SYSTEM DEMO REPORT

Purple Team

Athrey Ram Vinay
Jing Qiu
Ross O'Riordan
Vishnu Suresh
Xufeng Zhang

Contents

1. User Scenario: The Characters

We are basically targeting 2 groups of users:

- ***Tourists***
- ***Research students***

We have identified the target users based on the following observations:

1. Tourists:
Our application would assist tourists about the trending news of the place they are planning to visit. This would help them in deciding not to visit the place if there are any tensions going on. Also, this would help them by giving them weather forecast related information in order to plan the trip accordingly. The product would give the tourists a clear picture of the city/country.

2. Research students:
Since the world is moving towards internet, a lot of research students are researching news online to get the latest updates on news. Our product would provide a perfect platform to such users to learn a lot
about the current trending news of various countries along with tweets and videos and get different perspective about a particular news from different countries.Also, since we are pulling in the news from a wide range of newspapers, the user can
virtually go through all of them rather than only a single source.

2. Technical Problem: The Setting

Our application is a location based retrieval and display of trending news across the globe and the related tweets and YouTube videos. This system exists since it provides the user the freedom to pick the country he wishes to read the news from, a category to choose from and get a unique summary of articles rather than reading the entire news to understand it and view the response of the world through the tweets and have a visual of the news through YouTube videos.

We have four major technical problems:

1. Finding a common way to extract news from the newspaper
Since we are completely dependent on Newspapers to provide material to the user, only after the new sites update their news we can display the news we can take the news and check Twitter and YouTube for relevant/supporting materials. There might be delays in displaying the information. Also, we faced challenges in extracting the news and finding a common way to do it.

2. Finding Summary:
We are not just providing the hyper link of a newspaper to the user, we are providing a unique summary of each article but obtaining a summary would be difficult since we need to parse through the entire document and then pick those words that give the gist of the news. At the moment we are dependent on the Newspaper API but sometimes it may not give an apt abstract. Hence we are passing the first 1500 words of the news after performing a survey on multiple articles and found out that the first paragraph contains the summary of the news.

3. Relevancy of YouTube videos:
During the initial stages, we were passing the article heading as a query to YouTube to retrieve the videos and as a result the relevancy of the videos were very less. Therefore we decided to break the heading into multiple keywords and pass the array of keywords. This increased the accuracy of relevancy to some extent. We still face issues in finding the measure of the relevancy and have left it to the user and provide him a survey to rate the relevancy of the video.

4. Support of multiple languages:
The system only works well with English language. Since we have explored into many non English speaking countries, we have used Google Translate API to convert the languages into English. This leads to a delay in displaying the information. Also the accuracy of the language conversion might be an issue. The problem could be seen evidently when results for a specific query are returned. The tweets and videos are not related to the news that is displayed. Also, if there are no videos or tweets uploaded, the related fields are left blank.

Similar products are:

- ***NewspaperMap:***

Uses Google Maps API to display the map and the map is crowded with location markers which makes the user difficult to select the newspaper since he has to zoom in to the location and is not shown a clear display of location on the first go. Also, this does not display the trending/current news articles, instead redirects to the newspapers home page (which is like reading a normal newspaper). No tweets or YouTube videos or summary shown.

- ***NewsMap:***

Does not display an actual 2D map, instead provides tab of various countries. Once the country is selected, the trending headlines of the article is displayed and once clicked on them redirects to the homepage of the newspaper. Most latest news are shown as bigger boxes and less trending news are gradually decreased and are given less size boxes resulting in over crowding of the web page. The countries to select from are also very limited. But since they are big, they are getting the latest News from a very wide range of newspapers (i.e 400+). Hence the delay in getting the most updated news is low here. No tweets or YouTube videos or summary shown.

- ***NewsHunt:***

A mobile application that uses sources from national, regional and other websites for retrieval of news. This application displays news in multiple languages using the Google Translate API but has geographic limitations.

- ***Yahoo NewsDigest:***

Again another mobile application that displays news and a unique summary of the same. No tweets or YouTube videos shown.

