AMERICAN UNIVERSITY OF MADABA

Department of Data Science and Artificial Intelligence
Faculty of Information Technology

Emoona: Sentiment Analysis of Mental Health and Wellness Application

Ghazal Almidana Zaid Eltabatebi

Project Supervisor: Dr.Iman Abu Hashish

Academic Year 2024/2025

Abstract

This project explores emotion classification using the GoEmotions dataset leveraging modern language processing techniques and deep learning models. The main goal is to predict human emotions from text inputs. Following an object-oriented approach ensuring scalability, maintainability, and modularity across all stages of the pipeline including data loading preprocessing, model training and evaluation. RoBERTa is implemented and compared to assess performance and generalization, the data set is analyzed using both the original 27 label setup and custom groupings for practical deployment the trained model is integrated into a flatter-based mobile application offering real-time and emotion insights from user generated journal entries. This documentation presents the methodologies, system architecture, implementation details and evaluation metrics used throughout the project.

Table of Contents

Abstract …… i
Contents …… ii
List of Tables …… iii
List of Figures ……..iv

Chapter 1 Introduction …… 1
1.1 Problem Definition……... 2
1.2 Current and Existing System ……2
1.3 Objectives ……2
1.4 Stakeholders ……..3
1.5 Project Scope and Limitations ……..4

1.5.1 Project Scope ……4
1.5.2 Limitations ……4

Chapter 2 Literature Review ……5
2.1 The GoEmotion Dataset …… 5
2.2 RoBERTa for Sentiment Analysis …….. 5
2.3 Gaps and Experiments …… 6

Chapter 3 Methodology …… 8
3.1 Overview …… 8
3.2 Dataset Description …… 8
3.3 Model: RoBERTa (Theory) .....……8
3.4 Summary of Methodology Pipeline …… 9

Chapter 4 Implementation …… 10
4.1 Dataset Description …… 10
4.2 Dataset Preprocessing ..…… 10
4.2.1 Data Cleaning ..…… 10
4.2.2 Emotion Grouping ..…… 11
4.3 Exploratory Data Analysis (EDA) and Clustering …… 12
4.4 Model Training …….. 13
4.4.1 Model Comparison …… 14

Chapter 5 Proof of Concept …… 16
5.1 Requirement and Analysis …… 16
5.1.1 Information Gathering Techniques ……..…..16
5.1.2 Requirement specification …… 16
5.1.3 Use Case Specifications…… 17
5.1.4 Domain Object …….. 19

5.2 System Design …….. 20
5.3 System Implementation ……. 32
5.4 Deployment …….…….. 32

Chapter 6 Conclusions and Future Work …….…….. 33
6.1 Conclusion …….…….…….…….. 33
6.2 Future Work …….…….. 34

References …….……..…….…..… 36

List of Tables

Table 1: Ekman's theory based categories of the dataset. …… 11
Table 2: Final evaluation results of training RoBERTa on GoEMotions dataset ……. 14
Table 3: Grouped GoEmotions evaluation results ……..….. 15
Table 4: The original GoEmotions 27-label (No Neutral) Evaluation results ……. 15
Table 5: Final evaluation results of training RoBERTa of the GoEMotions dataset …… 15
Table 6: Functional Requirements (IEEE Format) …….. 16
Table 7: Non-Functional Requirements (IEEE Format) …….……. 17

List of Figures

Figure 1: Flowchart of the Pipeline ……..…… 9
Figure 2: Distribution of the 27 Label GoEmotions …… 11
Figure 3: Distribution of the 27 Label GoEmotions >= 1 Emotion ……. 11
Figure 4: Word Cloud highlighting that happy categories are dominant ……...….. 12
Figure 5: Pie chart of the Ekman's based categories ……..……. 13
Figure 6: Heatmap of the Categories Distribution across clusters …..……..…… 13
Figure 7: Line plot demonstrating the generalisation gap ……..…. 14
Figure 8: Domain object diagram ……..…….……. 19
Figure 9: Use Case Diagram ……..…….……. 20
Figure 10: Logging In Activity Diagram ……..…….…… 21
Figure 11: Journal Entry Activity Diagram ……..…….…….. 21
Figure 12: Logging In Activity Diagram ……..…….…… 22
Figure 13: Journal Entry Activity Diagram ……..…….…….. 22
Figure 14: Sequence Diagram of Logging In ……..……. 23
Figure 15: Sequence Diagram of Starting the App ……. 24
Figure 16: Sequence Diagram of Logging In ……. 25
Figure 17: Sequence Diagram of Starting the App ……26
Figure 18: Class Diagram Representing the App. ……. 27
Figure 19: Component Diagram of the System ……..……28
Figure 20: Deployment of the System Diagram ……..…….…. 29
Figure 21: The log of all previous journal entries …….… 29
Figure 22: Prediction results of a happy sentiment ……..……. 30
Figure 23: Results of summary of different Predictions ……..……. 30
Figure 24: Summary of the latest productions ……30
Figure 25: Dashboard Visualization ……..…….……30
Figure 26: Login Screen ……..……..……. 30
Figure 27: Sign Up Screen ……..…….…….……. 30
Figure 28: Dashboard Screen ……. 31
Figure 29: Journal Entry Screen ……. 31
Figure 30: Mood History Screen ……...……. 31

Chapter 1
Introduction

Natural Language Processing (NLP) is a field within artificial intelligence, enabling machines to understand, interpret, and generate human language. It combines machine learning along with linguistics to give deeper understanding of context and semantics which allow machines to process natural everyday text. In the context of this project NLP is essential to understand the journal entries and get the meaning behind the words so the machine can deal with it accordingly.

The sentiment analysis which is a subset of NLP focuses on identifying the emotional tone behind the text. Emotion classification takes sentiment analysis further by classifying into more meaningful labels as we implemented in our project here emotion classification is able to classify based on categories such as *happiness, sadness, anger, and surprise.* The system at hand gives the basic polarity a push forward to be more focused and relevant especially when dealing with a broad topic involving mental health and self-reflection journey.

The core of the project is the study of fine-grained emotions and how do they benefit the mental health domain by integrating fine-tuned machine learning model into a flutter-based mobile application that can predict from user's input journal to provide a reliable and relatable classification of the emotion behind the body of the text. This system goes beyond basic sentiment detection by leveraging a psychologically grounded emotion taxonomy (Ekman's theory) and a robust dataset (GoEmotions by Google) delivering insightful and precise emotional classification.

The *emoona* application provides a journaling interface that integrate NLP techniques, this present the user user with the ability to a daily input processed by the system and returns a predication from the fine-tuned RoBERTa-based deep learning model trained on the GoEmotions dataset, The app identify the emotions for each journal entry then visualise the results and trends over time as well as storing all entries with its predication to take the user on a smooth and helpful journaling and self discovery journey powered by AI-driven insights.

1.1 Problem Definition

While there is a growing interest in using AI and NLP for mental health support, many existing mental wellness applications rely on simple logging mechanisms such as emoji selection or basic text entries without any meaningful analysis. These systems do not process user input to provide feedback, trends, or emotional predictions. More importantly, they often fail to recognize subtle emotions or emotional combinations due to the absence of intelligent classification models.

The GoEmotions dataset released by Google provides a comprehensive collection of 58,000 Reddit comments labeled with 27 fine-grained emotions plus a neutral class. However, two major issues arise with using this dataset:
Imbalance: Some emotions like ”grief” and ”embarrassment” appear in less than 1% of samples, while emotions like ”approval” and ”admiration” dominate the dataset.
Overrepresentation of Neutral: The neutral class acts as a default label in ambiguous or unclear contexts, leading to models over-predicting neutrality and reducing sensitivity to actual emotional content.
Additionally, previous studies have shown that many models trained on this dataset fail to generalize or suffer from poor performance on rare labels. These challenges highlight the need for a more balanced, psychologically grounded, and technically optimized approach to emotion classification.

1.2 Current and Existing Systems

We have many variations that follow the same primary concepts of our idea some of the current landscape of emotion tracking tools includes:
• Manual entry applications: Apps like Daylio or Moodpath allow users to log moods via sliders or emojis, with no deep emotion recognition.
• Chatbots: Some use basic sentiment analysis (positive/negative/neutral) but lack multi-emotion classification or personalization.
• Journaling apps: Platforms such as Reflectly offer daily prompts but analyze content minimally.
Academic models like Bi-LSTM, BERT, and RoBERTa have shown progress in text classification and sentiment analysis. However, most of these systems were not tailored to psychological emotion frameworks like Ekman’s theory or optimized to handle highly imbalanced datasets like GoEmotions and do not integrate real-world applications for end-users. Our project addresses these gaps by restructuring the dataset using Ekman’s theory (Happy, Sad, Angry, Surprised), removing neutral, validating groupings through co-occurrence clustering, and building an app that visualizes predictions and supports journaling.

1.3 Objectives

The primary objective of this project is to build an AI-driven sentiment analysis system that classifies text entries into one of four emotion categories based on Ekman’s psychological theory. The system should function reliably within a mobile application for emotional self-assessment.
1. Efficient and accurate pattern recognition of the user's mental health.
2. Valid recommendations based on the user state from the system to help the users in their mental health journey.
3. Fully developed application that is both smooth and helpful to use.

These objectives were introduced first in the proposal of this project and moving on to Specific objectives included the following:

Dataset Processing: Cleaning the GoEmotions dataset by dropping irrelevant metadata, removing duplicates, and filtering out ”neutral” entries for a better balance of the data.
Emotion Grouping: Map 27 individual emotions into four psychologically validated groups: Happy, Sad, Angry, and Surprised. For a precise emotion classification,
Exploratory Data Analysis: Two parts of visualization were conducted distributions, top/bottom emotions and co occurrence of emotions were to understand the data primarily and cluster relationships to guide grouping. Then for the grouping results we explored the data by pie chart, heat map, cluster grouping visuals.
Model Development: Implement and fine-tune the RoBERTa model using the Hugging Face library transformers for multi-class single-label emotion classification.
Model Evaluation: Compare performance metrics (Accuracy, F1 Macro, Recall, Precision) across BERT, RoBERTa, and LSTM models under the same training setup.

Application Integration: Build a Flutter app that allows users to journal emotions, receive real-time predictions, and view their mood history graph.
Backend Implementation: Host the model using Render with Firebase as a database for user entries and predictions.

1.4 Stakeholders

The stakeholders included in this project are categorized into groups some of which are impacted directly by the project And some influence the project. We have primary, secondary and tertiary stakeholders.

Primary Stakeholders

End-users, who are the direct users of our journaling app , would use the regular journal entries which helps them reflect on their emotions and well-being and they are able to do so by our main functionality emotion classification and the helping visualization that captures the trends overtime.

Secondary Stakeholders

Mental Health Professionals like Psychologists, therapists, and counselors who may use emotional trend data to support diagnosis.

Data Scientists researchers studying emotion classification and NLP who can be built on our own methodology.

Tertiary Stakeholders

Academic Supervisor Including Dr. Iman Abu Hashish who oversees the project direction and scientific rigor.

