

Read me a story

IT 496: Graduation Project Report
Product Release-2

Prepared by

Fahda Almazyd	441200834
Lujain Alruwais	441201363
Reema Alowerdi	439200449
Shahad Aldebasi	441200794

Supervised by
Dr. Balsam AlSugair
&
Dr. Manal Albahlal

Third Semester 1444
Spring 2023



Table of Contents

1 Introduction	8
2 Background	13
2.1 Machine learning	13
2.1.1 Types of learning	13
2.1.2 Transformers	14
2.1.3 Multi-class and multi-label classification	17
2.2 Evaluation	18
3 Literature Review	20
3.1 Related research	20
3.1.1 Transformers used in applications	20
3.1.2 Children's Literature as a Medium of Moral Education	21
3.2 Children's book classification	22
3.2.1 Jarir bookstore	22
3.2.2 Amazon	22
3.2.3 Summary of book classification	23
3.3 Applications for children's stories	24
3.4 Competitive Product Analysis	28
3.5 Our solution	29
4 System Design and Development	31
4.1 Methodology	31
4.2 System requirements	33
4.2.1 System Users	33
4.2.2 Requirements Elicitation and Analysis	33
4.2.3 User Interactions	34
4.2.4 Roadmap and Product Backlog	35
4.3 System design	41
4.3.1 Architectural diagram	41
4.3.2 Class diagram	42
4.3.3 Component Level Design	43
4.4 Data Design	46
4.4.1 Data Models	46
4.4.2 Data Collection and Preparation and Analysis	49
4.5 Interface Design	56

4.5.1	Sitemap	59
4.6	System Implementation	61
4.6.1	RoBERTa model	61
4.6.2	Prepare Flutter environment	65
4.6.3	Firebase and its connection with Flutter and admin interface	65
4.6.4	Admin interface (website)	66
4.6.4.2	Edit story	67
4.6.5	Reader interface (application)	68
5	System Evaluation	72
5.1	Experimental Results	72
5.1.1	Experimental Set up	72
5.1.2	Comparison between RoBERTa and DistilBERT	72
5.1.3	Evaluation	73
5.2	User Acceptance Testing	74
5.2.1	Reader and admin experiments	74
5.2.2	Demographics of Participants	75
5.2.3	Questionnaire/Interview Results	76
5.3	Quality Attributes (NFR testing)	84
5.4	Discussion	86
6	Conclusions and Future Work	89
6.1	Global and local impact	90
6.2	Problems and challenges encountered during the software development	90
6.3	Limitations of the system	92
6.4	The main contribution of the project	92
6.5	Future work	93
7	Acknowledgments	95
8	References	97
9	Appendices	101
9.1	Appendix A: Dataset Collection	101
9.2	Appendix B: User Acceptance Testing	102
9.3	Appendix C: Tasks for admin and reader	115

LIST OF TABLES

Table 1- Competitive Product Analysis	28
Table 2-Product Backlog	37
Table 3-Keywords.....	51
Table 4-Keywords examples.....	51
Table 5-comparison (RoBERTa and DistilBERT)	72
Table 6-Accuracy.....	73
Table 7- Improve the accuracy	73
Table 8-Demographics of Participants for admin	75
Table 9-Demographics of Participants for reader	76
Table 10-test questionnaire for admin	76
Table 11-test questionnaire for reader	80
Table 12-Quality Attributes	84
Table 13- SUS score guide	86

LIST OF FIGURES

Figure 1-Transformer.....	14
Figure 2-Multi head attention	15
Figure 3-loss function	15
Figure 4-multi-label	17
Figure 5-multi class.....	17
Figure 6-Recall.....	18
Figure 7-Accuracy	18
Figure 8-Precision.....	18
Figure 9-F-score.....	18
Figure 10-Little Stories	24
Figure 11- Storytel	25
Figure 12- BOOKA.....	26
Figure 13-epic!.....	27
Figure 14-Use case diagram.....	34
Figure 15-Roadmap	36
Figure 16-System Architecture	41
Figure 17-Class diagram	42
Figure 18-Component level design(fine-tuning RoBERTa).....	43
Figure 19-Component level design(Add story)	44
Figure 20-Component level design(Read story)	45
Figure 21-ER diagram.....	46
Figure 22-NoSQL data model.....	47
Figure 23-Manual Annotation Flowchart	50

Figure 24-Preprocessing steps.....	52
Figure 25-distrbution of story across morals	53
Figure 26-frequency of keywords	54
Figure 27-unwanted words	54
Figure 28- Lengths, mean, and standard deviation before cleaning	55
Figure 29-Lengths, mean, and standard deviation after cleaning	55
Figure 30-Icon Usability	56
Figure 31-Reduce short-term memory load.....	57
Figure 32- see less button	58
Figure 33- “See more” button	58
Figure 34-Sitemap.....	59
Figure 35-installation	61
Figure 36-import	61
Figure 37-read dataset.....	62
Figure 38-Tokenize & encode code	62
Figure 39-Train the model code.....	63
Figure 40-Evaluation libraries	63
Figure 41-save model.....	64
Figure 42-predict code	64
Figure 43-admin library	66
Figure 44-flask library	66
Figure 45-classify story code	67
Figure 46- Edit story code.....	67
Figure 47- splitting code	68
Figure 48-splitting flowchart	68
Figure 49- text to speech code 1	69
Figure 50- text to speech code 2	70

Read me a story

Fahda almazyed¹, Lujain Alruwais², Reema Alowerdi³ and Shahad Aldebasi⁴

¹Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441200834@student.ksu.edu.sa

²Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441201363@student.ksu.edu.sa

³Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 439200449@student.ksu.edu.sa

⁴Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441200794@student.ksu.edu.sa

Abstract (English):

Morals play an essential role in why storytelling evolved and they give structure to stories. Whether shared by teachers, family, or authors, stories are useful to children by providing them with psychological and educational benefits. However, some authors use ambiguous titles to be creative. This makes it harder for the reader to find the book they want to read under a specific moral, since they could misinterpret the title.

Despite the availability of several reading apps, we have discovered that none of them fulfilled the urgent need for such applications since we found that parents/educators find it difficult to find books that are categorized according to specific morals via an online questionnaire.

“Read me a story” is a tablet application that helps parents and educators find stories about the morals they want to teach their children while also having the option to allow their children to listen to the stories directly. By using the latest technology of transformers, RoBERTa, we developed a classifier model based on a machine learning algorithm that classifies the children’s stories into five defined morals. We collected data, fine-tuned, and tested the model. We fine-tuned the model to evaluate the system’s effectiveness until we achieved 91% accuracy. By doing this, the reader is able to choose books based on what moral their child needs to learn.

Abstract (Arabic):

تلعب القيم الأخلاقية دوراً أساسياً في تطور القصص وتوفير هيكل لها. سواء كان يشاركها المعلمون ،الأسرة، أو المؤلفون ، فإن القصص مفيدة للأطفال من خلال تزويدهم بفوائد نفسية وتعليمية. ومع ذلك ، يستخدم بعض المؤلفين عنوانين غامضة كوسيلة للإبداع. وهذا يجعل من الصعب على القارئ العثور على الكتاب الذي يريد قرائته وفقاً لقيم أخلاقية معينة ، لأنهم قد يسيرون تفسير العنوان.

باستخدام أحدث تقنيات المحولات RoBERTa ، قمنا بتطوير نموذج مصنف يعتمد على خوارزمية التعلم الآلي التي تصنف قصص الأطفال إلى خمسة قيم أخلاقية محددة. قمنا بجمع البيانات وتدريبها واختبارها. قمنا بتدريب النموذج على تقييم فعالية النظام حتى حققنا دقة بنسبة 91٪. ، من خلال القيام بذلك ، يستطيع القارئ اختيار الكتب بناءً على الفئة (القيمة الأخلاقية) التي يحتاج الطفل إلى تعلمها.

وقد لاحظنا الحاجة الملحة لمثل هذه التطبيقات لأننا وجدنا أن الآباء / المعلمين يجدون صعوبة في العثور على كتب مصنفة وفقاً لقيم معينة من خلال الاستبيان. "اقرأ لي قصة" هو تطبيق لوجي يساعد الآباء والمعلمين في العثور على قصص حول الأخلاق التي يرغبون في تعليم أطفالهم أو جعل أطفالهم يستمعون إليها بشكل أسرع وأسهل من خلال تصنيف الكتب وفقاً لقيم أخلاقية محددة.

Keywords: Morals; Storytelling; Classified stories; Values; Children stories; Story; Reading; Classifier model; Audio story



INTRODUCTION

— READ ME A STORY 

1 Introduction

Reading and storytelling for children promote their brain development and imagination, it also develops language, emotions, and strengthens relationships. It is one of the best ways for teaching children. “Storytelling has always played an important role in human society. It serves as a useful tool to entertain and to educate adults and children alike” [1]. Storytelling is an ideal medium through which children can be taught valuable lessons about life because the engaging narratives captivate even young children’s attention. For this reason, stories have been told for generations and continue to be used throughout the world today to teach children moral values.

We want our children to develop moral principles and values like empathy, compassion, and respect as they grow up. However, some children's book authors get more creative than they need to be and name their books with ambiguous titles. This only makes it more difficult for educators and parents to choose what kind of topic they would like to teach their children. Countless situations would require a parent/educator to teach their child morals, such as if the child faces bullying in school. The parent/educator would want the child to be brave and patient in these situations, and what better way than with examples from a book? Dr. Seuss's world-famous book ‘The Cat in the Hat’ opens up about trust, responsibility, social expectations, and honesty [2]. But how will the parents/educators know what message the book is trying to spread when the title is this vague?

Raising children through stories is one of the most popular methods of explaining and applying concepts. It is sometimes difficult for parents and teachers to define the concept of the story from its title. So, by creating "Read me a story", we will provide a tablet application that helps parents and other adults in charge of children find stories that match the manners they are looking for.

The system will enable the users to perform the following functionalities:

- Users can read numerous books under specific morals which are friendship, respect, honesty, bravery, and patience.
- Users can add books to a favorite list for easy access.
- Users can listen to books.
- Users can bookmark a page to pick up where they left off.
- The books that have been bookmarked will show in continue reading for easy access.
- Users can read books under the favorite list offline.
- Admin will be able to add a book that will then be categorized to a specific moral by the classifier model.
- Admin will be able to disapprove the moral that the classification model categorized, which will save the story to the database to be used in improving the model.
- Admin will be able to edit and delete books.
- Admin will be able to search and filter stories in edit and delete page.
- Admin can view an overview of stories.

The “Read me a story” tablet application will be used by parents, children, or other adults who are in charge of children of a certain age (4-12) to teach them chosen manner/s through stories. The child him/herself can use the application too since it supports listening to the story functionality. The user will not be able to add stories or search for them online, but the application will provide a list of morals to select from and a list of stories under each value. A classifier model will be used to categorize the stories, but it will exclude any story that lacks morals and only supports the English language.

Our vision is to design an application for parents, teachers, and other adults in charge of children to make it easier for them to access books under the morals they would like to teach their children. “Read me a story” is a tablet application that provides books under specific morals, for whoever is responsible for a child.

We developed the application by following the Agile software development framework. We went through many stages during the software development process. First, we had to collect and manually annotate the data since we did not find datasets of children's books and their respective morals. Then, we had to clean the data by removing punctuation symbols and stop words, normalizing white spaces, etc. We then fine-tuned the model until it reached satisfactory accuracy. After that we then began developing the user and admin interface, the user application was developed using the Dart programming language while the admin website was developed using Python, specifically Flask web framework. User, admin, and books' information were stored using Firebase. Finally, we conducted user acceptance testing to evaluate the compliance of the system with our requirements and whether it performs up to expectations.

Our project designs and develops an iOS tablet application that gives parents, educators, and adults in charge of children easy access to books categorized under morals. The application uses a classifier model based on machine-learning algorithms that categorize the books based on the morals the book represents. This solution will help adults who are interested in teaching children morals through books. This will in turn help the child to build their character, tell right from wrong, boost their confidence, counter bad influence from peers, and many other benefits [3]. Providing an application with English books especially allowing the children to listen to the story will also help them improve their English.

“Read me a story” is different from other competitors since it specializes in categorizing books into morals. It provides features that will make it easier for parents to teach their children morals, whether by reading them the book under the moral they would like to teach or allowing the child to listen to the book themselves. Unlike storytel which specializes in audiobooks and lists books under superficial topics such as “Dragons” or “Princess”, our product is explicitly targeted at moral topics[4].

This document contains a total of 8 sections. The first is the introduction which includes the problem, the application scope, the vision, the objectives, the stages of the software development process, and the main contribution of our project. The second chapter covers background information that explains the domain of our application and its essential components, machine learning and its types, transformers and their examples, multi-class and multi-label classification, and evaluation. The third chapter is a literature review where we provided research related to our application, examples of how bookstores and applications

classify books, and a list of similar applications and their features which we compared to our application. The fourth chapter is system design and development which talks about the methodology we used to develop the software. This chapter also explains the system requirements, the user interactions with the software, and the application roadmap and backlog. In addition to covering the system design, the chapter also describes our data design and the implementation steps of key and challenging parts of the code. The fifth chapter is system evaluation, in this chapter we conduct system testing to ensure that the application and website are functioning as expected. To achieve this, we carried out user acceptance testing on both the application and the website. We have also conducted the NFR testing. The sixth chapter is the conclusion and future work that covers mainly our application's global and local impact, problems and challenges encountered during the software development, limitations of our application, the main contribution of our project, and some features that we plan to add in the future. The seventh chapter is acknowledgement, where we acknowledge everyone who contributed to the completion of this project. Finally, the last section is references, where we gathered all references that we used to get information.



BACKGROUND

— READ ME A STORY 

2 Background

Overview:

“Read me a story” is a tablet-based application, it focuses primarily on categorizing stories according to moral topics such as respect, honesty, etc. In this chapter, we will provide comprehensive information about machine learning algorithms, methods of evaluating models, transformers, and the methodology that we will use in our application.

2.1 Machine learning

Machine learning is a type of artificial intelligence (AI) that allows systems to learn and improve from experience without being explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it to learn for themselves [5]. Many of today's leading companies, such as Facebook, Google, and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies. Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions [6]. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

2.1.1 Types of learning

1. Supervised learning

In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm are specified.[7].

2. Unsupervised learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention [8].

3. Reinforcement Learning

Reinforcement Learning is a feedback-based machine learning technique. In such a type of learning, agents (computer programs) need to explore the environment, perform actions, and on the basis of their actions, they get rewards as feedback. For each good action, they get a positive reward, and for each bad action, they get a negative reward. The goal of a reinforcement learning agent is to maximize the positive rewards. Since there is no labeled data, the agent is bound to learn by its experience only [9].

4. Semi-supervised Learning

Semi-supervised Learning is an intermediate technique of both supervised and unsupervised learning. It performs actions on datasets having few labels as well as unlabeled data. However, it generally contains unlabeled data. Hence, it also reduces the cost of the machine learning model as labels are costly, "but for corporate purposes, it may have few labels. Further, it also increases the accuracy and performance of the machine learning model" [9].

2.1.2 Transformers

A transformer model is a neural network that learns context and thus meaning by tracking relationships in sequential data like the words in this sentence, "a model that uses attention to boost the speed with which these models can be trained"[10].

Transformer models apply an evolving set of mathematical techniques, called attention or self-attention ("Self-attention is the method the Transformer uses to bake the "understanding" of other relevant words into the one we're currently processing".[10]), to detect subtle ways even distant data elements in a series influence and depend on each other [11].

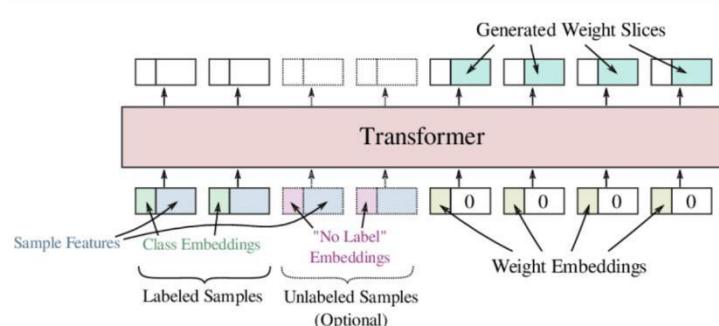


Figure 1-Transformer



In Figure-1 it shows the structure of the tokens passed to and received from a transformer model. Both labeled and unlabeled (optional) samples can be passed to the transformer as inputs. Empty output tokens indicate ignored outputs [12].

Word embedding- a learned representation for text where words that have the same meaning have a similar representation. It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems [13].

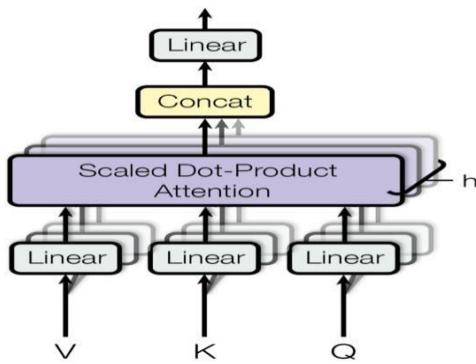


Figure 2-Multi head attention

This chart in Figure-2 represents *Multi-head Attention*, Multi-head Attention is a module for attention mechanisms that runs through an attention mechanism several times in parallel. The independent attention outputs are then concatenated and linearly transformed into the expected dimension [14].

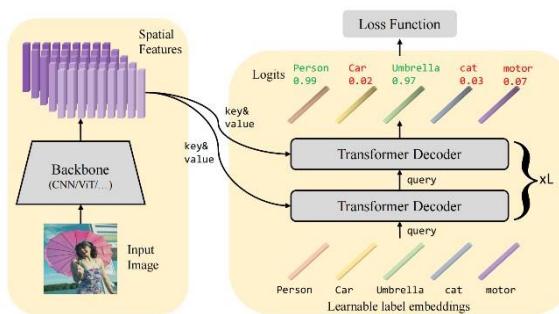


Figure 3-loss function

The *loss function* in Figure-3 is the function that computes the distance between the current output of the algorithm and the expected output. It is a method to evaluate how your algorithm models the data [15].

2.1.2.1 Example of transformer model

- **BERT**

BERT (Bidirectional Encoder Representations from Transformers) makes use of Transformers, an attention mechanism that learns contextual relations between words (or subwords) in a text. “In its vanilla form, Transformers include two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task”. Since BERT’s goal is to generate a language model, only the encoder mechanism is necessary [16].

- **ALBERT**

ALBERT is a Transformer architecture based on BERT but with much fewer parameters. This is accomplished by using two parameter reduction techniques. “The first is factorized embeddings parameterization”. By decomposing the large vocabulary embedding matrix into two small matrices, the size of the hidden layers is separated from the size of the vocabulary embedding. This makes it easier to grow the hidden size without significantly increasing the parameter size of the vocabulary embeddings. The second technique is cross-layer parameter sharing. “This technique prevents the parameter from growing with the depth of the network” [17].

- **ROBERTA**

RoBERTa (Robustly Optimized BERT pre-training Approach) is an NLP(Natural language processing) model and is a “modified version (by Facebook) of the popular BERT model”. This is more like an approach to better train and optimize BERT. Also, RoBERTa is a transformers model pretrained on a large corpus of English data in a self-supervised fashion. “It is a robustly optimized method for pretraining NLP systems that improves on”[18].

2.1.3 Multi-class and multi-label classification

In machine learning, multiclass or multinomial classification is a classification task with more than two classes. Each sample can only be labeled as one class. (classifying instances into one of two classes are called binary classification)[19]. On the other hand, multilabel classification is a generalization of multiclass classification, which allows us to classify data sets with multiple target variables. A multi-label classification involves several labels that represent the outputs of a prediction"; in the multilabel problem, there is no limit to the number of classes to which the instance can be allocated[20].

Multiclass classification should not be confused with multi-label classification, where multiple labels are to be predicted for each instance [19].

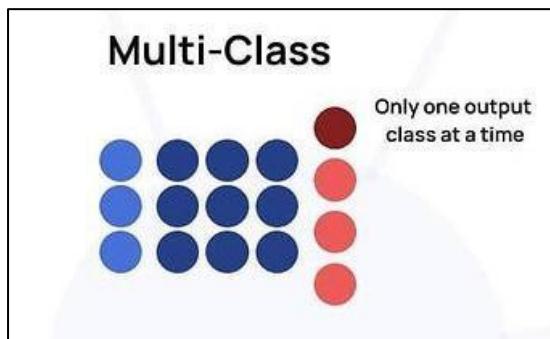


Figure 5-multi class

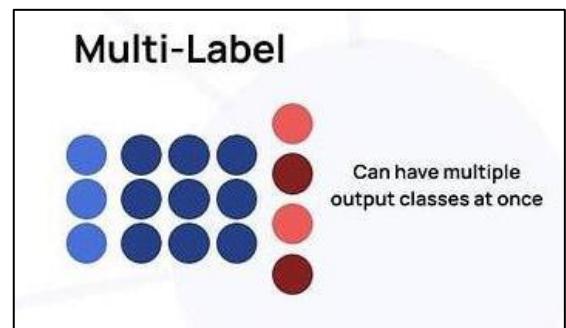


Figure 4-multi-label

Figure-4 is an example of multi-label, if we are building a model which predicts all the clothing articles a person is wearing, we can use a multi-label classification model since there can be more than one possible option at once [21].

Figure-5 is an example of multi-class, if we are making an animal classifier that classifies between Dog, Rabbit, Cat, and Tiger, it makes sense only for one of these classes to be selected each time [21].



2.2 Evaluation

Evaluation is an important step in classifying data. It helps in the prediction of how well the final model will work in the future, and how accurate this model is.

“Recall: The ability of a model to find all the relevant cases within a data set”[23]. Recall can be defined mathematically as the number of true positives divided by the number of false negatives [23].

$$Recall = \frac{TP}{TP + FN}$$

Figure 6-Recall

Accuracy represents “the number of correctly classified data instances over the total number of data instances” [22].

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Figure 7-Accuracy

Precision: The ability of a classification model to identify only the relevant data points. Mathematically, precision the number of true positives divided by the number of true positives plus the number of false positives [23].

$$Precision = \frac{TP}{TP + FP}$$

Figure 8-Precision

The F1 score is defined as the harmonic mean of precision and recall [24]. It is often useful when computing an average rate.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Figure 9-F-score



LITERATURE REVIEW

— READ ME A STORY 

3 Literature Review

Overview:

In this chapter, we first included some research that is relevant to our project to go into detail with. After that we looked through some bookstores/applications for books/stories to get a sense of how the majority of children's stories are being categorized, to consider finding out if categorizing children's stories according to their morals is actually a wise decision. Finally, we provided an analysis of related applications to identify their benefits and drawbacks.

3.1 Related research

In this section we will provide some research that is related to our application and will help give an insight into our project. We first mentioned transformers then talked about Multi-class Text Classification using BERT and TensorFlow, and finally morals with their significance.

3.1.1 Transformers used in applications

Tathagata Raha and his team elaborated on the impact of social media in our daily lives. They also highlighted the misleading information on social media and its effect on our lives. They proposed an approach to detect fake and real news about COVID-19. They used the dataset containing posts and tweets collected from Facebook, Twitter, and Instagram. They have split the dataset into training tests and validation parts. They tried different baseline models on this dataset and also used different transformer models. And results clearly show that their RoBERTa model achieves a 0.9864 F1-score, and their Electra model achieves a 0.9827 F1-score on the official test set[25].

Another application that used transformers was an application that focused on recommending topics for software repositories. This helps developers and software engineers' access, document, browse, and navigate through repositories more efficiently. By giving users the ability to tag repositories, GitHub made it possible for repository owners to define the main features of their repositories with a few simple textual topics. In this study, they proposed several multi-label classifiers to automatically recommend topics for repositories based on their textual information including their name, description, README files, wikipages, and their file

names. They first employed rigorous text-processing steps on both topics and the input textual information. They then mapped 29K sub-topics to their corresponding featured topics provided by GitHub. Then they trained several multi-label classifiers including DistilBERT-based models for predicting 228 featured topics of GitHub repositories. They evaluated the models both quantitatively and qualitatively. According to users' assessment, the approach can recommend on average 4.48 correct topics out of 5 topics[26].

3.1.2 Children's Literature as a Medium of Moral Education

O'Sullivan defines moral education as the “deliberate instruction in basic values and morals ideally woven into lessons throughout the curriculum and community”[27]. Compassion, fairness, kindness, honesty, perseverance, courage, justice, empathy, and acceptance are just a few of the universal principles he suggests should be taught in schools. Moral education has become a subject of interest to educators and parents. Schools should therefore teach moral education, values, and citizenship because building good character in children is an absolute necessity. According to Weaver, “literary characters have almost the same potential for influencing the reader as the real people with whom a reader might share a reading experience” [28]. He further affirms that character counts most in teaching children character education. He gives six (6) pillars of good character as: trustworthiness, respect, responsibility, fairness, caring, and citizenship. Leming asserts that many character education advocates suggest that educators should involve their local communities in identifying and defining the virtues to be woven into their educational goals[29].

