Verific8 Requirements Document

Date(s): *Wednesday, November 11, 2020*

Prepared by: *Maxwell Trdina, Kenzie Eggert, Anthony Oddo, Gia Thurton, Serge Tchabo*

# Introduction

This document contains the system requirements for Verific8. These requirements have been derived from several sources, including Pew and Stanford University.

## Types of Reader (Max)

**Upper Management and Executives:** Technical skills and implementation details are not critical for understanding Verific8 or the core problems it aims to solve. You will be best served by simply reading the contents found in the General Description.

**General Stakeholders:** Every stakeholder has a different set of needs. As there are a large quantity and variety of stakeholders, each one will need to identify and read the sections most critical to their needs and skill set. The best places to start are the General Description and the User Interface Requirements. This will give you an idea of what the app does and how it will appear to users.

**Project Managers/Scrum Masters:** Reading through the complete document is recommended. However, the most critical areas of focus are the General Description, Product Functions, User Characteristics, Constraints, and the entirety of the Specific Requirements section.

**Software Developers:** This entails anyone responsible for the coding and database implementation of the app, as well as those responsible for designing the software for analyzing news articles. Your primary areas of focus should be the Product Functions, Functional (User) Requirements, Availability and Performance Requirements, and System and Integration Requirements sections.

**UI Designers/UX Engineers:** This includes anyone who contributes to the overall visual representation of the app or anyone who has an impact on how the user interacts with the app. You will be primarily interested in reading the Functional (User) Requirements and the User Interface Requirements sections.

**Professional Fact Checkers:** Your role is primarily described in the Product Functions and Functional (User) Requirements sections. You may also find that reading the General Description will be beneficial.

**End Users:** In order to have a functional understanding of Verific8, what it does, and how you as a user will benefit from using it, we recommend that you read the General Description, Product Functions, and User Interface Requirements sections.

## Technical Background Required (Max)

This document has been written in a way that does not require a particularly significant technical background in order for readers to understand the various sections or the goals of the app. In fact, any user that has the ability to operate the app should be able to understand anything in this document. The only areas of note that use any industry jargon or acronyms are the Assumptions and Dependencies, the Constraints, and the System and Integration Requirements. For these areas, you can find an explanation of any acronyms in the glossary.

# General Description

This section will give the reader an overview of the project, including why it was conceived, what it will do when complete, and the types of people we expect will use it. We also list constraints that were faced during development and assumptions we made about how we would proceed.

## Why Verific8? (Kenzie and Max)

The malicious spread of false information, known more commonly as disinformation or fake news, has become a topic of rapidly increasing interest in recent years. Every day, an increasing amount of people engage with and possibly spread disinformation without even knowing it. One recent study found that as many as 86% of people using the internet are deceived by disinformation daily[[1]](#footnote-0). Nobody wants to spread false information, but when such a high rate of people are unknowingly exposed to disinformation, it is inevitable.

Our solution, Verific8, is a mobile application aimed squarely at solving this problem. With as many as 62% of Americans getting their news from online sources[[2]](#footnote-1), we know that our solution must embrace technology and keep pace in a modern, connected world. Verific8 is designed to provide people who use the internet as their primary source of news a quick and easy way to fact check the news they already encounter, as well as give them a source of reliable, highly truthful news.

Verific8 enables users to scroll with confidence, knowing that the truth is only a few taps away. We provide users with the means of fact checking any news article they may read, as well as providing them with multiple ways of viewing truthful news and a reliable, consistent method of identifying reliable news. Using our app, users will be more accurately informed than ever and have the power to break the cycle of spreading fake news.

## Product Functions (Max)

**Article Analysis:** The core of our program is the analysis of news articles. When a user provides the app with a news article they want to check, the app will use software to analyze the article and identify various features of the article, such as the author, source, and the references the author uses. One this information is extracted, we will compare it against our own database of known sources, authors, and references and their reputability maintained by our fact checking team. The trustworthiness of these data points is then quantified, along with other factors like age and number of references, to calculate a trustworthiness score.

Additionally, we can tell if an article is checked frequently as a result of our integrated search feature. If a specific article is analyzed frequently by users (ie if a particular link is searched for using our search tool at a particularly high rate), our fact checking team will additionally analyze the article manually. This will allow us to give more in depth scores and analysis for articles that many users are trying to see.

**Article Search:** Using an integrated search function, users will be able to either input articles by hyperlink or search by keyword for articles that have been analyzed previously to check whether a specific article is trustworthy or not. Users will be able to utilize a search bar where they can paste links to news articles, or search via keyword to find a specific article they’re looking for or to view articles focused on a specific topic. Once they find the article they’re looking for, users will be able to see statistics about the article, such as how many times it has been shared and on what platforms, as well as a scale of trustworthiness calculated based on a number of factors, such as the reputability of the source, author(s), and the references used in the article.

**News Feed:** Users of Verific8 will have the option of viewing a news feed that consists of new trending articles that have already been entered into our database and reported as truthful. This gives users access to a news feed consisting solely of reliable, trustworthy information about topics that are currently relevant.

**Trending page**: With this function, users will be able to view all popular and trending news articles, regardless of their trustworthiness rating. Along with each article, users will be shown an indicator bar displaying how the article has been rated. Also, if an article has been reviewed by our fact checking team, that info will also be displayed with the article. This will give users a different view of the current news landscape, with more content than would be displayed by just displaying truthful articles alone. This will enable users to still check if the content they’re reading is factual, as well as giving them a view of how much of the current trending news is factual.

## User Characteristics (Kenzie)

By design, Verific8 is designed for a wide range of users. This is because anyone who uses social media or reads news online could be a potential user for our solution. As a result, we are looking to target adults who use the internet, and especially social media, as a primary means of obtaining news to use our application.

This group of users does not need any particular technical background, training or education to use our app, allowing for an increased number of users. Being in both the midst of an Election and a pandemic, many adults fail to know the content that is accurate or not. As a result, they may find a strong sense of motivation to use Verific8 to gain a deeper understanding of disinformation and to increase their skills of identifying such fake news. Our users will be shocked to find out the high level of disinformation they encounter and possibly spread unconsciously each day.