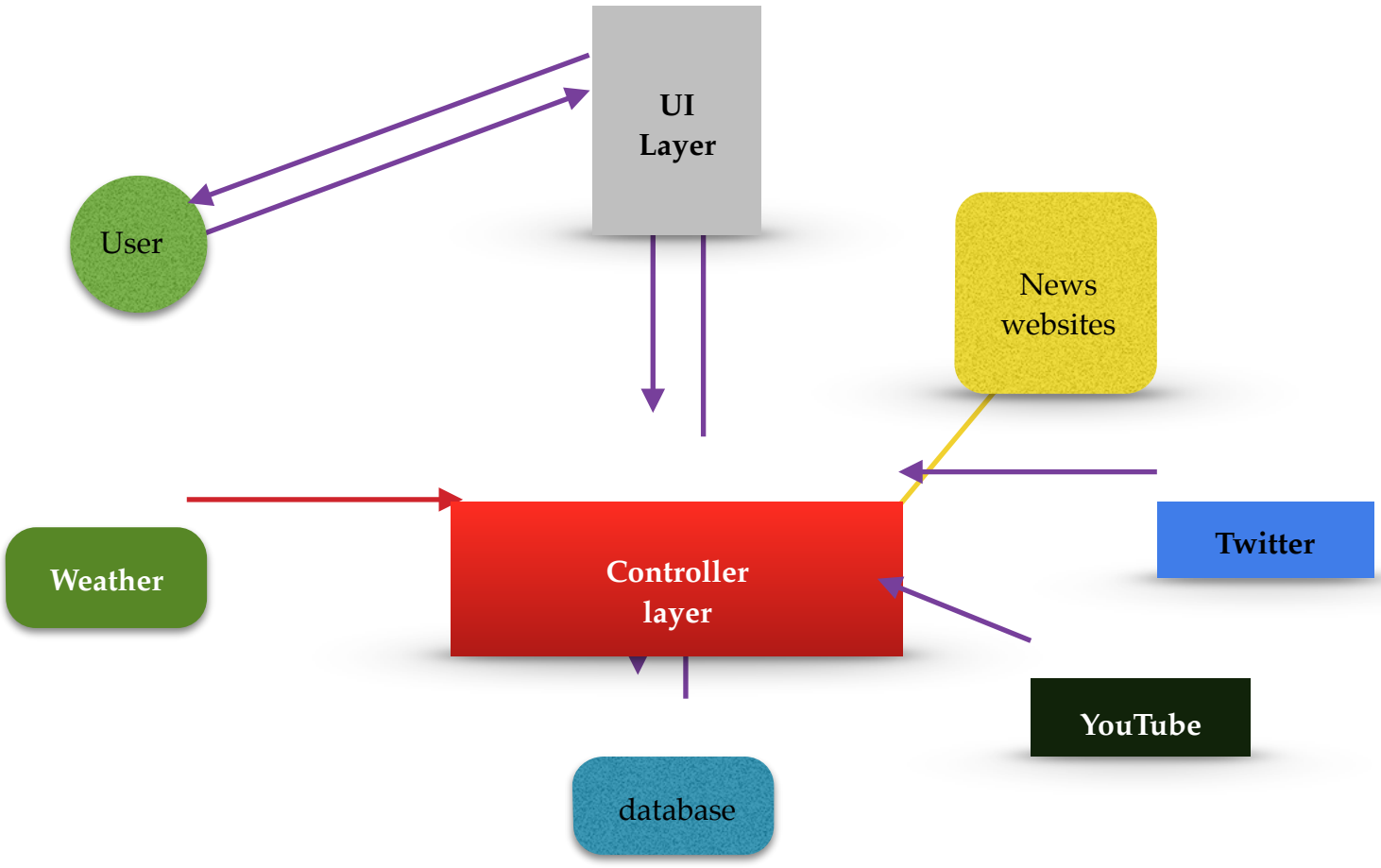
3. Technical Solution: The Plot

A location based application that retrieves and displays contextual information associated with selected articles from a specific country.

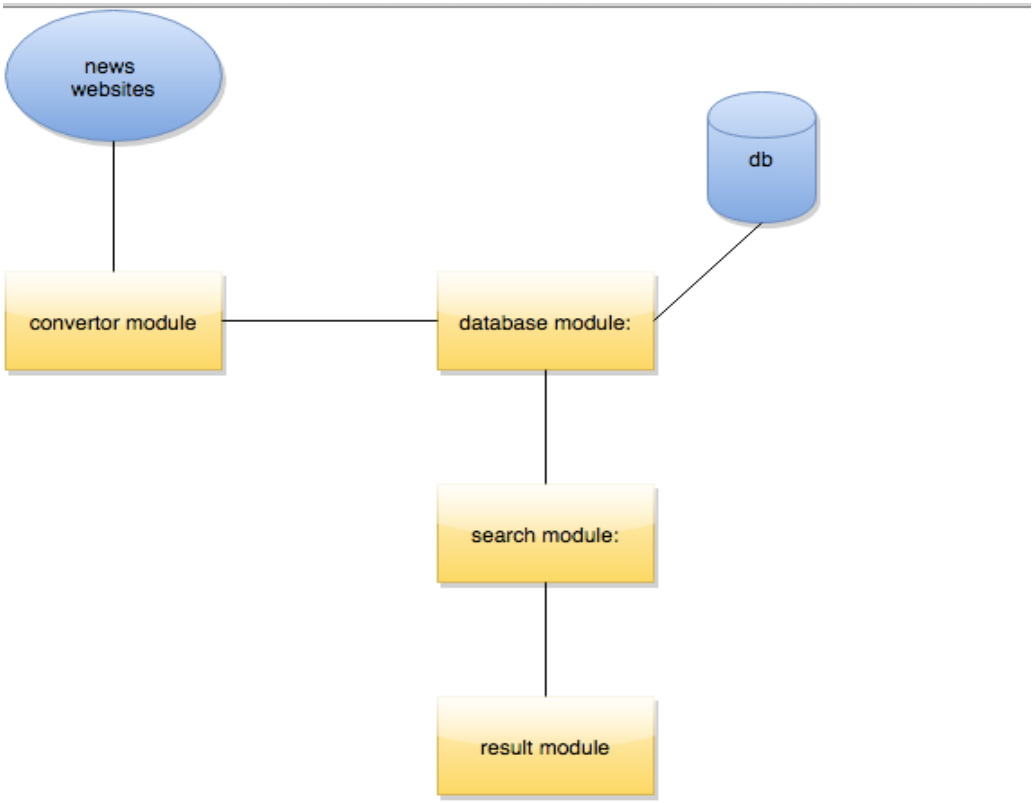
More specific:

1. A sophisticated UI that will display contextual information in a user-friendly manner.
2. Retrieve weather information based on country location.
3. Retrieve trending tweets from twitter API based on hashtags derived from news article's heading/sub-heading/main-body.
4. Retrieve relevant youtube videos related to selected articles
5. Display additional information based on the articles category e.g. sports, entertainment, politics. Such information will include e.g. betting odds (sport), local temperatures (weather), voting stats (politics) etc.