3.2 Children's book classification

We provided some examples of various bookstores/applications categorizations, because we thought getting a sense of how the majority of children's books are being categorized would help us learn more about our domain/research. We also decided to adopt Jarir and Amazon categorizations, because Amazon is the most well-known online bookstore and Jarir is the most well-known local physical bookstore, both offer English language books and stories.

3.2.1 Jarir bookstore

We visited Jarir bookstore to get an idea of how they categorize their books, we found out that they classify it into 8 categories which are:

- Activity & coloring
- Children reference
- Stories with sound
- Novelty books
- Islamic books
- Numbers, reading and writing learning
- Improving talents
- The child's age (1-5, 6-9, 10 and above)

3.2.2 Amazon

We also gave Amazon a look into their categorization of books. They classify it into 20 categories which are:

- Holidays & Celebrations
- Education & Reference
- Growing Up & Facts of life
- Activities, Crafts & Games
- Animals
- Arts, music & Photography

- Fairy tales, Folk tales & Myths
- Literature & Fiction
- Religions
- Geography & Cultures
- Science nature & how it works
- Biographies
- Sports & Outdoors
- Science fiction & Fantasy
- Humor
- Action & Adventure
- Comics & Graphic novels
- Early learning
- History
- Cars, trains & things that go

3.2.3 Summary of book classification

After looking at the book categories on Amazon & Jarir bookstore, which are both good in their own way, we realized that neither one of them categorizes books according to morals. Based on our research, we came to the conclusion that it is difficult to find a library or book-selling website that categorizes its books/stories into morals. Our application, which classifies stories based on morality, would be a perfect choice if parents need to quickly teach their children a lesson when they start displaying bad behavior.

3.3 Applications for children's stories

In this section, we will illustrate the applications that allow the user to read different stories.



Little Stories

Little stories is a storytelling mobile application that provides children stories.

Here is a list of little stories' main features:

- It provides stories with pictures.
- The child can play the main role by applying the child's name.
- You can make your own audio books (voice-over the story)

But just like any other application, Little Stories has some drawbacks as well. For starters, they only offer one story for free; the rest require a paid subscription. Additionally, their user interface is somewhat confusing because the stories are not categorized. Finally, they do not provide audio books unless you voice over the story.



Figure 10-Little Stories



Storytel

Storytel is a mobile application for reading stories/books and storytelling, it has audiobooks, eBooks, and stories.

Here is a list of Storytel's main features:

- It has a child's mode, where the child can find adventure in a space with just children's stories.
- Audio stories.
- Switch between listening and reading anywhere in a book.
- Browse and share reviews and reactions. You can build your bookshelf and get personalized recommendations.

The application has certain drawbacks as well. For instance, you may only listen to a 30-second sample of the book; the rest requires a paid subscription. In addition, we don't think the categorization of the stories is particularly obvious; for example, they have a category called "English" with 20+ random subcategories underneath it.

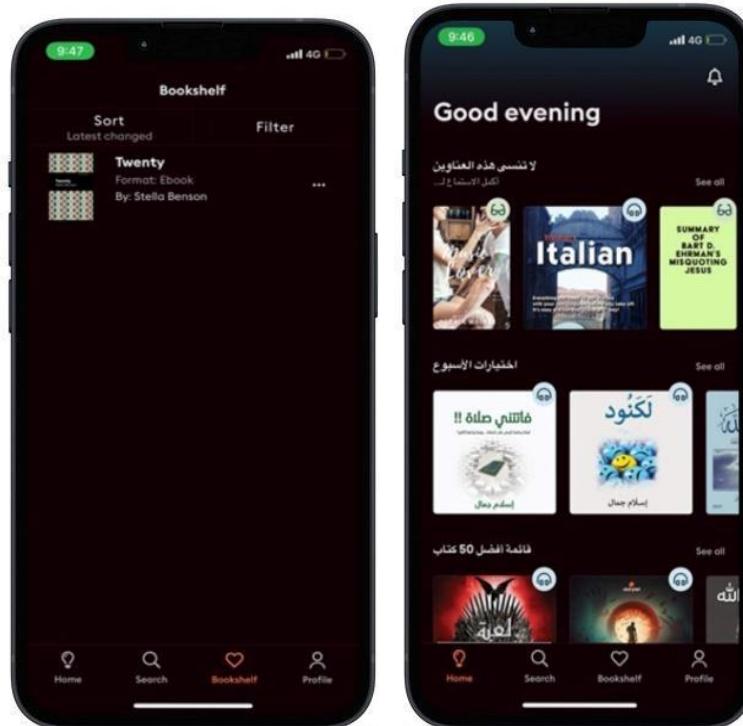


Figure 11- Storytel



BOOKA

BOOKA is an eBook mobile and tablet application directed towards children from ages 0-9.

BOOKA aims to enhance children's learning through fun and vibrant stories.

Here is a list of this app's main features:

- Provides plenty of books dedicated to educating young children.
- Provides a 'read to me' function that allows a narrator to read the story.
- Provides books in Spanish.
- It has a clear and child-friendly interface with certain pages that can only be accessed by an adult through a simple question.

After analyzing and interacting with this app, while it does have many interesting and new features, it has some downsides also. One of those being that the app is basically forcing the parent/educator to pay since it will not allow you to read past a few pages without the subscription plan showing up. Another drawback is that while it does limit the child from accessing the settings and the payment page by asking a multiple-choice question, couldn't the child just guess correctly and then pay for the subscription plan or mess up the settings?



Figure 12- BOOKA



epic!

Epic! is an interactive reading mobile and tablet application that provides educators and children access to books, audiobooks, learning videos, etc. Epic! aims to develop children's fluency and vocabulary skills through books.

Here is a list of this app's main features:

- Provides books on several different topics such as honesty and friendship and in multiple languages such as Spanish and Chinese.
- Provides personalized book recommendations based on the user's interests.
- The ability to put several books in the favorites' section.
- Enable the user to download books to read offline.
- Offers the user badges and rewards to encourage them to read more.

After interacting with this app, we noticed that it has a few downsides. First is that it only allows a child to read one book a day with a limited library unless they pay for a monthly or yearly subscription. Not only that, but it also only allows the child to download a book to be able to read offline after they pay for a subscription. However, it provides a filter function that allows you to categorize books based on age, whether the book is fiction or not, and several other characteristics, but not morals.

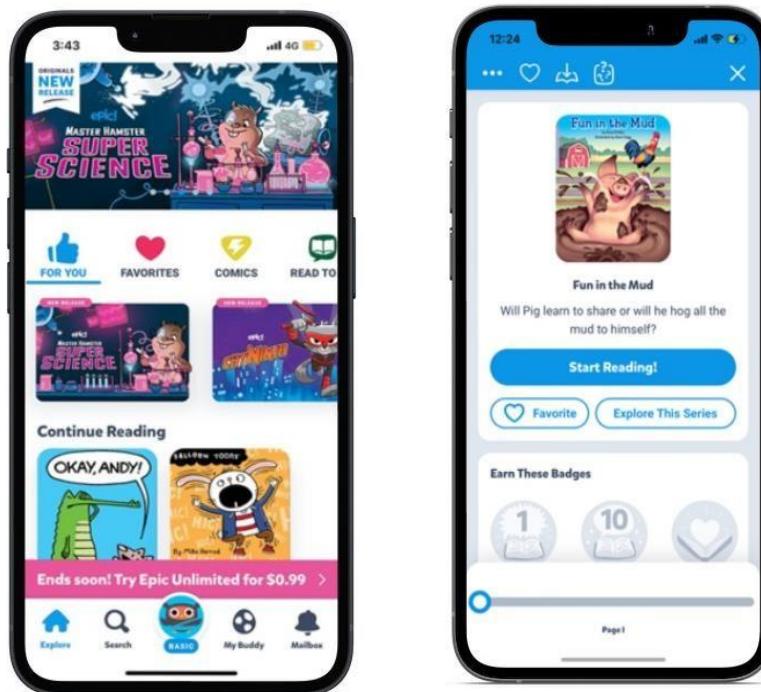


Figure 13-epic!

3.4 Competitive Product Analysis

Table 1- Competitive Product Analysis

Features	Applications				
					
Categorize stories based on moral topics		Little stories			Read me a story
Categorize stories based on different interests	✓		✓		
Audio stories	✓		✓	✓	✓
Stories with pictures		✓	✓	✓	✓
Free stories		✓			✓
Favorite List	✓			✓	✓
Supports Arabic	✓				
Bookmark					✓

Stories/books for all ages	✓				
Ratings and reviews	✓				

3.4 Summary of competitive product analysis

In conclusion, we analyzed several different competitors that have similar ideas to ours and noticed a few things. In Table-1 we have noticed that none of our competitors provide the functionality that our application focuses on, and that is categorizing stories based on moral topics. So, although some of these applications categorize based on interest, none of these applications focus on morals and raising awareness towards topics we place immense value on. For which we hope this will result in amassing the favor of educators and parents who also care about morals as much as we do. We also noticed that storytel, epic!, and BOOKA only allow access to books in exchange for a monthly or yearly subscription, thus not many applications allow access to free books like “Read me a story” does. Finally, we found that the favorite list feature is only available in one other application which is storytel. This is also considered an important feature since it gives parents/educators easy access to books they were planning to read.

3.5 Our solution

Although stories address morals and are a valid way to discuss them with children, we couldn't locate a bookstore or an application that categorizes books based on morals as ours does, which we consider as a big advantage. Also, machine learning plays a significant role in our project. Since we used supervised learning with evolving multi-class text classification as part of our project. We also worked with a transformer, specifically with RoBERTa, which is a Natural Language Processing (NLP) model since we have several topics of stories to categorize. After we fine-tuned the model, we calculated the accuracy using the formula we mentioned in the background. Additionally, we have found some datasets and an immensely helpful website containing e-books free of copyright.



SYSTEM REQUIREMENTS

— READ ME A STORY  —

4 System Design and Development

Overview:

In this chapter, we first discuss the methodology we used to develop the software. This chapter also explains the system requirements by including the system users, the requirement elicitation process, the user interactions with the software, and the application roadmap and backlog. In addition to covering the system design which includes the architectural diagram, class diagram, and component level design. Furthermore, the chapter describes our data design by displaying the data model and ER diagram, and by describing the data collection process. Then it goes into detail on the interface design by showing the application's structure using a sitemap and UX guidelines that were incorporated. Finally, the chapter explains the implementation steps of key and challenging parts of the code.

4.1 Methodology

We followed agile methodology to develop our application “Read me a story”. Agile is a project management methodology that emphasizes incremental and iterative project completion phases. Short-term development cycles are used to carry out a project's incremental components. Instead of top-down administration and adhering to a predetermined plan, the strategy places a higher priority on speedy delivery, flexibility, and cooperation[30].

Scrum is a framework that consists of roles, events, and artifacts. These elements work together to provide an agile project management methodology that helps teams manage their work better. As with anything in project management, the scrum methodology needs people to be executed. For this purpose, it defines three scrum roles, a scrum master, a product owner and a development team. The scrum master guarantees that everybody in the scrum team understands how the framework works and helps them adapt to the agile environment. The scrum product owner manages the product log, oversees sprint planning, and participates in scrum meetings. While The scrum development team is made up of all the team members who develop a software or product. For the scrum events, there are five events: sprint planning, daily scrum, sprint review, sprint retrospective, and the sprint itself. The events are meant to foster team collaboration and make sure that there's a constant line of communication among the scrum team members through the product or software development life cycle. Scrum artifacts help track the Scrum team's progress in any given scrum project. The three main

Scrum artifacts are the product backlog, sprint backlog, and product increment. The product backlog is the list of work that needs to be done which is ordered according to priority, the sprint backlog is the set of user stories in which the scrum team will be working in a single sprint, and the term product increment refers to all the product backlog items that have been completed during a sprint and it can also be used to describe the sum of all the completed backlog items and user stories.[31]

There are many benefits to working with agile methodology using the scrum framework, the benefit we feel benefited us most was the unparalleled flexibility this methodology provides. For example, whenever we needed to postpone or change a user story, it was easy to do so looking at how the project is planned, executed, and delivered incrementally. In addition, weekly meetings with our product owners helped us greatly in clarifying us on several topics as well as garnering feedback on our work.

We used Jira¹ to store our product backlog, plan our sprints in which we ordered our user stories based on their story points (in order to estimate how much effort is required for each feature), and also to write down our meeting notes with our supervisors. As for GitHub², we used it to create a repository in which we stored our application's source code. After we stored the source code it was easy to collaborate the code with team members where we can implement and remove changes via push and pull.

¹ Jira: <https://gp1-read-me.atlassian.net/jira/software/projects/GP/boards/1/backlog>

² GitHub: <https://github.com/reemaowerdi/GP1/tree/main>

4.2 System requirements

4.2.1 System Users

“Read me a story” application is aimed at users who are interested in teaching the child they are taking care of morals, or children who want to read or listen by themselves. The users of the application do not require any educational level, they do, however, need to read and understand the English language at an intermediate level at least to be able to read to the child different stories. In the case that a child was using the application, they only need to be able to understand English at an elementary level. Regarding technical expertise and experience, they only need to be familiar enough with IOS devices and IOS tablet applications to be able to use the application.

4.2.2 Requirements Elicitation and Analysis

In requirement elicitation, we have collected information by using a questionnaire that consists of important questions about the project's requirements and to know their opinions about applications that aims to provide stories for children based on morals. Also, from seeing some of our competitors, we noticed the lack of some important features, which inspired us to develop these in our app. The stakeholders who answered the questionnaire were a big group of end users who are interested in children's storytelling.

4.2.3 User Interactions

This use case diagram shows who will use the “Read me a story” application, as well as its main functionalities and a clear view of the steps the user will take to accomplish a specific task. "Admin" and "Reader" are the actors in Figure-14.

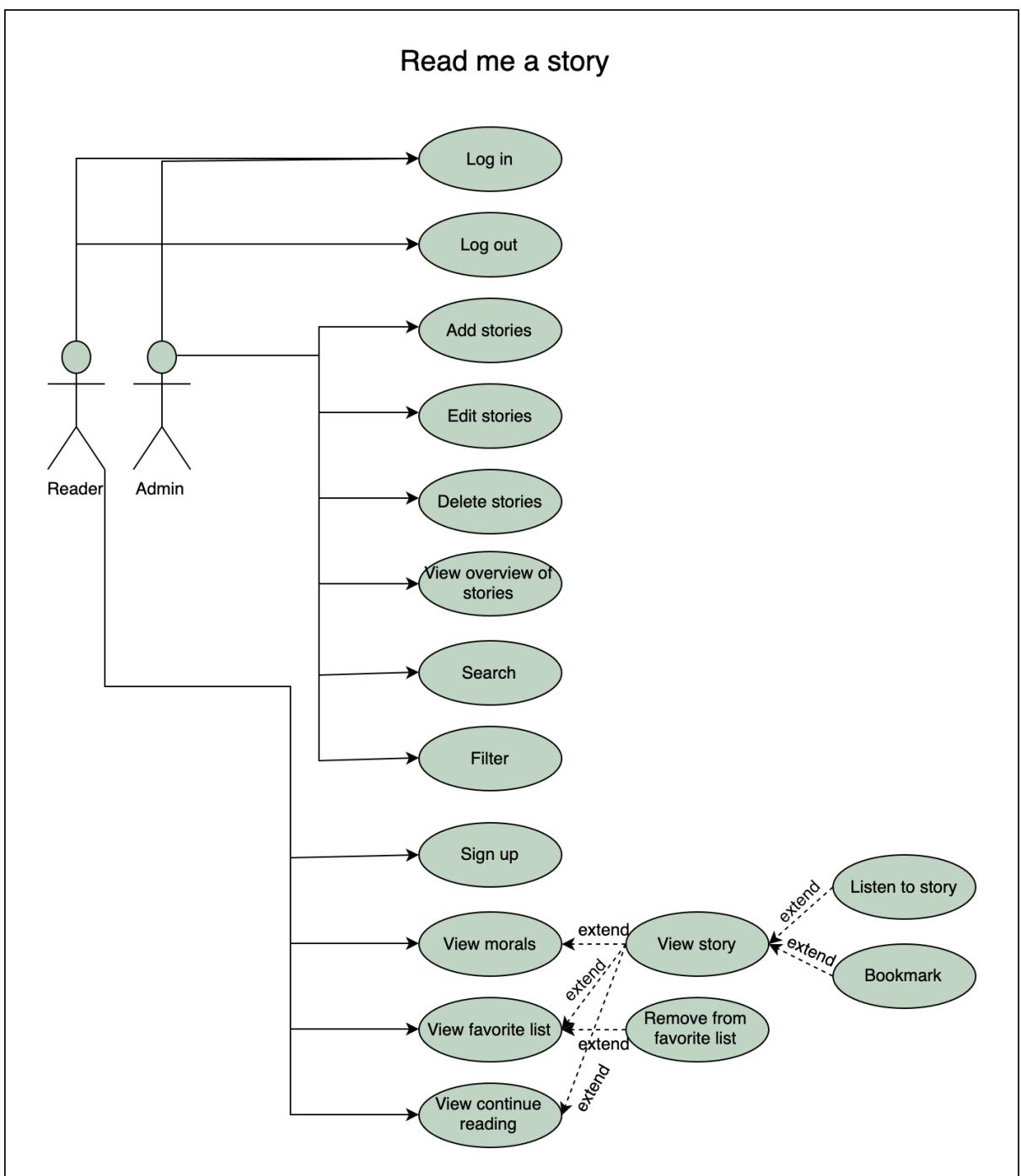


Figure 14-Use case diagram

4.2.4 Roadmap and Product Backlog

In this section, we will present the roadmap for the “Read me a story” application in Figure-15, a product roadmap is “a plan of action for how a product or solution will evolve over time[32]. In sprint 0, we performed domain analysis to provide the context for the information discussed in this report as well as provide research related to our project and comparisons to similar systems. Then in sprint 1, we began building our dataset from scratch, fine-tuning our classifier model in order for it to categorize a story to a specific moral, and preparing both the reader and admin interface so that the reader can view lists of books and morals and read the story, while the admin can add the story by providing its title, content, and moral. In sprint 2 we added features to both the admin and reader interface. Regarding the admin interface, we allowed the admin to be able to approve the moral after the classifier categorizes the story in case the admin thinks the moral classified is wrong, to log in and out of the website, and also to be able to add stories along with their respective picture to the database. The reader, however, was allowed to sign up, log in, and log out, as well as having the ability to add and delete stories to the ‘Favorites list’. In sprint 3, we made the reader be able to view the story with its picture along with read the story with a better layout which we described in detail in ([Splitting stories in the application](#)), and also added a bookmark feature so the reader can remember where they left off. While the admin can delete the story from the application. In sprint 4, we finished the edit story feature for the admin after it’s been carried over from sprint 3. Additionally, we added some minor but helpful features for the admin which are an overview of the stories, a search function, a filter for the stories, and a selection feature to be able to delete multiple stories at once. In this sprint we also improved the accuracy of the classifier model and enabled the reader to listen to the story via text-to-speech. Finally, in sprint 5, we added our final feature which is a ‘continue reading’ list on the homepage of the reader which contains all of the stories that the reader has previously bookmarked. We also did the hosting and the user acceptance test.

4.2.4.1 Roadmap

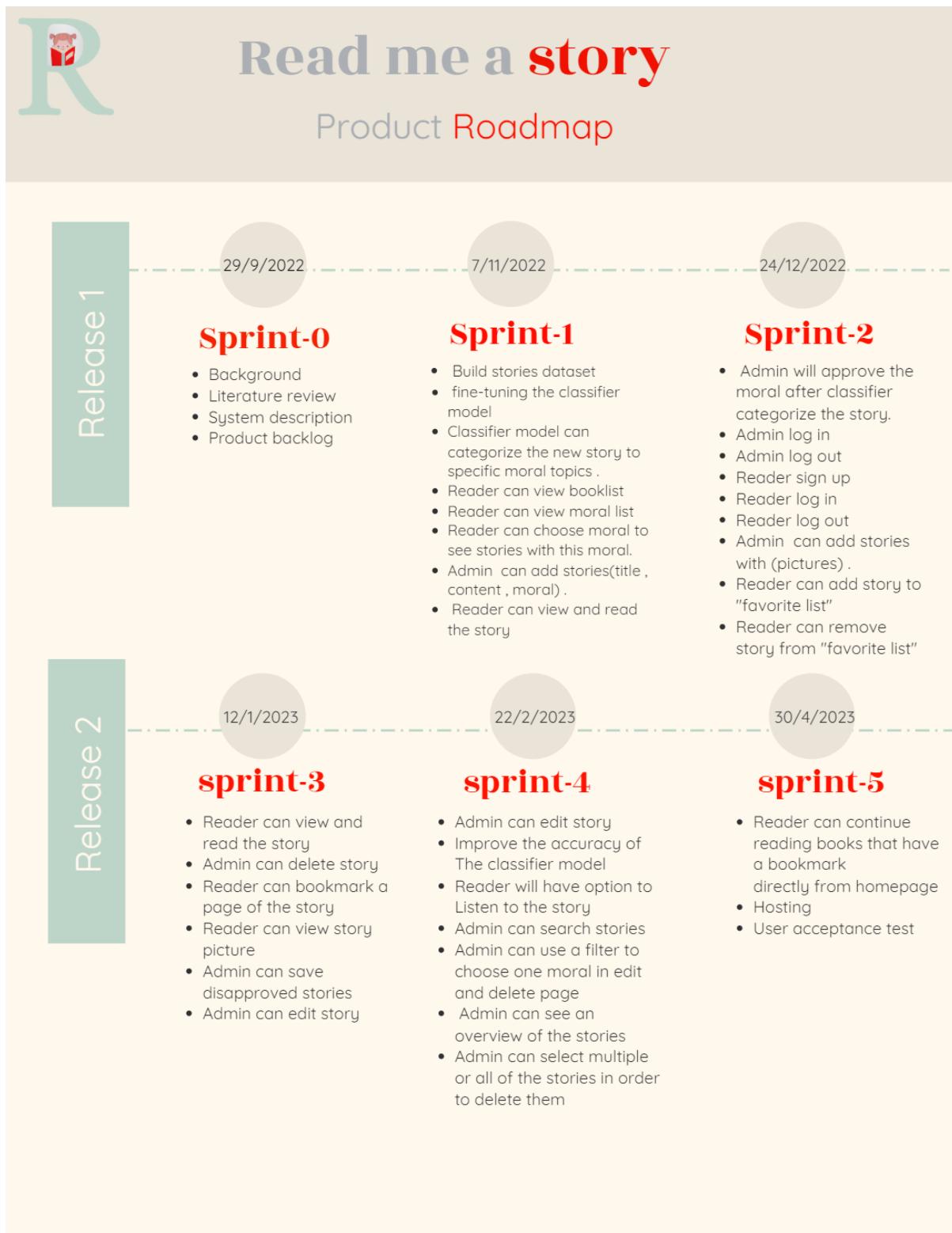


Figure 15-Roadmap

4.2.4.2 Product backlog

Table 2-Product Backlog

ID	PBI (user story)	Size (Story points)	Type (Feature, defect, technical work, knowledge acquisition)	Status (To do, progress, Done) in or	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
1	As an admin, I want to be able to add stories so that users can have more options of stories.	4	Feature	Done	<ul style="list-style-type: none"> As an admin, if I add a new story, then the message “added successfully” will appear and the story will be added to the database. As an admin, if I add a new story, then the story will go through the classification model and will be classified based on the most accurate moral.
2	When a story is added, the classifier model categorizes it to specific moral topics.	4	Technical	Done	If a story enters the classifier model, then the story will be classified to the appropriate moral.
3	When the classifier model categorizes the story to a specific moral, the moral of this story will appear directly in the admin interface.	3	Technical	Done	If the classifier classifies the story, then the admin can approve or reject it.
4	As a reader, I want to choose a moral from the morals list so that I can see stories under that moral.	1	Feature	Done	As a reader, if I click on a specific moral, then the system will redirect me to the stories that have been classified under that moral.
5	As a reader, I want to choose a story from the books list so that I can read it.	1	Feature	Done	As a reader, if I click on a specific story then the system will redirect me to the story that I chose.
6	As a reader, I want to have an audio option so that the child can listen to the story.	1	Feature	Done	<ul style="list-style-type: none"> As a reader, if I click on the listen button, then an audio

					of the content of the story chosen will start playing. <ul style="list-style-type: none"> As a reader, if I click on the pause button, then the story audio will stop playing.
7	As an admin, I want to be able to edit stories so that I can change what needs to be fixed.	3	Feature	Done	<ul style="list-style-type: none"> As an admin, if I click on the edit button, then I can have full access to change the story details. As an admin, if I click submit after editing the stories' information, then a confirmation message appears.
8	As an admin, I want to be able to delete stories so that unwanted stories disappear.	3	Feature	Done	<ul style="list-style-type: none"> As an admin, if I click on the delete button, then the story will be deleted and a confirmation message will appear. As an admin, if I click on the delete button after selecting multiple stories, then all of the selected stories will be deleted and a confirmation message will appear.
9	As a reader, I want to add a story to the favorite list so that it can become easier to access.	3	Feature	Done	As a reader, if I click on add to my (to favorite list) button, then the story will be added to my (to be favorite list).
10	As a reader, I want to access the favorite list so that I can view the stories I saved.	1	Feature	Done	As a reader, if I click on the favorite list button, then the list of stories I saved will appear.
11	As a reader, I want to remove stories from the favorite list so that I can delete the stories I no longer want.	2	Feature	Done	As a reader, if I unclick the favorite button, then the story will be deleted from the favorite list.
12	As a reader, I want to use the bookmark so that I can continue reading from where I stopped.	3	Feature	Done	As a reader, if I click on the bookmark icon then I can continue where I left off if I leave and return later.

