Chapter 1

Introduction

Our primary objective is to encourage people to take up reading by presenting them with a fresh and innovative technology which will make the reading experience more interesting and indulging and enhance a reader's experience and to create a tool that will help readers read in an effective and efficient way by combining music and reading by playing instrumental music in accordance with the emotion of a scene in an e-book. Music and literature have an intertwined past and through this application we present a method to automatically generate music from text. Thanks to smartphones and e-books, more people are reading on-the-go and as technology continues to expand, the figures are expected to rise. With the advancement in technology, we expect to see more from today's rich, increasingly sophisticated e-book content and more streamlined integration, enhanced usability, and cutting edge interactivity; all of which can be used on-the-go and with unprecedented cost effectiveness are a few of them.

1.1 Description

In this system, we present a method to automatically generate music from literature. Specifically, we focus on e-books and generate music that captures the change in the distribution of emotion words. We list below some of the benefits:

- Creating audio-visual e-books that generate music when certain pages are opened—music that accentuates the mood conveyed by the text in those pages.
- Mapping pieces of literature to musical pieces according to compatibility of the flow of emotions in text with the audio characteristics of the musical piece.
- Appropriate music can add to good visualizations to communicate information effectively, quickly, and artfully.

The system aims to develop an e-book which plays instrumental music in the background according to the emotion in a scene. Here the user will select an e-book which will then be playing music in the background. This will be done by analyzing the text and then using an emotion lexicon, the emotion density will be calculated which will then help in generating the music for a particular scene. Since music is dependent on the emotion words in the text, the challenge in composing new music, just as in creating a new story, is the infinite number of

choices and possibilities. Music will be generated using JFugue, which is an open-source Java API for programming music that generates the appropriate audio file.

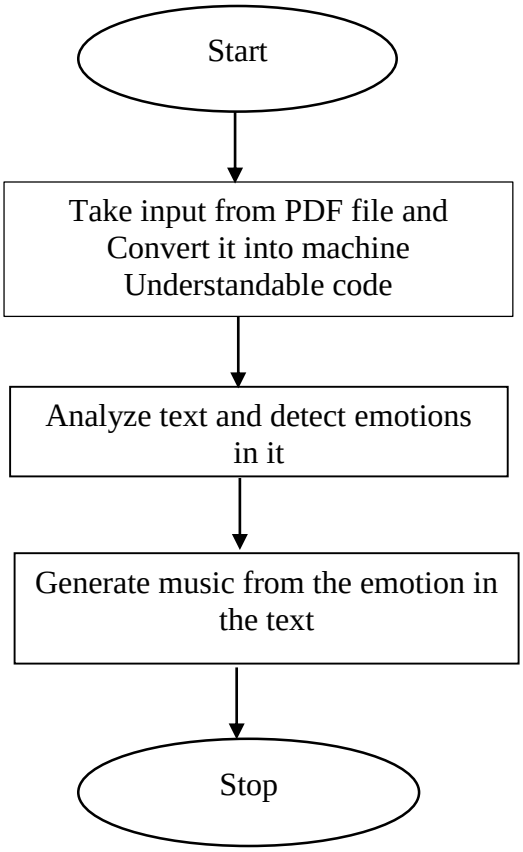


Fig 1.1.1 Working of the system

1.2 Problem Formulation

To create an application that will aim at encouraging people to take up reading by providing an innovative, fun and interesting way and to enhance the reader's experience by combining music and reading since any book combined with the right music is what readers think is the best.

1.3 Motivation

Music can be a great back drop to a theme, especially while reading. The greatest kick a reader can get while reading a book is when along with reading one can get the feel of watching a movie by playing instrumental music in the background. In addition to increasing one's

concentration, it is also quite soothing and might put you in the right spirit and enhance the experience.

We propose an improved electronic book that will analyze the emotions in a scene and generate music in tune with it. The proposed scheme will create a literary atmosphere, and will also drown out distractions so that the reader can fully focus on the page in front of him.

1.4 Proposed system

Step 1: Taking the input from the E-book

Most of the E-book is available in pdf format. So in our application we will have to create an interface that will allow user to read the e-book at the same time we will need the android device to analyze the text and generate music according to the emotions in the sentences. For doing this, an android device will take the pdf file and using a pdf Libraries in java it will convert it into machine understandable language.

Some of the Open Source PDF Libraries in Java:

i. **iText:**

iText is a library that allows you to generate PDF files on the fly. The iText classes are very useful for people who need to generate read-only, platform independent documents containing text, lists, tables and images. The library is especially useful in combination with Java(TM) technology-based Servlets: The look and feel of HTML is browser dependent; with iText and PDF you can control exactly how your servlet's output will look.

ii. **Gnujpdf:**

Gnujpdf is a Java package (gnu.jpdf.*) licensed under the LGPL. It provides a simple API to create pdf files and print using subclasses of java.awt.Graphics and java.awt.PrintJob. The PDF classes write to an OutputStream in pdf format instead of a typical Graphics object, but the method calls are the same as they would be in any Applet or Application drawing to a canvas.

iii. **PDF Box:**

PDFBox is a Java PDF Library. This system will allow access to all of the components in a PDF document. More PDF manipulation features will be added as the system matures. These ships with a utility to take a PDF document and output a text file.

iv. **FOP:**

FOP is an XSL formatter written in Java. It is used in conjunction with an XSLT transformation engine to format XML documents into PDF.

v. **JFreeReport:**

It is a Java reporting tool for formatting PDF reports. It is possible to simply hand off a swing TableModel to JFreeReport and get a paginated pdf as a result.

vi. **PJX:**

PJX is a general purpose PDF programming library for Java; with support for reading, combining, manipulating, and writing PDF documents.

vii. **PDF Clown for Java (PDF Jester):**

PDF Clown for Java (PDF Jester) is a Java 1.5 library for reading, manipulating and writing PDF files, with multiple abstraction layers to satisfy different programming styles: from the lower level (PDF object model) to the higher (PDF document structure and content streaming).