System Diagram



Back-end: Technical components/Data sources



Module description:

News websites:
Web source based on given urls. Chosen newspaper is based on its popularity and universal.
Ireland: The Irish Times *(**http://www.irishtimes.com/**)*
England: The Telegraph *(**http://www.telegraph.co.uk/**)*
Spain: Elpais *(**http://elpais.com/**)*
German: Beld *(**http://www.bild.de/**)*
Italy: (**http://www.corriere.it**)
France: (**http://www.thelocal.fr/**)

Convertor module: Fetch, analyse and process newspapers.
Fetch: retrivie news page source through python urllib library.
URLLIB provides a high-level interface for fetching data across the World Wide Web. In particular, the urlopen() function is similar to the built-in function open(), but accepts Universal Resource Locators (URLs) instead of filenames.

E.g. urllib.request.urlopen(url).read()

Analyse:
Analyse retrived news source to get articles' information (title, image, …), this is implemented mainly by regular expression , BeautifulSoup ,and newspaper lib.
Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

E.g.
```
soup.find_all(class_=re.compile('headlineImage'))
```

Newspaper provides simplified python article discovery & extraction.

e.g.
```
a = Article(url,language = 'de')
a.title # get articles title
```

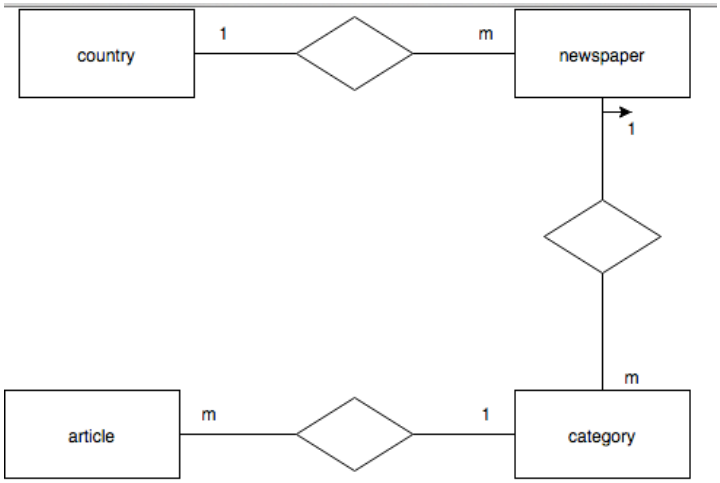
Process: Encapsulate results in appropriate data structure
E.g. for article entities

```
current_article_dict = {
     "title": title,
     "description": summary,
     "articleUrl": articleUrl,
     "imageUrl": imageUrl
     }
article_list.append(current_article_dict)
```

Database module:

Store newspaper and article information into database
Design and set up database
Design:
Requirements: For further expansion, many countries would be added to this project. Suitable newspapers are chosen based on survey. Analysed results are posed to frontend.
Entities: Accoding to requirements, what needs to be stored into database are country, newspaper, category and article entities.

ER Diagram:



Entities with attributes (primary key is underlined, foreign key is italic)
Country(country_id,country_name)
Newspaper(newspaper_id, url, name, country_id)
Category(basic_name, c_url,newspaper_id)
Articles(article_id,title,imageUrl,articleUrl,description,category_name)

Store data into tables

Based on development requirements, analysed results are stored into database, and frontend fetches them from database.
Database we use: SQLLite

Search module:
Getting articles and keywords to fetch related twitters and youtubes
According to given condition, search results from database
Condition: country and news category
Result: article details(title, image, summary etc)

Result module:
Using keywords to fetch related Tweets and YouTube videos.
Fetch tweets: using Twitter api
According to article keywords, fetch Tweets through Twitter python API tweepy.
e.g.

```
tweet=api.search(keyword,'en',count=5)
```

Fetch YouTube videos: YouTube API
According to article keywords, fetch Tweets through Twitter and YouTube API.

Front-end: User interface components

- ***Home page:***
  The home page displays the interactive map for selecting countries that the user wishes to browse articles from. We decided on 6 different countries to include in our application and each is highlighted on the map with a distinct color. All of the countries included are european and so the map is restricted to the european region. Other features on this page include a nav-bar displayed at the top of the page that informs the user of the current page he/she is on and also a title for the map was added. Pop-up boxes are also displayed showing information on the weather of selected countries e.g. current temperate, short summary.

- ***Articles page:***
  The articles page displays a scrollable list of article titles retrieved from the data base. Articles listed are arranged into four separate sections that organise the list according to article categories. Categories included are business, sports, politics and entertainment. Again a nav-bar is displayed at the top of the page and an additional nav-bar was included above the scroll box informing the user of the current category section being viewed. A selection of colours is also used to indicate the category of an article e.g. blue for business, green for entertainment.

