



Bilkent University

Department of Computer Engineering

CS 492 - Senior Design Project II

Unisphere: Global University Catalog

Final Report

Arkın Yılmaz	21502080
Doruk Çakmakçı	21502293
Hakan Sarp Aydemir	21501331
İrem Ural	21502278
Umut Berk Bilgiç	21502757

Supervisor: Uğur Doğrusöz

Jury Members: Fazlı Can and Çiğdem Gündüz Demir

Final Report

May 9, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1.Introduction	4
1.1 Engineering Report and Writing Standards	5
1.2 Use of New Tools and Technologies	5
1.2.1 Development Tools and Frameworks	5
1.2.1.1 NodeJS	5
1.2.1.2 React	5
1.2.1.3 Material-UI	6
1.2.1.4 AWS DynamoDB	6
1.2.1.5 Github	6
1.2.1.6 Postman	6
1.2.1.7 Selenium	7
1.2.1.8 Google Compute Engine	7
1.2.2 Libraries and Resources	7
1.2.2.1 BeautifulSoup Library for Python	7
1.2.2.2 Google Images Download Library for Python	7
1.2.3 Application Programming Interfaces	8
1.2.3.1 Open Weather Map	8
1.2.3.2 EventBrite API	8
1.2.3.3 World Weather Online	8
1.2.3.4 Currency Converter	8
1.2.3.5 Google Maps API	8
1.2.3.6 Google Places API	9
1.2.3.7 Google Directions API	9
1.2.3.8 Google Geocoding API	9
1.2.3.9 Google Roads API	9
1.2.3.10 Google Translate API	9
1.2.3.11 Twitter API	9
1.2.3.12 Tweepy	10
1.2.3.13 NLTK	10
1.2.3.14 Langdetect	10
1.2.3.15 BOTO3	10
1.2.3.16 SWOT API	10
1.3 Algorithms	11
1.3.1 Web Scraping and Data Gathering	11

1.3.2 Sentiment Analysis	12
2. Final Architecture and Design	12
2.1 System Overview	12
2.1.1 Subsystem Decomposition	13
2.1.1.1 Client	15
2.1.1.2 Server	16
2.1.1.2.1 Logic Subsystem	17
2.1.1.2.2 Data Subsystem	18
3. Impact of Engineering Solutions	18
3.1 Global Impacts	18
3.2 Economical Impacts	19
3.3 Environmental Impacts	19
3.4 Social Impacts	19
3.5 Impact on Security	20
4. Contemporary Issues	20
4.1 Data Gathering, Comparison and Visualization	20
4.1.1 Language of The Application	21
Appendix	22
Appendix A - User Manual	22
Landing Page	22
Login Page	23
Register Page	23
Main Application Page	24
Main View	24
Search Field	24
Search Results	25
Main View, After Search	26
Academic Detail	26
Basic Detail	27
Comment Detail	28
Comment Enter	28
Comment Delete Confirmation	29
Event Detail	30
Event Detail, Routing	30
Recommendation Detail	31
Housing Options Detail	32
Sentiment Detail	33
Current Weather Detail	33
Historical Weather Detail	34
Map View	34

Map View, Routing	35
Street View	36
Split View	36
Split View, University on the Left	36
Split View, Select Side Dialog	37
Split View, Both Sides Populated	38
Academic Detail	38
Basic Detail	39
Comment Detail	39
Event Select Side Dialog	40
Housing Options Detail	41
Sentiment Detail	41
Current Weather Detail	42
Historical Weather Detail	43
Profile and Options	43
Profile Menu Main View	43
Profile Menu, Options Display	44
Priority List Main View	45
Priority List Edit View	45
Change Password View	46
Appendix B - Class Diagrams	47
Data Layer	47
Logic View	49
References	50

1. Introduction

Today in the era of Internet, people are looking for the easiest and fastest way to access information. Instead of looking for every website, they try to find the one, which has all the information they are looking for. Starting from the high school years, students start to search universities and they try to find enormous and variety of information such as the ranking of the university, academic programs, price of the dorms/universities and some location-based data, for example the weather conditions in that city, social events and facilities near the university, campus photos etc. Besides, all that raw information, they want to compare the universities in several ways. Therefore, by following this urge, instead of searching all the universities in different tabs and then trying to compare them by themselves or by considering for some narrow aspects, Unisphere will overcome and abandon this tiresome searching for hours to find the best universities that suit people well.

Though there are some web applications which has some features of Unisphere, this application will gather all that information and will provide some new features. Different from other applications, it will combine both academic and social aspects of universities. It will display all the routes from selected university to all social facilities such as museums, concerts, shopping malls, hospitals, airports, subways etc. Besides, it will inform users about the events near the university. To feel the atmosphere in the university, 360° tour in the campus will be offered and general public opinion about the university will be analyzed and presented visually to the users.

This report aims to outline the final software architecture and design of Unisphere, as well the impact of the project to the world. Afterwards, the contemporary issues related to the project will be explained. Lastly, the used technologies, libraries and frameworks during the lifecycle of the project will be stated.

1.1 Engineering Report and Writing Standards

In this report, UML standard is used for representing class interfaces, hardware-software components and decomposing system to packages. The references follow IEEE citation standard, which is commonly used .

1.2 Use of New Tools and Technologies

In this section, the tools and technologies as well as frameworks and libraries used in Unisphere will be explained.

1.2.1 Development Tools and Frameworks

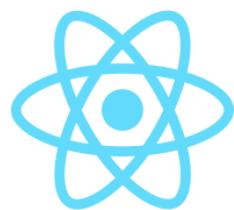
1.2.1.1 NodeJS

NodeJS is an open source, free to use and highly scalable Javascript runtime environment, which is built on top of Chrome's V8 JavaScript engine[1]. To build an interactive and responsive web application, NodeJS allowed us to build a software with Javascript both in client-side and server-side.



1.2.1.2 React

React is a stateful library for building dynamic interfaces. It utilizes virtual DOM technology in order to manipulate DOM's in an efficient manner, translating their structure and rendering them based on the state changes instantiated by other components, creating a dynamic web application.



[REACT]

1.2.1.3 Material-UI

Material-UI is a React framework that provides user interface components that are in-line with Google's Material Design design language guidelines. The design language consists of ordering based on elevation, material-like components such as paper, box shadows and glass. It also provides additional theming with MUI themes and overridable CSS.



1.2.1.4 AWS DynamoDB

Amazon provides a NO-SQL database service, which allows developers to store huge amount of data, and distributes the load/traffic with its auto-scaling functionalities[2].



1.2.1.5 Github

Github is version management control system, which allows developers to build a software together in an easy and efficient way[3].



1.2.1.6 Postman

Postman is a tool which enables developers to test requests to server[4]. In order to build a connection between client and server, Postman is used to test our requests to server-side.



1.2.1.7 Selenium

Selenium is a test-automation tool[5]. With the help of BeautifulSoup library for Python, Selenium is used to crawl data from the websites to build our university database.



1.2.1.8 Google Compute Engine

Google Cloud Platform provides Virtual Machines which are high performance and scalable and enables cloud computing. It allows customizable machines that can be adapted to developer's need [6].



Google Cloud Platform

1.2.2 Libraries and Resources

1.2.2.1 BeautifulSoup Library for Python

BeautifulSoup is a library for Python that is used for fetching the website content and it can be used for navigation to web elements[7]. In order to build our huge university database and to get universities academic information, tuition fee, historical information and logos of the university several websites are crawled by using BeautifulSoup.

1.2.2.2 Google Images Download Library for Python

GoogleImagesDownload is a library which allows developers to fetch images from Google into a directory, by using Google Search. By using keyword, according to the preference of the developer, the metadata as well as the images can be downloaded to local disk[8].

1.2.3 Application Programming Interfaces

1.2.3.1 Open Weather Map

OpenWeatherMap API supplies weather conditions with limited free access to users, more than 200,000 cities by using 40,000 weather stations[9]. Unisphere fetches current weather conditions of the cities.



1.2.3.2 EventBrite API

EventBrite gathers events around the world, by using EventBrite API Unisphere shows nearby events to universities on Google Maps[10].



1.2.3.3 World Weather Online

World Weather Online supplies historical weather data for several cities around the world starting from July 2008. By using its free tier functionalities, Unisphere accesses average temperature data for each city, and draws the content on a line graph[11].



1.2.3.4 Currency Converter

Currency Converter is an easy to use and easy to integrate API, which provides currency values and updates the data every hour[12].

1.2.3.5 Google Maps API

Google Maps Javascript API is a useful for customizing maps with your own content and imagery for display on web pages and mobile devices. It features different map types which



you can modify using layers and styles, controls and events, and various services and libraries. [13]

1.2.3.6 Google Places API

Google Places API is a service that returns information about places(establishments, geographic locations and prominent points of interest) using HTTP requests [14].

1.2.3.7 Google Directions API

Google Directions API is a service that calculates directions between locations[15].

1.2.3.8 Google Geocoding API

Google Geocoding API is a service that provides geocoding(street address to latitude and longitude) and reverse geocoding(latitude and longitude to street addresses)[16].

1.2.3.9 Google Roads API

The Roads API identifies the roads a vehicle was traveling along and provides additional metadata about those roads, such as speed limits[17].

1.2.3.10 Google Translate API

The Cloud Translation API can dynamically translate text between thousands of language pairs and lets websites and programs integrate with the translation service programmatically [18].

1.2.3.11 Twitter API

Twitter API is used to dynamically fetch tweets related to a string query. [19]

1.2.3.12 Tweepy

Tweepy is an easy-to-use Python library for accessing Twitter API.[20]

1.2.3.13 NLTK

NLTK is a platform for building Python programs to work with human language data. For this project, NLTK is used for text preprocessing (stop word removal, word tokenization etc.) and sentiment analysis(Vader sentiment analyzer[21]).[22]

1.2.3.14 Langdetect

A python library for language detection between 55 languages. [23]

1.2.3.15 BOTO3

Boto is the Amazon Web Services (AWS) SDK for Python. It enables Python developers to create, configure, and manage AWS services, such as EC2 and S3. Boto provides an easy to use, object-oriented API, as well as low-level access to AWS services [24].



1.2.3.16 SWOT API

SWOT API identifies email addresses that belong to colleges or universities. By using SWOT API, Unisphere allow students and academicians register to Unisphere [25].

1.3 Algorithms

1.3.1 Web Scraping and Data Gathering

To build our own database, we had to research a lot, and find the resources, in which we can fetch data from. These resources can be listed as following:

In order to create our base content for universities, the list of universities in the world, their contact information, historical information, departments are fetched from "<https://www.whed.net/>". To accomplish it, a crawler is written in Python by using BeautifulSoup module. Beautiful Soup library for Python allows us to extract data from HTML tags, however, WHED includes JavaScript in it. Therefore, in order to extract data, Selenium WebDriver is used to automate browsers. After running the crawler, fetching over 18,000 universities took a lot of time, almost 48 hours. As a result of this procedure, extracted data from the website was not formatted well and some of the informations were missing for most of the universities. We had to filter out some of the content.

