X-GEN

5/24/2019

Final Report of Spam Detection in Hotel Reviews

P. Thineshan(160633A)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Table of Contents

ABSTRACT SUMMARY

Some people leave fake reviews for products. There are many reasons such as to improve the product's ranking, to improve the seller's ranking and to balance negative reviews left for a product. So, spam detection is very important thing in this situation.

Commonly, in a hotel management system or other systems users use online reviews to make decisions about available products and services. Sometimes fake reviews are created. So, spam detectors help to find these fake reviews. This spam detection system finds fake reviews in a set of reviews. System is trained for find fake reviews. NLTK model classifier classify these fake reviews. So, a hotel management easily found fake reviews using this spam detector

Design a spam detection web application and detect review is spam or not spam. Input of the proposed system is a hotel reviews and get output as spam or not

1. INTRODUCTION

1.1 BACKGROUND OF THE APPLICATION DOMAIN/ PROBLEM

The background of this project is to develop a spam detection software to find fake reviews of the hotel reviews. The system trains a dataset of reviews and classify the fake reviews in hotel reviews. This will be very useful to many hotel management systems. User gives reviews about a hotel and system get user input and classify it as spam or not spam reviews and training to a given dataset and detect spam result.

1.2 MOTIVATION FOR THE SELECTED SYSTEM DEVELOPMENT

Users view their reviews with spam results and one user can't view other user reviews. When user register their hotel then automatically shows login page. No one cannot enter without login.

The main objectives of this project is,

1. Design and implement a spam detection user friendly web application
2. Provide a model to find fake reviews in a set of reviews
3. Automate classify particular review is spam or not spam

1.3 IMPORTANCE AND MAIN PURPOSE OF THE SYSTEM

There are many reasons why people leave fake reviews but, in short, the main reason is to increase hotel permutation. The main reasons for leaving fake reviews are to improve the hotel ranking, boost visibility of services which have just been listed, balance negative reviews left about hotel and advertise hotel services

Customers very disappointed in these fake reviews. So, spam detection is important thing in this situation. Spam detection system find fake reviews and remove it then customers easy to make decisions about hotels

1.4 OVERVIEW OF THE SYSTEM

Many refer to reviews prior to making a hotel reservation. Thus, these reviews can affect a brand positively as well as negatively. Design a spammer detection which can identify fake reviews in a hotel reservation site (eg: Trip advisor), overcoming the limitations of existing methods.

2. LITERATURE REVIEW

There are some other platforms which are providing this spam detection application. But the
facilities are restricted for the users.

ReviewMeta gather publicly available review data from platforms such as Amazon and Bodybuilding.com. This focus to collect set of reviews and also looking at the reviewers who are behind those reviews. ReviewMeta perform an analysis and run twelve different tests and use statistical modelling on any suspicious patterns for detect fake reviews.

- For the web application I plan to use Nodejs backend and bootstrap front end or other ways.
- I use natural language toolkit to automate classification of spam detection.

There are many spam detection applications like https://github.com/iliyaML/naive-bayes-spam-detector

3. SYSTEM MODELS

3.1 SYSTEM REQUIREMENT

3.1.1 FUNCTIONAL REQUIREMENTS

There will be lot of requirements related to the user roles and privileges.

1. *Spam detection of the hotel review*

   The system should be able to get the preprocessed review given by the user in order to process it to find the review spam result.

2. *View result of the review*

   The system should view the output as spam or not spam using an API.

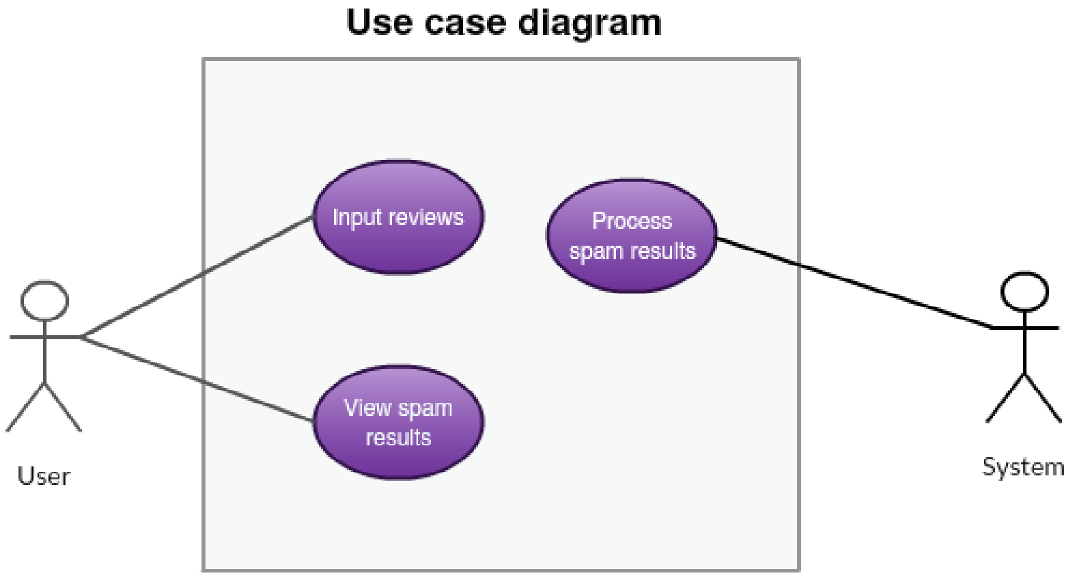
3. *System get the review*

   The system should get the user input from user interface of spam detector.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

- System availability main important non-functional requirement. The spam detection web application will be hosted through online server and it is available any time for any users. There are some server-side problems other way system will haven't limitations for the users.
- Response time also another important requirement. The time taken by the system to generate spam result usually depends on the availability of an existing pretrained modal or the need to train a modal for a new area. Time taken for the web application depends on train modal system. So, system used different classifiers and train the modals. Then system will generate output quickly.
- Friendly and attractive user interface is an important thing in a web application. Every user can easy to handle the user interface. User interface is very attractive and no ads, no unwanted pops up or other unwanted mages will be shown. So, user interface is very user friendly.
- Users don't know about technical knowledge (programming languages) or other extra skills. Basic knowledge of using a web application will be enough to use the system.
- User need an internet connection to reach this web application and no need for the configuration in this system. So, user can use this system easily.
- As it is a web-based system, any users can access this system using their web browsers easily. System is developed to respond to all popular web browsers. The system should be able to design all sizes of screens such as mobile phones and PCs.