- ***Summary page:***
  The summary page contains information related to the specific article selected by the user from the articles page. A short summary of the article is displayed to the user at the top of the page immediately under the nav-bar. The summary is presented through a scrollable box. Underneath the summary is an additional scroll box that displays trending tweets related to the selected article.
  Finally a slider is located at the bottom of the page that shows a selection of youtube videos related to the article. Videos will slide automatically to give a dynamic feel to the presentation of video content. All three sections are identified with a relevant title e.g. Article Summary, Twitter and also color is added to provide contrast to distinguish between sections e.g. blue alert boxes for tweets in Twitter section.
  In terms of the general design of the app many of the text boxes/navigational features are kept the same color (light grey) so as to create visual similarity between pages. We chose to keep the background white to provide effective contrast between features.

Example input/output and interaction with different components

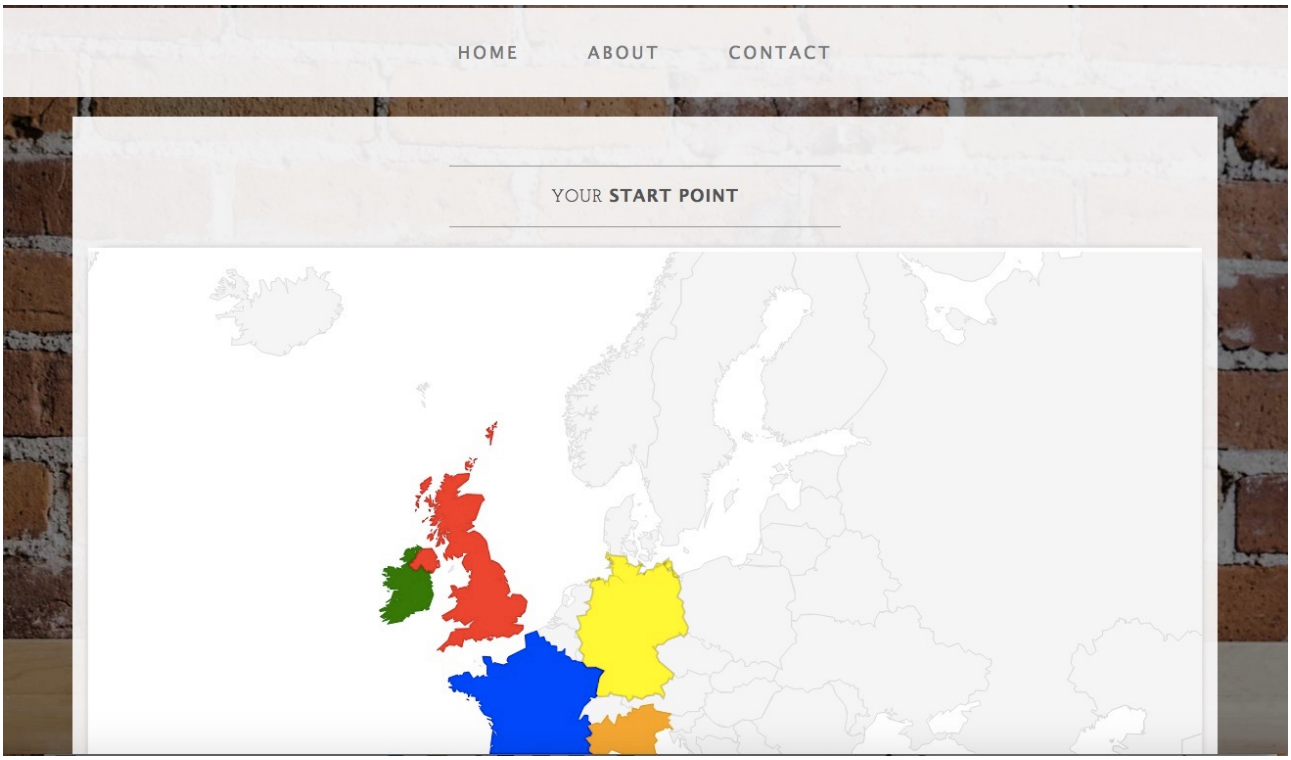


Figure 1

Figure 1 shows what is displayed to the user once he enters the site. A pop-up message is instantly brought to the foreground that contains instructions for the user.