Project Team Zaid Eltabatebi and Ghazal Almidana, leading the analysis, model training, interface design, and deployment.

Platform Providers The third party services hugging face which provided Transformers Library, Firebase the store for all input by user, and render where we have our model hosted.

1.5 Project Scope and Limitations

1.5.1 Project Scope

The project focus on building a mobile application that enhances the user experience by integrating journaling feature with the capabilities of the natural language processing featured cores of the system included:

- Allowing users to log personal journal entries securely.
- Analyzing entries by the classification model RoBERTa-based for understanding the meaning behind a body of a text.
- Based on predictions the results are Illustrated on a mood tracking line chart overtime.
- Firebase to store journal entries along with the Productions and keeping the user data safe and private
- Presenting recent data through a user friendly dashboard

1.5.2 Limitations

While the application provides advanced NLP techniques and machine learning that offer users a unique experience, that aids in self-reflection and emotional awareness there are certain limitations:

Data Source Bias: Reddit comments may not represent all demographics or psychological profiles.
Emotion Simplification: Using only four emotion groups reduces granularity and ignores complex mixed emotions.
Single-Modality: No use of speech, image, or physiological signals which could enhance emotion detection.
Latency: Real-time prediction depends on model hosting and device speed; delays may occur.
Generalization: The model was not fine-tuned with domain-specific pretraining (e.g., health forums or therapy transcripts), which might limit real-world generalization. Despite these limitations, the project represents a comprehensive approach to intelligent sentiment analysis for health and wellness, offering valuable insights for both users and professionals. The methodology, results, and implementation aim to contribute meaningfully to the growing field of emotion-aware artificial intelligence.

Chapter 2
Literature Review

2.1 The GoEmotion Dataset

Sentiment Analysis has developed immensely and with the help of Natural language processing - NLP tasks with multiple applications across various domains mental health chatbots, speech recognition and machine translation. Such applications gave a wider new range of approaches to integrate the NLB tasks in elaborated applications and one of the most important aspect is mental health that would always consider new ways that can reach easily and more pleasantly to people and when this is the orientation we are going we are in need of a constant improvements.

Ekman's theory (Ekman,1992), It is the psychological interpretation which state that of the universal emotions we have six grouped categories that combine all main emotions and define all the other subset emotions : joy, sadness, anger, fear, surprise and disgust.

The GoEmotions dataset provided by Demszky et al. (2020), Made substantial progress in this field by introducing a fine-grained and curated annotation for Over 58,000 reddit comments across 27 emotion labels or neutral, In spite of the data being the largest till this day a sitback in this dataset that it is biased as reddit users are mostly young males and this take from the diversity we see in other dataset that addresses emotions. Also it is highly imbalanced, Frequencies are highly skewed: positive states such as joy and approval are far more than 20% of the annotations where in other states like grief and embarrassment they represent < 1% And In this case the models would predict the common labels and miss rare ones or even ignore them completely, to Tackle this issue Wang et al.(2024) experimented with the GoEmotions dataset to train it on the multi label 27 emotions, group taxonomy by clustering three categories, and neutral For the last training they experimented with a group taxonomy following Ekman's Theory in addition to neutral leading to F1 score of 0.51 for RoBERTa.

But we need a better approach and higher evaluation scores therefore to deal with the imbalance in the dataset, following Ekman’s Theory The present work maps GoEmotions into four Ekman-based categories (dropping neutral). In Maheshwari & Varma (2022) Follow the same approach as Kaipeng et al.(2024) for handling the data but the difference lays where in this paper the model was Fine Tuned on a subset of GoEmotions as they included sentences that have only one of the seven emotions that matches the main data that their research was upon which is the Ekman’s theory categories to then further fine-tuning on other dataset that resulted poorly testing on essay while.

2.2 RoBERTa for Sentiment Analysis

Previous work on GoEmotions dataset rely on feature based and neural models. Integrating Bi-LSTM with Max pooling achieved the macro F1 of 0.46 Demszky et al. (2020) even though it out performed traditional TF-IDF + SVM baselines still it was struggling with minority emotions as these models are limited context and size wise. As for BERT, a language model developed in 2018 by Devlin et al., it's a breakthrough in understanding

language for emotion prediction in NLP tasks as it was built on transformers structure that was introduced by Vaswani et al., 2017 which replaced the use of recurrent networks(RNNs, LSTMs) and training BERT got a result of 0.64 for F1 Macro. And a common thing on Reddit its slang vocabulary and this gives poor results when evaluating models.
The model BERT affected many experiments since its release as well as RoBERTa which is an enhanced model based on BERT. And we can see the results of that, Maheshwari & Varma (2022) Approach the 27-label dataset and fine toned BERT-base and RobERTa-base on the original data, the next step was combining the models with BART-large. The evaluation test was done on essay data introduced by the workshop. Returning result of 0.585 for F1 this result was ranked top one on the leaderboard at the time of the shared task workshop (WASSA) competition. Even Though the test wasn't done directly on GoEmotions yet this is an example demonstrating that context limitation in transformers based models is much lower and therefore we see the development in the results and how RoBERTa and GoEmotions data affect the emotion classification as for comparison on the dev set solutions where RoBERTa and BART is used (F1-0.63) outperformed other solutions.
Optimizing RoBERTa by fine tuning through transfer learning on domain specific data pushed the model further. Wang, K., Jing, Z., Su, Y., & Han, Y. (2024) pursued pre-training RoBERTa-large on GoEmotions before the transfer learning, following the steps of the original paper Demzsky et al. data hierarchy wise later on fine tuned BERT on these three taxonomies the relevant result for our project is the Ekman's theory based and neutral which got results of 0.62 F1 Macro and for RoBERTa which is unusual a Macro F1 score of 0.51 making BERT outperform RoBERTa which lead to rejecting their previous hypothesis of the latter model being the superior and the justification for the model behaviour is task sensitivity in machine learning, suggesting BERT is the better model in this case. However, the justification depends solely on "task sensitivity" might not be accurate due to other factors such as different preprocessing or training strategy or model generalisation capabilities play a higher role in this case and can define more the reasons behind the lower results of RoBERTa. The results and experiments of our own are presented in section 3.

2.3 Gaps and Experiments

Discuss research papers on emotion classification using goemotions data set followed Eckman sex basic emotions framework(sadness, anger, Joy, fear, disgust, and surprise), as well as including neutral category studies such as- adopted this taxonomy and evaluating models like BERT and RoBERTa their results suggested that BERT outperform RoBERTa the this exact setup which they reported to task sensitivity.
However, in our work we observe the differences in that handling and the model. to improve emotional Clarity and label balance even more than the suggested taxonomy which is Ekman's theory. We categorized the data into Ekman's based four categories(happy, sad, surprise, and angry). Alongside these categories one of the biggest factors of the data imbalance is the neutral column and in our case there's no use for it so we dropped the neutral category. Based on the observations that the value the neutral category represents is very little and could lead to the model to have a category to fall back on and instead of understanding the data and categorize it based on the emotions in each group The model

would predict neutral every time there is a missing context or extra words that was already described as neutral. After applying our custom mapping on the data set we ensured consistency by training only on single-label rows.
Papers Wang et al. (2024) using BERT depends on hugging face Transformers but the key difference here we applied a minimum text length threshold of 248 characters and this was one of the fine tuning steps because we started with 128 and we see a noticeable difference when replacing to 248, and for RoBERTa It makes a difference because this help the model to retain more semantically rich and expressive text which improves our model.
Training for RoBERTa-base we fine-tuned the text using hugging face transformers library with no prior domain-specific pretraining. Also we implemented a custom train/test split using scikit-learn to reflect our data set restructuring. As for the hypermeters We experimented with different epochs with a prior stopping to reach the best result it was 2 epochs and a patch size of 16 with a learning rate of 5e-5 Which is a similar environment of the original paper Demzsky et al. (2020).
for the evaluation Our primary focused was on accuracy for a result of 72.6% as appropriate for single label classification but to have a fair comparison and adjust the model based on previous experiments from other papers we also considered the F1 Macro score of 0.64 which aligns closely and sometimes even exceeds some prior benchmarks.
The experiments we did wasn't solely focused on RoBERTa as we experimented with BERT and LSTM to prove that Roberta does have a leverage as an enhanced model from BERT and contrary to Wang et al. (2024) RoBERTa outperformed BERT and the appropriate justification is restructuring the data and preprocessing in a different pipeline - indicating that conclusion of the model results may not be a suitable generalized across variations of the GoEmotion task.

Chapter 3
Methodology

3.1 Overview

The proposed methodology in this project combines deep learning and natural language processing techniques to detect and classify emotions from text-based user input. The approach begins with robust data preprocessing, followed by exploratory data analysis, emotion grouping, and ends with a fine-tuned RoBERTa model trained to classify input into one of four core emotion groups: Happy, Sad, Angry, and Surprised. This methodology was driven by both empirical evidence and psychological theory (specifically Ekman's model of emotions), ensuring both technical rigor and emotional validity. Added Insight: Our approach is built with the real-world application in mind, especially in contexts such as wellness tracking, virtual therapy assistants, and emotional AI systems. Hence, we focused not only on achieving technical accuracy but also on ensuring the emotional classes we predict are psychologically meaningful and interpretable.

3.2 Dataset Description

We used the GoEmotions dataset, an emotion-labeled corpus from Reddit comments annotated with 27 fine-grained emotions. It includes over 58,000 samples, making it one of the most comprehensive emotion datasets for the English language. Initially, many texts had multi-label annotations, where a single text could have more than one emotion.

One of the key strengths of GoEmotions is that it captures a broad range of everyday emotions expressed in natural language, including slang, sarcasm, short comments, and emotionally charged sentences. This diversity allows models trained on it to perform well in unfiltered, real-world environments such as social media or user diaries in health apps.

3.3 Model: RoBERTa (Theory)

After the BERT model in Liu et al.(2019) RoBERTa was introduced as an enhanced model with features that outperform BERT by removing and modifying key elements of BERT training procedure the model is trained on 10x more data than Bert, removes the no next sentence prediction (NSP) which gives the data more determination in some tasks. And it increases compute, data, and training time that yields large gains. We explored the model even more and turned out to be the number one choice for emotion classification tasks like the one at hand.

3.4 Summary of Methodology Pipeline

• Data Cleaning & Filtering (removed neutral and duplicates).
• Emotion Grouping (based on semantic similarity + psychological theory).
• EDA & Clustering (to validate grouping) • Tokenizer Preprocessing (RoBERTa tokenizer).
• Model Comparison (RoBERTa vs BERT vs LSTM).
• Final Training & Evaluation (2 epochs, optimal results).
• Mental health application (Deploying the model, flutter).

This methodology ensures that the resulting model is not just accurate, but also practical, explainable, and adaptable for future real-world emotion-based AI systems.