13	As a reader, I want to sign up so that I can make my own account.	1	Feature	Done	As a reader, if I sign up successfully, then I will be redirected to the home page.
14	As an admin, I want to log in so that I can access my account.	1	Feature	Done	As an admin, if I log in successfully, then I will be redirected to the home page.
15	As an admin, I want to log out so that I can leave my account.	1	Feature	Done	As an admin, if I click on the log out button, then the system will redirect me to the log in page.
16	As a reader, I want to log in so that I can access my account.	1	Feature	Done	As a reader, if I log in successfully, then I will be redirected to the home page.
17	As a reader, I want to log out, so that I can leave my account.	1	Feature	Done	As a reader, if I click on the log out button, then the system will redirect me to the log in page.
18	As an admin, I want to disapprove the moral of the story so that I can save the story to the database to improve the model.	2	Feature	Done	As an admin, if I disapprove the moral of the story, then the message "story saved successfully" will appear and the story will be saved to the database.
19	As an admin, I want to be able to search when editing and deleting stories so that I can find them easily.	1	Feature	Done	As an admin, if I search the name of the story, then it will show immediately.
20	As an admin, I want to use the filter in the edit and delete pages so that stories with the same moral can be easily found.	1	Feature	Done	As an admin, if I choose one moral from the filter then it will display stories filtered under that moral.
21	As an admin, I want to see an overview of the stories so that I can know how many stories are in each moral and the total number of stories.	1	Feature	Done	As an admin, if I logged in, the system will redirect me to the home page then I can see the overview.

22	As a reader, I want to have a continue reading list on the home page so that I can find books that have bookmarks easily.	2	Feature	Done	As a reader, if I logged in, the system will redirect me to the home page then I can see the books in the continue reading list.
23	As a reader, I want to display the stories within one second when I choose the topic so that I can choose the story quickly [33].		NFR	Done	
24	As a reader, I want the story that I added to the favorite list to appear within one second so that I can view it in a short time [33].		NFR	Done	
25	As a reader I want the classification of the story topic to be accurate, 70% at least[34].		NFR	Done	
26	As a reader, I want the favorite list to be available offline at least 99% of the time, so that I would be able to read stories offline[35].		NFR	Done	
27	As a reader, I want to have a clear page layout with easy on the eye colors so that I will not be distracted. [43]		NFR	Done	
28	As a reader, I want to have commonly used UI icons so that dealing with the interface will be easy and obvious for me.		NFR	Done	
29	As a reader, I want the number of steps to get to the story to be as little as possible, so that I can reach the story quickly.		NFR	Done	

4.3 System design

4.3.1 Architectural diagram

In our “Read Me a Story” application we will apply the Model-View-Controller Architecture; we chose this architectural pattern because it is the most popular in-app architecture design. It is the one Apple recommends and it is easy to understand and use. In most code examples on the official Apple or Swift websites, applications are built using the MVC architecture. This is the default application architecture pattern used in all iOS, MacOS, and watchOS projects[36].

This architectural pattern consists of three levels: Model – this level is considered the lowest level when compared with the View and Controller. It primarily represents the data to the user and defines the storage of all the application’s data objects. Views – this level is majorly associated with the User Interface (UI), and it is used to provide the visual representation of the MVC model. In simpler terms, this level deals with displaying the actual output to the user. It also handles the communication between the user (inputs, requests, etc.) and the controller. Controller – this level takes care of the request handler. It is often considered as the brain of the MVC system- a link between the user and the system. The controller completes the cycle of taking the user output, converting it into desired messages, and passing them on to the views (UI). Using the MVC model holds many advantages, including being easily modifiable, having a faster development process, as well as having easy planning and maintenance[37].

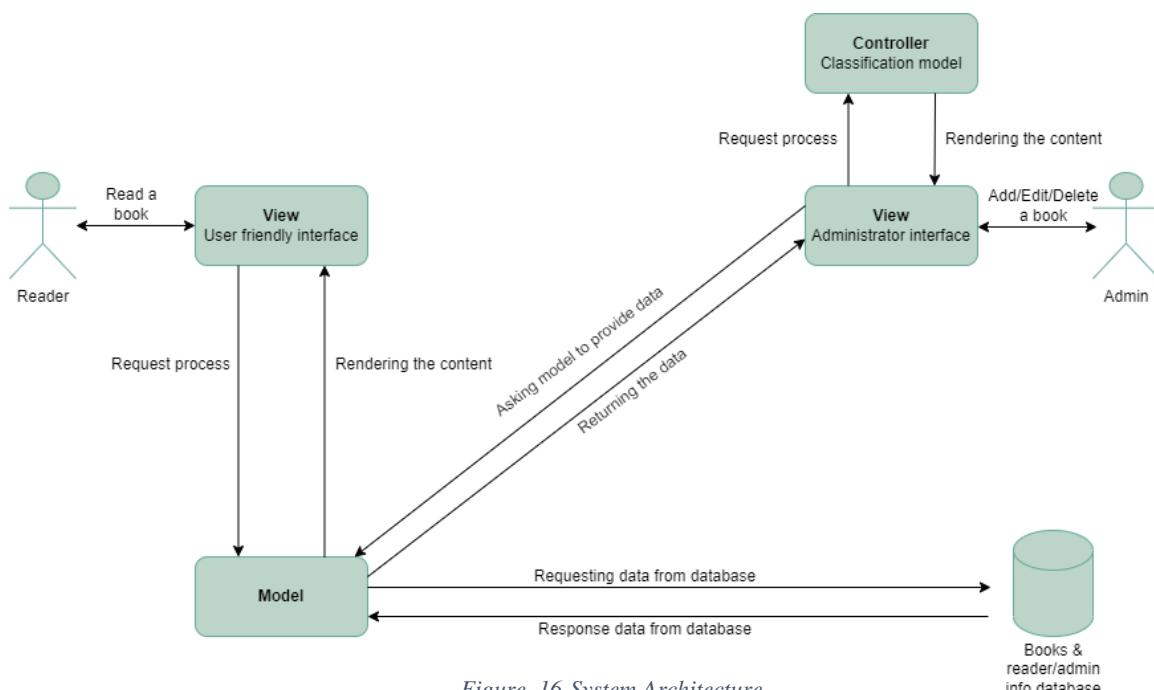


Figure 16-System Architecture

4.3.2 Class diagram

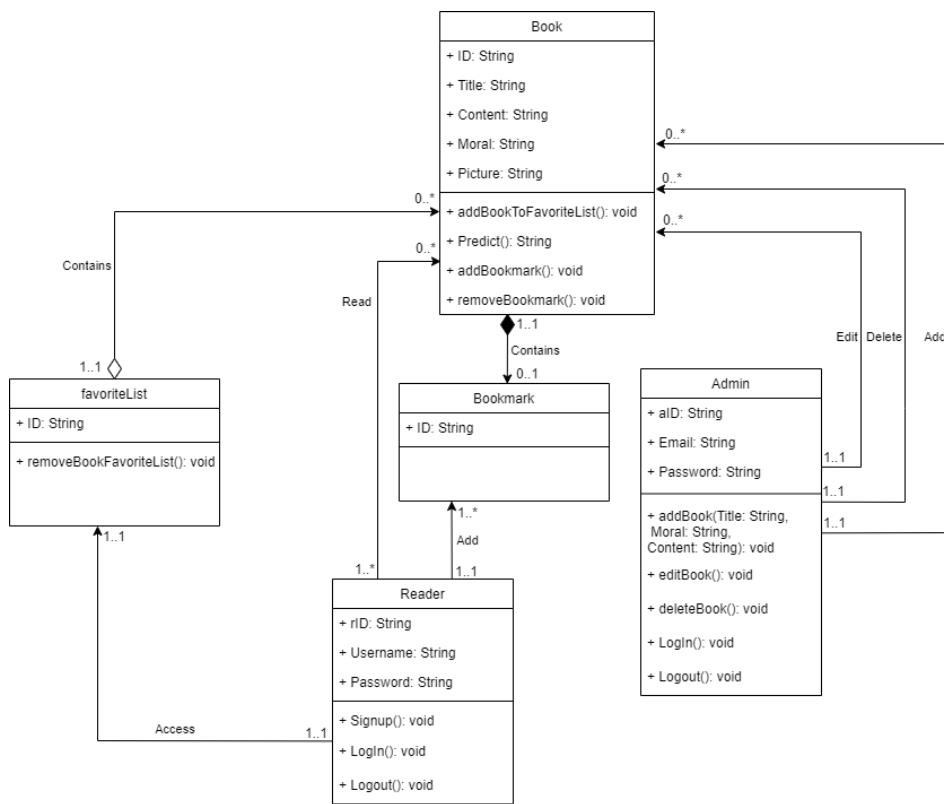


Figure 17-Class diagram

4.3.3 Component Level Design

1- Fine-tuning RoBERTa model on the stories dataset to predict the moral of the stories.

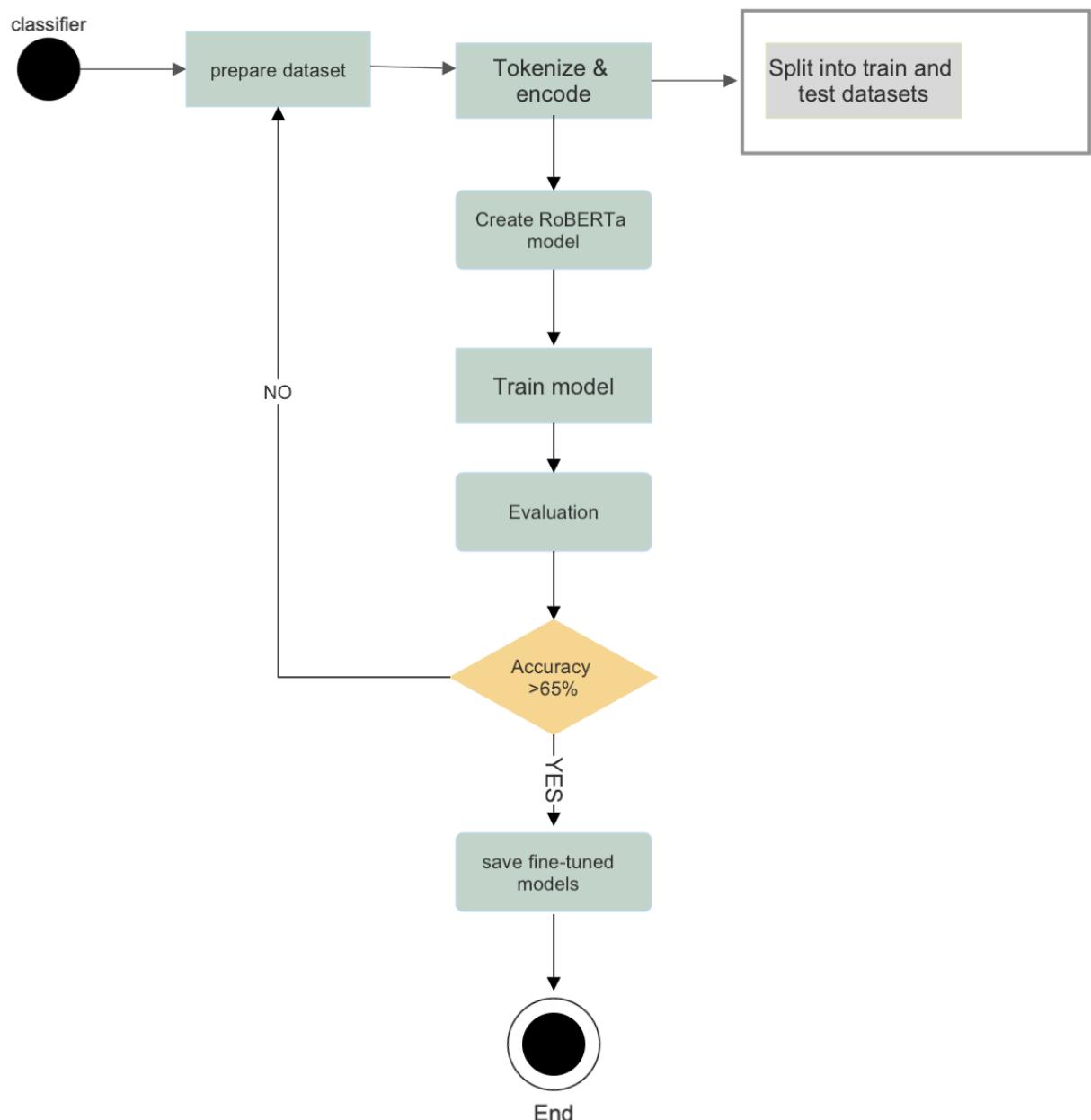


Figure 18-Component level design(fine-tuning RoBERTa)

- 2- As an admin, I want to be able to add stories so that users can have more options of stories.

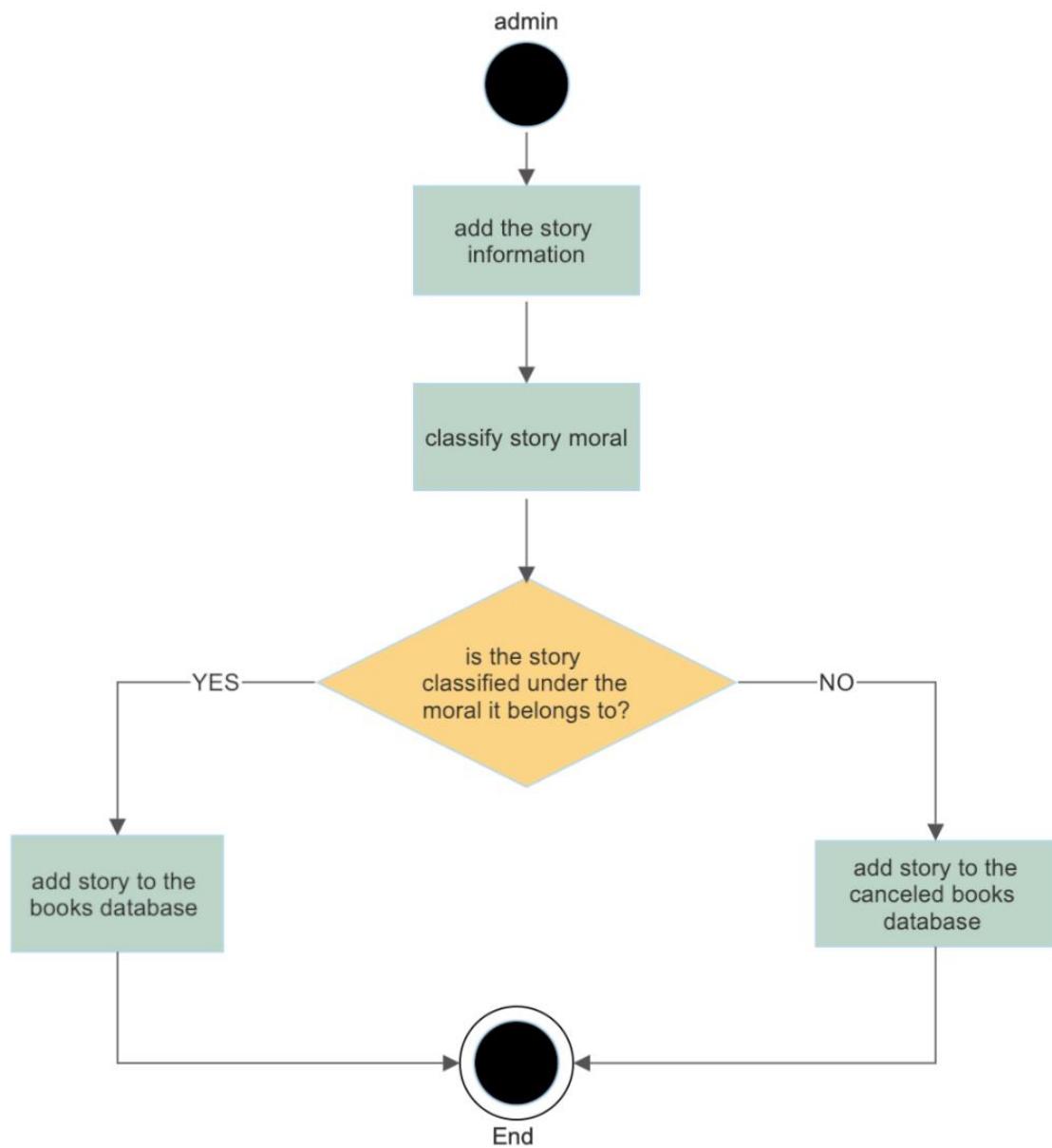


Figure 19-Component level design(Add story)

3- As a reader, I want to choose a story from the list of stories so that I can read it to the child.

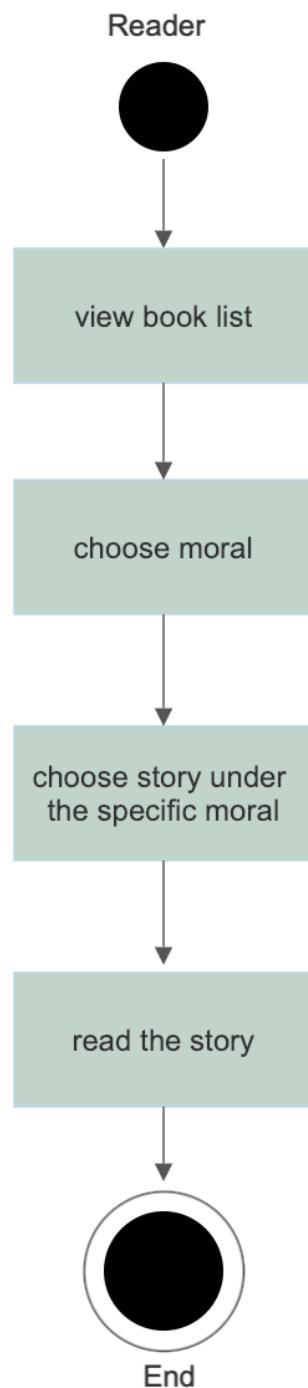


Figure 20-Component level design(Read story)

4.4 Data Design

4.4.1 Data Models

- **ER diagram**

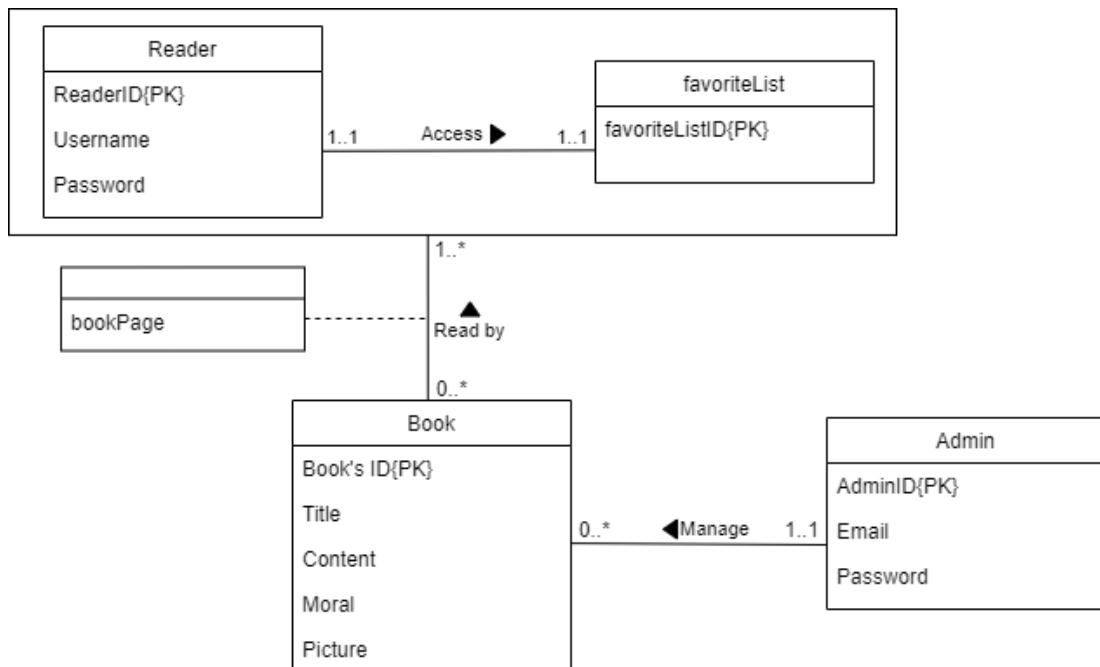


Figure 2I-ER diagram

- Data model

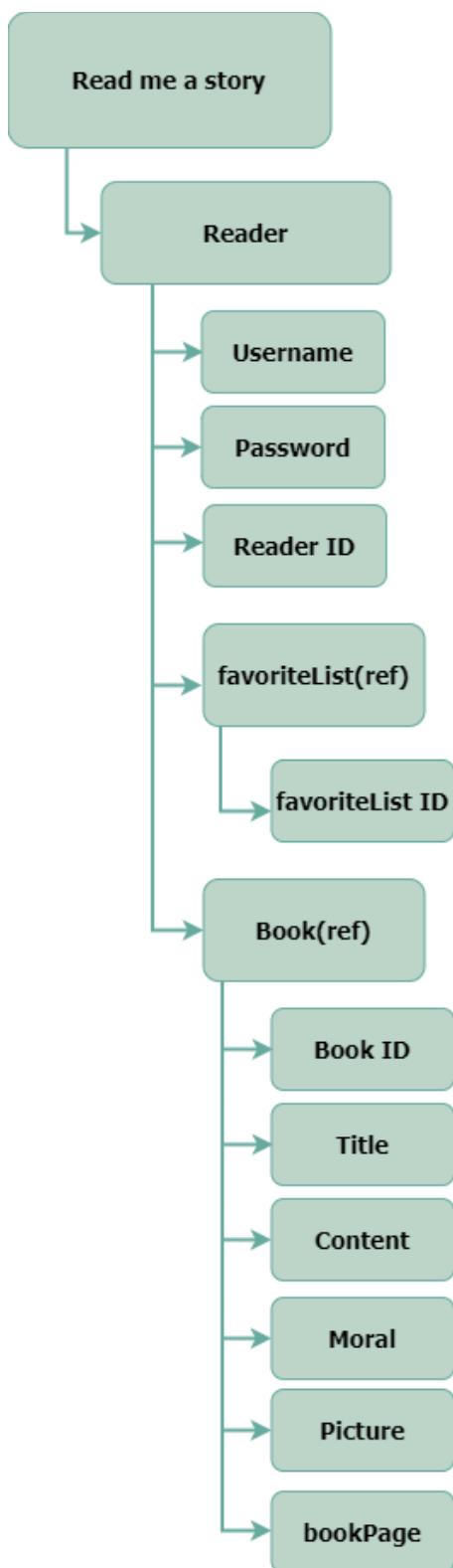


Figure 22-NoSQL data model

Figure-22 shows the structure that represents the system's data model which is made up of the Reader collection.

Firestore is a “flexible, scalable NoSQL cloud database to store and sync data. It keeps your data in sync across client apps through real-time listeners and offers offline support so you can build responsive apps that work regardless of network latency or Internet connectivity”.[38]

We have found that this database system best suits our application’s needs, and since it is a non-relational database, we found that it is important to better understand the connections between the entities in the ER diagram by representing the data structure as a tree.

First, “Read me a story” is the parent node to “Reader”. Which is a collection and child node to “Read me a story”. “Reader” is the parent node to two other referenced collections which are “favoriteList” and “Book”. Second, the documents are “Read me a story”’s grandchildren. Each document contains a set of key-value pairs.

4.4.2 Data Collection and Preparation and Analysis

4.4.2.1 Data Collection

First, we looked for datasets that contained the story's contents and its morals. After spending a substantial amount of time looking for datasets in reports and research, we stumbled upon research that contained how to classify stories based on a few morals. After looking through the paper though, we, unfortunately, did not find a classified dataset. We tried contacting the author of the paper for it[39], but they did not respond. Therefore, we then concluded that we had to prepare the dataset from scratch.

Our classification method was based on this paper: (A Corpus for Understanding and Generating Moral Stories)[39]. In this research, we found a clear method for collecting stories, classifying them, and how to manually annotate stories. We explained the method in further detail in the [Manual Annotation](#) section. We also found several sources containing English-language stories that are free of copyright.

We collected the stories from different websites free of copyright such as Gutenberg[40] Stories were gathered in two stages, 79 in the initial stage and 61 in the model improvement stage with a total of (140) English raw text stories, we show the full list of these websites in [Appendix A](#).

We took about 8 stories from the site Liz Story Planet, 24 stories from Gutenberg, 2 stories from Library of Congress, 30 stories from Free Kids Books, 47 stories from MoralStories26, 12 stories from English for Students, and 17 stories from taleswithmorals.