Step 2: Analyzing Text and assigning emotion to it

We use the NRC Emotion Lexicon (Mohammad and Turney, 2010; Mohammad and Turney, 2013) to identify the number of words in each chapter that are associated with an affect category. We generate counts for eight emotions (anticipation, anger, joy, fear, disgust, sadness, surprise, and trust) as well as for positive and negative sentiment. We partition a paragraph into four sections representing the beginning, early middle, late middle, and end. If required each section is further partitioned into four sub-sections. The number of sections, the number of subsections per section, and the number of notes generated for each of the subsections together determine the total number of notes generated for the novel. Even though we set the number of sections and number of sub-sections to four each, these settings can be varied, especially for significantly longer or shorter pieces of text. For each section and for each sub-section the ratio of emotion words to the total number of words is calculated. We will refer to this ratio as the overall emotions density. We also calculate densities of particular emotions, for example, the joy density, anger density, etc. As described in the section ahead, the emotion densities are used to generate sequences of notes for each of the subsections. Each of the pieces presented in this paper are for the piano with three simultaneous, but different,

melodies coming together to form the musical piece. Two melodies sounded too thin (simple), and four or more melodies sounded less cohesive.

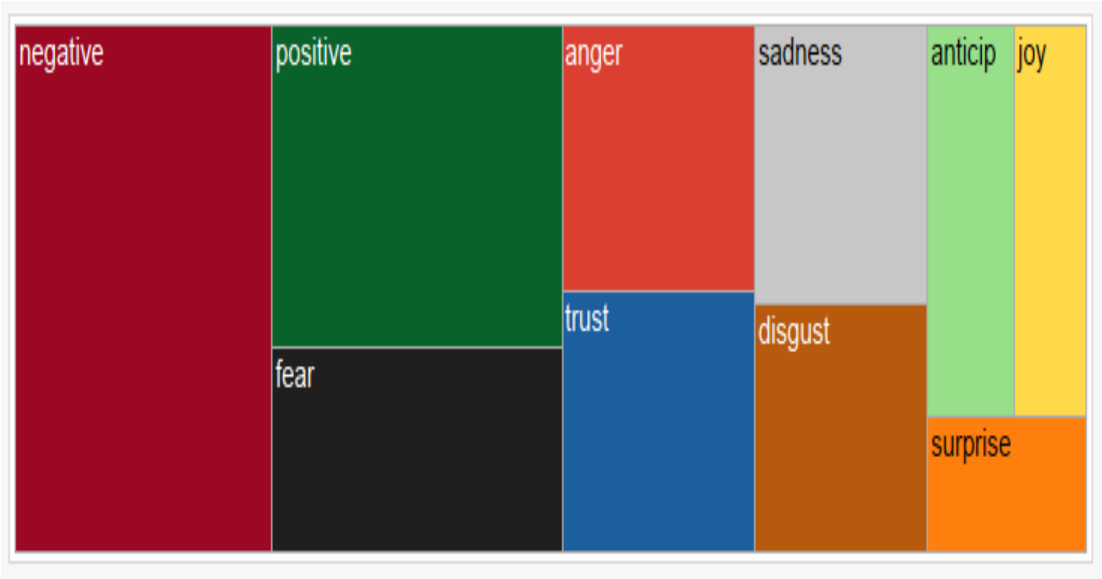
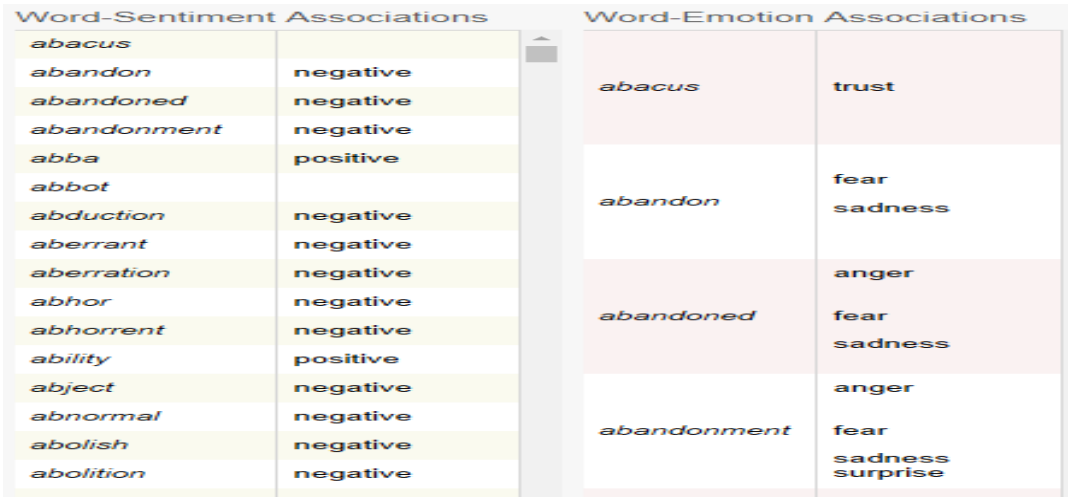


Fig 1.4.1 Affect categories: A treemap showing the number of words associated with each affect category [Reference:http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm]

Fig 1.4.2 Set of categories: A treemap showing the number of words associated with sets of categories [Reference: http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm]



Word-Sentiment Associations

abacus	
abandon	negative
abandoned	negative
abandonment	negative
abba	positive
abbot	
abduction	negative
aberrant	negative
aberration	negative
abhor	negative
abhorrent	negative
ability	positive
abject	negative
abnormal	negative
abolish	negative
abolition	negative

Word-Emotion Associations

abacus	trust
abandon	fear sadness
abandoned	anger fear sadness
abandonment	anger fear sadness surprise

Fig 1.4.3 Word-Sentiment association and Word-Emotion analysis [reference: http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm]

Step 3: Generating Music from the Emotion in text

JFugue is an open-source Java API that helps create generative music.7 It allows the user to easily experiment with different notes, instruments, octaves, note durations, etc within a Java program. JFugue requires a line of specifically-formatted text that describes the melodies in order to play them. The initial portion of the string of JFugue tokens for the novel Peter Pan is shown below. The string conveys the overall information of the piece as well as the first eight measures (or one section) for each of the three melodies (or voices).

KCmaj X[VOLUME]=16383 V0 T180

A6/0.25 D6/0.125 F6/0.25 B6/0.25

B6/0.125 B6/0.25 B6/0.25...

K stands for key and Cmaj stands for C major.

This indicates that the rest of the piece will be in the key of C major. The second token controls the volume, which in this example is at the loudest value (16383). V0 stands for the first melody (or voice). The tokens with the letter T indicate the tempo, which in the case of this example is 180 beats per minute. The tokens that follow indicate the notes of the melody. The letter is the pitch class of the note, and the number immediately following it is the octave. The number following the slash character indicates the duration of the note. (0.125 is an eighth-note (1/8th), 0.25 is a quarter note, 0.5 is a half note, and 1.0 is a whole note.) We used JFugue to

convert the specifications of the melodies into music. JFugue saves the pieces as a midi files, which we converted to MP3 format.