## Constraints (Serge and Max)

Given that our solution is currently only app based, we must strive to ensure that our interface is of very high quality and our app is easy to use. Because users will have only one way of interacting with our solution, if the interface is not convenient for them to use, it is likely they will lose interest in our solution altogether. Expansion to other platforms, such as a web based interface are something we will want to expand to eventually, but for now, we must focus on ensuring that the interface we do have provides users with the absolute best user experience.

Another concern we must be aware of is how our app handles checking articles and sources from subscription based or paid sources. For many common paid sources, it is feasible for our solution to bear the cost of subscriptions for our fact checking team. However, if a user were to search an article from a paid source that we have not seen before, we would not have much in the way of a proactive solution. In these cases, we would likely just have to explain to the user that we would not be able to analyze the article at the time. If this source were attempted to be analyzed by many users, as we could tell from our own internal analytics, then we would consider paying to access the source and adding its information to our repertoire.

We must also consider the liability we entail with applying trustworthiness judgements to journalists/authors and their work. Labeling an author or their work as untrustworthy could in some extreme cases be considered libel or defamation. While it is up to our professional fact checkers to be as objective and justified as possible, we must acknowledge that this could still become a problem anyways. To resolve this, we will need to budget for legal fees as early on as possible, so that when a problem like this arises, we are not blindsided and have the capital to respond.

Finally, we must also consider our users, more specifically their privacy and their ability to use the app. Although we don’t need to collect much info about users of our app, users still can create an account for the app, and thus we need to prioritize the security of their account and any information we may collect about them. We must also consider differently abled individuals and ensure that our app is in accordance with WCAG and its various requirements. This will allow us to extend the usability of our app to many more individuals, all while ensuring that their data is secure. Because our solution is not a government owned or funded entity, we are not bound to the regulations of 508C, and our efforts to ensure a usable interface for all is voluntary[[3]](#footnote-2).

## Assumptions and Dependencies (Serge and Gia)

Verific8 requires several assumptions and dependencies for successful use. On the user end, we assume that they have a smartphone or tablet with basic knowledge of its use. We also assume that these devices have internet connection via cellular data or a WiFi connection. On the technical side, Verific8 depends on the use of SEO spider tools for website crawling as well as the use of APIs for database management. We are dependent on our professional fact checking team as well that is unbiased. Finally, on the administrative side, Verific8 is dependent on administrators with troubleshooting and feedback skills and documentation of any issues for updating. Our administrative team must also have disaster recovery skills.

# Specific Requirements

This section of the document lists specific requirements for Verific8. Requirements are divided into the following sections:

1. User requirements. These are requirements written from the point of view of end users, usually expressed in narrative form.
2. Availability and Performance requirements.
3. System and Integration requirements. These are detailed specifications describing the functions the system must be capable of doing.
4. Sustainability Requirements
5. User Interface requirements. These are requirements about the user interface, which may be expressed as a list, as a narrative, or as images of screen mock-ups.

## Functional (User) Requirements (Gia)

**Registration**: Once the app is downloaded, users are prompted to either sign up using either their email or their Facebook account. If they choose to sign up using Facebook, the feed on their homepage will synchronize with the posts shared on their Facebook feed and indicate which ones contain disinformation.

* Case: John signs up for Verific8 using his facebook account. Upon logging in his feed synchronizes and he notices that a post that his friend Kelly recently shared contains disinformation

**Sharing**: Users have the option of sharing posts that they see on our app via text message, email, twitter, or a copied link. Users that have apple products can also share these posts via airdrop. Sharing a post through these external methods invites other people to join the app in order to view them.

* Case: Sophia logs in to our app and recognizes a post that she recently talked about with her friend, Cheyenne. When she clicks on it she sees that it was actually disinformation because the article was originally posted two years ago. She decides to share the post with Cheyenne via text message.

**Search bar**: Users are able to search for articles using keywords that are relevant to the article. The search results show a list of articles containing the keywords, listed by relevancy and date posted. When the user clicks on an article, our app shows an analysis of the article’s credibility based on its source, author, and references. To discover articles, our search bar works through the processes of *crawling and indexing*. Our system will crawl through billions of article web pages, downloading them, and following their links to other web pages on a regular basis. This is how we gain access to the article’s author, source, and references. Once article web pages are discovered, they are added to the general index database to store and organize their content. The general index database stores keywords within the article’s content, the date it was posted and updated, its user engagement, and the type of content it includes.

* Case: Malcolm has a research paper to complete based on the COVID-19 pandemic. He only wants to use credible sources, so he searches ‘coronavirus’ on our app for articles related to this keyword. From the search results, he only picks out the ones that are from credible authors, sources, and references.

**Article disinformation analysis**: Users can view the disinformation analysis of each article through our app. Once the crawling and indexing process is completed for an article, our professional fact checking team will analyze the article’s index to determine if there is any disinformation detected. Our fact checking team looks not only for accurate and up-to-date content, but also possible bias based on three categories: the source, author, and references. Once this process is complete, our fact checking team provides an analysis on the article’s credibility for each category.

If, for some reason, our fact checking team has not gotten to a particular article or they aren’t sure about the article’s disinformation status, the article will be forwarded to be analyzed by our algorithm. The algorithm will then analyze the article’s index database for its source, author, and references. It will then compare the article’s data to our general index database of sources, authors, and references collected through our crawling and indexing process across all articles in order to rate each category’s credibility. Each category will be assigned a score based on the data retrieved. To make this process faster, the algorithm will implement machine learning by continuously adding to our general database to get better at predicting the status of each category.

* Case 1: Mya sees an article online titled, “Joe Biden Predicted to Win the Upcoming US Election by a Landslide,” and decides to search it up through our app to figure out if it contains disinformation. She types in the exact title and the article comes up. When she looks at the disinformation analysis she sees that our app has indicated that although the source and author are credible, the article references are untrustworthy because the poll statistics are drawn from a survey placed on a democratic website. Mya now knows that this article contains bias.
* Case 2: Rodrick finds an interesting article on the trending page titled, "California to cut power for 1 million residents,” but notices that it is pending verification from the fact checking time. When he looks at the disinformation analysis, he sees that the algorithm has scored this article as potentially trustworthy because although there is no data on its references, its sources and authors are credible.

**Statistics**: Users can view statistics regarding what social media platforms each article is being shared on. This information is retrieved through analyzing the user engagement of the article’s index.