4.4.2.2 Manual Annotation

In this phase we performed manual annotation. During this phase, we observed that there are no indicators such as "This story is about honesty" to detect the story's moral in most raw texts. So, in this step, we made 3 annotators read the content of the story and judge whether it violates general ethical principles and whether they contain a moral. If it doesn't violate ethical principles and it does contain a moral, we then check if 2 annotators at least agree on one moral. If they did agree on the moral, then we add the story to our dataset. The manual annotation tasks are summarized as followed in Figure-23:

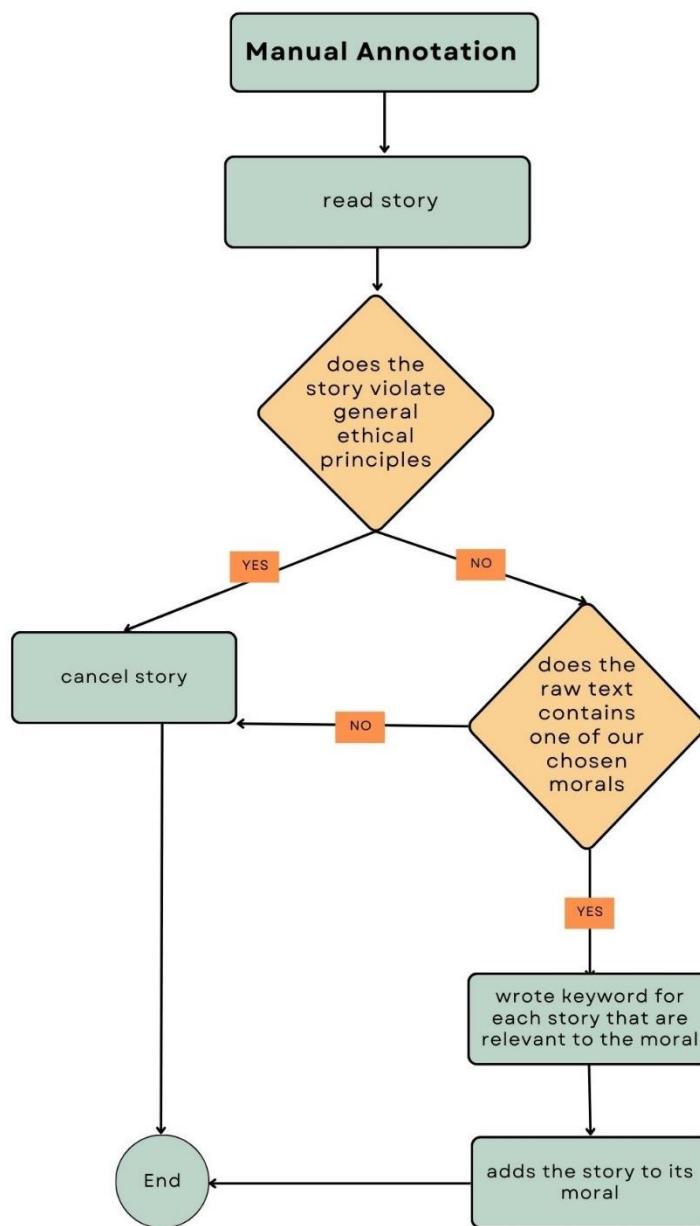


Figure 23-Manual Annotation Flowchart

Keywords:

During the manual annotation phase, we found that there were some keywords that are uncommon between the same moral and are not straight forward, such as curiosity which falls under patience. This made us focus more on the stories' keywords and meanings that refer to them. Also, we noticed that keywords can be synonymous or antonymous as shown in Table-3:

Table 3-Keywords

Moral	Keywords
Patience	wait, curious, impatient, impatiently, waiting, persevering, anxious
Honesty	honesty, lied, truth, liar, trick, truthful, cheat, lying, honest
Bravery	brave, courage, danger, afraid, courageously, determined, scary, fearless, trouble, fear, heroism, help, confront
Respect	respect, obey, rebellious, naughty, argue, rude, respectful, yell
Friendship	friend, love, lonely, play, family, care, alone

Table 4-Keywords examples

Moral	Example	Keywords
Patience	Charlie had no <u>patience</u> , he didn't like what the teacher was saying to him, and did not <u>wait</u> until the teacher finish the class.	<u>patience</u> , <u>wait</u>
Honesty	who lived a very simple life. He was poor but he was a good man he never <u>lied</u> he always told the <u>truth</u> .	<u>lied</u> , <u>truth</u>
Bravery	The whole kingdom fought until the shy brother had the <u>courage</u> to beat the evil forces and save the kingdom.	<u>courage</u>
Respect	After her speech, all the children realized that they had been <u>rude</u> to Anna, and felt sorry.	<u>rude</u>
Friendship	Nita feels the right way round, thanks to the <u>new friend</u> she's found. Upside down was never fun. Now she <u>plays</u> with him and everyone.	<u>new friend</u> , <u>plays</u>

4.4.2.3 Preprocessing

We used preprocessing on our data for two reasons:

- To provide us with an overview of the dataset, we need to know if the data is balanced or not.
- To prepare the data to be used in fine-tuning the RoBERTa model.

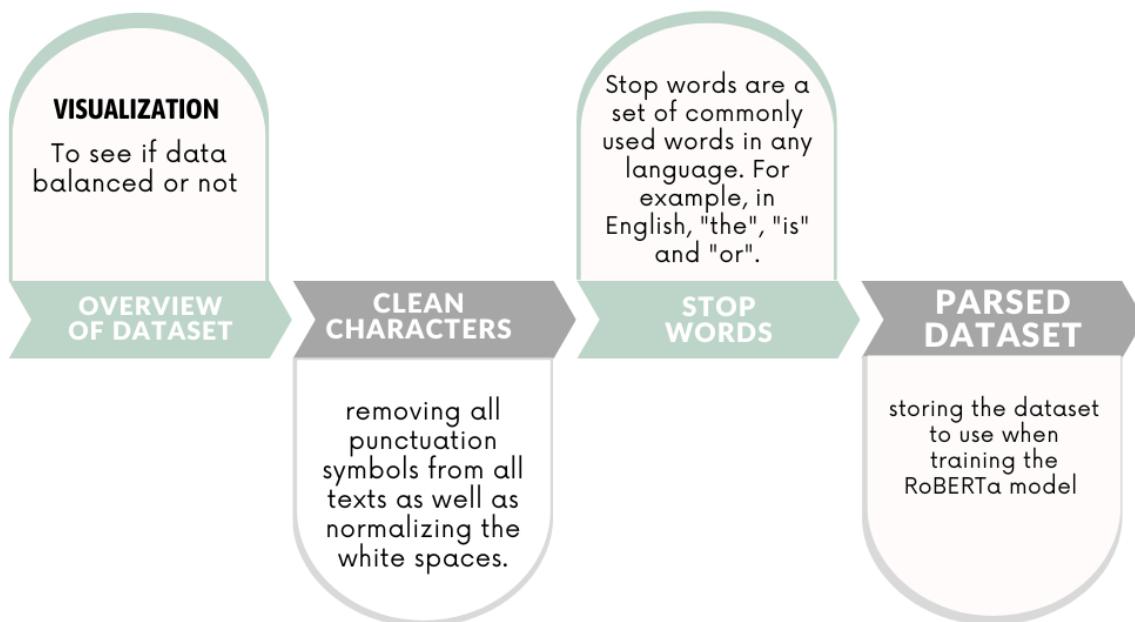


Figure 24-Preprocessing steps

After preprocessing we found that our data was balanced so we have left it as it is. Unbalanced data appears when there is a label that contains more stories than others, for example, if the honesty class is very few compared to the respect class, here we will have to apply techniques to balance it and that could cause problems during the training. Also, we cleaned the data and removed most of the unnecessary words to use when training the model so that the accuracy became higher.

4.4.2.4 Data analysis

For our data analysis, we will present the distribution of the stories across different morals, identify the most commonly occurring keywords for each moral, and examine if there has been any change in the number of words after data cleaning and what are these words that have been deleted.

We have 141 stories. 30 friendship, 29 honesty, 26 brave, 25 patience, and 30 respect as shown in Figure-25.

To know if our data is balanced, we calculated it using the ratios equation [41]. We divided each class with the total number of stories, 140, then we simplified the ratios by dividing each of them by the smallest ratio then multiply all of them by 10 to get rid of the decimal points. Now, the ratios are 12:12:12:10:10. Then, we simplified them to be 6:6:6:5:5, the differences between them are small so it is balanced.

As a result, we were able to determine that our data was balanced.

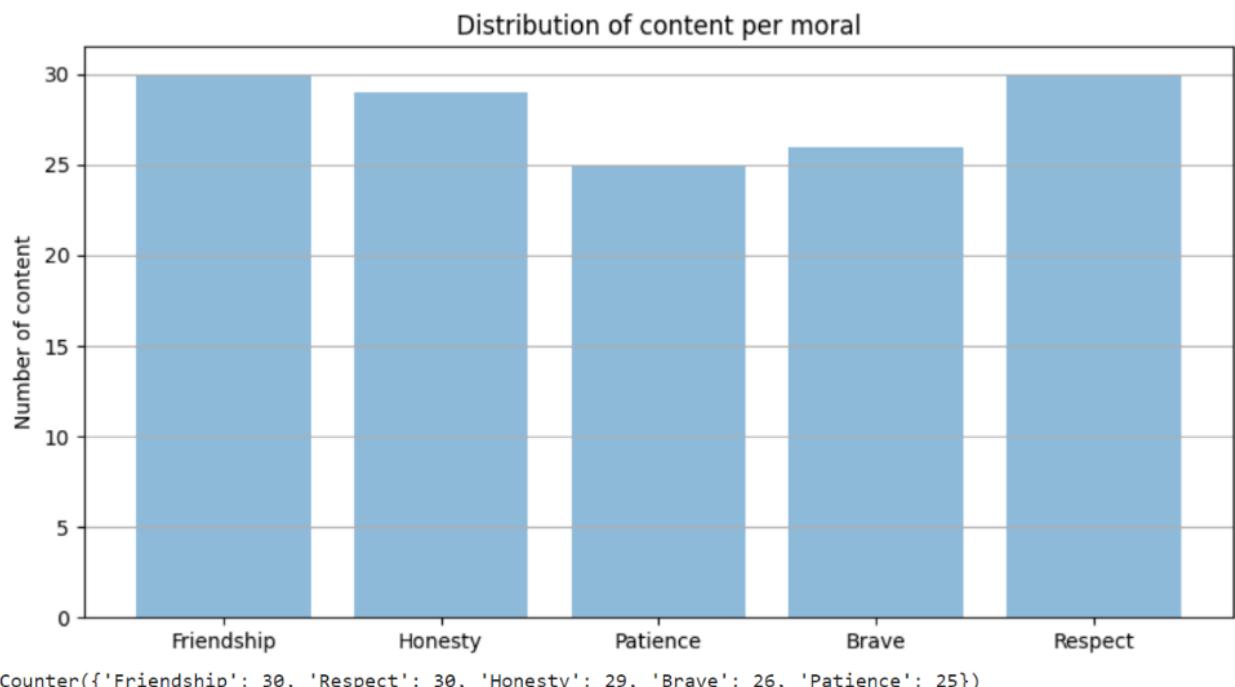


Figure 25-distrbution of story across morals

We selected keywords from each moral to determine their frequency in the stories and identify the least common and most common words as shown in Figure-26.

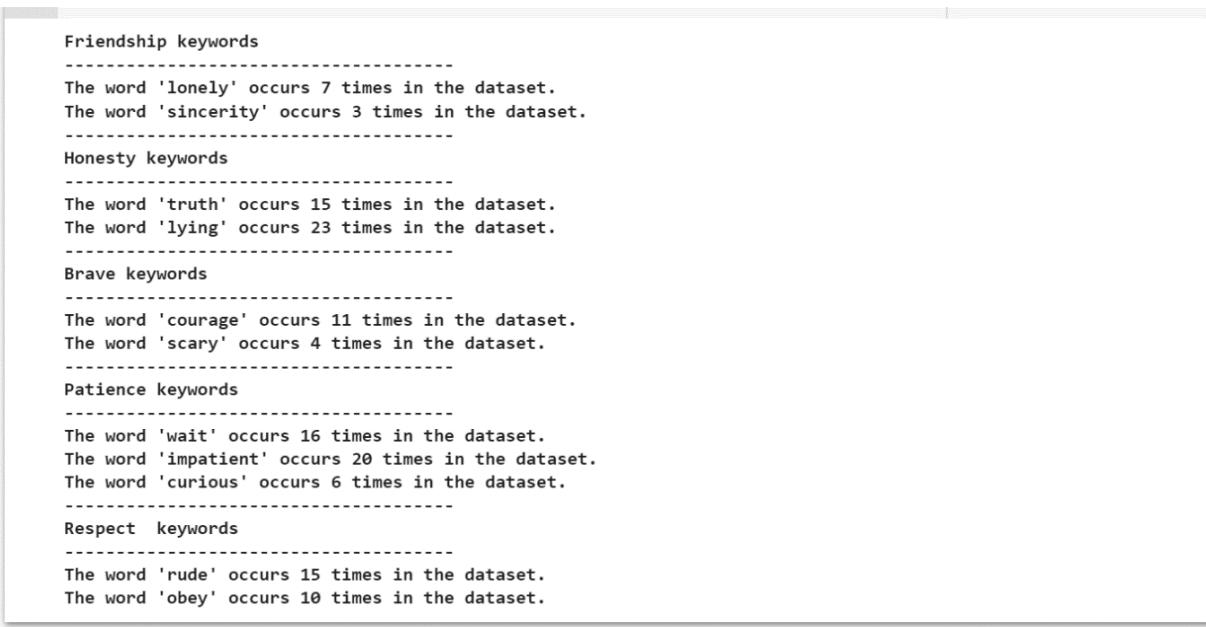


Figure 26-frequency of keywords

For cleaning the data, we utilized a library called spacy to eliminate stop words from the data. Stop words are a set of commonly used words in any language. For example, in English, "the", "is" and "or", would easily qualify as stop words. These words are not semantically useful in the sentences; thus, it is a good idea to remove them and focus on important words instead. Additionally, we observed that some words in the stories were repetitive, but they did not belong to any moral and their repetition could affect the accuracy of the classification such as king, princes, and animals as shown in Figure-27.

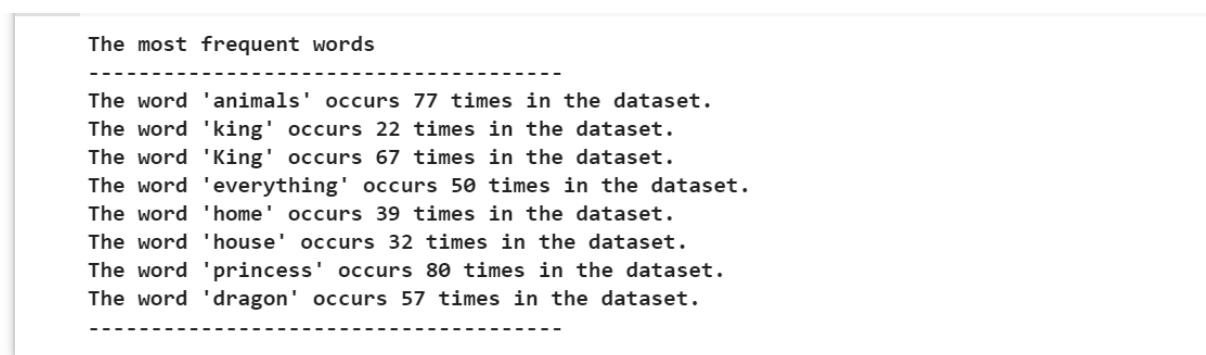


Figure 27-unwanted words

Figure 28-29 demonstrates the impact of removing stop words and other frequently repeated words that did not belong to any moral. By comparing the length of sentences before and after the cleaning process, it was observed that there was a noticeable difference in length after the removal of such words. This implies that the remaining words after cleaning were more pertinent to the storyline.

Lengths, mean, and standard deviation before cleaning in Figure-28:

		lengths	mean	stdev	
Friendship	[186, 324, 211, 119, 559, 223, 364, 476, 569, ...]	460.294118	373.937790		
Honesty	[172, 240, 330, 494, 436, 804, 192, 591, 370, ...]	397.750000	173.590130		
Patience	[407, 534, 262, 595, 261, 220, 144, 365, 764, ...]	394.600000	359.435629		
Brave	[1165, 801, 205, 511, 500, 498, 776, 319, 631, ...]	530.937500	266.077800		
Respect	[231, 364, 445, 609, 227, 480, 222, 417, 392, ...]	311.600000	139.747630		

Figure 28- Lengths, mean, and standard deviation before cleaning

Lengths, mean, and standard deviation after cleaning in Figure-29:

		lengths	mean	stdev	
Friendship	[73, 201, 105, 49, 369, 120, 157, 211, 351, 16...]	228.117647	183.912983		
Honesty	[82, 121, 136, 243, 185, 320, 73, 249, 140, 93...]	170.312500	73.653892		
Patience	[145, 236, 140, 258, 97, 88, 54, 162, 334, 711...]	168.666667	172.990366		
Brave	[528, 361, 137, 222, 223, 213, 405, 163, 326, ...]	237.437500	123.761043		
Respect	[96, 141, 199, 235, 89, 175, 88, 178, 149, 77,...]	123.933333	56.468027		

Figure 29-Lengths, mean, and standard deviation after cleaning

4.5 Interface Design

Many successful companies, including Apple, Google, and Microsoft, used UX guidelines for interface design. The Read me a story application is built following Schneiderman's eight golden rules, and Nielsen's heuristics as they aid in the design of a user-friendly application and improve its usability and accessibility. And for the proper text font and size for children reading tablets we used 2 research papers[44][45].

UX guidelines:

- Icon Usability:** “a user’s understanding of an icon is based on previous experience.[39]”. Use either an icon or a title to explain what a button does especially when children use your application. For example, in our application we use the favorite icon which is a heart icon. If they click it, the story will be saved in their list. Also, we used a female and male icon. If they click it, they can listen to the story with either a male or female audio as shown in Figure-30.

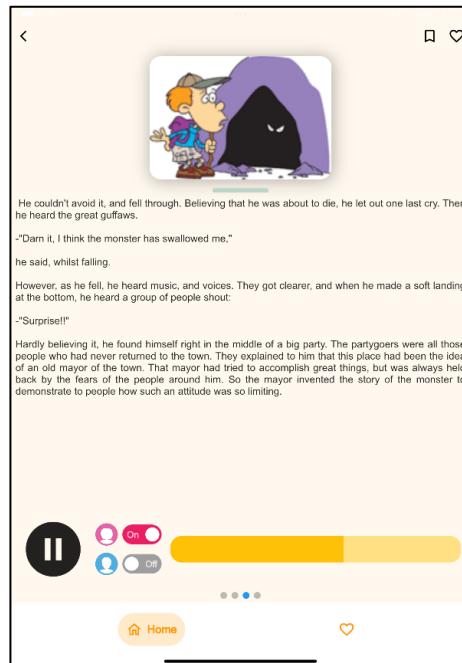


Figure 30-Icon Usability

2- Reduce short-term memory load: reduce short term memory load by designing screens where options are clearly visible. As Nielsen says, “recognizing something is easier than remembering it. Minimize the user's memory load by making objects, actions, and options available”[42]. We apply this in the booklist where each moral is clearly displayed as shown in Figure-31. If the user chooses a specific moral and goes to the booklist page for this moral, the name of the moral will appear there. So, it will be easy for the user to remember which moral they chose.

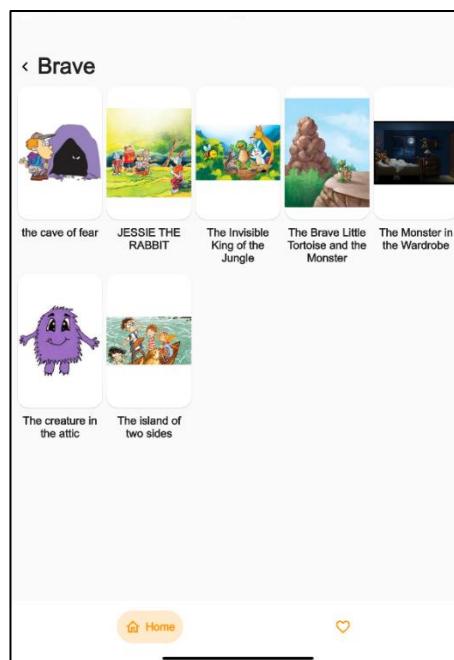


Figure 31-Reduce short-term memory load

3- Consistency and standards [43]: Users should not have to wonder whether different words, situations, or actions mean the same. We apply this by placing the back icon on the bottom left of the form and also the search bar on the top left on the admins interface. We also made sure in the application that all morals on the pages looked the same and were easy to find. The buttons for adding to favorites, controlling audio, and logging out were always in the same spot, so it was easy to use.



- 4- Aesthetic and minimalist design [43]:** Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility. We apply this by having a “See more” button on the admins interface as shown in Figure-32 and Figure-33.

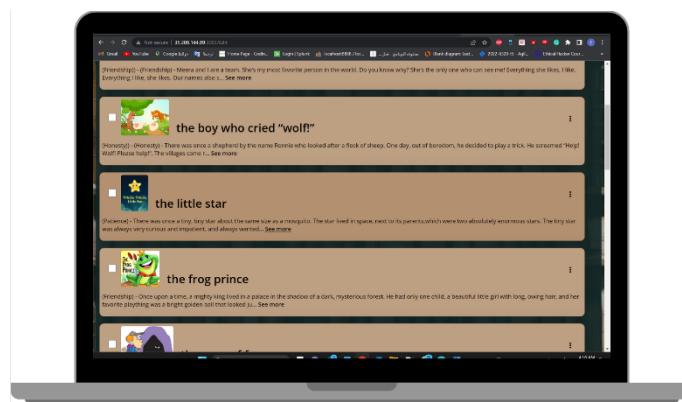


Figure 33- “See more” button

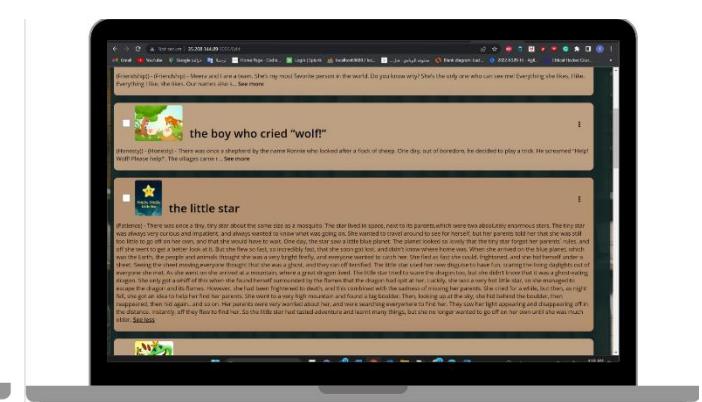


Figure 32- see less button

- 5- User control and freedom[43]:** Users often perform actions by mistake. They need to be able to back out of an unwanted action without having to go through an extended process. We apply this by allowing the user to undo their unintentional clicks. For example, removing a bookmark or a story from favorites list once they add to it.
- 6- Text font and size[44]:** We have conducted research about the most proper text font and size for children reading on tablets, we found that bigger sizes for the text is not always better, and we concluded that the best font and size to use are 14-point size and comic sans MS[45].