Table 1: Emotion and audio features of a few popular novels that were processed by TransProse. The musical pieces are available at: http://transprose.weebly.com/final-pieces.html.

Book Title	Emotion 1	Emotion 2	Octave	Tempo	Pos/Neg	Key	Activity	Joy-Sad
A Clockwork Orange	Fear	Sadness	5	171	Negative	C Minor	0.009	-0.0007
Alice in Wonderland	Trust	Fear	5	150	Positive	C Major	0.007	-0.0002
Anne of Green Gables	Joy	Trust	6	180	Positive	C Major	0.010	0.0080
Heart of Darkness	Fear	Sadness	4	122	Negative	C Minor	0.005	-0.0060
Little Prince, The	Trust	Joy	5	133	Positive	C Major	0.006	0.0028
Lord of The Flies	Fear	Sadness	4	151	Negative	C Minor	0.008	-0.0053
Peter Pan	Trust	Joy	6	180	Positive	C Major	0.010	0.0040
Road, The	Sadness	Fear	4	42	Negative	C Minor	-0.002	-0.0080
To Kill a Mockingbird	Trust	Fear	5	132	Positive	C Major	0.006	-0.0013

Fig 1.4.4 Emotions and audio features of Transpose system

[Reference :http://transprose.weebly.com/final-pieces.html]

1.5 Scope of the project:

In this system we will be implementing a e-book that can generate music according to the emotion in a scene. In addition to increasing one's concentration, it is also quite soothing and might put you in the right spirit and enhance the experience. Therefore, along with reading one can get the feel of watching a movie which is something that all readers would love.

Chapter 2

Review of Literature

Review of literature has been done based on different techniques used for sentiment analysis and what kind of music should be played in the background.

Literature survey on E-books

AC Nielsen conducted a survey on around 2,000 adults in urban cities for their India Book Market Report. In lieu with the technological advancements, 56 percent of the respondents consume at least one eBook in their lifetime. Another global survey conducted by Nielsen stated that, 54 percent use their smartphones to read books for at least 3-4 hours a day.

On the Guardian site, when readers were asked what background music they prefer to read to, a lot of them suggested some great tunes to read to – with consensus building around classical music or vocal-light tunes. Many others said they play film soundtracks, which usually combine no lyrics and an epic or soothing feel.

Literature survey on Music

Music and literature have an intertwined past. It is believed that they originated together (Brown, 1970), but in time, the two have developed into separate art forms that continue to influence each other [1]. So, based on this we decided to generate instrumental music in our application using the JFugue which is an open-source Java API for programming music that generates the appropriate audio file.

Literature survey on Sentiment Analysis

Analysis of emotions has been considered a challenging and interesting task by various researchers. However, there are few prior works who work with textual input to analyse these emotions. Emotion analysis (EA) from text is the task of predicting emotion in a piece of text. There are two modes possible for the expression of emotion which are emotive vocabulary (use of actual words) and affective items (which may be expressed through actions). As per Emotion Analysis from Text: A Survey by IIT, Bombay, datasets that have been introduced for computational studies in emotion analysis include (a) LIWC, (b) EmoLexi, (c) Wordnet-Affect, (d) ANEW, (e) ANEW for Spanish and (f) Chinese emotion lexicon.

According to Sentiment Strength Detection in Short Informal Text, a paper published by Statistical Cybermetrics Research Group, School of Computing and Information Technology, University of Wolverhampton, there have been some previous attempts to develop algorithms to detect the strength or prevalence of sentiment or emotion in text, or to differentiate between several types of emotion.

The LIWC (Linguistic Inquiry and Word Count, www.liwc.net) software from psychology, for example, uses a list of emotion-bearing words to detect positive and negative emotion in text in addition to three specific emotions of particular use in psychology and psychotherapy: anger, anxiety and sadness. It uses simple word counting, measuring the proportion of words falling within an extensive predefined list (e.g., 408 positive and 499 negative words or word stems). The list includes some words that are associated with emotions but do not describe them. For example 'lucky' is a positive keyword and 'loses' is a negative keyword. In contrast to the machine learning approaches discussed above, these lists have been compiled and validated using panels of human judges and statistical testing. LIWC calculates the prevalence of emotion in text, rather than attempting to diagnose a text's overall emotion or emotion strength. It is most suited to longer documents, for which its statistics would be useful indicators of the tendency for emotion to occur. The program uses word truncation for simplicity (e.g., joy* matches any word starting with joy), rather than stemming or lemmatisation, but does not take into account booster words like "very" or the negating effect of negatives (e.g., not happy). LIWC has been used by psychology researchers to investigate the connection between language and psychology (Pennebaker et al., 2003) [2] and also as a practical tool, for example to detect how well people are likely to cope with bereavement based upon their language use (Pennebaker, Mayne, & Francis, 1997) [3]. A related emotion detection approach differentiates between happy, unhappy and neutral states based upon words used by students describing their daily lives (Wu et al., 2006). This is similar to the typical positive/negative/neutral objective for opinion mining, however.

One computer science initiative has attempted to identify various emotions in text, focusing on the six so-called basic emotions (Ekman, 1992; Fox, 2008) [4] of anger, disgust, fear, joy, sadness and surprise (Strapparava & Mihalcea, 2008) [5]. This initiative also measured emotion strength. A human-annotated corpus was used with the coders allocating a strength from 0 to 100 for each emotion to each text (a news headline), although inter-annotator agreement was low (Pearson correlations of 0.36 to 0.68, depending on the emotion). A variety of algorithms were subsequently trained on this data set. For example, one used WordNet

Affect lists to generate appropriate dictionaries for the six emotions. A second approach used a Naive Bayes classifier trained on sets of LiveJournal blogs annotated by their owners with one of the six emotions. In our paper "Linguistic predictors of adaptive bereavement, Journal of Personality and Social Psychology," [3], we focus on using Twitter, the most popular microblogging platform, for the task of sentiment analysis. We show how to automatically collect a corpus for sentiment analysis and opinion mining purposes. We perform linguistic analysis of the collected corpus and explain discovered phenomena. Using the corpus, we build a sentiment classifier that is able to determine positive, negative and neutral sentiments for a document.

The best system (for fine-grained evaluation) was one previously designed for newspaper headlines, UPAR7 (Chaumartin, 2007), which used linguistic parsing and tagging as well as WordNet, SentiWordNet and WordNet Affect, hence relying upon reasonably correct standard grammar and spelling.

