



A Machine Learning approach to detect Fake News

University of the West of England

Name: Juan Camilo Rodriguez

Student ID: 16020551

Module: Digital Systems Project

Module Code: UFCFXK-30-3

Word Code:

TABLE OF CONTENTS

| | |
|--|-----------|
| TABLE OF CONTENTS | 2 |
| ACKNOWLEDGMENTS | 5 |
| TABLE OF FIGURES | 6 |
| TABLE OF TABLES..... | 7 |
| INTRODUCTION | 9 |
| 1.1. Define Problem | 9 |
| 1.2. Aim and Objectives | 11 |
| 1.3. Chapter Breakdown..... | 12 |
| LITERATURE REVIEW | 13 |
| 2.1. Fake News | 13 |
| 2.1.1. What is Fake News | 13 |
| 2.1.2.Types of Fake News | 13 |
| 2.2. Linguistics Concepts | 15 |
| 2.2. How is the meaning captured from phrases? | 15 |
| 2.3. Relevant Research | 16 |
| 2.3.1. Techniques used..... | 16 |
| 2.3.2. In Depth Machine Learning Approach | 16 |
| 2.3.3. Prior Attempts | 16 |
| 2.4. Gaps Identified..... | 17 |
| 2.5. Hypotheses..... | 17 |
| REQUIREMENTS | 18 |

| | |
|--|-----------|
| 3.1. MoSCoW Methodology | 18 |
| 3.2. Functional Requirements | 19 |
| 3.3. Non-Functional Requirements | 20 |
| METHODOLOGY | 22 |
| 4.1. Software Development Strategies | 22 |
| 4.1.1. Traditional Development Models | 22 |
| Waterfall Model | 22 |
| 4.1.2. Agile Development Models | 23 |
| SCRUM | 23 |
| 4.2. Method | 23 |
| 4.2.1. SCRUM Approach for Project | 24 |
| DESIGN, IMPLEMENTATION AND TESTING | 27 |
| 5.1. Software Development Tools | 27 |
| 5.2. Project Architecture | 28 |
| 5.2.1. Web Architecture and Workflow | 28 |
| What is MVC? | 28 |
| MVC and MVT | 29 |
| 5.2.2. Machine Learning Architecture | 29 |
| 5.3. Design, Implementation and Testing | 31 |
| 5.3.1. Stage 1: Front-end | 31 |
| Sprint Backlog | 31 |
| Planning & Design | 32 |
| Wireframe | 32 |
| Implementation | 33 |
| Development Tools | 33 |
| Django Template Language | 34 |
| Template Inheritance | 35 |
| Django Snippets | 37 |
| Testing | 39 |

| | |
|--|-----------|
| Review | 40 |
| 5.3.2 Stage 2: Back-End | 41 |
| Sprint Backlog | 41 |
| Planning & Design | 42 |
| Implementation | 43 |
| <i>Development Tools</i> | 43 |
| Testing | 47 |
| Review | 48 |
| 5.3.3 Stage 3: Machine Learning | 49 |
| Sprint Backlog | 49 |
| Planning & Design | 49 |
| <i>Sequence Diagram</i> | 49 |
| Implementation | 50 |
| Testing | 50 |
| Review | 50 |
| PROJECT EVALUATION | 52 |
| REFERENCES | 53 |
| APPENDICES | 55 |
| Appendix A: Project Tools | 55 |

Juan Camilo Rodriguez

16020551

Spread the News

UFCFXK-30-3

ACKNOWLEDGMENTS

TABLE OF FIGURES

| | |
|---|----|
| FIGURE 1 - WATERFALL MODEL | 22 |
| FIGURE 2 - SCRUM SPRINT | 24 |
| FIGURE 3 -PROJECT SCRUM DEVELOPMENT | 25 |
| FIGURE 4 - MVC SYSTEM ARCHITECTURE | 28 |
| FIGURE 5 - MVT SYSTEM ARCHITECTURE | 29 |
| FIGURE 6 - TYPICAL MACHINE LEARNING SYSTEM ARCHITECTURE | 30 |
| FIGURE 7 - PROJECT'S MACHINE LEARNING SYSTEM ARCHITECTURE | 30 |
| FIGURE 8 - WEBSITE WIREFRAME | 32 |
| FIGURE 9 - PART OF HOME HTML FILE | 34 |
| FIGURE 10 - BASE TEMPLATE (BASE.HTML)..... | 35 |
| FIGURE 11 - EXAMPLE BLOCK TAG | 36 |
| FIGURE 12 - TEMPLATE MACHINE LEARNING SNIPPET | 38 |
| FIGURE 13 - WHERE MACHINE LEARNING SNIPPET IS ADDED ON HOME PAGE TEMPLATE | 38 |
| FIGURE 14 - PROJECT URL PATH..... | 43 |
| FIGURE 15 - WEBSITE URL PATHS..... | 44 |
| FIGURE 16 - EXAMPLE VIEW | 45 |

TABLE OF TABLES

| | |
|---|----|
| TABLE 1 - CHAPTER BREAKDOWN | 12 |
| TABLE 2 - FUNCTIONAL REQUIREMENTS | 19 |
| TABLE 3 - NON-FUNCTIONAL REQUIREMENTS | 20 |
| TABLE 4 - SCRUM SPRINT STEPS | 25 |
| TABLE 5 - REQUIREMENTS IN CORRESPONDING STAGE | 26 |
| TABLE 6 - LANGUAGE: PYTHON | 27 |
| TABLE 7 - FRAMEWORK: DJANGO | 27 |
| TABLE 8 - LIBRARY: SCIKIT-LEARN | 27 |
| TABLE 9 - REQUIREMENTS FOR FRONT-END | 31 |
| TABLE 12 - REQUIREMENTS FOR BACK-END | 41 |
| TABLE 10 – UNDERSTANDING PROJECT URL PATHS..... | 44 |
| TABLE 11 – UNDERSTANDING WEBSITE URL PATHS | 45 |
| TABLE 13 - REQUIREMENTS FOR MACHINE LEARNING..... | 49 |

Report quality

SUPERVISOR

Report is incomplete and contains many notes to the author rather than the complete work.

Missing an abstract.

Overuse of superlatives and personal feelings

Tables and figures are incomplete or missing.

Grammatical and spelling errors throughout the document.

References section contains some references that are unused in the document

MARKER

No use of the appropriate terminology. Fails to describe the problem and the work done. Shows a lack of structure, comprehensibility, clarity and grammatical quality. Results and products of the project are insignificant, poor and/or incomplete.

Report has not be finalized and contains many working notes rather than complete content for the report.

Chapter 1:

INTRODUCTION

1.1. Define Problem

News corporations are critical, necessary and influential on a global scale. They are the main and most consumed source of communication that informs about events that occur worldwide. Due to this, news organisations are expected to deliver rich, trustworthy and quality content. However, the way our society functions does not always and necessarily award those that are honest, hard-working and/or humble, especially when it comes to honesty.

These news companies with time have become more oriented and focused on what will generate the most profit. This has led to a very grievous problem with news reports being misleading and containing false information. Misleading and misinformed news are also known by the term of **fake news**. This is not a new term; it is just one that has been catching attention lately. Nowadays there has been a shift, and many people are starting to question the content from news reports. In fact, there has been survey studies questioning the public about news and their opinions. An eye-catching fact in the Reuters Institute annual report (2019) was how continuously over time the trust level dropping, worst case of 2019 was the United Kingdom with 70% of the people questioning the content from news media. The trust level worsens when it is about politics or when a country is going through eventful circumstances (Reuters Institute, 2019).