To fill out the tuition fee field in our database, we tried to fetch data from Google Search by writing a crawler, however it does not work well as the page content differed a lot in each search and the necessary information could not be fetched as a result. Afterwards, we had to write another crawler to extract data from "<https://www.unipage.net/en/universities>" for tuition fees. In order to get the logos of the university, google_images_download library for Python is used, it allowed us to get image references, hyperlinks. Lastly, the university rankings are gathered by using CWUR ranking for 2018-2019 world university ranking. The subject rankings are gathered by using QS rankings available content for community.

1.3.2 Sentiment Analysis

Sentiment Analysis of tweets is done using vader sentiment analyzer of NLTK library of Python. However, vader sentiment analyzer works only for English texts. So in order to use vader sentiment analyzer as a multilingual sentiment analysis tool, the following pipeline is used: First hashtags, mentions and URLs are in the tweet is discarded. Then, the language of tweets is detected. Then, if the tweets are written in a language other than English, they are translated to English. Furthermore tweets are preprocessed to remove stop words. Finally, vader sentiment analyzer processes the tweet and produces positivity, negativity and neutrality of the input tweet in terms of percentage.

One problem we encountered while fetching tweets is Retweets. The retweets are eliminated using regular expressions.

2. Final Architecture and Design

2.1 System Overview

To be able to achieve the design goals of Unisphere, the system should be designed in a neat and proper way. The whole system has an enormous complexity and to overcome this chaos, the system can be decomposed into smaller chunks, which gathers similar functionalities together and collects the classes which has more associations with each other into one package. Unisphere sticks to those design goals and gives importance to subsystem decomposition. Unisphere decoupled the subsystems and tried to collect the classes which are relatively coherent with each other. Decomposition provides some abstraction and modularity, hence it will allow different teams to work simultaneously and optimize the implementation process. As each subsystem provides an interface to others, it eases the error revisions as well and minimizes the dependency of each team to others. It minimizes the communication between teams during implementation. Furthermore, in a well-defined subsystem hierarchy, when a change is applied, other subsystems did not affected by this change. As a result, these advantageous aspects of subsystem

decomposition are realized and used. In latter sections, the subsystems, their relations and their functionalities will be explained.

2.1.1 Subsystem Decomposition

Since Unisphere is a web application, its main functionalities are divided into two parts, the functionalities related to user interface and the functionalities related to data preservation and manipulation. In order to provide this separation Client/Server architecture will be used as a core subsystem decomposition. In this case, client requests services of server, while server hosts the data and makes calculations, manipulations using the data and provides those services to Client. The central database connection will be established in server side.

To separate pure data storage, from manipulation of these data, three tier architecture will be used. The client-side will include the presentation layer, view subsystem. The presentation layer is the top-most level of Unisphere. The main function of this layer is to provide an interface between the user and the application, which translates and propagates user's requests to the server side for request processing. Also, the data provided from the server side will be visualized in this layer. By requesting necessary data from server, it will display the information related to universities, show the events, facilities and routes to these in an attractive way. Furthermore, the server-side will be divided to two layers: Logic and Data layer. Data layer is the layer that the information related to universities and user's themselves are stored. The information will be propagated to Logic layer for processing and provides a connection between client and server. The Logic layer will be responsible for the calculations on the stored data.

Since the subsystem decomposition diagram of Unisphere is too large to show fully we have separated the subsystem contents and overall subsystem decomposition. The general subsystem decomposition of Unisphere is below:

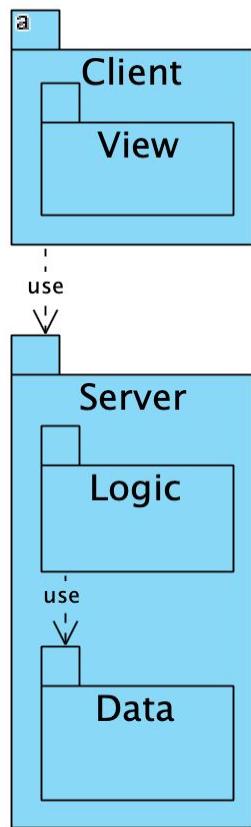


Figure 1. Subsystem Decomposition of Unisphere

2.1.1.1 Client

Client side of the system interacts with the end user, it is basically what the end user sees. When the user enters login information, chooses universities to compare, wants to view sentiment analysis or changes the settings on his/her account, then a request will be sent to the server.

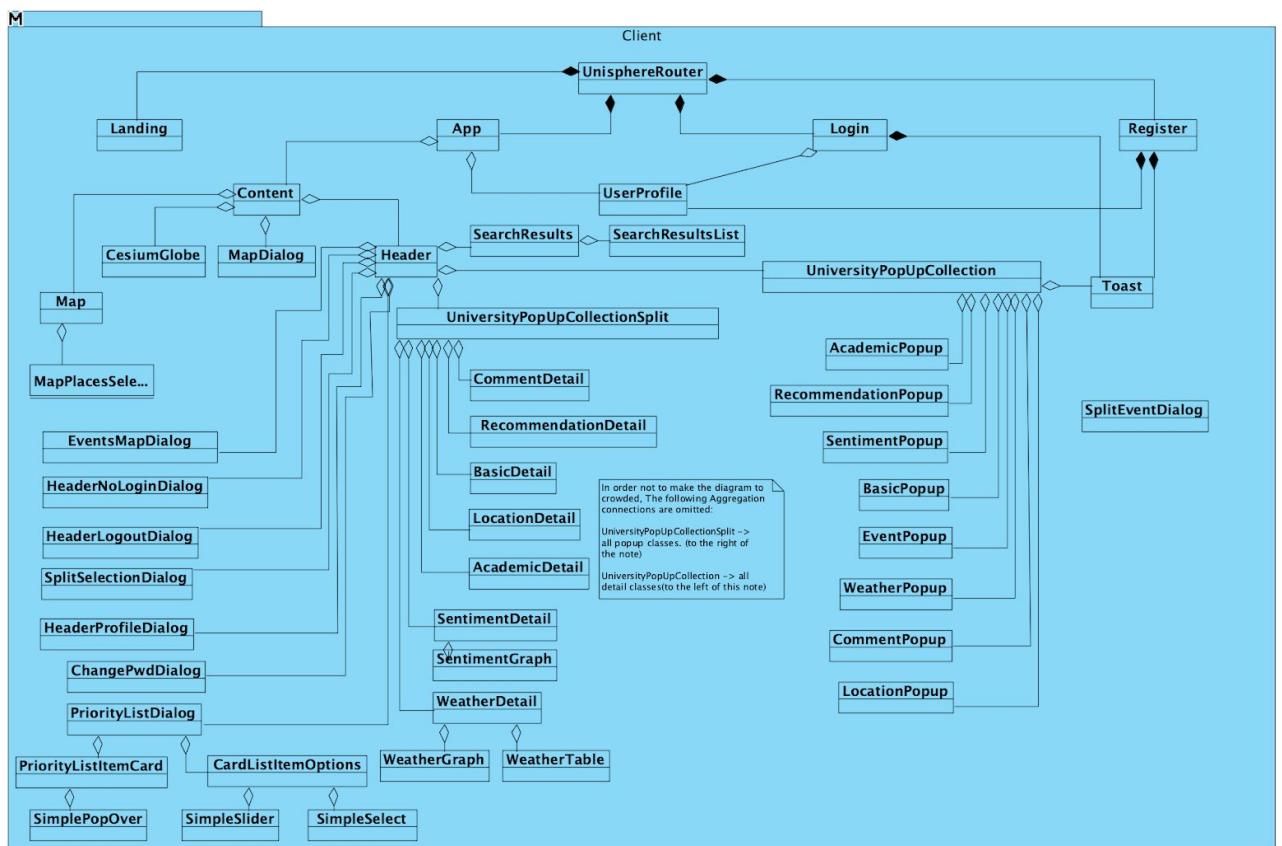


Figure 2.Client Subsystem

2.1.1.1.1 View

The main functionality of view subsystem is to interact with the user. The web pages, login page, user profile page, settings page, all those different views will be supplied by this subsystem. It interacts with the logic layer of the Server as well. As a design decision, in order to ease the implementation procedure and according to our

decision for implementation language and used frameworks, which are Javascript, ReactJS and Nodejs, it was better to combine controller and view subsystem into one presentation layer package. It allowed us to better handle the asynchronous operations.

2.1.1.2 Server

The crucial and main functionality of this subsystem is to store and manage data, including both persistent and dynamic data. As a database Amazon Web Services and its cloud infrastructure will be used. To separate raw data storage, from manipulation of these data, the server side is decomposed into two parts, ie. Logic and Data. Most of the user actions requires the connection to the server. Starting from the login events to each university search by user, each activity will require data manipulation or data fetching.

When user sign ups to the system, it will store user data and save it to the database. When user logins to the system, the credentials of the user will be checked by using user information in database. When user clicks to a university or searches a university, general information such as logo of the university, departments etc. will be provided by using database connection. These operations require an access to the database class in the Data layer. When user tries to access more information about a university, as well as location based informations, such as social facility, social event locations and weather conditions in that city, data manipulation is required . For instance, in order to show locations of the facilities, several routes from a university must be drawn as well. Hence in Logic subsystem, the raw data will be manipulated further to draw these routes in the client side. Besides, the general opinion about universities are displayed as well, in order to show these, the comments fetched from twitter and reddit will be analyzed in Logic layer.

In general, server subsystem is decomposed into two parts, Logic and Data. When user requests data, view package will interact with App class in Logic layer in the Server. Data subsystem is composed of fetchers (crawlers) for university information, logoFetcher, tweetFetcher etc. , as well as the Database class.

