Software Requirement Specification

Submitted as part of the requirements for:

CE903 Group Project

Topic: Fact-checking (based on Wikipedia)

Group Number: **6**

Registration Number of Group Members:

1802772
1806447
1804523
1806152
1807600
1807846

Date of Submission: 7 February, 2019

Version: 1 Date: 7 February, 2019

Table of Contents

List of Figures

List of Tables

|

1 Introduction

The news is an integral part of our daily life. It keeps us up to date with the current events that are happening around us and throughout the world. For instance, by considering the news about weather forecasting, we often decide whether it is necessary to carry an umbrella or not. In olden days we used to get to know about news through newspapers. However, at present with the help of the internet we are getting the news immediately. Moreover, through various social media platforms like Facebook, Twitter, and many more we can share the news with others without knowing the authenticity.

News which is not true is called fake news. With the continuous growth of digital data, anyone can produce news and can spread it across social media. Often we are not familiar with the source of news. Current social media exhibit an increasing amount of misleading information and fake news. Therefore, combatting against fake news is an important issue for our society [1], [2]. The source of the fake news varies and depends on the different person, ethnicity, nation and other factors. We have seen many politics affected by the fake news. Consequently, there should be a proper method to detect various fake news and misleading information.

One possible way to combat fake news is the fact-checking process [2], [3], and [4]. Fact checking provides the evidence to rebut the misleading news which is often published in different Media. With the increasing amount of false news, the reliance on fact-checking systems have been increased and it is one of the means to determine the correctness of factual statements in social media. Therefore, our proposed system is based on the fact-checking process to identify the correctness of a statement provided or asked by a user.

In order to implement the proposed system, Wikipedia has been selected as the basis of the fact-checking process. Moreover, a flexible agile model have been chosen as project management tools. Next two subsections describe the purpose and scope of the proposed fact-checking process.

1.1 Purpose

The main purpose of the proposed system is to provide a comprehensive solution to perform the fact-checking against online Wikipedia pages for a given user query. The query is mainly based on a domain which is restricted to a person name. The project mainly deals with the fact-checking process by using the natural language processing methods and information retrieval models to detect the factual claim by using the Wikipedia. Moreover, the proposed system meant to be a useful tool with further development scope to identify the facts by taking account of a different layer of natural language processing, text analytics, information retrieval, and machine learning algorithms to produce a valuable database which can be used for other NLP analysis related tasks.

1.2 Scope

The proposed fact-checking system will produce an automatic fact-checking method that is capable to identify the fact across several Wikipedia pages to determine the authenticity of a statement provided by the user. The solution will also able to preserved processed Wikipedia pages for future reference. Furthermore, preserved Wikipedia data will enable the system not to do the same pre-processing task for similar page retrieval. Machine learning algorithm will be a useful medium to broaden the scope of the proposed system by identifying the pattern in preserved data. As a part of the future scope, the system can be expanded across all domain available at Wikipedia and across different news medium which ultimately leads towards the fact-checking of false news. It could be done for online and offline versions.

1.3 Abbreviations

NER:	Name Entity Recognition
NLP:	Natural Language Processing
NLTK:	Natural Language Toolkit
RE:	Regular Expression

2 System Requirement Specification

In general fact-checking process mainly provides a verdict on whether a specific claim is a true, false, or mixed reaction based on a fact-checking methodology [2]. Several algorithms have been undertaken by different organizations to design the fact-checking process based on different approaches and domains define by a fact-checking community [2], [4], and [5]. This proposed project mainly deals with fact-checking for a user query by

using the online Wikipedia [6], [7], [8], and [9]. While user queries are mainly related to a person, Wikipedia is the fact-checking basis to retrieve relevant documents for further processing and to check the validity of user query.
For instance, the user query is related to a person called Barak Obama. Then the user will devise a query/ claim about Barak Obama. The system will take the query and will check the fact based on the user query and available fact retrieved from Wikipedia. Based on the available fact in relevant Wikipedia pages, the system will revert back positive or negative feedback to the user for that particular claim. Below are some examples of the possible query by the user and the fact-checking method of the proposed system.

Example 1: Incorrect Claim.
The claim by user: "Barack Obama born in UK."

The necessary fact that will be checked is, whether Barak Obama born in the UK or not. Identification of fact will be based on available information of relevant Wikipedia pages.

- Step 1: Find name from the query by pre-processing (NER) pipeline and redirect to appropriate URL; for this example relevant page is Barak Obama.
- Step 2: Extract data from url and search the url data with a verb in the query using a regular expression and fetch the sentence.
- Step 3: Convert fetched sentence into NER.
- Step 4: Compare both query and fetched sentence (NER equivalent) and if there are some differences between comparison and the fact is wrong.
- Step 5: Display the message saying that the query is wrong. Display fetched sentence from Wikipedia.

Example 2: Correct Claim.
The claim by user: "Donald Trump is the president of U.S.A"

The necessary fact that will be checked is, whether Donald Trump is the president of USA or not. Identification of fact will be based on available information of relevant Wikipedia pages.

- Step 1: Find the name from the query by pre-processing (NER) pipeline and redirect to appropriate URL, in this case, the name is Donald Trump.
- Step 2: Extract data from url and search the url data with a verb in the query using a regular expression and fetch the sentence.
- Step 3: Convert fetched sentence into NER.
- Step 4: Compare both query and fetched sentence (NER equivalent) and if they are same then the fact is correct.
- Step 5: Display the message saying that the query is correct. Display fetched sentence from Wikipedia.

Moreover, this system requirement specification intensified and focuses on different steps of proposed software development to understand the nature of the programs to be built, and other systems it interfaces with. The system requirement specification has been articulated in four different steps called feasibility study, methods for requirements elicitation and analysis, system models, and requirements specification.

2.1 Feasibility Study

In order to evaluate the project’s potential and success, four types of the feasibilities study have been conducted. Next four subsections will illustrate them briefly.

2.1.1 Economic Feasibility

Wikipedia is a free, enormous, cross-language NLP repository with more than 36 million articles. It continues to grow while having more than 10,000 users across the world for maintenance and information updates. In this project, we will use Wikipedia as our database, through the natural language pre-processing of text data, and then pass the processed data as a search key to Wikipedia, get structured data from Wikipedia [7], [8], and [9]. After that, we will analyse the language and finally report the results to the user.

2.1.2 Technical feasibility

Selenium is a powerful web application testing tool that can simulate interactions between the clients and server, it is also implemented in popular programming languages including C#, JAVA, PYTHON, RUBY and more. Nltk and WordNet provide tools for natural language processing and text analysis. Beautiful Soup can easily crawl webpage's feedback data. Pandoc is capable of converting and manipulating various types of structured data. Mysql and Mybatis are able to create relational mapping databases (ORMs), and database operations will be more convenient.

2.1.3 Theoretical feasibility

N-grams, k-dependent Markov chain, CFG can be used as a theoretical tool for text and machine language processing. Related information retrieval theory, text symbol substitution, Bag of Words, and free-text indexing.

2.1.4 Operational feasibility

Our project will continue to be carried out, and an iterative mode is used for continuous optimization and update. We expect that we will open the test version first. This version has no user interaction interface. When all the integration tests are finished, a new version is iterated to the software with a visual interface. Our ultimate goal is to build a website that can be accessed by users on their mobile phones, including the corresponding supporting facilities.

2.2 Requirements Elicitation and Analysis

With the expansion of social networks and the continued growth of intelligent network hardware access which enrich people's cultural and spiritual satisfaction, meanwhile, brings very favourable conditions for the rapid spread of fake news, these false news are often increased through blogs, social networks, online magazines, and others, causing numerous problems for citizens and may even endanger the safety and order of society. For example [10], after the explosion of the Fukushima nuclear power plant in Japan on April 12, 2011, rumours released a message on the Internet that residents living in the Liaodong Peninsula may be threatened by nuclear radiation. Therefore, a large number of residents flocked to major supermarkets to buy salt and daily necessaries. Until now, variations of various false news still challenge the real information, which is quite confusing to people's audio-visual. Moreover, according to [11] there are restrictions to websites and other tools that can automatically detect false news, such as the US website "The Onion" (Rubin et al., 2016), which was founded in 1996, and the virus tracking website "Buzzfeed" (Potthast). Et al., 2017) also has a fact-finding website “PolitiFact” (Wang, 2017). Although the above-listed websites can achieve basic statement verification, for satirical, humorous or absurd sentences, although they have used text analysis technology to conduct very useful explorations, they are still powerless. More importantly, most websites that trace on false news only support detection in specific areas, such as politics, entertainment, sports, and others. Above all, the database cannot support such large-scale data storage at this stage.

