This project has received funding from the European Union's Horizon 2020 research and innovation programme under **grant agreement No 780355**

European Commission

FANDANGO

FANDANGO DELIVERABLE

Deliverable No.:	D5.2
Deliverable Title:	System and User Interfaces prototypes
Project Acronym:	Fandango
Project Full Title:	FAke News discovery and propagation from big Data and artificial intelliGence Operations
Grant Agreement No.:	780355
Work Package No.:	5
Work Package Name:	The FANDANGO software stack
Responsible Author(s):	Daniele Vannella, Alessandro Pisu, Camila Garcia, Marco Coluzza
Date:	30.07.2019
Status:	v1.0 - Final
Deliverable type:	OTHER
Distribution:	Confidential, only for members of the consortium (including the Commission Services)

Revision History

Version	Date	Modified by	Comments
V0.1	07.07.2019	Alessandro Pisu	First draft
V0.2	14.07.2019	Alessandro Pisu	Contribution
V0.3	19.07.2019	Camila Garcia	Contribution
V0.4	25.07.2019	Marco Coluzza	Review
V1.0	30.07.2019	Daniele Vannella	Final review

TABLE OF CONTENTS

List of Figures

List of Tables

ABBREVIATIONS

ABBREVIATION	DESCRIPTION
H2020	Horizon 2020
EC	European Commission
WP	Work Package
EU	European Union
UI	User Interface
GUI	Graphical User Interface
NAVBAR	Navigation bar
REST	Representational State Transfer
API	Application Programming Interface

EXECUTIVE SUMMARY

This document belongs to the FANDANGO project funded by the European Union's Horizon 2020 (H2020) research and innovation programme under grant agreement No 780355.

It is a confidential report that describes the system and user interface prototypes, implemented during the course of the project, starting from the first mock-up to the latest release. Each web-UI release was reviewed by the end-user, whose provide feedback to improve its design and usability.

Moreover, this document contains a description of current architecture and some specification about how front-end and back-end communicate.

INTRODUCTION

Currently, usability and ergonomics of the user interface (UI) in interactive applications are critical factors for achieving users' acceptance. Thus, from the perspective of the end-users, the success of the entire software development process depends on the quality of the interaction between user and system.

Fandango has mainly set itself the goal of usability as long as it is a necessary condition for survival on the Web. The most noted reasons why a user stop visiting/navigating a website are:

- the website is difficult to use
- the homepage fails to clearly state what users can do on the site
- users get lost on it
- information is hard to read or doesn't answer users

The probability that a user reads a website manual or spends time trying to figure out an interface is very low, since there are other websites available. Leaving is the first line of defense when users encounter a difficulty.

These considerations are, of course, applied in Fandango's web site, where users are journalists and demand for a friendly user interface.

Usability has many aspects, and is often associated with the following 5 attributes:

Easy to learn: The user can quickly go from not knowing the system to getting some work done with it.

Efficient to use: Once the user has learned the system, a high level of productivity is possible.

Easy to remember: The infrequent user is able to return to using the system after some period of not having used it, without having to learn everything all over.

Few errors : Users do not make many errors during the use of the system, or if they do make errors they can easily recover from them. Also, no catastrophic errors should occur.

Pleasant to use: Users are subjectively satisfied by using the system; they like it.

1. Iterative Process

Iterative user interface development involves steady refinement of the design based on user testing and other evaluation methods. Typically, once the design is completed, the problems which several test users have encountered using the application, must be noted. These problems will be fixed in a new iteration and then will be tested again to ensure that the "fixes" solved the problems and to find any new usability problems introduced by the changed design. The design changes from one iteration to the next are normally local to those specific interface elements that caused user difficulties. An iterative design methodology does not involve blindly replacing interface elements with alternative new design ideas. If two or more interface alternatives needs to be chosen, it is possible to perform comparative testing to measure which alternative is the most usable. Iterative design is specifically aimed at refinement based on experience obtained from previous iterations. In Figure 1 is reported the interactive process used.

1. The process begins with the translation of the user requirements into mockups.

   User requirements (Deliverable 2.3) specification is a documentation, which defines computer system (application) functionality, completely independent of any solution-oriented bias or software vendor. It must be understandable by the project stakeholders (users, developers, designers and project managers).

2. In the second step the mock-up is implemented and the interaction with the end-users starts.

3. The last step is the iterative phase where user feedback is retrieved and new versions of the interface are implemented. This process ends when users are satisfied in terms of usability and the interface has no bugs.

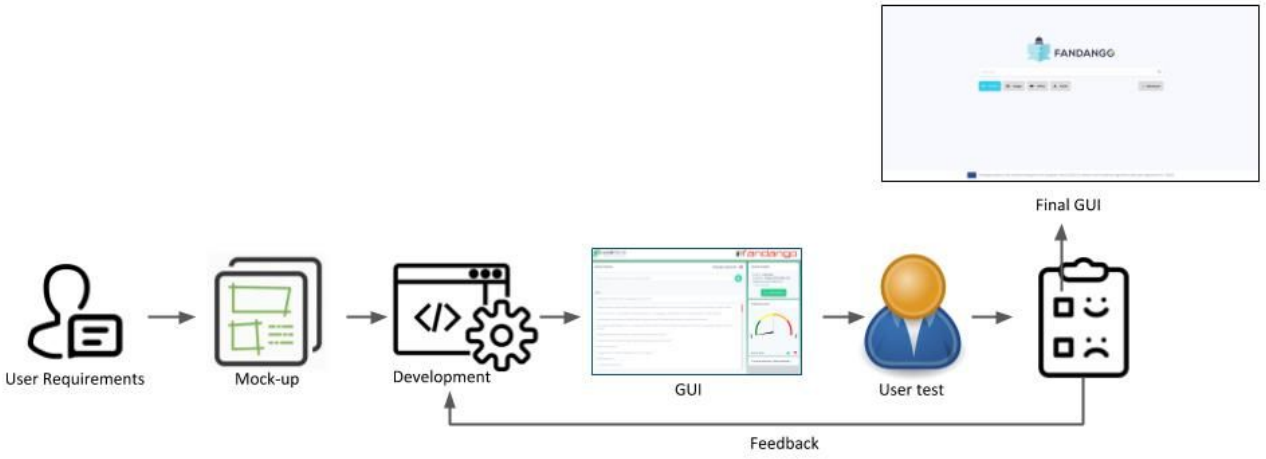

Figure 1 - Iterative process

The users' feedback of each iteration resulted in change requests as reported in Table 1.

VERSION	DATA	DESCRIPTION AND CHANGES	FIGURE
Mock-up	17/07/2018	User Interface Mock-up	2
V 0.1	29/10/2018	first implementation of the interface. Functionality: ● Text analyzer ● Fakeness level ● Feedback	3
V 0.2	27/02/2019	Implementation of the new interface style and features after first feedback of end-users Functionality and changes: ● EU brand navbar ● Authors score ● Claim selection ● New GUI color theme	4
V 0.3	15/05/2019	Implementation of new features inside the interface Functionality and changes: ● Images analyzer ● Videos analyzer ● Siren Dashboard button ● Similar news	5
V 0.4	18/06/2019	Restyling of interface with a new easier and simple version Changes: ● New “Google style” main homepage ● Separated pages based on analysis type (article, image, video and claim)	6-7
V 1.0	25/07/2019	Implementation of the style and features after first feedback about new interface style Changes: ● Minor UI styles fix	8-12

		• Images and videos analyzed in own page • Claims analyzed in own page	

Table 1 - Change Request list (release basis)

The following sections briefly show and describe the evolution of the User Interface.