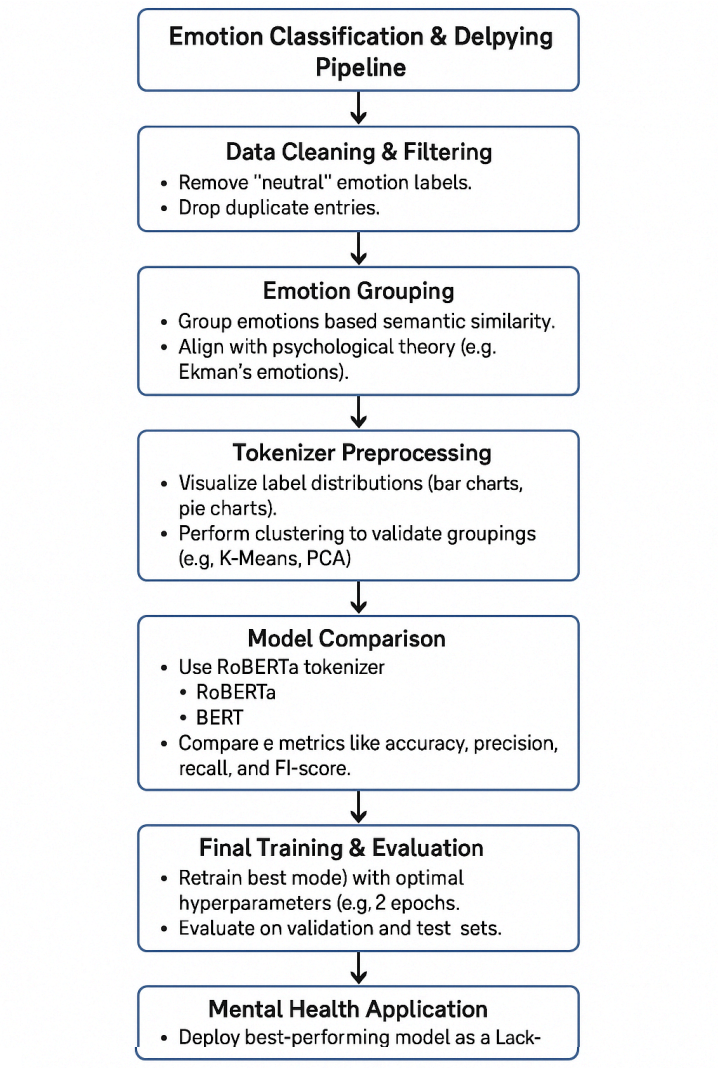


Figure 1: Flowchart of the Pipeline

Chapter 4
Implementation

4.1 Dataset Description

The original GoEmotions dataset was imported and preprocessed using pandas; originally the dataset included 58,000 records with over 37 features. The data was structured in a way that each text would have one or more emotions identified by 0 or 1 including neutral columns. The data included 27 emotions and neutral. Our main focus was the emotion for each text. After primary exploration we used matplotlib to visualise the original emotions (see Figure 2). Exploring the data made the high imbalance clearer now with neutral taking 15,000 of the records and emotions like grief have less than 300 records of the dataset which would be handled as mentioned in section (4.2).

4.2 Dataset Preprocessing

4.2.1 Data Cleaning

To make the dataset more usable and improve model performance all non-textual and any extra columns that won't add any value to the model training like (e.g., author, subreddit, link id). We dropped these columns using pandas library keeping the column example very unclear for exploration purposes, As well as all rows labeled as ”neutral” were dropped. The neutral class had disproportionately high frequency, introducing imbalance and making it difficult for the model to distinguish nuanced emotional signals. During our initial experiments, including neutral led to accuracy drops and underfitting. After removing neutral The dataset became more focused and emotionally rich. Accuracy improved significantly. Class balance was better distributed across emotion groups.

Additionally, we observed that many ”neutral” labels were assigned to unclear, vague, or emotionless sentences. Keeping them caused the model to generalize poorly, misclassifying clear emotional inputs as neutral. By removing them, the model was forced to focus on discriminating real emotional patterns. After that we dropped duplicated records from the data as well and other than the imbalance of the dataset the data is mostly clean.

Output of the data is 61,297 with 29 columns till this step, next we explore the co–occurrent emotions (see Figure 3) and that shows we have a values with 0 emotions related to them therefore were dropped after clarifying they are under the *example very unclear* category which was made by the original annotaters.

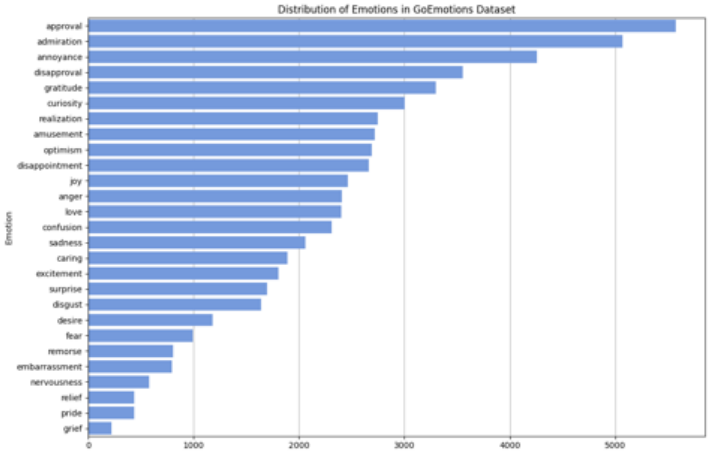


Figure 2: Distribution of the 27 Label GoEmotions

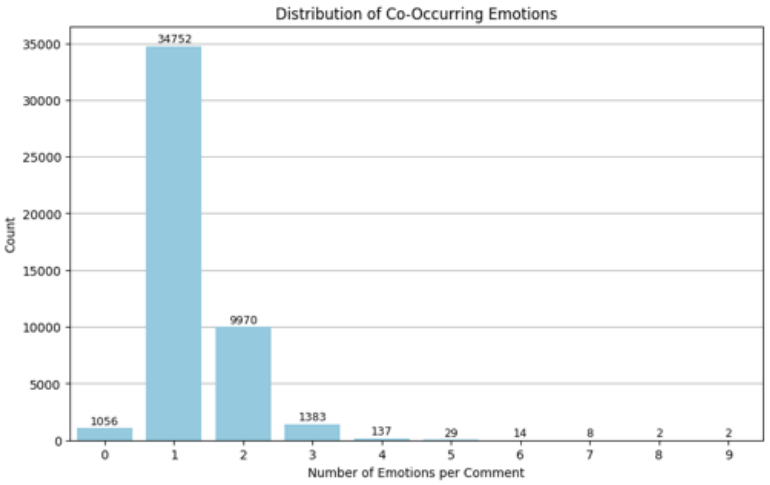


Figure 3: Distribution of the 27 Label GoEmotions >= 1 Emotion

For a clean text we process the text by RobertaTokenizer. The tokenizer uses subword encoding, that is provided by hugging face transformers library which is a very common tool to use with training transformers based models like RoBERTa. Tokenize each comment, Pad/truncate sequences to 248 tokens which also was the optimal length for the data any increase or decrease returned a lower performance, and Prepare attention masks This step produced consistent inputs for the transformer model, ensuring smooth training.
These features put the performance of it ahead of other tokenizers that don't typically deal with complexity as well, in addition its optimized to the RoBERTa model so it suits it well.

4.2.2 Emotion Grouping

Ekman's theory is a followed procedure for the GoEmotions dataset but in an attempt for better results when evaluating the model, we followed the theory in addition to the co-occurrent records to group the 27 emotions into **four** different categories: Happy, Sad, Surprised, and Angry. This grouping fixed the high imbalance problem and also since our application for the model is a mental health context for the text so we decided to drop the **neutral** category because for this context it isn't relevant and for mental health even with broad categories we need specific emotions therefore we custom mapped the data using python dictionary as follows (see Table 1).

Table 1: Ekman's theory based categories of the dataset.

Emotion Group	**Sub-Emotions**
Happy	admiration, amusement, desire,approval,caring,excitement, gratitude, joy, love, optimism, pride, relief.
Sad	disappointment, grief, nervousness, sadness, remorse, embarrassment.
Angry	anger, disapproval, disgust, fear, annoyance.
Surprised	curiosity, surprise, realization, confusion.

To validate the semantic coherence of our emotion groups, we applied KMeans clustering:

• Adjusted Rand Index (ARI): 0.118

• Normalized Mutual Info (NMI): 0.248

These metrics validated that the manually defined emotion groups are statistically distinguishable, even when clustered without labels - offering confidence in the robustness of our categories. The emotion groups were selected to be non-overlapping and psychologically grounded, ensuring that the final predictions are intuitive for human interpretation, a key requirement for mental wellness and applications.

4.3 Exploratory Data Analysis (EDA) and Clustering

To get a deeper understanding of the data and to make sure the mapping is relevant to the scope of the application we performed visualisation using matplotlib and seaborn after calculating the frequency we can see the visualisation highlighting different aspects of the data (see Figure 4).

Word Cloud of Most Common Words in Dataset
Thank
one
Love
people
good
make
know
think
feel
way
sorry
don
t
Yeah
still
got
go
now
going
man
m
guy
want
look
much
need
will
bad
time
really
well
lol
someone
even
hope
game
right
see
thing
great
work
feel
say
fuck
mean
friend
wrong
comment
hate
kid
best
made
better
post
actually
sure
probably
sound
nothing
Maybe
dude
glad
day
let
though
Oh
u
damn
happy
keep
something
never
pretty
take
Wow
re
Ye
thought
nice
find
joke
point
enjoy
back
always
weird
feeling
video
wait
world
help
give
haha
kind
stupid
terrible
play
funny
seem
believe
remember
amazing
year
two
hard
sad
without
change
lot
exactly
understand
watch
money
come
call
care
idea
looking
fucking
girl
happen
put
making
reason
teams
didn t
wish
life
person
part
everyone
definitely
interesting
awesome
tell
shit
hell
joke
new
win
terrible
another
getting
honestly
anyone
anything
place
old
problem
read
tell
little
live
least
use
dear
real
hear
start
around
guess
first
sub
saying
enough
lmao
hear
amazing
remember
two
believe
seem
funny
change
play
kind
haha
stupid
work
great
lot
exactly
understand
without
sad
take
hard
pretty
Wow
re

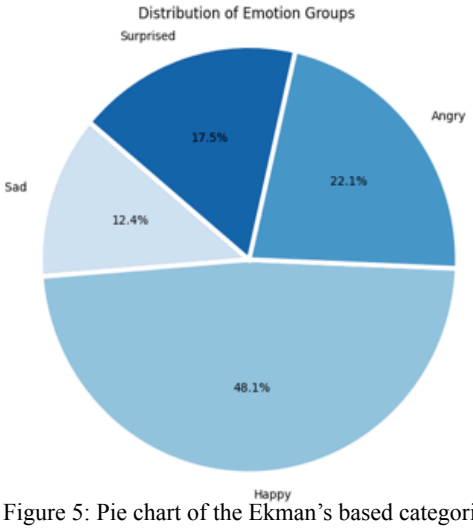


Figure 5: Pie chart of the Ekman's based categories

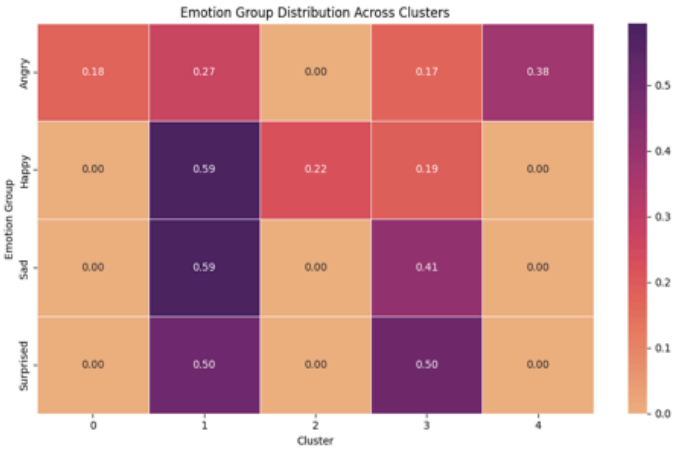


Figure 6: Heatmap of the Categories Distribution across clusters

This analysis not only helped understand the statistical distribution of the data, but also highlighted the emotional patterns and natural co-occurrence of emotions, guiding our grouping decisions and later influencing model design.