* Case: Sarah reads an article on our app that was verified as disinformation. She is curious about which apps this article is mostly being spread on so she presses the “statistics” button. From this data, she sees that the article had the most interactions on Instagram.

**Trending Page:** Users can view articles that have the most user engagement through the trending page. As the crawling and indexing process progresses, our app picks out the top ten articles with the most user engagement and displays them on this page. The user engagement data is found in each article’s index database.

* Case: Jake notices an article on the trending page that he saw a few times before on other apps. After looking at the disinformation analysis, he learns that the article contained disinformation despite its popularity.

## Availability and Performance Requirements (Anthony)

As Verific8 is a mobile application, we expect a 99.9% uptime and a response time of two seconds or less. As the application is not using outside services, but rather a database, this uptime and response time should be easily met by our engineers. Response time can vary on the user's end due to network connection, speed, latency, etc. but that will not affect the overall speed of the application. We will initially launch in the USA and eventually move to Europe and worldwide as we scale upwards. We expect increased usage during regular active hours within the USA, typically from 9:00 AM to 9:00PM EST. We also expect an increase in user activity during government debates, hearings, and nightly news. During these times, the application will be able to handle the increased workload and will not suffer in performance or response time. It is important to maintain high performance in the application, as a downgrade in performance will cause a loss in users.

Verific8’s information database will be regularly updated and maintained, as well as backed up to our servers. Regular maintenance will be performed during down hours, between 2:00 and 3:00 AM EST. This will ensure the application will perform sufficiently with regular updates and maintenance. Maintenance during low activity hours will affect the least amount of people as possible, and ensure the application runs smoothly during active hours.

## **System and Integration Requirements** (G**ia)**

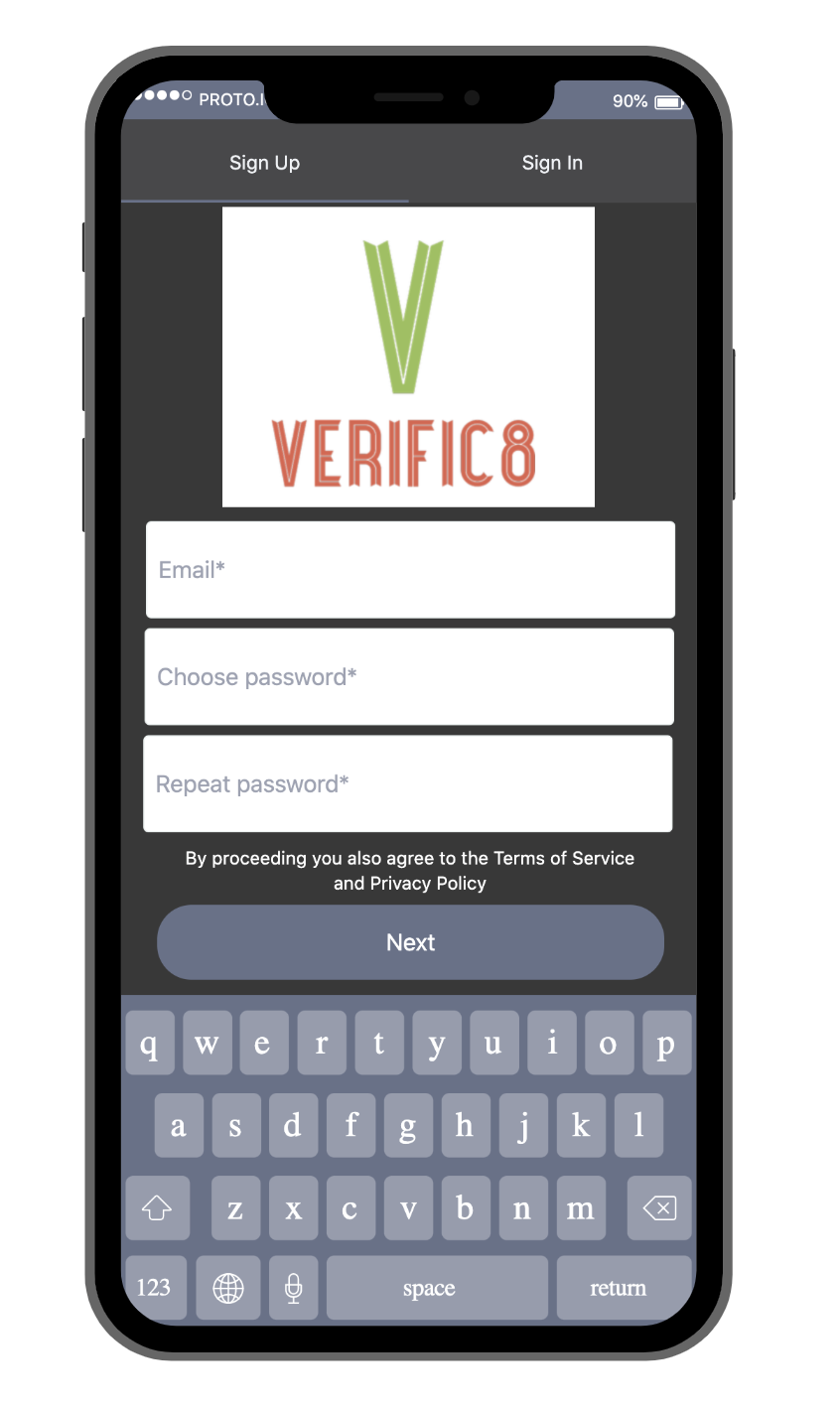
To ensure that Verific8 is compatible with most devices, it will run on iOS 12.0 and later for Apple products and Android 5.0 and later. Verific8 will be most compatible with the latest iOS 14.1 update as well as the Android 11 update. We chose this compatibility based on an Apple report that states that as of June 17, 2020, 94% of all Apple devices use iOS 12.0 and later, and a Statista report that states that 94.1% of Android devices use Android 5.0 and later. To personalize the user’s feed, the app will collect data from the articles they search and click on to determine what news categories should be displayed on the home page.

