integrates well with other Python applications, making it a popular choice for PDF manipulation tasks.

- **Transformers :**
  The Transformers library by Hugging Face is a powerful Python toolkit for natural language processing (NLP). It provides access to state-of-the-art pre-trained models for tasks like text classification, translation, summarization, and question answering. The library supports models from BERT, GPT, T5, and many others, leveraging the capabilities of transformer architectures. The pipeline API simplifies the application of these models, enabling quick and easy integration into projects without extensive NLP expertise. Users can perform complex tasks with just a few lines of code. The library is highly extensible, supports fine-tuning, and integrates seamlessly with popular deep learning frameworks like TensorFlow and PyTorch. It's widely used in both research and industry for its efficiency and ease of use.

5.4.2 Main Functions

1. PDF Text Extraction Function:

This initial step involves creating a function responsible for extracting text from a PDF file. The function should accept a PDF file as input and output the text content contained within.

```python
def extract_text_from_pdf(pdf_path):
    text = ""
    with open(pdf_path, "rb") as pdf_file:
        pdf_reader = PyPDF2.PdfReader (pdf_file)
        for page_num in range(len(pdf_reader.pages)):
            page = pdf_reader.pages[page_num]
            text += page.extract_text ()
    return text
```

2. Paragraph Splitting Function:

Following the extraction of text from the PDF, the next task involves splitting the extracted text into paragraphs. This function should be designed to split the text into paragraphs ranging from 3 to 7 sentences each.

```python
def divide_text_into_paragraphs(text):
    num_paragraphs = random.randint(3, 7)
    words = text.split()
    total_words = len(words)
    words_per_paragraph = total_words // num_paragraphs

    paragraphs = []

    for i in range(num_paragraphs):
        start_index = i * words_per_paragraph
        end_index = start_index + words_per_paragraph

        if i == num_paragraphs - 1:
            paragraph = " ".join(words[start_index:])
        else:
            paragraph = " ".join(words[start_index:end_index])

        paragraphs.append(paragraph)

    return paragraphs
```

3. Ensuring Correct Choices for Questions:

- **Appending Choices:**
  When creating multiple-choice questions, it's crucial to ensure the correct answer is included in the choices provided to the participants. This involves adding the correct answer to the choices list but placing it at a different index compared to the distractors.

```python
def append_element_at_index(lst, element):
    index = random.randint(0, 3)
    lst.insert(index, element)
    return lst
```

- **Removing Duplicates:**
  To maintain clarity and fairness in assessments, it's essential to remove any duplicate choices from the options provided to respondents. This ensures that each choice is unique and contributes meaningfully to the question.

```python
def remove_duplicates(lst):
    unique_list = []
    for item in lst:
        if item not in unique_list:
            unique_list.append(item)
    return unique_list
```

4. Generate Summary or Quiz :

Here we call all the helpers functions and the pipeline API to complete our task.

```
def generate_summary(pdf_text):
  summarizer = pipeline("summarization" , model = "pszemraj/led-large-book-summary")
  summary = summarizer(pdf_text,
                       max_length=(math.trunc(len(pdf_text) * 0.18)),
                       min_length=(math.trunc(len(pdf_text) * 0.15)),
                       do_sample=False)[0]['summary_text']
  summary = summary.replace(". ",".\n")
  return summary
```

```
def generate_quiz(paragraphs):
  questions = []
  answers = []
  choices = []
  for paragraph in paragraphs :
    inputs = tokenizer(paragraph, return_tensors="pt")
    outputs = model.generate(**inputs, max_length=100)
    question_answer = tokenizer.decode(outputs[0], skip_special_tokens=False)
    question_answer = question_answer.replace(tokenizer.pad_token, "").replace(tokenizer.eos_token, "")
    question, answer = question_answer.split(tokenizer.sep_token)
    question = question.strip()
    answer = answer.strip()

    input_text = " ".join([question, tokenizerD.sep_token, answer, tokenizerD.sep_token, paragraph])
    inputs = tokenizerD(input_text, return_tensors="pt")
    outputs = modelD.generate(**inputs, max_new_tokens=128)
    distractors = tokenizerD.decode(outputs[0], skip_special_tokens=False)
    distractors = distractors.replace(tokenizerD.pad_token, "").replace(tokenizerD.eos_token, "")
    distractors = [y.strip() for y in distractors.split(tokenizerD.sep_token)]

    distractors = append_element_at_index(distractors, answer)
    distractors=remove_duplicates(distractors)
    questions.append(question)
    answers.append(answer)
    choices.append(distractors)

  return questions, answers, choices
```

Chapter 6 : Conclusion and Future Work

In this final chapter, we thoroughly analyze our findings, comparing them with existing methods to highlight strengths and weaknesses. We offer insights into future improvements and address limitations for further exploration. The chapter concludes with a concise summary of our system's effectiveness and potential enhancements.

