# **Chapter 4 System Analysis and Design**

# **Introduction**

System analysis and design focuses on the intrinsic and deep analysis of sentiment analysis, the detailed flow of entities, architecture and the design of the system. This is a comprehensive architectural overview of the system, using different analytic tools to depict different parties of the system. Sentiment analysis activities were executed in accordance to the chosen methodology (CRISP DM). Systems analysis and design deals with planning, analysis, design and implementation of the build system. Along the phases the developer deals with the planning of the information system through understanding and specifying in detail what the system should do and how the components of the system should be implemented and work together. Logically the preliminary stage of software development is to establish precisely and accurately what the users of the system want. The dominant part of this stage is communication between the users and the software developer. When the engineers is dealing with requirements, he/she is normally referred to as a systems analyst or, simply, “analyst”.

# 2. **System Analysis.**

System analysis focuses on the process of collecting factual data understanding the process involved identification of problems in the data and preparing the data for the model.

* 1. **Data Understanding.**

This project is based on the CRISP DM system development methodology. The six stages of the CRISP DM contains Data Understanding as the second stage of the development cycle. After the first stage of the cycle that called for understanding the project and its goals, the second stage guided in finding the correct dataset that will help to build the sentiment analysis model using Support vector machine - the model of choice.

The data that is suitable for the system is textual data. Textual data contains different words of which some contains sentiment value and others don’t. The dataset gathered was from Kaggle that was gathered from YouTube video comments.

The dataset was about 300 MB in size. The format of the dataset was .*csv* and it had 1048576 rows and 6 columns. The dataset was a labelled dataset and one of the columns contained the sentiment value of every sentiment. The others column contained the sentiments. The feature of the dataset being labelled made this dataset suitable for use in this project as it removed the need for labelling the dataset. It didn’t call for reading each and every sentiment and understanding it before loading the dataset in the model.

The dataset wasn't clean. It contained all types of symbols and short words that people use when communicating on daily basis. It also contained emoticons that people use to express their emotions. So there was need to deal with the emoticons too to make them valuable in this project.

Since the sentiment analysis is for video libraries, dataset from YouTube comments was really suitable as YouTube is also a video library. YouTube also contains videos from different subjects of life that makes the dataset suitable for generalised video libraries. The sentiment analysis isn't for a specific chapter of life, but for general videos.

* 1. **Exploring and Processing Dataset(Data preparation)**

The third stage of the CRISP DM is data preparation. Preparation of dataset is exploring and processing the dataset so that it fits perfectly well and also produces reliable model results. In trying to do so a number of processes are done that where much of data cleaning processes. All the data preparation stages were seen necessary after understanding the data structure in stage two of CRISP DM.

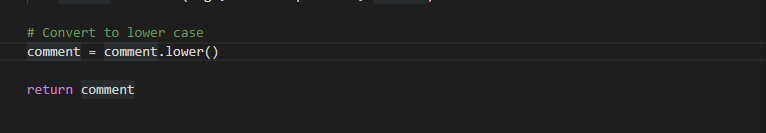
For a sentiment analysis to work well some data features must be removed. This will save time taken by the system analysing data since some features that are used when people are communicating contains no sentimental value at all. In trying to pave or remove features without sentimental value a number of processes were done to the dataset as the steps for cleaning it.

1. **Column deletion.**

The first data cleaning stage that was done was to remove some columns from the dataset. The dataset contained 6 columns of which the columns of value to the project where only two, the sentiment value column and the sentiment column. Other 4 columns where the date and time column, user id column, user name column and the other column which was written ‘NOQUERY’ throughout to the 1048576*th* column. The column removal was necessary as a way to reduce the dataset size.