3.1.3 USECASE DIAGRAM



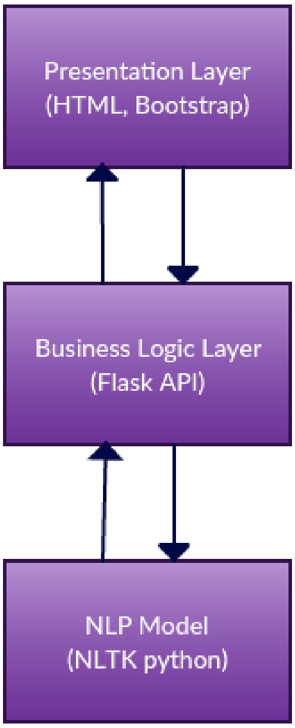
Use case name	Input reviews	
Actor	User	
Description	When a user wants to find the fake reviews then user has to input the reviews	
preconditions	-	
Main flow	User	System
	1. Go to the web application 2. Input review	1.1 Display web application page 2.1 Get the review 2.2 Detect the result 2.3 Display result and give access to next review
Successful end/post condition	Review must be upload to the system Model must be trained	
Fail end/post condition	Give access to re-input the review	
Extensions	N/A	

Use case name	View spam results	
Actor	User	
Description	When system detect the spam result then user want to see the results.	
preconditions	-	
Main flow	User	System
	1. Input review 2. View result	1.1 Detect the spam result 1.2 Display result
Successful end/post condition	Review must be upload to the model Model must be trained	
Fail end/post condition	-	
Extensions	N/A	

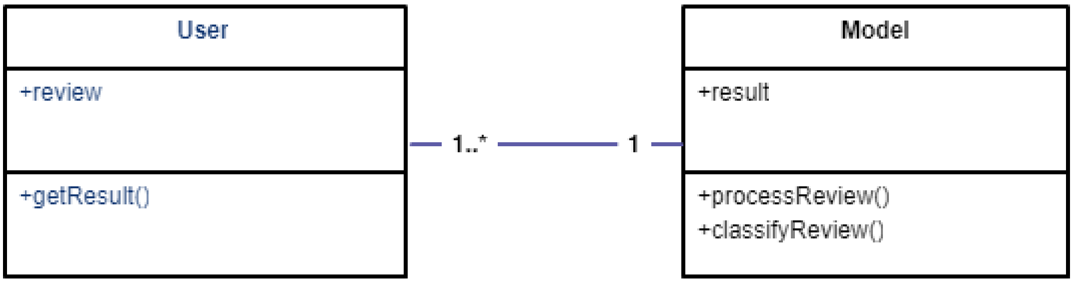
3.2 SYSTEM DESIGN

3.2.1 SYSTEM ARCHITECTURE

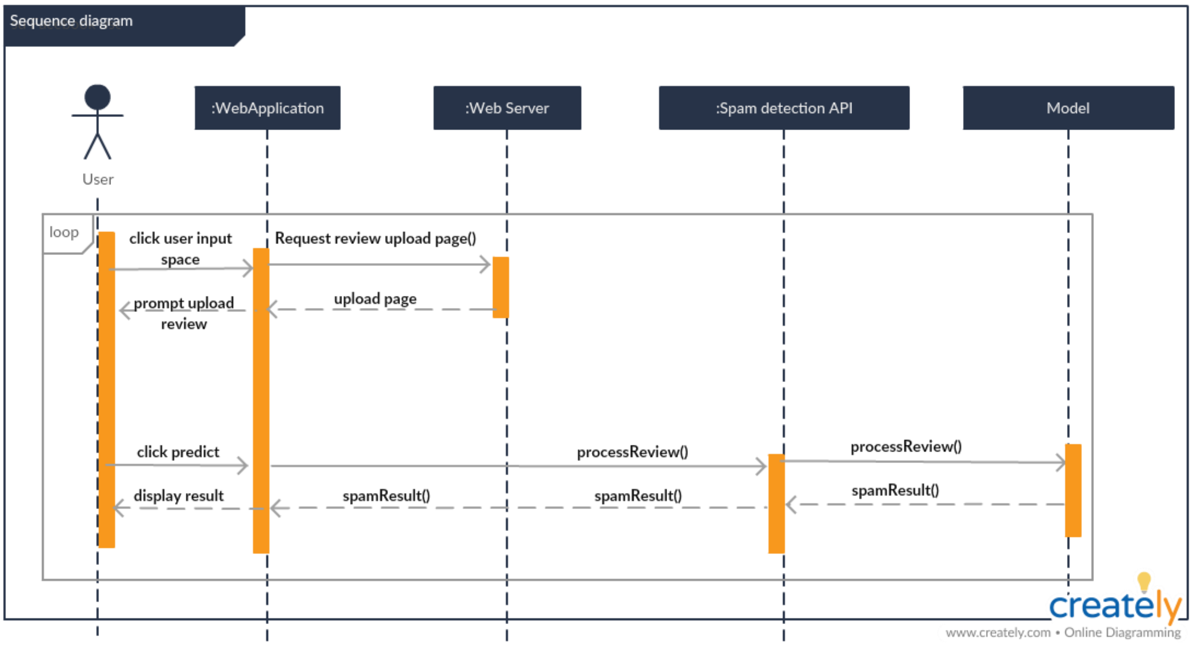
Following diagram will express the logics view of the architecture. We have analyzed most part of the system and future risks to finalize this following diagram. System will be developed according to this logical architecture diagram. The logical view is concerned with the functionality that the system provides to end-users. UML diagrams used to represent the logical view include class diagrams and state diagrams.



3.2.2 CLASS DIAGRAM

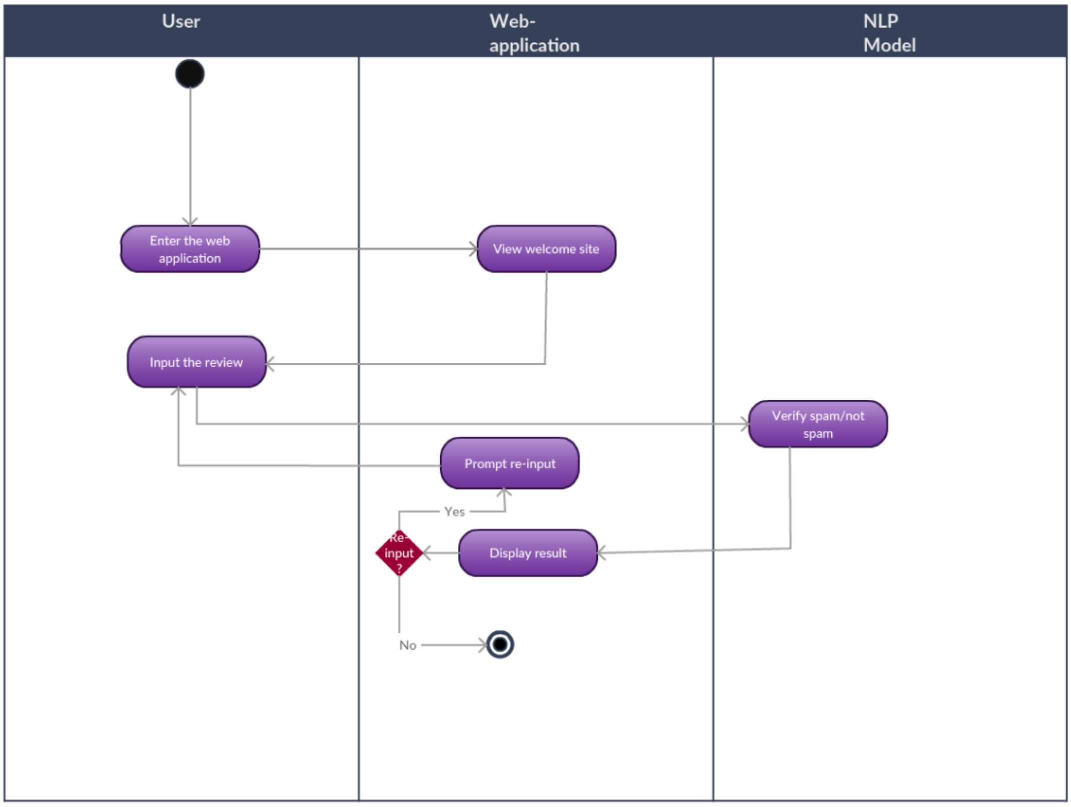


3.2.3 SEQUENCE DIAGRAM



3.2.4 ACTIVITY DIAGRAM

Activity Diagram



3.3 DATABASE DESIGN

Database has user and review tables. In user table has user details with user id. In review tables every user (hotel) reviews with user id. we can view each hotel reviews with the use of this user id (user id is foreign key in review table). In this way designed the database schema of spam detection web app.