Saif M. Mohammad and Peter D. Turney of Institute for Information Technology, National Research Council Canada, stated in a survey paper that even though considerable attention has been given to the polarity of words (positive and negative) and the creation of large polarity lexicons, research in emotion analysis has had to rely on limited and small emotion lexicons. So, in this application we will use the NRC Emotion lexicon which is a dataset of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing. This dataset consists of Word Sentiment and Word Emotion Assosiations and can be used in a multitude of contexts such as sentiment analysis, product marketing, consumer behaviour analysis, and even political campaign analysis. This has greater than 14,000 unigrams (words) and approximately 25,000 word senses. Each entry in this lexicon includes a term, an emotion, and a measure of how strongly the term is associated with the emotion

EmoLex was used in a system "Generating Music from Literature" by Hannah Davis (New York University) which is similar to the ReaMu application to identify the number of words in each chapter that are associated with an affect category. The overall emotion density and densities of particular emotions are also calculated based on which the music is generated using JFugue which is an open-source Java API.

Chapter 3

System Analysis

3.1 Functional requirements:

- The system requires that ReaMu application be installed and user has an account and if not, user creates an account and text be the input.
- The system will analyze the text and apply NRC lexicon to extract annotation in each sentence.
- The system will then perform sentiment analysis and find keywords that have significant emotion.
- According to this emotion density, using JFugue, music will be generated.
- User can pause the music using a button that will be provided for the same.
- User can also upload an e-book of their choice and hence the system is not limited to the books in our database.

3.2 Non- Functional requirements:

- Performance: The response time of the system is very low, that is, the text will be analyzed and music will be played according to the emotion in a few seconds.
- Usability: The system can be used by the user whenever he/she wants a movie-like feel while reading and will also have the option to pause the music.
- Availability: The system is available for the user 24x7.
- Robustness: The system proceeds when it finds the emotion density of the text. This makes the system highly robust.
- Reliability: The system will play appropriate music according to the emotion and thus it is reliable.
- Scalability: The system will detect and recognize text in the ebook reader and analyze it for emotions according to which music will be played.

3.3 Specific requirements:

Software Requirements:

Following are the software requirements:

- Operating system: Android

- NRC Lexicon(Open source Lexicon dataset)
- Android Studio IDE (For Application development)

Hardware Requirements:-

Following are the hardware requirements:

- A device running Android 5.0 (API 21) or higher
- Since the application must run over the internet, we require an android mobile with inbuilt internet support on it. The application does not require any specialized hardware.

3.4 Use-Case diagram and Description:

Our use case diagram is a simple representation of a user's interaction with the system that shows our relationship between the user and the different use cases in which the user is involved. Fig 3.4.1 represents the use case to analyze the sentiment.

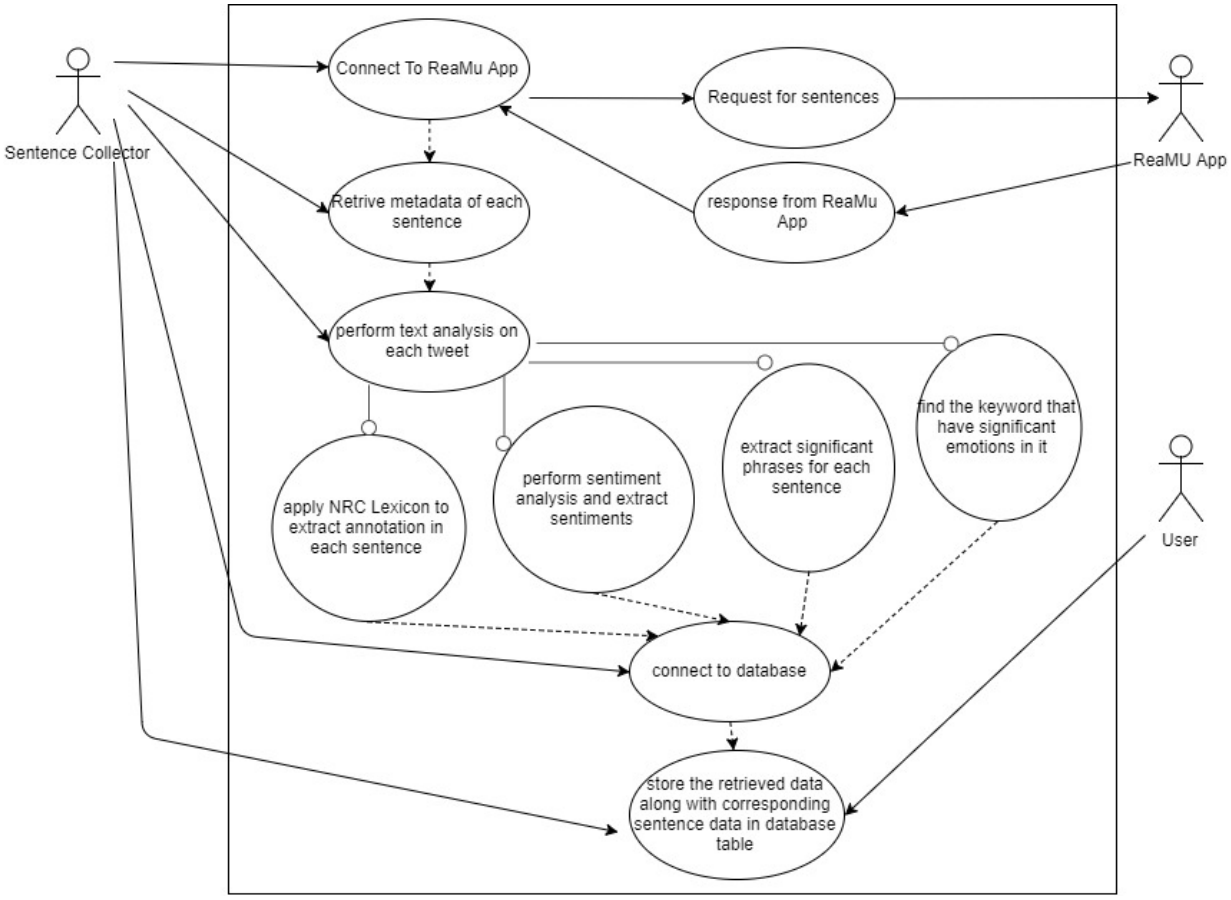


Fig 3.4.1 Use case to analyze the sentiment

In our use case diagram for **Sentiment Analyzer** we have three users:

1. The book Reader
2. App Interface(sentence generator)
3. Sentiment analyzer

The book Reader: This user uploads the books existing in PDF format in our application. The books are further simplified into sentences using the PDF libraries.

