

Detecting Fake Reviews

Software Requirement Specification (SRS)

Amira Ashraf, Marim Mohamed, Yara Ahmed Nader, Yasmine Maged

Supervised by Dr.Ammar Mohammed, Eng.Huda Mohammed

October 2018

1 Introduction

1.1 Purpose of this document

The purpose of this software requirements specification document is to present the requirements to detect Fake reviews by filtering it from authentic data review .Since reviews on products have critical impact on the product's reputation and service providers,So we detect the fake positive and fake negative review from the real reviews.

1.2 Scope of this document

System will detect fake reviews by using some machine learning algorithms according to:

1. Text pre-processing.
2. Extract the feature from text reviews, user behavior and user account.
3. Classify the reviews.

1.3 Overview

Fake reviews are the reviews which is designed specifically to give a false impression to consumers on the point of purchasing. The problem of fake online reviews not only concerns individual consumers but it can also lead to an erosion of consumer confidence in the online market, which can reduce competition. There are two types of fake reviews. A positive fake review which can give you a false image to persuade you to buy a product or a service and a negative fake review which is written for to damage the company's reputation and go on to recommend a different product from a competitor. This document proposes "Detecting user fake reviews" to provide a real positive and real negative reviews system.

1.4 Business Context

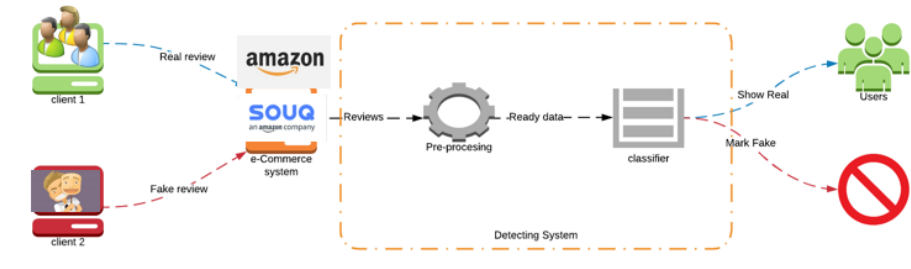


Figure 1: System Overview

1. We have two clients one who wrote a real review and another one who wrote fake review.
2. The reviews will be shown at the e-commerce system.
3. Detecting System:
 - (a) Passing the reviews to the pre-processing phase which is preparing the data for the classifier.
 - (b) Then classify the data either it is fake or real review.
4. The real reviews will be shown to the users of this e-commerce system and the fake reviews will be marked as fake and sent to another file.
5. We contact someone who works at Jumia who offers help and loved our project idea.

2 General Description

2.1 Product Functions

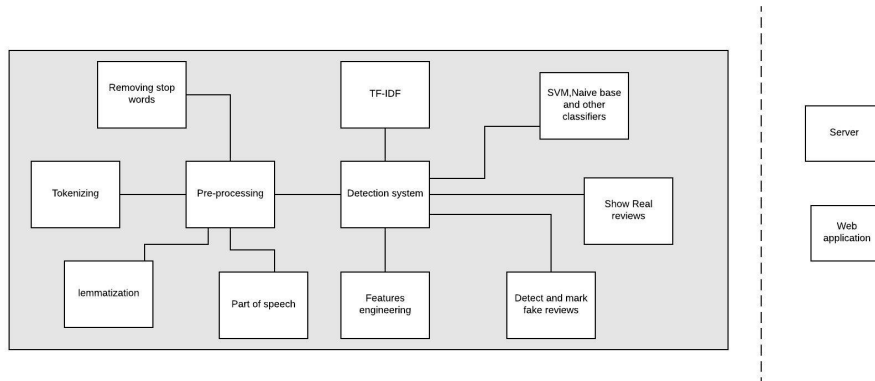


Figure 2: System context

1. Pre-processing
 - (a) The system will split text into separate words.
 - (b) System removes all the stop-words to filter the document.
 - (c) System returns back the base or dictionary form of a word.
2. Feature extraction
 - (a) The system will extract the feature from the text.
 - (b) System will use the TF-IDF to convert each word into her weight.
3. Classification
 - (a) System use SVM and Naive Bayes.
4. Admin
 - (a) The admin can add o customer, product, data consultant and respond for verification.
 - (b) Admin can search for any product and all customers whom writing the reviews.
5. Customer
 - (a) Customer can add , edit and delete his review.
6. Data consultant
 - (a) He is responsible for training the model ,detect real fake reviews and send verification to admin for marking the fake.