4.4 Model Training

RoBERTa was introduced as an enhanced model with features that outperform BERT, Fine tuning could be done using many ways like changing parameters, restructuring the data and changing the layers of the model. Fine tuning a model is a non-linear process and this will be mentioned in detail later on that will clarify the choice of model. The Train/Test split is 80% training and 20% validation - 32737 training and 8185 validation - which is a standard random split, it uses a fixed seed to ensure reproducibility and the ability to compare fair results.

Model base is roberta-base that is (pre-trained by Facebook AI). Optimising the data using AdamW with a learning rate of 5e-6, batch size of 16, and 2 Epochs these parameters were chosen after many experiments.

The model was trained for more than 2 Epochs and that lead to high overfitting (see Figure 7) in the training and other parameters changes affected the results of the evaluation poorly so based on the different adjustments we did this environment was the best for training RoBERTa (See Table 2) and to test the other models as well.

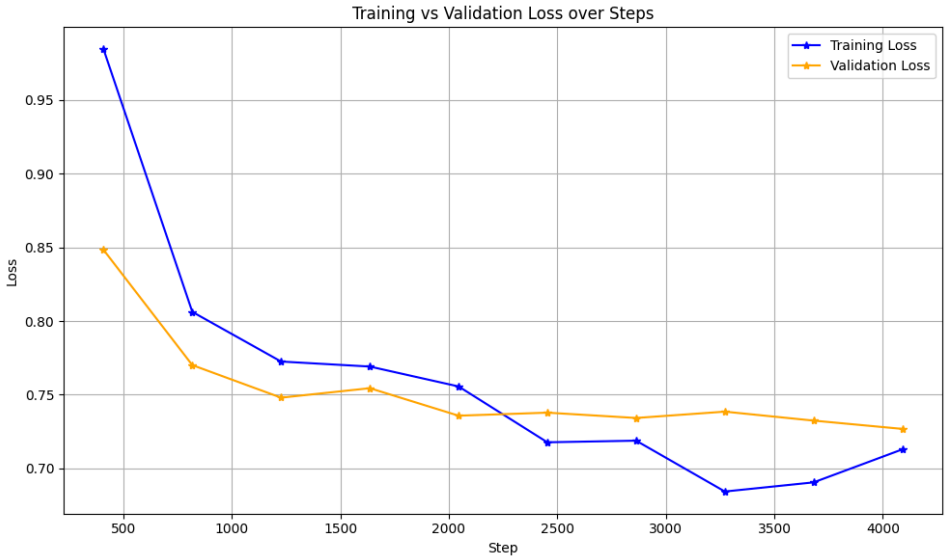


Figure 7: Line plot demonstrating the generalisation gap.

Table 2: Final evaluation results of training RoBERTa full metrics evaluation

Epoch	Training Loss	Validation Loss	Accuracy	F1-Macro	F1 Micro	Precision Macro	Recall Macro
1	0.77	0.80	0.72	0.64	0.72	0.67	0.62
2	0.77	0.80	0.72	0.64	0.72	0.66	0.63

4.4.1 Model Comparison

To make certain of the model we had chosen three models were used for comparison:

• RoBERTa – a robust transformer-based NLP model.

• BERT – a foundational NLP model.

• LSTM – a traditional RNN used for sequence modeling.

Each model was trained for 2 epochs, with a learning rate of 5e-6 and batch size of 16.

As shown in Liu et al.,(2019), RoBERTa improves on BERT by removing the Next Sentence Prediction objective and training with larger mini-batches, more data, and longer sequences making it more robust and expressive.

To Prove the choice for RoBERTa as the model for training this data we choose similar model BERT and LSTM which was used in earlier experiments on the GoEmotions Demzsky et al. (2020) both were trained in the same preprocessing of the data, tokenization and environment to ensure reliable resource and fair comparison. And this model comparison (see Table 3) proved training RoBERTa is the suitable model for the single label data.

Table 3: Grouped GoEmotions (Happy, Sad, Angry, Surprised) evaluation results.

Model	Accuracy	F1-Macro	Generalization Gap
RoBERTa	0.72	0.64	0.04
BERT	0.68	0.60	1.09
LSTM	0.52	0.17	0.001

The following training was done to justify the choice of the data restructuring as we kept the tokenization and environment the same but the preprocessing of data followed the same steps before the grouping and trained on the 27-multi label emotions. And the results for same environment performed poorly(see Table 4) compared to the Ekman's theory based categories

Table 4: The original GoEmotions 27-label (No Neutral) Evaluation results.

Model	Accuracy	F1-Macro	Generalization Gap
RoBERTa	0.28	0.36	0.02
BERT	0.23	0.31	0.01
LSTM	0.23	0.16	0.01

A loss curve was plotted to monitor training progress and training loss steadily decreased, reaching ~0.59 and for validation loss flattened, then slightly increased, indicating the beginning of overfitting after 2 epochs This confirms that the No. of epochs was optimal. Training longer leads to overfitting and reduced generalization; this was also the following parameter by Wang et al.(2024) as we mentioned.
The following results are the final results for training RoBERTa on our hierarchy; these results reflect balanced, stable performance with high sensitivity to emotional nuance.

Table 5: Final evaluation results of training RoBERTa on 4 categories of the GoEMotions dataset.

Metric	Value
Accuracy	72.67%
F1 Macro	0.646
F1 Micro	0.726
Precision Macro	0.661
Recall Macro	0.635

Chapter 5
Proof of Concept

5.1 Requirement and Analysis

5.1.1 Information Gathering Techniques

For accurate definition for the system requirements , we combined formal and informal methods. The techniques chosen were suitable for the project scope, timeframe, and target audience.

• Informal Interviews

The interviews were conducted with friends and family members who represent potential users of the system. The discussion provided insights into their expectations and recommendation upon using similar applications and the challenges they face. These users were an important part of our audience because we want an age inclusive app and at the same time focus on the students like us.

• Observation

After preparing a prototype of the application pipeline and how it can possibly be used by users. We gave the prototype to some of the expected users from the target audience and took reviews from them based on using the app.

5.1.2 Requirement specification

Functional Requirement: Following the object-oriented approach , the system's requirements describing the system functions and primary actions that are implemented for the mobile application are presented using IEEE format(See Table 6) for clarification as well as supporting use case diagrams in the next section.

Table 6: Functional Requirements (IEEE Format)

ID	Requirement Description	Priority	Source
FR1	Each user should be able to securely log in.	High	User interview
FR2	Each user can enter a journal entry through the interface.	High	User interview
FR3	The system shall analyze the text using the predication model.	High	Model Specifications
FR4	The system should predict the emotion category from the model and return it.	High	Model Specification
FR5	Each predication should be visualized in the dashboard line chart.	Medium	Informal Testing
FR5	The System shall display journal entries and predication in history visualization.	Medium	Implementation Details
FR6	The system shall store user journal and predication ensuring secure database.	High	Implementation Details

Non-Functional Requirements:This part follows the same approach and here we can find the description of the second part of the requirements (See Table 7).

Table 7: Non-Functional Requirements (IEEE Format)

ID	Requirement Description	Priority	Source
NFR1	The system should ensure user login and journal data transmitted and stored securely.	High	security standard
NFR2	The application shall be responsive and operate smoothly.	High	usability expectation
NFR3	The system shall respond to user action within 3 seconds.	High	Literature Review
NFR4	The app will have user-friendly interface for all functions.	High	UI/UX best Practices
NFR5	The models shall provide 72 percent accuracy based on validation.	Medium	literature review
NFR6	Back and server and emotion label shall be available during operational hours.	Medium	Reliability Expectation
NFR7	the system shall be maintainable and allow updates to the model or UI without user data loss.	Medium	implementation guidelines
NFR8	All personal Journal data should remain private and accessible only to the user.	High	Data Privacy Guidelines

5.1.3 Use Case Specifications

Here we present use case specifications for the main functions of the app following an object oriented approach, each use case illustrates the actions made by the user and how the system operates to execute the actions to achieve a certain goal wanted by the user or set by the app system.

Use Case 01: Submit Journal Entry & Get Emotion Prediction
Actors:
User, ML Prediction Model(backend), Firestore Database

Preconditions:
User is logged in and the Internet connection is active.

Main Flow:
Users navigate to the journal entry page.
User writes a journal entry.
User submits the entry.
The system sends the entry to the ml model.
The model returns a protected emotion labeled.
The system saves the journal plus prediction to Firestore.
The app stores the data locally as well for offline access.

User sees a confirmation that the entry has been saved and analyzed.

Alternate Flows:
If the internet is unavailable, the journal will still be saved locally and sync when connection is back.
If the model API fails, the system displays an error and saves only the entry(prediction later).

Postconditions:
Journal entry along with its prediction will be saved locally and in cloud storage.
Predication is displayed on the visualisation for mood tracking.

Use Case 02: View Predictions on Dashboard
Actors:
User, Firestore Database, App UI

Preconditions:
User is already logged in and there is a saved journal entry along with the predication

Main Flow:
Users navigate to the dashboard page.
The app restores the predictions from the Firestore.
The system preprocesses emotion abels.
The labels are visualised on the line chart.
The model returns a protected emotion labeled.
The user views visualisation on dashboard and mood history to keep track of the emotions over time.

Alternate Flows:
If the internet is unavailable,the visual will show saved locally predictions.
If no entry exist, empty state would be the current.

Postconditions:
The user is able to view their insights and Trends over time.

Use Case 03: Sign Up & Log In to the App
Actors:
New User, Registered User, Firebase Authentication

Preconditions:
App is installed and running, internet connection

Main Flow:
New users open the app and click on sign up.
User enters an email and password.

Firebase creates an account and saves the new user.
The app shows the dashboard page.
User data is stored in a firestore under a unique ID .

Main Flow:
Registered users open the app.
User enters an email and password.
User clicks on the login button.
Firebase validates the account.
The app shows the dashboard page.