2. MOCK-UP

Agile approach, one of the key methodologies used in today's software projects, often rely on user interface mockups for capturing the goals that the system must satisfy. User Interface mockups are useful tools for describing scenarios where real-life data are used for exemplifying the use case instead of abstract descriptions. However, the information provided by mockups are informal and allows misunderstandings by different stakeholders

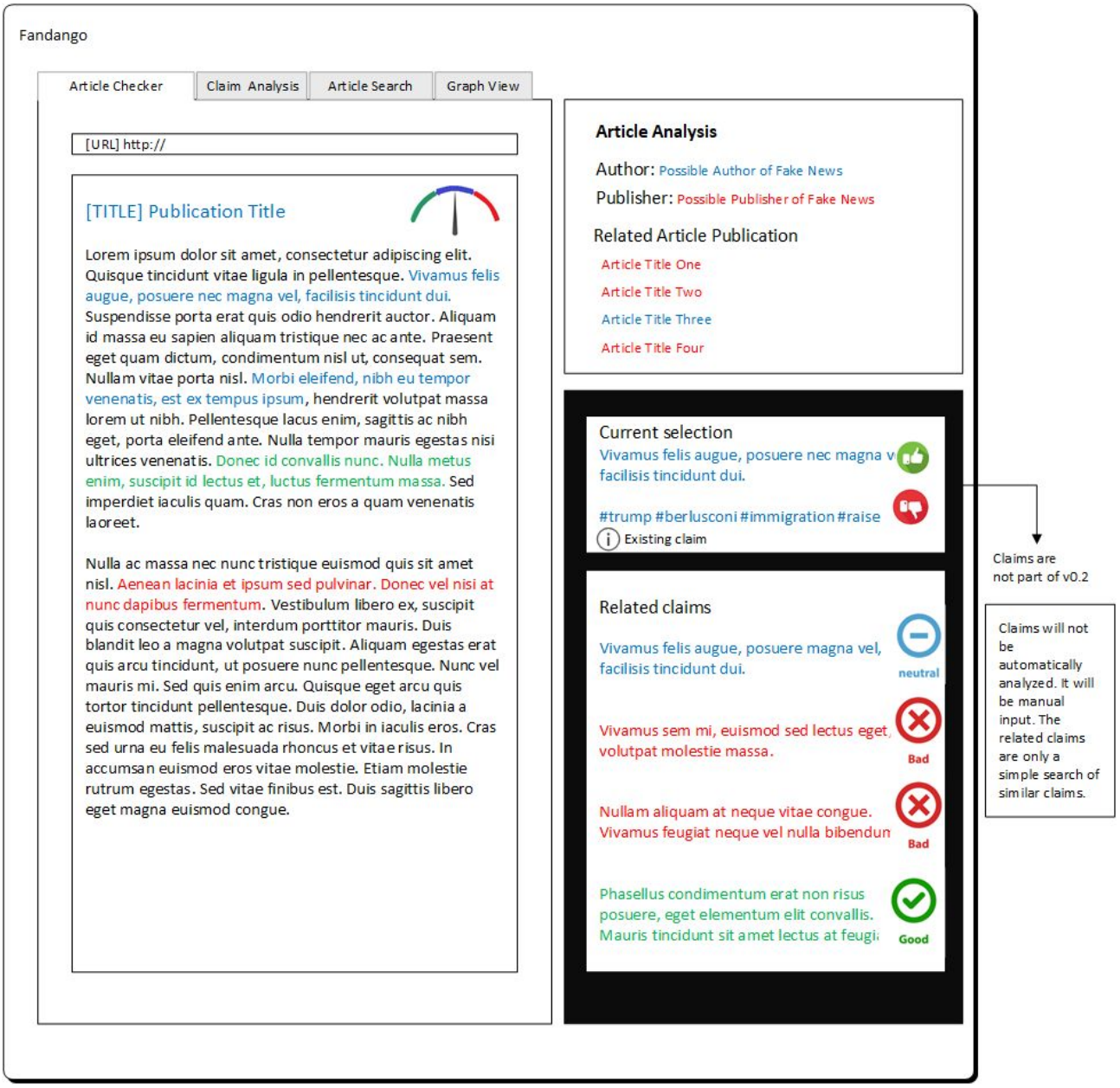

Figure 2 - Mock-up

Mockups show the mainly functionality requested by end-user in User Requirements:

- Text Analysis
- Authors and Publishers analysis
- Claims Analysis

3. FANDANGO RELEASES

3.1. VERSION 0.1

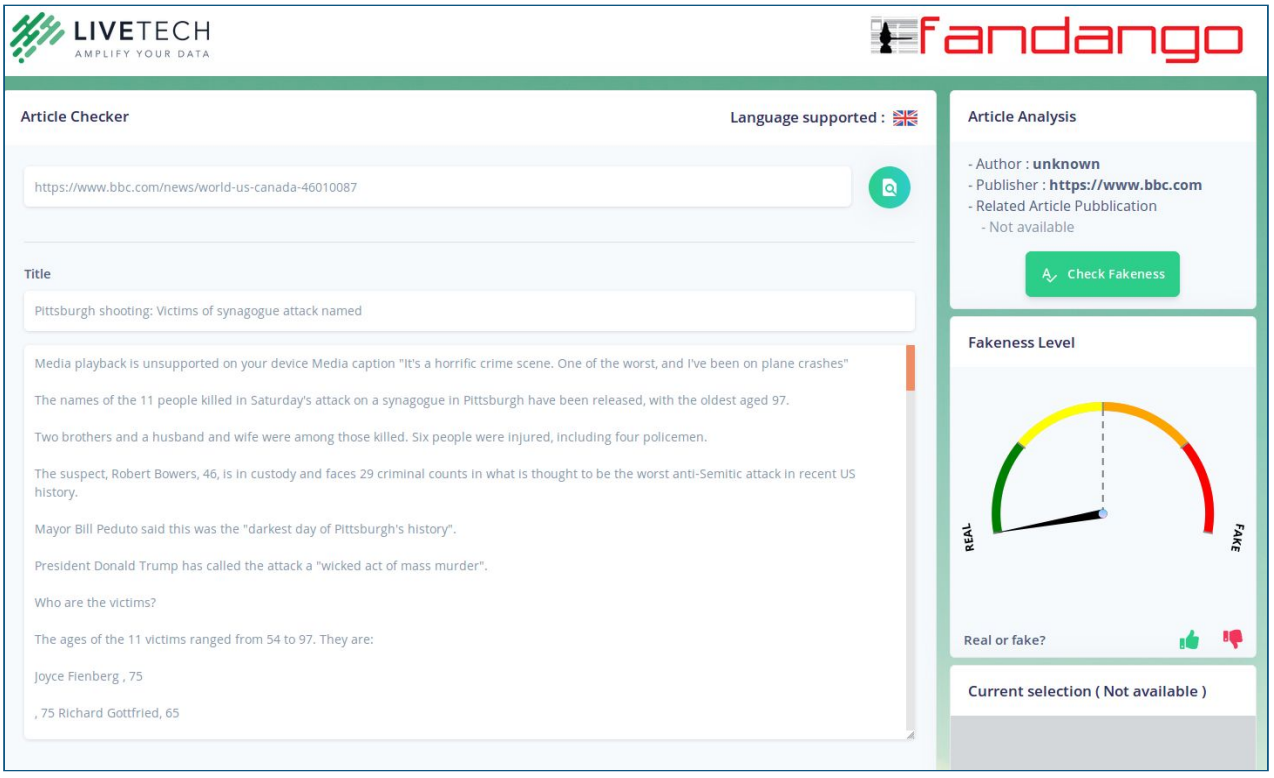

Figure 3 - Article Analysis Page V0.1

Figure 3 shows the first implementation of Fandango Project UI. In this phase, the GUI has been implemented with the following functionalities:

- Article visualization (Title, body, authors and publisher)
- Fakeness Level (represented to the right with a gauge chart and using a draft of machine learning model)
- Fakeness reliability feedback

  Main feedback from users were:

  - Unprofessional design
  - Missed functionality

3.2. VERSION 0.2

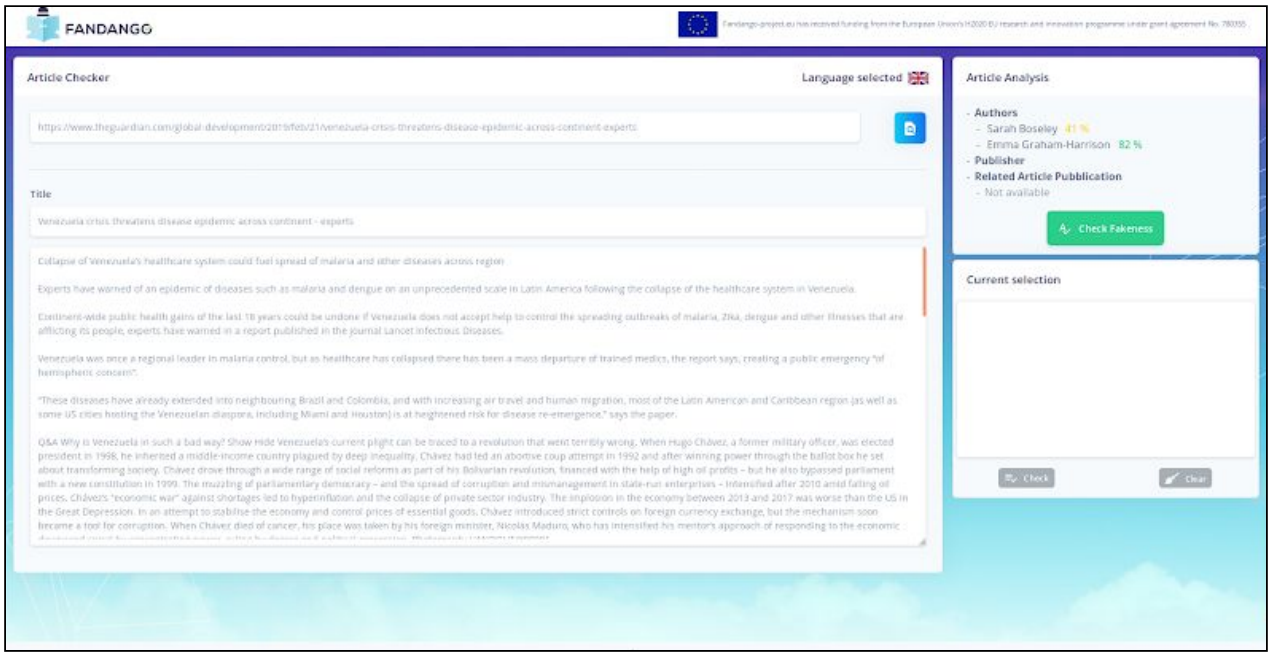

Figure 4 - Article Analysis Page V0.2

Figure 4 shows the second Fandango UI version which implements the new features and first feedback of end-users about the usage of interface.

The most important functionalities added was:

- Authors score
- Selection of claim inside an article body
- Similar claims service

The minor changes about UI style were:

- New UI style with Fandango colors
- New navigation bar with EU and Fandango logos

3.3. VERSION 0.3

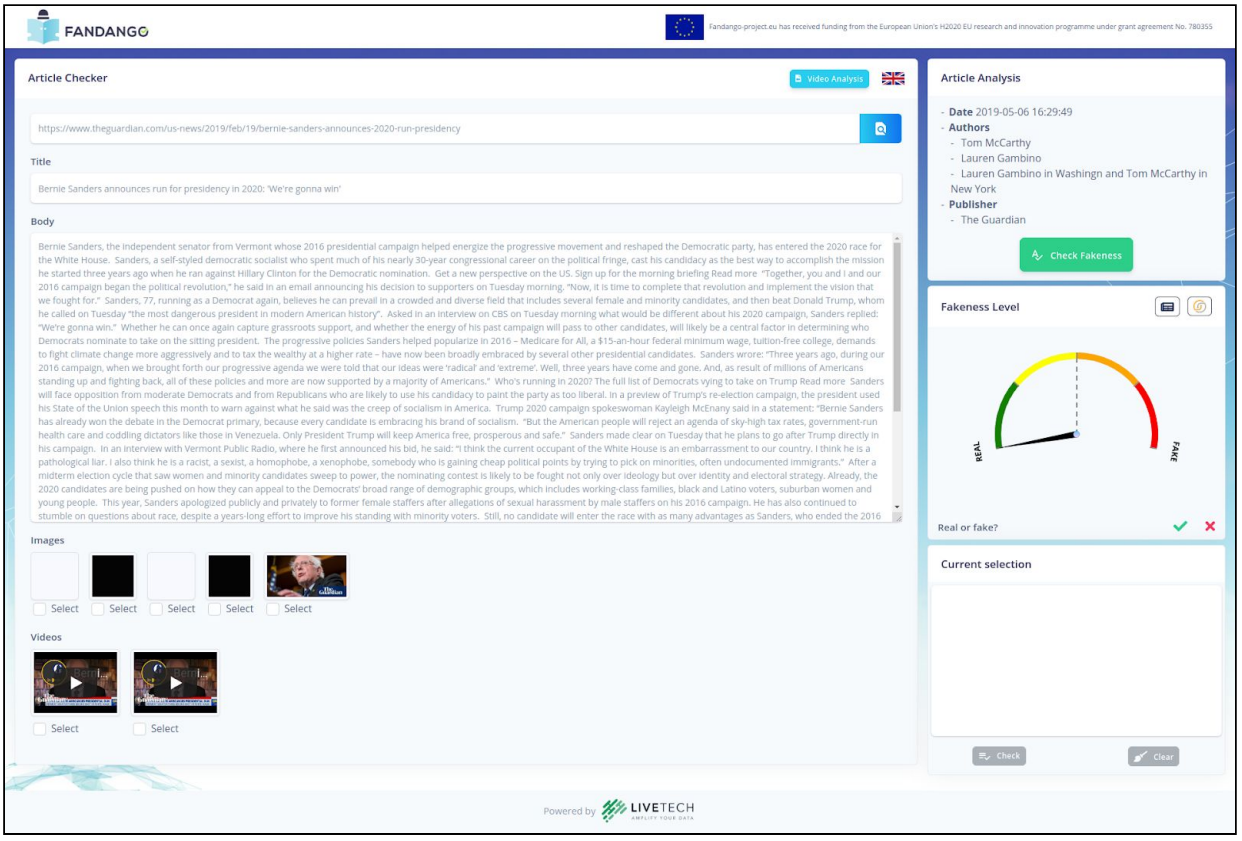

Figure 5 - Article Analysis Page V0.3

Figure 5 shows the third Fandango UI version which implements the following features:

- Article language recognizer
- Article date
- Images analysis
- Videos analysis
- Similar news button
- Siren Dashboard button

The minor changes about UI style were:

- New footer at the bottom of page

3.4. VERSION 0.4

The version 0.4 has a new style interface. Infact, the structure of sites is fully changed. Now the website is split into functionality pages:

- Homepage (Main page)
- Article Analysis page
- Image Analysis page
- Video Analysis page
- Claims search page

In this version, we show only Homepage and Article Analysis because these are the main pages that change in the next version (interaction).