It is a fact that fake news is a pernicious problem, it just becomes difficult to determine what degree and how much of an impact it has and will have in the future. A recent event which saw some of the consequence of fake news was the elections of United States of America in 2016. Postelection there were findings of fake propaganda and the usage of fake news to fool Americans, here are some of the reported cases:

- Within five months prior to the elections, there was roughly 170 million tweets. Of which 30 million were about the election. It was found that 7.5 million of those 30 million, which equates to 25%, contained fake news and/or were extremely bias (Bovet, A., and Makse, H.A, 2019)
- Guess, A., Nyhan, B. and Reifler J. (2018) found out that 1 in 4 Americans visited a fake news website.

- Additionally, Guess, A.'s (2018) report also claim that more than 40% of individuals do not know and/or did not conduct fact-checking when reading news about the elections.

Another motivation for this project is that this year the United States either re-elects or elects a new president. Based off prior presidential run, many companies and individuals have attempted to prevent for this time the spread of fake news. This project is another attempt to aid and counter fake news for the public.

One last point, it terribly sad to see how much money twists people. Really makes some individuals do some outrageous actions. This is exactly what has happened with news companies. What once was their aim to produce honest news is practically non-existent today. Even if there are honest and good journalist, there is always the motive to hesitate whether what you are reading is true. Users should not be worried about what the news publishes as its meant to be re-writing an event. The fact that 1 in 2 people do not trust news is no acceptable and just proves how society crumbles (Reuters Institute, 2019). Therefore, a standard need to be set, and if news companies will not make the effort to produce such content then it us up to others to catch their lies and make them public. Which will consequently make improve their standards.

Aims and objectives

SUPERVISOR

The objectives are not detailed enough. They should be based on critical review of the relevant products and projects found in the literature review. Objectives relate to the expected outcomes of the project

MARKER

Aims barely specified;not clear if aims and objectives have been met.

1.2. Aim and Objectives

These are the aim and objectives set for this project

AIM: *To create an intelligent tool that can autonomously be capable from fed text, determine whether a given news report is real or fake*

OBJECTIVES:

1.3. Chapter Breakdown

The remaining of the project will be going through the chapters mentioned in the table below.

Table 1 - Chapter Breakdown

| Chapter | Description |
|----------------------------------|---|
| Literature Review | This chapter goes over content that was relevant for the problem at stake. Terminologies, concepts, and precious attempts are brought up. |
| Requirements | In this chapter the requirements that were proposed are stated. As well as determining their importance towards the project. |
| Methodology | This chapter goes over the strategy that was used for the software development of the project. |
| Design, Implementation & Testing | In this chapter the planning, designing, development and testing of the software is introduced. In essence, the creation of the application |
| Project Evaluation | In this chapter a reflective and critical analysis is made on the project conducted and determining how successful the application was. |

Literature review

SUPERVISOR

The literature review gives a reasonable background overview on the different types of Fake News.

Section 2.3.1 lists some approaches that have been previously used to detect Fake News, but does not give sources, or critically examine previous works.

Contains incomplete sections and tables.

Alessandro and Marcelloni reference missing from References section.

MARKER

Poor analysis of relevant works. Demonstrates little or no insight into the problem.
Incomplete in parts, with little critique offered for what has been discussed.

Chapter 2:

LITERATURE REVIEW

This chapter introduces the research conducted for this project. Brings

2.1. Fake News

This section

2.1.1. What is Fake News

There are a lot of misconceptions and definitions when it comes to the term of **fake news**. The term definition really relies from what perspective and branch of knowledge you are using fake news for (Clayton, K., 2019).

This project will follow the definition of fake news by Parikh, S.B. and Atrey, K. (2018), which says “Fake news can be any content that is not truthful and generated to convince its readers to believe in something that is not true.” Regardless of the term, fake news has become a representation/symbol of false information in press sources.

An additional controversy is with terms that are similar to fake news, terms such as: “fraudulent news”, “misleading news”, and more. Some papers have different views as to what each mean and some treat these similar terms as synonyms for fake news. Therefore, it is worth noting that this project will treat those variations as synonyms unless mentioned otherwise.

2.1.2.Types of Fake News

As seen from the above subsection controversy, the reader may also wonder “what is considered fake news?” or “what properties classifies news as fake?”. Once again this really depends on the research conducted and what the authors decide to portray fake news as. Therefore, this project will also be using Rubin, Chen and Conroy (2016) to determine the types and what classifies as fake news:

1. **Serious Fabrications**

This type of fake news can be found mainly in **yellow press** and tabloids. These are areas of journalism where exaggerated headlines are used to catch the attention of readers. Along with that the story tends to be scandalous and over-dramatic. Some examples of this type of fake news are found in gossip columns, crime stories and astrology.

2. **Large-scale Hoaxes**

This type is similar to "Serious Fabrications", as the intentions to lie to the audience. However, what differs with this type are the malicious intents of the journalist(s) with their publication. These writers are more careful with the content and wording of their reports, to make them sound as legitimate as possible. This is for any reader that is uninformed or not cautious to be misled in some manner, typically for the benefit of the journalist(s)

The degree of severity with this type is above the other types of fake news. For instance, in 2016, an individual was deceived and entered to a restaurant firing an armed weapon (Tandoc Jr, E.C *et al.*, 2017). This was all due to some distinct political views that the journalist had which led them write about a scandalous hoax about one of the candidates for presidency.

3. **Humorous Fakes**

The news produced by this type are for individuals that are conscious of the topic and that can realise that the news article produces was fabricated with the intentions to humour the audience. However, for readers that are unaware of the topic of the written article will most likely take it as a factual news reportage. Which in consequence will have the same effect as the other types of fake news.

2.2. Linguistics Concepts

2.2. How is the meaning captured from phrases?

2.3. Relevant Research

2.3.1. Techniques used

2.3.2. In Depth Machine Learning Approach

2.3.3. Prior Attempts

2.4. Gaps Identified

2.5. Hypotheses

Requirements analysis

SUPERVISOR

Good use of MoSCoW method for requirements analysis.

Requirement table should ideally contain a columns for the ID, the requirement, the dependencies (if any), priority, and description.

Non-Functional requirements have the same IDs as functional requirements.

Both tables incomplete with only 10 out of a minimum of 20 requirements detailed.

MARKER

Little or no description, analysis and presentation of the requirements.

Chapter 3:

REQUIREMENTS

This chapter will go through the requirements that have been set for the development. They have been ordered in a prioritisation manner, which will help complete elements of the project that are of more importance over others. The requirements have also been split into the services that will and will not be used, as well as how such services will be measured.

3.1. MoSCoW Methodology

There is the uncertainty of the amount of time that can be invested into this project is **unknown**. Therefore, it is of importance to prioritise the requirements into essential and non-essential as that will provide a target as to what is being achieved. A common method to be able to hierarchize the requirements is by using requirement prioritisation methods.