## **Sustainability Requirements** (K**en**zie)

Verific8 will initially launch in the USA in the English language, eventually scaling to Europe and eventually worldwide. Additionally, accompanying the product to its new locations will include the various languages. We plan to expect roughly 250 users for our launch, and expect as many as thirty million users after the app has scaled to cover the globe. Throughout the process, scalability is key for our product in a number of ways. After our initial launch, we plan to use agile development in order to provide updates and changes as they are produced. While we plan to expand our product, it becomes equally important to expand our maintenance to ensure users are provided with the high quality experience we strive for. As a result of this, we plan to remain up-to-date with our product by providing updates, with updates being pushed out every week. This process will ensure each user is being provided with the most current and error-free version of Verific8. Social media users are constantly being bombarded with news, and should be able to turn to Verific8 for answers at any time. News changes too quickly for our application to be unavailable or facing technical difficulties, therefore, expanding our maintenance team will support this.

## User Interface Requirements (Gia)

The simple design of our Verific8 app will make it easy to use for its users. Our app is compatible across iPhones, Androids, iPads, and Tablets so that users can easily access our database. Once the app is downloaded and installed, the user will be given the choice to either sign in (figure 1) or sign up (figure 2). On the sign in page, users have the option of signing up with Facebook which allows the app to access all of the posts on their Facebook feed and synchronize it with the app’s feed. If the user chooses not to connect their Facebook account, their feed will be a random list of recent news.

**

*Figure 1 Figure 2*

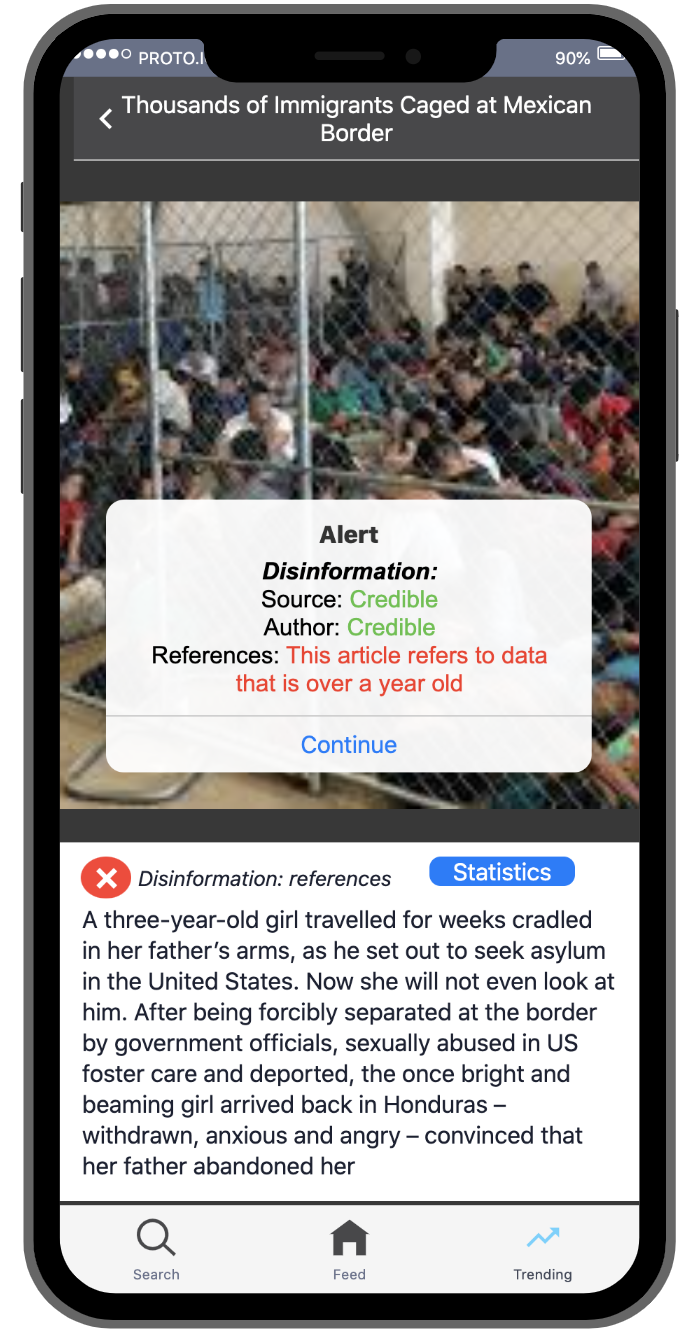
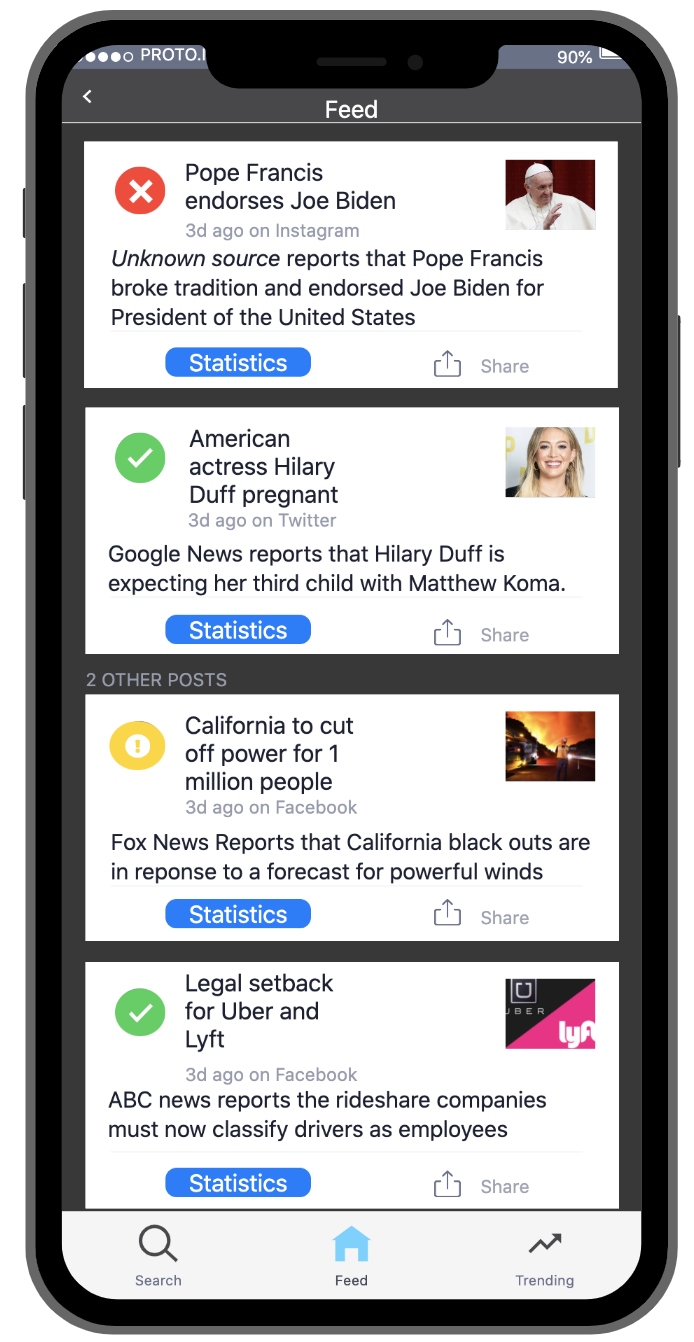
**Feed**: Upon logging into the app, users are taken to the home page, also known as the feed. This page (Figure 3) features a list of the user’s feed of posts. To determine if an article contains disinformation, our professional fact checking team analyzes it based on three categories: the article’s source, author, and references.