2.2 Similar System Information

The system is a web application in which will be able to detect negative deceptive opinion from Tweets. The system is a stand alone application .This paper is a bit similar to our application as it use the personal profile and behavioral features to detect the deceiver.

2.3 User Characteristics

The system will be used by large e-commerce companies, their are two types of users both types of users have different use of the system .

Data Consultant:

- Must have domain knowledge and skills to use the computer.
- Must have the Knowledge to use database .
- Must be able to interact with UI of the system to add, delete and edit records

Admin:

- Must be able to work with databases
- Must be able to manage databases through the user interface that provides with the system.
- Must be able to monitor the security and integrity of data.

2.4 User Problem Statement

Fake reviews affect the e-commerce company's reputation by giving untrusted reviews that affect customer decision to buy the product. So we propose the system that detect fake reviews which will help in giving accurate data about the product to the customer.

2.5 User Objectives

The data consultant will be able to see the classification of reviews if its real or fake review and then mark them by the permission of the admin. The system will save the data consultant time and effort for detecting the fake reviews as the system need only the reviews dataset and then classify them without any external help.

2.6 General Constraints

The main constraint of the system is to have a review of a new customers

3 Function Requirements

Table 1: FR1: postReview.

Description	Customer write his opinion whether it's honest opinion or fake , after clicking the post button our system will start running and call the preprocessing FR4,FR5 and FR6 then pass it to the features extraction FR10,F11.get features and pass it to classifiers returning the detected fake reviews in separate file by FR2 without showing it on the e-commerce system.
Input	review text ,Product ID
Output	review loaded successfully or no new reviews to post
Pre-condition	no review text has been uploaded
Post-condition	review Successful upload without failure
Dependencies	(login)
Criticality	10/10

Table 2: FR2: FakeReviewFile.

Description	after detecting reviews in FR1 the fake reviews will be saved in file as an output of FR1 to be checked data consultant
Input	review info
Output	deleted successfully or failed
Pre-condition	the list of all the real and the fake reviews
Post-condition	delete fake review Successful without failure
Dependencies	FR1:(Post)
Criticality	10/10

Table 3: FR3: CustomerReview.

Description	Data consultant can list all the reviews have been written by the specific customer.
Input	customer ID or email
Output	all the customer reviews records
Pre-condition	no show of the list of reviews
Post-condition	show the updated list of reviews for this customer
Dependencies	None
Criticality	7/10

Table 4: FR4: TokenizeText.

Description	It takes each review and splits it to array of words "Tokens " also removing all the spaces and the breaks in each review ,for simplifying the review processing.(NLTK built in function)
Input	Review Text
Output	Array of words
Pre-condition	text in it's original form
Post-condition	it gets text without spaces or breaks
Dependencies	FR1:(PostReview)
Criticality	10/10

Table 5: FR5: RemoveStopWords

Description	It take the array of words output from FR5 and removes any stopword such as :("as", "the", "a") from the text as a kind of filtration.(NLTK built in function)
Input	array of words
Output	array of words without any stopwords
Pre-condition	text in it's original form
Post-condition	it gets text filtered from stopwords
Dependencies	FR4:(TokenizeText)
Criticality	10/10

Table 6: FR6: lemmatization

Description	It take the array of words output from FR6 and get the single forms of derived words such as:(”students ”:”student”). (NLTK built in function)
Input	Array of words of the review
Output	array of words consist the singular of each word
Pre-condition	text in it’s original form
Post-condition	it gets array of words with it’s single or If it gets confused between words
Dependencies	FR5:(removeStopWords)
Criticality	10/10

Table 7: FR7: listUser.