A well-known requirement prioritization technique is **MoSCoW**, it is a common practice in the agile development environment (Vestola, M., 2010). This method has four different levels of prioritisation (Kuhn, J., 2009). These are the definitions proposed by Achimugu P. *et al* (2014) for each of the levels:

| | |
|---------------|---|
| Must | <i>Requirements are not negotiable; the failure to deliver these requirements would result in the failure of the entire project</i> |
| Should | <i>Features that would be nice to have if at all possible</i> |
| Could | <i>Features that would be nice to have if at all possible but slightly less advantageous than the "S" (Should)</i> |
| Won't | <i>These requirements are not unimportant, but they will definitely not be implemented in the current software project. They may, at a later stage, be created.</i> |

In addition, the requirements have been split up into **functional requirements** and **non-functional requirements**. This project will be using the following definitions:

**Functional
Requirements (FR)**

"Functional requirements specify the functions of the system, how it records, computes, transforms, and transmits data." (Lausen, S., 2002)

**Non-Functional
Requirements (NFR)**

"Non-functional requirements describe the nature and limitations on the project instead of its functionality, also this term describes the non-behavior aspects and attributes of the system including usability, portability, security, understandability, reliability, and modifiability. In general, the non-functional requirements highlight the requirements that describe "how good" the software." (Hudaib, A. et al, 2018)

3.2. Functional Requirements

In this section the services that the final application must, should, could and won't have are covered. These requirements set out a standard for the application at the end of the project as well as other functions that can be used for further development.

Table 2 - Functional Requirements

| ID | Priority | Requirement Description |
|-------|----------|---|
| FR-01 | Must | The intelligent system will use Machine Learning techniques only to be able to determine the legitimacy of news reports |
| FR-02 | Must | The system will only accept English written newspapers |
| FR-03 | Must | The only accepted input from the user should be URLs, anything else will be rejected |
| FR-04 | Must | The system has to be able to extract the content from the news report |
| FR-05 | Must | The system should be able to determine if the given URL comes from a news source |
| FR-06 | Must | The Machine Learning application after thorough testing should be at least achieving 75% |

| | | |
|--------------|---------------|---|
| FR-07 | Should | There system will announce to the user when the intelligent system has come to a decision with the given URLs |
| FR-08 | Should | The system should be able to process more than one URL at a time |
| FR-09 | Should | The systems output will provide a reason as to how it came to its output decision |
| FR-10 | Should | The website application should be able for other users to then extend upon the current project easily |
| FR-11 | Should | The machine learning application should be ready to use for other users to test with other datasets and/or models |
| FR-12 | Should | The machine learning within the website should be easily exchangeable if needed |
| FR-13 | Won't | The system will use different artificial intelligent approaches to create a hybrid system |

3.3. Non-Functional Requirements

In this section, it will be covered the requirements that will judge the functional requirements. Essentially, these requirements/judgements will allow quantitatively and qualitatively measure the application.

Table 3 - Non-Functional Requirements

| ID | Priority | Requirement Description |
|---------------|-----------------|--|
| NFR-01 | Must | The website will have a responsive web design for desktop users |
| NFR-02 | Must | The websites functionality shall fully work on Google Chrome browser engine. |
| NFR-03 | Must | The project should attempt to follow the most up-to-date technologies and practices. |
| NFR-04 | Should | The website should work on the most popular browser engines |

| | | |
|---------------|---------------|--|
| NFR-05 | Should | The website should follow recent styling trend |
| NFR-06 | Should | The website should process the data for the Machine Learning dynamically |
| NFR-07 | Should | The website should be able to classify in less than a minute per news report |
| NFR-08 | Should | Operations should be carried out in a time frame that does not cause the user to believe that the system has become unresponsive |
| NFR-09 | Should | The programming should adopt Python PEP8 standard format |
| NFR-10 | Could | Have security measures for malicious input |
| NFR-11 | Could | The project to be entirely documented |

Chapter 4:

METHODOLOGY

In this chapter different approaches to develop software will be presented and compared. Followed by the technique that this project will use throughout the project, with some modifications to the method.

4.1. Software Development Strategies

Software Development Life Cycle (SDLC) is defined as the method that is used to develop software, these models have steps and techniques as to how the software should be developed in order to achieve the highest success (Mishra, A. and Dubey, D., 2013). The SDLC models can generally be categorised into “Traditional Development” and “Agile Development” (Leau, Y.B. *et al*, 2012).

4.1.1. Traditional Development Models

The most recognised traditional software development models include: Waterfall, V-Model, and many more. In this section, Waterfall will be explained and later discussed with the agile methodology.

Waterfall Model

Model introduced by Winston Royce in 1970 (Mishra, A. and Dubey, D., 2013), it a linear development method, what this means is that a phase once completed is not revisited again. It is easy model to understand, however, its rigid architecture creates is hard to be able to correct mistakes from previous phases.

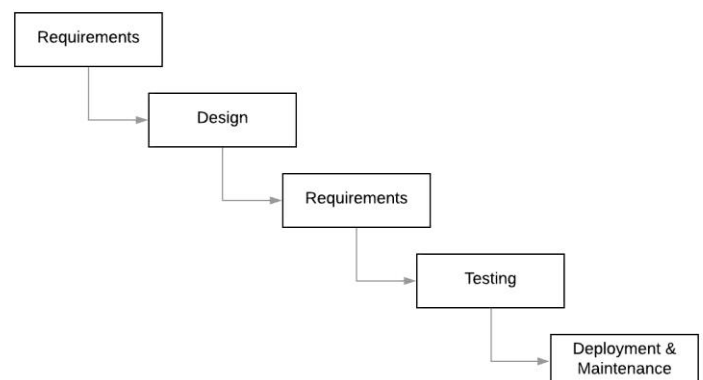


Figure 1 - Waterfall Model

4.1.2. Agile Development Models

A group of seventeen individuals, referred to as “the Agile Alliance”, met in 2001 and discussed about the need for an alternative way to develop software. This meeting is now known as “The Agile Manifesto and what sprung was a new series of “lightweight methodologies” and rules for Software Development and now also entering into other business frameworks (Hudaib, A. *et al*, 2001).

Within Agile there are several popular models, most popular ones include: SCRUM, Extreme Programming and more. A benefit with agile methods is the adaptability that can be used if necessary, this allows for unforeseen problems to be addressed and modified on time.

SCRUM

SCRUM like other agile models is a dynamic method that revolves around teamwork, working software, customer interaction and adaptability to change (Schwaber and Sutherland, 2017). What is of interest about SCRUM is how the sprints work and what they recommend developing software.

Sprints ~~is was creates~~ SCRUM such a flexible Software Development approach. Each Sprint is given a goal, all must work to be able to accomplish this target in the proposed time frame. To be able to achieve the highest success rate the sprint is split into further stages. There is no standard as to what steps need to be taken as this varies with every project. Generally speaking, the recommended stages include planning, development work, sprint review and sprint retrospect (Schwaber and Sutherland, 2017).

4.2. Method

SCRUM is the chosen methodology to help control and organise the development of the project. This method is based upon the principles of the Agile Manifesto, along with its own set of techniques. It provides tremendous flexibility and adaptability for the development of the application. This is a particular point as to why SCRUM was chosen because it is likely that there will be at some point a delay as there are many concepts being learnt for the first time and other work being done simultaneously.

As Verheyen, G. (2013) points out, SCRUM is not a strict process that has obligatory and exhaustive steps. Instead there are proposed techniques that is left for the team to decide whether or not to opt. This is brought up as this project will be using SCRUM as its basis, however, there has been some additional decisions that are not particularly part of the SCRUM framework that will disclosed in the following paragraphs.

4.2.1. SCRUM Approach for Project

It has been decided that the total project sprints will be divided in these stages: Front-end, Back-end and Machine Learning. All of these contain parts of the requirements that are of great importance (the MoSCoW “must” requirements). Therefore, the idea behind this method is that in the worst-case scenario the end application will contain all of the “must” requirements. Anything else will be an application with additional features, which would improve the User Experience (UX) and the usages for the software.