2.3 Different System Models and UML

In order to understand the functionality of the system and overall architecture of the proposed software, different system modelling has been incorporated. Moreover, different system models mainly represent the different perspective of the overall systems. Furthermore, illustrated system modelling is mainly based on the Unified Modelling Language (UML) [12]. Next two subsections describe the proposed software with appropriate system models.

2.3.1 Data Processing Model

The proposed fact-checking project towards an automated fact-checking method with the focus on the user query and the system which determines the factual user queries that are worth checking from online Wikipedia. The proposed system will be deployed and substantially tested in different use cases. The overall architecture of the system is as follows.

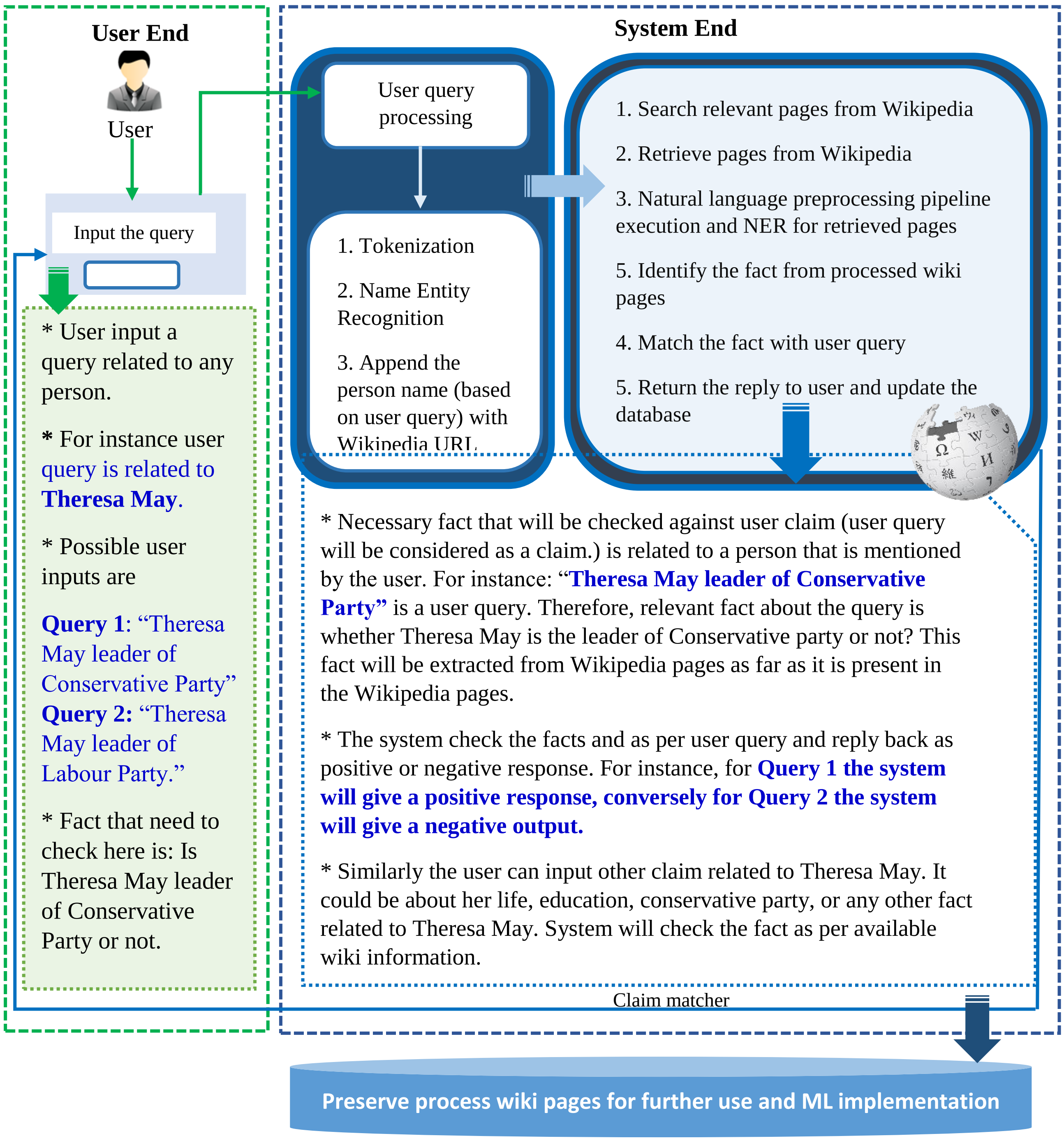

User End
User
Input the query
* User input a query related to any person.
* For instance user query is related to Theresa May.
* Possible user inputs are
Query 1: "Theresa May leader of Conservative Party"
Query 2: "Theresa May leader of Labour Party."
* Fact that need to check here is: Is Theresa May leader of Conservative Party or not.
System End
User query processing
1. Tokenization
2. Name Entity Recognition
3. Append the person name (based on user query) with Wikipedia URL
1. Search relevant pages from Wikipedia
2. Retrieve pages from Wikipedia
3. Natural language preprocessing pipeline execution and NER for retrieved pages
5. Identify the fact from processed wiki pages
4. Match the fact with user query
5. Return the reply to user and update the database
* Necessary fact that will be checked against user claim (user query will be considered as a claim.) is related to a person that is mentioned by the user. For instance: "Theresa May leader of Conservative Party" is a user query. Therefore, relevant fact about the query is whether Theresa May is the leader of Conservative party or not? This fact will be extracted from Wikipedia pages as far as it is present in the Wikipedia pages.
* The system check the facts and as per user query and reply back as positive or negative response. For instance, for Query 1 the system will give a positive response, conversely for Query 2 the system will give a negative output.
* Similarly the user can input other claim related to Theresa May. It could be about her life, education, conservative party, or any other fact related to Theresa May. System will check the fact as per available wiki information.
Claim matcher
Preserve process wiki pages for further use and ML implementation

Figure 1: Major Components of Fact Checking process

The overall data flow (up to level 1) of proposed fact-checking software can be illustrated as follows.