Description	It list all the system users by the admin , by checking the user id (admin Id) to access this function if he is the admin only then listing all the users in details
Input	user ID
Output	all the system users
Pre-condition	no show of the user records
Post-condition	show the users records
Dependencies	(login)
Criticality	7/10

Table 8: FR8: searchUser.

Description	Function takes user name or his Id to loop through the system users and select this specific user ,this function must be run by the admin
Input	user name or ID
Output	user info
Pre-condition	having the name or Id of the user
Post-condition	Give the Name or the Id to the function and it Search by the admin then return the Record the admin wants
Dependencies	(login)
Criticality	7/10

Table 9: FR9: Ngram.

Description	Function takes the output of the preprocessing functions :FR4,FR5,FR6 .Then extract the feature of Bigram in each review by generate such word pairs from the existing sentence and maintaining their current sequences.
Input	the array of words
Output	feature file
Pre-condition	the array of words in it's preprocessing form
Post-condition	taking each pair of words with each other respecting to their sequence.
Dependencies	FR6:lemmatization
Criticality	10/10

Table 10: FR10: TF-Idf.

Description	Function takes the output of the preprocessing functions ,then extract the feature in each review by calculating the frequent words (TF) and calculate the ratio bet. the frequent term and total no. of doc words (idf)
Input	feature file
Output	TF-IDF featurefile
Pre-condition	feature file without calculating TF-IDF
Post-condition	it gets the frequent terms in the dataset then giving them higher weight
Dependencies	FR9:Ngram
Criticality	10/10

Table 11: FR11: partOfSpeech.

Description	It takes the output of the function FR6 ,then categorize syntactic analysis(noun-adjective-verb) each word in the dataset.
Input	Array of words of the review
Output	array of words consist the origin of each word
Pre-condition	text in it's original form
Post-condition	Categorize each word to its syntactic analysis(noun-adjective-verb).
Dependencies	FR6:lemmatization
Criticality	10/10

4 Interface Requirements

4.1 User Interfaces

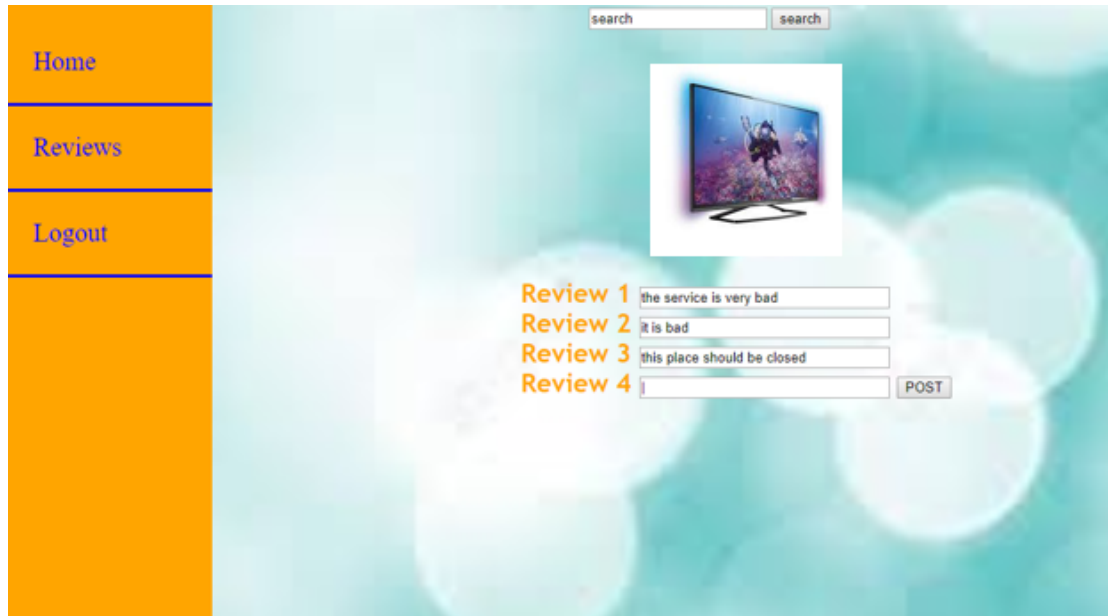


Figure 3:



Figure 4:

4.1.1 API

- Natural Language Tool Kit Library (NLTK): it is used for the text (reviews) preprocessing for filtration, tokenization and lemmatization

- The NLTK Version is 3.3

4.2 Communications Interfaces

In this part,the system only needs internet connection.

5 Performance Requirements

The system requires specific hardware computational power to perform in the best way, (8GB RAM,Intel core family (i5-i7)).

6 Design Constraints

6.1 Standards Compliance

IEEE Standards

6.2 Hardware Limitations

No strict hardware limitations as its a web application so, it works on the devices that supports the internet access.

6.3 Software Language

Python and HTML5 will be used.

6.4 Implementation Constraints

The implementation constraint is that the system is a web application.Therefore it will be available on the devices that supports internet access.

7 Other non-functional attributes

7.1 Scalability

Scalability is the ability to appropriately handle increasing and decreasing of reviews or users. So, this will let the system to work on large data or small data.by using MYSQL

7.2 Portability

This system's functions are all written in python language. Python was used as it runs on unix , mac , windows , web applications and android. So it is very portable. Also this function will be used in the change of architecture MVC.

7.3 Reliability

This system trains on a huge dataset which raises the accuracy of classification. so the system should be assured that result is precise.

8 Preliminary Object-Oriented Domain Analysis

8.1 Inheritance Relationships

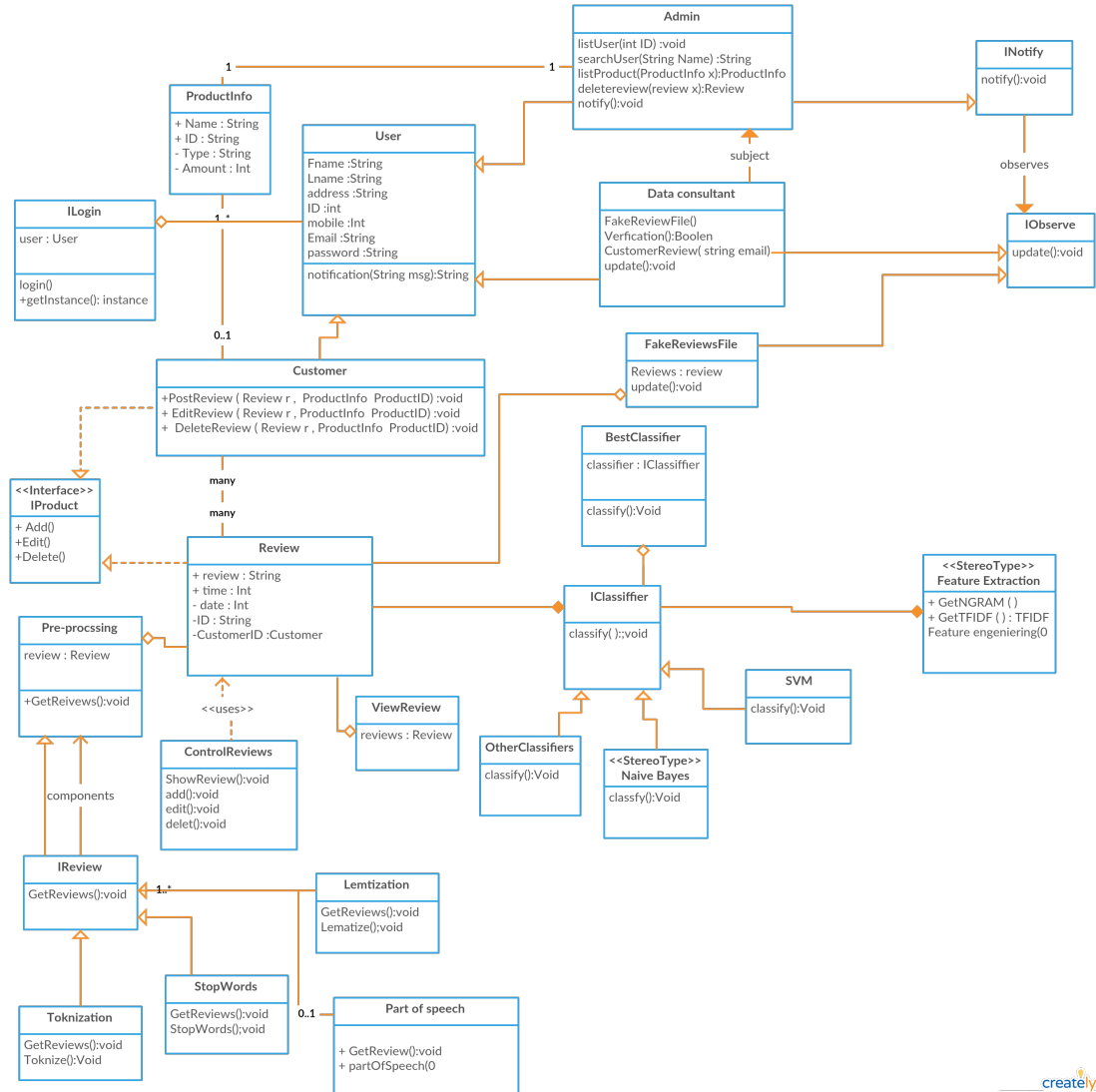


Figure 5: class diagram

8.2 Class descriptions

8.2.1 User

List of super-class	None
List of sub-class	Admin,Data consultant,Customer
Purpose	The user inputs the reviews to detect them to get the fake and real reviews
Collaborations	Inherited by class admin , data consultant and Customer