1. **Lowercase.**

The dataset contained mixed text sizes that is capital letters and small letters. While the sentiment value between "LOVE" and "love" are the same. So to keep the dataset in a uniform format and to make sure a word in caps and in small caps are treated the same there was need to change the data to lowercase. To make this work *lower()* function from python was used as shown below



1. **Remove punctuation.**

Removal of punctuation symbols from the dataset was also necessary as the dataset contained a number of punctuation symbols. Punctuation doesn't add any extra information to the sentiment value. Removal of punctuation symbols reduce/ scales down dataset size and also improves computation efficiency of the model. For this feature *regex* and *replace* python function were used.

1. **Removing stop words.**

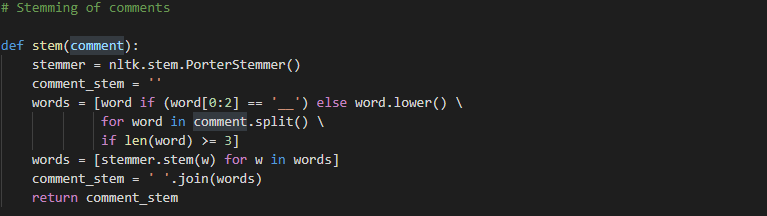
Removal of words that carries no sentimental value was a necessary step in this project. So stop words were also removed. Stop words are very common words that carry no meaning or less meaning compared to other keywords. If we remove the words that are less commonly used, we can focus on the important keywords instead. Making the model know best the well and commonly used words

1. **Tokenizing the data.**

Tokenising is splitting text into minimal meaningful units. There is a sentence tokenizer and word tokenizer. The one that was used in this project was a word tokenizer. The reason for using the word tokenizer was to ensure the model uses perfectly tokenized words, this improves model accuracy. Though this valued accuracy over time and CPU overuse. For this task *trim()* method from *nltk* package was used. The reason was its simplicity approach to tokenizing data.

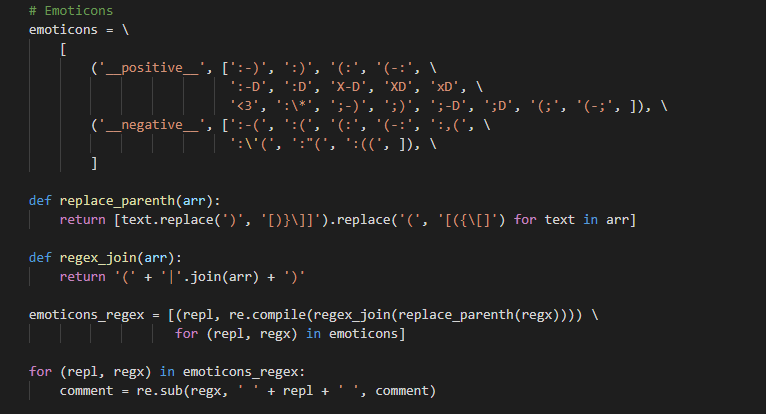
1. **Stemming.**

Stemming was used so as to try to extract root words from the words that were contained in the dataset. Words like "like", "likes" and "liked" were being stemmed into "like". Stem function from *nltk* library was used for this feature.



1. **Emoticons.**

The dataset contained sentimental value and the sentiment value need to be extracted as well. For this feature a small dictionary was created for emoticons that are positive and negative. Then the sentiments in the dataset were being compared with the small created dictionary as shown below.



**2.3 Feature Engineering.**

Feature Engineering is the procedure of converting raw text data into machine understandable format (numbers). For this project there was need to convert the dataset into a format that the model will understand before passing the data into the model. Different feature extraction methods can be used that include TF-IDF, One Hot Encoding, N-grams, Count Vectorizer and Hash Vectorize.

Each of the above methods have got an area in which they suit best and for the sentiment analysis project TF-IDF (Term Frequency- Inverse Document Frequency). The suitability of the method is measured based on the type of the data and the structure of the data. The feature that need to be engineered so as to end up with highly accurate results in the model also helps in considering the method.