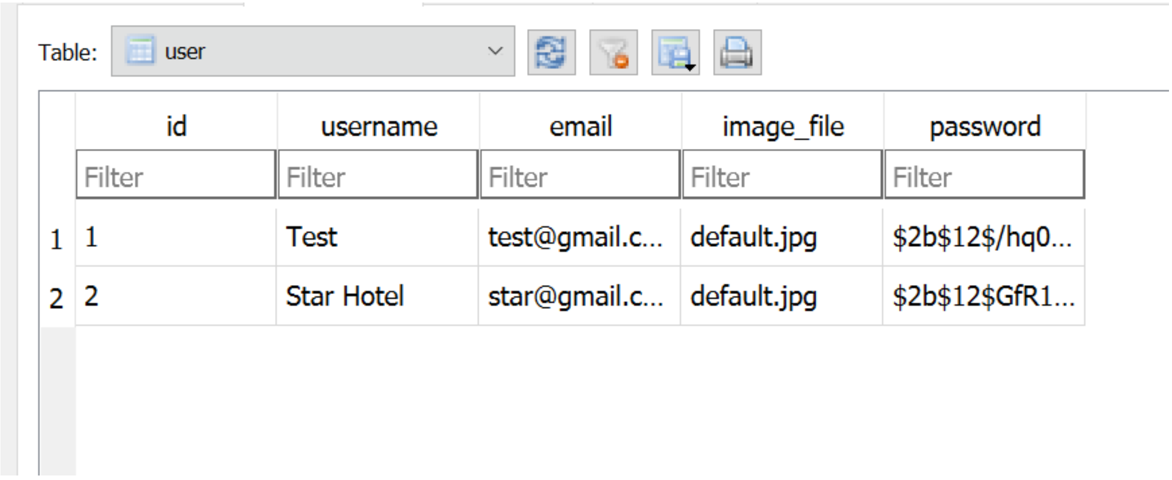


Table: user

	id	username	email	image_file	password
	Filter	Filter	Filter	Filter	Filter
1	1	Test	test@gmail.c...	default.jpg	$2b$12$/hq0...
2	2	Star Hotel	star@gmail.c...	default.jpg	$2b$12$GfR1...

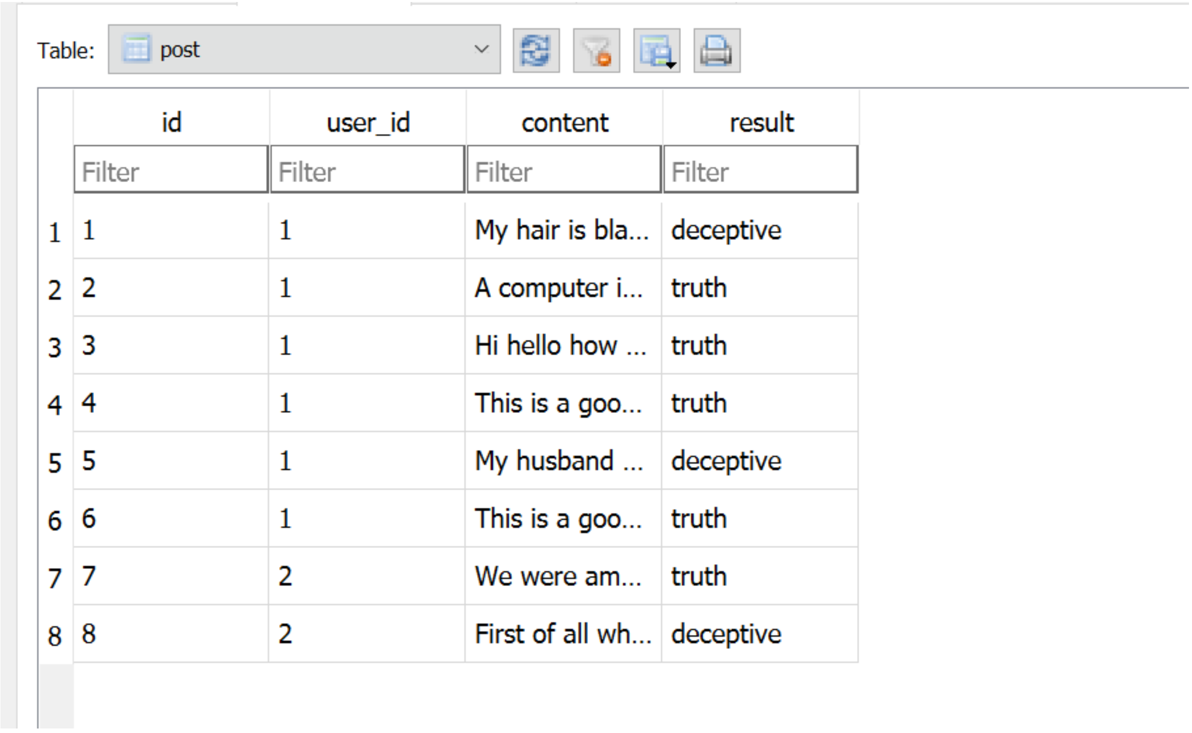


Table: post

	id	user_id	content	result
	Filter	Filter	Filter	Filter
1	1	1	My hair is bla...	deceptive
2	2	1	A computer i...	truth
3	3	1	Hi hello how ...	truth
4	4	1	This is a goo...	truth
5	5	1	My husband ...	deceptive
6	6	1	This is a goo...	truth
7	7	2	We were am...	truth
8	8	2	First of all wh...	deceptive