4.5.1 Sitemap

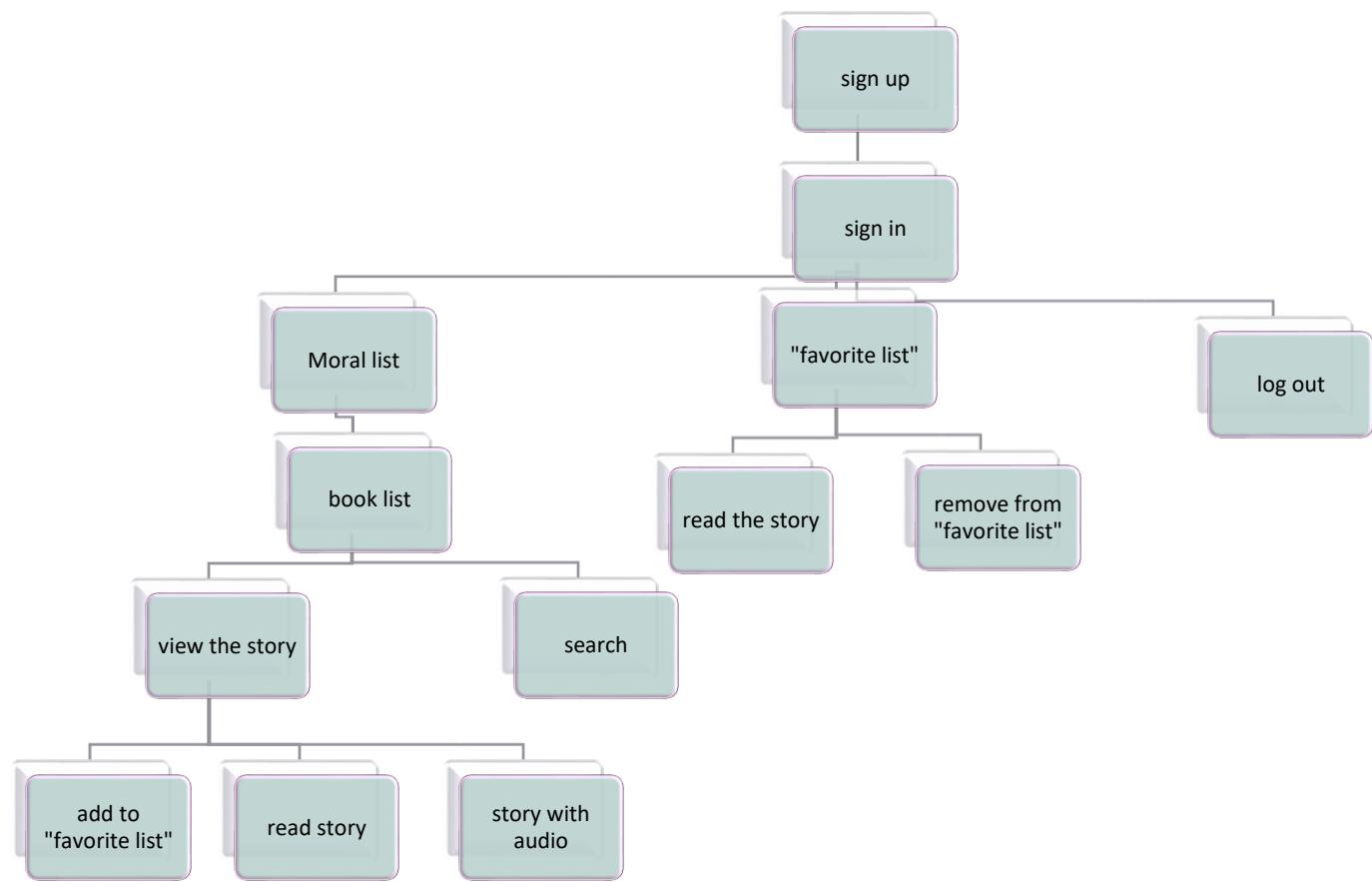


Figure 34-Sitemap

SYSTEM IMPLEMENTATION

— READ ME A STORY  —

4.6 System Implementation

In this section, we discuss the major and challenging aspects of our code. Additionally, we outline the key implementation procedures and steps. Our team encountered several challenges starting from fine-tuning the RoBERTa model to programming the most important features of the application and the website.

4.6.1 RoBERTa model

We tried both TensorFlow and PyTorch. Both of them worked fine with training but when writing prediction code, the resources on PyTorch were more available than TensorFlow. Because of that, we decided to write the RoBERTa code using a PyTorch library from hugging face. Hugging Face is a "community and data science platform that provides tools that enable users to build, train and deploy ML models"[46].

To begin, we needed to install and import these libraries in Figure-35 and Figure-36:

```
installation
[ ] !pip install transformers
```

Figure 35-installation

```
import
[ ] import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, precision_recall_fscore_support
import torch
from transformers import TrainingArguments, Trainer
from transformers import AutoModelForSequenceClassification, AutoTokenizer, BertTokenizer, BertForSequenceClassification
from transformers import EarlyStoppingCallback

from collections import Counter
import matplotlib.pyplot as plt
import regex as re
import statistics
import math
```

Figure 36-import

Prepare dataset:

We had to start building our dataset from scratch and manually annotate each story, which we also discussed this issue in detail on the ([Problems and challenges encountered during the software development](#)) section.

By using this code, we allow Google Collab access to Google Drive so it can read the dataset, in Figure-37:

▼ Prepare dataset

```

23s ❶ # Mounting Google Drive
      from google.colab import drive # to mount Google Drive to Colab notebook
      drive.mount('/content/gdrive')
      path='./gdrive/My Drive/Colab Notebooks/'

      ▷ Mounted at /content/gdrive

0s [4] df = pd.read_csv(path+'Children-Books(141).csv')
      df.head()

      Title          Author        Moral      Content
      0   Meera and Ameera  Nimmy Chacko  Friendship  Meera and I are a team. She's my most favourit...
      1   Every Minute Counts  Shabnam Minwalla  Friendship  Adil is always late to school.\nHis class teac...
      2   Mali's Friend       Rujeko Moyo  Friendship  Mali enjoys playing by himself.\nHe likes to b...
      3 My Best Friend – short story with imagination  Anupa  Friendship  I have a friend. She lives in my house.\n\nWhe...
      4 Same-Same or Different? Being friends despite ...  Roopa Pai  Friendship  Mamma put Sparrow's lunchbox into her schoolba...

```

Figure 37-read dataset

Tokenize & encode:

- 1- Define pretrained tokenizer and model.
- 2- We imported tokenizer from HuggingFace to use it here.
- 3- Assigning x to content and y to moral.
- 4- Split into train and test datasets.
- 5- RoBERTa's input accepts up-to 512 tokens, thus we must truncate the tokenized texts.

In our case, we truncate to 512 tokens because our story contains a lot of lines as shown in Figure-38.

```

[58]
# Define pretrained tokenizer and model
model_name = "roberta-base"
# Import tokenizer from HuggingFace
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=5)
# device = torch.device("cuda") if torch.cuda.is_available() else torch.device("cpu")
# model.to(device)
# print(f'Using {device}')
# ----- 1. Preprocess data -----#
# Preprocess data
X = list(df["Content"])
y = list(df["Moral"])
# Split into train and test datasets
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)
X_train_tokenized = tokenizer(X_train, padding=True, truncation=True, max_length=512)
X_val_tokenized = tokenizer(X_val, padding=True, truncation=True, max_length=512)

```

Figure 38-Tokenize & encode code

Create RoBERTa model:

- 1- Create torch dataset.
- 2- Fine-tune pretrained model.

Train model:

- 1- Define Trainer parameters.
- 2- Define Trainer (batch size, epochs) as shown in Figure-39.

```
# Define Trainer
args = TrainingArguments(
    output_dir= output_dir,
    evaluation_strategy="steps",
    eval_steps=15,
    logging_steps =15 ,
    save_steps=15,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=10,
    num_train_epochs=6,
    seed=0,
    load_best_model_at_end=True,
)

trainer = Trainer(
    model=model,
    args=args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=3)],
)
```

Figure 39-Train the model code

Evaluation:

Import accuracy score, precision, recall, and F-score support to print them in our code as shown in Figure-40.

```
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, precision_recall_fscore_support
```

Figure 40-Evaluation libraries

- **Save the model:**

After our model achieved the required accuracy which is 91%, here we needed to save the weight and vocab in an external folder so we can be able to use it in the predict code. The challenge here is that most of the RoBERTa codes we found did not need to save their model in an external folder because they predict the model in the same training code or they just need to see the accuracy and compare it to another model, so that took us a long time to figure out how to save it.

We saved the model using this code in Figure-41:

▼ save model

```
[ ] classifier_directory = "./drive/My Drive/Colab Notebooks/classifier_directory2"
tokenizer.save_pretrained(classifier_directory)
model.save_pretrained(classifier_directory)

tokenizer config file saved in ./drive/My Drive/Colab Notebooks/classifier_directory2/tokenizer_config.json
Special tokens file saved in ./drive/My Drive/Colab Notebooks/classifier_directory2/special_tokens_map.json
Configuration saved in ./drive/My Drive/Colab Notebooks/classifier_directory2/config.json
Model weights saved in ./drive/My Drive/Colab Notebooks/classifier_directory2/pytorch_model.bin
```

Figure 41-save model

- **Predict code:**

This code defines a function called prediction() that makes a prediction based on a text input using a pre-trained language model.

The function starts by printing the text input from the front end. It then uses the Hugging Face library to load a pre-trained tokenizer for the RoBERTa model. The text input is then tokenized using the tokenizer. Next, the tokenized input is used to create a PyTorch dataset for the input. The pre-trained model is loaded from a specified directory, with the number of labels set to 5.

The np.argmax() function is used to find the index of the highest prediction score, which corresponds to the predicted label. The predicted label is then converted into a string label ('Friendship', 'Honesty', 'Patience', 'Brave', or 'Respect') and returned by the function as shown in Figure-42.

```
259     # prediction
260     tokenizer = AutoTokenizer.from_pretrained("roberta-base")
261     X_test = [request.form.get('content')]
262     #X_test=['honesty truth lie honest']
263     print(X_test)
264     X_test_tokenized = tokenizer(
265         X_test, padding=True, truncation=True, max_length=512)
266     print(X_test_tokenized)
267     # Create torch dataset
268     test_dataset = Dataset(X_test_tokenized)
269     # Load trained model
270     model_path = "./classifier_directory"
271     model = AutoModelForSequenceClassification.from_pretrained(
272         model_path, num_labels=5)
273
274     # Define test trainer
275     test_trainer = Trainer(model)
276     # Make prediction
277     raw_pred, _, _ = test_trainer.predict(test_dataset)
278     # Preprocess raw predictions
279     y_pred = np.argmax(raw_pred, axis=1)
280     pred = ''
281     if y_pred == 0:
282         pred = 'Friendship'
283     elif y_pred == 1:
284         pred = 'Honesty'
285     elif y_pred == 2:
286         pred = 'Patience'
287     elif y_pred == 3:
288         pred = 'Brave'
289     elif y_pred == 4:
290         pred = 'Respect'
291     return pred
292
```

Figure 42-predict code

4.6.2 Prepare Flutter environment

After our comprehensive research, we concluded that Flutter which is a Google open-source framework for creating, natively built, multi-platform applications[47],and Dart which is a “client-optimized language for developing fast apps on any platform”[48], are the most suitable tools for us. Since they have multiple resources and tutorials online and they work on both macOS and Windows so the whole team will be able to work on the code.

- **Visual studio code**

To start coding and implementing our application we had to download a tool, which was Visual Studio Code. Visual Studio Code is a Microsoft free open-source text editor. Linux, macOS, and Windows all support VS Code.[4]

- **Xcode**

As for macOS users, we had to download Xcode environment which is: “The integrated development environment (IDE) from Apple that is used to create, compile and test macOS and iOS (iPhone/iPad/iPod) applications” [50].

- **Flutter coding preparation**

After setting up flutter we finally started working on the code. We encountered a lot of errors while working on it and problems related to software version issues, that we talked about in detail on the ([Problems and challenges encountered during the software development](#)) section, we searched about every error we faced, and Stack overflow[51] was the most useful tool that we used in solving errors. And the front-end was finally ready!

4.6.3 Firebase and its connection with Flutter and admin interface

4.6.3.1 Firebase connection to Flutter

We used firebase which is “an app development platform that helps you build and grow apps”[52] to store our dataset. After that we needed to connect it to our flutter code to retrieve the data. To do that, we first connected the firebase to flutter, after we connected the firebase to flutter, we then retrieved the stories.

4.6.3.2 Firebase connection to Admin interface

We had to use python language to implement the interface, so we can connect it with the classifier model which also uses python. First, we had to connect it to firebase by using the firebase_admin library and URL as shown Figure-43:

```

14 import firebase_admin
15 from firebase_admin import db
16 from firebase_admin import credentials
17 from flask import Flask, request, url_for ,flash , jsonify
18 from firebase_admin import firestore
19 import pyrebase
20 from datetime import timedelta
21 from flask import session, app
22 import time
23
24 # access database
25 databaseURL = 'https://read-me-a-story3-default.firebaseio.com/'
26
27 cred_object = credentials.Certificate("./config.firebaseio.json")
28 default_app = firebase_admin.initialize_app(cred_object, {
29     'databaseURL': databaseURL,
30 })
31

```

Figure 43-admin library

Then import the Flask class as shown in Figure-44 “Flask is a light-weight, modular, server-side Python framework that allows you to develop web applications”[53]:

```

31
32 # How to run flask program reference: https://flask.palletsprojects.com/en/2.2.x/quickstart/#:~:text>To%20run%20the%20application%2C%20us
33 from flask import Flask
34 app = Flask(__name__)
35

```

Figure 44-flask library

4.6.4 Admin interface (website)

We proceeded to develop the administrator's interface, which contained a homepage displaying an overview of the stories. This interface also included an "Add" page, where the admin could input the story's title, image and content, then classify the content in order to add to the database, which would then be displayed in the application. Additionally, the interface featured an "Edit" and "Delete" page, permitting the admin to reclassify the story's content, modify the title, image, or content, or remove the story entirely.



4.6.4.1 Classify story

This code in Figure-45 is used to classify a story by sending its content to the server using a POST request and updating the webpage with the result. While the classification is being performed, the user interface keeps showing a "waiting" message. If the classification is successful, the classification result, moral class, is shown to the admin.

```

345      // classifying the story
346      document.querySelector("#classifyButton").addEventListener("click", (event) => {
347          event.preventDefault();
348          const form = new FormData();
349          form.append('content', document.querySelector("#textarea").value);
350          document.getElementById("message1").innerHTML = "waiting ..."
351          axios.post("/Add_pred", form).then(response => {
352              audio.play();
353              console.log(response.data);
354              document.getElementById("message1").innerHTML = ""
355              document.querySelector("#moral").value = response.data
356          }).catch(e => {
357              console.log(e);
358              alert("Failed to add, check developer logs");
359          })
360      }, false);
361
362

```

Figure 45-classify story code

4.6.4.2 Edit story

This function is used to update the story's information in the Firestore database with new information provided by the admin as shown in Figure-46.

```

650      // when the admin edits the book and submits the form,
651      //this function will initiate and replace old values
652      //in the database with new values entered by admin
653      window.completeEdit = async () => {
654          const id = document.getElementById('editId').value
655          const title = document.getElementById('editTitle').value
656          const content = document.getElementById('editContent').value
657          const moral = document.getElementById('editMoral').value
658          const picture = document.getElementById('editPicture')
659
660          document.getElementById(id).innerHTML += 'updating...';
661          document.getElementById('editTitle').value = title
662          document.getElementById('editContent').value = content
663          document.getElementById('editMoral').value = moral
664
665
666          let update = {
667              title: title,
668              titleSmall: title.toLowerCase(),
669              content: content,
670              moral: moral
671          }
672

```

Figure 46- Edit story code

4.6.5 Reader interface (application)

4.6.5.1 Splitting the stories in the application

This code is used to split the content of a book into multiple pages with a maximum word count per page. Additionally, it ensures that sentences are not split mid-way and are only split once they are completed with a period, exclamation mark, question mark, or quotation mark. Figure-47 shows the implemented code and the flowchart in Figure-48 explains the splitting process scenario.

```

for (var i = 0; i < bookPages.length - 1; i++) {
    List<String> linesOfCurrentPage = bookPages[i].split('. ! ? "'");
    String lastLineOfCurrentPage = linesOfCurrentPage.last;
    List<String> correctCurrentList = linesOfCurrentPage..removeLast();
    bookPages[i] = '${correctCurrentList.join('. ! ? "')}.';
    bookPages[i + 1] = '$lastLineOfCurrentPage ${bookPages[i + 1]}';
}

```

Figure 47- splitting code

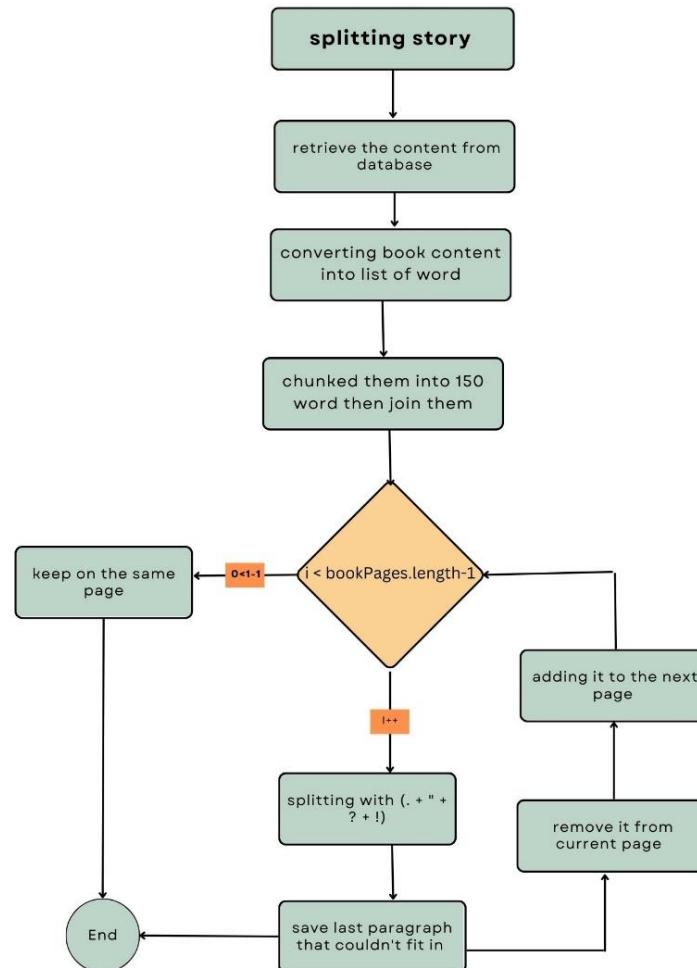


Figure 48-splitting flowchart

4.6.5.2 Text to speech

The code in Figure-49 is used to automate the reading of a book using text-to-speech technology, with the ability to pause and resume the reading process. The progress of the reading is tracked and updated in real-time.

```

147   for (var i = indexOfBookPage.value; i < bookPages.length; i++) {
148     var bookPage = bookPages[i];
149
150     var list = bookPage.split(" ");
151
152     int wordIndex = findWordIndex(
153       list, detectWords.isEmpty ? "abcdefg" : detectWords.last);
154     int index = wordIndex == -1 ? 0 : wordIndex;
155     bookPage = list.sublist(index).join(" ");
156
157     if (isPlaying.value) {
158       pageController.jumpToPage(i);
159
160       tts.setProgressHandler((text, start, end, word) {
161         spokenWords.add(word);
162         detectWords.add(word);
163         prgressValue.value = (spokenWords.length) / (bookWords.length);
164       });
165       await tts.speak(bookPage);
166       await tts.awaitSpeakCompletion(true);
167       if (i == (bookPages.length - 1)) {
168         await tts.stop();
169         isPlaying.value = false;
170         prgressValue.value = 1;
171         spokenWords.clear();
172       }
173     } else {
174       // spokenWords.clear();
175       await tts.stop();
176       pageController.jumpToPage(indexOfBookPage.value);
177     }

```

Figure 49- text to speech code 1

Figure-50 shows the code that is used to read a single page of a book using text-to-speech technology, with the ability to pause and resume the reading process. The only difference it has from Figure-49 is that it starts reading a specific page the reader has chosen or the page the reader bookmarked.

```

278 void playerStory(int i) async {
279     // for (var i = indexOfBookPage.value; i < bookPages.length; i++) {
280     final bookPage = bookPages[i];
281     storyWords = bookPage.split(' ');
282     await tts.setLanguage("en-US");
283     await tts.setVolume(1.0);
284     await tts.setSpeechRate(0.5);
285     await tts.setPitch(1.0);
286
287     if (isPlaying.value) {
288         pageController.jumpToPage(i);
289
290         tts.setProgressHandler((text, start, end, word) {
291             spokenWords.add(word);
292             detectWords.add(word);
293             prgressValue.value = (spokenWords.length) / (bookWords.length);
294         });
295
296         int wordIndex = findWordIndex(storyWords, detectWords.last);
297         int index = wordIndex == -1 ? 0 : wordIndex;
298         String story = storyWords.sublist(index).join(" ");
299
300         await tts.speak(story);
301         // await tts.speak(bookPage.substring(wordIndex, bookPage.length));
302         await tts.awaitSpeakCompletion(true);
303         if (i == (bookPages.length - 1)) {
304             await tts.stop();
305             isPlaying.value = false;
306             prgressValue.value = 1;
307             spokenWords.clear();
308         }
309     } else {
310         spokenWords.clear();
311         await tts.stop();
312         pageController.jumpToPage(indexOfBookPage.value);
313     }
314 }

```

Figure 50- text to speech code 2

We included our GitHub repository which contains all of the code for both the application and the website.

GitHub: <https://github.com/reemaowerdi/GP1/tree/main>



SYSTEM TESTING

— READ ME A STORY 

5 System Evaluation

This chapter involves conducting system testing to ensure that the application and website are functioning as expected. To achieve this, we carried out user acceptance testing on both the application and website.

5.1 Experimental Results

5.1.1 Experimental Set up

We began building RoBERTa model using the PyTorch library from Hugging Face and provided detailed pictures of the libraries used in [RoBERTa model](#). Next, we had to manually annotate each story to start building our dataset from scratch then we did pre-processing and we found that our data was balanced, we discussed this process thoroughly in the [data collection](#) section. After that, we tokenized and encoded the data, splitting it into train and test datasets. We truncate to 512 tokens because our story contains a lot of lines. Then, we fine-tuned RoBERTa model by creating a torch dataset and defining all these specifications (batch size, epochs, max size).

5.1.2 Comparison between RoBERTa and DistilBERT

We tried the DistilBERT model to see if it would give a better result than RoBERTa or not. We compared it when the number of stories was 79 stories and when epoch was 8. In Table-5, we concluded that RoBERTa was better. The reason behind it is the size of the models. RoBERTa is a larger model with significantly more parameters than DistilBERT, which is a distilled version of BERT, another popular language model. RoBERTa was pre-trained on a larger and more diverse dataset than BERT, which led to better performance on many natural language processing tasks[54].

Table 5-comparison (RoBERTa and DistilBERT)

Model name :	Epoch	Training Loss	Validation Loss	Accurac y	F-score	Precision	Recall
RoBERTa	6	0.20	0.68	0.84	0,85	0.89	0.84
DistilBER T	6	0.55	0.82	0.73	0.75	0.84	0.73

5.1.3 Evaluation

- Performance:**

When training the model for the first time the accuracy was 21% which is very bad and low, so we had to work on the dataset itself by:

- 1- Preprocessing (cleaning) by removing stop words.
- 2- Increasing the dataset size by adding more stories from 50 to 79.
- 3- Changing the number of epoch and max capacity multiple times until we reached a higher accuracy.

Finally, we were able to achieve 88% accuracy.

Table 6 shows the epochs we have tried (2, 3, 4, 6, 8) with a max capacity= 512:

Table 6-Accuracy

Epoch	Training Loss	Validation Loss	accuracy	F score	precision	recall
2	1.5	1.35	0.42	0.40	0.49	0.42
3	0.49	0.61	0.77	0.79	0.88	0.77
4	0.70	0.74	0.77	0.79	0.87	0.77
6	0.20	0.68	0.84	0.85	0.89	0.84
8	0.21	0.54	0.88	0.88	0.90	0.88

- Improve the performance:**

To increase the accuracy and to ensure that the classifier categorizes the stories better, we increased the number of stories from 79 to 141 until we reached 91% accuracy as shown in Table-7. It was the same number of epoch which is 8, the only difference was the number of stories.

Table 7- Improve the accuracy

Epoch	Training Loss	Validation Loss	accuracy	F score	precision	recall
8	0.20	0.38	0.91	0.91	0.93	0.91

5.2 User Acceptance Testing

User Acceptance Testing (UAT), or application testing, is “the final stage of any development process to determine that the software does what it was designed to do in real-world situations”[55].

In this section, we conducted user acceptance testing on a total of 30 users, 10 of which were admins while 20 were readers, by allowing them to use the application and website. We collected the information via a questionnaire that asked a variety of questions about their experience using the application/website.

5.2.1 Reader and admin experiments

In this section, we discuss how we conducted the experiments for both the admins and the readers.

5.2.1.1 Reader experiment

The test experiment for the reader was set up as follows:

- Two of the team members tested the system with 4 end-users (readers) and the other two members tested 6 end-users (readers).
- The users underwent testing in two different environments - either at home or at the university. In testing, we divided the application users into 9 children and 11 adults.
- Adults who participated in the testing were between the ages of 20 and 48, while children ranged from 4 to 13 years old. The test was conducted using an iPad running on iOS.
- We wrote a set of tasks for the readers to perform while using the system. We show these tasks in [Appendix C](#).
- Children answered questions either by providing the answers directly or by relying on our observations of their behavior when interacting with the application.
- We observed them during the test and asked them directly for their opinion.

5.2.1.2 Admin experiment

The test experiment for the admin was set up as follows:

- Each team members tested 2 administrators except for one member who tested 4 administrators.
- The test was conducted using a laptop.
- The admins underwent testing in two different environments - either at home or at the university.
- We wrote a set of tasks for the admin to perform while using the system. We show these tasks in [Appendix C](#).
- We observed them during the test and asked them directly for their opinion.

5.2.2 Demographics of Participants

The actual questionnaire is in Appendix B.