The problem that was in the dataset was the repetitive occurrence of the same words. For example the repetition of the word "love" throughout the dataset. This was causing a bias in the outcome of the model since a repetitive appearance of a word achieve higher importance in other methods that was first tried that are Count Vectorize and Co-occurrence. TF-IDF deals with this problem to avoid repetitions of words overpowering other words with sentiment value within a statement.

TF-IDF is a combination of TF and IDF. Term Frequency is the ratio of the count of a word present in a sentence to the length of the sentence. TF just basically captures the importance of the word with respect to the length of the document. This implies that a word with frequency of three within a sentence with length ten is not the same as when the word length of the sentence is 100 words. It must get more importance.

Inverse Document Frequency (IDF), of each word is the log of length of the total number of rows to the number of rows in a particular document in which that word is present. IDF measures the rareness of a term. That is a word is appearing in almost all documents, then that word is of no use to us since it is not helping to classify or in information retrieval. IDF will nullify this problem.

# **System Design**

System design focuses on designing the architecture of the interfaces, the modules and the data for the system to satisfy the required requirements of the product/ system (Hull et al, (2017). This is now the application of all the theories about the system into development of the product. The aim of the design phase is on how to accomplish the system objectives.

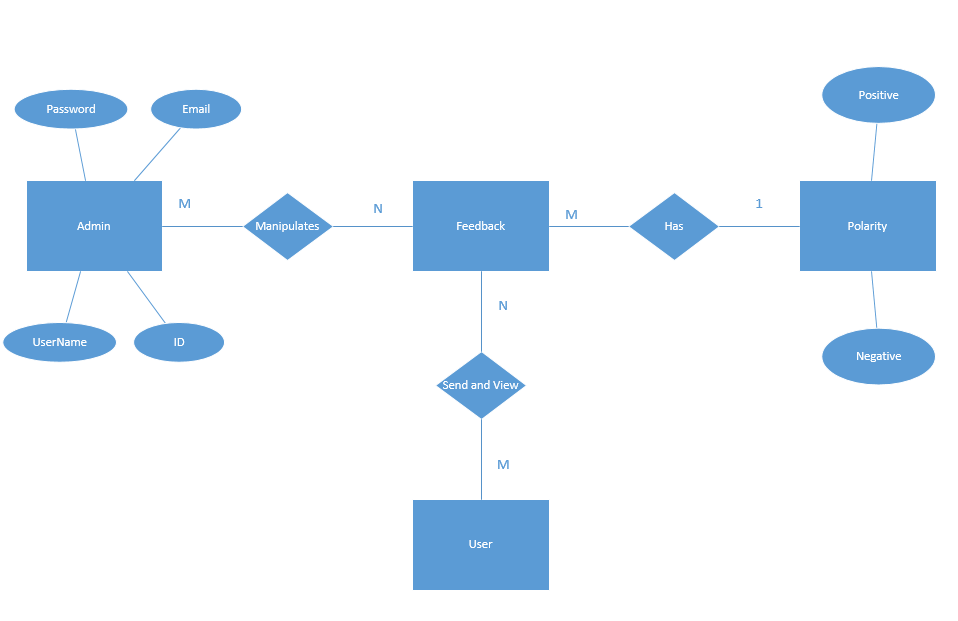