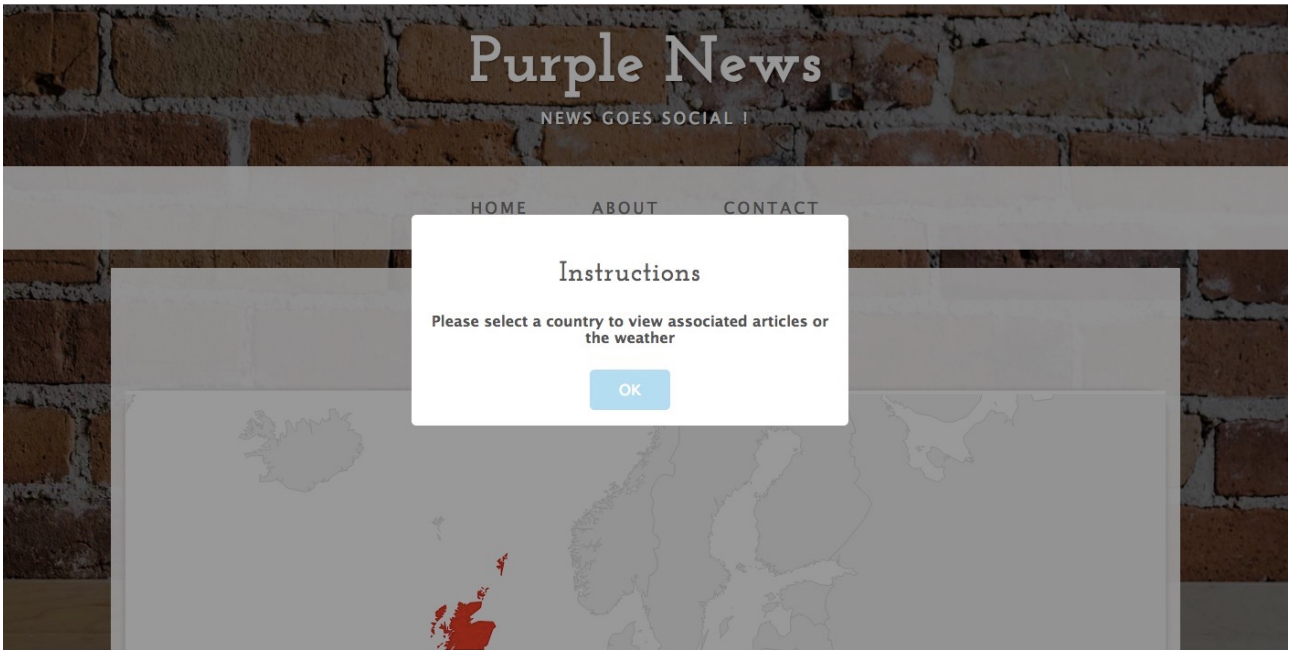


Figure 2

Once the user has read the instructions and closes the alert box he then proceeds to select a country from the interactive map ***(Figure 2)***. The user can only select countries which are highlighted with their unique color e.g. green for Ireland.

Selecting a country will lead to two of the following operations that the user is to choose from (Figure 3);

1. Display weather information for that country
2. Re-direct the user to the articles page where they can browse the list of articles retrieved from database.

Figure 3

If 'Weather' is selected a new pop-up message is displayed showing the weather forecast for the specific country ***(Figure 4)***.

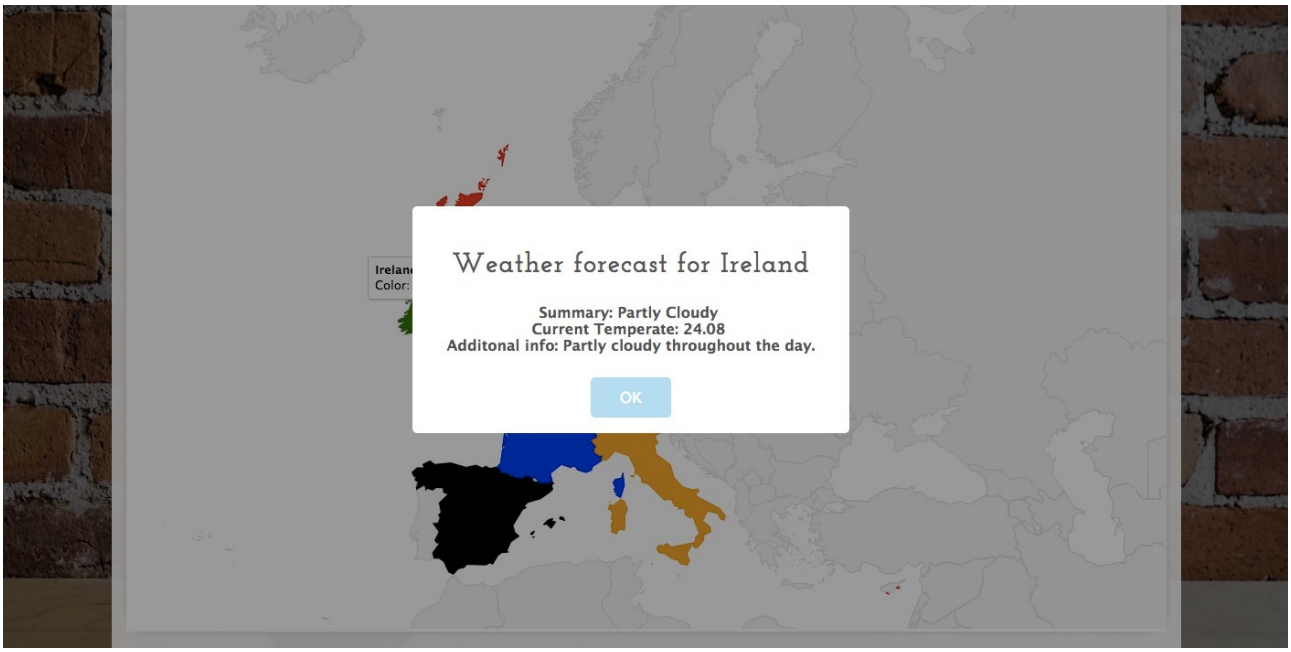


Figure 4

If the 'Articles' option is selected the user will be redirected to the articles page ***(Figure 5)***. Articles under the category 'Business' will be the first to be displayed and the user can then scroll down to explore articles from other categories.