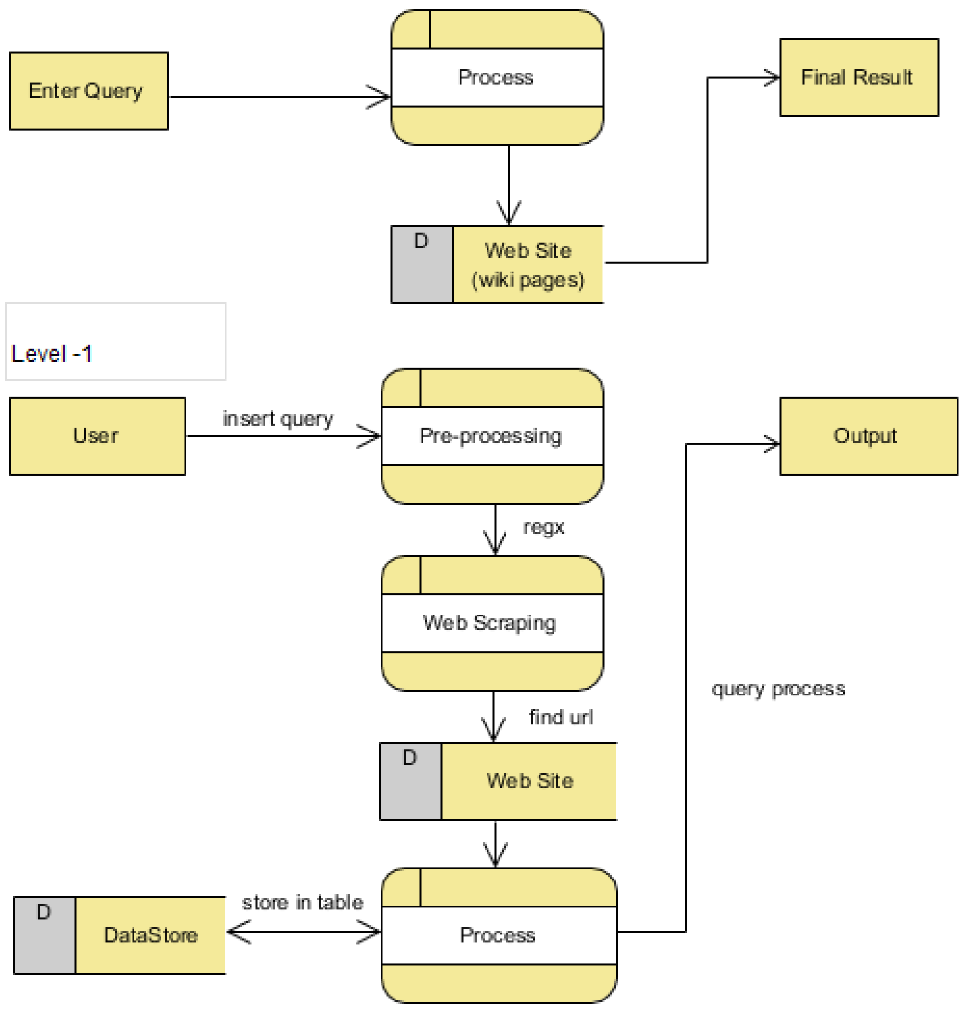


Figure 2: Data Flow Diagram for Fact Checking

2.3.2 Scenarios and Use Case Models

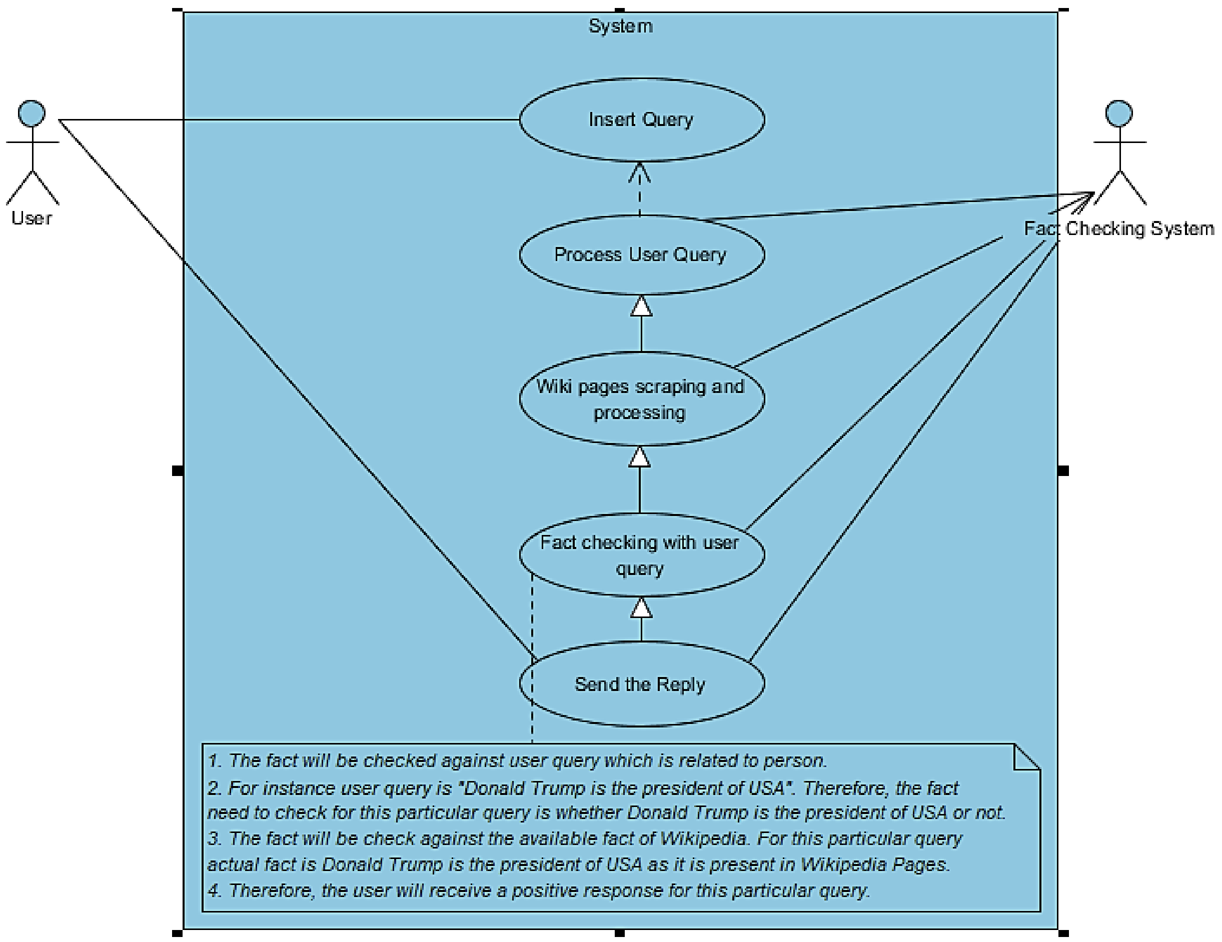
This section will describe different UML diagram for necessary logical clarification and a better understanding of the overall implementation of the proposed software system. Used UML for coherent representation is a Use Case Diagram, Activity Diagram, and Class Diagram.

Use Case Diagram:

Mainly, the working process of the proposed software system has been described in two expressions. The first expression is the External Working Procedure of the system. The second expression is the Interior Working Procedure Representation

External Working Procedure of the system:

The external working process of use case diagram describes the outer job of the scheme.



1. The fact will be checked against user query which is related to person.
2. For instance user query is "Donald Trump is the president of USA". Therefore, the fact need to check for this particular query is whether Donald Trump is the president of USA or not.
3. The fact will be check against the available fact of Wikipedia. For this particular query actual fact is Donald Trump is the president of USA as it is present in Wikipedia Pages.
4. Therefore, the user will receive a positive response for this particular query.