If a red ‘X’ is placed in the top left corner of a post, our fact checking system has determined that the post contains some form of disinformation because of a problem found in one or more of those three categories. Once one of these articles is clicked , it gives an explanation on which category our system has detected disinformation in (figure 4).

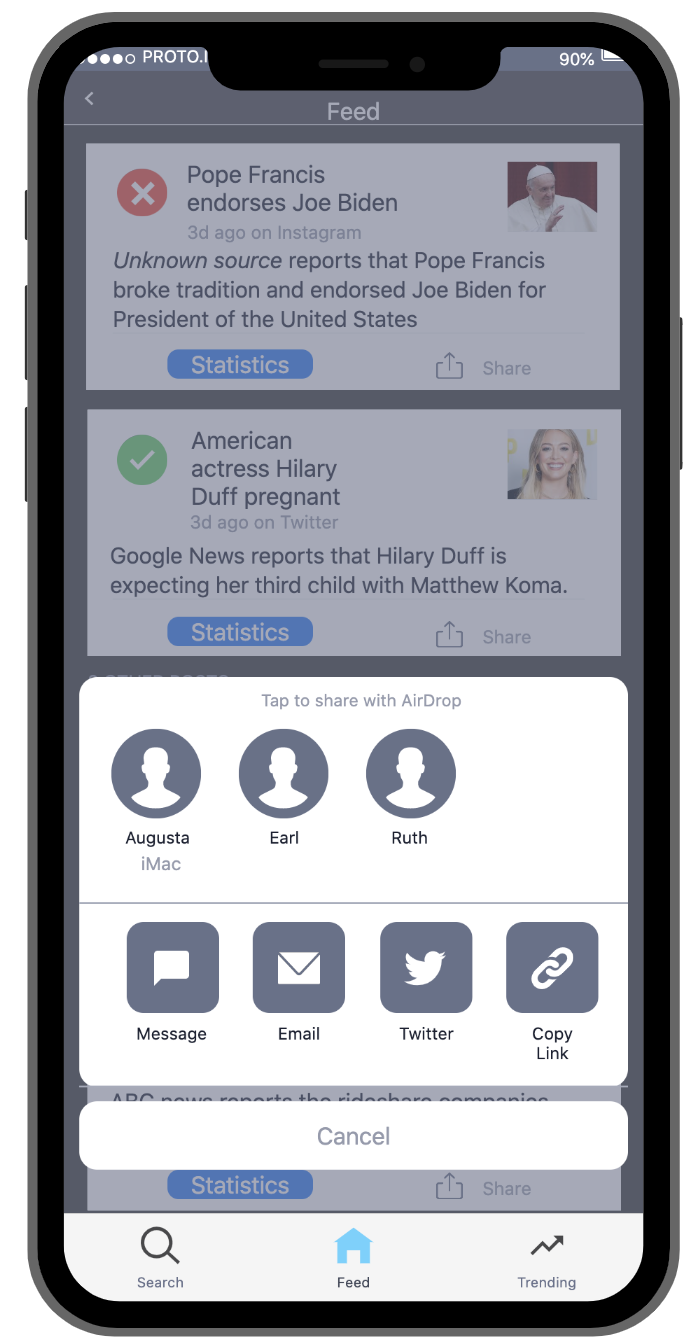
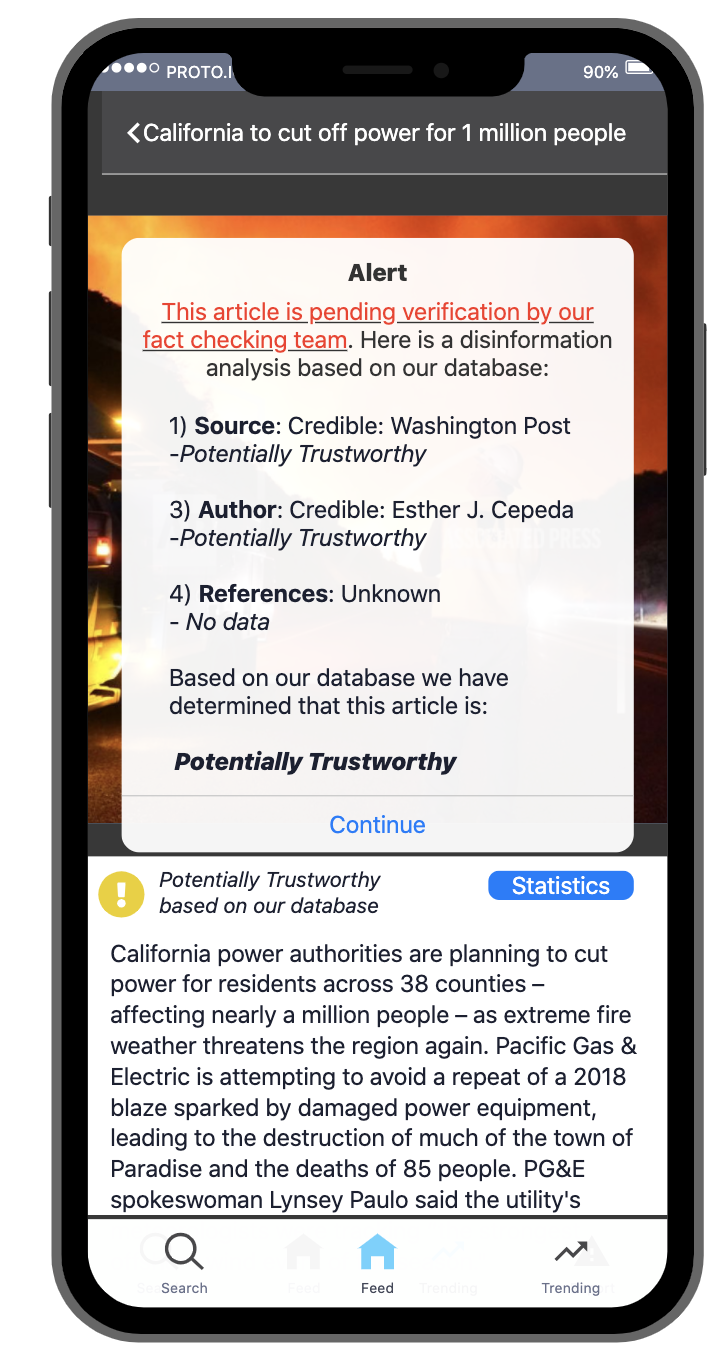
If a green check is placed in the top left, our system has verified that the article is credible across all three categories.