5.2.2.1 Demographics of Participants for the admin

Table 8-Demographics of Participants for admin

Demographics of Participants	
What is your age?	<ul style="list-style-type: none"> • (20-30) 4 admins (40%) • (31-40) 4 admins (40%) • (41-50) 2 admins (20%) • (Over 50) 0%
What is your native language?	<ul style="list-style-type: none"> • (Arabic) 100% • (English) 0%
What is the highest degree or level of education you have completed?	<ul style="list-style-type: none"> • (Graduated high school) 3 admins (30%) • (Associate degree) 0% • (Bachelor's degree) 5 admins (50%) • (Post-graduate degree) 2 admins (20%)

5.2.2.2 Demographics of Participants for the reader

Table 9-Demographics of Participants for reader

Demographics of Participants	
What is your age?	<ul style="list-style-type: none"> (Under 15) - 9 children (45%) (15-25) - 2 readers (10%) (26-35) - 3 readers (15%) (36-45) 4 readers (20%) (Over 45) 2 readers (10%)
What is your native language?	<ul style="list-style-type: none"> (Arabic) - 20 readers (100%) (English) - 0%
What is the highest degree or level of education you have completed?	<ul style="list-style-type: none"> (2th grade or less) - 9 children (45%) (Graduated high school) - 2 readers (10%) (Associate degree) - 0% (Bachelor's degree) - 8 readers (45%) (Post-graduate degree) - 1 reader (5%)

5.2.3 Questionnaire/Interview Results

5.2.3.1 Questionnaire/Interview Results for the admin

We summarized the results of the questionnaire in the form of a bar chart. The answers range from 1 to 5(1 for Strongly disagree – 5 for Strongly agree). The actual questionnaire is in [Appendix B](#).

Table 10-test questionnaire for admin

Experience as an admin ^[OBJ]	
Questions	Responses
What is your level of technical experience?	<ul style="list-style-type: none"> (1 - Novice) - 0% (2 - Advanced beginner) - 0%

	<ul style="list-style-type: none"> • (3 - Competent) - 3 admins (30%) • (4 - Proficient) - 4 admins (40%) • (5 - Expert) - 3 admins (30%)
Reaction to the system	
Questions	Responses
The icons found in the website are well known and indicate what it does.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 10 admins (100%)
It is clear that the Disapprove button in add form saves the story in another dataset collection.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 2 admins (20%) • (5 - Strongly agree) - 8 admins (80%)
I found the error messages in the website are clear and indicate the details of the error.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 10 admins (100%)
System Usability Scale	
Questions	Response
I think that I would like to use this system frequently.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 1 admin (10%) • (5 - Strongly agree) - 9 admins (90%)
I found the system unnecessarily complex.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 10 admins (100%) • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0%

	<ul style="list-style-type: none"> • (5 - Strongly agree) - 0%
I thought the system was easy to use.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 10 admins (100%)
I think that I would need the support of a technical person to be able to use this system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 9 admins (90%) • (2 - Disagree) - 1 admin (10%) • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%
I found the various functions in this system were well integrated.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 10 admins (100%)
I thought there was too much inconsistency in this system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 10 admins (100%) • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%
I would imagine that most people would learn to use this system very quickly.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 2 admins (20%) • (5 - Strongly agree) - 8 admins (80%)
I found the system very cumbersome to use.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 10 admins (100%) • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%

I felt very confident using the system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 1 admin (10%) • (5 - Strongly agree) - 9 admins (90%)
I needed to learn a lot of things before I could get going with this system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 8 admins (80%) • (2 - Disagree) - 2 admins (20%) • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%

5.2.3.2 Questionnaire/Interview Results for the reader

Table 11-test questionnaire for reader

Experience as a reader	
Questions	Response
Familiarity with iOS devices.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 1 reader (5%) • (3 - Neutral) - 1 reader (5%) • (4 - Agree) - 2 readers (10%) • (5 - Strongly agree) - 16 readers (80%)
Reaction to the system	
Questions	Response
Stories were displayed quickly when clicking on the moral.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 1 reader (5%) • (5 - Strongly agree) - 19 readers (95%)
The icons found in the application are well known and indicate what it does.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 2 readers (10%) • (5 - Strongly agree) - 18 readers (90%)
I found the error messages in the application are clear and indicate the details of the error.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 20 readers (100%)
Reactions to the system interface design	
Questions	Response
User-friendly interface design.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 1 reader (5%)

	<ul style="list-style-type: none"> • (5 - Strongly agree) - 19 readers (95%)
The colors are comfortable and not distracting.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 20 readers (100%)
The app used an appropriate font type.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 1 reader (5%) • (4 - Agree) - 2 readers (10%) • (5 - Strongly agree) - 17 readers (85%)
System Usability Scale	
Questions	Response
I think that I would like to use this system frequently.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 3 readers (15%) • (5 - Strongly agree) - 17 readers (85%)
I found the system unnecessarily complex.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 20 readers (100%) • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%
I thought the system was easy to use.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 2 readers (10%) • (5 - Strongly agree) - 18 readers (90%)
I think that I would need the support of a technical person to be able to use this system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 19 readers (95%) • (2 - Disagree) - 1 reader (5%) • (3 - Neutral) - 0%

	<ul style="list-style-type: none"> • (4 - Agree) - 0% • (5 - Strongly agree) - 0%
I found the various functions in this system were well integrated.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 20 readers (100%)
I thought there was too much inconsistency in this system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 20 readers (100%) • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%
I would imagine that most people would learn to use this system very quickly.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 1 reader (5%) • (5 - Strongly agree) - 19 readers (95%)
I found the system very cumbersome to use.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 20 readers (100%) • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 0% • (5 - Strongly agree) - 0%
I felt very confident using the system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 0% • (2 - Disagree) - 0% • (3 - Neutral) - 0% • (4 - Agree) - 1 reader (5%) • (5 - Strongly agree) - 19 readers (95%)
I needed to learn a lot of things before I could get going with this system.	<ul style="list-style-type: none"> • (1 - Strongly disagree) - 18 readers (90%) • (2 - Disagree) - 2 readers (10%) • (3 - Neutral) - 0% • (4 - Agree) - 0%



	• (5 - Strongly agree) - 0%
Comments	
please give us a comment or feedback	
7 responses	
I loved it	
عجيبتي الالوان واني اقدر اسمع القصه بصوت بنت او ولد	
I liked the pictures of stories	
التطبيق فكرته رائعة خصوصا لي انا كأم احتاج الي افرا لبني قصص توصل قيمة معينه انا حابه اوصلها لها ، بال توفيق يارب ❤️ ❤️	
I liked the application very much, especially the feature of listening to the story. My five-year-old son loved it very much	
متحمسه جدا لاستخدام تطبيقكم على طلباتي الشي الناقص الوحيد هو اني تمنيت يكون فيه قيم اكتر مثل المشاركه مع الاخرين ، المسؤوليه غير كذا كل شي ممتاز ننتظره بصير موجود ونقدر نستخدمه ❤️ 👍	
Having stories categorised based on morals made it easier for me to choose specific morals, but I wish there were more moral choices and it would be nice if the font were a bit bigger, but the rest is great especially the colors.	

5.3 Quality Attributes (NFR testing)

We tested the NFR on Read me a story application and all the three non-functional requirements (Performance, Usability, and Availability) were successfully implemented as shown in the Table below.

Table 12-Quality Attributes

User story	Quality Attribute	Measure	Results
As a reader, I want the number of steps to get to the story to be as little as possible, so that I can reach the story quickly.	Performance: How responsive is the system and its components?	Compute the time it takes the user to get to the story.	<ul style="list-style-type: none"> It takes the user 2 steps to get to the story from the morals page. 1-Choose the moral. 2-Choose the story. 8 users completed the test. On average, it took them 3.29 seconds to get to the story.
As a reader, I want a friendly interface so that I can interact with the interface easily, and that I don't make more than 5 mistakes while using the Read me a story application.	Usability: How the user can learn and operate through interactions with the system.	Users need to complete the 3 main functionalities without making any mistakes.	<ul style="list-style-type: none"> 8 users tested 3 main features: login, add the story to favorite list, and listen to the story. We found that 8 out of the 8 users successfully completed the sign-in process. 8 out of 8 users added the story to the favorites list successfully.

			8 out of 8 users completed the listening process successfully.
	Availability: How likely the system is accessible to a user at a given point in time.	Compute the availability of the favorite list by adding many stories at the same time (over loading the system).	We measured the favorite list to be available offline, by adding many stories to the favorite list, using 4 different accounts, tablets, network speed and user behavior. All of the stories were able to be accessed offline and have been added successfully.

5.4 Discussion

After analyzing the results from both the admin and the readers sides, we concluded that the overall results from the system evaluation were good based on the results of the user acceptance testing. The admin's reaction to the system went well, if one thing required improvement it would be to make the disapprove button clearer on its intention since 2 out of 10 users found it unclear. For the admin's system usability scale results, we calculated it to be an average of 98.25 which is an excellent rating based on the table below obtained from this research(Measuring and Interpreting System Usability Scale (SUS))[56].

Table 13- SUS score guide

SUS score	Grade	Adjective rating
> 80.3	A	Excellent
68 - 80.3	B	Good
68	C	Okay
51 - 68	D	Poor
< 51	F	Awful

Since the results range from 1 to 5 (1 being strongly disagree and 5 being strongly agree), we assign those points for each response received. The calculation procedure for the SUS process[56] was as follows:

- Add up the total score for all odd-numbered questions, then subtract 5 from the total to get (X).
- Add up the total score for all even-numbered questions, then subtract that total from 25 to get (Y).
- Add up the total score of the new values (X+Y) and multiply by 2.5.
- We repeat these steps for every user, then we divide it by the number of users to get the final SUS score.

During the testing phase of the application, we collected feedback from both children and adults. Children showed a strong interest in the visually appealing graphics, story pictures, and audio storytelling. They particularly enjoyed being able to select between a male and female voice. On the other hand, we also tested the application with mothers, kindergarten teachers, and aunts in the adult group. We observed that they were more interested in the story content and appreciated the idea of categorizing the stories based on morals, which was not present in other applications. The font size was suitable for most users, with 17 finding it appropriate. However, 3 of them expressed a preference for a larger

font size. For the reader's system usability scale results, we calculated it to be an average of 98.75, which is also an excellent rating based on Table-13. We calculated the SUS score using the steps mentioned above. Also, all the three non-functional requirements (Performance, Usability, and Availability) were successfully implemented. As shown in Table 12, we have represented how we measured each one of them. For the performance we have calculated the steps taken by the user to get to the story, which were 2 steps: 1- choose the story, 2- choose the moral. 8 users completed the test and on average it took them 3.29 seconds to get to the story. As for the availability, we have tested the favorite list to be available offline, since it is the feature that needs availability the most. Furthermore, we calculated the usability during the testing by allowing users to test the first main functionality which is log in, and then test the other 2 main functionalities in the application. 1-add the story to the favorite list and 2-listen to the story. We observed them while they were testing these 3 main functionalities, and all 8 users completed the test successfully without making any mistakes.

CONCLUSIONS AND FUTURE WORK

— READ ME A STORY  —

6 Conclusions and Future Work

We presented “Read me a story”, an application that classifies children’s stories based on morals. “Read me a story” project's development phases are described in this document. Starting with the problem, solution, and scope of the project are all introduced in the first chapter. In the background chapter, we presented a brief overview of our knowledge of the types of machine learning and the importance of using transformers because they can classify the children’s stories into morals in our app. Then, in the literature review section, we looked at and discussed related research papers, analyzed and reviewed related apps, and viewed several children’s book classifications connected to our idea. Additionally, none of them were categorizing their stories into categories based on morals and values. So, we presented our solution.

We started the chapter on system description by introducing our system and handing out questionnaires to gather more information for a deeper understanding. We found that educators, parents, and other relatives need an application that helps them find stories based on morals in order to teach their children how to distinguish between right and wrong. Furthermore, we described the criteria for our system, including both functional and non-functional requirements. Our system's architectural diagram was covered in the chapter on system design, we chose Model-View-Controller Architecture because it's the default application architecture pattern used in all iOS. Then, it was followed by class diagram, data design, component level designs, and lastly user interface design. Finally, "Read me a story" application was built using flutter framework and the RoBERTa model developed using Python with Google Collab. Which we covered in the implementation process in the chapter on implementation. Finally, In the testing chapter, we observed that the system was well-received by users, with excellent ratings in system usability. The feedback received from both children and adults helped us identify areas for improvement, such as making the disapprove button clearer and allowing for a larger font size option.

6.1 Global and local impact

- Local impact

Our application focuses on teaching children morals through the use of stories in English. By that, it will not only help children learn morals, but it will also serve to be beneficial in that it will be a useful resource to teach Arab children the English language.

- Global impact

According to the UNESCO Institute of Statistics: One third of the globe's youth is bullied.[57] Moral self-view influences both a person's capacity for positive social interactions and their potential for aggressive and bullying behavior. Because they treat other people with respect, someone who is socially decent and ethical may be less prone to bullying. By teaching morals to children, it will not only decrease the risk of them being bullies, but it will also help them act appropriately in situations. For example, a morally engaged student may be bothered by the mistreatment of others and find it harder to ignore acts of misconduct happening around them.[58]

6.2 Problems and challenges encountered during the software development

One of the hardest challenges we faced was gathering knowledge about the classifier packages. We also faced a lot of errors while training the model. When we took the RoBERTa code from GitHub it had some errors in the tokenizer and libraries installation. At first, we fixed these errors from stack overflow. Then, we fine-tuned it in order to fit with our dataset. We were able to train the model very well using TensorFlow (which is a software library for machine learning), but unfortunately, we couldn't fix the prediction part, so we had to try another code using PyTorch.

Another difficulty we faced was finding a suitable platform to run the RoBERTa code efficiently. We tried Visual Studio Code then created venv(virtual environments) using terminal to install all libraries, but we found that Google Collab was the most suitable option for us.

We also struggled with the accuracy at first, when we fine-tuned the model for the first time, the accuracy only amounted to 21% which was very poor, and we kept trying to improve

it by adding more stories to the dataset and changing the epoch value until it reached 91% which took a long time to do. Moreover, learning how to program the Dart language using Flutter framework was also a huge challenge, as it not only required a lot of time to learn a new programming language, but we also faced a lot of errors starting from software version issues to firebase connection.

We also faced challenges while downloading the Xcode, first we downloaded flutter and then we downloaded “Xcode 13.3 version” it took us 12 hours to download it, and finally after it finished we connected it with Visual Studio Code and flutter but unfortunately, it did not work, it kept showing errors. After research we realized that 13.3 version is not compatible with our macOS big sur 11.6 version, so we had to delete it and reinstall the compatible version, so at the end it took us 3 days to finally be able to start working on the code, which was exhausting.

We used firebase as mentioned above to store our dataset, and regarding the dataset, there were no datasets that included each story with its moral. So, we had to start building our dataset from scratch and manually annotated each story, which took us almost 1 week to do, while also taking into consideration if they were free of copyright. After storing the dataset in the firebase, we then tried to connect it to our flutter code to retrieve the data, but it did not work at first. We followed multiple tutorials, used a lot of resources, watched a lot of videos but it kept showing errors. Overall, it took us more than 2 weeks for the connection to be successful.

For the admin interface, the most challenging part of it is that we had to use Python language to implement the interface, since it was our first time implementing a web page interface using python. The reason for using python was so we would be able to connect it with the RoBERTa model python code. We then had to connect it with the firebase, which required a lot of research and time. But thankfully, both were successfully done.

When integrating the classifier model code with the flask code we faced a lot of problems particularly with the prediction part of the classifier code. This problem lowered our morale since it took us about 2 weeks to fix but, in the end we solved the problem successfully.

We also encountered a problem in dividing the story into pages. How do we divide it and based on what? This feature took more time than expected until we reached a solution. After enabling the Text-To-Speech feature, it functioned correctly. However, when we attempted to add a progress bar to pause and resume the sound, we encountered numerous

issues on the page that resulted in many bugs. It took us a considerable amount of time to identify the root cause, which we eventually found on the website pub.dev[59]. Once we discovered the issue, we were able to resolve it and prevent further problems.

6.3 Limitations of the system

“Read me a story” is a tablet application that aims to make it easier for parents and educators to teach children the morals they want to teach. Whether it is to better their character, help them respond to specific situations, or help them become a better person overall. Unfortunately, due to limited time and scope, “Read me a story” does not support any other language other than English. In addition, the number of stories and morals are only limited.

6.4 The main contribution of the project

Our project contributed to documenting the importance of teaching morals to children as well as providing the means to do so. We made it possible for parents and educators to find stories based on the morals they want to teach. We hope that will help spread the importance of this topic and potentially see the teaching of morals through stories implemented in schools.

In our project, we learned many things about working with agile software development such as the importance of communication and organization, the significance of maintaining a constant and consistent pace of work, along with responding to change and not always sticking to a plan. We also acquired skills in time management while considering the impact of other factors, i.e., providing time for other subjects. We also learned new programming concepts and technologies such as programming by Dart language using Flutter framework, using a NoSQL database for the first time and connecting it to our system, fine-tuning a machine learning model, developing an interface also using python to be able to connect it with the model in a future sprint, and finally, solving problems encountered during the development phase. Overall, we believe that our project helped improve our teamwork skills such as collaboration, active listening, responsibility, etc. as well as help improve our programming skills.



6.5 Future work

On our next journey, "Read me a story" plans to provide more features to make the users have a better experience using our application. Such as:

- Providing books on multiple languages.
- Adding more morals and books.
- Supporting the Arabic language.
- Allowing readers to rate the books.
- Recommending books to the reader based on what books they have read in the past.



ACKNOWLEDGMENTS

— READ ME A STORY  —

7 Acknowledgments

First and foremost, we would like to praise Allah the Almighty, the Most Gracious, and the Most Merciful for giving us the opportunity, courage, and energy in completing our graduation project. We would also like to express our deepest gratitude to our supervisors Dr. Balsam Alsugair and Dr. Manal Albahal for their continuous support and encouragement that made all this possible. We were extremely lucky to have supervisors who cared so much about our work and progress. Their guidance and advice helped us tremendously throughout the completion of the project. In addition, we would like to express our special thanks to the committee members of this project for providing guidance in areas of their expertise. Their honest judgement and recommendations only motivated us to do better. Finally, we would like to thank our scrum master, Dr. Maha Alyahya, for shedding light on several areas where we left confused and for always being willing and enthusiastic to help us.

REFERENCES

— READ ME A STORY  —

8 References

- [1] Can classic moral stories promote honesty in children? - researchgate. (n.d.). Retrieved September 13, 2022, from :
https://www.researchgate.net/publication/263098641_Can_Classic_Moral_Stories_Promote_Honesty_in_Children.
- [2] Prindle Institute. 2022. The Cat in the Hat - Teaching Children Philosophy - Prindle Institute. Available at: <https://www.prindleinstitute.org/books/the-cat-in-the-hat/>.
- [3] Ponio, J. (2019). How Important Are Moral Values in Children? | Our Father's House. [online] Our Father's House Soup Kitchen. Available at:
<https://ofhsoupkitchen.org/importance-moral-values-children>.
- [4] peechify.com. (2022). Audible Vs. Storytel Audiobooks: What You Need To Know. [online] Available at: https://speechify.com/blog/audible-vs-storytel-audiobooks/?landing_url=https%3A%2F%2Fspeechify.com%2Ftext-to-speech-online%2F.
- [5] Selig, J. (2022c, July 4). What Is Machine Learning? A Definition. Expert.ai. Retrieved September 27, 2022, from <https://www.expert.ai/blog/machine-learning-definition/>
- [6] Burns, E. (2021b, March 30). machine learning. SearchEnterpriseAI. Retrieved September 19, 2022, from
<https://www.techtarget.com/searchenterpriseai/definition/machinehttps://www.techtarget.com/searchenterpriseai/definition/machine-learning-MLlearning-ML>
- [7] Education, I. C. (2021, June 30). Supervised Learning. Retrieved September 25, 2022, from <https://www.ibm.com/cloud/learn/supervised-learning>
- [8] Education, I. C. (2022, March 31). Unsupervised Learning. Retrieved September 18, 2022, from <https://www.ibm.com/cloud/learn/unsupervised-learning#:~:text=Unsupervised%20learning%2C%20also%20known%20as,the%20one~ed%20for%20human%20intervention>
- [9] Basic Concepts in Machine Learning - Javatpoint. (n.d.). [www.javatpoint.com](http://www.javatpoint.com/basic-concepts-in-machine-learning). Retrieved September 25, 2022, from <https://www.javatpoint.com/basic-concepts-in-machine-learning>
- [10] Alammar, J. (n.d.). The Illustrated Transformer. Retrieved September 19, 2022, from <http://jalammar.github.io/illustrated-transformer/>
- [11] Merritt, R. (2022, September 16). What Is a Transformer Model? | NVIDIA Blogs. NVIDIA Blog. Retrieved September 19, 2022, from
<https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/>
- [12] structure of the token (n.d.-b). Retrieved September 25, 2022, from
https://www.researchgate.net/figure/Structure-of-the-tokens-passed-to-and-received-from-ahttps://www.researchgate.net/figure/Structure-of-the-tokens-passed-to-and-received-from-a-transformer-model-Both-labeled-and_fig2_357791308transformer-model-Both-labeled-and_fig2_357791308
- [13] What Are Word Embeddings for Text? (2019, August 7). Retrieved September 20, 2022, from <https://machinelearningmastery.com/what-are-word-embeddings/>
- [14] Papers with Code - Multi-Head Attention Explained. (n.d.). Retrieved September 20,

- 2022, from <https://paperswithcode.com/method/multi-head-attention>
- [15] Pere, C. (2020). What are Loss Functions? [online] Medium. Available at: <https://towardsdatascience.com/what-is-loss-function-1e2605aeb904>.
- [16] Horev, R. (2018b, November 17). BERT Explained: State of the art language model for NLP. Medium. Retrieved September 19, 2022, from <https://towardsdatascience.com/bertexplained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>
- [17] Lan, Z. (2019, September 26). ALBERT: A Lite BERT for Self-supervised Learning of Language. . . arXiv.org. Retrieved September 19, 2022, from https://arxiv.org/abs/1909.11942?source=post_page-----1647f4438797-----
- [18] Definition of Roberta. (n.d.). In www.dictionary.com. Retrieved September 25, 2022, from <https://www.dictionary.com/browse/roberta>
- [19] 1.12. Multiclass and multioutput algorithms. (n.d.). Scikit-learn. Retrieved September 29, 2022, from <https://scikit-learn.org/stable/modules/multiclass.html> .
- [20] Multi-Label Classification with Scikit-MultiLearn. (n.d.). Engineering Education (EngEd) Program | Section. Retrieved September 29, 2022, from <https://www.section.io/engineering-education/multi-label-classification-with-scikit-https://www.section.io/engineering-education/multi-label-classification-with-scikit-multilearn/multilearn/>.
- [21] Analytics Vidhya. (2021). Difference between Multi-Class and Multi-Label Classification. [online] Available at: <https://www.analyticsvidhya.com/blog/2021/07/demystifying-the-difference-between-multihttps://www.analyticsvidhya.com/blog/2021/07/demystifying-the-difference-between-multi-class-and-multi-label-classification-problem-statements-in-deep-learning/class-and-multi-label-classification-problem-statements-in-deep-learning/>.
- [22] B, H.N. (2020). Confusion Matrix, Accuracy, Precision, Recall, F1 Score. [online] Medium. Available at: <https://medium.com/analytics-vidhya/confusion-matrix-accuracyhttps://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cdprecision-recall-f1-score-ade299cf63cd>
- [23] Built In. (n.d.). When Accuracy Isn't Enough, Use Precision and Recall to Evaluate Your Classification Model. [online] Available at: <https://builtin.com/data-science/precisionhttps://builtin.com/data-science/precision-and-recalland-recall>.
- [24] Koehrsen, W. (2021, August 3). Use Precision and Recall to Evaluate Your Classification Model When Accuracy Isn't Enough. Built In. Retrieved September 29, 2022, from <https://builtin.com/data-science/precision-and-recall>
- [25] www.hindawi.com. (n.d.). A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification. [online] Available at: <https://www.hindawi.com/journals/jhe/2022/3498123/#literature-review> [Accessed 19 Sep. 2022].
- [26] Izadi, M., Heydarnoori, A. and Gousios, G. (2021). Topic recommendation for software repositories using multi-label classification algorithms. *Empirical Software Engineering*,
- [27] O'Sullivan, S. (2004). Books to live by: Using children's literature for character education.
- [28] Weaver, C. (1994) Reading process and practices: From socio-psycholinguistics to whole language. Portsmouth: Heinemann.
- [29] Salih, I. (no date) Using children's literature in the teaching of moral education in