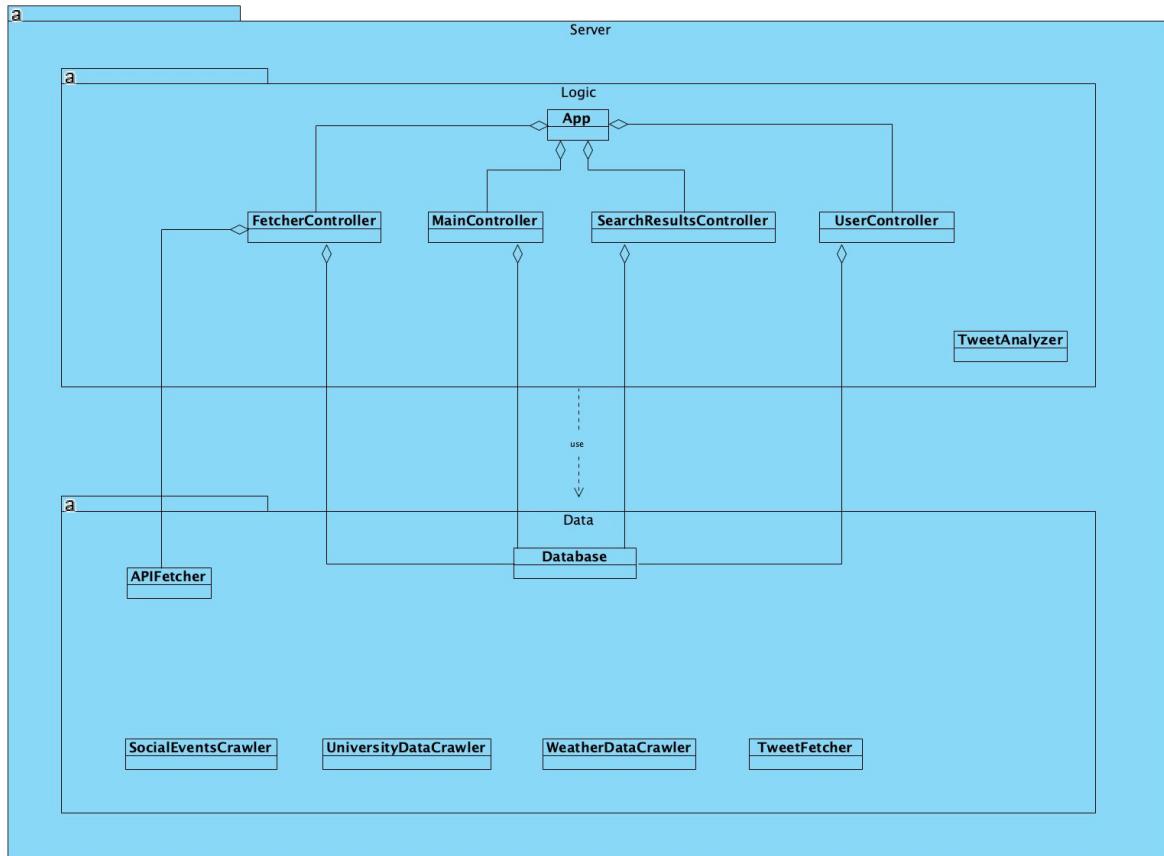


Figure 3. Server Subsystem

2.1.1.2.1 Logic Subsystem

The main functionality of Logic subsystem is to manipulate raw data, and provide it to the Client package.

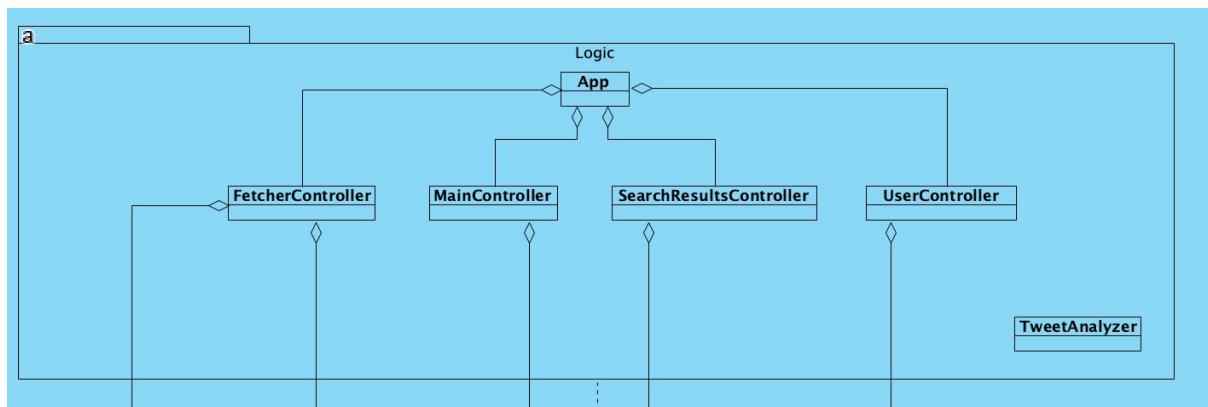


Figure 4: Logic Subsystem

2.1.1.2.2 Data Subsystem

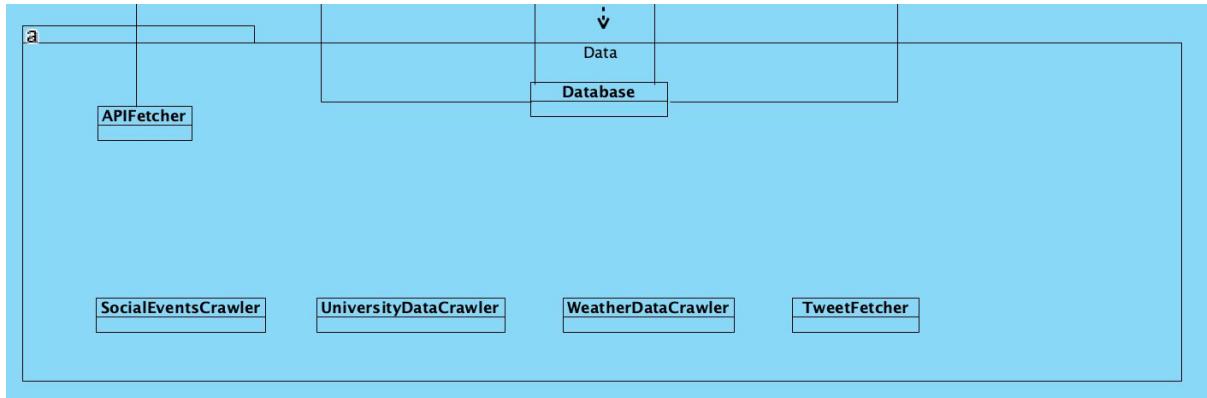


Figure 5: Data Subsystem

3. Impact of Engineering Solutions

Every developed software will affect the world in a specific way. During the implementation of Unisphere, the following impacts of the application are considered:

3.1 Global Impacts

Unisphere gathers, displays and visualizes information of universities all around the globe thus catering to academicians, fresh high school and college graduates looking to take their next step in their academic career to experienced postgraduates looking to finally solidify their profession in their area with a PhD programme. From its start, accessibility and ease-of-use was forefront when it came to the development stage of Unisphere. By reducing barriers of entry and providing an elegant and fluent design language, Unisphere aims to decrease the learning curve and provide the best UX it can. Additionally by creating an environment where users can share their opinions and thoughts about a subject, Unisphere aims to bring together a sense of community and connection around the globe.

3.2 Economical Impacts

The API endpoints and other services used by Unisphere tend to charge monthly proportional to the amount of usage:

- As Unisphere, deals with huge data, to lower the response time of the application, some paid functionalities of AWS DynamoDB is used.
- For the deployment and daily usage of the project, Google Compute Engine charges \$24 monthly.
- Google Translate API is used to fetch Twitter comments, in order to provide better sentiment analysis, we need to gather further Tweets and apply Natural Language Processing on these data, requires translation of these Tweets into English, hence it brings some cost with it. Before 500,000 character limit is reached, Twitter API does not require charges however after that it charges \$20 per million characters.
- Unisphere is free to use for everyone.
- Most of the API's and frameworks used in our application are free, such as EventBrite, OpenWeatherMap etc.
- For the “.net” domain of the website \$15 is paid to NameCheap.

3.3 Environmental Impacts

Unisphere is a web application, therefore it only requires an active Internet connection and a device that can run a browser with Javascript enabled. As a result, it does not bring any further environmental effect other than the inherent carbon footprint associated with hosting any web application (such as electricity usage of server, battery usage of user device etc.)

3.4 Social Impacts

As Unisphere ease the searching process for a university, by its simplistic yet meaningful UI components, it can attract users and motivate them to further studies. As users can access detailed information about universities, including academic,

social and location based information, they can make decisions easily and accurately. Additionally, users find it enjoyable to mess around on the globe interface and check out the campus.

3.5 Impact on Security

Unisphere doesn't keep any data about the usage habits of the users, it just saves the account data and doesn't share this data with other parties. The account data is kept in a DynamoDB system which is a secure system maintained by Amazon.

4. Contemporary Issues

In this section; some of the issues which we have faced during development is discussed. These issues were difficult in terms of design choices and each decision made brought along a trade-off.

4.1 Data Gathering, Comparison and Visualization

In the internet, it is observed that there is no unified source for academic and social information about the universities. Since the data is distributed to many different sources, it requires an exhaustive search. Additionally, some of the available data about the universities are represented in text format, which is hard to understand or not informative at all. Therefore, the user interface of Unisphere is designed in a way where users can easily see the desired information in a compact manner. It also allow users to compare universities by providing an attractive split view representation. Moreover, Unisphere used charts and tables when needed to create more informative representation of the data.

4.1.1 Language of The Application

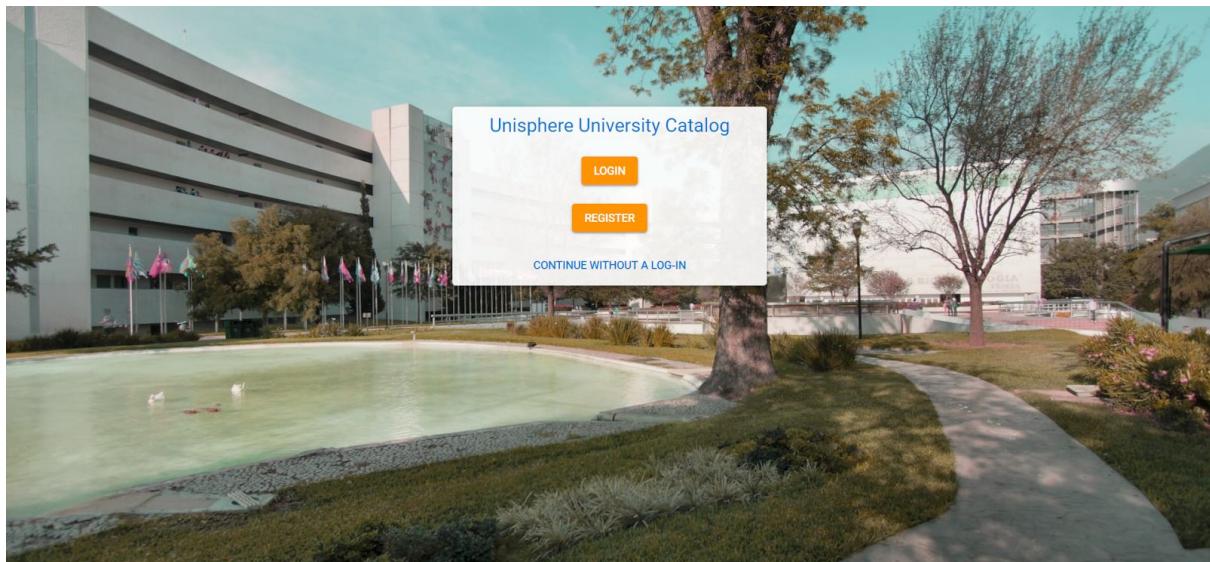
The main objective of Unisphere is to gather various information related to universities all around the world. Accordingly, the target audience of the application are students and academicians worldwide. To create an effective communication environment and allow users to be able to use easily, Unisphere is implemented in English. However, some of the information related to nearby social events are still in local language.