Each stage is comprised of a number of sprints. Each sprint will follow roughly conventional SCRUM sprint protocol, [Figure 2](#) gives and illustrative representations as to what comprises each sprint and what exactly does each step of the sprint mean.

Figure 2 - SCRUM Sprint

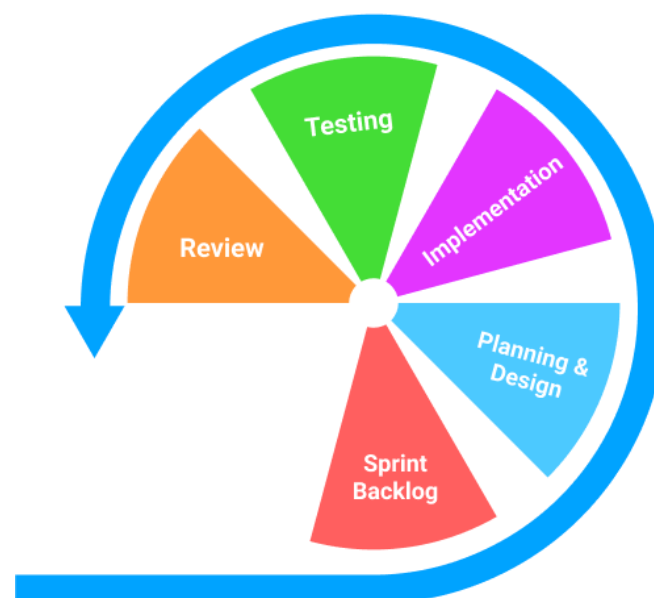
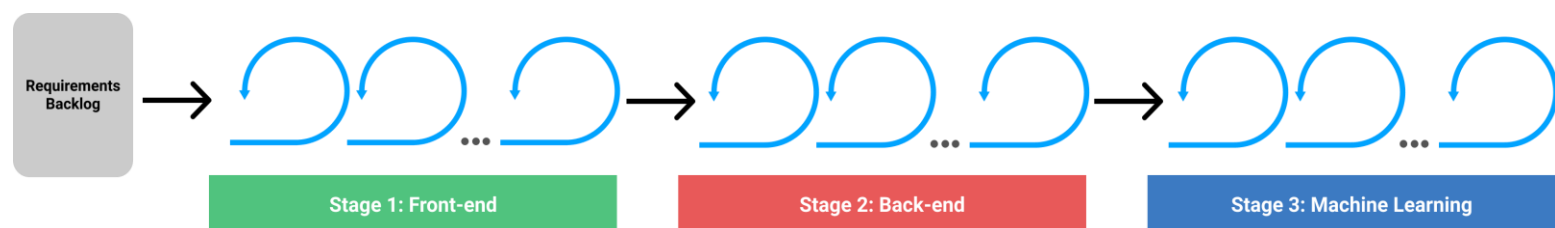


Table 4 - SCRUM Sprint Steps

| SCRUM Sprint Step | Description |
|------------------------------|---|
| Sprint Backlog | <p>Requirements, from Chapter 2, that are related to the stage name are selected to be completed within the sprint.</p> <p>This stage formalizes the specific tasks that need completion.</p> |
| Planning & Design | <p>Focuses on how the current stage will be implemented. Therefore, will have a series of diagrams, sketches, charts, or other relevant planning or designing techniques that have been used to condensate the implementation process.</p> |
| Implementation | <p>Overviews the key implementations, in general will consist mainly around code. This is conveyed through snippets of code and explain the chosen approaches/techniques.</p> <p>This stage highly focuses on the sprint backlog being developed and completed.</p> |
| Testing | <p>Testing that the program functions adequately. Regardless, most emphasis on the testing will surround that the sprint backlog has been properly implemented.</p> |
| Review | <p>At the end, some observations are left for later development and potential comments that can be reused in "Project Evaluation" chapter.</p> |

Overall, the project will have the following development overlook:

Figure 3 -Project SCRUM Development



In addition, the table below contains the same requirements brought up in *Chapter 3: Requirements*. However, now they have been categorised to their corresponding stage section.

Table 5 - Requirements in corresponding Stage

| Stages | Requirement ID's |
|------------------|--|
| Front-end | FR-07, FR-10, NFR-01, NFR-02, NFR-03, NFR-04, NFR-05, NFR-06, NFR-08 |
| Back-end | FR-02, FR-03, FR-04, FR-05, FR-08, FR-09, FR-10, FR-12, NFR-03, NFR-06 |
| Machine Learning | FR-01, FR-06, FR-11, NFR-03, NFR-07 |

Chapter 5:

DESIGN, IMPLEMENTATION AND TESTING

5.1. Software Development Tools

In this section the most important software development language, framework and library are covered. In *Appendix A: Project Tools* all the software utilities are covered and explained as to how they were used.

Table 6 - Language: Python

| Language – Python | | URL: https://www.python.org/ |
|---------------------|---|--|
| <i>Description:</i> | Python is an extremely popular language in multiple fields of Computer Science, it is used in Data Science, Web Development and more. Its readability and flexibility create an attraction to it. On top of that Python has a very dedicated community for creating libraries and Frameworks for all disciplines. | |

Table 7 - Framework: Django

| Framework – Django | | URL: https://www.djangoproject.com/ |
|---------------------|--|--|
| <i>Description:</i> | This is one of the many frameworks that has been dedicated for Python. It is mainly used for Web Development. Such framework is meant to help developers, mainly with menial tasks that are common within many projects. | |

Table 8 - Library: Scikit-Learn

| Library – Scikit-Learn | | URL: https://scikit-learn.org/stable/ |
|------------------------|--|--|
| <i>Description:</i> | This is a Machine Learning library. This library mainly covers supervised and unsupervised learning algorithms. Similarly to Django, this library is meant for developers to use the pre-defined models. | |

5.2. Project Architecture

This project is about the production of two different applications and then combining both of them into a single application.

5.2.1. Web Architecture and Workflow

Since Django is a framework, then its intentions are to make the life of developers much easier. One of which is having an architectural system for development. The architectural design that Django follows is known as **Model-View-Template (MVT)**. It is practically a synonym for the more commonly recognised design pattern known as **Model- View-Controller (MVC)**.

What is MVC?

Model-View-Controller is a software system design that is used greatly on the web, due to the typically of the two different roles of Front-end and Back-end. MVC is split up into three components:

- **Model:** This section is meant for all the data and logical operations of the program. (Krasner, G.E. and Pope, S.T., 2014)
- **View:** This is the actual presentation of the application, known as the User Interface. No logical process is done in this area of the application. (Krasner, G.E. and Pope, S.T., 2014)
- **Controller:** This is the connecting point between the View and the Model, can be referred as the middleman. It is what will collect the input from the user and send it to the model. Can also collect an item from the model and present it to the view. (Krasner, G.E. and Pope, S.T., 2014)

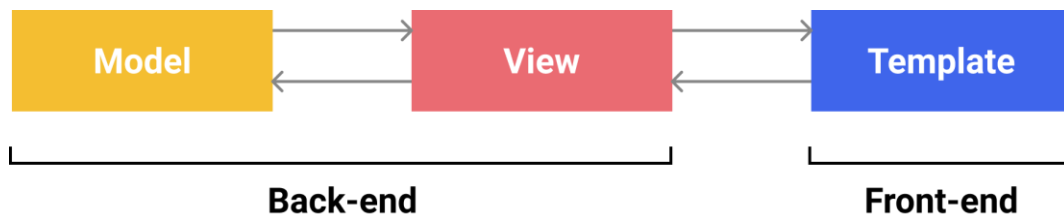
Figure 4 - MVC System Architecture



MVC and MVT

The main difference between MVT and MVC lies on how each section of the design is named and the task that each section has; the figure below illustrates which section of MVT is paired with MVC.