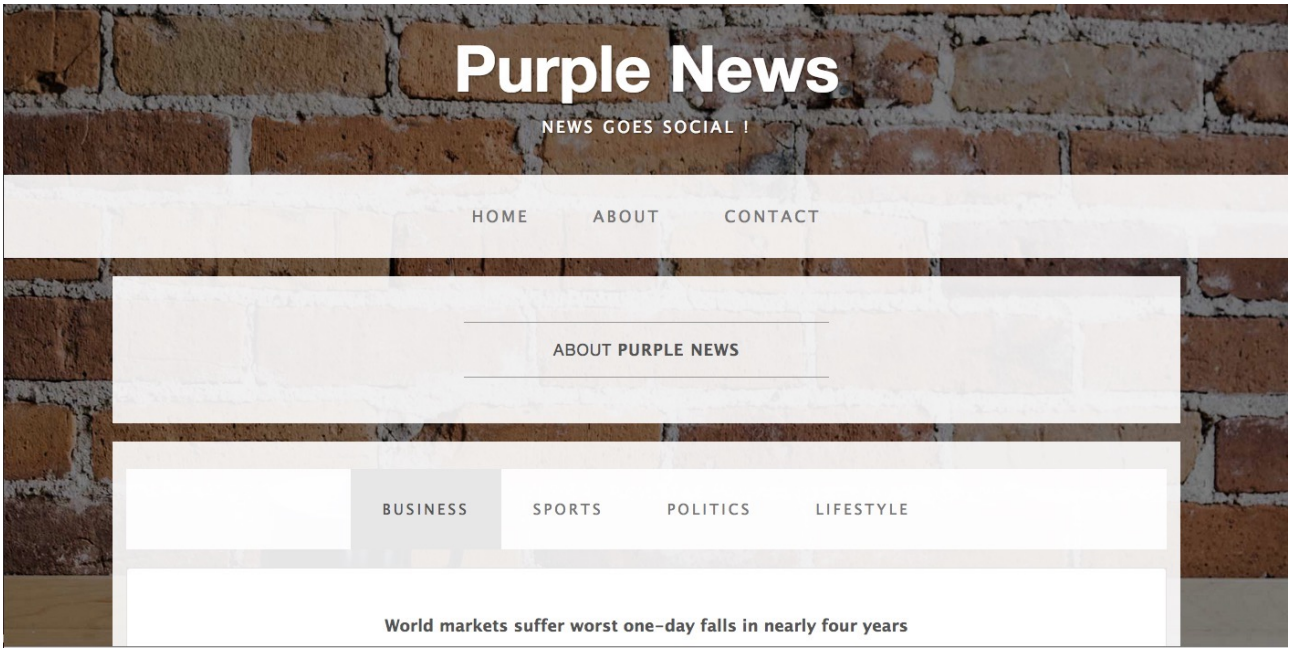


Figure 5

The 'Current category' nav-bar will inform the user of the category for visible articles e.g. 'Sports' ***(Figure 6)***, 'Politics' ***(Figure 7)***.