Attributes	fname,lname,address,ID,email,mobile,password
Operations	notification(msg)

8.2.2 Customer

List of super-class	None
List of sub-class	None
Purpose	Write , edit and delete reviews on product
Collaborations	inherit from class user assisted by review class and associates ProductInfo and review classes
Attributes	None
Operations	PostReview(Review r, ProductInfo ProductID),EditReview(Review r , ProductInfo ProductID),DeleteReview(Review r , ProductInfo ProductID),

8.2.3 ProductInfo

List of super-class	None
List of sub-class	None
Purpose	It has all the product information
Collaborations	assist admin and customer classes
Attributes	Name,ID,type,amount
Operations	None

8.2.4 IProduct

List of super-class	None
List of sub-class	None
Purpose	contains add , edit and delete operations on reviews
Collaborations	customer and review implements this class
Attributes	None
Operations	add(),edit(),delete()

8.2.5 Admin

List of super-class	None
List of sub-class	None
Purpose	runs the online store
Collaborations	inherits from class user and INotify and assisted by ProductInfo
Attributes	None
Operations	listUser(int ID),SearchUser(String Name), ListProduct(ProductInfo x),notify()

8.2.6 INotify

List of super-class	None
List of sub-class	Admin
Purpose	notifies the admin when any change happens
Collaborations	inherits Admin class
Attributes	None
Operations	notify()

8.2.7 IObserve

List of super-class	None
List of sub-class	DataConsultant , FakeReviewsFile
Purpose	updates when there is a change
Collaborations	inherits DataConsultant and FakeReviewsFile class
Attributes	None
Operations	update()

8.2.8 FakeReviewsFile

List of super-class	None
List of sub-class	None
Purpose	stores all the fake reviews that have been removed from the website
Collaborations	inherits aggregates from Reviews class and inherits from IObserve
Attributes	None
Operations	update()