- Nigerian primary schools, Globalacademicgroup.com. Available at:<https://www.globalacademicgroup.com/journals/knowledge%20review/Ibrahim.pdf>
- [30] Association for Project Management (2016). Agile project management. [online] Apm.org.uk. Available at: <https://www.apm.org.uk/resources/find-a-resource/agile-project-management/>.
- [31] Westland, J. (2019). Understanding Scrum Methodology - A Guide. [online] ProjectManager.com. Available at: <https://www.projectmanager.com/blog/scrum-methodology>.
- [32] Atlassian (n.d.). Agile roadmaps: build, share, use, evolve. [online] Atlassian. Available at: <https://www.atlassian.com/agile/product-management/roadmaps#:~:text=agile%20product%20roadmap%3F->.
- [33] What Is Response Time & How to Reduce It. (2022, August 2). Sematext. Retrieved September 27, 2022, from <https://sematext.com/glossary/response-time/> .
- [34] How To Know if Your Machine Learning Model Has Good Performance | Obviously AI. (n.d.). Retrieved September 29, 2022, from <https://www.obviously.ai/post/machinelearning-model-performance> .
- [35] Golovnia, S. (2022) 15 non-functional requirements in software engineering [2023], Custom Software Development & IT Staff Augmentation in USA. from <https://www.sfappworks.com/blogs/non-functional-requirements-in-software-engineering> .
- [36] The 7 Most Popular iOS Architecture Patterns Explained - Zero To App Store. (n.d.). Retrieved September 20, 2022, from <https://www.zerotoappstore.com/the-most-popular-ios-architecture-patterns-explained.html> .
- [37] Benefit of using MVC - GeeksforGeeks. (n.d.). Retrieved September 20, 2022, from <https://www.geeksforgeeks.org/benefit-of-using-mvc/amp/> .
- [38] firebase.flutter.dev. (n.d.). Cloud Firestore | FlutterFire. [online] Available at: <https://firebase.flutter.dev/docs/firestore/overview/> [Accessed 2 Nov. 2022].
- [39] Guan, J., Liu, Z. and Huang, M. (no date) A corpus for understanding and generating moral stories, ACL Anthology. Available at: <https://aclanthology.org/2022.nacl-main.374.pdf> .
- [40] Project Gutenberg. (2019). Project Gutenberg. [online] Available at: <https://www.gutenberg.org/>.
- [41] Mum, M.W. (2018). How to Calculate a Ratio of 3 Numbers. [online] Maths with Mum. Available at: <https://www.mathswithmum.com/calculate-ratio-3-numbers/>.
- [42] Nielsen, J. (1994). 10 Heuristics for User Interface Design. [online] Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- [43] Wong, E. (2018). Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces. [online] The Interaction Design Foundation. Available at: <https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces>.
- [44] Katzir, T., Hershko, S., & Halamish, V. (2013). The Effect of Font Size on Reading Comprehension on Second and Fifth Grade Children: Bigger Is Not Always Better. *PLOS ONE*, 8(9), e74061. <https://doi.org/10.1371/journal.pone.0074061>
- [45] Bernard, M. E., Chaparro, B. S., Mills, M., & Halcomb, C. G. (2002). Examining children's reading performance and preference for different computer-displayed text. *Behaviour & Information Technology*, 21(2), 87–96. <https://doi.org/10.1080/01449290210146737>

- [46] Hachcham, A. (2021). Natural Language Processing with Hugging Face and Transformers. [online] neptune.ai. Available at: <https://neptune.ai/blog/natural-language-processing-with-hugging-face-and-transformers>.
- [47] Flutter - Build apps for any screen. (n.d.). Retrieved November 2, 2022, from <https://flutter.dev/>
- [48] Dart overview. (n.d.). Dart. Retrieved November 2, 2022, from <https://dart.dev/overview>
- [49] What is Visual Studio Code? (n.d.). Educative: Interactive Courses for Software Developers. Retrieved November 2, 2022, from <https://www.educative.io/answers/what-is-visual-studio-code>
- [50] PCMag. (n.d.). Definition of Xcode. PCMag. Retrieved November 2, 2022, from <https://www.pcmag.com/encyclopedia/term/xcode>
- [51] Error on pod install. (2013, August 14). Stack Overflow. <https://stackoverflow.com/questions/18224627/error-on-pod-install>
- [52] Google (2018). Firebase. [online] Firebase. Available at: <https://firebase.google.com/>.
- [53] scoutapm.com. (n.d.). Python Flask Tutorial - Getting Started with Flask | Scout APM Blog. [online] Available at: <https://scoutapm.com/blog/python-flask-tutorial-getting-started-with-flask>.
- [54] Sanh, V. (2019). Smaller, faster, cheaper, lighter: Introducing DilBERT, a distilled version of BERT. [online] Medium. Available at: <https://medium.com/huggingface/distilbert-8cf3380435b5>.
- [55] Elazar, E. (2018). What is User Acceptance Testing (UAT) - The Full Process Explained. [online] Panaya. Available at: <https://www.panaya.com/blog/testing/what-is-uat-testing/>.
- [56] Smyk, A. (2020). The System Usability Scale & How it's Used in UX | Adobe XD Ideas. [online] Ideas. Available at: <https://xd.adobe.com/ideas/process/user-testing/system-usability-scale-ux/>.
- [57] uis.unesco.org. (2018). New SDG 4 Data on Bullying. [online] Available at: <http://uis.unesco.org/en/news/new-sdg-4-data-bullying>.
- [57] Affairs (ASPA), A.S. for P. (2019). Preventing Bullying Through Moral Engagement. [online] StopBullying.gov. Available at: <https://www.stopbullying.gov/resources/research-resources/preventing-bullying-through-moral-engagement-research-summary>.
- [59] Dart packages. (n.d.). flutter_tts | Flutter Package. [online] Available at: https://pub.dev/packages/flutter_tts.

9 Appendices

9.1 Appendix A: Dataset Collection

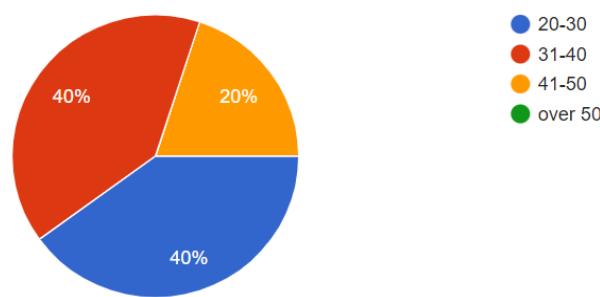
- [1] Project Gutenberg. (2019). Project Gutenberg. [online] Available at: <https://www.gutenberg.org/>.
- [2] www.lizstoryplanet.com. (n.d.). Read Interesting Moral Stories for Kids - Liz Story Planet. [online] Available at: <https://www.lizstoryplanet.com/>.
- [3] Library of Congress, Washington, D.C. 20540 USA. (n.d.). Free to Use and Reuse: Classic Children's Books | Free to Use and Reuse Sets | Library of Congress. [online] Available at: <https://www.loc.gov/free-to-use/classic-childrens-books/>.
- [4] Free Kids Books. (n.d.). Public Domain. [online] Available at: <http://freekidsbooks.org/license/public-domain/> .
- [5] Anon, (n.d.). MoralStories26.com - A Collection of Best Stories on Moral Values in English. [online] Available at: <https://moralstories26.com/>.
- [6] www.taleswithmorals.com. (n.d.). Aesop's Fables. [online] Available at: <https://www.taleswithmorals.com/>.
- [7] English for Students. (n.d.). Moral Stories. [online] Available at: <http://www.english-for-students.com/Moral-Stories.html> .

9.2 Appendix B: User Acceptance Testing

Demographics of Participants for admin:

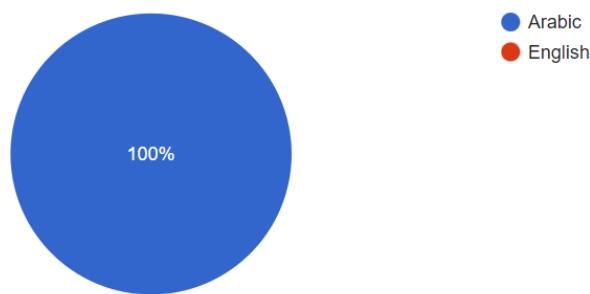
What is your age ?

10 responses



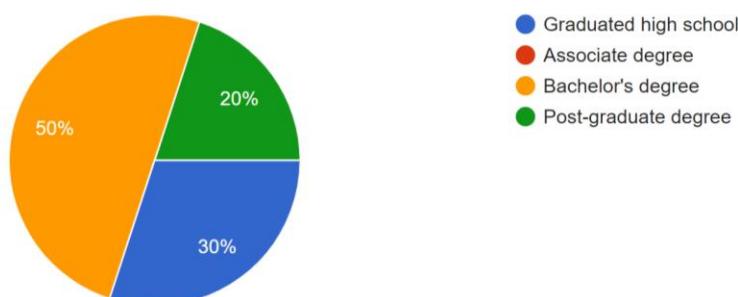
What is your native language?

10 responses



What is the highest degree or level of education you have completed?

10 responses

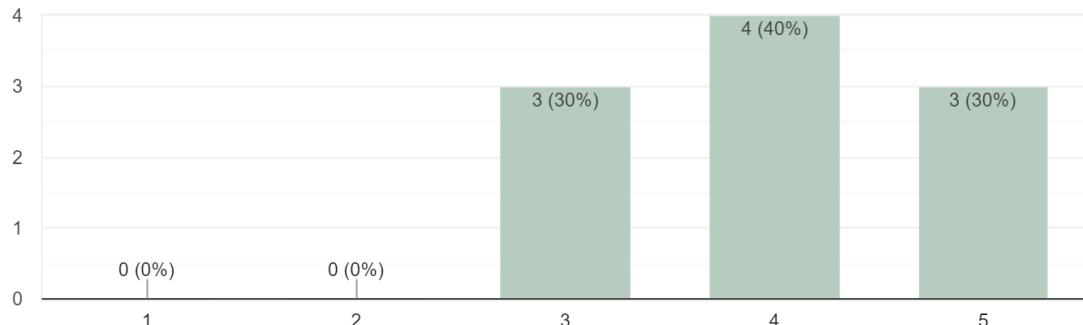


Questionnaire/Interview Results for admin:

What is your level of technical experience?

 Copy

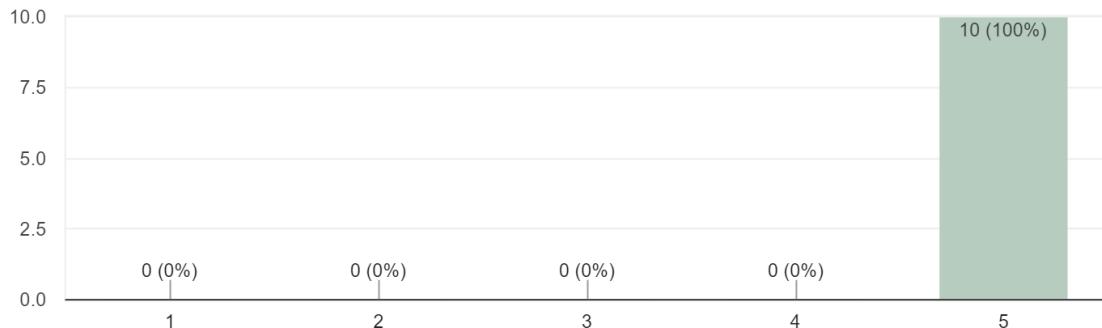
10 responses



The icons found in the website are well known and indicate what it does.

 Copy

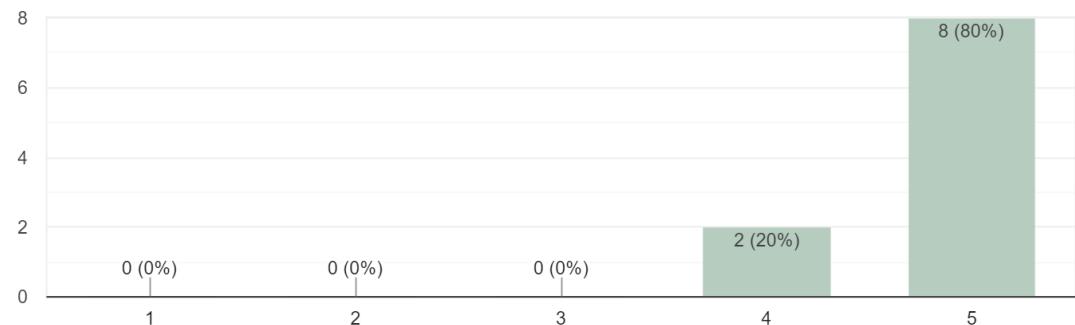
10 responses



It is clear that the Disapprove button in add form saves the story in another dataset collection

 Copy

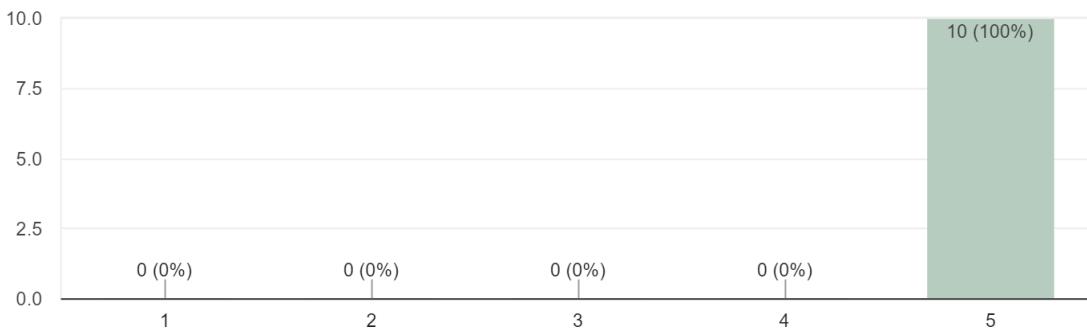
10 responses



I found the error messages in the website are clear and indicate the details of the error.

Copy

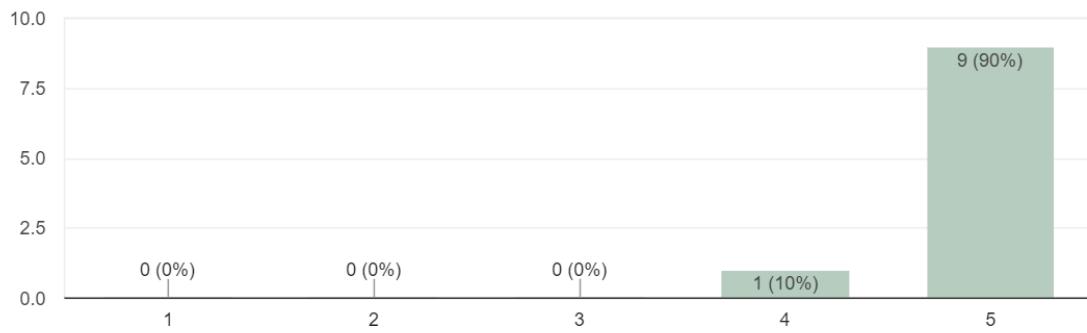
10 responses



I think that I would like to use this system frequently.

Copy

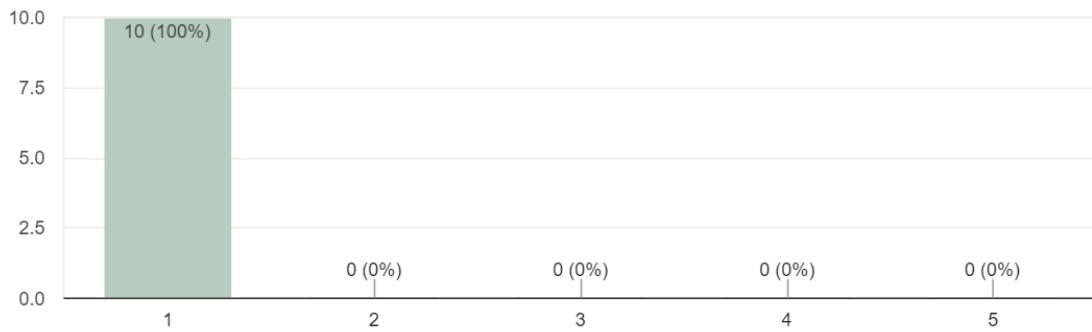
10 responses



I found the system unnecessarily complex.

Copy

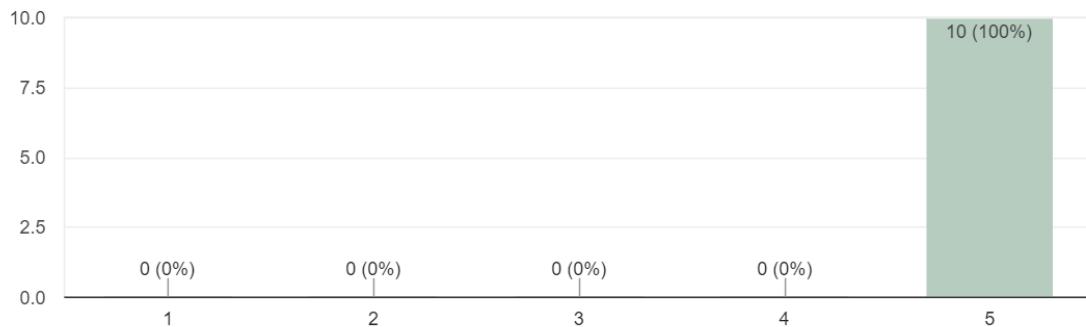
10 responses



I thought the system was easy to use.

 Copy

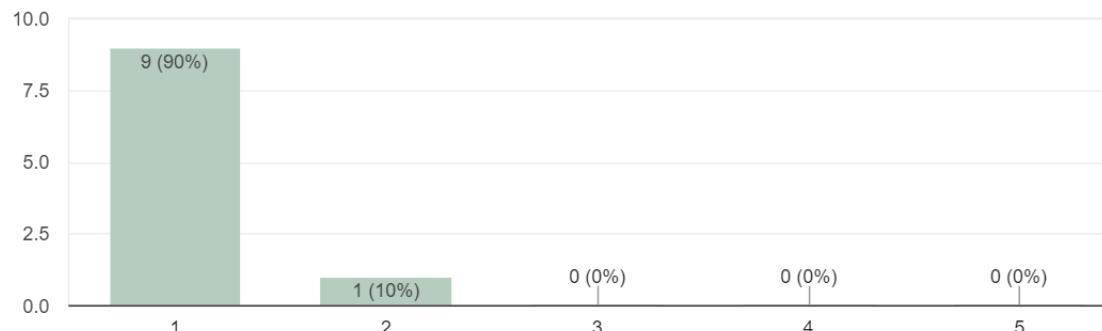
10 responses



I think that I would need the support of a technical person to be able to use this system.

 Copy

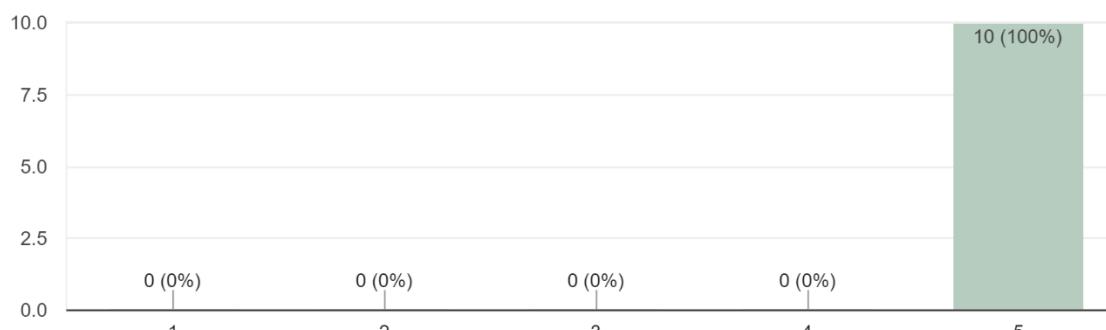
10 responses



I found the various functions in this system were well integrated.

 Copy

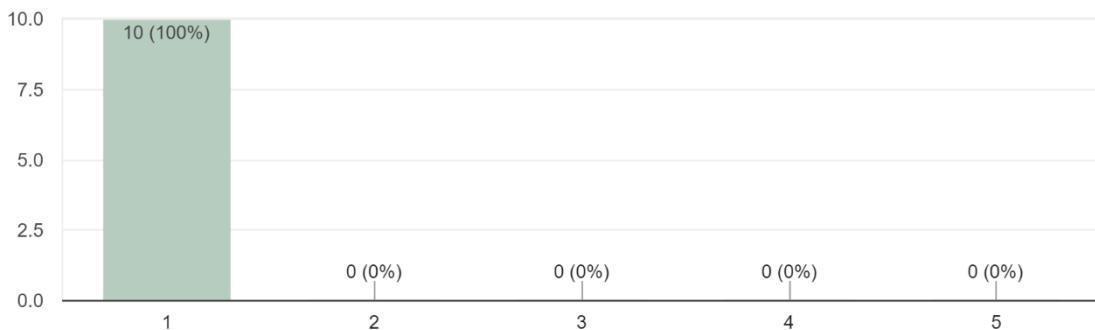
10 responses



I thought there was too much inconsistency in this system.

 Copy

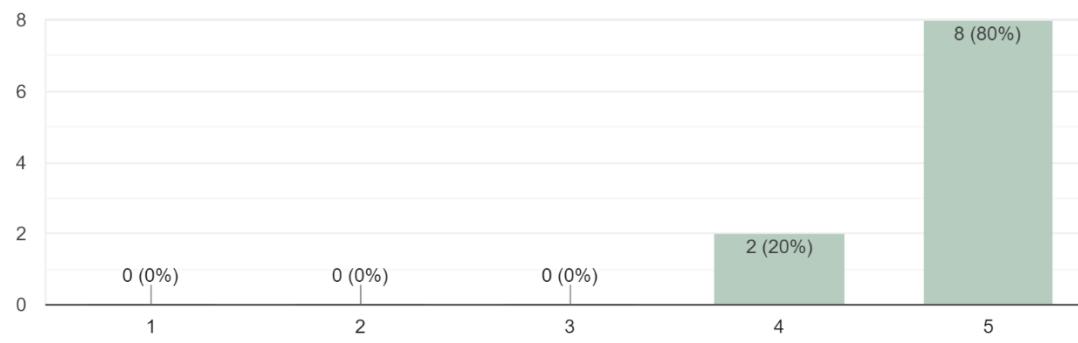
10 responses



I would imagine that most people would learn to use this system very quickly.

 Copy

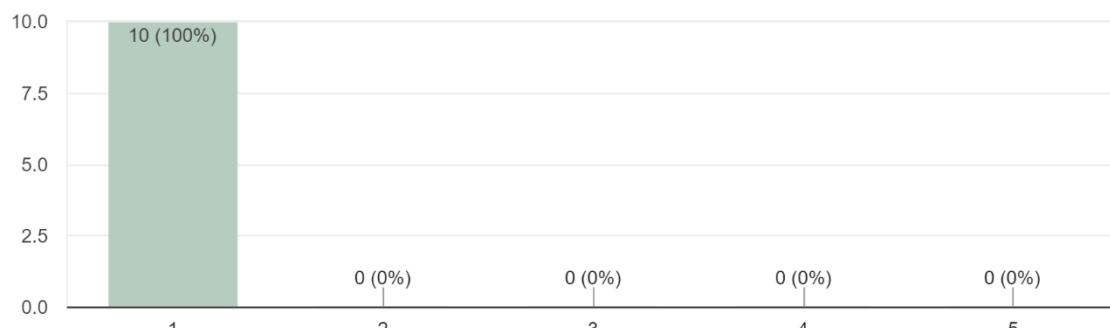
10 responses



I found the system very cumbersome to use.

 Copy

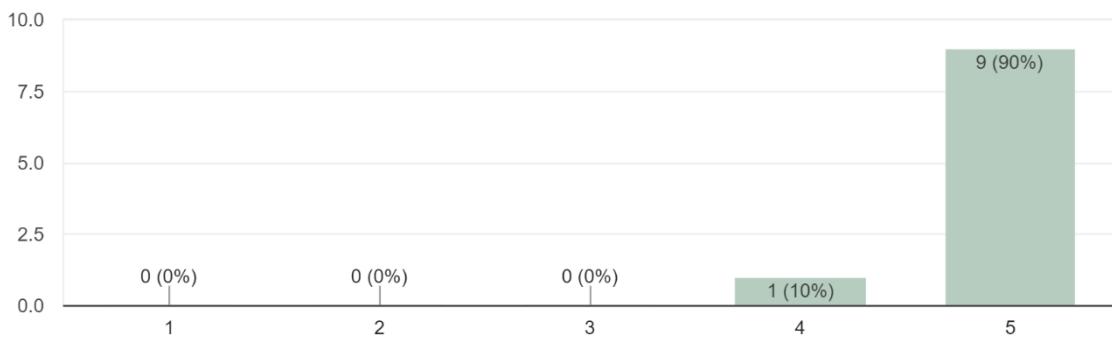
10 responses



I felt very confident using the system.

Copy

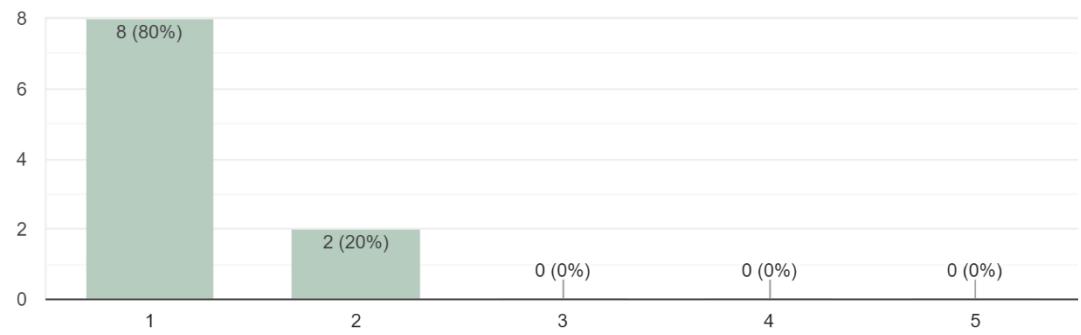
10 responses



I needed to learn a lot of things before I could get going with this system.