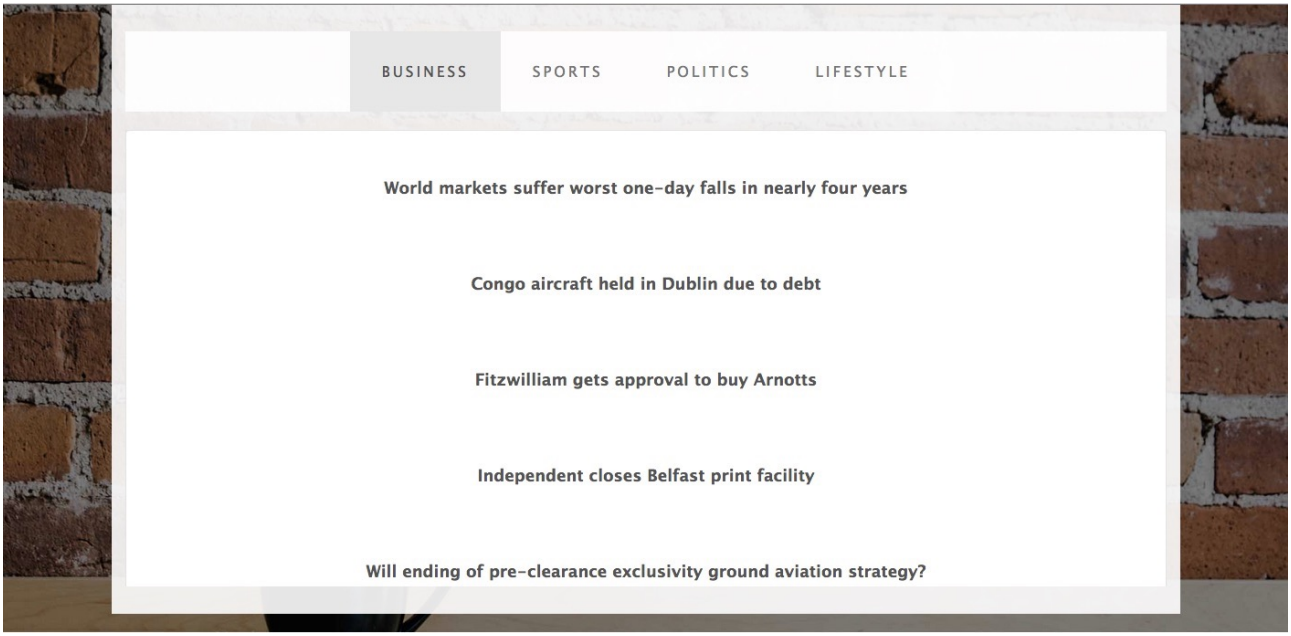


Figure 6

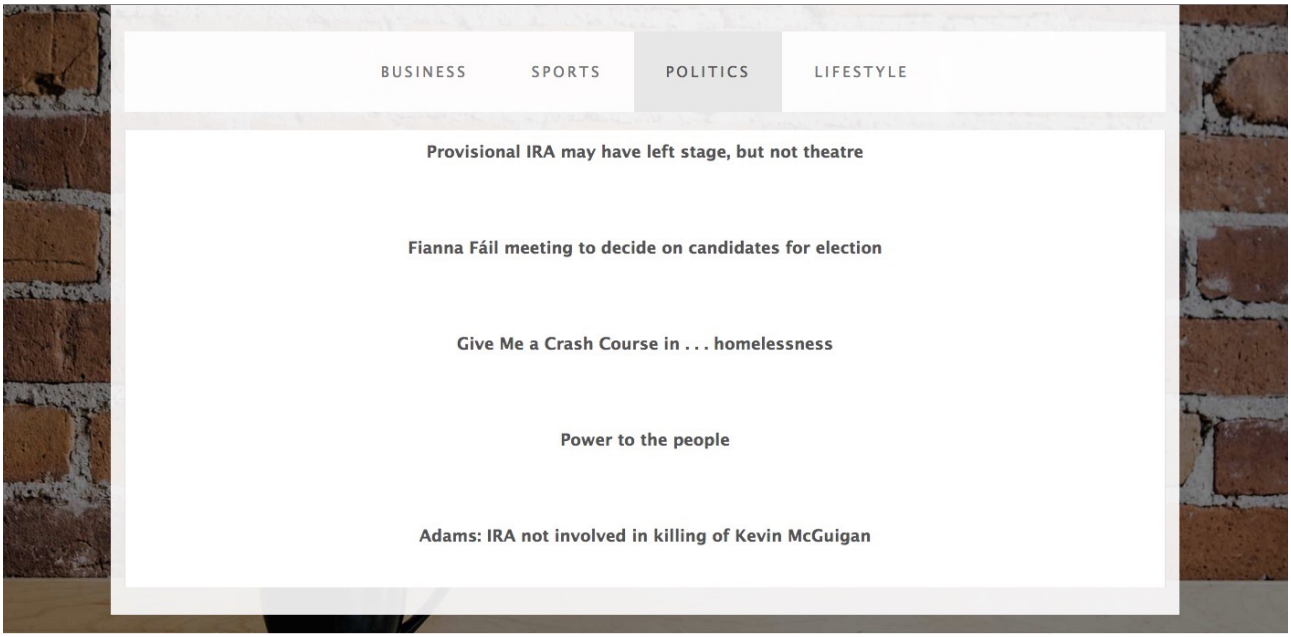


Figure 7

Once an article is selected (in this case 'World markets suffer worst one-day falls in nearly four years') the user is redirected to the summary page. The summary page displays a short summary for the article ***(Figure 8)***, relevant trending tweets ***(Figure 9)*** and relevant youtube videos ***(Figure 10)***.