HOMEPAGE

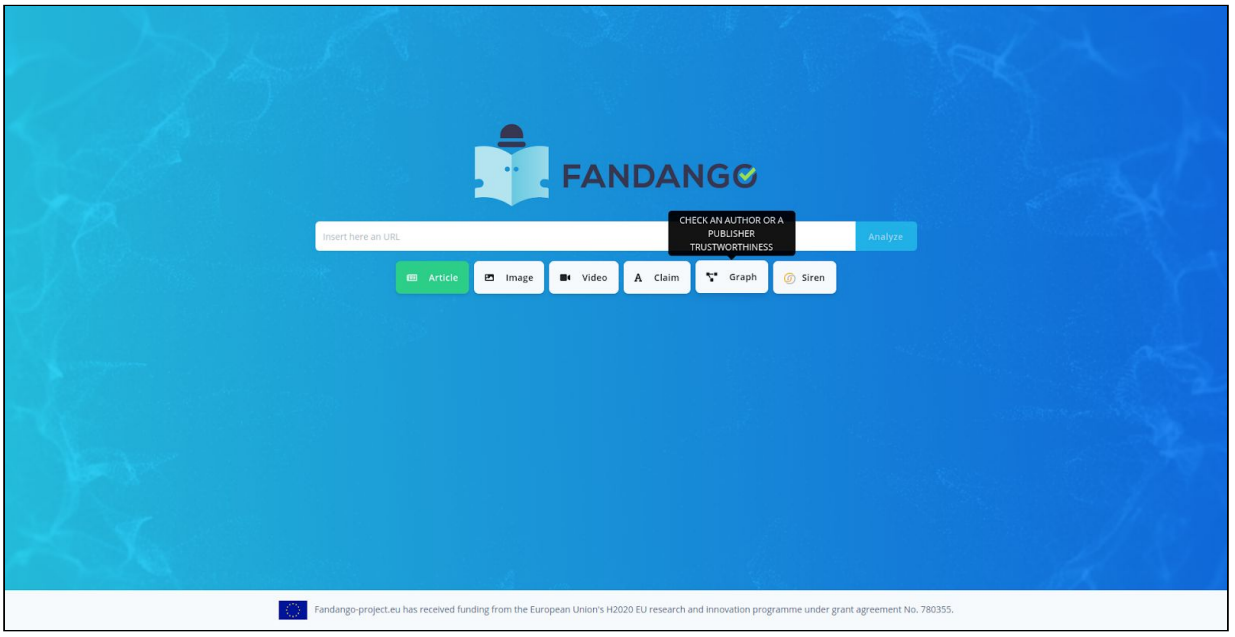

Figure 6 - Homepage V0.4

Figure 6 shows the new Fandango UI homepage. After the need for a more intuitive and simple interface, this version implements a totally new and different GUI.

The main changes were:

- New main homepage to switch functionality (analysis types)
- New homepage style
- New footer with EU logo
- Analysis infos on buttons mouse-over

ARTICLE ANALYSIS PAGE

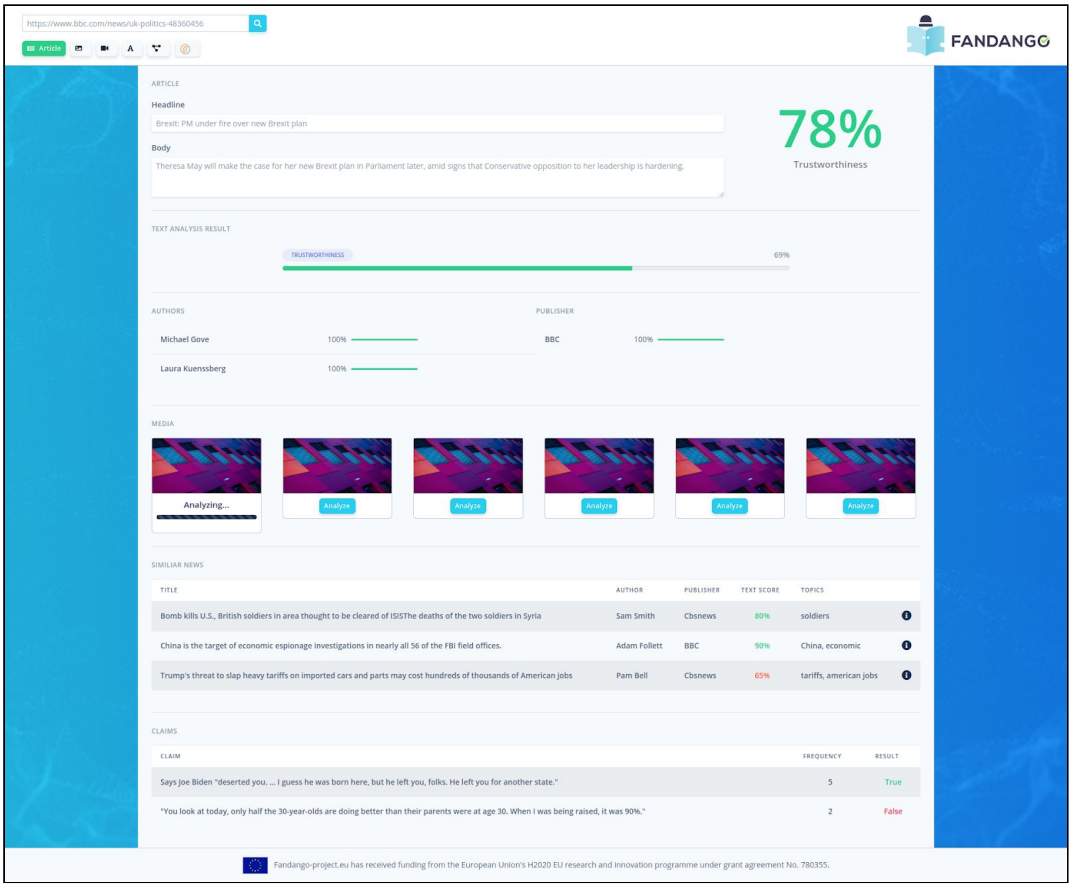

Figure 7 - Article Analysis Page V0.4

Figure 7 shows the new Article analysis page of Fandango UI, based on new feedback and mock-up received during Thessaloniki meeting.

The main changes were:

- New UI pages style
- New type of trustworthiness visualization (from chart to percentage)
- Text analysis result (represented on progress bar)
- Images and videos analysis
- Similar news
- Claims
- New navigation bar with search bar and the possibility to switch type of analysis with buttons

4. Latest Release (M19)

The latest version of Fandango UI (Figure 8) implements some minor UI fixing according to end-users feedback to the new design.

All the UI features described in this section are supported by FANDANGO's services which are fully described in D2.4 and D3.1.

Homepage

In Homepage of Fandango (Figure 8), the users can navigate the different functionality of Fandango:

- Article Analysis - a user, typing a url, can analyze an article
- Image Analysis - a user, typing a url of an image, can analyze it
- Video Analysis - a user, typing a url of a video, can analyze it
- Claims Search - a user can search a claim by typing a sentence

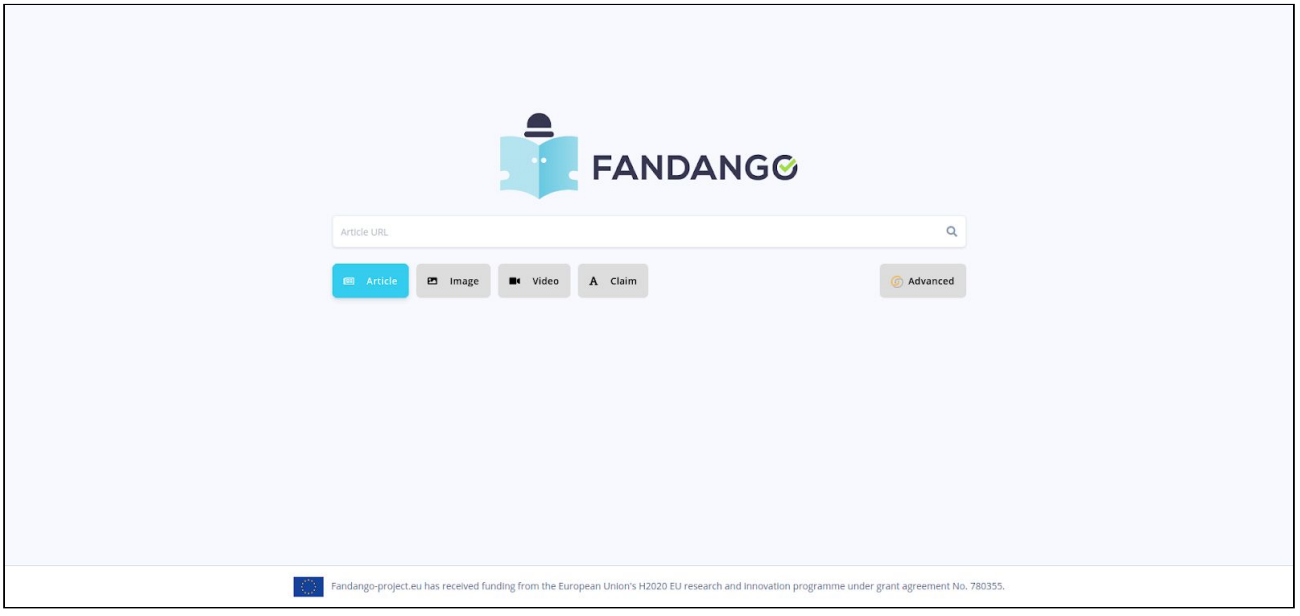

Figure 8 - Homepage

How to use fandango's user interface:

1. Select the desiderated function clicking on the respective button.
2. Type/Copy a url or a claim
3. Click on search button result

The homepage changes are:

- Clear background
- Separation of Siren Dashboard (Advanced) from Article, Image, Video and Claims search
- Type checker based on type of analysis (For example: Article analysis accept only URL inside textbox)

The navigation bar (Figure 8) changes are:

- Fandango logo moved on the left
- Buttons on navbar now are always expanded with its label
- Clicking a button on the navbar, the interface will redirect the user to the homepage with the selected analysis

ARTICLE ANALYSIS PAGE

In this page (Figure 9), the user can visualize results of the news analysis in terms of text, authors and publishers information.

- The text analysis score (Figure 9.a) is reported in a progress bar. It represents how reliable a news is (from 0 to 100%).
- The Authors and Publishers scores (Figure 9.b) are reported in a progress bar as well, one for each author and publisher. These scores represent the credibility of authors and publishers

In Figure 9.c and 9.d are shown the media sections, where users can choose an image or a video to analyze (clicking on one of them will open the respective analysis page)some of their aspects(editing, heatmap, etc.).

Moreover, in the last section of the same page (Figure 9.e), users can explore a list of 10 similar news, already analyzed by the system and retrieved from Fandango's silos.

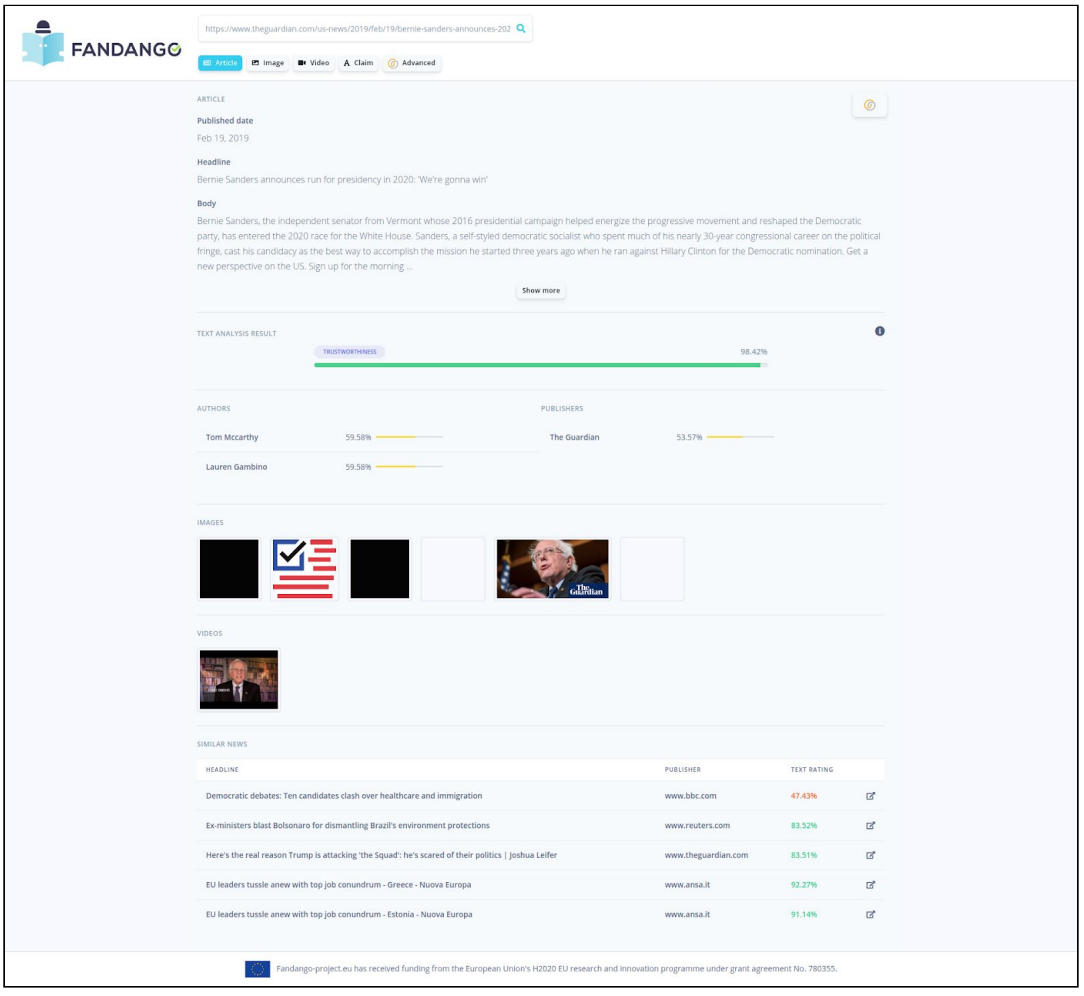

Figure 9 - Article Analysis Page

Figure 9.a - Article Analysis Page: text analysis details

Figure 9.b - Article Analysis Page: Authors and Publisher analysis details

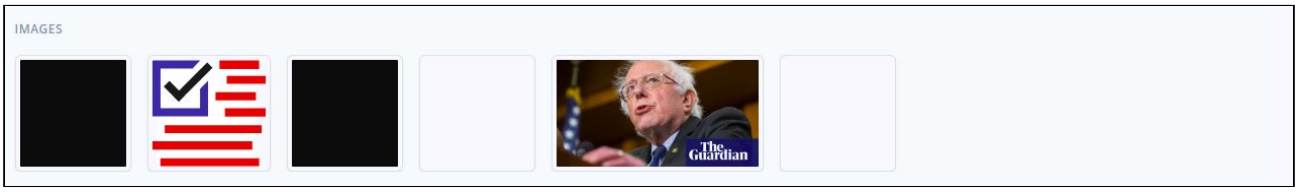

Figure 9.c- Article Analysis Page: Images

Figure 9.d - Article Analysis Page: Videos

Figure 9.e - Article Analysis Page: Similar news

The article analysis page changes are:

- Clean background
- Edit boxes removed on article analysis result to prevent misunderstandings
- Article date added
- Siren investigation button added to explore dashboards
- Images and videos analysis results shown in single pages
- Claims removed from article analysis page(Figure 9)

IMAGE ANALYSIS PAGE

In this page (Figure 10), the user can analyze an image to see if the image has been manipulated or not. The different analyses are reported in D4.3.

The user can see every analysis made through a list of images. Each image represents a different algorithm applied on the original one. For instance, if there are manipulations on a single section of an image, the user will find a highlight of the manipulated section with a boundbox.

