Software Requirements Specification for Fake News Detector

Lestyn Brooks

4/30/2021

Table of Contents

Revision History

Date of Change	Reason for Change	Version
12/14/2020	Initial Specification	1
4/15/2021	Remove info about Android capabilities (time)	1.1
4/30/2021	Cleaning up document for submission	1.2

1. Introduction

1.1 Purpose

The purpose of this document is to detail the project plan for the development of the Fake News Detector application.

1.2 Scope

This document specifies requirements for an application for determining whether a news article found online is providing real, honest and truthful information.

The application will allow users to:

- Submit an article link to be inspected by the program to determine its legitimacy
- View articles that have been previously checked by the application and whether they were determined to be real or potentially fake
- Be informed about whether an article comes from a satirical source

1.3 Product Perspective

1.3.1 System Interfaces

The application will run on Windows devices. I had initially planned to make it possible to run on Android devices as well, but due to time constraints, I had to cut that feature for now. I also chose not to make it iOS compatible as a paid developer license is required to deploy to iOS, and that is not something I can afford right now.

1.3.2 User Interfaces

The application GUI provides menus, buttons, containers, and grids, allowing for easy control by using either a keyboard and mouse or a touch screen.

1.3.3 Software Interfaces

The application allows the import of a website URL.

The application allows the export of article information to an MS Excel sheet via CSV data format.

1.4 Product Functions

Product Functions are summarized in Section 2 of this document.

1.5 User Classes and Characteristics

End User – The End User is the person who wishes to find out if the news in an article is legitimate or if it's potentially fake. The End User will be able to be informed about article legitimacy by means of submitting article links for review, viewing previously checked articles and their results, and being informed the reasoning behind an article being labelled as potentially fake.

1.6 Acronyms and Abbreviations

FND: Fake News Detector

CSV: Comma Separated Values

GUI: Graphical User Interface

URL: Uniform Resource Locator (For the purpose of this project, it refers to a web resource)

ID: Identifier

HTML: Hypertext Markup Language

iOS: iPhone Operating System

SRS: Software Requirements Specification

2. Requirements

2.1 Functions

2.1.1 Submit Article for Review

[FND-SRS-1] The application shall allow End Users to submit a link to an article so that it can determine the article's legitimacy using an algorithm.

2.1.2 Collect Article Information from URL

[FND-SRS-2] The application shall detect the article's title and contents so that it is easy for both the End User and the machine learning algorithm to read and process.

[FND-SRS-3] The application shall pass the article's title and contents into a CSV file to be read by the machine learning algorithm for review.

2.1.3 Determine an Article's Legitimacy

[FND-SRS-4] The application shall use the news dataset from Andrew Thompson (www.kaggle.com/snapcrack/all-the-news) as a test dataset as it includes thousands of articles spanning from 2015 to 2017.

[FND-SRS-5] The application shall use a Passive Aggressive Classifier so that the application makes updates to correct itself in the event of an article being mislabeled.

2.1.4 Reject Invalid or Duplicate Articles

[FND-SRS-6] The application shall inform the End User if the article they submitted comes from a satirical source, such as The Onion or Babylon Bee, so that the End User knows that those articles are intentionally fake.

[FND-SRS-7] In the event that a URL submitted by the End User is not a news article, the application shall display an error so that the algorithm isn't thrown off by things that aren't articles.

[FND-SRS-8] In the event that a URL submitted by the End User is not found (i.e. it returns a 404 error), the application shall display an error so that the algorithm isn't thrown off by a blank page.

[FND-SRS-9] In the event that a URL submitted by the End User has already been checked, the application shall let the user know that so the algorithm doesn't have duplicate entries to learn from.

2.1.5 Review Previously Checked Articles

[FND-SRS-10] The application shall allow End Users to see what articles have been previously fact checked and what the results were so that the End User can have as much information as possible available to them.

2.1.6 About Page

[FND-SRS-11] The application's GUI shall include a section with information about how the software was written so that the End User can see that the application is unbiased, in part because it is not human.

2.2 Usability Requirements

[FND-SRS-12] The application shall have an average uptime of 99% so that End Users can fact check news articles at their convenience, regardless of the time of day.

2.3 Performance Requirements

[FND-SRS-13] The application shall display the GUI within 10s of it being loaded so that the End User can use it quickly.

2.4 Logical Database Requirements

[FND-SRS-14] The application shall contain an SQL Database so that every article that is fact checked can be stored within the application for the End User to see.

2.5 Software System Attributes

[FND-SRS-15] The application shall run on any Windows device.

[FND-SRS-16] The application shall be written in Python as it has many deep learning frameworks and libraries available to work with and make the development process relatively simple.