4. SYSTEM IMPLEMENTATION

4.1 IMPLEMENTATION PROCEDURE

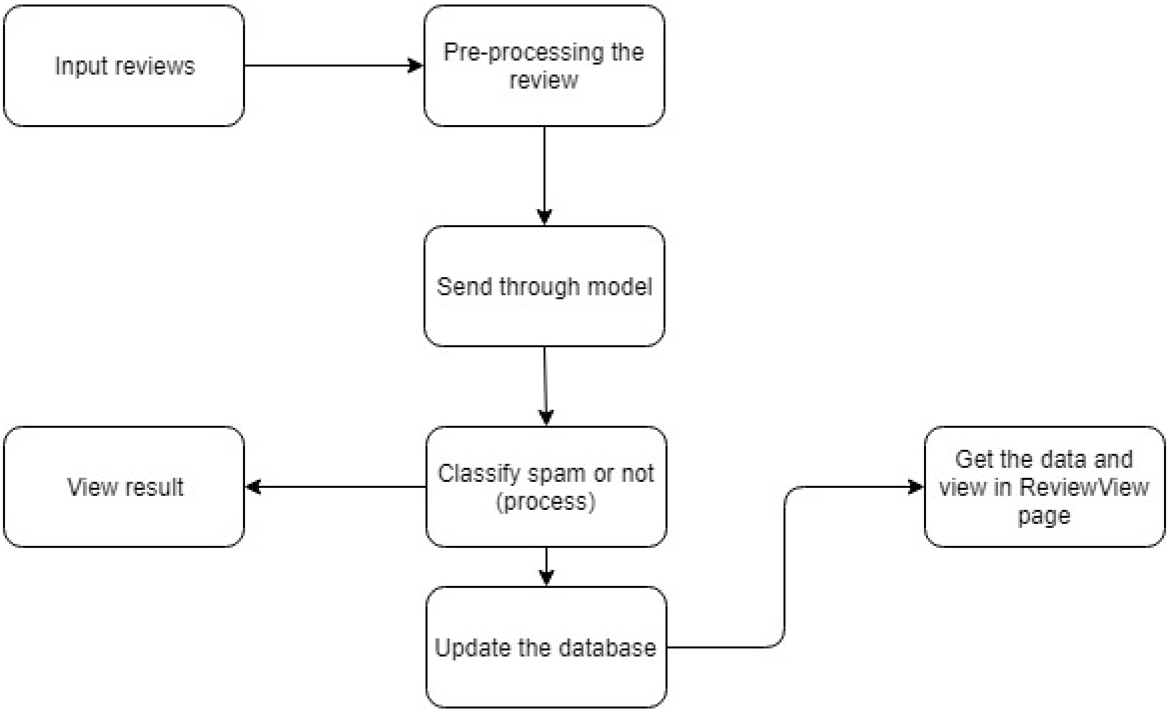
Technologies	*Reason for usage*
Python and Flask framework	*This the main part of the system which includes the model and its components.*
HTML, CSS	*This allows to create the GUI for the system*
Bootstrap	*This make the system attractive and responsive*
PyCharm	*Development IDE*

Classify model is created by using multinomial classifier. If using this classifier then accuracy is more than 80%. So, our company used this classifier to classify spam results. SQLite database is used to save user details and reviews.

4.2 MATERIALS

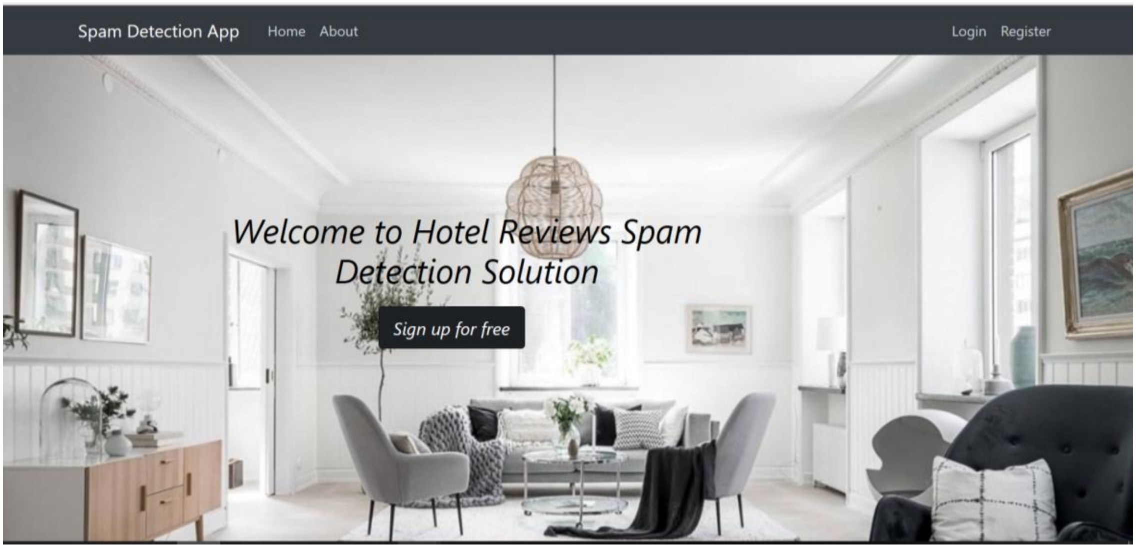
1. The machine learning model used in this system to classify spam results. It is created by using multinomial classifier.
2. Reviews dataset collected from hotels websites(yelp.com) using web scraping.

4.3 THE ALGORITHM


Input reviews
Pre-processing the review
Send through model
View result
Classify spam or not (process)
Get the data and view in ReviewView page
Update the database