Alternate Flows:
The account not created yet the app shows error message
the email has already been signed up with before error message

Postconditions:
The user authenticated and redirected to the dashboard.

5.1.4 Domain Object
This conceptual model represents the general idea and main functions that are implemented for the system and their relationships.

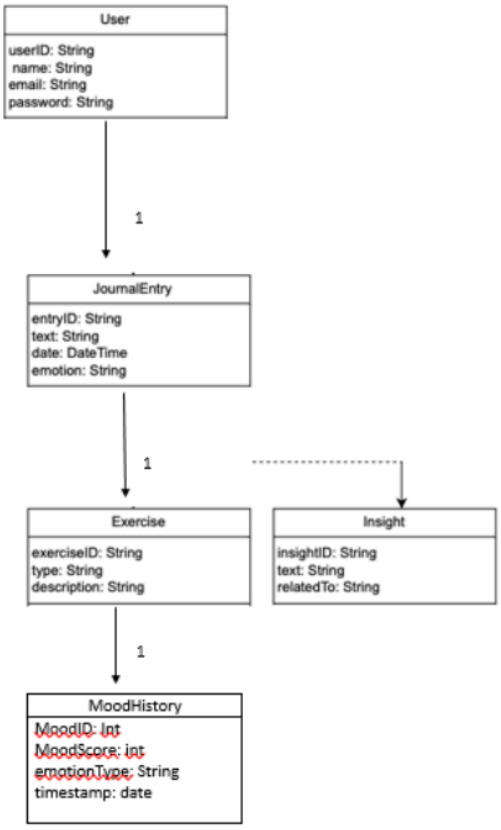


Figure 8: Domain object diagram

5.2 System Design

As described in detail in the previous section, the system was designed following an object-oriented approach. The preceding diagrams were created to illustrate the systems architecture and behaviour.

The use case diagram for the *Emoona* mobile application provides a comprehensive view of the system and mainly there are two factors where the user interacts with the system, system (Emoona): Represents the app boundary. Illustrated clearly (See Figure 9).

Use Cases:

- **Sign up / Log in** – Required to access other features (included use case).
- **Analyze emotion** – The user writes text and receives emotional analysis.
- **View mood history** – The user can view past emotional trends.
- **Access exercises** – The user gets recommendations like breathing or journaling.
- **Update settings** – The user can change app preferences.

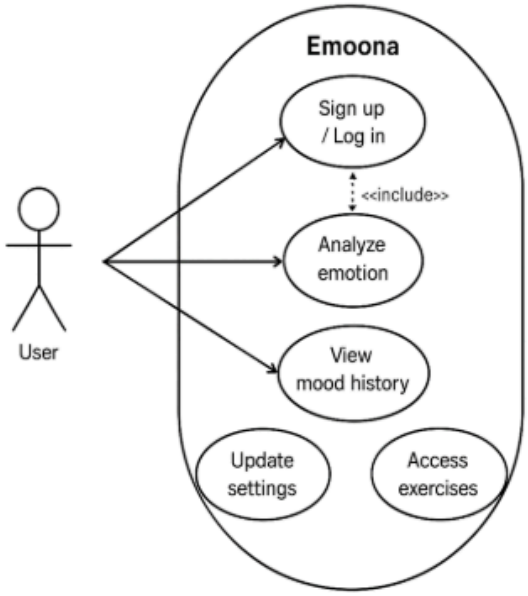


Figure 9: Use Case Diagram

The following diagrams provide details of the activity (See Figures 10-13).

The first activity diagram (See Figure 10) for the logging In process in the app. It illustrates that the user starts the login process by entering a username and password and when The credentials are submitted the system checks if the credentials are valid.

- If valid, the user is taken to the home screen.
- If not valid, an error message is shown.

The Second activity diagram (See Figure 11) for the writing journal entry process in the app. It describes the flow of making a new journal entry, the user starts the process then the journal editor screen opens. Lastly a decision is made either.

- If the user does not write any text, the app saves the (empty or draft) entry.
- If the user writes text, the process ends normally.

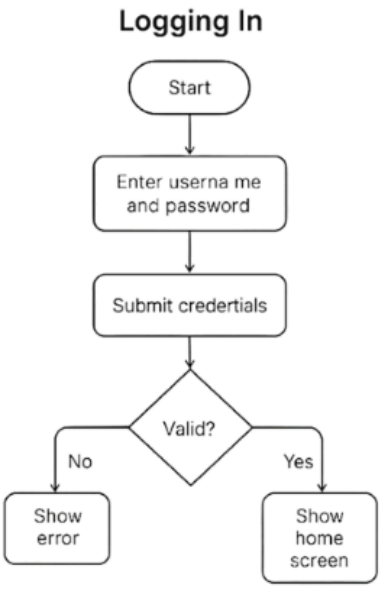


Figure 10: Logging In Activity Diagram

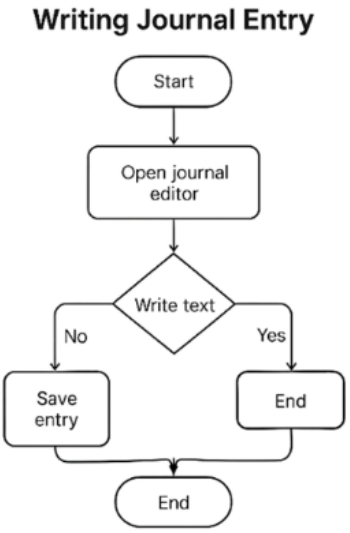


Figure 11: Journal Entry Activity Diagram

The Third activity diagram (See Figure 12) for the *sentiment analysis* feature in the app. It shows the process flow as follows: The user starts the process and enters a journal text then the system checks if the sentiment can be analyzed.

- If yes, the process ends without showing a result (this might indicate an error or early exit).
- If no (i.e., analysis is done), the app displays the result and then ends.

The last activity diagram (See Figure 13) for viewing mood history in the Emoona app.. It shows the process flow as follows: The user starts the process selecting to view mood history. After that the app checks if any mood entries exist.

- If yes, the process ends without showing a result (this might indicate an error or early exit).
- If no (i.e., analysis is done), the app displays the result and then ends.

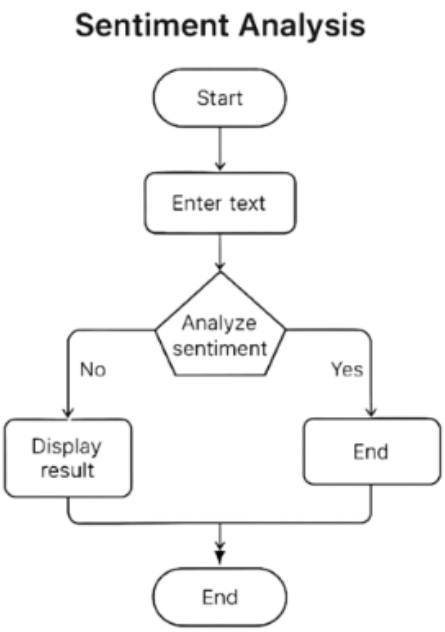


Figure 12: Sentiment Analysis Activity Diagram

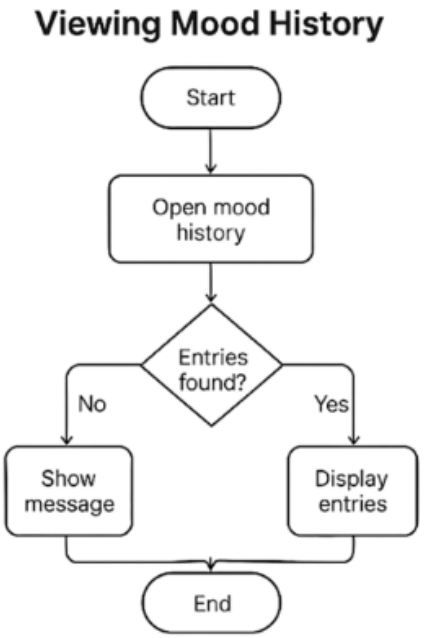


Figure 14: View Mood History Activity Diagram

The following section we have the sequence diagram (See Figure 14-16) that describes the flow of interactions between the user and the system calls.

This is a sequence Diagram (See Figure 14) outlines the login process it visually explains how the login validation process directs users to the correct screen.
Flow:

- The user opens the app
- Then enters their credentials (email and password)
- The system checks whether the credentials are valid or not

The results:

- If valid, the home screen is displayed
- If invalid, an error message is shown
- In either case, the flow ends after attempting to log in

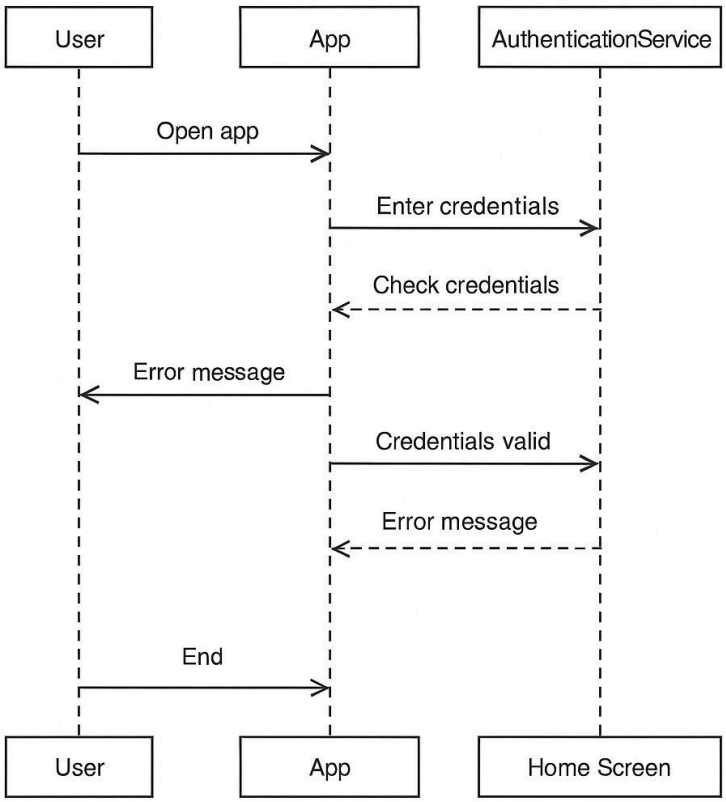


Figure 14: Sequence Diagram of Logging In

This is a sequence Diagram (See Figure 15) for starting the app Flow and dealing with the interactions in the Emoona application, this diagram represents the steps a user follows to open the app and analyze a journal entry.
Steps:

- The process starts when the user opens the Journal Entry page.
- The user either types their feelings or selects a predefined emotion.
- When they click the “Analyze Sentiment” button, the entry is sent to a Machine Learning model.
- The ML model returns the sentiment result.

The results:

- Displayed to the user
- Saved in Firebase
- And used to update both the Mood History and Dashboard, ending tthe process.