App Interface (sentence generator): The iText PDF library simplifies the PDF file into sentences. The iText is capable to generate read-write file, platform independent documents containing text. This information is stored in the Application database.

Sentiment Analyzer: The sentiment analyzer will first connect to the database of the Application and retrieve the metadata of the sentences. Then analysis will be performed on these sentences according to the following steps:

1. Apply NRC Lexicon to extract annotations in each sentence.
2. Perform sentiment analysis an extract sentiments.
3. Extract significant phrases for each sentence.
4. Find the keyword that has significant emotion in it.

Fig 3.4.2 represents the use case to generate the music according to the sentiments analyzed.

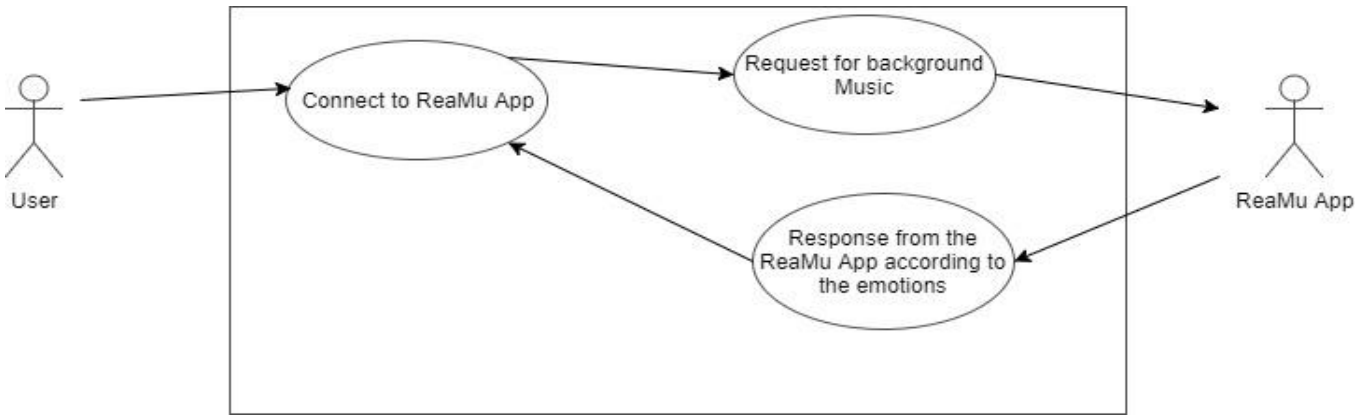


Fig 3.4.2 Use case to generate the music according to the sentiments analyzed

In our **Music Generator** we have two users:

1. The book Reader.
2. The application Interface(Music Player)

The book Reader: The book Reader when opens the book he will be displayed with an interface that that allow him to read his eBook at the same time listen to the background music. To play background music the app will input the output from the sentiment analyzer into the jfudge API. These inputs consist of numeric values which are applied to a formula to get the output in following parameters i.e. Type (positive or negative) Emotion, Octaves, Tempo, Key.

Music Player: Using these parameters the music player in the app will generate music.

Chapter 4

Analysis Modeling

4.1 Data Modeling

The ER diagram in Fig 4.1.1 describes how the data are related to each other. The user has an e-book reader that can play music. User can login to their account and select the book he/she wants to read. E-book reader then analyzes the text and plays music in accordance to the emotions.