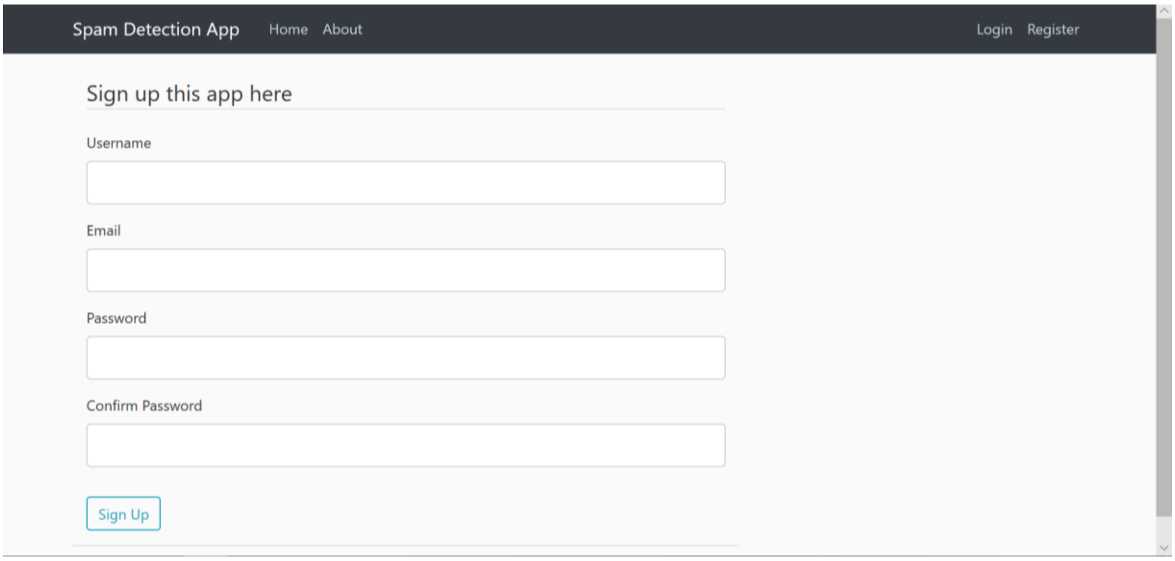
4.4 MAIN INTERFACES

1. Welcome page



Welcome to Hotel Reviews Spam Detection Solution

2. Register page
   User is registered by username, email, password fields. Every field is fill not optional.



Sign up this app here

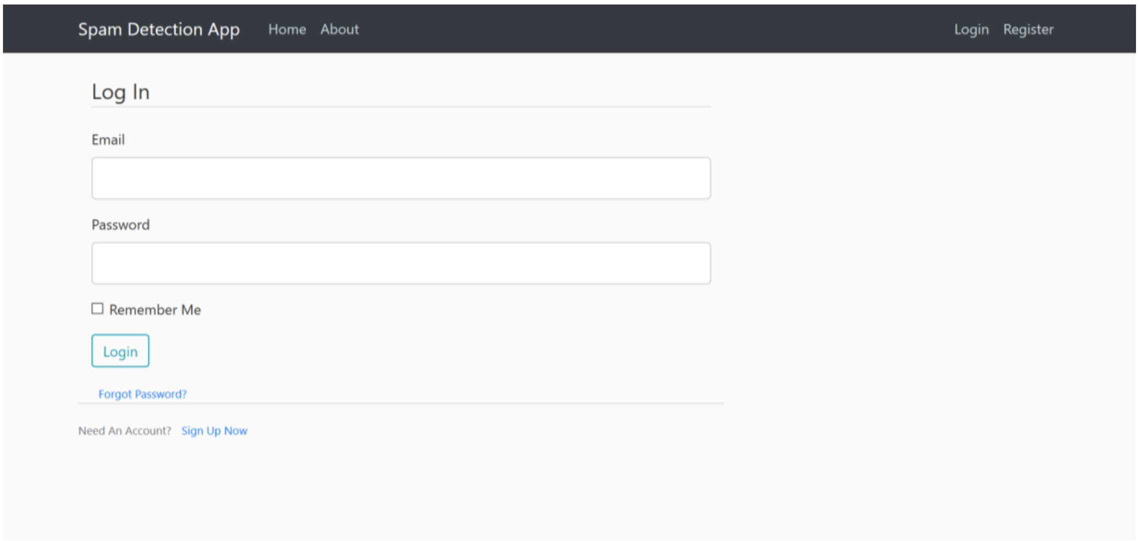
Username

Email

Password

Confirm Password

3. Login page
   Login Page for the User login, It will require the users to enter the Email and Password to authenticate. If someone haven't user account in Spam detection app, then they can easily create the user accounts using the Sign up now link. When user is trying to login the system with wrong login then login unsuccessful message will be flashed.



Spam Detection App Home About Login Register

Log In

Email

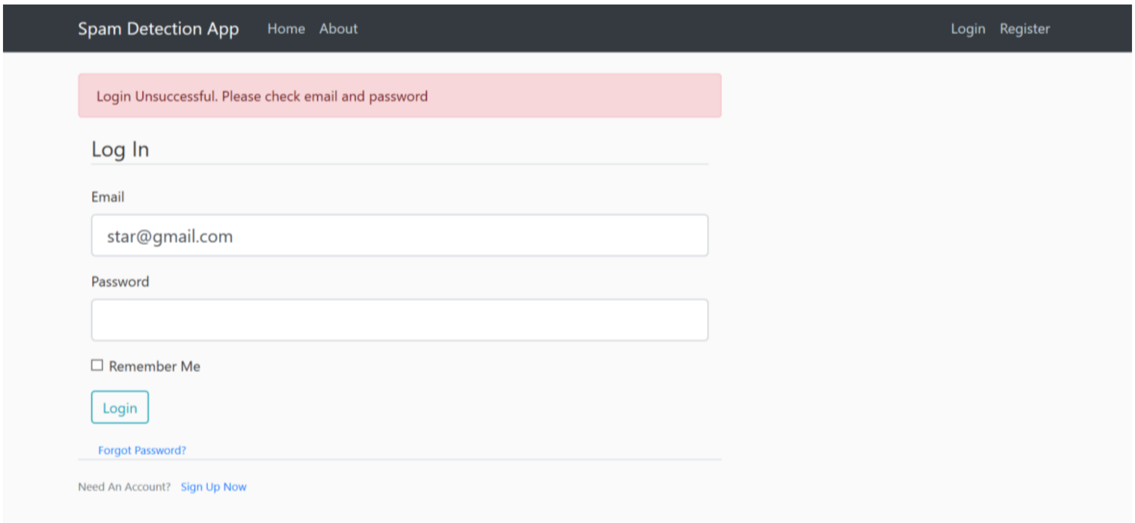
Password

□ Remember Me

Login

Forgot Password?

Need An Account? Sign Up Now



Spam Detection App Home About Login Register

Login Unsuccessful. Please check email and password

Log In

Email

star@gmail.com

Password

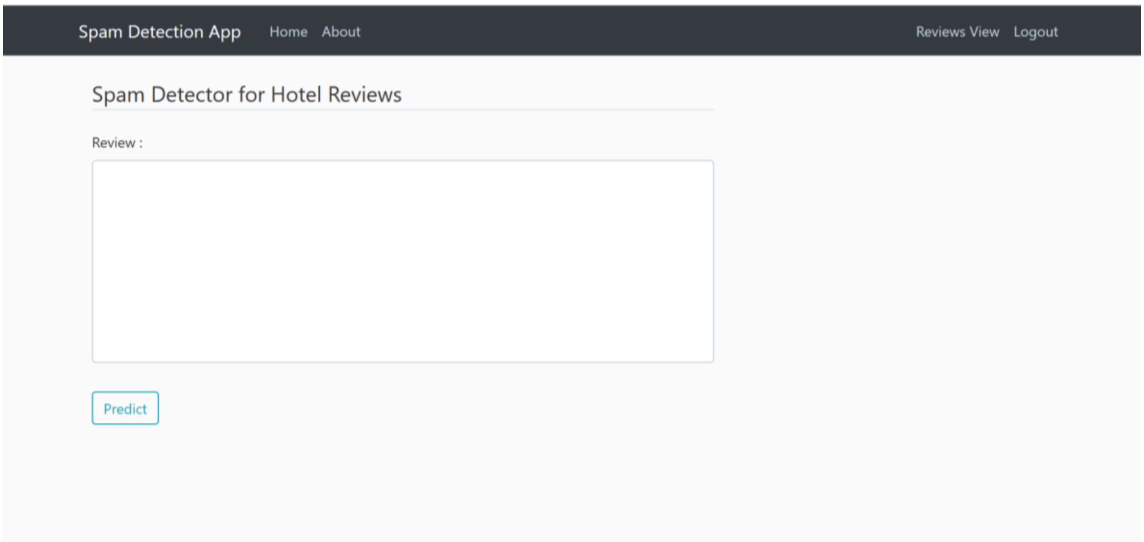
□ Remember Me

Login

Forgot Password?