Figure 5 - MVT System Architecture



In MVT these are the tasks of each section:

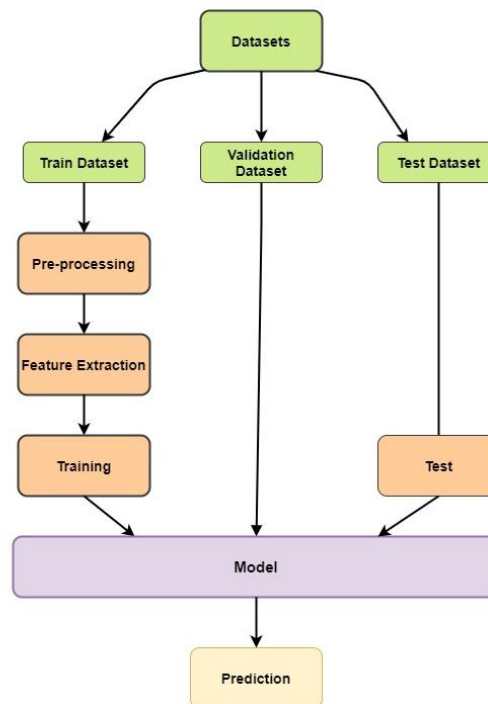
- **Model:** This section is meant for the database of the application.
- **View:** Similarly to the Controller of MVC, the View is the middleman between the Template and the Model. But in addition, it is also the place where the logical operations processes occur.
- **Template:** This just like the View of MVC, is the place that has to do with anything that has to do with graphical for the user.

5.2.2. Machine Learning Architecture

The architecture for the machine learning project will be different than the conventional way as to how other developers implement it. Main reason being that the project has set a requirement for other individuals to be able to use the project to test with other variables.

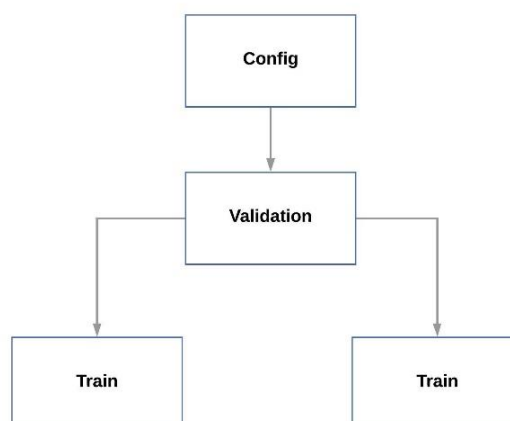
Conventional Machine Learning applications for this type of problem have the following architecture:

Figure 6 - Typical Machine Learning System Architecture



Whereas this projects main high-level difference lies that the Machine Learning training and testing will be done separately. This will deliver for external users to try out different models and datasets, and so satisfy the due requirement.

Figure 7 - Project's Machine Learning System Architecture



5.3. Design, Implementation and Testing

This subsection is the culmination effort of all the prior topics that have been discussed up to this point. As stated, the project has been developed in Front-end, Back-end and Machine Learning stages. Each stage was developed using SCRUM approach, which consists of five steps, for greater detail about each step refer to *Chapter 4: Methodology*.

5.3.1. Stage 1: Front-end

This stage focuses on the front-end of the application, that being the User Interface (UI) and User Experience (UX) of the website application.

Sprint Backlog

These are the functional and non-functional requirements that were split in Chapter 4 for the Front-end Stage.

Table 9 - Requirements for Front-end

| Stage 1: Front-end Requirements | | |
|---------------------------------|----------|---|
| ID | Priority | Requirement Description |
| FR-07 | Should | There system will announce to the user when the intelligent system has come to a decision with the given URLs |
| FR-11 | Should | The website application should be able for other users to then extend upon the current project easily |
| NFR-01 | Must | The website will have a responsive web design for desktop users |
| NFR-02 | Must | The websites functionality shall fully work on Google Chrome browser engine. |
| NFR-03 | Must | The project should attempt to follow the most up-to-date technologies and practices. |
| NFR-04 | Should | The website should work on the most popular browser engines |

| | | |
|---------------|---------------|--|
| NFR-05 | Should | The website should follow recent styling trend |
| NFR-06 | Should | The website should process the data for the Machine Learning dynamically |
| NFR-08 | Should | Operations should be carried out in a time frame that does not cause the user to believe that the system has become unresponsive |

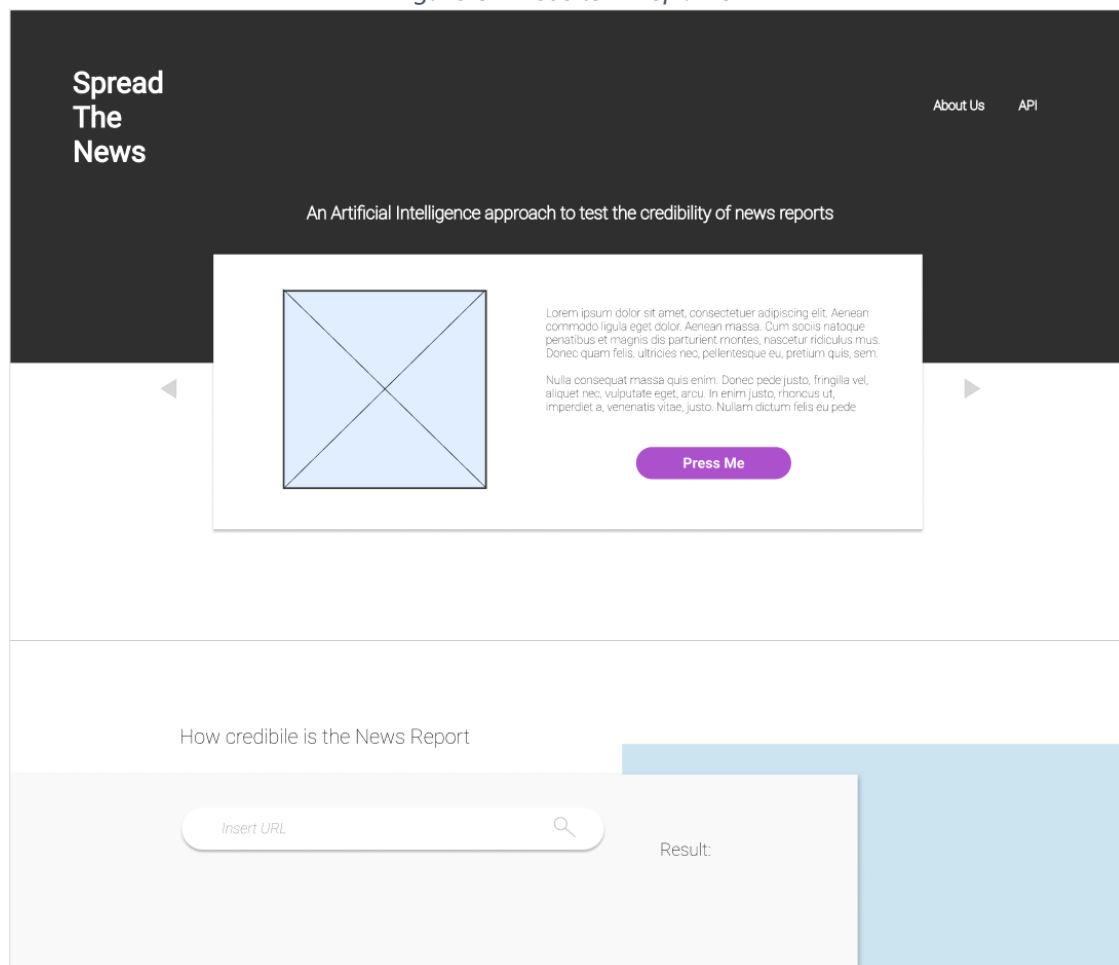
Planning & Design

The sub-section covers the planning that was undertaken to be able to develop the User Interface of the website application

Wireframe

Wireframe is a sketch/mock-up of the website User Interface. This planning technique allowed for many of the requirements to be thought of and to be able to have an idea as to how to solve them each. After several sketches, the image below shows the final sketch.