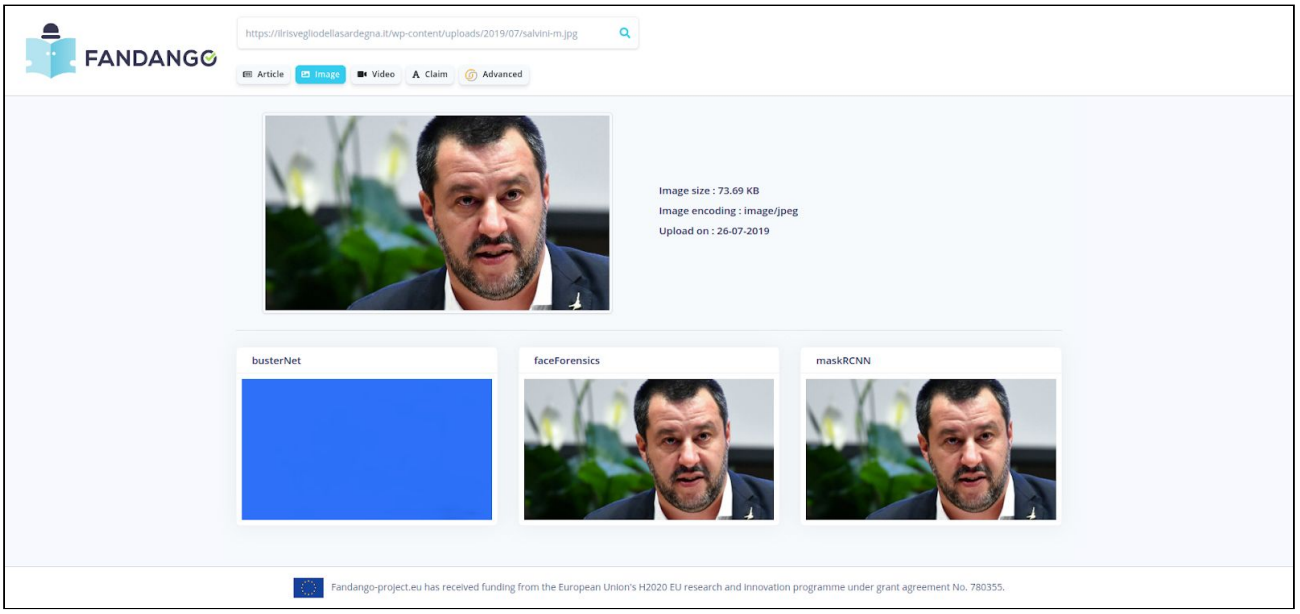

Figure 10 - Image Analysis Page

The new image analysis now includes:

- Image Analysis
- Image info (size, encoding and upload date)
- Image with different analyzer type (busterNet, faceForensics, maskRCNN)

VIDEO ANALYSIS PAGE

In this page (Figure 11), the user can analyze a video to see if the video has been manipulated or not. A more detailed explanation is reported in D4.3

This is the slowest functionality in Fandango. The reason is that the analyzer check each frame on the video.

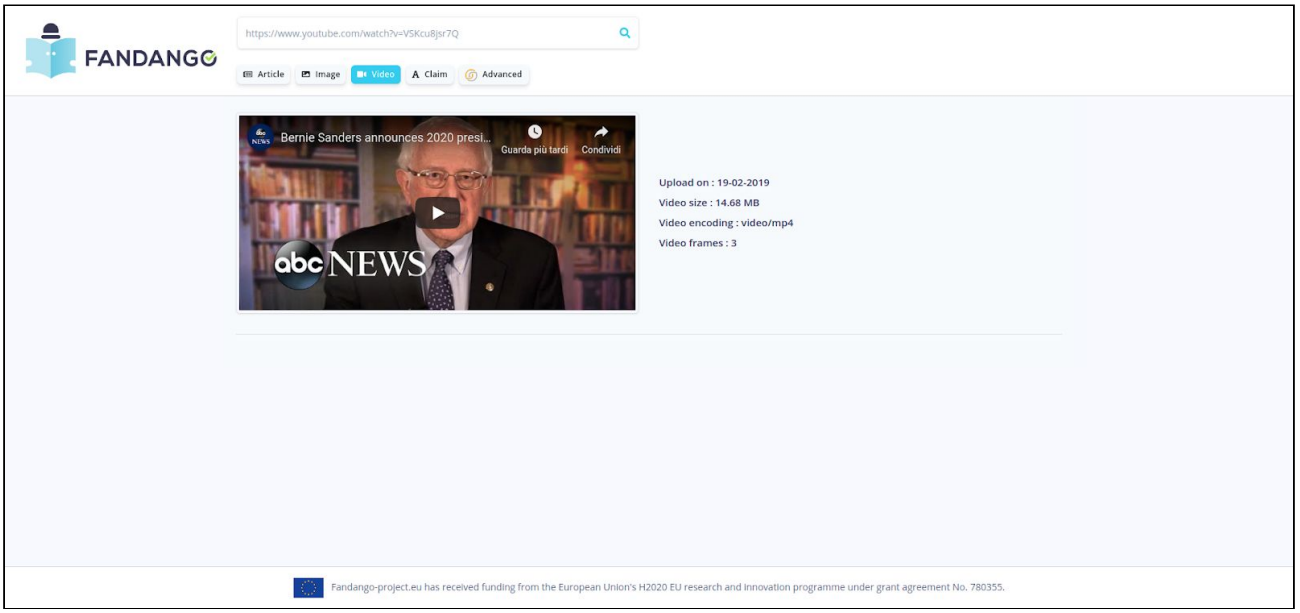

Figure 11 - Video Analysis Page

The new video analysis now includes:

- Video analyzed
- Video info (size, encoding, upload date and frames)

In the next version, it will be added analysis of the single frames of the video.

CLAIM SEARCH PAGE

In this page (Figure 12), the user can research claims that have been already reviewed by trusted sources.

Claim details (Figura 12.a) shown are:

- Author name
- Label
- Rating value
- Worst rating
- Best rating
- Reviewed body
- Link to navigate the webpage from where the claim comes from

Claim's details can be expanded("see more") by clicking the arrow on the right.

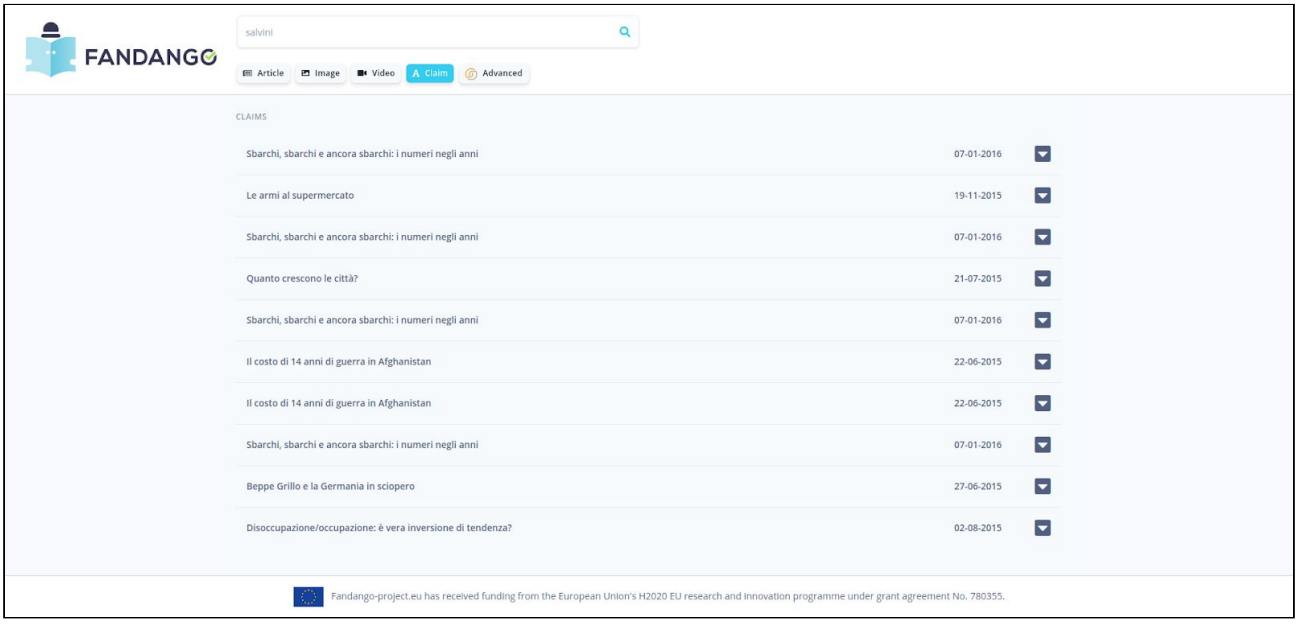

Figure 12 - Claims Page

Figure 12.a - Claims Page details

The new claim analysis now includes:

- List of claims found

5. UI PROPOSAL

Subsequently to last version, a UI proposal has been created from the following feedback given by end-users (Figure 13-16):

- Switch analysis type with buttons under textbox is not clear
- Button mouse-over information not intuitive
- Differences between Analysis button and Advanced button not evident
- Buttons inline visualization not optimized in small desktop / mobile
- Navigation bar not optimized in small desktop / mobile

HOMEPAGE (DESKTOP VERSION)

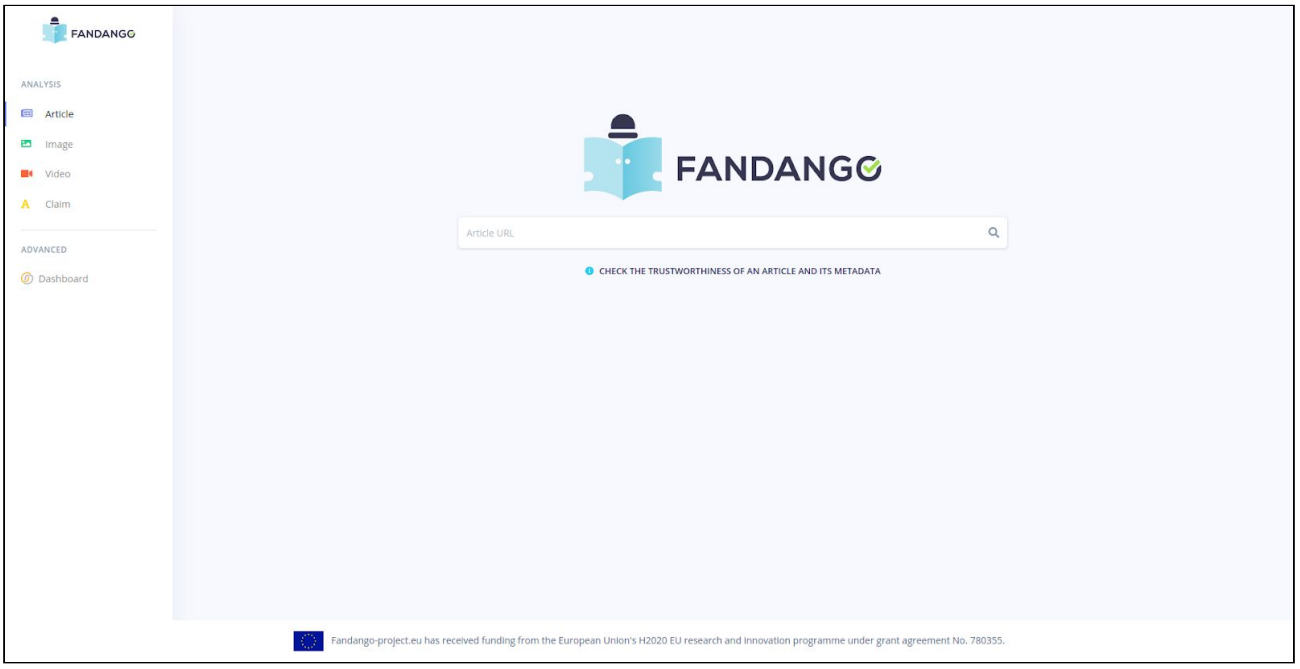

Figure 13 - Homepage (Desktop version)

HOMEPAGE (MOBILE VERSION)

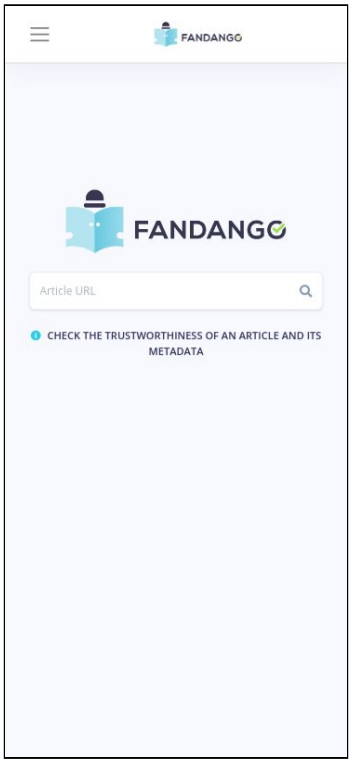

Figure 14 - Homepage (Mobile version)

ARTICLE ANALYSIS PAGE (DESKTOP VERSION)

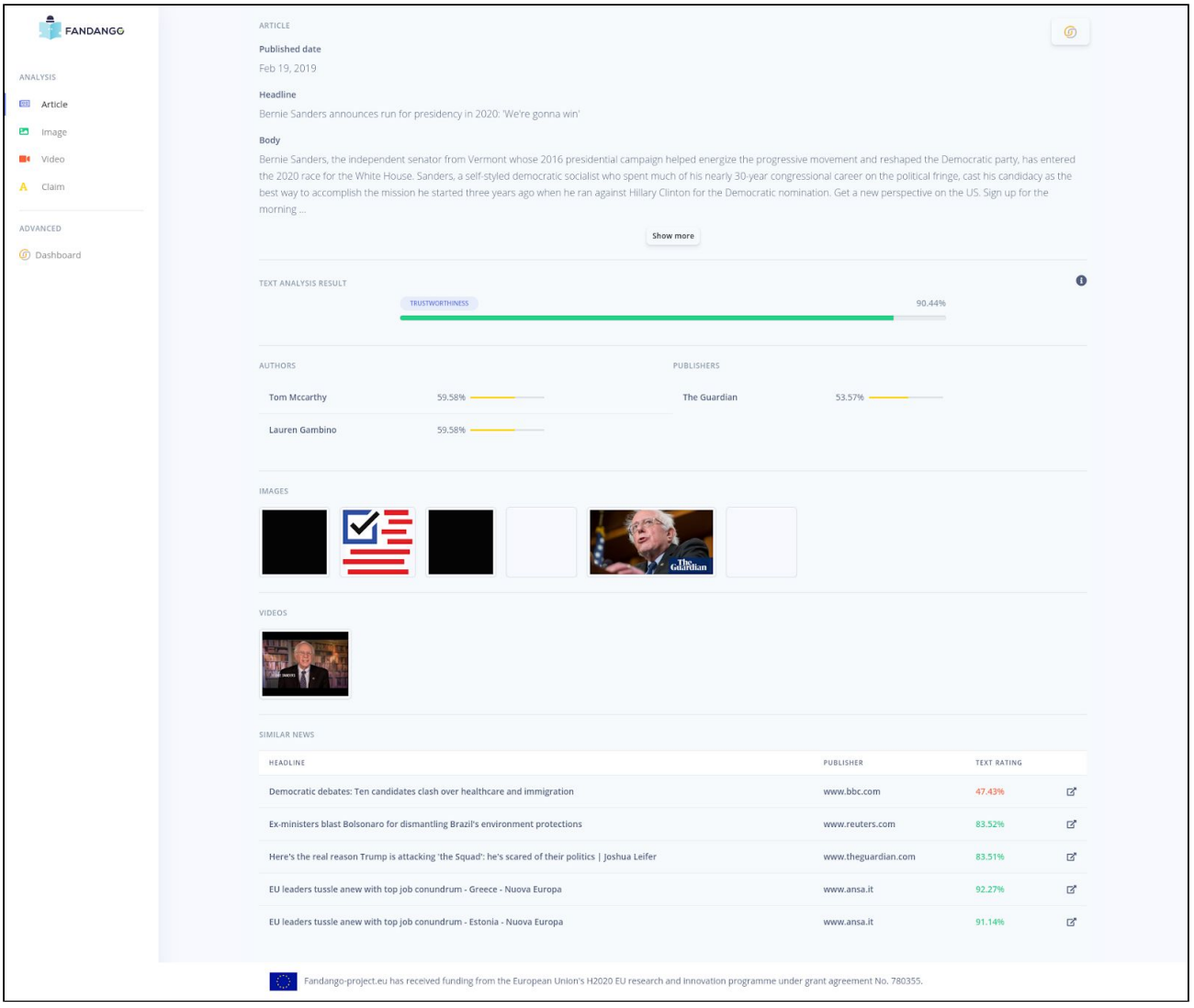

Figure 15 - Article Analysis Page (Desktop version)