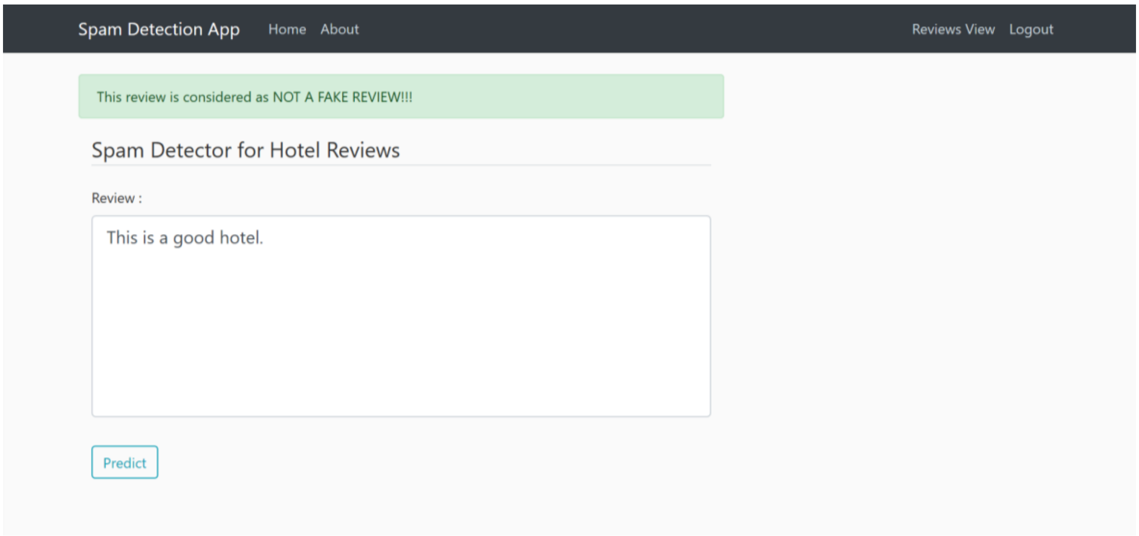
Need An Account? Sign Up Now

4. Home page
   This is main part of the system. Here user input their reviews and get their results in the same page.



Spam Detector for Hotel Reviews

Review :



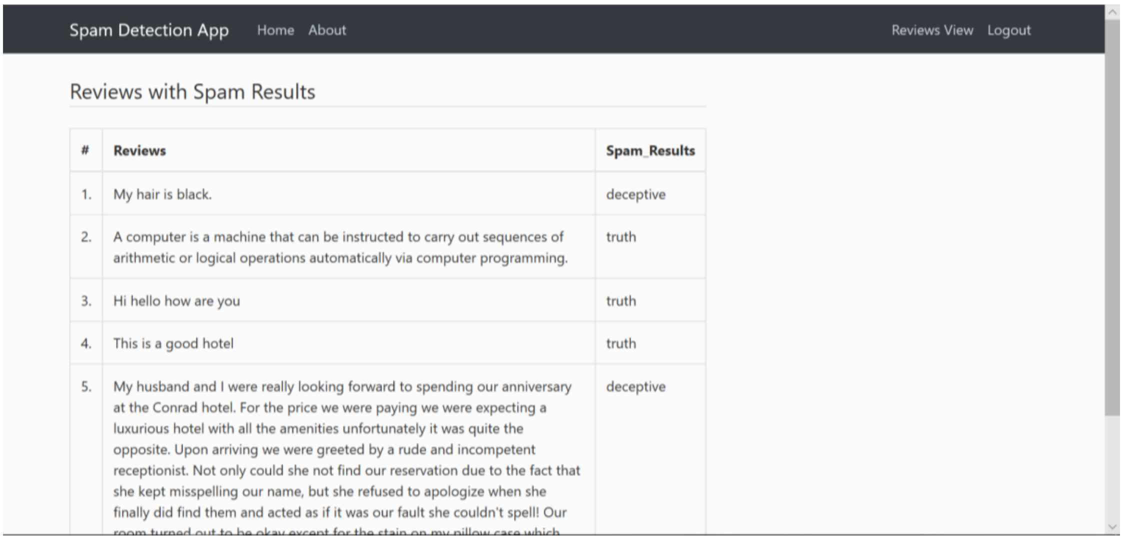
This review is considered as NOT A FAKE REVIEW!!!

Spam Detector for Hotel Reviews

Review :

This is a good hotel.

5. Review View page
   User can view their all reviews with spam results in one page. This very useful for users.



Reviews with Spam Results

#	Reviews	Spam_Results
1.	My hair is black.	deceptive
2.	A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming.	truth
3.	Hi hello how are you	truth
4.	This is a good hotel	truth
5.	My husband and I were really looking forward to spending our anniversary at the Conrad hotel. For the price we were paying we were expecting a luxurious hotel with all the amenities unfortunately it was quite the opposite. Upon arriving we were greeted by a rude and incompetent receptionist. Not only could she not find our reservation due to the fact that she kept misspelling our name, but she refused to apologize when she finally did find them and acted as if it was our fault she couldn't spell! Our room turned out to be okay except for the stain on my pillow case which	deceptive

5. SYSTEM TESTING AND ANALYSIS

5.1 TESTING APPROACH

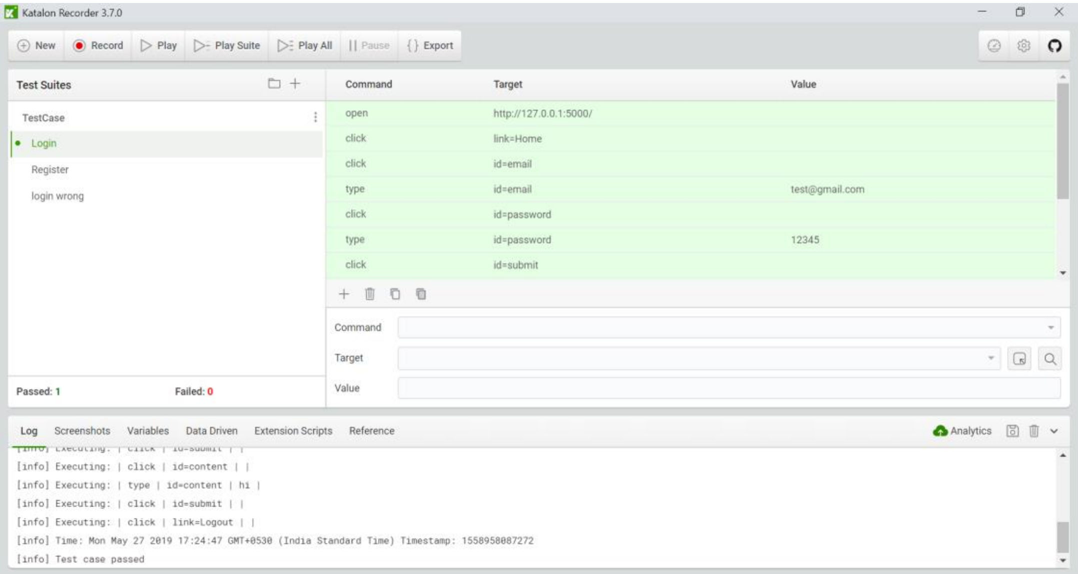
This project is about spam detection of hotel reviews. User of the system is hotels. Every user hotel signup the spam detection web app then only they enter the spam detector web page. That page user input the reviews and see their spam results in that page. Then they see all reviews with spam results in table view. After user use the app then logout their web app. This is a implementation view of spam detection app.