6.1 Conclusion

“The learning picnic” system successfully addresses the critical need for effective educational tools that support both teachers and students in the learning process especially for the fourth grade. By developing a system that automatically summarizes educational content and generates quizzes tailored to fourth-grade students, we have created a valuable resource that enhances classroom instruction and individual learning experiences.

Our system leverages advanced natural language processing techniques to provide clear and concise summaries, making complex material more accessible to young learners. The quiz generation feature offers diverse question formats that promote engagement and reinforce comprehension, ensuring that students can test and solidify their understanding of the material.

Throughout the development and implementation phases, we prioritized usability and adaptability, resulting in a user-friendly interface that teachers can easily integrate into their existing workflows. Preliminary feedback from educators has been overwhelmingly positive, highlighting the system's potential to save time and improve educational outcomes.

While our project has achieved its primary objectives, we recognize that there is significant potential for future enhancements. Incorporating adaptive learning algorithms, expanding to support multiple languages, and integrating multimedia elements are just a few of the many possibilities for further development. Additionally, extending the system's capabilities to include voice interaction, Real-Time Collaboration with AI Tutors and offline functionality will broaden its applicability and impact.

Ultimately, our project represents a significant step forward in leveraging technology to improve education. By continuing to innovate and expand upon our initial work, we can contribute to a more personalized and effective learning environment for students worldwide. We are excited about the future prospects of our system and its potential to make a lasting difference in the field of education.

6.2 Future Work

6.2.1 Expanded Grade Levels and Subjects and Multilingual Support:

For future enhancements we can expand the system to include additional grade levels and subjects will significantly increase the utility of our website. Currently, our platform is specialized for fourth-grade science. However, future developments could allow us to extend our services to encompass any grade level and subject area desired by teachers or students. This expansion will enhance the website's versatility and broaden its educational impact. We also can Extend the system to support multiple languages, making it accessible to a broader range of students. This is particularly useful in multilingual classrooms or regions.

6.2.2 Adaptive Learning Capabilities:

We also can Develop adaptive learning algorithms that customize quiz questions based on individual student performance. This can help address each student's strengths and weaknesses more effectively.

6.2.3 Collaborative Learning:

Further we can Add features that support collaborative learning, allowing students to work together on quizzes or share their summaries and insights. Which will provide students with a competitive advantage by fostering a sense of competition among their peers.

6.2.4 Voice Interaction:

Integrate voice recognition and synthesis to allow students to interact with the system using speech, making it more accessible for younger students or those with reading difficulties.

6.2.5 Location detection:

Further, we can enhance the website by incorporating location detection features. This would enable the platform to identify the region and governing educational authority of the student, allowing it to recommend questions that are likely to appear on their exams based on regional exam trends and standards.

6.2.6 Offline Capabilities:

Implement offline functionality so that students can access summaries and quizzes without an internet connection. This is particularly useful in areas with limited internet access.

6.2.7 Mobile Application:

In today's digital age, almost everyone possesses a personal mobile device. Therefore, developing a mobile application for our learning system would be highly beneficial. This would facilitate easy access for users at any time and from any location, while also enabling parents to conveniently monitor their children's progress.

6.2.8 Real-Time Collaboration with AI Tutors:

Develop real-time collaboration features where students can interact with AI tutors for instant feedback and assistance while working on quizzes or reviewing summaries.

References

[1] COLING 2022 Shared Task : LED Finetuning and Recursive Summary Generation for Automatic Summarization of Chapters from Novels.

[2] Transformer-based Models for Long Document Summarization in Financial Domain.

[3] LHS712EE at BioLaySumm 2023 : Using BART and LED to summarize biomedical research articles.

[4] Longformer : The Long-Document Transformer.

[5] Attention Is All You Need.

[6] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

[7] Efficient Few-Shot Fine-Tuning for Opinion Summarization.

[8] FINE TUNING T5 MODEL FOR TEST EXAMINATION AND ASSESSMENT.

[9] GPT-4 Technical Report.

[10] MQAG : Multiple-choice Question Answering and Generation for Assessing Information Consistency in Summarization.