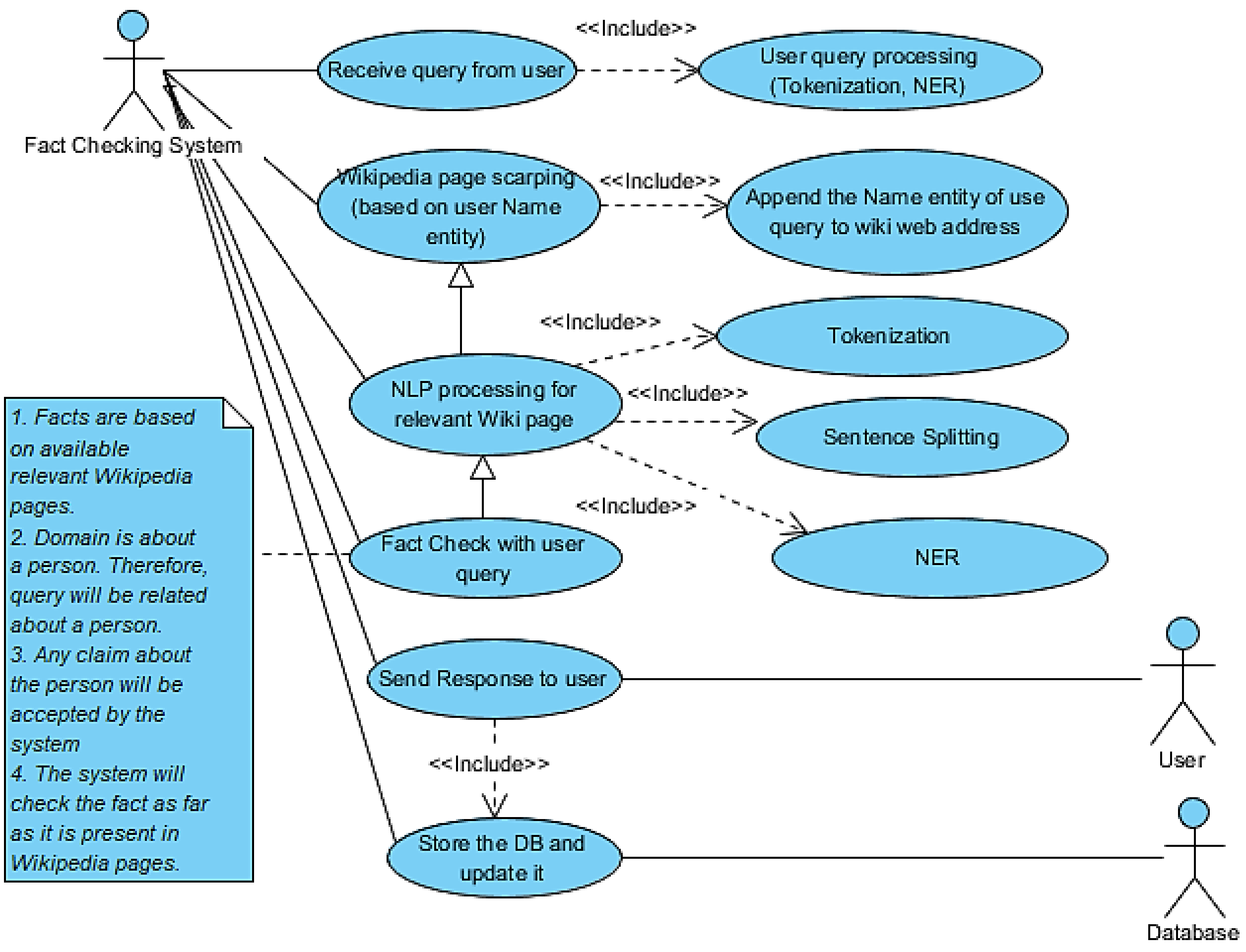
Figure 3: Outer View of Fact Checking System

Figure 3 is the outer view of the proposed software system. The user first inserts the query. Based on the user query the Fact-Checking System process the user query. After retrieving and processing the Wikipedia pages, the system checks the fact against user query and send back the response. Therefore, the whole fact-checking process come to an end with the response to the user.

Interior Working Procedure of the system:

The inner work representation describes the working procedure of the fact-checking system, which is unknown to the user. It describes the system design for the planned system in different dimensions. This dimensional description is done in three parts.

Moreover, this diverse explanation of the use case diagram helps to understand the system's structure.



Fact Checking System
Receive query from user
<<Include>>
User query processing (Tokenization, NER)
Wikipedia page scarping (based on user Name entity)
<<Include>>
Append the Name entity of use query to wiki web address
<<Include>>
Tokenization
NLP processing for relevant Wiki page
<<Include>>
Sentence Splitting
<<Include>>
NER
Fact Check with user query
1. Facts are based on available relevant Wikipedia pages.
2. Domain is about a person. Therefore, query will be related about a person.
3. Any claim about the person will be accepted by the system
4. The system will check the fact as far as it is present in Wikipedia pages.
Send Response to user
User
<<Include>>
Store the DB and update it
Database

Figure 4: Use Case Diagram for Proposed Fact Checking System

Above mentioned use case diagram describes how a Fact-Checking System process the user query. Moreover, it also describes how necessary Wikipedia pages can be retrieved and used for fact checking. Furthermore, it also illustrates that how database update related with fact checking system. After necessary cross-checking, the system sends back the response to the user.

Activity Diagram:

In order to describe the workflow of a Use Case Diagram, an Activity diagram has been deployed. The below-mentioned Activity Diagram can easily be described as the logical steps involved in the execution of Fact-Checking use case.

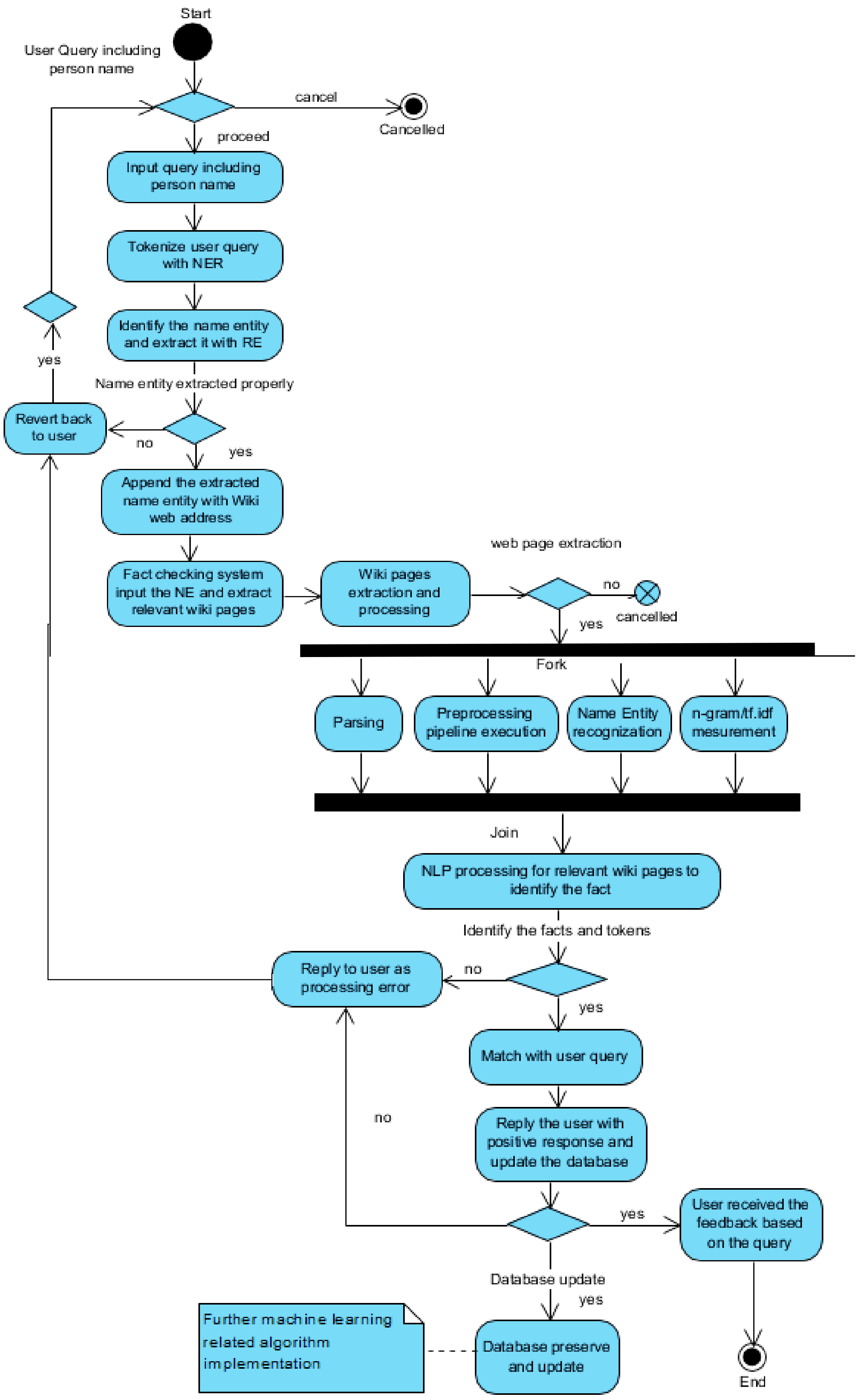

Start
User Query including person name
cancel
Cancelled
proceed
Input query including person name
Tokenize user query with NER
Identify the name entity and extract it with RE
Name entity extracted properly
yes
Revert back to user
no
yes
Append the extracted name entity with Wiki web address
Fact checking system input the NE and extract relevant wiki pages
Wiki pages extraction and processing
web page extraction
no
cancelled
yes
Fork
Parsing
Preprocessing pipeline execution
Name Entity recognization
n-gram/tf.idf mesurement
Join
NLP processing for relevant wiki pages to identify the fact
Identify the facts and tokens
Reply to user as processing error
no
yes
Match with user query
Reply the user with positive response and update the database
no
yes
User received the feedback based on the query
Database update
yes
Further machine learning related algorithm implementation
Database preserve and update
End

Figure 5: A complete Activity Diagram for Fact Checking Use Case

Class Diagram:

A Class Diagram is one of the prime sources of code generation [12]. Therefore, below mentioned class diagram illustrates different attributes with properties and association between classes to describe a static view of the elements that make up the proposed fact checking system.