A test plan has been created to make sure that the system conforms to the specifications, design and to perform quality assurance on the final product. It will help to remove bugs in the system as much as possible and reduce the risk of software failure. The test plan will help to verify whether the final product delivers the expected outcomes and fulfils the requirements successfully with a variety of testing techniques. Moreover, it will also help to detect faults by using the designed test cases.

The main tests included are

- Unit test and Integration testing using Python unit tests
- User-interface testing is done through Selenium, Katalon recorder

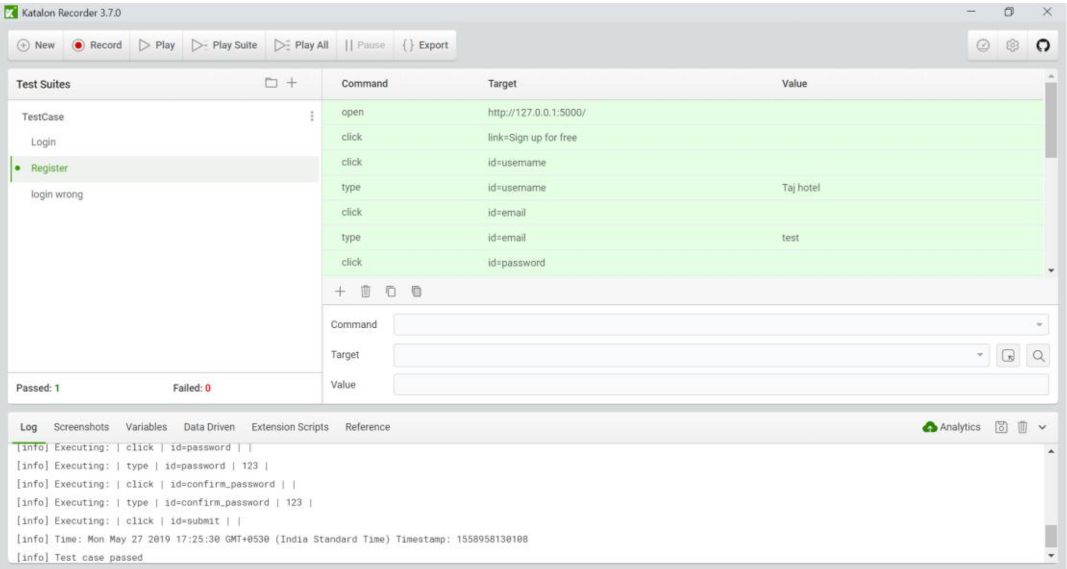
5.2 Unit Testing, Results and analysis of testing



Command	Target	Value
open	http://127.0.0.1:5000/	
click	link=Home	
click	id=email	
type	id=email	test@gmail.com
click	id=password	
type	id=password	12345
click	id=submit	

Passed: 1 Failed: 0

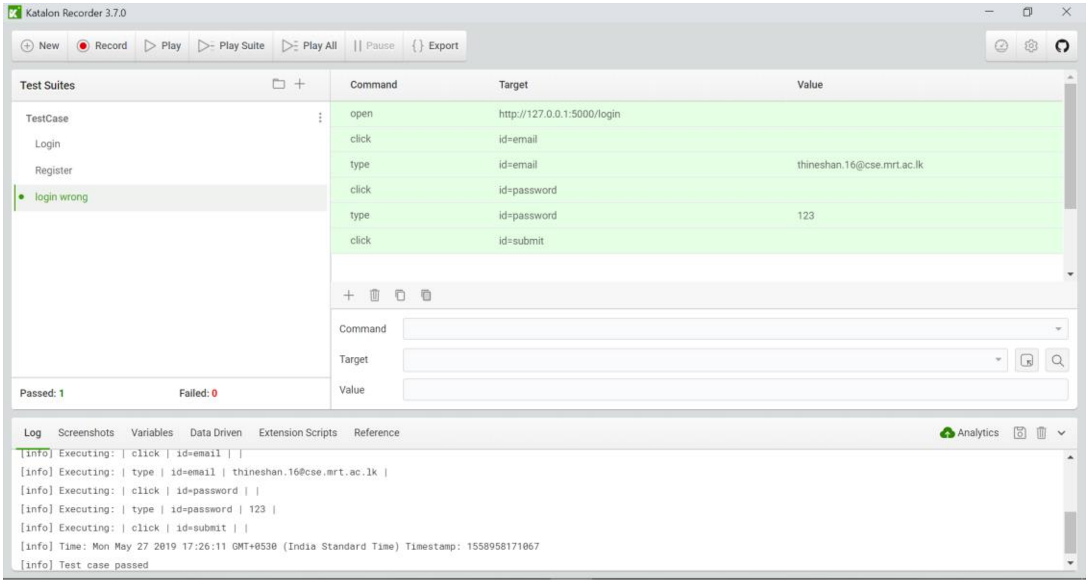
```
[info] Executing: | click | id=submit | |
[info] Executing: | click | id=content | |
[info] Executing: | type | id=content | hi |
[info] Executing: | click | id=submit | |
[info] Executing: | click | link=Logout | |
[info] Time: Mon May 27 2019 17:24:47 GMT+0530 (India Standard Time) Timestamp: 1558958087272
[info] Test case passed
```



Command	Target	Value
open	http://127.0.0.1:5000/	
click	link=Sign up for free	
click	id=username	
type	id=username	Taj hotel
click	id=email	
type	id=email	test
click	id=password	

Passed: 1 Failed: 0

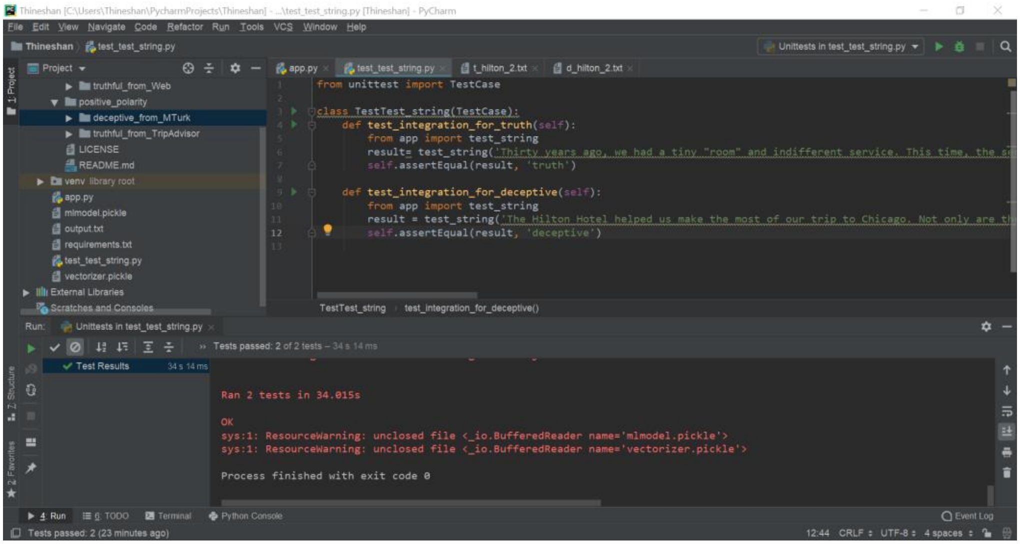
```
[info] Executing: | click | id=password | |
[info] Executing: | type | id=password | 123 |
[info] Executing: | click | id=confirm_password | |
[info] Executing: | type | id=confirm_password | 123 |
[info] Executing: | click | id=submit | |
[info] Time: Mon May 27 2019 17:25:30 GMT+0530 (India Standard Time) Timestamp: 1558958130108
[info] Test case passed
```