If a yellow exclamation point is placed, that indicates that our professional fact checking team is not sure yet of the article’s disinformation status. In this case, our system references a database containing information about news sources, authors, and references maintained by our fact checkers. From here, our software will compare the information it has gathered from the article with our database. For each piece of data (authors, sources, references) the software will assign a score for the data from one of the given options: Not Trustworthy, Potentially Untrustworthy, Potentially Trustworthy, or No Data. After all of the data has been analyzed, a score is returned that is computed based on the analyzed data (figure 5).

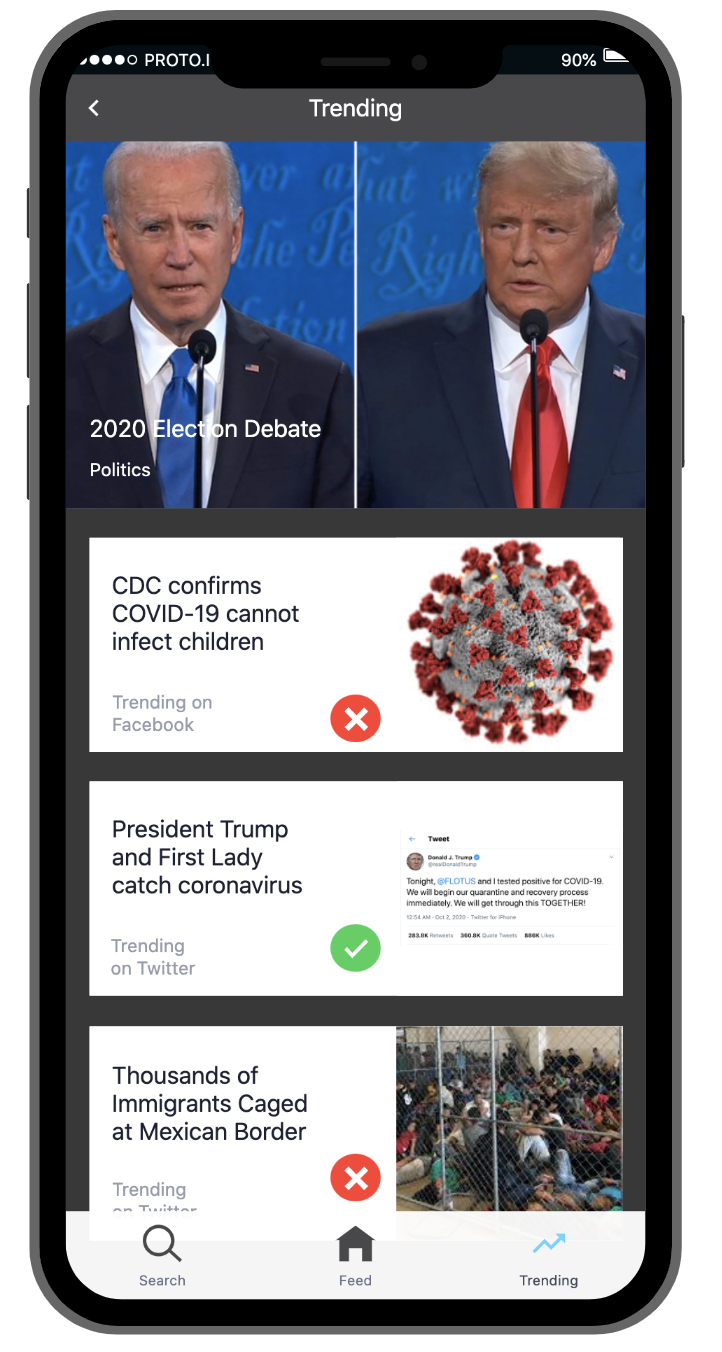
Each post lists the article title, a short excerpt of its content, when it was shared, and which social media platform it was most shared on. As shown in figure 6, when they press the ‘share’ button, users also have the option of sharing the posts that they see on our app via text message, email, twitter, or a copied URL link. For Apple products, sharing is also possible via airdrop. Users are able to navigate to other pages on the app using the tab bar on the bottom.