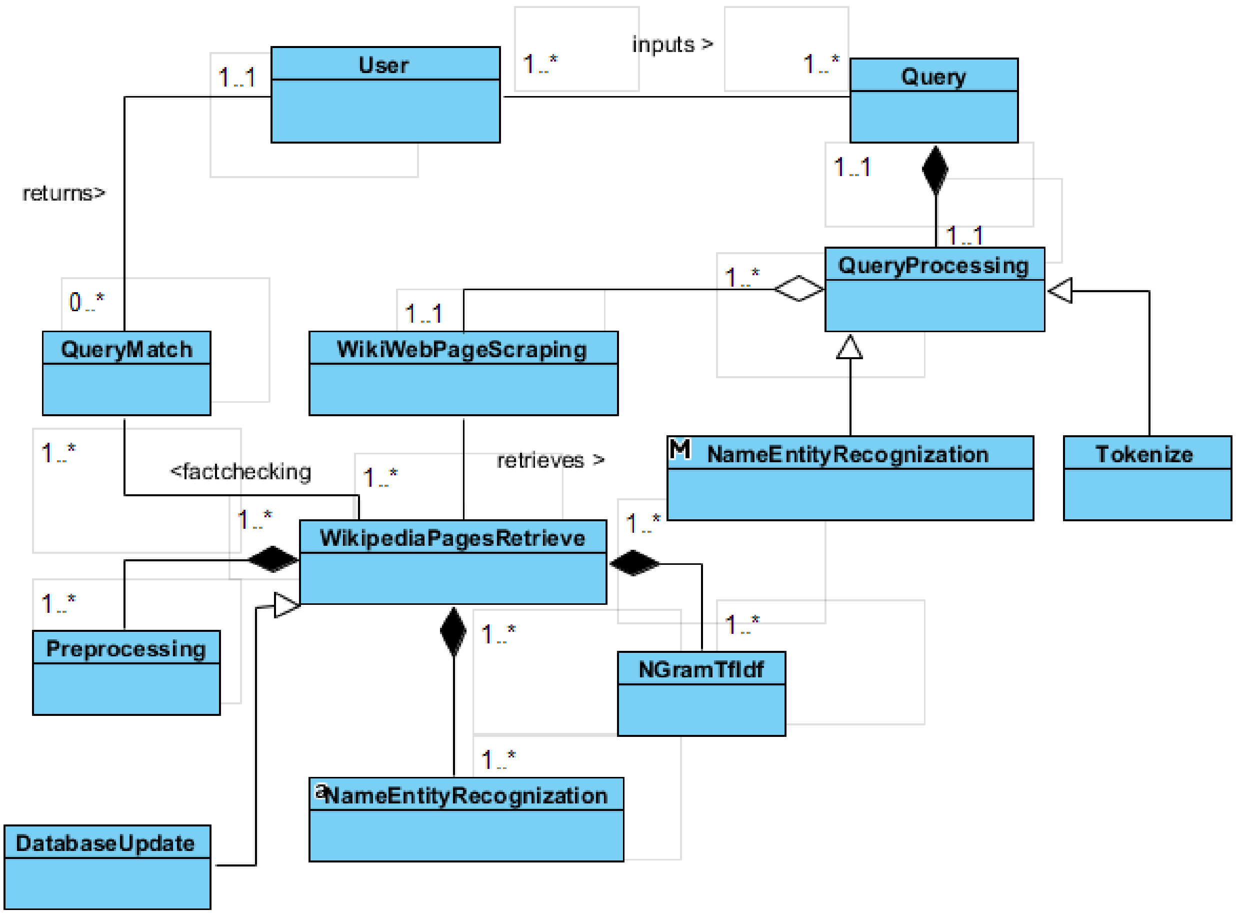


Figure 6: Class Diagram for proposed fact checking system

According to the above-mentioned class diagram, inputs, retrieves, and fact-checking describe the association between the user query and fact-checking system. While, webpage pre-processing pipeline, NER, n-gram, and tf-idf are the composition of retrieve/processing of Wikipedia page, web page scraping is the aggregation of user query processing. The relationship between attributes can be defined as 1 to many and others based on the proposed system implementation.

2.4 Requirements Specification

Requirements Specifications outlines the overall working process of the proposed fact-checking software. As this document is mainly to evaluate the requirements before the design phase, it is useful to define the main software architecture in order to avoid the design related complexity during final implementation. Next two subsections will briefly describe the required steps of requirements specification.

2.4.1 Functional Requirements

The proposed system is mainly dealing with fact-checking based on user input and Wikipedia's online pages. Therefore, a user inputs the query along with person name based on desire information needs. The proposed system process the query and takes it as input for the fact-checking process.

Functional requirements of the proposed system can be summarized as follows**.**

- Input: System accepts inputs from the user in the form of a sentence. It is necessary that user has some prior knowledge about how to input data in the system for finding the information.
- Pre-processing: After input doing pre-processing and web scraping convert input into a valid and redirect to the web.
- Fact checking: finally using some NER techniques identify the fact.
- Output: Display the fact to the user in which he/she is looking

Some services and functions that are relevant to fact-checking process are prime factors to the functional requirements. Necessary descriptions are below.

Function Number	Name of Software	Version Number	Description
1	Python 3+	3+(Windows, Mac, Linux)	The object-oriented python programming language is widely used for Natural Language Processing and Information Retrieval field. Python is a simple yet effective programming language that has a enrich collection of the library. Moreover, it is easily compatible with other programming languages. This proposed system will be developed by using the Python 3+.
2	NLTK	3.4+	Natural Language Toolkit is a widely used platform for python programs. A wide range of available packages makes the toolkit very effective in terms of pipeline processing, web scraping, parsing, and other NLP related works. NLTK will be used for web scraping, pre-processing pipeline execution and other relevant works to extract relevant information from online Wikipedia pages.

3	Scikit-learn	0.18.2+	Scikit-learn is an effective python library for data mining and machine learning related implementation. Scikit-learn will be used to implement some machine learning algorithms to find a more effective way of fact checking and database preservation.
4	Stanford/NLTK Parser	3.4+	Stanford or NLTK parser will be used to parse the retrieved data or to find out the Name Entity Relationship for processed tokens.
5	spaCy/TensorFlow	N/A	spaCy/TensorFlow could be used as an alternative to Scikit-learn.
6	Appropriate front end application for user input	N/A	It will be determined during the software development phase.

2.4.2 Non-Functional Requirements

Any project's requirements need to be properly articulated and clearly understood by both the client and developers to make it a successful execution. Therefore, both functional and non-functional requirements must have an appropriate manner with necessary detail and comprehensive collections of relevant requirements. Proper identification of non-functional requirements is as important as the functional requirements. Non-functional requirements can be defined as the global constraints of a software system. However, defining the non-functional requirements is not very easy as it is difficult to impose during the development of the software. Moreover, it is difficult to evaluate before the implementation phase as it can be changed with the development and time constraints.

In order to define the non-functional requirements, either product-oriented approaches or process-oriented can be implemented. While a product-oriented approach gives more focus on the quality of the system, process-oriented approaches give more empathises on the design process. Another way to identify the non-functional requirements is by quantitative or qualitative approaches. The quantitative approaches deal with measurable scale for quality attributes. Conversely, qualitative approaches identify the relationship of quality goals.

The non-functional requirements of the proposed fact checking system are mainly based on process-oriented approaches. The non-functional approaches can be defined as follows.