Command	Target	Value
open	http://127.0.0.1:5000/login	
click	id=email	
type	id=email	thineshan.16@cse.mrt.ac.lk
click	id=password	
type	id=password	123
click	id=submit	

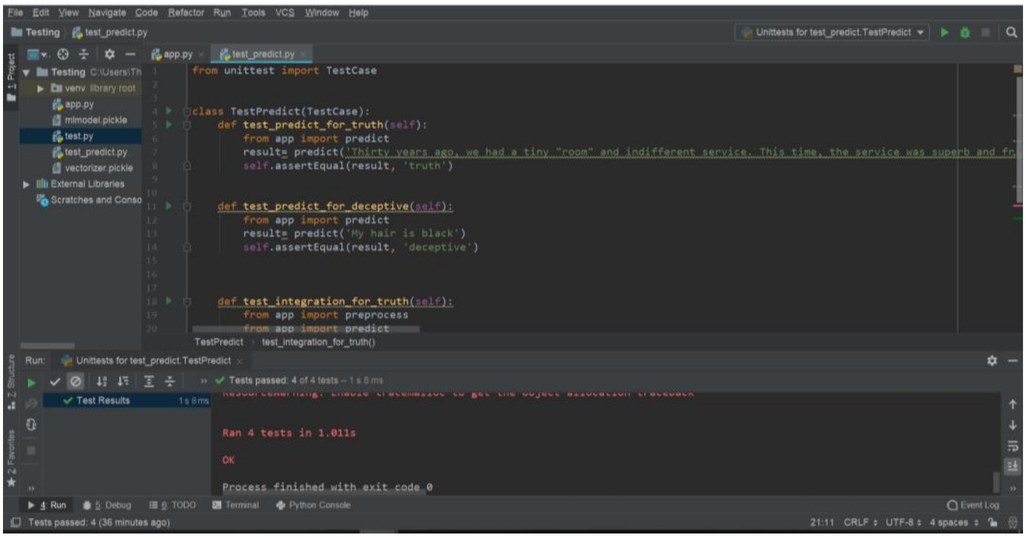
Passed: 1 Failed: 0

```
[info] Executing: | click | id=email | |
[info] Executing: | type | id=email | thineshan.16@cse.mrt.ac.lk |
[info] Executing: | click | id=password | |
[info] Executing: | type | id=password | 123 |
[info] Executing: | click | id=submit | |
[info] Time: Mon May 27 2019 17:26:11 GMT+0530 (India Standard Time) Timestamp: 1558958171067
[info] Test case passed
```



```
from unittest import TestCase

class TestTest_string(TestCase):
    def test_integration_for_truth(self):
        from app import test_string
        result= test_string('Thirty years ago, we had a tiny "room" and indifferent service. This time, the s
        self.assertEqual(result, 'truth')

    def test_integration_for_deceptive(self):
        from app import test_string
        result = test_string('The Hilton Hotel helped us make the most of our trip to Chicago. Not only are th
        self.assertEqual(result, 'deceptive')
```

```
Ran 2 tests in 34.015s

OK
sys:1: ResourceWarning: unclosed file <_io.BufferedReader name='mlmodel.pickle'>
sys:1: ResourceWarning: unclosed file <_io.BufferedReader name='vectorizer.pickle'>

Process finished with exit code 0
```



```
from unittest import TestCase


class TestPredict(TestCase):
    def test_predict_for_truth(self):
        from app import predict
        result= predict('Thirty years ago, we had a tiny "room" and indifferent service. This time, the service was superb and fr
        self.assertEqual(result, 'truth')


    def test_predict_for_deceptive(self):
        from app import predict
        result= predict('My hair is black')
        self.assertEqual(result, 'deceptive')



    def test_integration_for_truth(self):
        from app import preprocess
        from app import predict
```

Tests passed: 4 of 4 tests – 1 s 8 ms

```
ResourceWarning: Enable tracemalloc to get the object allocation traceback


Ran 4 tests in 1.011s

OK

Process finished with exit code 0
```

Tests passed: 4 (36 minutes ago)

6. CONCLUSION AND FUTURE WORK

Spam detection of hotel reviews system is a very crucial factor for business. Now a day's fake reviews are a biggest problem to hotel management systems. Many popular companies want spam detection tools. This will be very helpful for anyone who like to detect fake reviews about their hotels.

Future work

- Improve the user interfaces
- Include some add in our pages
- User create their own model
- Improve the speed of detect spam results
- Improve the performance of the system and accuracy
- Model is learning user input reviews using deep learning procedure.

REFERENCE

- Reviewmeta.com. (2019). *ReviewMeta.com - Amazon Review Checker*. [online] Available at: https://reviewmeta.com/ [Accessed 22 Jan. 2019].
- Ramirez, V. (2019). *3 Free Fake Reviews Detectors to help you Spot Fake Reviews on Amazon - AMZDiscover Blog*. [online] AMZDiscover Blog. Available at: https://www.amzdiscover.com/blog/3-free-fake-reviews-detectors-to-help-you-spot-fake-reviews-on-amazon/#C1 [Accessed 22 Jan. 2019].
- GitHub. (2019). *Build software better, together*. [online] Available at: https://github.com/topics/spam-detection?fbclid=IwAR03HRVyKhfgTq9yjYAJ3yBsiIBb3JnGDmmxR3_n6H_D25E5lfl2gSwShS0 [Accessed 22 Jan. 2019].
- Towards Data Science. (2019). *Spam Detection with Logistic Regression – Towards Data Science*. [online] Available at: https://towardsdatascience.com/spam-detection-with-logistic-regression-

23e3709e522?fbclid=IwAR0fBE09zxegzzpnaTSjp4zhmMFDouGhaJBl6xHl7Y60XOHWFeHuffmHaP4 [Accessed 21 Jan. 2019].

- Selenium available at https://www.seleniumhq.org/
- Slideshare.net. (2019). *4+1 View Model of Software Architecture*. [online] Available at: https://www.slideshare.net/bashcode/41-view-model-of-software-architecture [Accessed 22 Feb. 2019].
- Slideshare.net. (2019). *Ch6-Software Engineering 9*. [online] Available at: https://www.slideshare.net/SE9/ch6-5147962 [Accessed 22 Feb. 2019].