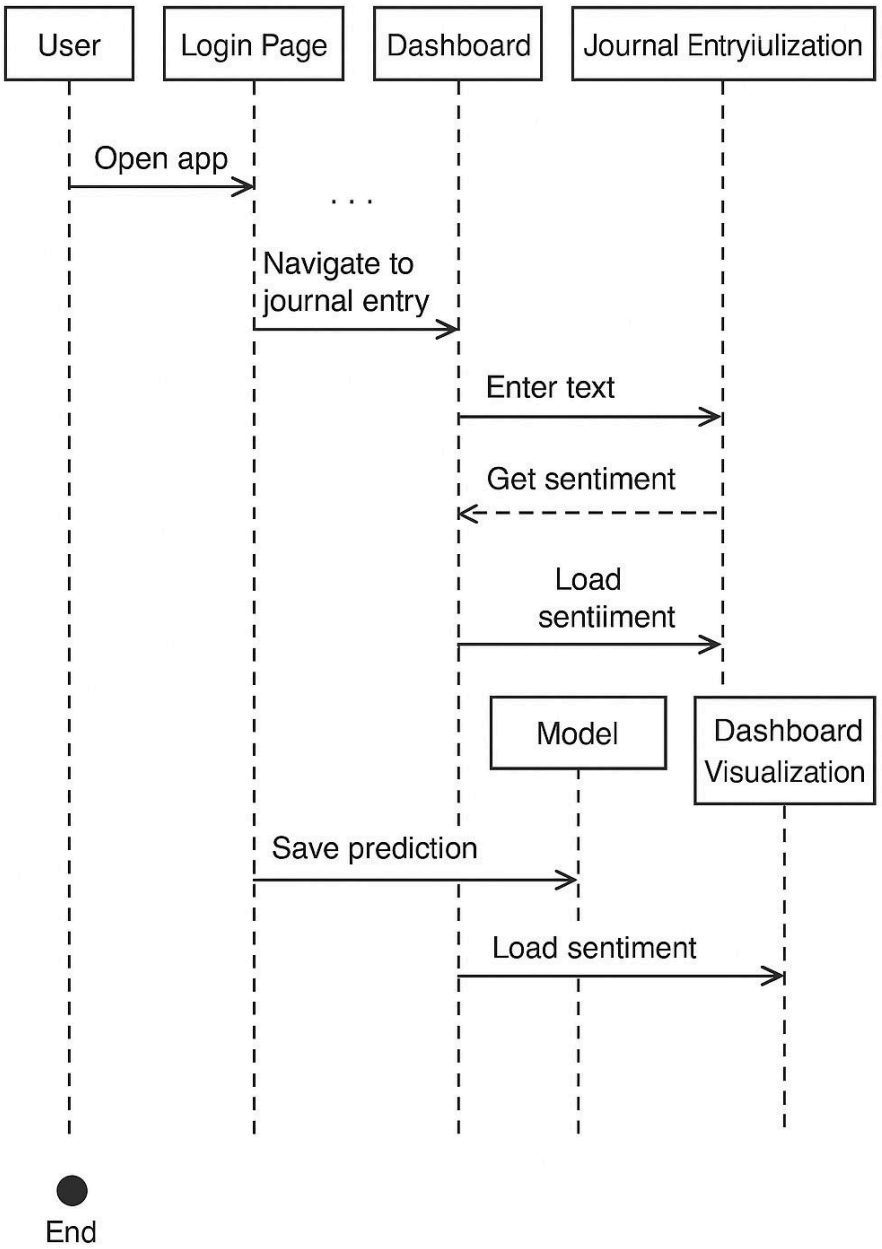


Figure 15: Sequence Diagram of Starting the App

This is a sequence Diagram (See Figure 16) for Sentiment Analysis, this diagram shows the flow of actions when the user analyzes their emotions:

Start

- User opens the app and enters a journal entry
- The text is preprocessed (e.g., cleaned, tokenized)
- Sentiment analysis is performed using the ML model

Then, based on the result:

- If the emotion is Happy, display the “Happy” label
- If not happy, the system checks if it’s Angry, Sad, or Surprised and shows the corresponding label

Finally:

- The analysis result is saved
- The application session ends

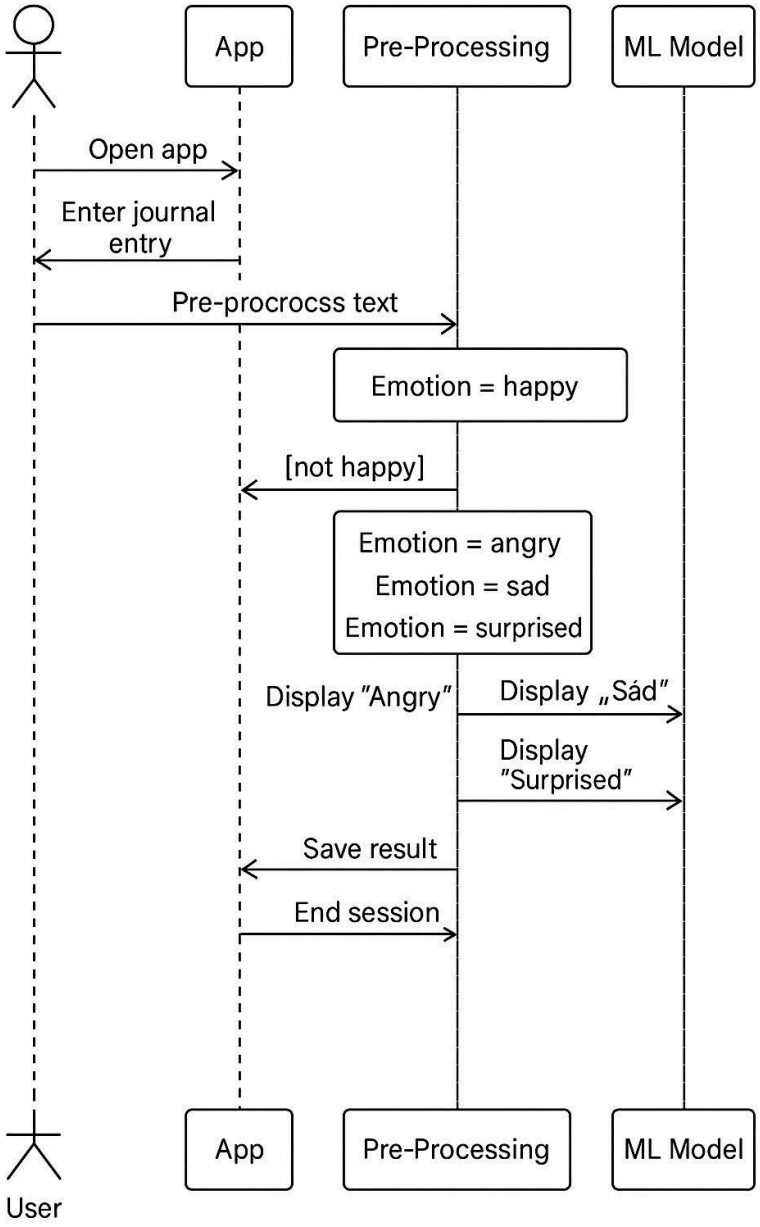


Figure 16: Sequence Diagram illustrating
the Sentiment Analysis Process

This sequence Diagram (See Figure 17) for visualisation for mood history page, this diagram shows the flow of actions when the user analyzes their emotions:

Start

- User opens the app and enters a journal entry
- Sentiment analysis is performed using the ML model
- The predication is stored locally and in firestore

Then, based on the result:

- The visualisation will show one of the four categories of emotion by a dot on the line chart
- For each new dot added the line connects all of them until we have a line chart that represent the users emotions

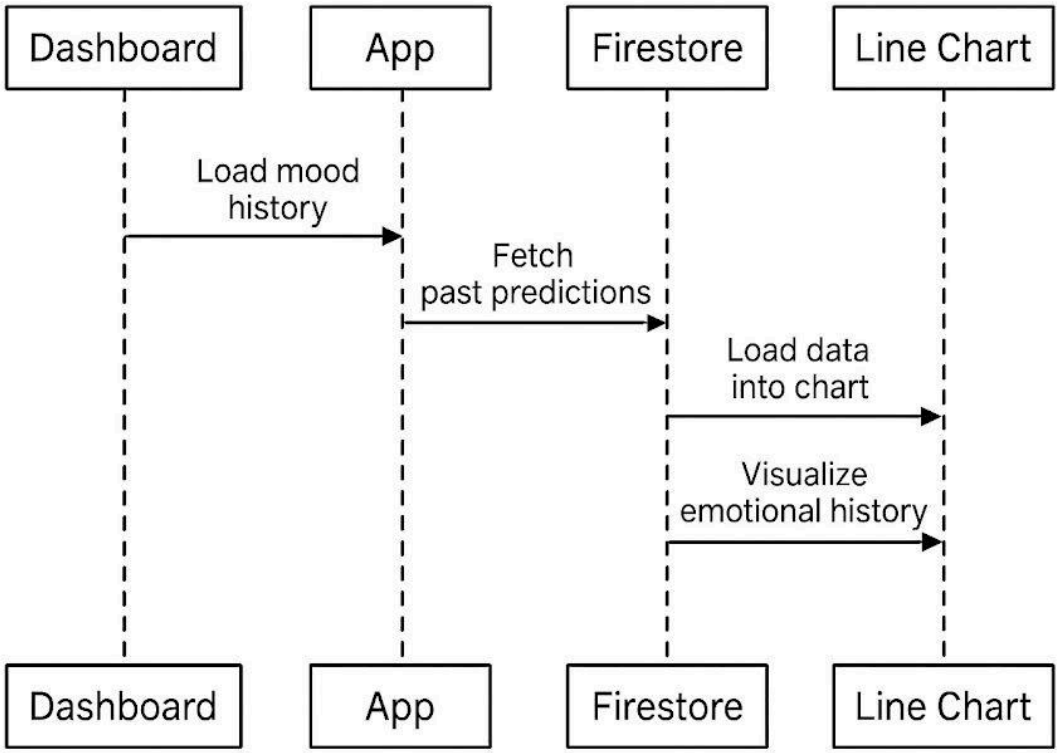


Figure 17: Sequence Diagram Illustrating the Mood History process

Class diagram for emoona app (See Figure 18) this class diagram represents the structure of the Emoona mental health application using object-oriented design. It includes the key classes, their attributes, and methods:

User: In the app attributes like `userID`, `name`, `email`, and `password`, identify the user and those two are methods to `signUp()`and `login()`.

JournalEntry: Stores each journal entry by the user. Includes `entryID`, `text`, `date`, and `emotion`, with a method `analyzeSentiment()` to trigger emotion detection.

MoodHistory: follows the user's mood predications with fields like `moodID`, `moodScore`, `emotionType`, and `timestamp`.

Exercise: presents well-being exercises with attributes such as `exerciseID`, `type`, and `description`. The method `startExercise()` allows users to begin an activity.

Insight: Contains personalized insights generated from user behavior and mood trends.

SentimentAnalyzer: A helper class that is for analyzing using `analyze(text: String)` and returning an emotion based on the text.