Figure 8 - Website Wireframe



Implementation

Development Tools

The tools that were used to be able to develop the front-end stage of the application includes:

HTML5, CSS AND JAVASCRIPT

The tools of HTML5, CSS and JavaScript are traditional tools for Web Development. Django allows the usage of this tools along with the framework. HTML5 was used to generate most of the static content within the webpage. CSS was used to style the HTML structure and content. JavaScript was used to create dynamic functionality for the website, such as us Asynchronous JavaScript and XML (AJAX) technique to be able to process data in the background and allowing the user to freely scroll around the page, overall delivering a more satisfying user Experience, covered more in the back-end stage.

DJANGO FRONT-END FEATURES

Django Features such as:

- Django Template Language (DTL)
- Template Inheritance
- Template Snippets

The tools provided by Django promotes highly loose coupling and Don't Repeat Yourself (DRY) design. Which then allows the program to be scalable, maintainable, and more readable for further development.

The next subsection goes further in depth as to how these Django features works and how they were used within the program. Keep in mind that for now the back-end needs to be considered as a black-box, therefore, must be assumed that the information being exchanged between front-end and back-end are correct.

Django Template Language

Once the data has successfully been sent to the Template from the View. It is then possible to access that data with the use of Django's Template Language (DTL), which is triggered by using different combinations of curly braces and symbols, such as “{{ }}”, “{% %}”, and more. Each of these symbols have different representations. Will be explaining key ones as they appear in the examples.

For instance, a common piece of data that is fed to the templates is title variable and this goes within the “title” HTML tags. What DTL does is complete that sent data in the tags, this is seen in [Figure 11, line 43](#).

Figure 9 - Part of Home HTML File

```
1  {% extends "website/base.html" %}
2
3  {% block title %} {{ title }} {% endblock title%}
4
5
6  {% block header-content %}
7  <div class="page-intro">
8
9  </div>
10 {% endblock header-content%}
11
12
13 {% block main-content %}
14
15 <!-- Detector -->
16 <div class="detector-section">
17     <h1 class="search-bar-heading h-tf">Automatic News Checker</h1>
18     <div class="detector">
19         {% include 'website/snippets/detector.html' %}
20     </div>
21     <div id="detector-response">
22     </div>
23 </div>
```

In [Figure 9, line 3](#), this line contains:

- A **Django Variable**, which is surrounded by “{{ }}”
- A pair of **Django Block Tags**, which is surrounded by “{% %}”.
 - o concept that will be explained in the upcoming Django features sections.

There are two important things to take from this line:

1. The title data is being called within the Django Variable

2. The entirety of line 3 is using DTL. This is language is what helps communication between Template with the View. This emphasizes the use of the MVT architectural design

Template Inheritance

This is one of the most powerful features about Django and why it is very appealing for many people to use Django. This feature allows massive website to be easier to program and maintain. The reader may have observed that when going through webpages, there always seems to be a trend in design, layout, and style. Template inheritance is what helps accomplish this by using the similar concept of inheritance in Object-Oriented programming.

That being said, there is a template that is typically names **base.html** that will act as the fundamental all or a group of templates, below is the base template for this project:

Figure 10 - Base Template (base.html)

```

1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0, viewport-fit: cover">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8
9      <link rel="stylesheet" type="text/css" href="{% static 'website/css/main.css' %}">
10
11      <!-- External Styling -->
12      <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
13
14
15      <title> {% block title %} TITLE HERE {% endblock title %} </title>
16  </head>
17  <body>
18      <div class="grid-wrapper">
19          <header class="header-container">
20
21              {% include 'website/snippets/navbar.html' %}
22
23
24              {% block header-content %}{% endblock header-content %}
25
26          </header>
27          <main class="main-container">
28              {% block main-content %} {% endblock main-content %}
29          </main>
30          <footer class="footer-container">
31              <p>FOOTER</p>
32          </footer>
33      </div>
34      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
35      <script src="{% static 'website/js/main.js' %}"></script>
36  </body>
37  </html>

```

From this file a series of other templates with different contents can be generated easier. This is achieved through the use of Django Block Tags. This works by having some tags as shown in [Figure 12, line 24](#) which states that any child template/inheriting template when using this tag will put its content within this area. For instance, the following HTML Template is the one for home page and look careful at [lines 13 and 42](#). As seen this is the same blocks used in [figure 12](#). Basically, the content of the base template is overwritten by what is in home page HTML.

Figure 11 - Example Block tag

```

1  {% extends "website/base.html" %}
2
3  {% block title %} {{ title }} {% endblock title%}
4
5
6  {% block header-content %}
7  <div class="page-intro">
8
9  </div>
10 {% endblock header-content%}
11
12
13 {% block main-content %}
14
15 <!-- Detector -->
16 <div class="detector-section">
17     <h1 class="search-bar-heading h-tf">Automatic News Checker</h1>
18     <div class="detector">
19         {% include 'website/snippets/detector.html' %}
20     </div>
21     <div id="detector-response">
22     </div>
23 </div>
24
25
26 <!-- News Feed -->
27 <!--<div class="news-section">-->
28 <!--     <div class="today-news-section">-->
29 <!--         <h1 class="news-title h-tf">Today's Stories</h1>-->
30 <!--         <div class="newsfeed">-->
31 <!--             {% include 'website/snippets/newsfeed.html' %}-->
32 <!--         </div>-->
33 <!--     </div>-->
34 <!--     <div class="search-new-section">-->
35 <!--         <h1 class="news-title h-tf">Search News</h1>-->
36 <!--         <div class="newsfeed">-->
37
38 <!--             </div>-->
39 <!--     </div>-->
40 <!--</div>-->
41
42 {% endblock main-content %}

```

Additionally, this Django feature also helps with the implementation of styling with CSS. This is because all the pages have been strategically designed so that the HTML structure is similar. This allows to re-use most of the CSS selectors for different pages, only in cases when there is unique HTML does the CSS have to be different and more specific. But when it comes for instance with **responsive web design**, which has been achieved using the combination of Grid Layout and Flexbox Layout, then the CSS is the same for all of them. Overall, once again and maximising reusability, maintenance, and organisation.

Django Snippets

What this feature does is being able to create Template snippets and integrate them anywhere within the designated Template File. For instance, in the project the section that is meant for the user to input the URL and where the Machine Learning responds was turned into a snippet. Meaning that in home page HTML that area is not explicitly there, but more as created a link to the file to extract the HTML from. The two main advantages that this brings is:

- Whenever another developer would want to specifically work on the Machine Learning part of the website then they will know exactly where to find it and how much of the HTML it covers. Instead of perhaps accidentally mixing some other areas of the webpage.
- Another benefit that this feature has is how the part that is snipped can be integrated in any area that the user would desire. Which means for instance if I had also would want to integrate the Machine Learning part into another webpage, instead of having to copy paste the entire HTML, would just had to direct and link the snippet that I would want to be present.