**

*Figure 3 Figure 4*

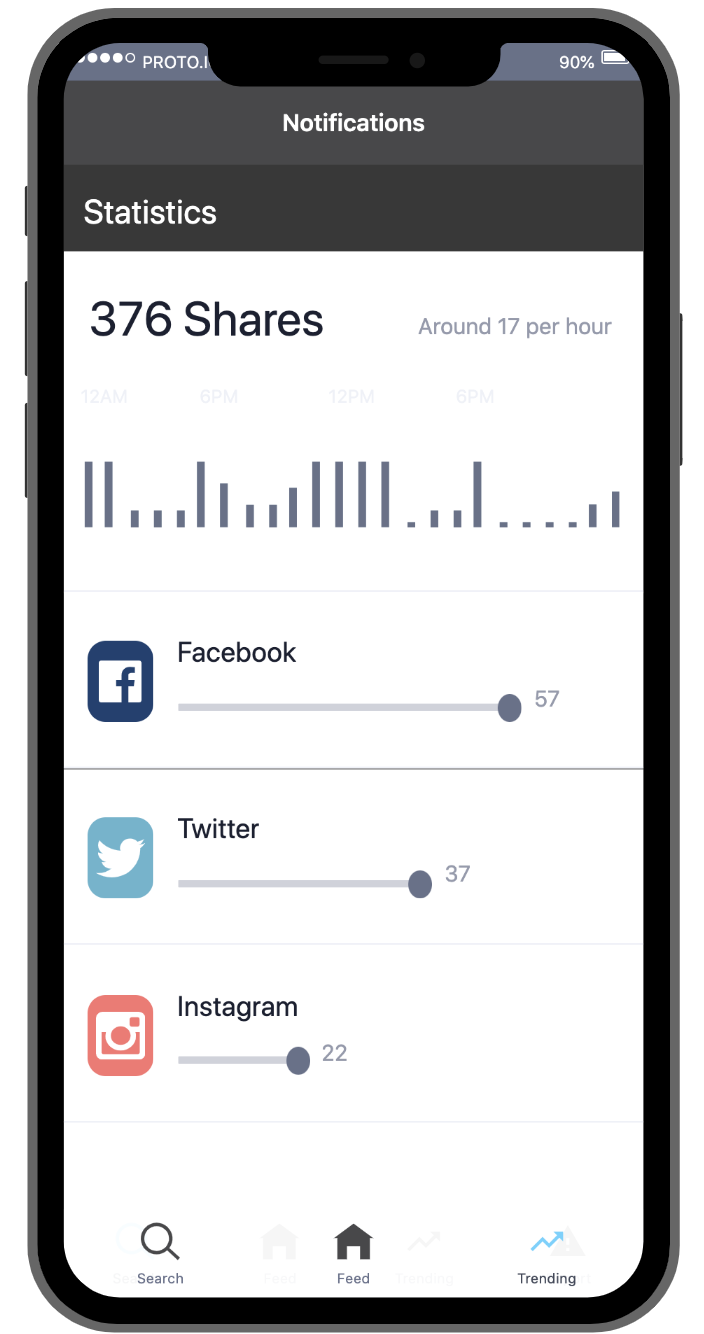
**

*Figure 5 Figure 6*

**Trending page**: By pressing the ‘trending’ tab at the bottom of the screen, users are taken to the trending page (figure 7). Here, the most shared posts across all social media platforms are listed.**

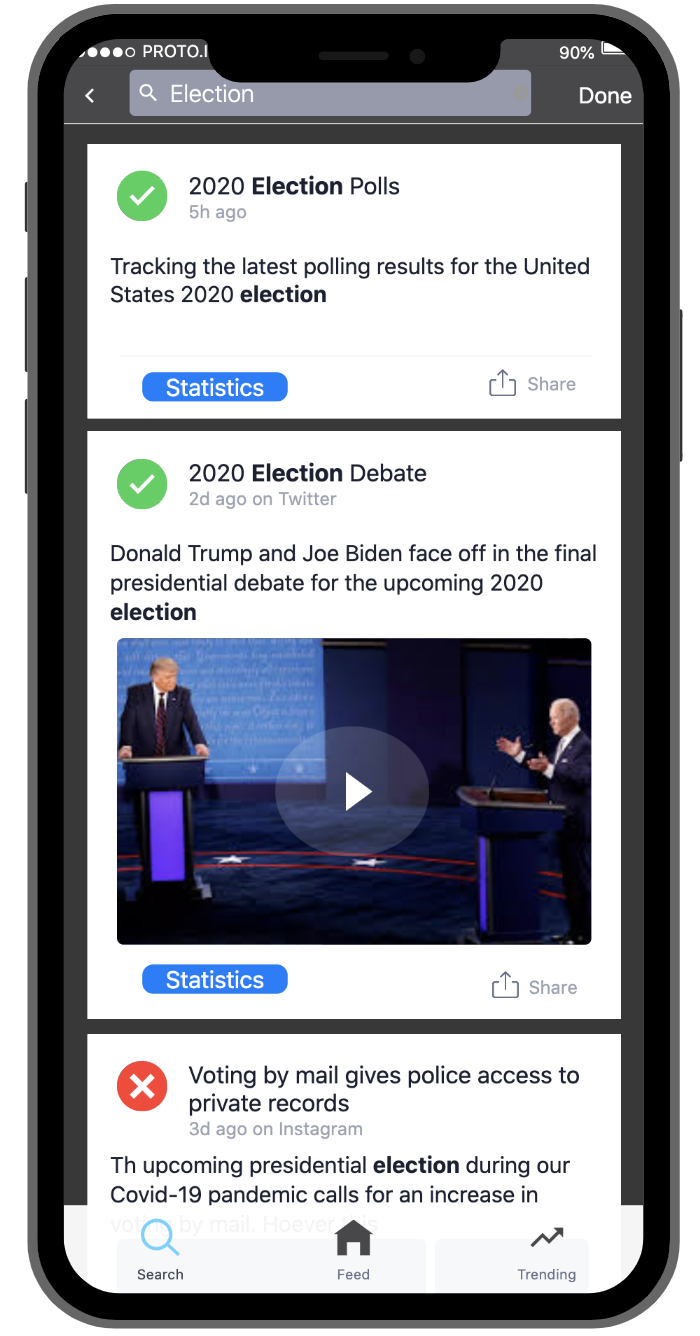
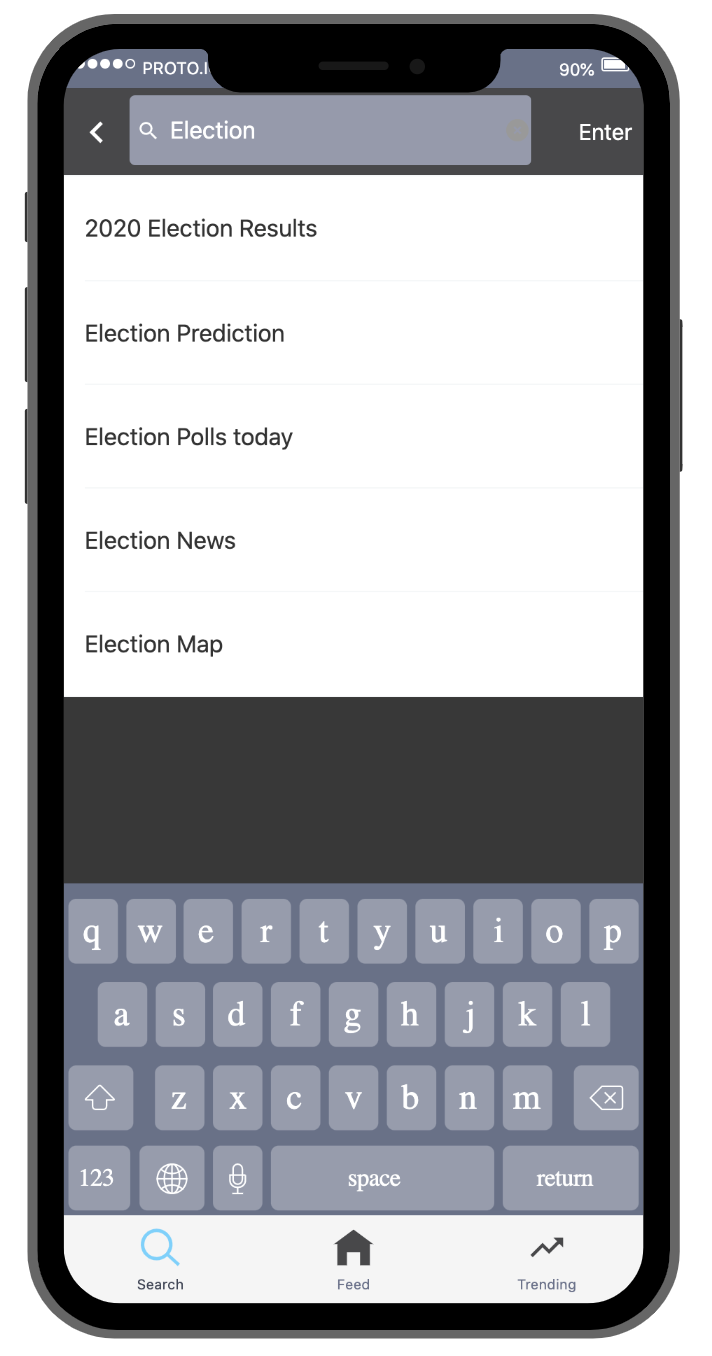
*Figure 7*