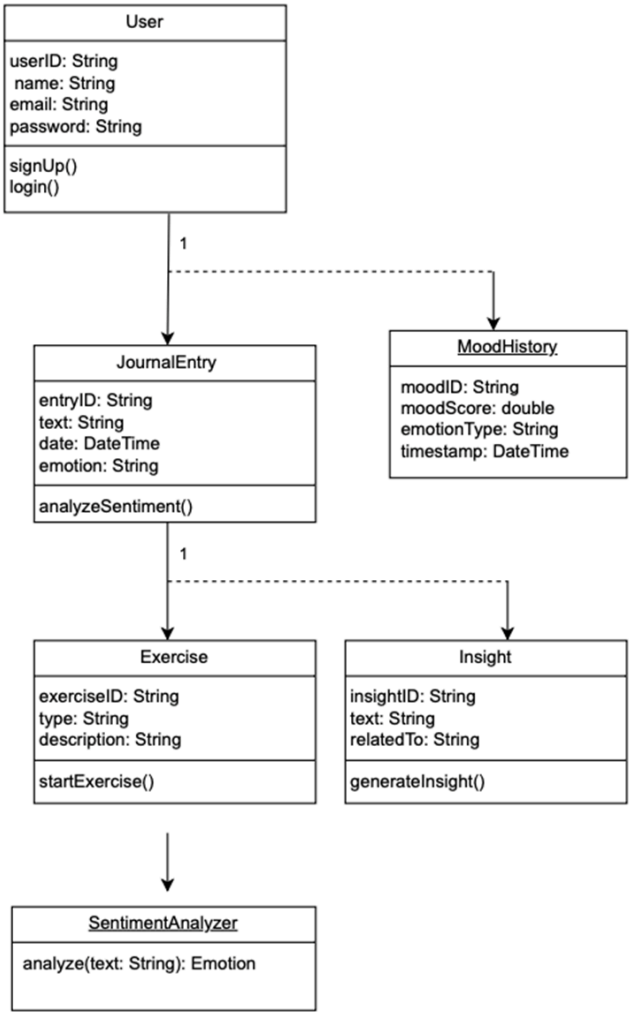


Figure 18: Class Diagram Representing the App.

Component Diagram (See Figure 19) focuses on five main components of the system and how they interact during the process.
Mobile Application: UI components, Navigation Logic, Firebase Integration and HTTP Client.
The sentiment analysis process is invoked through API
Sentiment Analysis API: RoBERTa Model, Pre Processing Logic, Emotion Grouping Logic.
The results are retrieved using firebase for storing.
Firebase Services: Firestore Database, Authentication, and Cloud Storage.
Receives data from the prediction system
Admin Dashboard: User Analytics, Logs, and System Monitoring
Frequently updated insights helps manage the tracking system

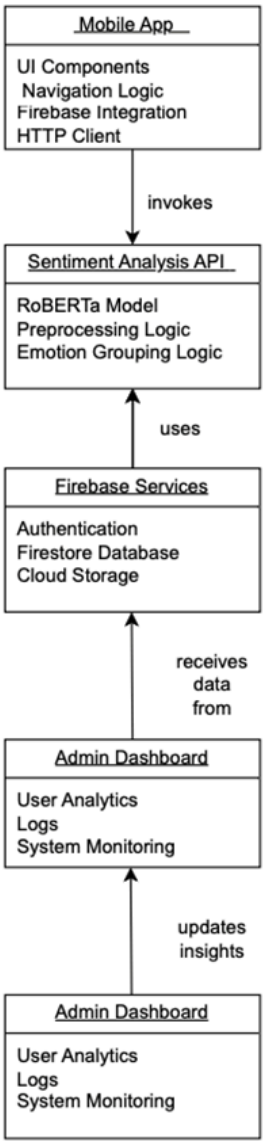


Figure 19: Component Diagram of the System

The following is a deployment diagram (See Figure 20) showing different aspects of the Emoona system deployed and their interaction during run time.
User Mobile Device: Flutter app (emoona) which contains the UI and local logic that runs on the user's phone.

Web Server / API: RoBERTa trained model uploaded on Huggingface website, hosted the model on Render, and the flutter app sends requests to the Render hosted model and receives predictions.
Firebase Backend: Firebase database for storing journal entry and mood history and storage for user information.

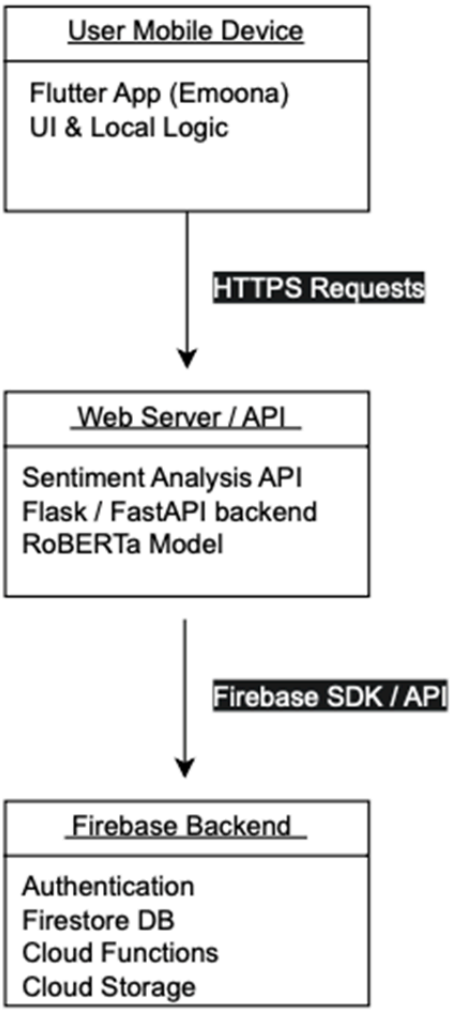


Figure 20: Deployment of the System Diagram

Output layout In this section we can see the output from the application and different scenarios where the intertext is something like ” I'm feeling really good today” which predict a happy category or “ I'm shocked that my friend didn't invite me for their important event” this results in Surprise category and based on this examples and extra ones we can see different outputs

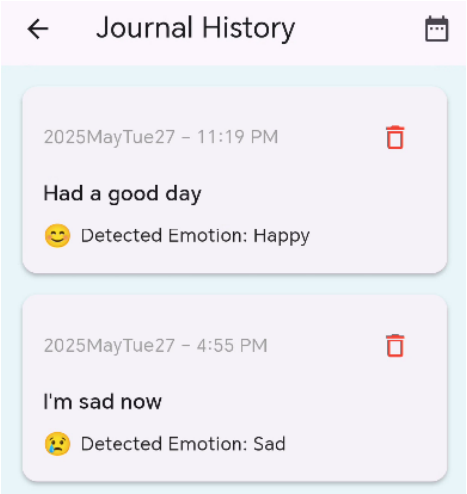


Figure 21: The log of all previous journal entries

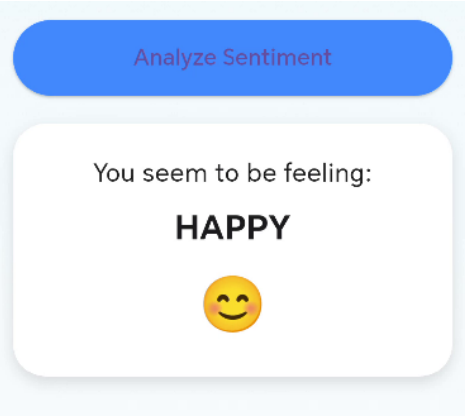


Figure 22: Prediction results of a happy sentiment.

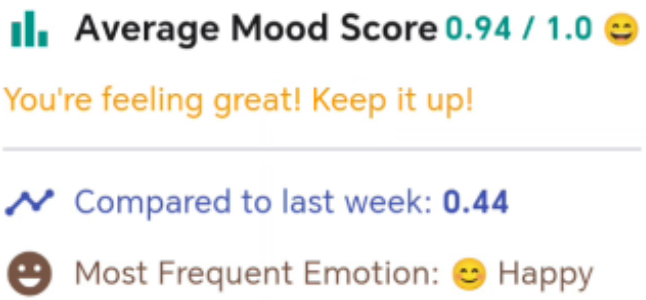


Figure 23: Results of summary of different Predictions

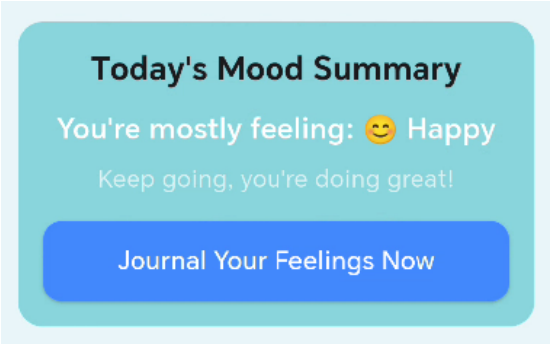


Figure 24: Summary of the latest productions

Figure 25: Dashboard Visualization

Input design output of the previously explained system with main screens, all details from the sections and diagrams (See Figure 26 - 30) show the results of the conducted work regarding the *Emoona* application.