8.2.9 review

List of super-class	None
List of sub-class	None
Purpose	takes reviews from database
Collaborations	assists customer , implements from IProduct class and composes Iclassify review and aggregates pre-procssing and FakeReviewsFile
Attributes	review,time,date,ID,customerID
Operations	None

8.2.10 ControlReviews

List of super-class	None
List of sub-class	None
Purpose	Have all the functions that is used in the review class
Collaborations	uses review class
Attributes	None
Operations	showReview(),add(),edit(),delete()

8.2.11 ViewReviews

List of super-class	None
List of sub-class	None
Purpose	outputs the results of control class
Collaborations	aggregates from review
Attributes	reviews
Operations	None

8.2.12 Data consultant

List of super-class	None
List of sub-class	None
Purpose	It detects the reviews
Collaborations	inherits from class user and IObserve
Attributes	None
Operations	FakeRviewsFile(), CustomerReview(ID), update()

8.2.13 Pre-procssing

List of super-class	None
List of sub-class	IReview
Purpose	prepares the data
Collaborations	aggregates from review class, inherits IReview
Attributes	review
Operations	GetReviews()

8.2.14 IReview

List of super-class	Pre-procssing
List of sub-class	Tokenization,stopwords,lentization
Purpose	Have the getreviews functions which gets all the reviews from dataset
Collaborations	inherit from Pre-procssing
Attributes	None
Operations	GetReviews()

8.2.15 Toknization

List of super-class	None
List of sub-class	None
Purpose	splits reviews into separated words
Collaborations	inherit from Pre-procssing
Attributes	None
Operations	GetReviews(),Toknize()

8.2.16 StopWords

List of super-class	None
List of sub-class	None
Purpose	remove all the stopwords
Collaborations	inherit from Pre-procssing
Attributes	None
Operations	GetReviews(),Stopwords()

8.2.17 Lemtization

List of super-class	None
List of sub-class	None
Purpose	change the plural words into single
Collaborations	inherit from Pre-procssing
Attributes	None
Operations	GetReviews(),lemtize()

8.2.18 Iclassify

List of super-class	None
List of sub-class	Naive Bayes , otherClassifiers , SVM
Purpose	it classifies the reviews by using some machine learning algorithms
Collaborations	svm and naive bayes implements this class aggregates best-classifier class
Attributes	None
Operations	classify()

8.2.19 SVM

List of super-class	Iclassify
List of sub-class	None
Purpose	it classifies the reviews
Collaborations	implements from Iclassifier
Attributes	None
Operations	classify()

8.2.20 Naive Bayes

List of super-class	Iclassify
List of sub-class	None
Purpose	it classifies the reviews
Collaborations	implements from Iclassifier
Attributes	None
Operations	classify()

8.2.21 OtherClassifiers

List of super-class	Iclassify
List of sub-class	None
Purpose	it classifies the reviews
Collaborations	implements from Iclassifier
Attributes	None
Operations	classify()

8.2.22 BestClassifier

List of super-class	None
List of sub-class	None
Purpose	it classifies the reviews with the best algorithm taken after comparing between all used algorithms and taking the best one with highest accuracy
Collaborations	aggregates from Iclassifier
Attributes	classifier
Operations	classify()

8.2.23 Feature Extraction

List of super-class	None
List of sub-class	None
Purpose	it is used to remove unneeded attributes from data that may actually reduce the accuracy of the model
Collaborations	composes from classification class
Attributes	None
Operations	GetNGRAM(), GetTFIDF() , FeatureEngineering()

8.2.24 ILogin

List of super-class	None
List of sub-class	None
Purpose	makes user login
Collaborations	aggregates from user
Attributes	user
Operations	login(),getInstance()