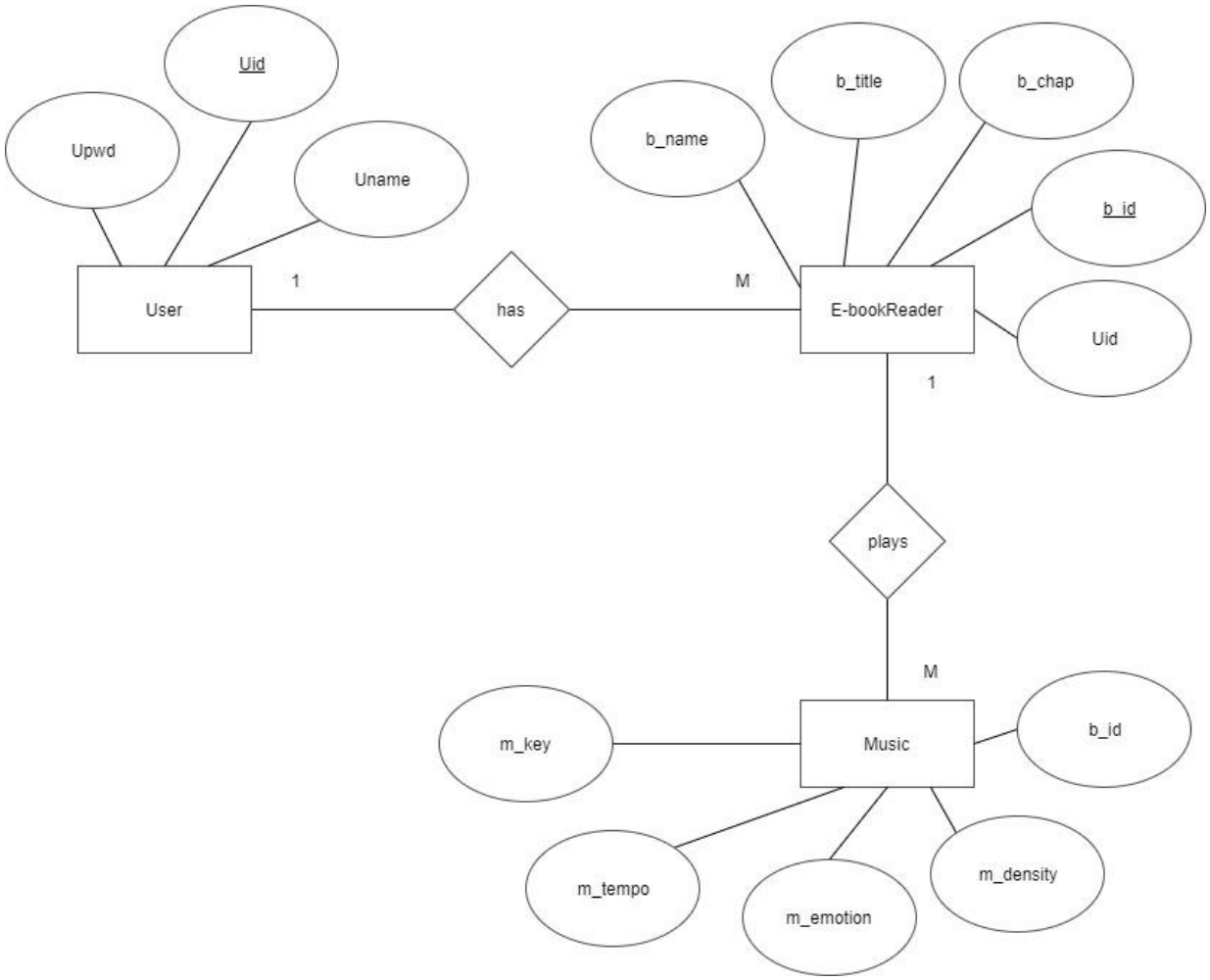


Fig 4.1.1 ER Diagram

4.2 Class Diagram

Fig 4.2.1 represents the class diagram of the system.

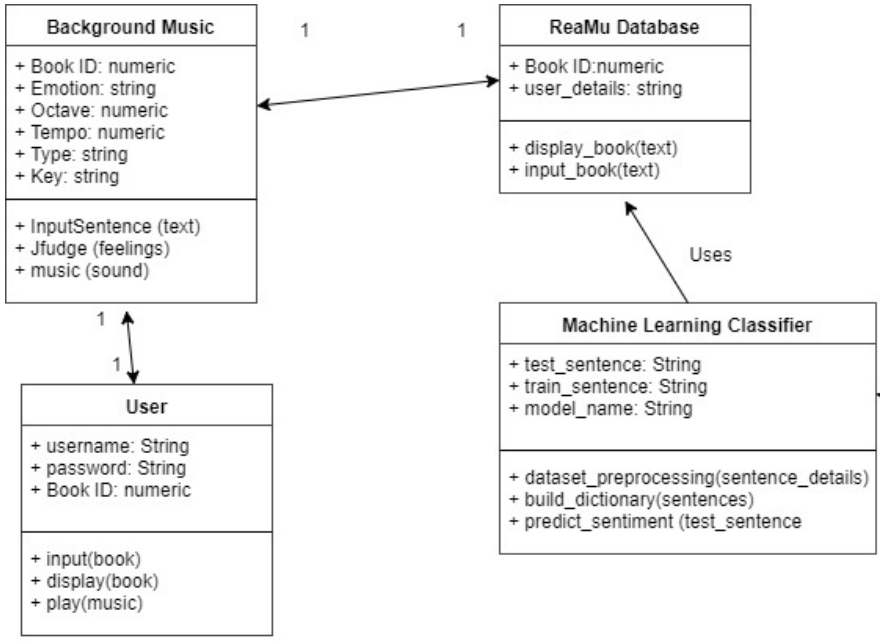


Fig 4.2.1 Class Diagram

Our application consists of four major classes:

1. **User:** The user has privileges like inputting the books of their choice, Read books of their choice and listen to the music.
2. **Machine Learning Classifier** (Sentiment Analyzer)**:** It has to carry out three major functions**.** These function includes creating dataset preprocessing (sentence details), build dictionary of sentences and predict sentiments.
3. **ReaMu Database:** It is used to store the Sentences generated from the PDF library iText. It is also used to store the values from the NRC lexicon. The music generated from the jfudge is also stored in the app database.
4. Jfudge Music Generator: It inputs parameters and generate music accordingly.

4.3 Functional Modeling

Figure 4.3.1 shows the level 0 DFD of ReaMu system. User gives input to the system in the PDF format which is converted to text to analyze the sentiments in the sentences. Using these parameters, the music is generated which is then played in the background.

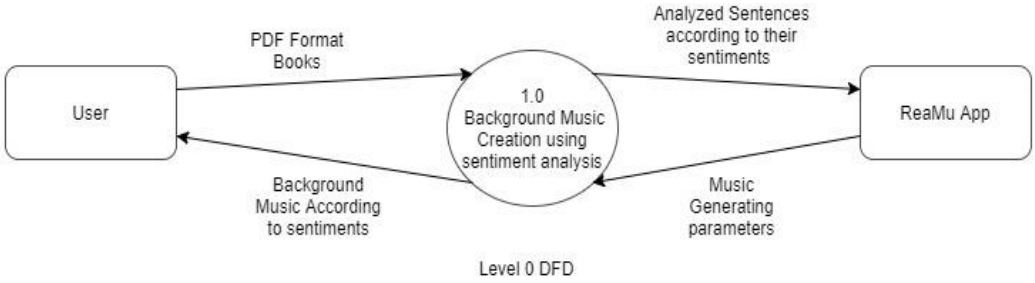


Figure 4.3.1 Level 0 DFD

Figure 4.3.1 shows the level 1 DFD of ReaMu system. It describes in detail what the system is supposed to do. Sentiment Analyzer takes sentences and storees the details in the database. When user opens a book to read it, JFugue generates music by getting the parameters from the application database.

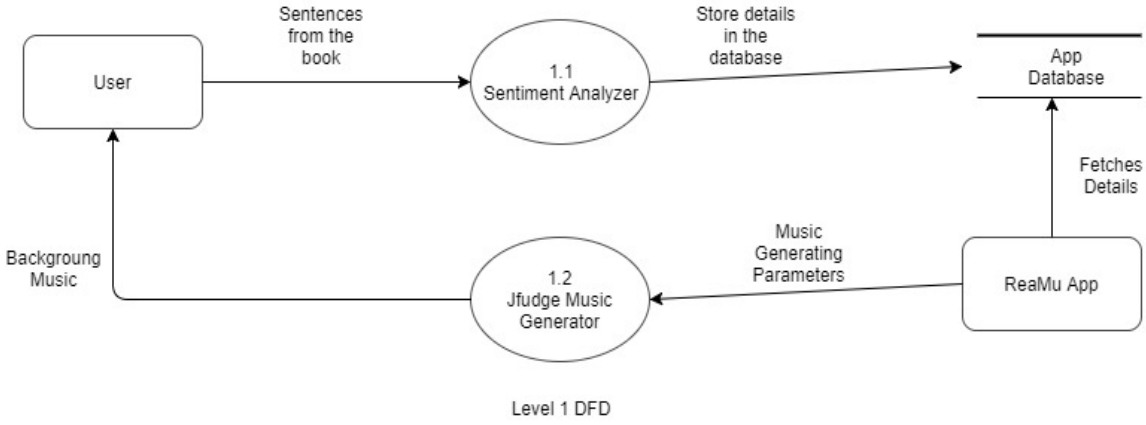


Figure 4.3.2 Level 1 DFD

4.4 Sequence Diagram:

The sequence diagram shows the flow the system will follow. Fig 4.4.1 represents the sequence diagram of ReaMu application. The user first registers and these details are forwarded to system/admin which are saved to the database. After confirmation and verification, user can login to the system to use the application. Our application support addition of E-books of user's choice. Sentences are then analyzed to get parameters and then the music is generated using the JFugue API and then it is played in the background corresponding to the text.