Here is the snippet created for the Machine Learning part of the website. With **Figure 14** showing the easiness to integrate the snippet onto the Template.

Figure 12 - Template Machine Learning Snippet

```
1
2 <form action="/input/" method="POST" id="detector-form">
3
4     {% csrf_token %}
5
6     <div class="search-bar">
7         {{ form.user_input }}
8         <button type="submit" id="detector-submit" aria-label="Search"><i class="material-icons">search</i></button>
9     </div>
10 </form>
```

Following figure shows the integration of this component into the Template that is displayed to the user.

Figure 13 - Where Machine Learning Snippet is added on Home Page Template

```
15 <!-- Detector -->
16 <div class="detector-section">
17     <h1 class="search-bar-heading h-tf">Automatic News Checker</h1>
18     <div class="detector">
19         {% include 'website/snippets/detector.html' %}
20     </div>
21     <div id="detector-response">
22     </div>
23 </div>
```

This concept helps readability, maintenance, and scalability for the website. But most importantly, this concept is to compartmentalise key areas of Templates, in other words makes emphasis on the DRY concept.

Testing

This section will cover the testing performed to test the User Interface of the website. The way to test it was by using the requirements defined and categorised for this stage and to determine if the page does what was set to complete.

Testing Requirement(s): FR-11

| ID | Priority | Requirement Description |
|--------|----------|--|
| FR-07 | Should | There system will announce to the user when the intelligent system has come to a decision with the given URLs |
| FR-11 | Should | The website application should be able for other users to then extend upon the current project easily |
| NFR-01 | Must | The website will have a responsive web design for desktop users |
| NFR-02 | Must | The websites functionality shall fully work on Google Chrome browser engine. |
| NFR-03 | Must | The project should attempt to follow the most up-to-date technologies and practices. |
| NFR-04 | Should | The website should work on the most popular browser engines |
| NFR-05 | Should | The website should follow recent styling trend |
| NFR-06 | Should | The website should process the data for the Machine Learning dynamically |
| NFR-08 | Should | Operations should be carried out in a time frame that does not cause the user to believe that the system has become unresponsive |

Review

This stage covered the extensively as to how the Front-end was developed, as well as the Django concepts that were needed to be able to develop the application. It is believed that the techniques used throughout the entire front-end were ones that

5.3.2 Stage 2: Back-End

This stage focus focuses on the back-end of the application. In other words, the functional part of the website and how the data provided by the user is processed.

Sprint Backlog

These are the functional and non-functional requirements that were split in Chapter 4 for the Back-end Stage.

Table 10 - Requirements for Back-end

| Stage 2: Back-end Requirements | | |
|--------------------------------|----------|---|
| ID | Priority | Requirement Description |
| FR-02 | Must | The system will only accept English written newspapers |
| FR-03 | Must | The only accepted input from the user should be URLs, anything else will be rejected |
| FR-04 | Must | The system has to be able to extract the content from the news report |
| FR-05 | Must | The system should be able to determine if the given URL comes from a news source |
| FR-08 | Should | The system should be able to process more than one URL at a time |
| FR-09 | Should | The systems output will provide a reason as to how it came to its output decision |
| FR-10 | Should | The website application should be able for other users to then extend upon the current project easily |
| FR-12 | Should | The machine learning within the website should be easily exchangeable if needed |
| NFR-03 | Must | The project should attempt to follow the most up-to-date technologies and practices. |
| NFR-06 | Should | The website should process the data for the Machine Learning dynamically |

Juan Camilo Rodriguez

16020551

Spread the News

UFCFXK-30-3

Planning & Design

Implementation

The website application does not have much emphasis when it comes to the Baack-end. Regardless there are some key implementations that are still worth noting about.

Development Tools

The tools that were used to be able to develop the back-end stage of the application includes:

VIEWS

Views are a key component in Django, they are where the logical operations occur. It is also the place that links the Model with the Templates. Therefore, this section will go through as to how many of the operations in the front-end were able to occur. Before going through that it is important to have a better grasp as to who Django workflow is and how Views play a part in it.

Django uses a built-in feature known as **URL Mapping**. When a user enters a URL then Django looks for the URL that matches said input. With each URL there is designated action to it, generally it will link to a View that will then be called.

Figure 1 demonstrates the URL path for the entire project. However, it is a slightly different to what is mentioned above, but overall works similar.

Figure 14 - Project URL Path

```
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('', include("website.urls")),
22 ]
```

Explaining the URL paths for the project:

Table 11 – Understanding Project URL Paths

| URL path: Admin | |
|-------------------------|---|
| How to access: | Localhost:8000/admin (This is if using local machine to run the server with port 8000) |
| Understanding the path: | This is an inbuilt URL that is provided by Django. Which just contains features for the admin. |
| URL path: blank | |
| How to access: | Localhost:8000 (This is if using local machine to run the server with port 8000) |
| Understanding the path: | This blank String URL is for the website application. The keyword “include” essentially extends the URL path to another URL file with more paths. |

In Figure 14, line 21 makes a includes another URL file path. Those URLs are shown below in Figure 15, and these URL paths are the ones that the project uses.

Figure 15 - Website URL Paths

```

1  from django.urls import path
2
3  # Views
4  from . import views
5
6  urlpatterns = [
7      # Home URL(s)
8      path('', views.home_page, name="Spread the News"),
9      path('input/', views.url_prediction),
10
11     # About URL(s)
12     path('about/', views.about_page),
13
14     # Contact Us URL(s)
15     path('contact/', views.contact_page),
16
17     # API URL(s)
18     path('api/', views.api_page),
19
20 ]

```

All of these paths have similar principles in functionality. Here is how they work and the purpose for each:

Table 12 – Understanding Website URL Paths

| URL path: <i>blank</i> | |
|--------------------------------|---|
| How to access: | Localhost:8000 (This is if using local machine to run the server with port 8000) |
| Understanding the path: | Once again this is intentionally left blank, it is if there was a domain then it would just be the name given instead of localhost. This directs to the homepage view, which later directs to the homepage template. |
| URL path: <i>input</i> | |
| How to access: | Cannot access this link |
| Understanding the path: | This URL path is for machine learning section in the home page. When the user submits input into the system then JavaScript is used to call the URL path, that then calls the View with the steps to validate the input and if valid then predict what the news report classifies as. |

Once called the View, then it will perform some tasks, some of which will end up in the Front-end. [Figure 3](#), demonstrates the process that the Home page view looks like and its operations

Figure 16 - Example View

```

21  """ # Home View(s) # """
22
23  # Call to retrieve news and display on screen
24  def home_page(request):
25      ### FORM
26      # Unbound Form
27      form = DetectorForm()
28
29      ### API
30      # API CALL
31      newsapi = NewsApiClient(api_key='9dff3b262af247178cba410205157829')
32
33      # Type of News requested from API
34      top_headlines = newsapi.get_top_headlines(language='en')
35
36
37      ### Prep for Response
38      # Designated Template
39      template_sent = "website/home.html"
40
41      # Data sent to Template
42      context = {
43          "title": "Spread the News",
44          "form": form,
45          "top_headlines": top_headlines,
46      }
47
48      return render(request, template_sent, context)

```

As seen in [Figure 16](#), this is the home page view and it is performing a number of statements which give results needed for the Template. At line 39, it is specified as to what is the Template destination. Lines [42-46](#) the dictionary named “context” will contain the data that is needed to be sent to the template, in this case to the HTML file. Line 48 is when the data is sent to the home page template.