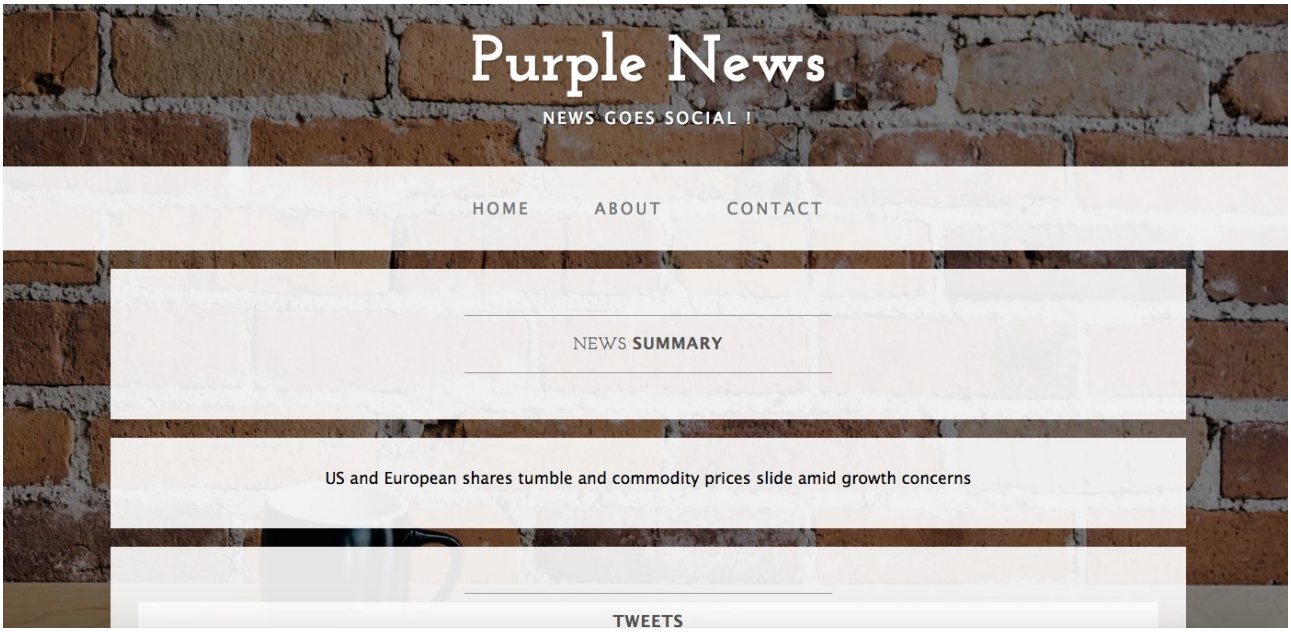


Figure 8

Figure 9


World Economy. Sep 17. Banks Suffer as Market Plunge.
0:17 / 1:57
CNN

Figure 10

4. Integration: The Sub-plots

The application uses the Tweepy to get the tweets, *urllib* Python package to get the YouTube videos, news API to retrieve the news from different newspaper and google charts for the map. To design the webpage, we use Django as the framework and HTML/CSS and Bootstrap to make the page much more beautiful and easy to read. In order to make the application run faster, we build databases to store the news, so it will be much easier to deliver the content to the new page so the application will load faster than without database. We use nltk to get the keywords from the news, then use these keywords to fetch the twitter and YouTube videos. Maybe sometimes the keywords may not work very well, so we will find a new method to get the right keywords then the accuracy of the twitter and YouTube videos related to the news will much better. Also we will give the weather forecast of the country using the Forecast.io API.
Our product delivers all kinds of news from different countries to the users, users can choose what they want to read and we provide tweets and YouTube videos related to the news, this make the news more comprehensive.

5. Impact: The Resolution

In 2014, there were 1.133 billion international tourist arrivals worldwide

The top 10 international tourism destinations in 2014 were:

Country	Number of visitors (in million)
France	83.7
United States	74.8
Spain	65.0
China	55.6
Italy	48.6
Turkey	39.8
Germany	33.0
United Kingdom	32.6
Russia	29.8
Mexico	29.1

From the data of **UNWTO**, 1.133 billion tourist arrivals worldwide and France, Germany, Spain, Italy and UK are in top 10 international tourism destinations, so our products will have big impact in these tourists, most of them will want to get the latest news of the country they visit.

And our product is very comprehensive it will attract many users to use our product. We will promote out product by twitter or Facebook then more people will know our product, if they think it is good, they will promote it to others, so more and more people will know our product, after that the impact will begin.

As it is a new product, first we will test our product, let the users to use the products and give us some advises. According to the review, we will improve the product to make it satisfy the users' needs. After the testing, the complete product will publish, then we will make it stable, as the society grow quickly, we need to learn the new things quickly and put the right things into our product, making our product more competitive, not be eliminated. Then it will move into mature stage. For the first step it may take 3 months and after 6 months our product will into stable then it will developing step by step.

6. Reflections: The Reviews

As compared to similar existing products, the application is making the user experience more simple by integrating a 2D map. The application was developed with simple and ease of access in mind and put the user at the top.

The application was built with a specific user group in mind, and since the success rate can be much higher than other existing offerings. Compared to the early stage prototype models, the application has become faster in rendering the pages as well as fetching the news content and translating the articles for countries with non-English news articles.

Before the technology stack where are used in developing the project is finalised, the team considered various technologies and debated whether the front-end had to be developed as a web interface or a mobile app. The final choice were made to develop the application as with a web interface using Python and Django with HTML5+CSS+Javascript and Bootstrap. The choice was made as all the team members were comfortable with the language as well when the application grows, Python is easy to build a solid backbone for scalable system. Since Python is already an established Object Oriented language with a wide variety of plugins to choose from, it helped the team to reduce the work overhead a lot simpler. Once such instance was to process the extracted article content and to summarise it along with keyword extraction. With existing Python packages such as *nltk* and *summarize*, this process was straight forward by consuming them and making fine adjustments as required.

When turning the idea of building the application to actual development, the team faced numerous road blocks.

One major challenge which the team still face is to find the Youtube videos relevant to the news article. The algorithm used in the project has be improved far from the initial phases, but still is facing some hiccups when fetching the videos using keywords.

Since the team went with a modular approach when developing the application, merging the modules together created a bit of trouble. As most of the team members were not experienced in using version control systems, the team had to work individually and test the piece of code before merging together. Hence, it took a couple of weeks to build a single point of access where most of the features are incorporated to run and test.

The team had a solid foundation with project management and software development. The team identified tools which can be put into use for effective management and dividing tasks between he team. By following Agile project management methodology, all members of the team were well aware of their tasks as well as the other member of the team.

During the initial phases of software development, prototype designs were made in the form of wireframes using pen and paper as well as online wire framing tools such as NinjaMock. This gave the team a clear idea on how the application should look like, identifying the core features, deconstructing the problem into independent modules so that the members can work without an issue of bottleneck.

When the product development started, the entire team was focused on software development and there was little thought give to developing the business model. But soon, the team realised this has to be given almost equal importance as to developing the application itself. Thinking about the user base and refining them into two categories helped to shape the features the application.

The team's perspective of giving almost all the focus to development has changed to delegating tasks within team members, multi-tasking and learning about taking ownership if independent modules and conforming to the timeline set during Scrum meetings.

7. References and key resources

Data sources:

The following websites were scraped for the news article contents and to extract related social media information;

http://www.irishtimes.com/
http://www.telegraph.co.uk/
http://elpais.com/
http://www.bild.de/
http://www.thelocal.fr/
http://www.corriere.it

Social media:

http://www.twitter.com
http://www.youtube.com
http://forecast.io

Software:

Framework: **Django 1.8.3**
Languages: **Python 3, Javascript, HTML5, CSS**
IDE: **PyCharm, Sublime Text, Atom**
Source Control: **Github**
Project Management: **Trello**
Communication: **Facebook Messenger, F2F meet**
Wire-framing and design: **Ninjamock, Adobe Photoshop**