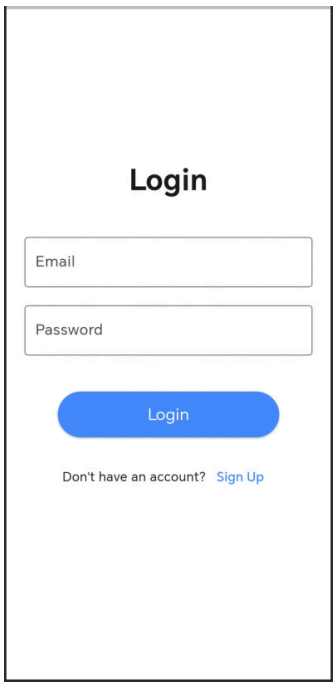


Figure 26: Login Screen

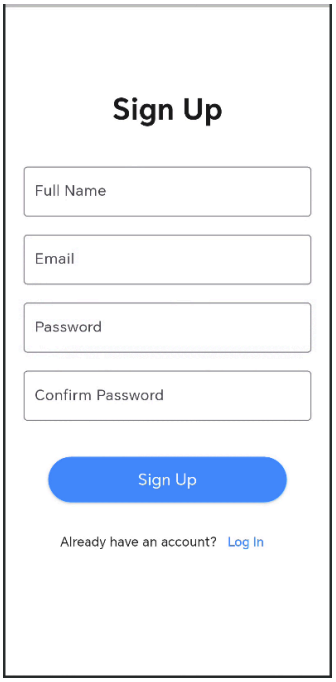


Figure 27: Sign Up Screen

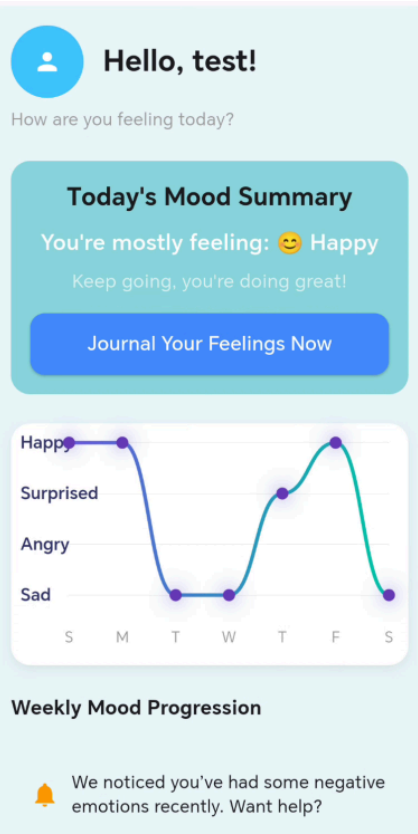
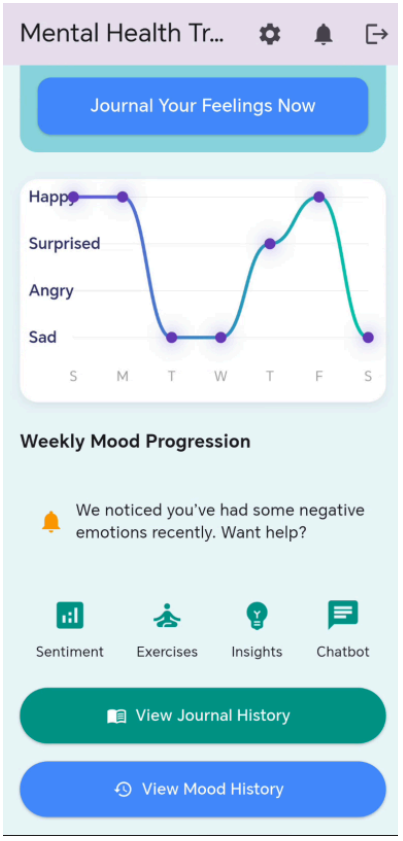


Figure 28: Dashboard Screen



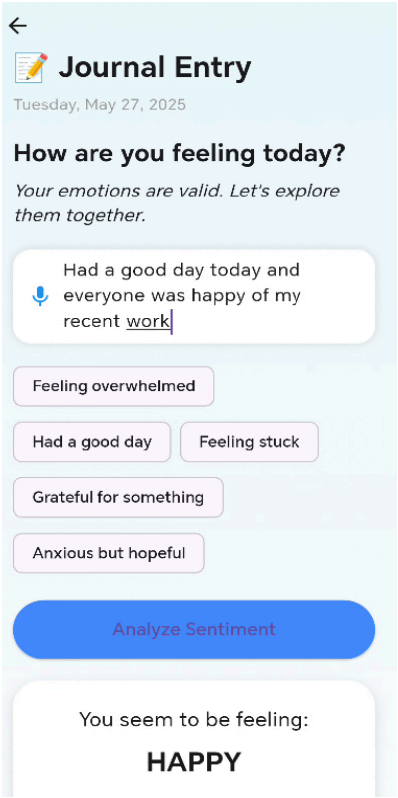


Figure 29: Journal Entry Screen

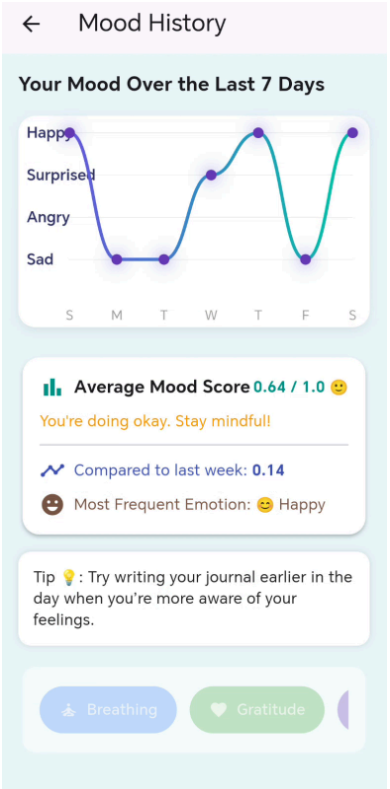


Figure 30: Mood History Screen

5.3 System Implementation

And describes the development of the application and how the designs are translated into a working system. all designs are laid out in chapter three and the models integration was tested and verified during development. Our implementation includes:

- Using Dart we built a flutter base mobile application.
- For authentication purposes and the cloud database we used firebase.
- RoBERTa Model was hosted on hugging face and served via fast API deployed on Render
- Journal entry takes user input, sends it to the back end model, receives an emotion reduction and visualizes that in a line chart.
- all journal entries and corresponding emotion predictions are stored and firestore and displayed hysterically.

5.4 Deployment

For this section we have the final system that has been deployed and integrated Allowing real time journal entry submission, emotion, analysis, storage and visualization forming a complete proof of concept for an AI-powered mental health journaling assistance, as follows:

For **Front-end** first step was choosing the name EMOONA - fusion of emotions and the moon which is changing like a person's state. Developing a front-end that ensured an easy and smooth experience for the user this included:
Sign up/ Login pages
Dashboard page(visualisation, summary of the user state)
Journal entry page(takes text from user, predict sentiment category)
Exercise page(advices that would benefit user)
Settings page(for flexibility to the user)

Backend of the application we connected the firebase with the app to keep record of new users that sign up for the app and secondly created a database that keeps track of each user journal entries along with its prediction.

Model Hosting the app sends an HTTP POST request to the backend with journal entry then the model that is hosted via render responds with the prediction which is one of the four categories.fine-tuned RoBERTa hosted on hugging face and served via fast API deployed on Render
Database Integration firebase Firestore is used for storing user journal entries and predicted emotions.
Visualization: Summaries of these entries and predication gets sent to the dashboard page to be visualized as a line chart to be saved in the mood history visualisation as feedback for the user along with text-based summary.

Chapter 6
Conclusions and Future Work.

6.1 Conclusions

This project successfully delivered a fully integrated, end-to-end *emotion* classification *system* based on cutting-edge Natural Language Processing (NLP) techniques and deep learning models, grounded in psychological theory, and implemented in a real-world mobile application called *Emoona*.

The core mission of the project was to help users track and understand their emotional state by analyzing free-text journal entries using AI. We utilized the *GoEmotions dataset* by Google, which includes over 58,000 Reddit comments annotated with 27 emotion labels. This dataset provided a rich, real-world source of emotional language and was an ideal starting point for model training and experimentation.

- Significant class imbalance, with rare emotions like *grief* and *embarrassment* underrepresented.

- Over-reliance on the "neutral" label, which blurred distinctions between true emotional expressions and led to poor classification performance.

- Multi-label annotations, which complicated the use of traditional classification algorithms.

To address these challenges, we implemented a multi-step preprocessing pipeline that included:

- Dropping irrelevant metadata columns to focus solely on emotional content.

- Removing the "neutral" class, which acted as a catch-all and reduced model sensitivity to subtle emotional cues.

Mapping the original 27 emotions into 4 psychologically grounded categories based on Ekman's theory: *Happy, Sad, Angry, and Surprised*. Filtering for single-label samples only, which simplified the learning task and boosted model performance. These refined datasets were then used to train and compare three models: LSTM, BERT, and RoBERTa. All models were trained under identical conditions for a fair evaluation. The RoBERTa-base model emerged as the most effective, achieving:

Accuracy: **72.67%**

Macro F1 Score: **0.646**

These results confirmed the advantage of dataset restructuring and domain-specific grouping, demonstrating that RoBERTa, when trained on a psychologically informed dataset, can

outperform other models even in scenarios where previous research showed BERT had an edge. Following training, the model was successfully integrated into the Emoona mobile app, built using Flutter, with Firebase as the backend for storage and user management. The app allows users to:

- Write journal entries expressing their daily emotions.

- Receive instant predictions of emotional states (Happy, Sad, Angry, Surprised).

- View personalized mood history charts and emotion trends over time.

- Access supportive tools such as breathing exercises and journaling prompts.

In conclusion, From a human-centered design perspective, Emoona bridges the gap between advanced AI research and emotional self-care. It transforms abstract predictions into actionable insights and creates a private, accessible space for users to reflect on and manage their mental well-being. This project also illustrates the power of combining:

Technological innovation (transformer-based NLP models like RoBERTa), Psychological frameworks (Ekman's theory of emotion), and mobile computing (Flutter and Firebase). To build a system that is accurate, meaningful, and useful. It lays a strong foundation for future research and applications in emotion-aware AI, personalized mental health tools, and empathetic technology proving that emotional intelligence can be built into machines in a way that supports, rather than replaces, human experience.

6.2 Future Work

While the current system has achieved strong results, and as an other applications whether related to sentiment analysis of other domains several areas remain open for further development and improvement:

1. Train on Emoona's Own Dataset

The most significant next step is to build and train the model on Emoona's own user-generated dataset, which will be collected directly from journal entries submitted in the app. Unlike the general-purpose GoEmotions dataset, this domain-specific data will better reflect the emotional expressions and vocabulary of our target users, improve accuracy by aligning training with the app's use-case, allowing for more personalized and relevant emotion predictions.

2. Expand Emotion-Based App Features

We plan to integrate additional app features powered by emotion analysis, including:

- Emotion-driven activity suggestions (e.g., breathing, music, meditation).
- Notifications and reminders based on negative mood trends.
- Personalized mental wellness advice.

Collaborate with Mental Health Experts

- We aim to work closely with psychologists and mental health professionals to:
- Validate the app's emotional predictions.
- Ensure clinical accuracy and ethical use of emotional data.
- Potentially integrate professional insights and therapy tools.

Multilingual Expansion

- The app will eventually support Arabic and other languages, requiring new datasets and retraining the model for each language to maintain high accuracy and cultural relevance.

References

[1] J. Demszky, M. Movshovitz-Attias, R. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," Google AI Blog, 2020. [Online]. Available: https://github.com/google-research/goemotions. [Accessed: March 10, 2025].

[2] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Transformers: State-of-the-Art Natural Language Processing," Hugging Face, 2020. [Online]. Available: https://huggingface.co/transformers/. [Accessed: April 1, 2025].

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, Jul. 26, 2019. [Online]. Available: https://arxiv.org/abs/1907.11692. [Accessed: April 5, 2025].

[4] K. Wang, Y. Su, Z. Jing, and Y. Han, "Large Language Models on Fine-grained Emotion Detection Dataset with Data Augmentation and Transfer Learning," arXiv preprint arXiv:2404.06042, Apr. 9, 2024. [Online]. Available: https://arxiv.org/abs/2404.06042. [Accessed: May 9, 2025].

[6] H. Maheshwari and V. Varma, "An Ensemble Approach to Detect Emotions at an Essay Level," in Proceedings of the Second Workshop on Natural Language Processing for Non-Monolingual EdTech, Association for Computational Linguistics, 2020, pp. 1–6. [Online]. Available: https://aclanthology.org/2020.nlpedtech-1.1. [Accessed: May 12, 2025].