8.3 Block Diagram

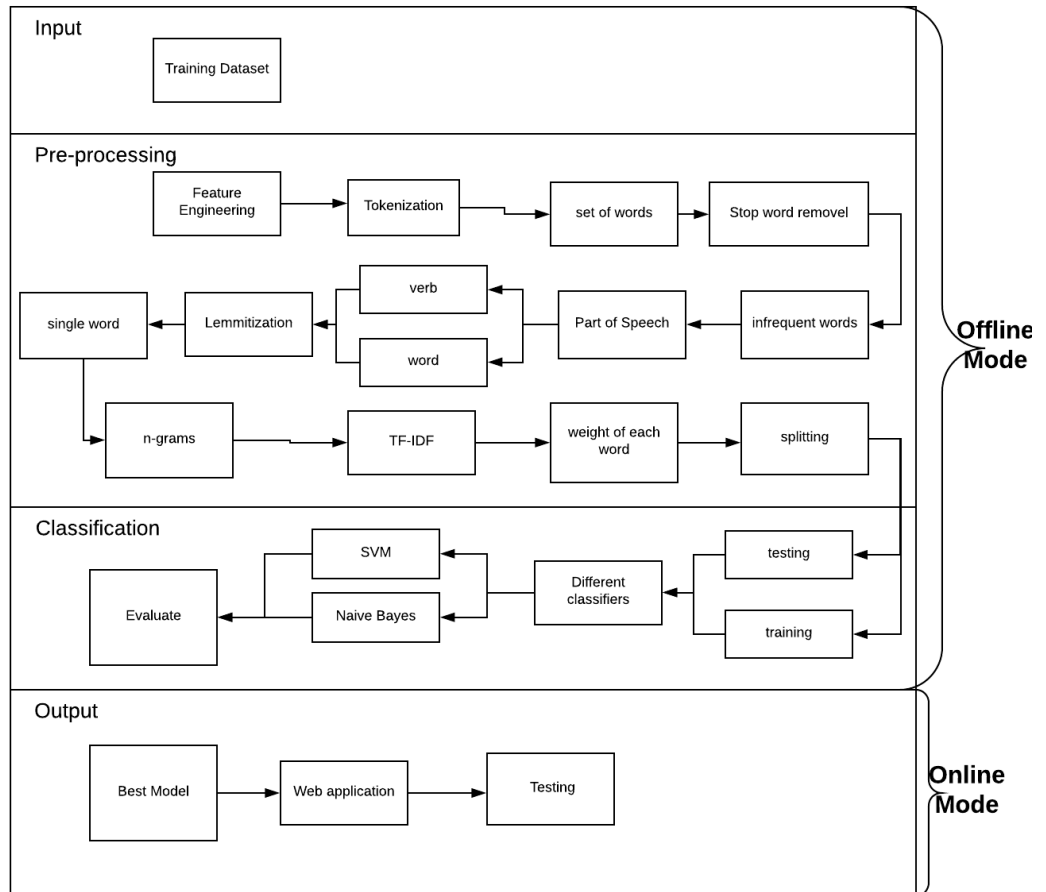


Figure 6: Block Diagram

9 Operational Scenarios

9.1 Users scenario

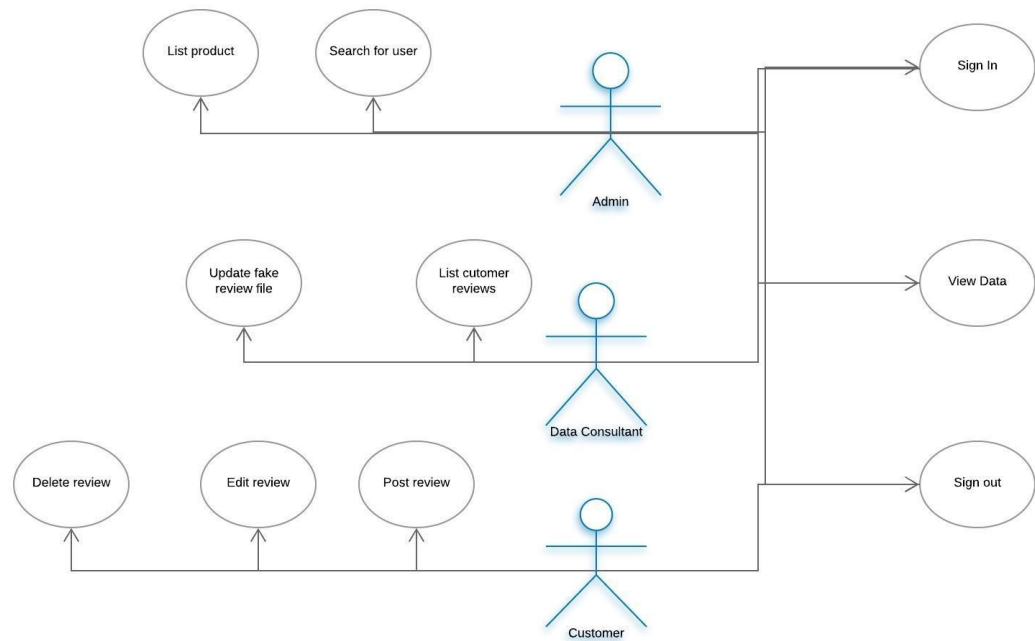


Figure 7: Users scenario

In this system, the user must have an account to sign in, so the first thing is registration to have a username and password for each user. The customer can post his review and have the ability to edit or delete it. The data consultant after logging in he can list all the users and update the fake reviews file by adding the new fake reviews that detected by the system. Also the admin after logging in he can search for any user in the system and see his history.

9.2 System scenario

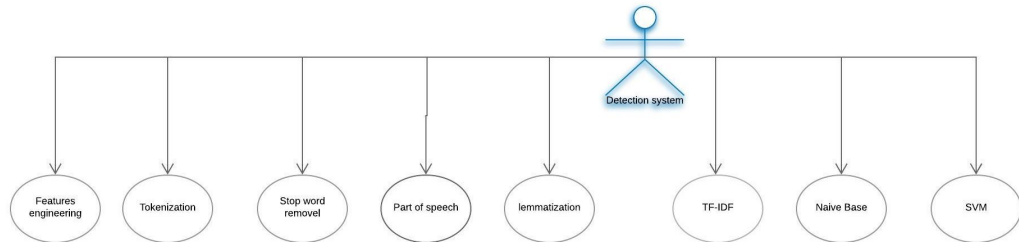


Figure 8: System scenario

Files are entered the system to firstly preprocessed by tokenization then stop word removal then apply part of speech to make system differentiate between noun , verb and adjective and finally we apply lemmatization .Then Implementing feature engineering on the user behavior and the user history and also feature extraction process will occur to the words which is being weighed by TF-IDF to determine the highly weighted words and to determine lexical