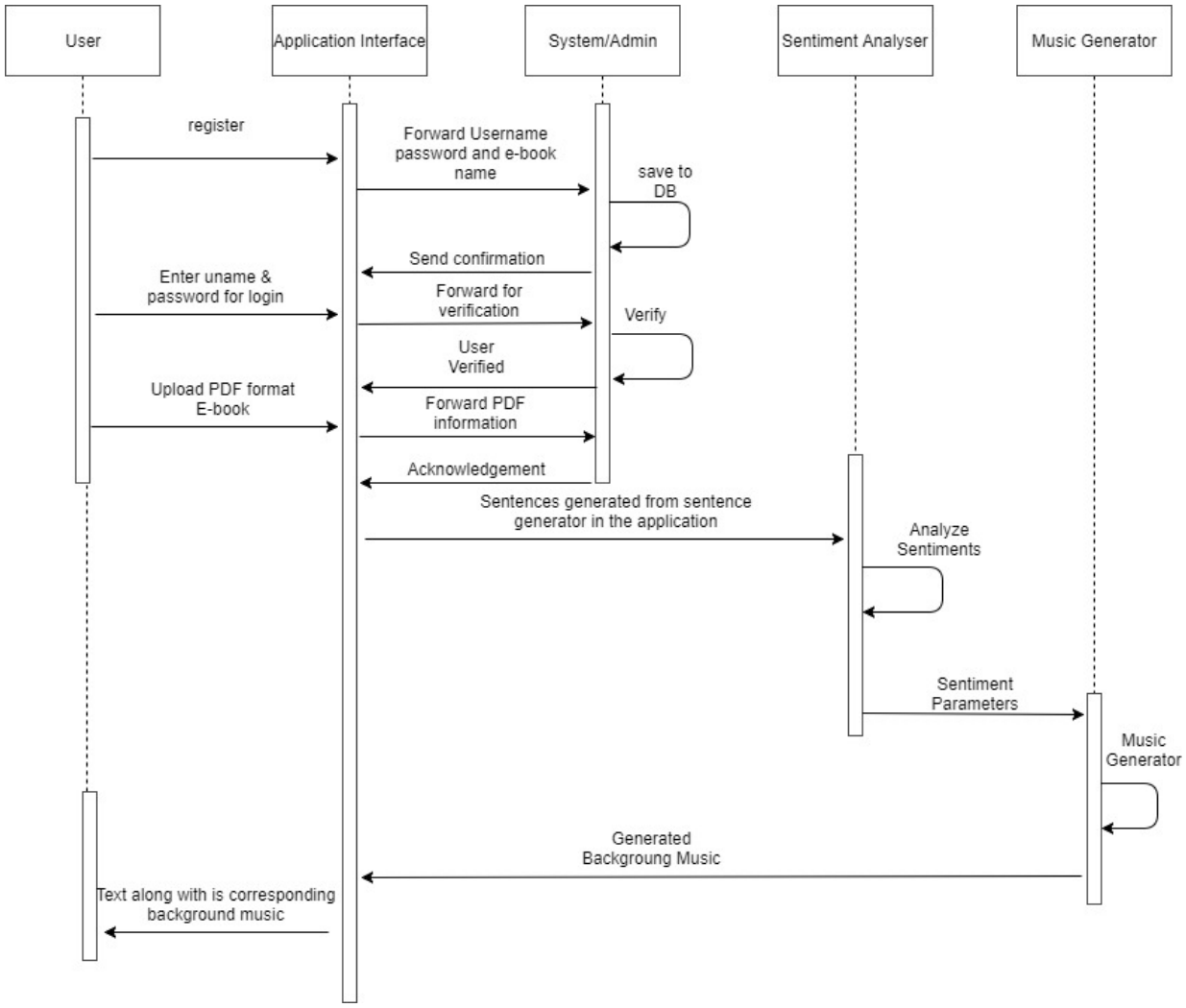


Fig 4.4.1 Sequence Diagram

4.5 Time Line Chart

The Time line chart in Fig 4.5.1 displays a list of events in chronological order. It is a graphic design showing a long bar labeled with dates alongside itself and labeled with dates where the events would happen.