Appendix

Appendix A - User Manual

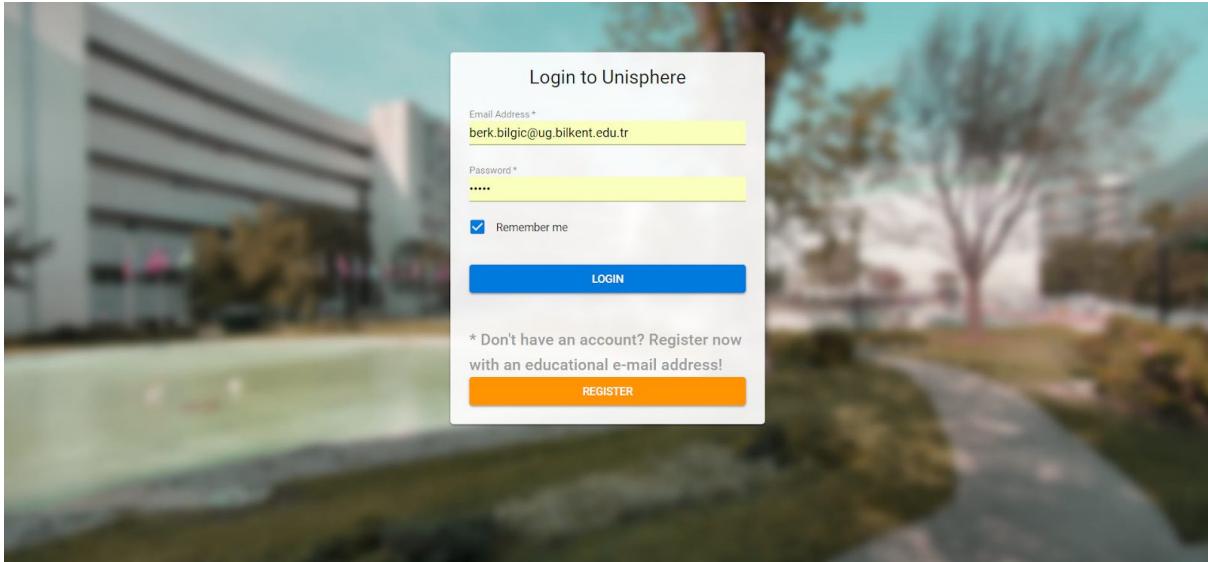
Unisphere is a web-application hosted on a remote server and thus needs to special installation process. Anyone can visit <http://www.uni-sphere.net> on a browser with modern JavaScript enabled browser can see the application in action. The rest of this section aims to serve the purpose of guiding and outlining the flow and experience of Unisphere by using screenshots from the live website.

Landing Page



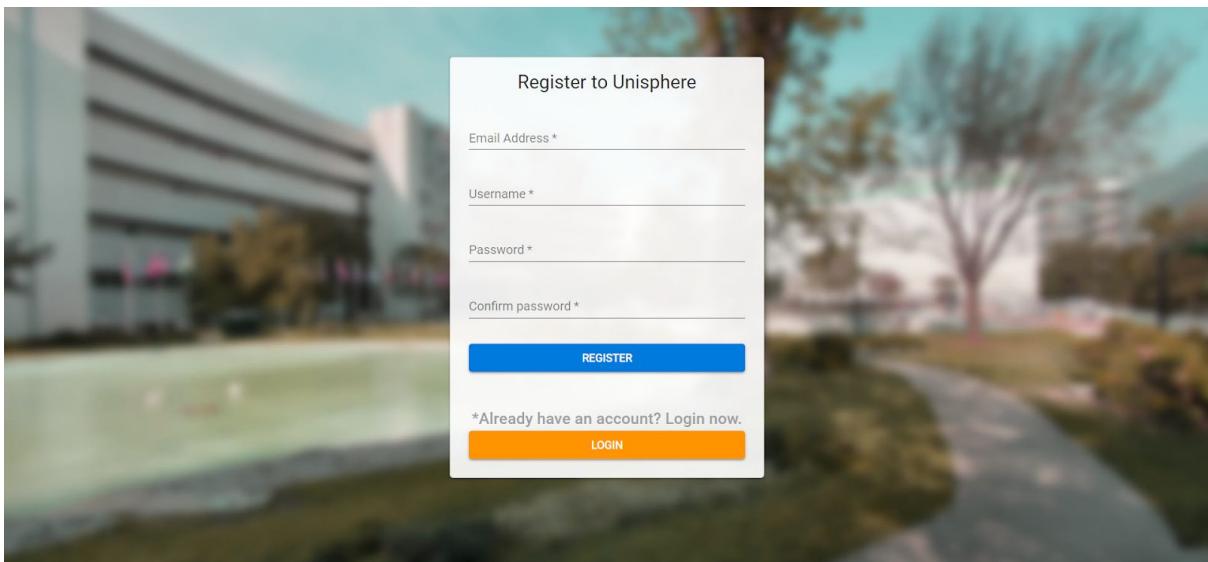
When a user visits [uni-sphere.net](http://www.uni-sphere.net) they are greeted with the landing page which provides clear navigational components: Login button, Register button and link that allows users to continue to the main application page without logging in. User who do not log in cannot get recommendations since they don't have a priority list set and cannot view and enter comments.

Login Page



Users login with their accounts in the login page. When the remember me button is set, the browser will store the login information as cookies so that the next time the user visits the login page they are redirected to the main application page.

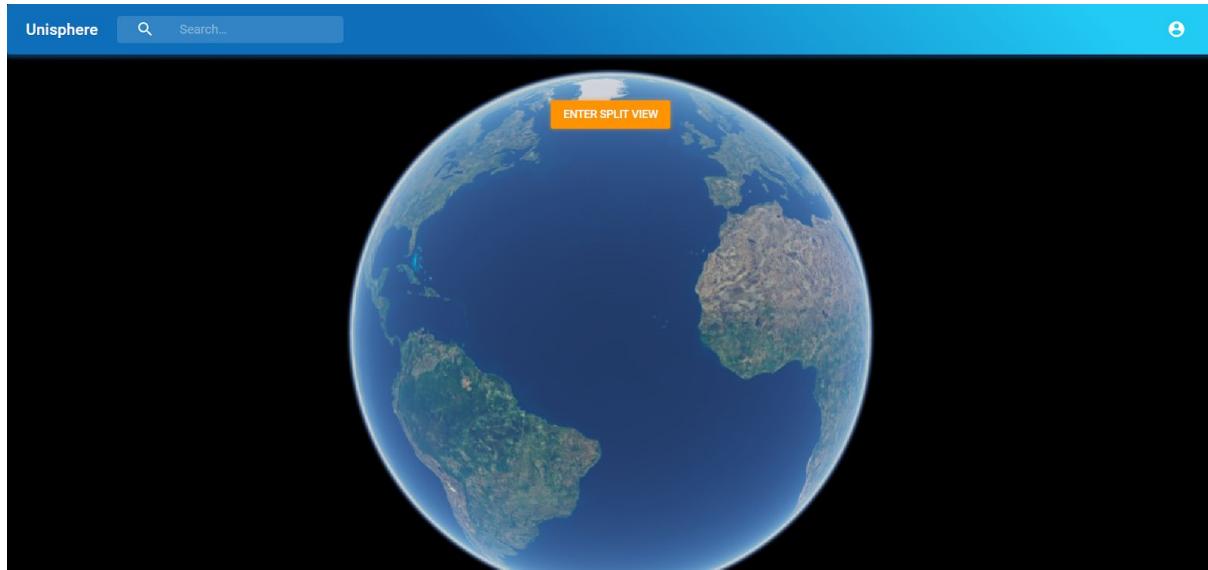
Register Page



Users are allowed to register to Unisphere only with an email address that is of a recognized academic institution. After registering, the cookies on the browser is set and the user is redirected to the main application page.