Article analysis page (mobile version)

Figure 16 - Article Analysis Page (Mobile Version)

UI proposal advantages

- Analysis type info always visible under textbox
- Better space optimization
- Sidebar always available during navigation
- Most obvious difference between Analysis functions and Advanced function
- Sidebar is fully responsive and adaptive with large, medium and small desktop, tablet and mobile
- Cleaner design and more intuitive for the first usages

6. ARCHITECTURE

The Web application is entry point of Fandango (Figure 17).

Fandango Web application is a Single Page Application (SPA). it is the dynamic framework that covers the interaction with the user by the updated data within the existing page without loading totally new pages from the server, this way the application requests only needed content details. AJAX, Asynchronous JavaScript, and JSON are the foundation of the page communication. Additionally, the HTML5 History API provides the ability to adjust the URL of the page without its reloading, simultaneously creating separate URLs for different views. Generally, the server queries can carry various types of data, using the form of JSON payloads or HTML elements.

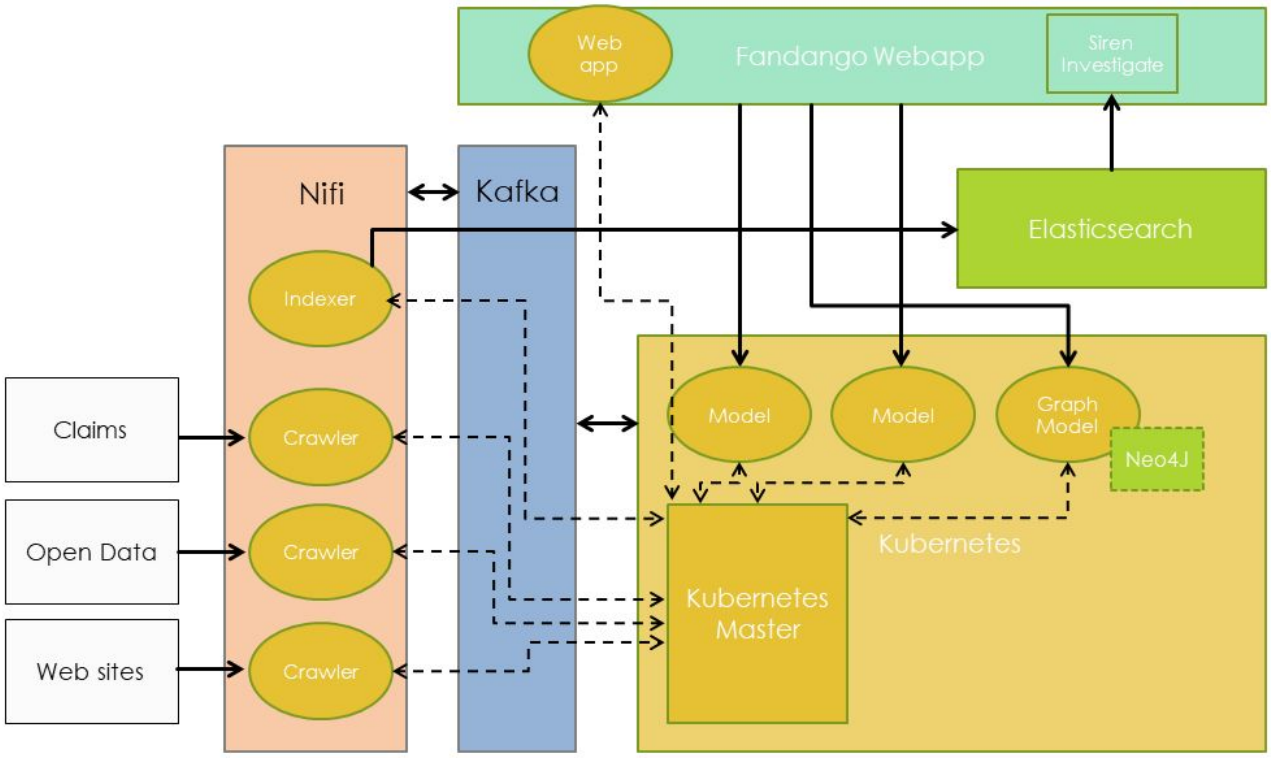

Figure 17 - Architecture

Fandango Web application combines both server-side (Back-end) and client-side (Front-end) scripts. The server-side scripts are responsible for the analysis of the news, image, video and claim, while the client-side present the data to customers. Web application architecture diagram is reported in Figure 18.

Process of the Web application:

- The user sends a request to the server through the Internet, using the browser.
- The web server is responsible for forwarding the command to the requested server.
- The server processes the user's request.
- The web application delivers the processed information by the server to user.

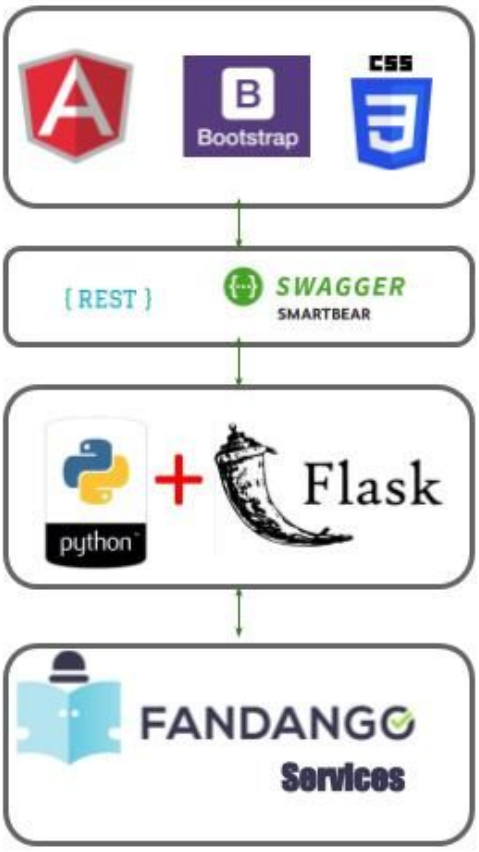

Figure 18 - Web application

Components:

- Front-end is written in JavaScript programming language, HTML5 and CSS3. It used two frameworks called Angular and Bootstrap, as well. The version of the frameworks used are Angular 7 and Bootstrap 4. Front-end receives the information through web-services via RESTful API from the back-end side.

  Angular is a TypeScript-based open-source web application framework led by the Angular Team at Google.

  Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

- Back-end side is written in Python programming languages. It uses a framework called Flask. Through Flask the back-end can expose RESTful API to communicate with the front-end part. The version of Python used is 3.6.

Flask is a lightweight WSGI[1] web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug[2] and Jinja[3] and has become one of the most popular Python web application frameworks.

The Swagger[4] framework has been used to facilitate the modeling of the RestFul API. Infact, the API design is prone to errors, and it's extremely difficult and time-consuming to spot and rectify mistakes when modeling APIs.

Swagger UI is one of the most popular tools for generating interactive documentation from your OpenAPI document. Swagger UI generates an interactive API console for users to quickly learn about your API and experiment with requests.

[1] https://wsgi.readthedocs.io/en/latest/
[2] https://palletsprojects.com/p/werkzeug/
[3] http://jinja.pocoo.org/
[4] https://swagger.io/