- **Interface requirements**

  Interface requirements can be measured in terms of user-friendliness and the interfaces with other systems. The proposed fact-checking system is based on python programing language, which is object-oriented language and quite easy to implement. Similarly, it can be easily interfaced with other systems.

- **Performance requirements**

  The performance requirement is mainly based on below parameters.

  i. **Time/Space bounds**

     Overall response time, available space for storage, and workload are some prime factors for time and space bound. Initially, the implementation of the proposed system required less space and time as the fact-checking database will be relatively small. However, the increasing amount of stored data might increase the response time for the fact-checking process. Similarly, with time the database will require more space to preserve the future data.

  ii. **Reliability**

     Reliability can be defined by components availability and integrity of information. The proposed system needs to be more reliable with a minimum amount of system downtime. It can be only defined after the implementation of the system.

  iii. **Security**

     In order to ensure the security of the dataflow, some cryptographic algorithm can be deployed to make sure the permissible information flow.

- **Operating requirements**

  The proposed system mainly based on simple yet effective implementation method. Moreover, it can be accessed without much difficulty and standard level of skills.

- **Lifecycle requirements**

  The proposed system is easily maintainable and portable. As it is a web-based application, it can be accessed from different places. Furthermore, it also has a great potentiality for further development.

- **Economic requirements**
  As a part of the initial implementation, the proposed system will cost a minimum level of cost as most of the development tools are open source. However, the exact cost can be measured after full implementation and by taking the account of the increasing amount of data.

3 Requirements Validation

Testing is a process of requirements validation to find the error in software and creates our software error free. To make our software perform well it should be error free. For removing errors from software testing is compulsory. If testing is done successfully the software is error free.

As per IEEE standard [13], 1059-1993 Testing is defined as

"*The process of analysing a software item to detect the differences between existing and required conditions (that is bugs) and to evaluate the features of the Software Item*."

There are many principles of testing. Some of them are as follows.

- All the test should meet the customer requirements. Because any unattended error causes the failure of the system.
- To make our software error free, testing should be performed by the third party. Because software engineer who creates the software is not capable to detect an error.
- Always start testing with small parts and as the test progress the focus of testing shift to find the final error to integrate the final component.
- Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- All the test to be conducted and should be planned before the implementation.
- It follows pareto rule (80/20 rule) which states that 80 percent of all errors come from 20 percent of program components.

Here in our model (Software), we use the Verification and Validation model. Moreover, these two terms are very confusing for most of the peoples and there are many peoples who use them interchangeably but these two are very much different.

Before proceeding let's have a look at the important difference between them.

Verification – Are we building the product right?

Validation – Are we building the right product?

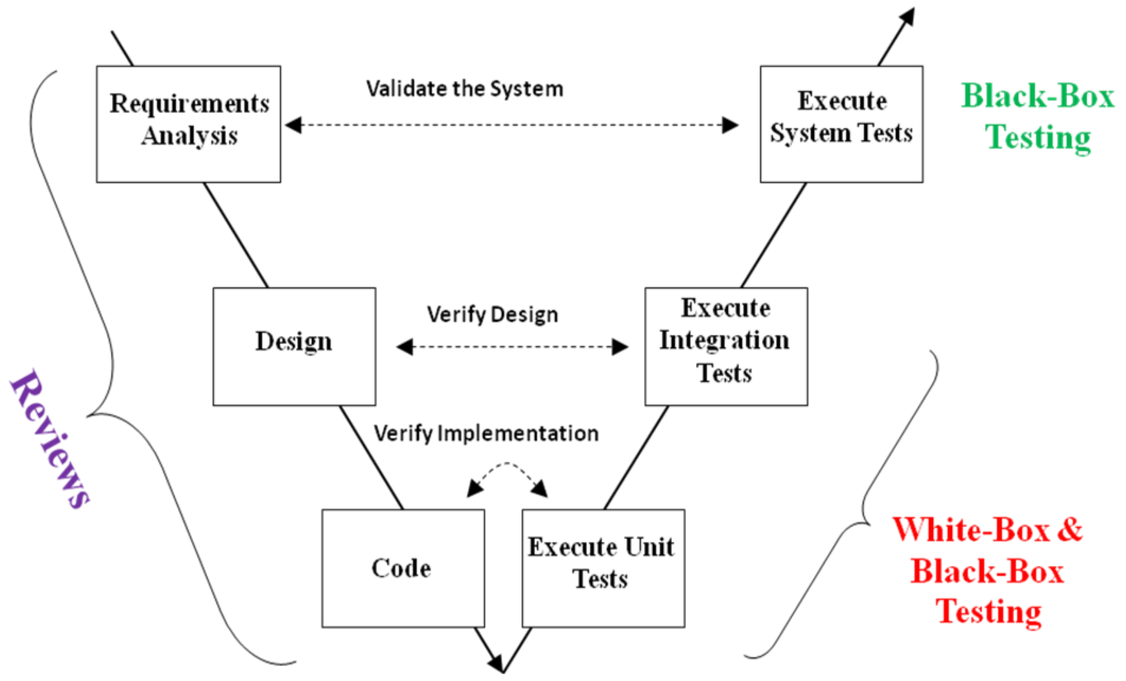


Figure 7: An overview of System Verification and Validation process

3.1 Different Methods of Testing

There are different methods for software testing, which are illustrated below.

3.1.1 Black-Box Testing

In this technique, the tester does not have any knowledge about the internal working process. The technique of testing without having any knowledge of the interior application, a tester will interact with the system interface provided by programmer only insert the input and examining the output without knowing how the system is. Sometimes unit testing is the part of Black-Box Testing.

3.1.2 White-Box Testing

This testing is a detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. For white-box testing, a tester needs to know the internal workings of the code. Sometimes unit testing is a part of White-Box testing as in our case. The programmer is an initial tester as he/she knows the logical structure.

There are different levels of testing that include different methodologies and can be used while conducting software testing. The main levels of software testing are as follows.

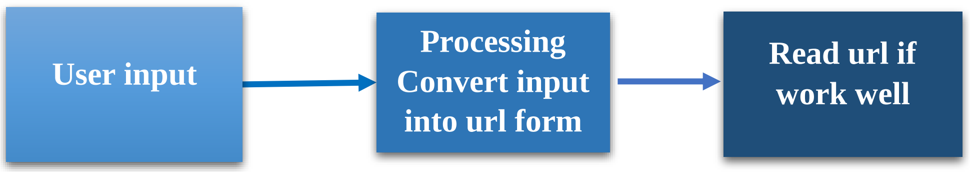
- Functional Testing
- Non-functional Testing

Functional Testing

Functional testing is a type of Black-Box testing that is based on the specifications of the software that is to be tested the application by providing the input. After that, the results are examined, that need to conform to the functionality it was intended for. The functional testing of software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

- ### Unit testing

Its focus on the smallest unit of software and ensure that they operate fine. In this proposed system we test the individual or interrelated units. This type of testing is performed by the programmer by taking random input and then check this to identify the desired result. For example in our system the user enter any query then our system converts it into url format and check the url is exist or not.



Example

User input BARACK OBAMA

url form https://en.wikipedia.org/wiki/barack_obama

- ### Integration Testing

This is the second level of testing. Integration testing is a systematic technique. This testing is performed any time in between the process to check whether any stage of the software is still working well or not. In this testing, many unit-tested modules are combined into a subsystem. As in our software when we redirect the user to desire url then we apply some nltk and NER technique to find the proper information and check with the customer (Supervisor) and if it is agreed then proceed further. Otherwise again work with this and do this or adopt the regression testing

- **Regression Testing**

This activity helps to ensure that any changes (Due to customer requirement change or for other reason) do not cause any additional error.

- **System Testing**

This testing process is focused on finding the error which occurs due to the interaction between different subsystems and system components. It also concerned with validating that the system meets functional and non-functional requirements. There are three main kinds of system testing which are mentioned below.

- Alpha Testing
- Beta Testing
- Acceptance Testing

The proposed system is based on Beta Testing because

- **Beta Testing**

Beta testing is the testing that performs by only some selected group of the friendly customer which is fully applied in our system because we test our system with supervisor and other team members.