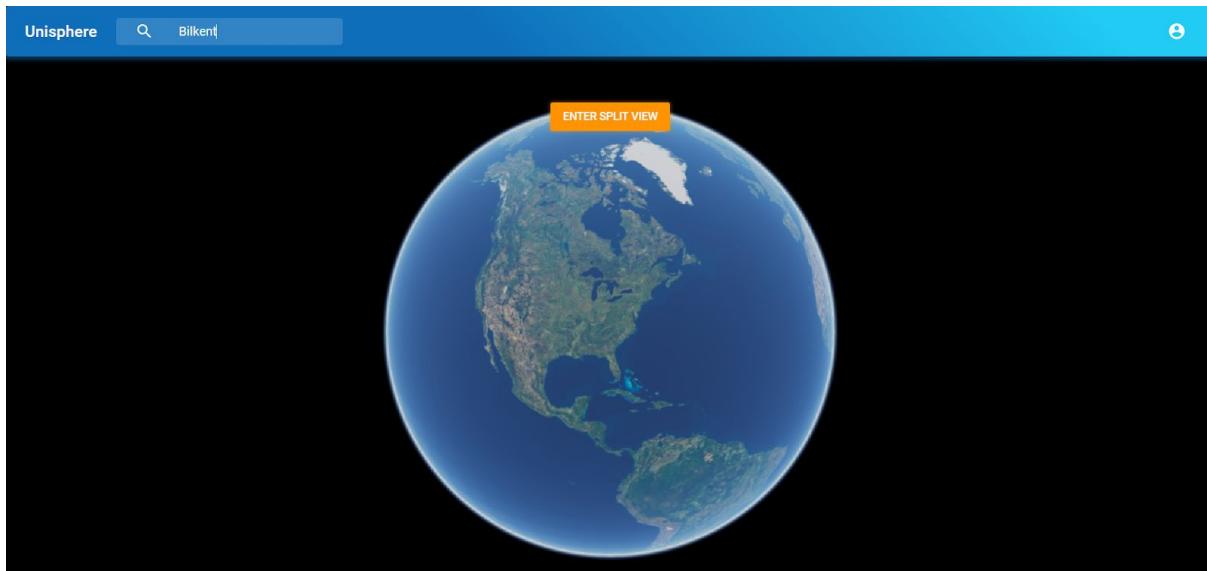
Main Application Page

Main View



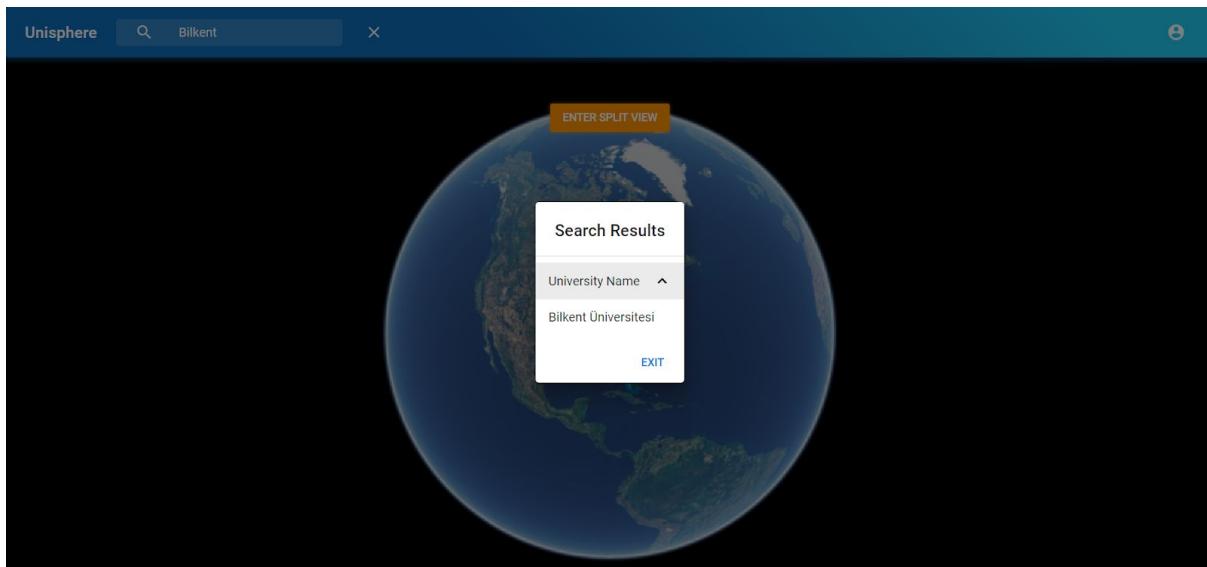
This is the main application page. When users first successfully log in or enter the application without logging in, they see the globe and the header with search bar and profile icon button.

Search Field



Users can enter a university name to the search field and hit their enter key in order to search for that university on Unisphere's database.

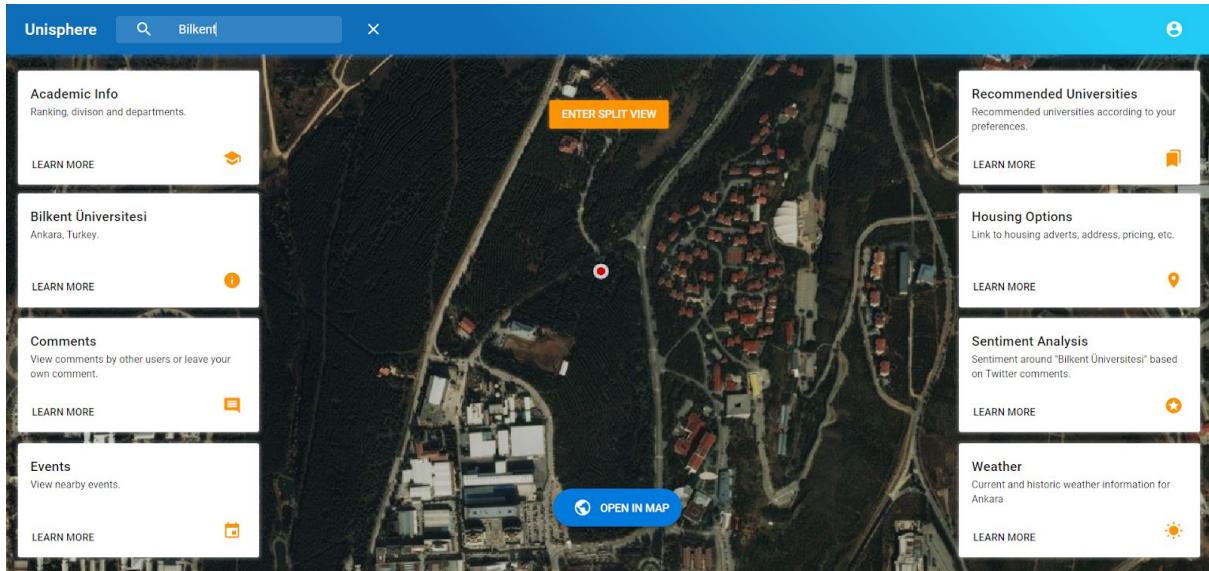
Search Results



After the search is complete and the results have returned via a POST request response from the server, the results are displayed under the

search results list as list items. When a result is clicked, the globe will slowly move the camera to the selected university's campus.

Main View, After Search



After the globe animation is done, eight pop-ups appear with a growing animation. These pop-ups allow users to access the following detail views; academic information, basic information, comments, events, recommended universities based on the priority list, housing options, sentiment analysis of Tweets about the university, historical and current weather information for the city the university resides in.

Academic Detail

The screenshot shows the Unisphere interface for Bilkent University. At the top, there's a search bar with 'Bilkent' and a magnifying glass icon. Below the search bar, there's a 'Recommended Universities' section with a small map thumbnail. The main content area features a large satellite map of the university campus. Overlaid on the map is a white box containing the university's logo (a red circle with 'Ihsan Doğramacı' and 'Bilkent Üniverstesi') and the text 'Academic information for Bilkent Üniversitesi'. Below this, there's a detailed description of the university's history and admission requirements, along with a ranking of 456. To the left of the map, there are sections for 'Academic Info', 'Comments', and 'Events'. To the right, there are sections for 'Housing Options', 'Sentiment Analysis', and 'Weather'. A blue button at the bottom of the central box says 'OPEN IN MAP'.

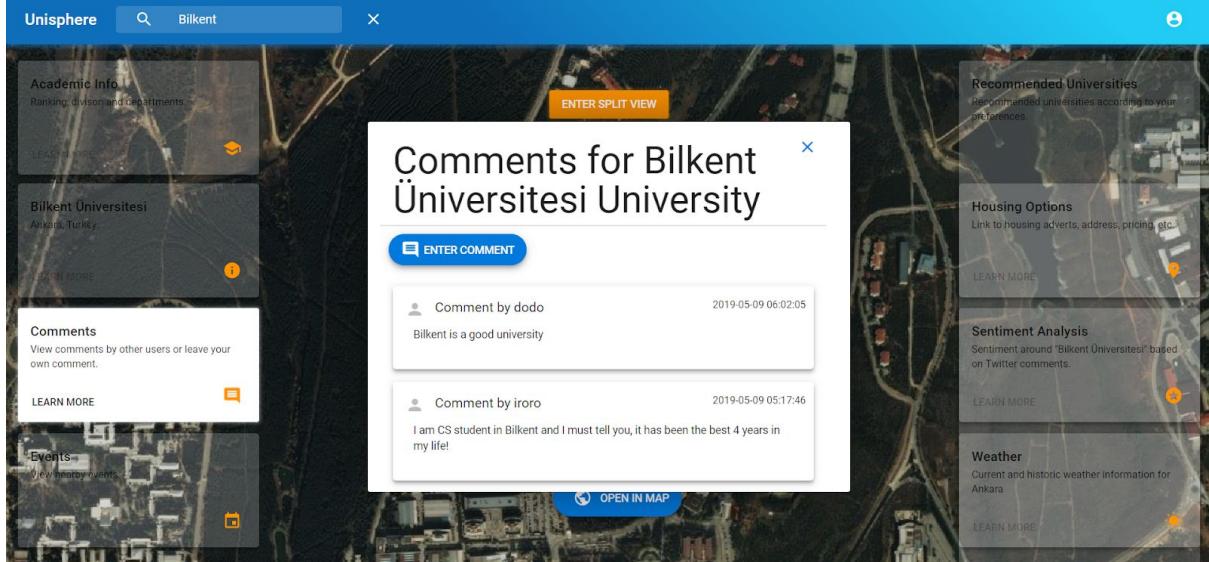
The academic information for a university is displayed in the academic detail view. This information includes but not limited to: agency, university history, general global academic ranking, available departments and fields of study, education language and available degrees.

Basic Detail

This screenshot shows the Unisphere interface for Bilkent University, similar to the previous one but with less detailed academic information. The main content area features a white box with the university's logo and the text 'General information for Bilkent Üniversitesi'. Below this, there's a brief description of the university's location (Ankara, Turkey), postal code (06800), contact phone number (+90(312) 290-40-00), and heads of the institution (Abdullah Atalar as Head, Kürsat Aydogan as Senior Administrative Officer, and Secretary General). A blue button at the bottom of the central box says 'OPEN IN MAP'.

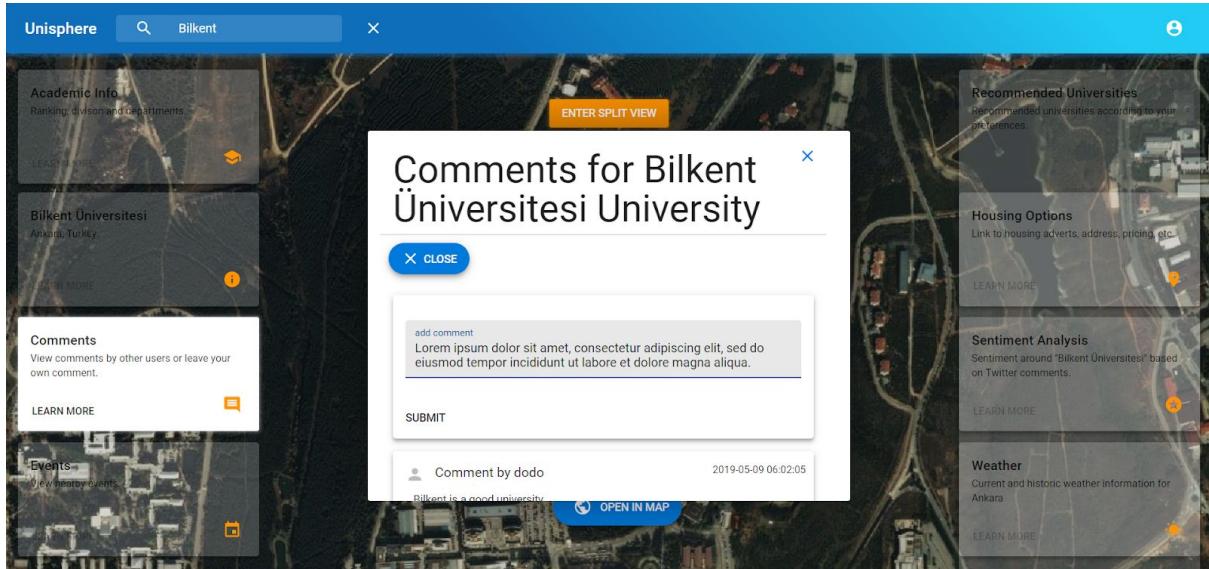
The basic and contact information for a university is displayed in the basic detail view. This information includes but not limited to: country, city, postal code, contact phone number, rector, general or any other high ranking individuals and the university link.