**Statistics page**: By pressing the ‘statistics’ button under a post, users are shown how many times the article was shared through each social media platform (figure 8). This page also shows how many times the article was shared by the hour.

**

*Figure 8*

**Search page**: Finally, Under the search tab, users are able to search for an article using keywords (figure 9). As the user types the app will list a prediction of what the user is trying to type so that the user can click on one for easy searching. If the search predictions aren’t quite what the user wants, they can finish typing what they want to search and then press send. The search results page will show articles that contain the keywords that were searched, filtered by relevancy and date posted (figure 10).

**

*Figure 9 Figure 10*

# Glossary

AI : Artificial intelligence

API: Application Programming Interface

iOS: The Operating System found on Apple mobile devices; the platform on which mobile apps run on Apple Devices

SEO: Search Engine Optimization

WCAG: Web Content Accessibility Guidelines

# References

Allcott H. and Gentzkow M., Social media and fake news in the 2016 election. Stanford University, Journal of Economic Perspectives 31(2): 211-236, 2017, p. 221

A. (2020, June 17). App Store. Retrieved October 28, 2020, from https://developer.apple.com/support/app-store/ps://developer.apple.com/support/app-store/

Clayton, Kathrine, et al. "Real Solutions for Fake News?" 11 February 2019. *Dartmouth.* Web Document. 27 October 2020. <https://www.dartmouth.edu/~nyhan/fake-news-solutions.pdf>.

Geiger, A.W. “17 Striking Findings for 2017.” *Pew Research Center*, Pew Research Center, 27 July 2020, www.pewresearch.org/fact-tank/2017/12/26/17-striking-findings-from-2017/.

Liu, S. (2020, July 06). Android versions market share 2019. Retrieved October 28, 2020, from https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/

Marsden, S. (2018, May 10). How do Search Engines Work? Retrieved November 10, 2020, from https://www.deepcrawl.com/knowledge/technical-seo-library/how-do-search-engines-work/

Decker, N. (2019, April 05). 508c vs WCAG - Comparing Web Accessibility Laws, Guidelines, &amp; Standards. Retrieved November 11, 2020, from https://www.velir.com/blog/2016/11/22/508c-vs-wcag-comparing-web-accessibility-laws-guidelines-standards

# Index

Below are clickable links to each heading and subheading in the document.

[Introduction](#_heading=h.7ktruanlz3ya)

[Types of Reader (Max)](#_heading=h.q6ypoqtyij08)

[Technical Background Required (Max)](#_heading=h.psupis3j4339)

[General Description](#_heading=h.k5lehgvarow0)

[Why Verific8? (Kenzie and Max)](#_heading=h.12jx58lzgp95)

[Product Functions (Max)](#_heading=h.e1a82ilso6m6)

[User Characteristics (Kenzie)](#_heading=h.hwt9oz6i6t86)

[Constraints (Serge)](#_heading=h.sk0vk078ik4p)

[Assumptions and Dependencies (Serge)](#_heading=h.th2mbpquofof)

[Specific Requirements](#_heading=h.fv1rl92wpelf)

[Functional (User) Requirements (Gia)](#_heading=h.3rqjzpjcwyzy)

[Availability and Performance Requirements (Anthony)](#_heading=h.rjsccz9qv23n)

[System and Integration Requirements (Gia)](#_heading=h.jojddsi54a79)

[Sustainability Requirements (Kenzie)](#_heading=h.oqv121dr1g6b)

[User Interface Requirements (Gia)](#_heading=h.8dagp4569qmd)

[Glossary](#_heading=h.b0a2j25xjmdk)

[References](#_heading=h.o251krt4g1on)

[Index](#_heading=h.4j74bzc3a5e)

1. Allcott H. and Gentzkow M [↑](#footnote-ref-0)
2. Clayton, Blair and Busamm [↑](#footnote-ref-1)
3. Decker [↑](#footnote-ref-2)