- **Acceptance Testing**

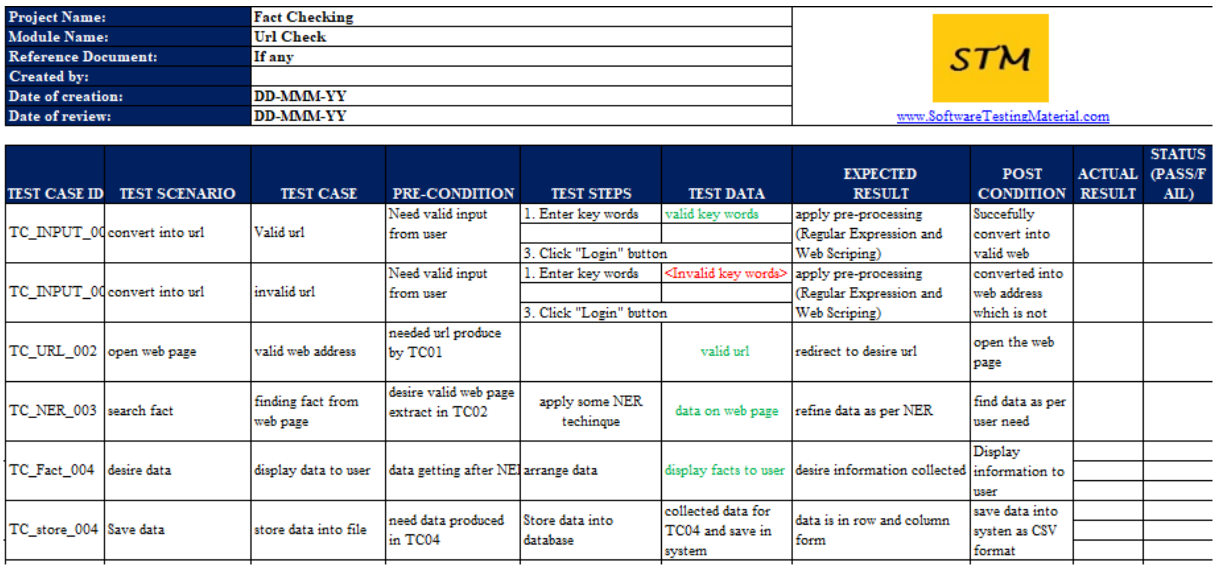
This is the testing which is performed by the customer itself to determine whether the final system delivery is accepted or rejected. A series of acceptance test is performed to enable the customer to check all requirements are matched.

Non-Functional Testing

Non-functional testing involves testing software from the requirements which are non-functional in nature but important such as performance, security, user interface, and others.

3.2 Test Cases and Test Recording Procedures

A test case has components that describe input, action, and an expected response, in order to determine if a feature of an application is working correctly [16]. A test case is a set of instructions on “HOW” to validate a particular test objective/target, and when it followed will tell us if the expected behaviour of the system is satisfied or not.



Project Name:	Fact Checking
Module Name:	Url Check
Reference Document:	If any
Created by:	
Date of creation:	DD-MMM-YY
Date of review:	DD-MMM-YY

www.SoftwareTestingMaterial.com

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_INPUT_00	convert into url	Valid url	Need valid input from user	1. Enter key words 3. Click "Login" button	valid key words	apply pre-processing (Regular Expression and Web Scriping)	Succefully convert into valid web		
TC_INPUT_00	convert into url	invalid url	Need valid input from user	1. Enter key words 3. Click "Login" button	<Invalid key words>	apply pre-processing (Regular Expression and Web Scriping)	converted into web address which is not		
TC_URL_002	open web page	valid web address	needed url produce by TC01		valid url	redirect to desire url	open the web page		
TC_NER_003	search fact	finding fact from web page	desire valid web page extract in TC02	apply some NER techinque	data on web page	refine data as per NER	find data as per user need		
TC_Fact_004	desire data	display data to user	data getting after NEI	arrange data	display facts to user	desire information collected	Display information to user		
TC_store_004	Save data	store data into file	need data produced in TC04	Store data into database	collected data for TC04 and save in system	data is in row and column form	save data into systen as CSV format		

Table 1: Test Cases for Fact Checking

For detail please click on the excel file.

test case.xls

3.3 Testing Schedule

Possible testing schedule is as follows.

General:

No	Principal	Modules	States	Prior	Start time	End time
1	*	Test the development environment	~	1	Feb 8,19	Feb 11,19
2	*	Unit testing	~	1	Feb 16,19	Feb 27,19
3	*	Integration testing	~	1	Feb 28,19	Mar 09,19
4	*	Acceptance testing	~	1	Mar 10,19	Mar 10,19

Detail:

No	Principal	Interface	States	Prior	Start time	End time
1	*	Nltk&WordNet preprocess testing	~	1	Feb 8,19	Feb 11,19
2	*	Web crawling testing	~	1	Feb 16,19	Feb 20,19
3	*	Data parsing testing	~	1	Feb 21,19	Feb 27,19
4	*	Data structure conversion testing	~	1	Feb 28,19	Mar 03,19
5	*	Database upload testing	~	1	Feb 28,19	Mar 03,19
6	*	Information retrieval testing	~	1	Mar 03,19	Mar 10,19
7	*	Information comparison testing	~	1	Mar 03,19	Mar 10,19
8	*	Pre output testing	~	1	Mar 08,19	Mar 10,19
9	*	Output testing	~	1	Mar 10,19	Mar 10,19
10	*	General test	~	1	Mar 10,19	Mar 10,19

Table 2: Possible Test Schedule

4 Overall Project Planning and Management

4.1 Methodology

In our fast-paced world of technological advancements, it is not uncommon to have the client requirements changing every other day, neither is it common to implement only one design model for an entire project. Hence it is essential at the Project planning stage to have a model that has that high level of confidence to meet with the ever-demanding requirements, design and model evolutions, code changes, code optimization, test upgrades, and changes in deployment specifications. Much of the traditional software engineering projects make use of a stepwise procedure more commonly known as the waterfall model but what separates the waterfall model from some of the more advanced methodologies is its lack of adaptability to changes [14], [15]. Hence the method which will be used in approaching our project is the agile methodology.

An agile methodology is in a way an enhancement to the waterfall model. In this methodology, the changes in any stage of the SDLC is maintained by carrying out a newly updated prior stage in tandem with the current stage in the form of an iterative approach. Agile as such does not necessarily have a fixed process, rather it is a process that engages the software development principle members and stakeholders to respond in the case of change rather than following a fixed plan [15]. In other words, respond to unpredictability rather than work on a fixed system.

In our case of identifying the facts from a source like Wikipedia. The source like Wikipedia is not static and is continuously evolving and furthermore a future requirement might involve having the system work on other sources and adapt our code to run on a more advanced technique of data processing or machine learning or furthermore to fix any defects that may arise of such future versions, hence we are employing the use of an Agile approach of methodology. As the principal use of our agile approach, we will be making use of the Scrum methodology which will have us communicate through a common portal in our case the JIRA tool, enable us to create Sprints and Sprint reviews and set backlogs to further ensure pending tasks to synchronize action and ensure smooth transition of our project unto deployment and further.

4.2 Necessary tools to deploy the proposed system

In order to complete the task of the entire project, relevant tools need to be used. Several important tools, along with the usage rules are as follows.

- The first tool is NLTK, which mainly includes 5 functions.

  I. **Search term**

  The search term is the context in which the word appears. In addition to the initial exploration of the semantics of words in the text, this function can also be applied to the task of word retrieval.

  II. **Finding approximate words**

  According to the context of the word, if they find similar structures, they are considered to be approximate words.

  III. **Lexical diversity**

  Compare the richness of vocabulary between different texts by calculating the value of the length of the different words.

  IV. **Vocabulary map**

  This method can be used if we want to see how the relevant words appear throughout the frequency, that is, when certain words appear in the front, middle, and back of the text.