Comment Detail



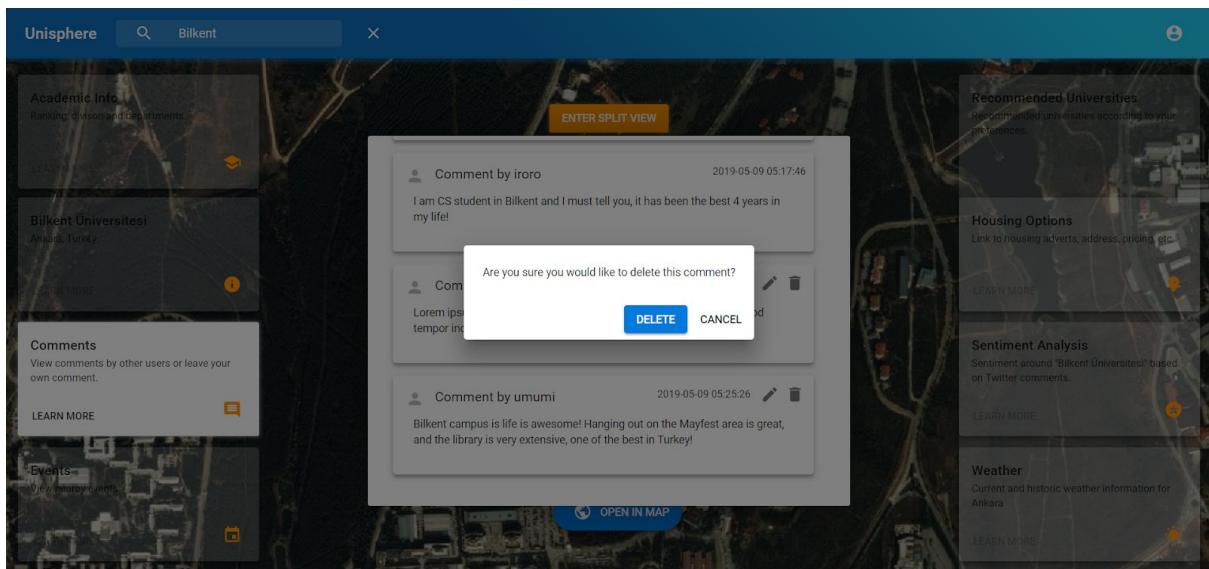
The comments entered on a university is available in the comment view. Here, users can enter comments, view comments made by themselves and others or edit and delete their own comments. Comments also display the time in which they were entered.

Comment Enter



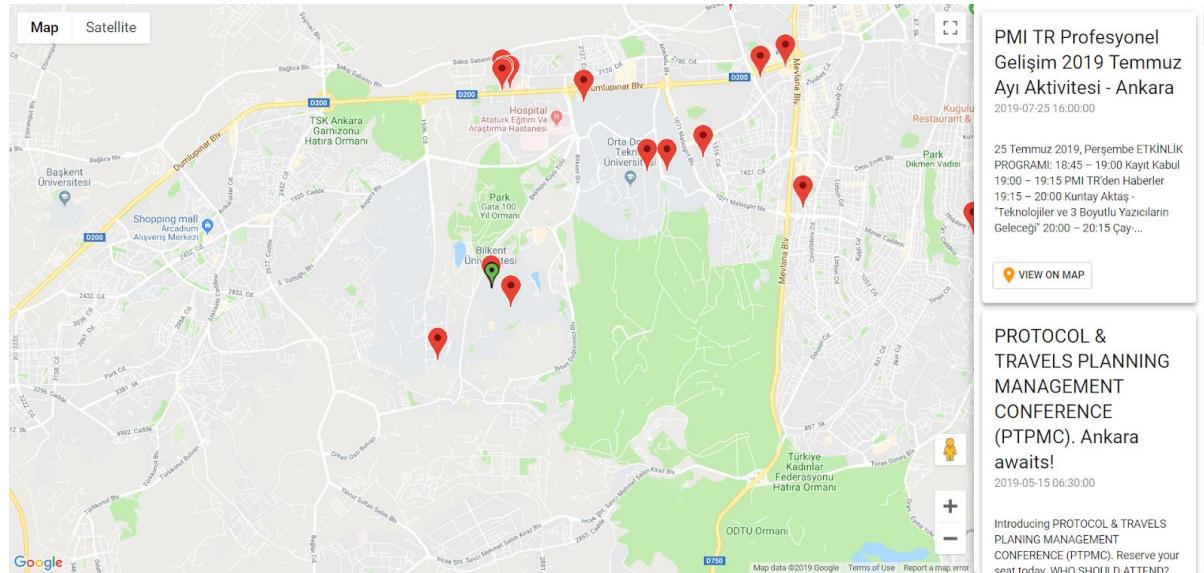
When a comment is entered and submitted, it is sent off to the server and the comment detail view is automatically refreshed to displayed the new comment.

Comment Delete Confirmation



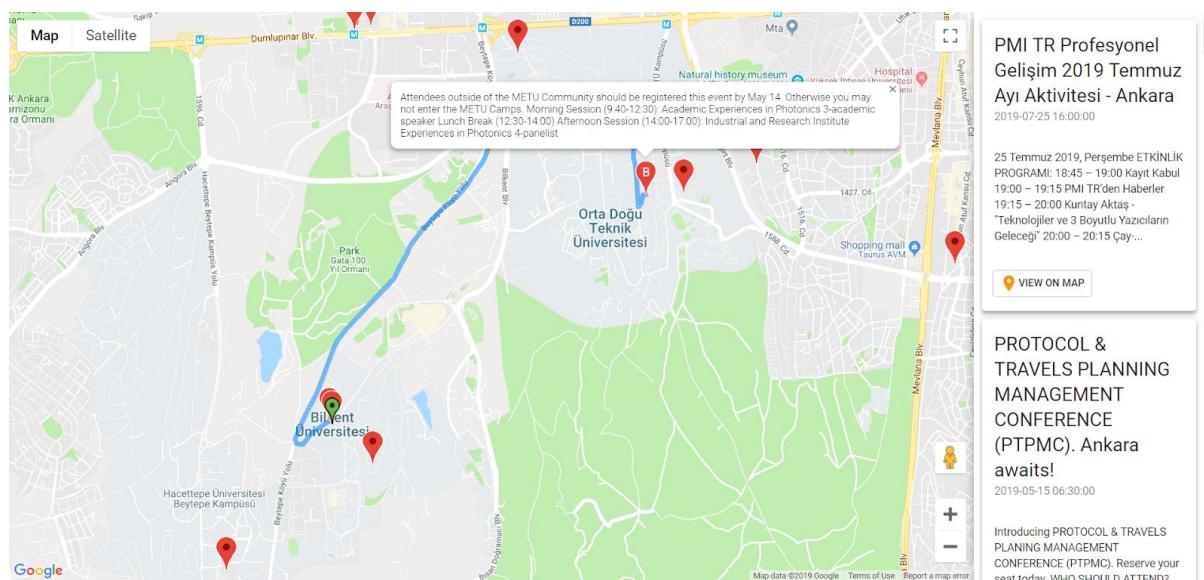
After deleting or editing comments, the comment detail view is refreshed in order to reflect the changes to the comments for the university.

Event Detail



Nearby events for a given university is shown to the user when they click the “Learn More” button of the event pop up. Upon click, a map view provided by Google is displayed with pin drops indicating the locations of the events. The users can scroll through the scroll cards on the right to gain more information about an event and see exactly when it will be held.

Event Detail, Routing



When a pin drop on the map is clicked, Unisphere will calculate a route between the university center and and the selected event. Users can go back to Unisphere main application view by clicking the “Escape” key on their keyboards.

Recommendation Detail

Recommended Universities
Calculating recommendations may take some time.

- List of Universities According to Your Preferences
- List of Universities According to Your First Four Preferences
- List of Universities According to Your First Three Preferences
- List of Universities According to Your First Two Preferences
- List of Universities According to Your First Preference

Recommended Universities
Recommended universities according to your preferences.

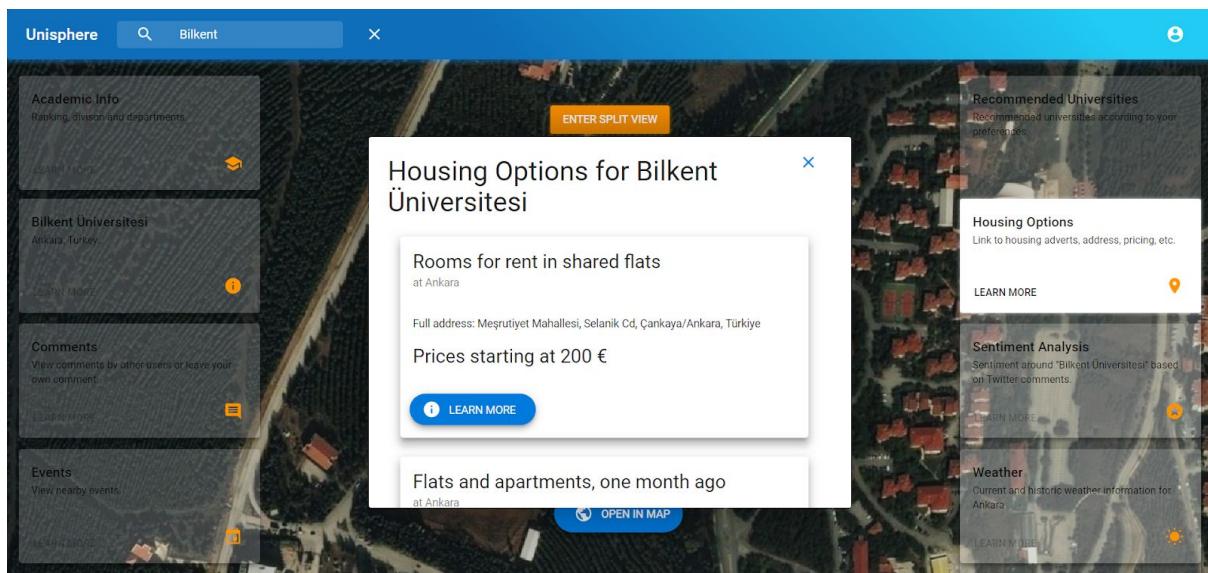
Housing Options
Link to housing ads/info, address, pricing, etc.