Now that some fundamental Django concepts and workflow have been covered, it will make the following and future concepts much easier to understand.

ASYNCHRONOUS JSON AND XML (AJAX) AND VIEWS

Asynchronous JSON And XML (AJAX) is a technology in web development that allows a web page to request data asynchronously. The purpose of asynchronous programming in web development is to avoid having to go through a whole web page request to be able to obtain something in return (Ford, J.L.E, 2009). This technology has been one that many high earning companies have invested, such as Google, into their webpages as it results to deliver a more satisfying website experience (Woychowsky, E., 2007).

Although AJAX is regarded more of a front-end technology over a back-end. It is included within this stage as the particular AJAX and View that is used throughout the project is the one link with the Machine learning part of the website. The reason as to why thi

Testing

This section covers the testing for the requirements stated for Back-end in the Requirements Backlog section of this stage

| | |
|----------|-----------------------------|
| Test 001 | Detector - Only Accepts URL |
|----------|-----------------------------|

| | |
|-------------------|--------------|
| Latest Test Date: | 04 June 2020 |
|-------------------|--------------|

| | |
|-----------|----------------------------|
| Objective | Form must only accept URLs |
|-----------|----------------------------|

| | |
|----------------|-------|
| Requirement(s) | FR-11 |
|----------------|-------|

Covered

Expected Result:

| | |
|--------------|------|
| Test Result: | PASS |
|--------------|------|

Evidence:

Juan Camilo Rodriguez

16020551

Spread the News

UFCFXK-30-3

Review

5.3.3 Stage 3: Machine Learning

In this stage the focus is the development of the Machine Learning application and explaining as to what route was taken and explaining as to how this path works in detecting fake news.

Sprint Backlog

These are the functional and non-functional requirements that were split in Chapter 4 for the Machine Learning Stage.

Table 13 - Requirements for Machine Learning

| Stage 3: Machine Learning Requirements | | |
|--|----------|---|
| ID | Priority | Requirement Description |
| FR-01 | Must | The intelligent system will use Machine Learning techniques only to be able to determine the legitimacy of news reports |
| FR-06 | Must | The Machine Learning application after thorough testing should be at least achieving 75% |
| FR-11 | Should | The machine learning application should be ready to use for other users to test with other datasets and/or models |
| NFR-03 | Must | The project should attempt to follow the most up-to-date technologies and practices. |
| NFR-07 | Should | The website should be able to classify in less than a minute per news report |

Planning & Design

It has been established that it is desired that the Training and testing of the model will be separate. However, this section will cover the other aspects in which that application will be developed in order function and complete the requirements

Sequence Diagram

The following diagram will illustrate how objects within the Machine Learning Program interact

Implementation

Testing

Review

Software development, implementation and testing

SUPERVISOR

Section is incomplete with little evidence of implementation. Most parts of this section seem to contain a collection of notes for the author.

The section appears to contain notes on sources for implementation, is missing diagrams and relevant code snippets with explanations of how this was implemented

The discussion regarding stemming and what it is may have been better suited in the design section.

No test plan devised with a cross reference to the requirements table and no testing performed.

No source code provided.

MARKER

Little or no demonstration of programming skills. Demonstrates little or no insight into the technologies employed. Little or no software testing; limited or no reflection on the tests' results.

Much of this section appears to be a collection of notes rather than a report

Chapter 6:

PROJECT EVALUATION

Project evaluation

SUPERVISOR

Section incomplete. No evaluation, conclusions or reflection on the project

You must critically evaluate your results in the light of your tests, describing its strengths and weaknesses.

Ideas for improving the project can be carried over into a Future Work section. Remember: no project is perfect, and even a project that has failed to deliver what was intended can achieve a good pass mark, if it is clear that you have learned from the mistakes and difficulties.

Reflection should focus on the skills and lessons that can be carried over into future activities and the impact of what you have done on the assumptions, concepts and ideas that you used to make decisions about your work.

Conclusions should be a summary of the aims of project and a restatement of its main results, i.e. what has been learnt and what it has achieved.

An effective set of conclusions should not introduce new material. Instead it should briefly draw out, summarise, combine and reiterate the main points that have been made in the body of the project report and present opinions based on them.

MARKER

Little or no evaluation of the project; demonstrates little or no reflection on the aspects of the project (research, requirements, methodology, design etc.); little or no discussion of the limitations; provides little or no suggestions for further work;

Little or no use of the feedback provided in the PiP and/or during meetings with the supervisor.

REFERENCES

Achimugu, P., Selamat, A., Ibrahim, R. and Mahrin, M.N. (2014) A Systematic Literature Review of Software Requirements Prioritization Research. *Information and Software Technology* [online]. 56 (6), pp. 568-585. [Accessed 22 June 2020].

Beck, K., Beedle, M., Bennekum, A.V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, S., Sutherland, K., and Thomas, D. (2001) Manifesto for agile software development. Available from: <https://agilemanifesto.org/>. [Accessed 15 July 2020]

Ford, J.L.E (2009) *Ajax Programming For the Absolute Beginner*. Canada: Cengage Learning.

Hudaib, A., Qasem, M.A.H., Masadeh, R.M.T. and Alzaqebah, A.I. (2018) Requirements Prioritisation Techniques Comparison. *Modern Applied Science* [online]. 12 (2), pp. 62-80. [Accessed 23 June 2020].

Krasner, G.E. and Pope, S.T. (2014) A Description of the Model-view-controller Userinterface Paradigm in the Smalltalk-80 System. *Joop - Journal of Object-oriented Programming* [online]. [Accessed 18 July 2020].

Kuhn, J. (2009) Decrypting the MoSCoW Analysis. *The Workable, Practical Guide to Do It Yourself* [online]. 5 (44), pp. 1-2. [Accessed 22 June 2020].

Lausen, S. (2002) *Software Requirements Styles and Techniques*. Great Britain: Biddles Ltd.

Leau, Y.B., Loo, W.K., Tham, W.Y. and Tan, S.F. (2012) Software Development Life Cycle Agile Vs Traditional Approaches. *2012 International Conference on Information and Network Technology (Icint 2012)* [online]. 37 (1), pp. 162-167. [Accessed 15 July 2020].

Mishra, A. and Dubey, D. (2013) A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios. *International Journal of Advance Research in Computer Science and Management Studies* [online]. 1 (5), pp. 64-69. [Accessed 15 July 2020]

Schwaber, K. and Sutherland, J. (2017) *The Definitive Guide to Scrum: The Rules of the Game*. *The Scrum Guide* [online]., pp. 1-19. [Accessed 17 July 2020].

Verheyen, G. (2013) *Scrum - A Pocket Guide - A Smart Travel Companion*. Premedia, Amersfoort - NL: Van Haren Publishing.

Vestola, M. (2010) A Comparison of Nine Basic Techniques For Requirements Prioritization. Helsinki University of Technology [online]., pp. 1-8. [Accessed 04 June 2020].

Woychowsky, E. (2007) Ajax - Creating Web Pages with Asynchronous Javascript and Xml. : Pearson Education, Inc.

APPENDICES

Appendix A: Project Tools

Juan Camilo Rodriguez

16020551

Spread the News

UFCFXK-30-3