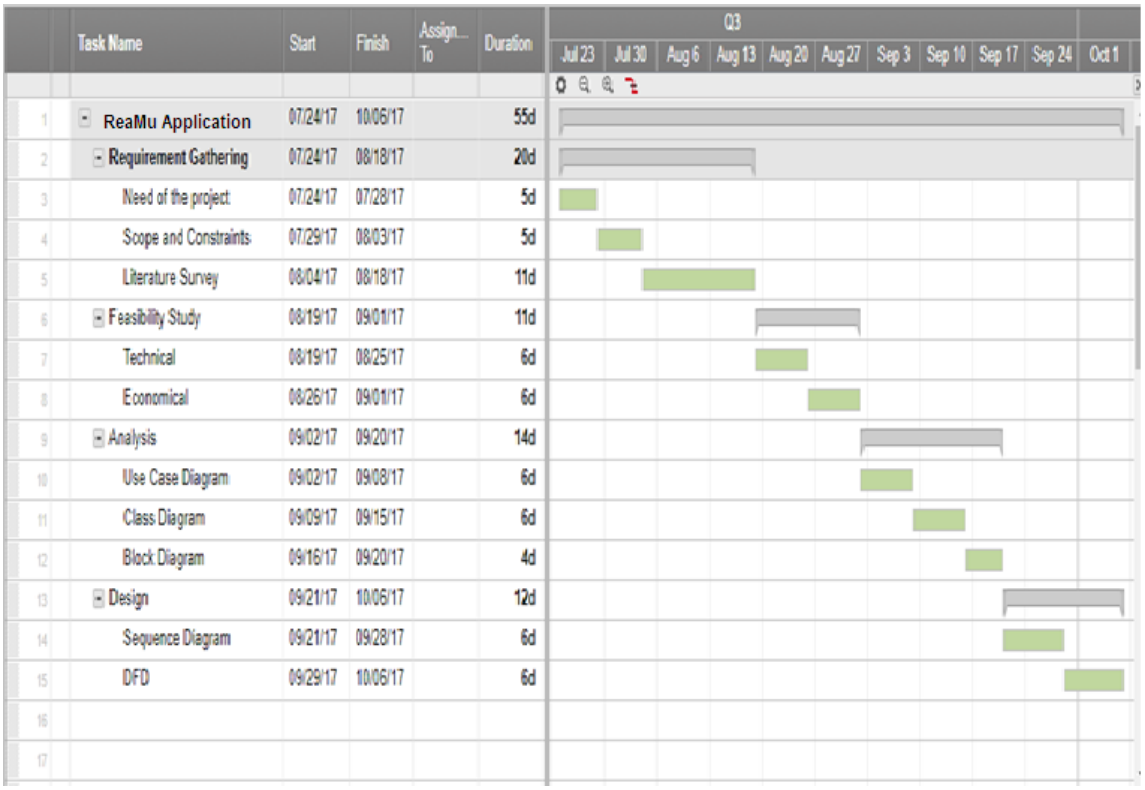


Fig 4.5.1 Time line chart

Chapter 5

Design

5.1 Architectural Design

The proposed system has the following application architecture. Fig 5.1.1 explains the architecture of ReaMu application.

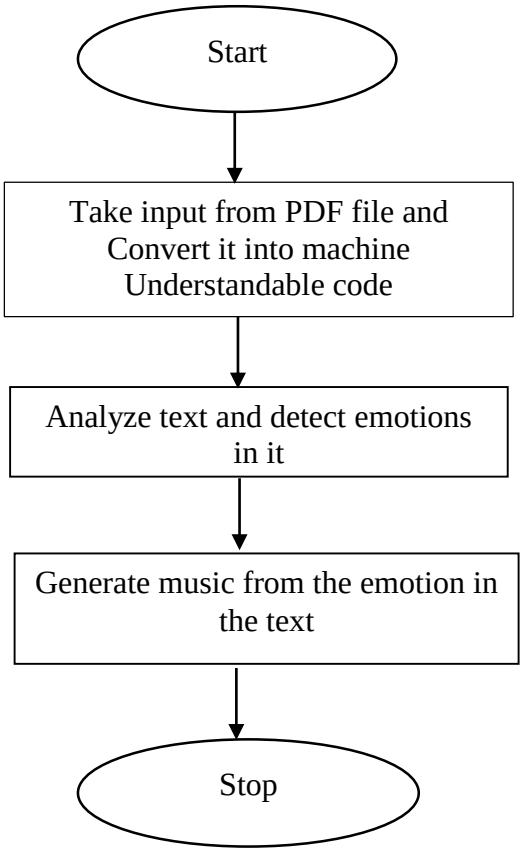


Fig 5.1.1 Architectural design of ReaMu

Here the user will select an e-book which will then be playing music in the background. This will be done by analyzing the text and then using an emotion lexicon, the emotion density will be calculated which will then help in generating the music for a particular scene. First, input will be taken from PDF file and converted into machine understandable code. Next, the text will be analyzed to detect the emotions in it. From the emotions detected, music will be generated using the JFugue API.

5.2 User Interface Design:

Following are the user interfaces required in the system:

- E-book reader installed in the device
- A page/activity that plays songs according to emotion and also has an option for pause.

Chapter 6

Conclusion and Future work

The aim of this system is to automatically generate music from literature. Literature survey helped us understand the existing systems and their drawbacks which in term helped us develop a system overcoming those drawbacks. With the advancement in technology, we expect to see more from today's rich, increasingly sophisticated e-book content and more streamlined integration, enhanced usability, and cutting edge interactivity. Music and literature have an intertwined past and through this application we present a method to automatically generate music from text. This will be the final chapter of the report. A brief report of the work carried out shall form the first part of the Chapter. Scope for future work should be stated lucidly in the last part of the chapter.

Literature Cited

[1] Calvin S Brown, “The relations between music and literature as a field of study. Comparative Literature,” 22(2):97–107, 1970.

[2] Pennebaker, J., Mehl, M., & Niederhoffer, K.,”Psychological aspects of natural language use: Our words, our selves. Annual Review of Psychology,” 54, 547-577, 2003.

[3] Pennebaker, J. W., Mayne, T., & Francis, M. E., “Linguistic predictors of adaptive bereavement. Journal of Personality and Social Psychology,” 72(4), 863-871, 1997.

[4] Fox, E. ,” Emotion science ,Basingstoke: Palgrave Macmillan,”, 2008.

[5] Strapparava, C., & Mihalcea, R.,”Learning to identify emotions in text, Proceedings of the 2008 ACM symposium on Applied computing,” 2008, New York, NY: ACM, (pp. 1556-1560).

[6] Emotion Analysis from Text: A Survey, IIT Bombay, India

[7] Sentiment Strength Detection in Short Informal Text, Statistical Cybermetrics Research Group, School of Computing and Information Technology, University of Wolverhampton, Wulfruna Street, Wolverhampton WV1 1SB, UK.

[8] Twitter as a Corpus for Sentiment Analysis and Opinion Mining, Alexander Pak, Patrick Paroubek Universit´e de Paris-Sud, Laboratoire LIMSI-CNRS, Bˆatiment 508, F-91405 Orsay Cedex, France

[9] Generating Music from Literature, Hannah Davis New York University, Saif M. Mohammad National Research Council Canada.

Acknowledgement

A project is an activity that needs involvement of a lot of people from different cadres and calibres. Hence it would be highly unfair on our part to claim credit for the whole project. With due respect, we express our sincere gratitude to our esteemed Principal Dr. Sincy George for this teaming experience and for giving us an opportunity to work on this project.

We sincerely want to thank our project guide Ms. Ankita Karia for her valuable guidance and advice she has given us when faced difficulties while working on the project. We are also grateful for her constructive criticism and for helping us develop this idea to pursue the project.

We would also like to thank our Class Co-ordinator Ms. Vincy Joseph for her notable inputs and the advice she has given us in our projects.

We would also like to thank our HOD Dr. Kavita Sonawane and the Computer Department staff who have helped us throughout this project. We are fortunate enough to get encouragement and support from all teaching staff of computer department. We would also like to extend our sincere regards to all the non-teaching staff of computer department for their timely support.