Sentiment Analysis
Sentiment around "Bilkent Üniversitesi" based on Twitter comments.

Weather
Current and historic weather information for Ankara.

Unisphere calculates recommendations based on the priority list set by the user on their profile settings page. The recommendation detail view includes five list item buttons. The first button tries to calculate and retrieve universities that are exactly compliant to the user's priority list. If the users is not satisfied with the results or no matching university is found, they can try the button below the one they just tried in order to omit one element of their priority list. Doing so will result in a relaxation of constraints on the calculation of recommendation, thus resulting in a bigger list of universities, with the cost of the calculation taking more time and trading off one priority list element.

Housing Options Detail



Housing options detail view includes housing options in the city that the selected university is in. The individual housing option cards include the description, city, full address, price and a link to the original advert.

Sentiment Detail

The screenshot shows the Unisphere platform interface for Bilkent University. The main content area displays a pie chart titled "Sentiment analysis for Bilkent Üniversitesi". The chart shows the distribution of sentiments: Positive (49%), Neutral (31%), and Negative (20%). Below the chart is a button labeled "OPEN IN MAP". To the left of the chart, there is a sidebar with sections for "Academic Info", "Comments", and "Events". To the right, there are sections for "Recommended Universities", "Housing Options", "Sentiment Analysis", and "Weather". A map of the university campus is visible in the background.

The sentiment and public consensus around a given university is regularly checked by Unisphere using Twitter API. The sentiment results are aggregated into three categories: Positive, Negative or Neutral.

Current Weather Detail

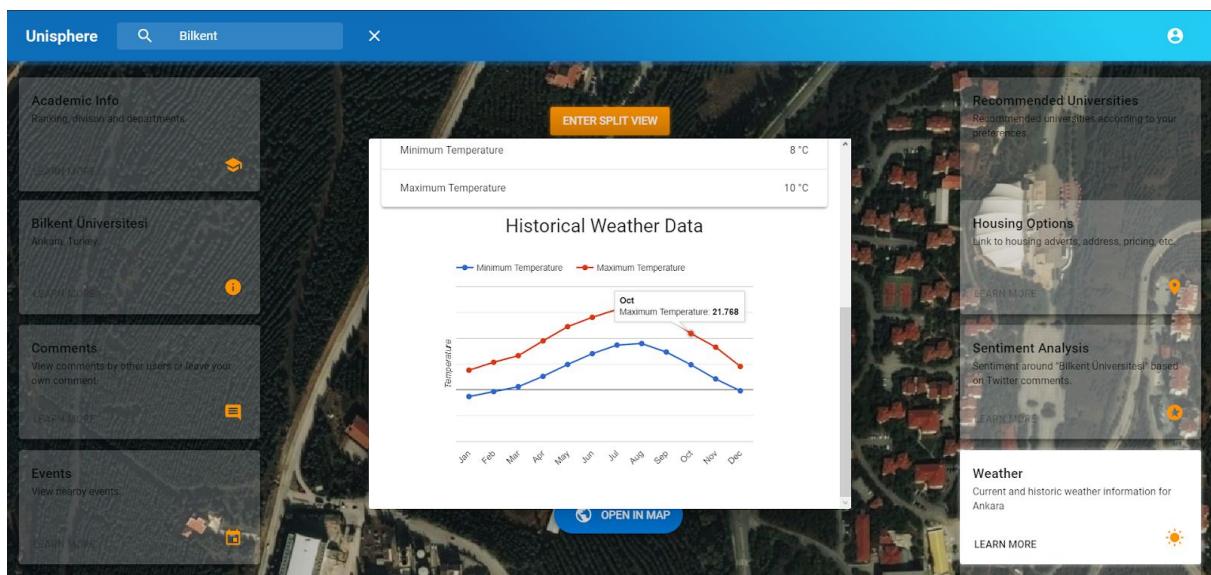
The screenshot shows the Unisphere platform interface for Ankara. The main content area displays a weather summary for Ankara: "9 °C" and "broken clouds". Below this is a table titled "Current Weather Conditions" with the following data:

Description	Value
Humidity	75 %
Pressure	1015 hpa
Minimum Temperature	8 °C
Maximum Temperature	10 °C

Below the table is a button labeled "OPEN IN MAP". To the left of the table, there is a sidebar with sections for "Academic Info", "Comments", and "Events". To the right, there are sections for "Recommended Universities", "Housing Options", "Sentiment Analysis", and "Weather". A map of the city is visible in the background.

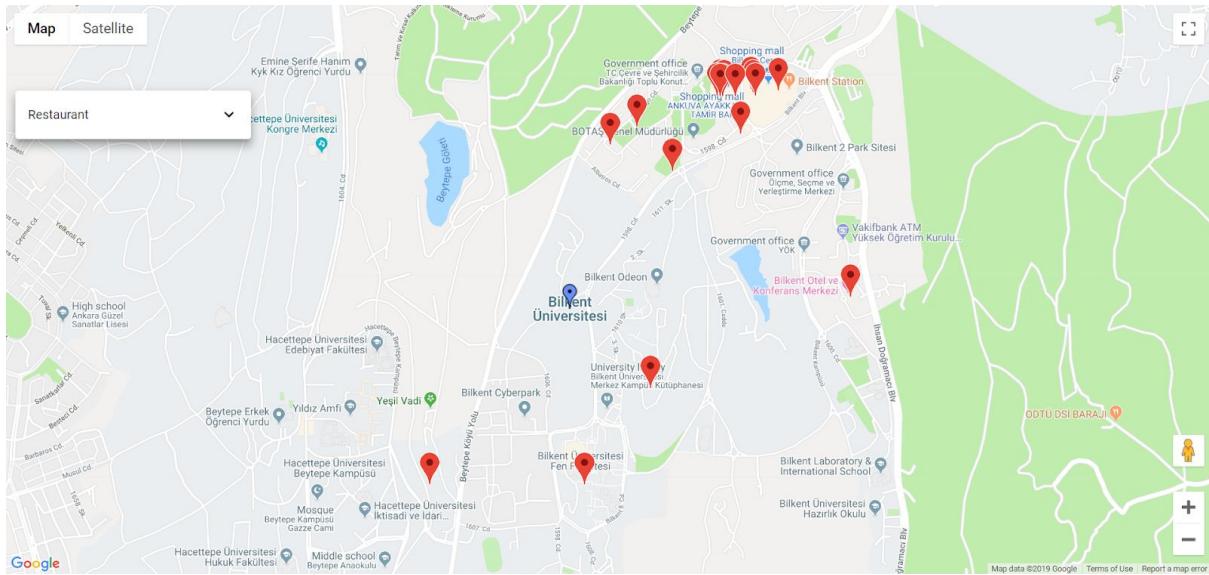
Users can see the current weather for the city that the selected university is at in the weather detail view. The current weather data is not fetched on a regular basis but instead requested when a user asks for it. The current weather table includes the current weather in Centigrade, an icon to indicate the condition, description, humidity, pressure, and minimum and maximum temperatures forecasted for the day.

Historical Weather Detail



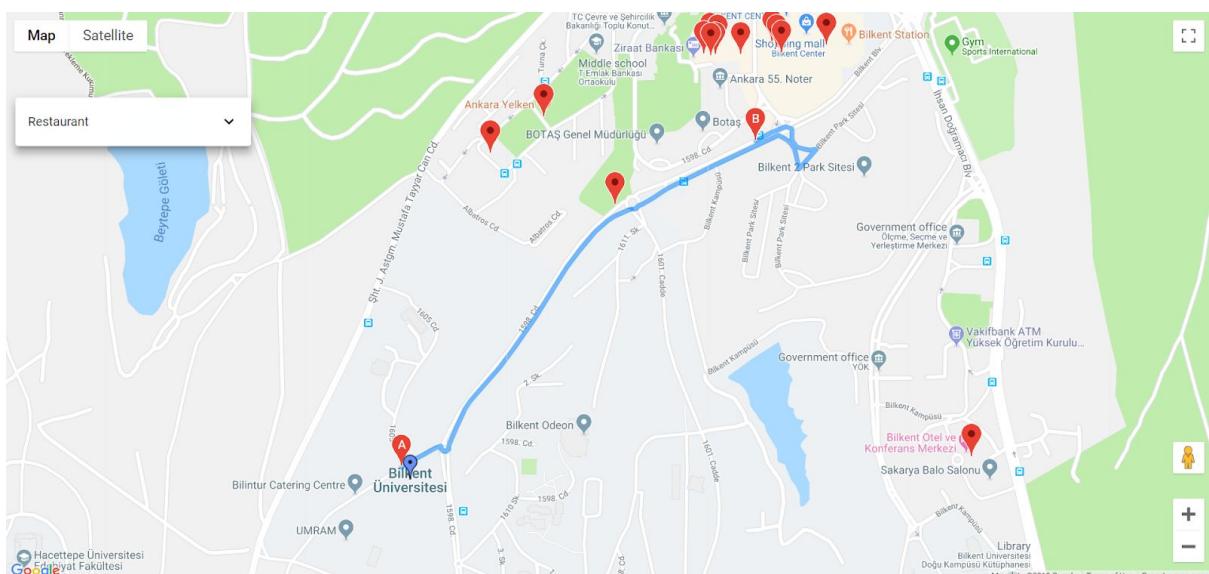
Users can see a historical weather line graph indicating how the minimum and maximum temperatures changed throughout the past year. This is especially useful for a user that might not want to be somewhere very cold or too hot, or for a user that especially prefers colder cities.

Map View



The map view fab (stands for floating action button, documented as a button that provides a single and important functionality to a page) is always present whenever there is a current university present in the app. In the map users can select from a wide variety of social facilities such as restaurants, night clubs, hotels, hospitals, city halls and much more.

Map View, Routing



By clicking on any of the facility pins a route to that facility from the selected university is drawn on the map.