# **Database design**

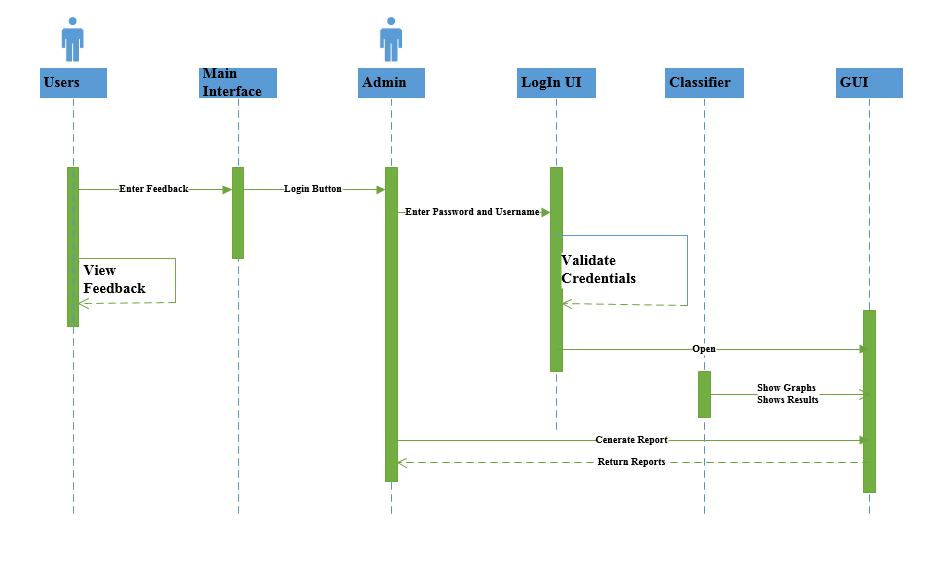
By database design we will be trying to produce a detailed model of the database. This include all the logical and designing choices and the physical storage parameters needed for generating a designing a data definition language. The data definition language is the one that can be used to generate or create a database.

* 1. **Entity - Relationship Diagram**

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. The elements of an ERD are entities, relationships and attributes. The Entity relationship diagram below shows the entities that are modelled to the system database and the association between the entities.



* 1. **Sequence Diagram**



# **System Modelling**

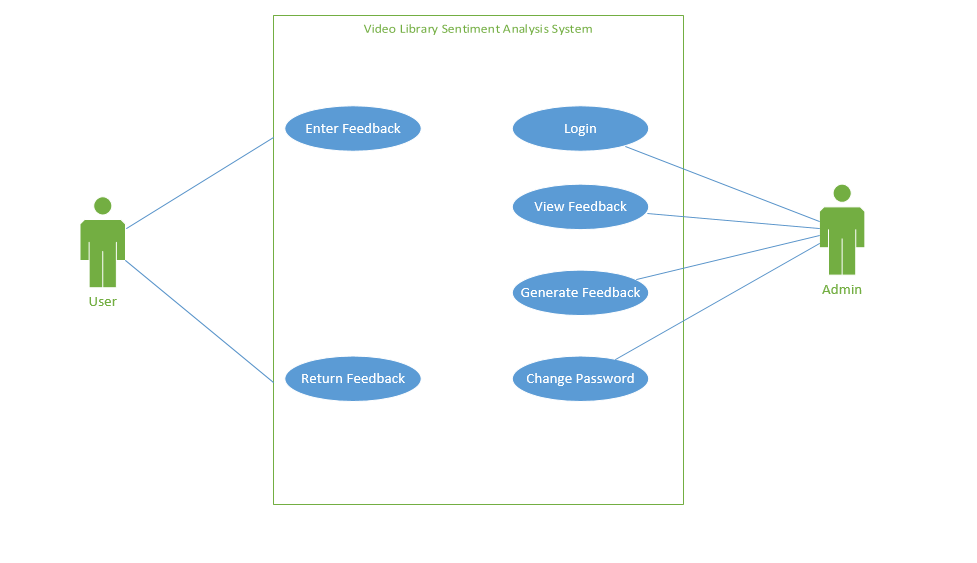
System modelling helps to visuals how the system work and interacts with external systems and objects. The modelling tries to visualise the requirements of external entities separately from the internal designs. Once there is a visualization of how external objects and internal objects will interact this aid in the development of the system as a whole. The easier it is to develop the sentiment analysis system the faster objectives are met. In other words system modelling marks the start of the implementation and testing phase of the software development methodology. According to the methodology chosen (CRISP DM) the stage of modelling helps to see if the project was clearly understood. If well understood then proceed to evaluation stage, on condition that the modelling stage is completely done. If the modelling stage is failing to come out well then a push back to understanding the project objectives is required.