Copy

10 responses



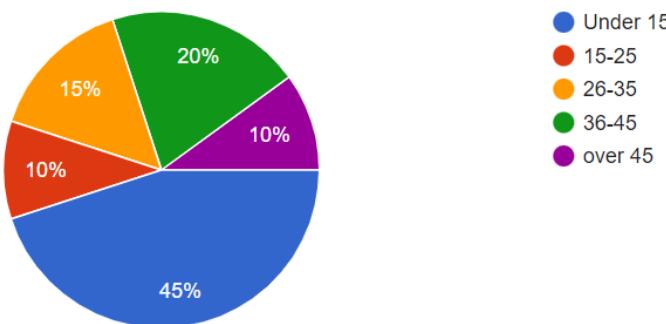
Admin questionnaire:

<https://docs.google.com/forms/d/1ohW7Au0IqlNqgrSHXsPDHTCPX69vEucRsseyQnsQlk/>

Demographics of Participants for reader:

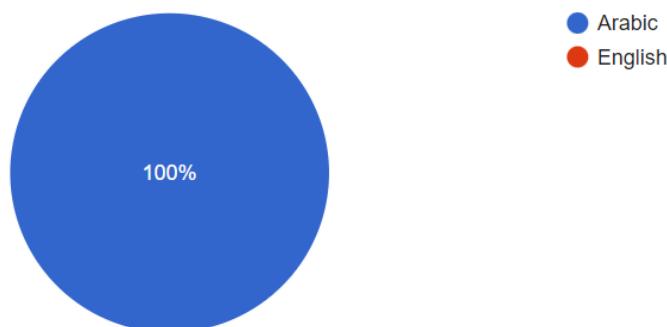
What is your age?

20 responses



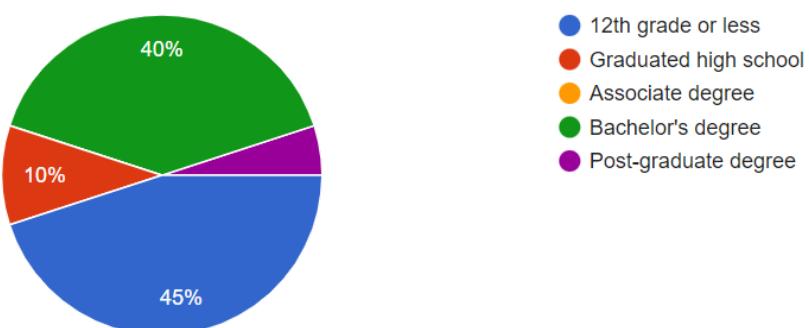
What is your native language?

20 responses



What is the highest degree or level of education you have completed?

20 responses

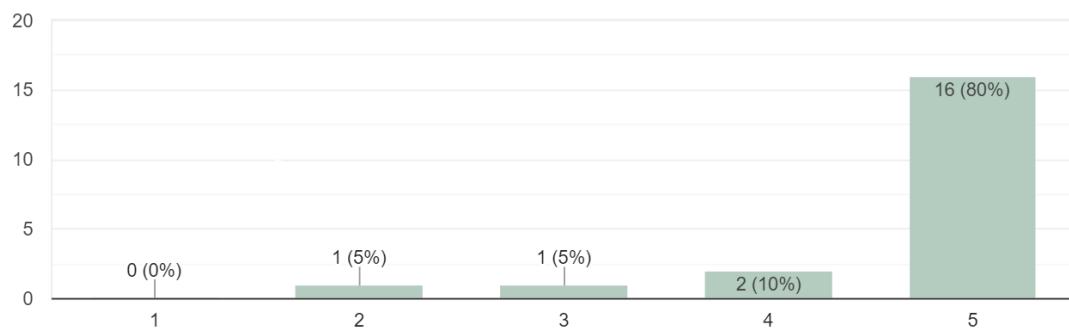


Questionnaire/Interview Results for reader:

Familiarity with ios devices

Copy

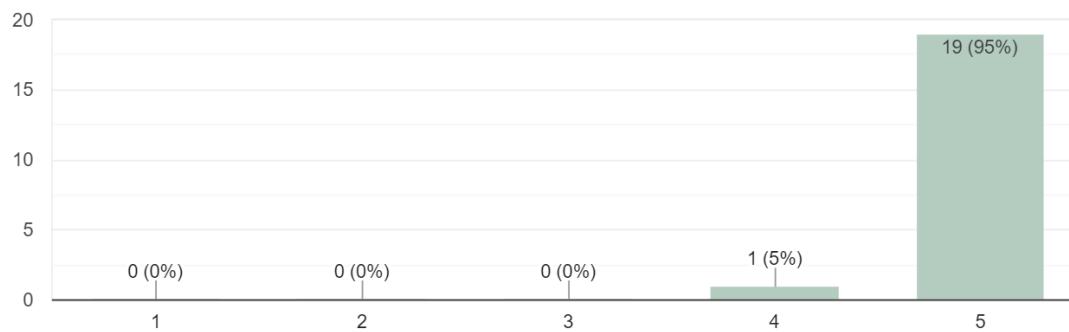
20 responses



Stories were displayed quickly when clicking on the moral

Copy

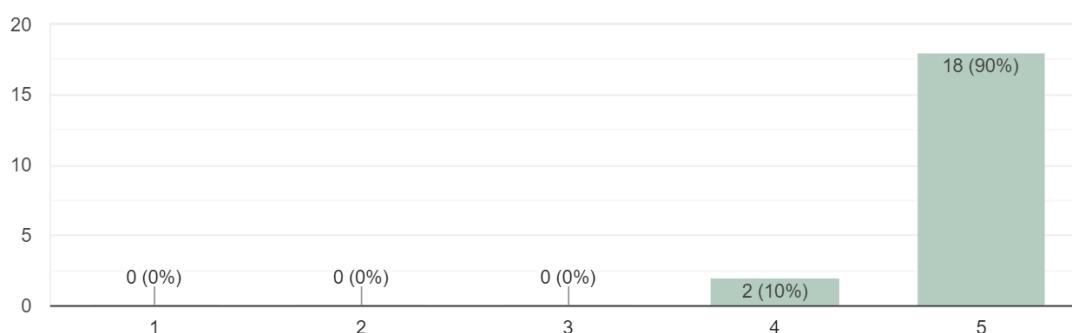
20 responses



The icons found in the application are well known and indicate what it does.

Copy

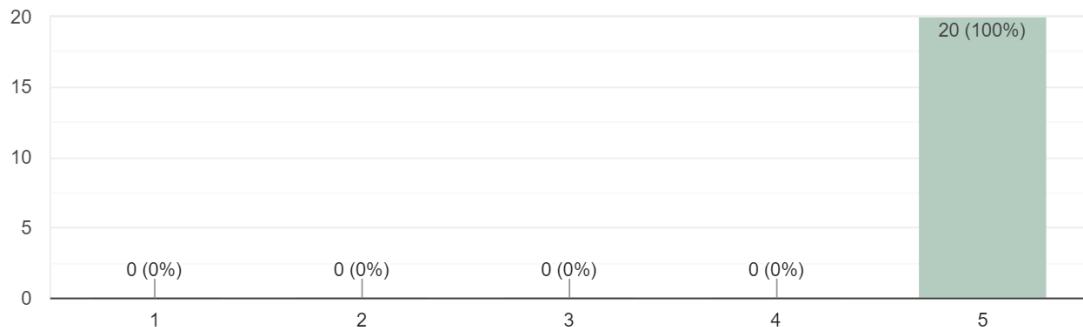
20 responses



I found the error messages in the application are clear and indicate the details of the error.

 Copy

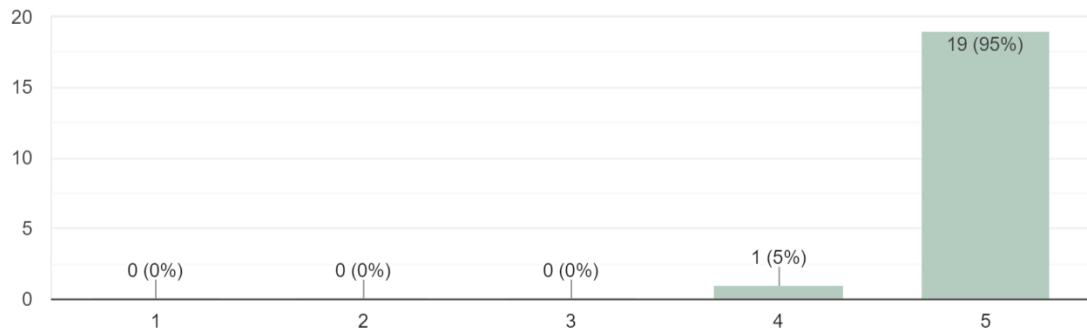
20 responses



User-friendly interface design.

 Copy

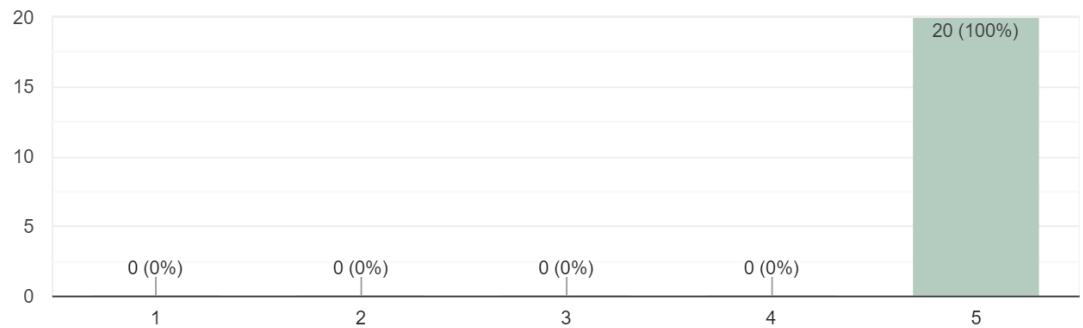
20 responses



The colors are comfortable and not distracting.

 Copy

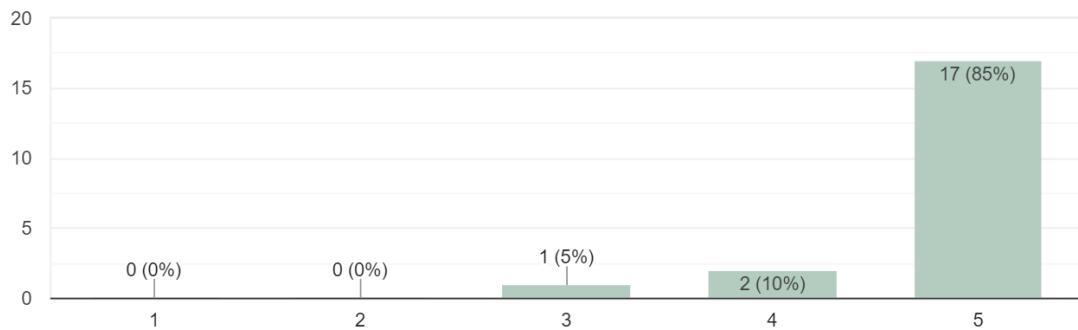
20 responses



The app used an appropriate font type

 Copy

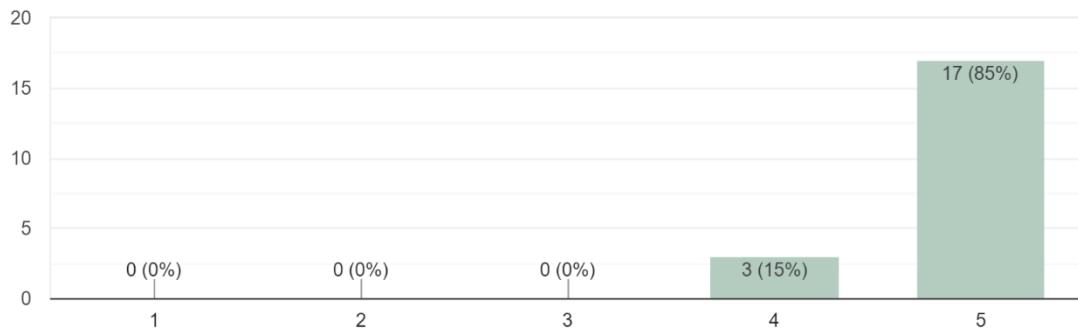
20 responses



I think that I would like to use this system frequently.

 Copy

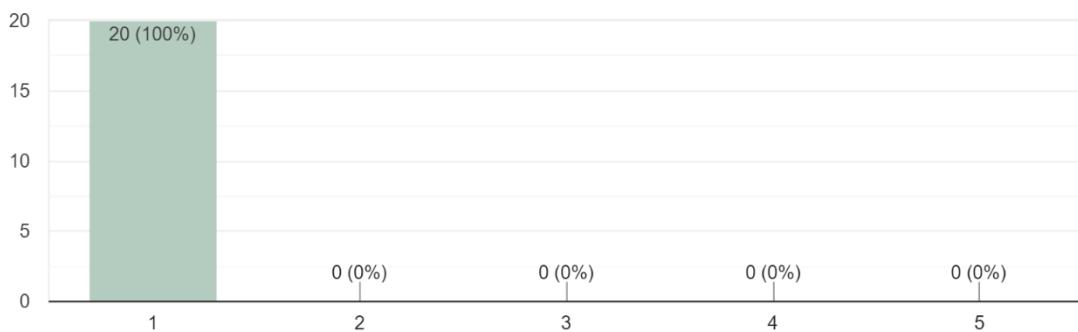
20 responses



I found the system unnecessarily complex.

 Copy

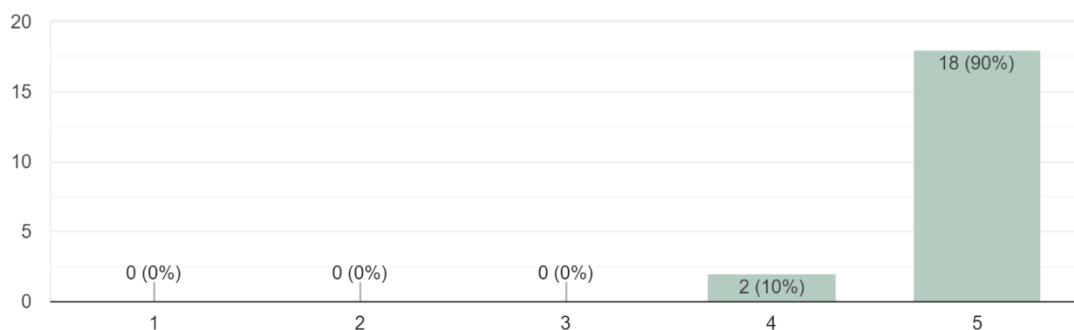
20 responses



I thought the system was easy to use.

 Copy

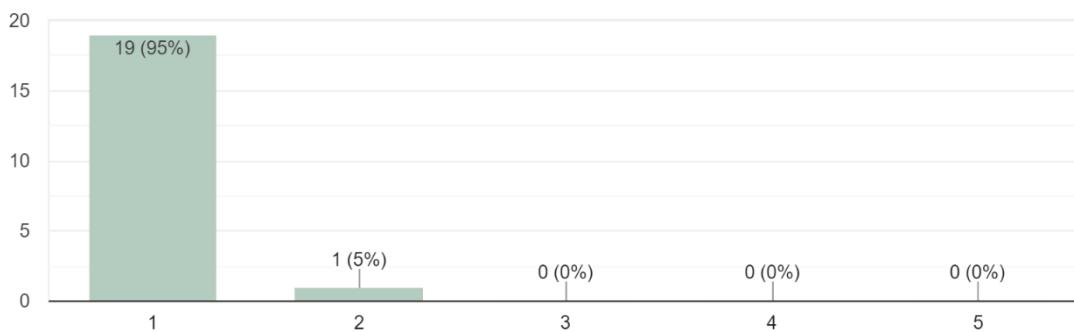
20 responses



I think that I would need the support of a technical person to be able to use this system.

 Copy

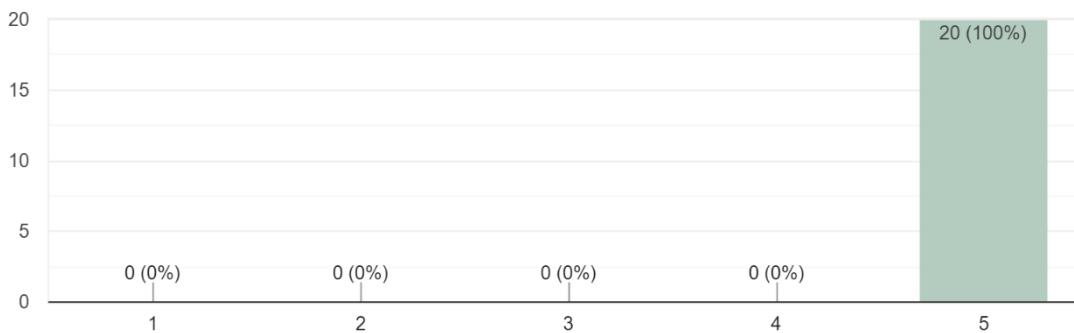
20 responses



I found the various functions in this system were well integrated.

 Copy

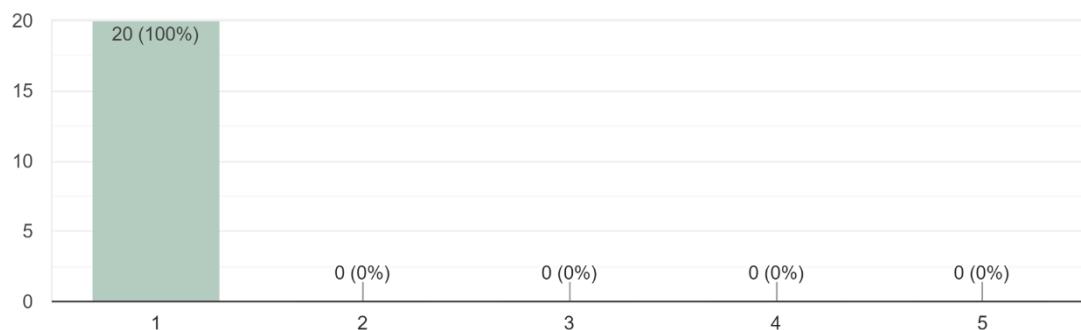
20 responses



I thought there was too much inconsistency in this system.

 Copy

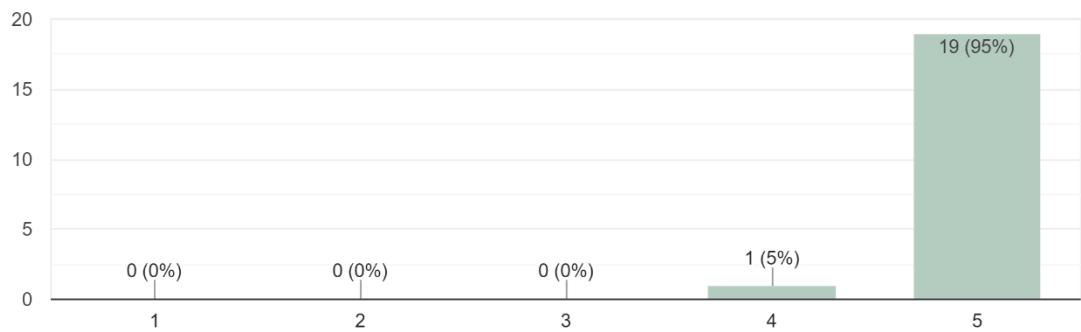
20 responses



I would imagine that most people would learn to use this system very quickly.

 Copy

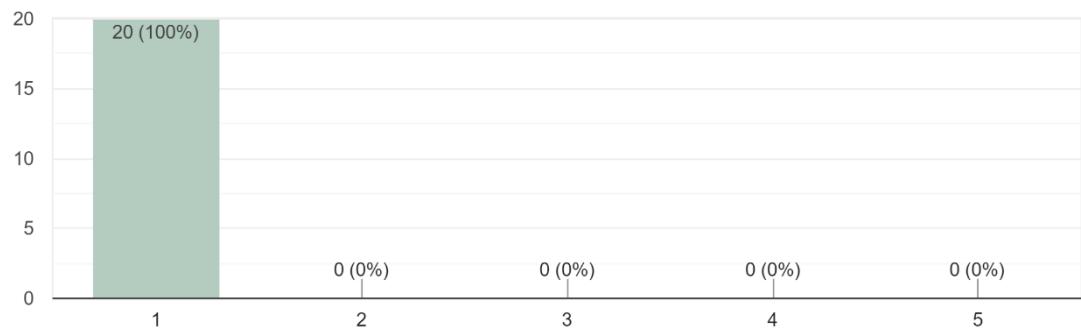
20 responses



I found the system very cumbersome to use.

 Copy

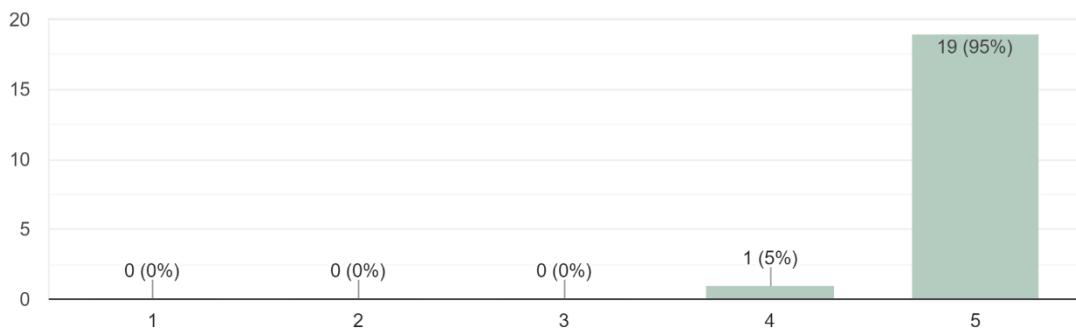
20 responses



I felt very confident using the system.

Copy

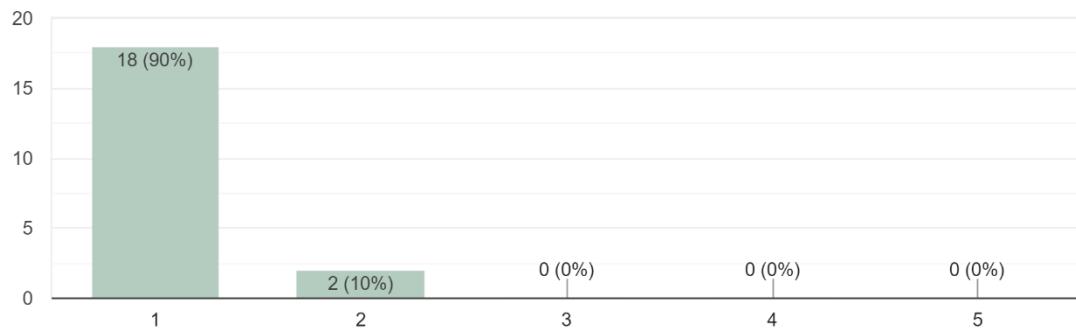
20 responses



I needed to learn a lot of things before I could get going with this system.

Copy

20 responses



Reader questionnaire:

<https://docs.google.com/forms/d/1fnEYTuanCbfhUISAw3rLmx4ySEu1KWMQj6sWns-r574/>



9.3 Appendix C: Tasks for admin and reader

Tasks for the admin:

1. Login with:

-Email: test@gmail.com

-Password: 123456

2. Add story page containing the following tasks:

2.1. Fill the form with a story from this site (<https://read.gov/aesop/081.html>) called (The Dog in the Manger).

2.2. Classify story.

2.3. Disapprove moral of the story.

2.4. Fill the form again with a story from this site (<https://read.gov/aesop/102.html>) called (Mercury & the Woodman).

2.5. Classify story.

2.6. Add story.

3. Edit story information page contained the following tasks:

3.1. Filter the stories according to moral.

3.2. Search for story.

3.3. Reclassify story content.

3.4. Edit story fields.

3.5. Delete story.

4. Logout.



Tasks for the user(reader):

1. Sign up.
2. Home page contained the following tasks:
 - 2.1. Choose a moral.
3. List of stories page contained the following tasks:
 - 3.1. Choose a story.
4. Page of a story contained the following tasks:
 - 4.1. choose male or female voice then listen to the story.
 - 4.1.1. pause the voice.
- 4.2. Add the story to the favourites.
- 4.3. Bookmark any page you stop at .
5. Go to favourites list and remove the story from the favourites.
6. Go to the home page and open the story from the continue reading list.
 - 6.1 Remove bookmark.
5. Logout.