features of review and then determine the personal profile of the writer. finally the classification process which will be performed by SVM classifier and Naive base Classifier to determine real reviews from fake one's.

10 Preliminary Schedule Adjusted

Task Name	Start	Finish
Ideas of graduation projects	7/2/2018	7/15/2018
Idea Research	7/16/2018	9/1/2018
Survey and Proposal	9/1/2018	9/18/2018
Rehearsal of proposal	9/19/2018	9/19/2018
Proposal Presentation	9/26/2018	9/26/2018
Designing system	9/27/2018	10/1/2018
Designing Class Diagram	10/11/2018	10/18/2018
Submitting Survey Paper	10/20/2018	10/20/2018
SRS Writing	10/22/2018	11/9/2018
SRS presentation	11/14/2018	11/14/2018
External Examiner	12/3/2018	12/11/2018
SDD Writing	1/25/2019	1/30/2019
SDD Presentation	2 week of Feb	2 week of Feb
IMPLEMENTATION	2/25/2019	3/15/2019
IMPLEMENTATION Evaluation	After spring break	After spring break
Writing Paper		
Delivering Paper	3 days after Spring Vacation	
TECHNICAL EVALUATION	1st week of may	
Writing Thesis	5/9/2019	5/15/2019
Final Presentation	25-Jun-19	

Figure 9: comparison

11 Preliminary Budget Adjusted

we will use IBM Blumix cloud services (server-storage) to store the database run the system .