The models will be used as a basis for system tests, making a clear relationship between the tests and the objectives and when the objectives change, this relationship is used to update the test

* 1. **Use case diagram**

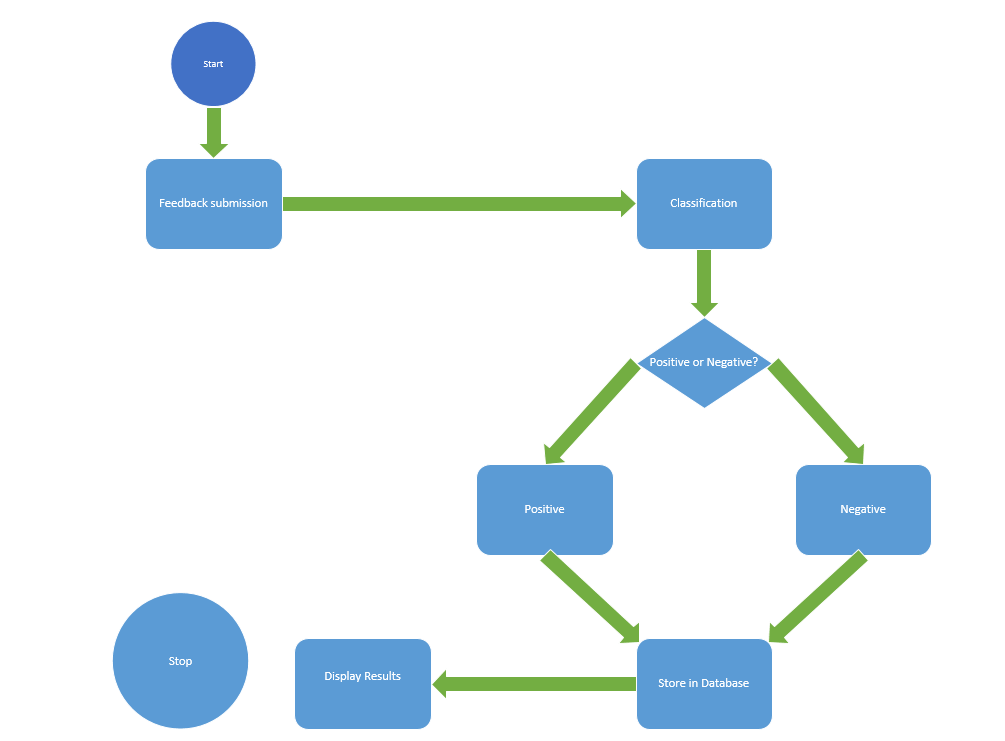
System have got behaviours. The behaviour of the system need to be visualise. One of the tools that visualise the behaviour of the system best is use case diagrams. Use case diagrams is an object oriented model construct.

Interactions between the user and the system are described through a prototypical course of actions along with a possible set of alternative courses of action (Grechanik et al, 2007). The use case diagram of the video sentiment analysis system is shown in figure below.



* 1. **Activity Diagram**

Events in a system need also to be clarified. To show them clearly an activity diagram can be used. Unlike other diagrams this helps to visualise activities. It gives the overall view of the system showing all the possible results of an event occurring and how they tangled. Below is an activity diagram of the sentiment analysis as whole for a standard sentiment analysis.



1. **Conclusion**

This chapter discussed the system requirements elicitation process which is the most important stage in the software development process as it ensures that a system being developed fully satisfies the client needs. With the help of UML diagrams were also incorporated to define and discuss the analysis and design concepts of the system using use-cases, class diagrams, sequence diagrams and activity diagrams which convey the structure and functionality of objects in a way that can be better understood by the developers. The analysis and the design phases provided the foundation for the implementation of the system.