  V. **Combination of text.**

- The second tool is Scikit-learn, whose basic functions are mainly divided into six parts, called classification, regression, clustering, data dimensionality reduction, model selection, and data pre-processing.

  i. Classification refers to identifying the category of a given object, which belongs to the scope of supervised learning. The most common application scenarios include spam detection and image recognition.

  ii. Regression refers to predicting continuous value attributes associated with a given object. The most common application scenarios include predicting drug response and predicting stock prices.

iii. Clustering refers to automatically identifying a given object with similar attributes and grouping them into collections, which belongs to the category of unsupervised learning. The most common application scenarios include customer segmentation and grouping of test results.

iv. Data dimensionality reduction refers to the use of dimensionality reduction techniques such as principal component analysis (PCA), non-negative matrix factorization (NMF) or feature selection to reduce the number of random variables to be considered. The main application scenarios include visualization processing and efficiency improvement.

v. Model selection refers to the comparison, verification and selection of a given parameter and model. Its main purpose is to improve the accuracy through parameter adjustment.

vi. Data pre-processing refers to feature extraction and normalization of data.

- Third tool is Pandoc. The Pandoc is a development of mark-up language conversion tool, which can realize format conversion between different mark-up languages. Pandoc is written in Haskell language, realizes interaction with users in command line form, and supports multiple operating systems. Similarly, Pandoc can generate a variety of target formats for the above source files, including word processing software format, e-book format, technical document format, page layout format, etc.

- Forth tool is WordNet, which is a different English dictionary. It has two main purposes. It is both a dictionary and a dictionary. It is easier to use than a simple dictionary or dictionary. It also supports automatic text analysis and artificial intelligence applications. WordNet internal structure is, In WordNet, nouns, verbs, adjectives, and adverbs are each organized into a synonym network. Each synonym collection represents a basic semantic concept, and these collections are also connected by various relationships.

- The fifth tool is NumPy. Numpy is an extension library for the Python language. Supports a large number of high-dimensional arrays of dimensions and matrix operations, in addition to providing a large number of mathematical function libraries for array operations. The core function of NumPy is the "ndarray" data structure, which is an array object that represents multiple dimensions, homogeneity, and fixed size. Moreover, a data type object that is related to this array describes the data format of its array elements. The grammar mainly includes four kinds, and that are creating arrays, basic operations, global methods, and linear algebra.

- The final one is Beautiful Soup which is a Python package that includes parsing HTML, XML documents, and fixing documents with errors such as unclosed tags. This extension pack creates a tree for the page to be parsed in order to extract the data, which is very useful for network data collection.

4.3 Project Execution Plan

As the method we have used in our project curtails to an agile approach, the plan below is simply a skeleton sequence of how we will approach this project and there is always room for accommodating necessary changes that may result during the Software Development Life Cycle. In other words, this project development plan holds the current process, or first iteration for development of the project and will be changed as and when needed.

General Plan:

No	Interface	Start time	End time
1	Perform Literature survey and come up with a solution.	Jan 21st 2019	Feb 1st 2019
2	Build an SRS Report, and Design Document	Feb 1st 2019	Feb7th 2019
3	Testing of Development Environment and components	Feb 8th 2019	Feb 11th 2019
4	Identifying Source datasets, and data collection	Feb 11th 2019	Feb16th 2019
5	Web crawling coding and testing	Feb 16th 2019	Feb 20th 2019
6	Nltk pre-processing and Data parsing coding and testing.	Feb 21st 2019	Feb 27th 2019
7	Data structure conversion and data structure testing completion	Feb 28th 2019	3rd March 2019
8	Information retrieval coding and output testing.	3rd March 2019	10th March 2019
9	Output Fact Checking from query. And deployment.	10th March 2019	10th March 2019
10	Building our Final Report, Risk assessment, and deliverable documentation.	10th March 2019	21st March 2019

Table 3: Overall project execution plan

4.4 Change Management

Something that is really important in the life of any good software is its adaptability to change. This adaptability to change would ensure that given software would be stable in the market. To ensure change happens smoothly, one must focus more on how one’s software can learn to adapt in terms of a crisis, in terms of a complete model change, or data change.

Our case primarily focuses on extracting information from a single source like Wikipedia. Given the simplicity of our model, and its approach to web scraping for the relevant information from a website. It will be quite simple to adapt our system to handle say another organization’s websites, simply through small modification to the code. Another focus on change will occur in terms of defect management and crisis management. Having our system work with just one source of a website will nullify most defects. Furthermore, it is also possible to run our code by web scraping through multiple sources, hence there are chances of coming across some defects through fetching information or data pre-processing, these we will manage through making use of simplistic yet efficient language processing models. Also, most of the defects will arise from the unwanted crisis in terms of power failure, or an error with loading server. As far as a power failure is concerned we will be making use of storing a backup database of our processed information per each query.

Further enhancements in our code in the future can include management of concurrent query processing. As far as server loading errors are concerned, the websites used in our case, Wikipedia are publicly well-renowned websites working in an integrated cloud environment; hence this eliminates the need of concern for server going down which can be eliminated. Furthermore, this project curtails to websites that are continuously upgrading data and hence all data updates will be properly managed at any time.

References

[1] P. B. Brandtzaeg, and A. Folstad, "Trust and distrust in online fact-checking services", *Communications of the ACM*, vol. 60, no. 9, pp. 65-71, 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=3122803. [Accessed: February 2, 2019].

[2] C. Conforti, M.T. Pilehvar, and N Collier, "Towards Automatic Fake News Detection: Cross-Level Stance Detection in News Articles", *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp. 40-49, 2018. [Online]. Available: http://www.aclweb.org/anthology/W18-5507. [Accessed: February 2, 2019].

[3] V. Huynh, and P. Papotti, "Towards a Benchmark for Fact Checking with Knowledge Bases", *Companion Proceedings of the The Web Conference 2018*, pp. 1595-1598, 2018. [Online]. Available: https://dl.acm.org/citation.cfm?doid=3184558. [Accessed: February 2, 2019].

[4] N. Hassan, F Arslan, C li, and M. Tremayne, "Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1803-1812, 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=3098131. [Accessed: February 2, 2019].

[5] "The State of Automated Factchecking", *fullfact.org*, 2016. [Online]. Available: https://fullfact.org/blog/2016/aug/automated-factchecking/. [Accessed: February 2, 2019].

[6] D. Jimenez, "Towards Building an Automated Fact-Checking System", *ACM International Conference on Management of Data*, pp. 7-9, 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=3055176. [Accessed: February 2, 2019].

[7] F. Wu, and D. S. Weld, "Autonomously Semantifying Wikipedia", *ACM conference on Conference on information and knowledge management*, pp. 41-50, 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=1321449. [Accessed: February 2, 2019].

[8] F. Wu, and D. S. Weld, "Automatically Refining the Wikipedia Infobox Ontology", *7th international conference on World Wide Web*, pp. 635-644, 2008. [Online]. Available: https://dl.acm.org/citation.cfm?id=1367583. [Accessed: February 2, 2019].

[9] J. Nothman, J. R. Curran, and T. Murphy, "Transforming Wikipedia into Named Entity Training Data", *Proceedings of the Australasian Language Technology Workshop*, vol. 6, pp. 124-132, 2008. [Online]. Available: http://www.aclweb.org/anthology/U08-1016. [Accessed: February 2, 2019].

[10] "Fukushima Daiichi nuclear disaster", *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Fukushima_Daiichi_nuclear_disaster. [Accessed: February 2, 2019].

[11] V. Rosas, B. Kelinberg, A. Lefevre and R. Mihalcea, "Automatic Detection of Fake News", *8th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, art no. 82, 2015. [Online]. Available: https://dl.acm.org/citation.cfm?id=2857152. [Accessed: February 2, 2019].

[12] T. A. Pender, "*UML Weekend Crash Course*", New York, Wiley Publishing, 2002.

[13] "IEEE Recommended Practice for Software Requirements Specifications", *Software Engineering Standards Committee of the IEEE Computer Society*, 1998.

[14] "Security Development Lifecycle for Agile Development", *Microsoft*. 2009.

[15] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements Engineering and Agile Software Development", *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 2003.

[16] I. Sommerville, "*Software Engineering*", Addison Wesley, 2010.

Appendix

Document Approvals