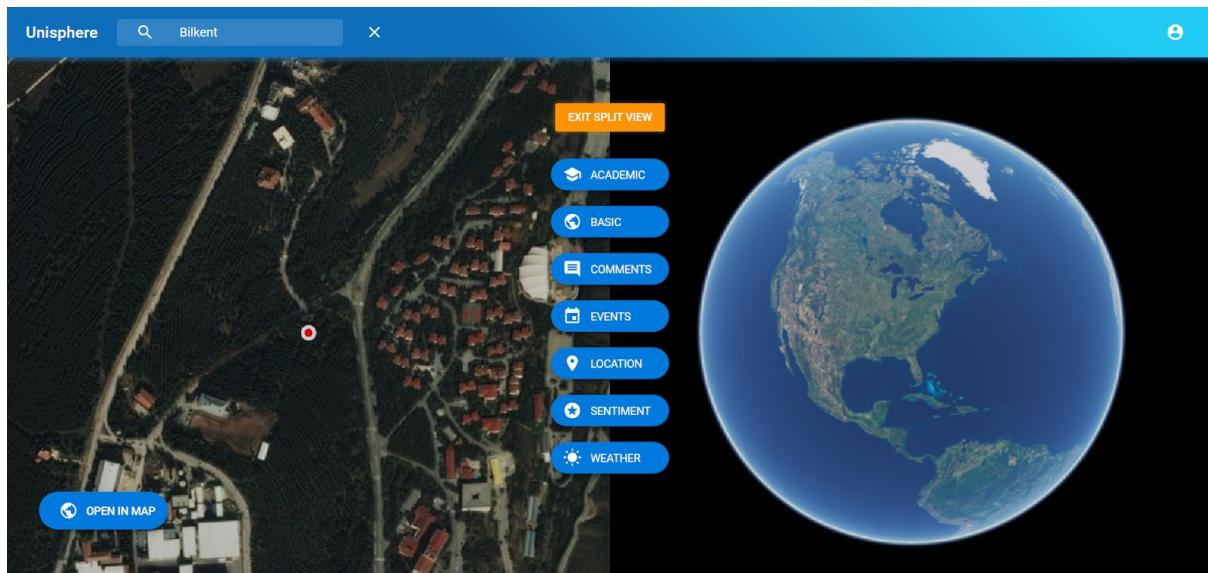
Street View



By dropping the street view pin on a location on the map view, users can enter the street view where they can wander around and discover the location that they dropped the pin into.

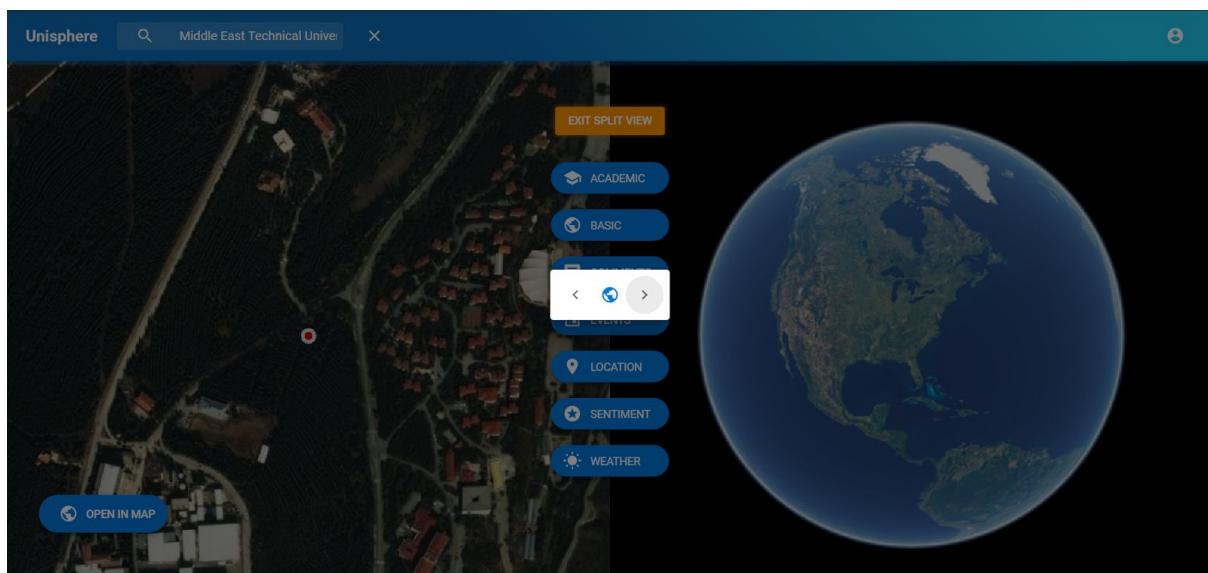
Split View

Split View, University on the Left



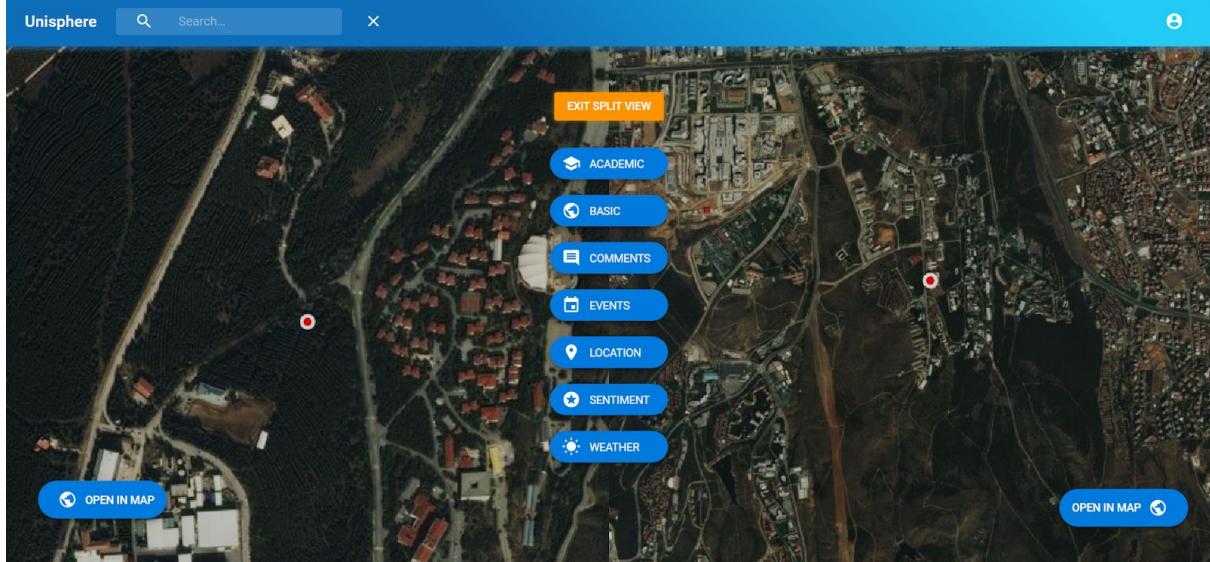
In the split view screen, users can search for universities and put these universities on either the left or the right side of the screen to compare them.

Split View, Select Side Dialog



In the split view screen, users can put the university that they search for to left or right side of the screen by clicking on the left or right button that appears after clicking on the name of the searched university respectively.

Split View, Both Sides Populated



After searching for universities and populating both sides of the screen users have several options and metrics to compare the universities on left and right sides of the screen.

Academic Detail

The screenshot shows a Unisphere interface with a split-screen view. On the left, the 'Academic' tab is selected for Bilkent University. It displays the university's logo, name, and basic information: Agency for Bilkent Üniversitesi is Council of Higher Education (YÖK), founded in 1984, and its admission requirement is a secondary school certificate or equivalent. On the right, the 'Academic' tab is selected for Middle East Technical University. It displays the university's logo, name, and basic information: Agency for Middle East Technical University is Council of Higher Education (YÖK), established in 1956, and its admission requirement is a secondary school certificate or equivalent. A central sidebar provides navigation options: EXIT SPLIT VIEW, Academic, Basic, Comments, Events, Location, Sentiment, and Weather.

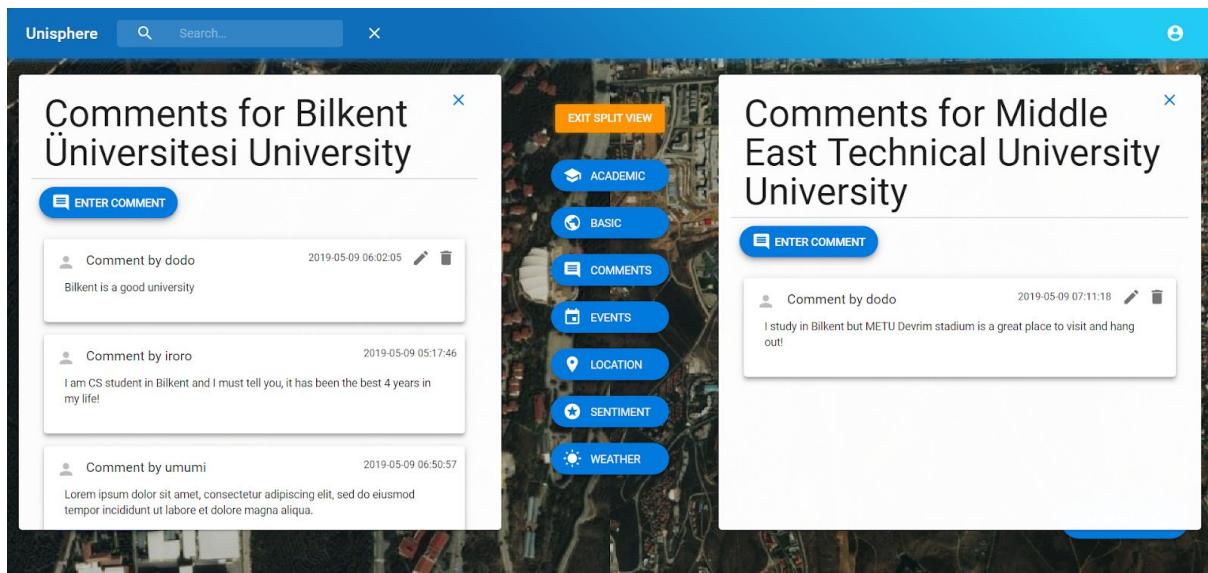
By clicking on the “academic” button while on the split view screen users can view the academic information for both the universities and compare them.

Basic Detail

The screenshot shows a Unisphere interface with a split-screen view. On the left, the 'Basic' tab is selected for Bilkent University. It displays the university's location (Ankara, Turkey), postal code (06800), contact phone number (+90(312) 290-40-00), head (Abdullah Atalar), job title (Rector), senior administrative officer (Kürsat Aydogan), and secretary general. On the right, the 'Basic' tab is selected for Middle East Technical University. It displays the university's location (Ankara, Turkey), postal code (06800), contact phone number (+90(312) 210-11-00), head (Mustafa Versan Kok), job title (Rector), and a note to visit their website for more information. A central sidebar provides navigation options: EXIT SPLIT VIEW, Academic, Basic, Comments, Events, Location, Sentiment, and Weather.

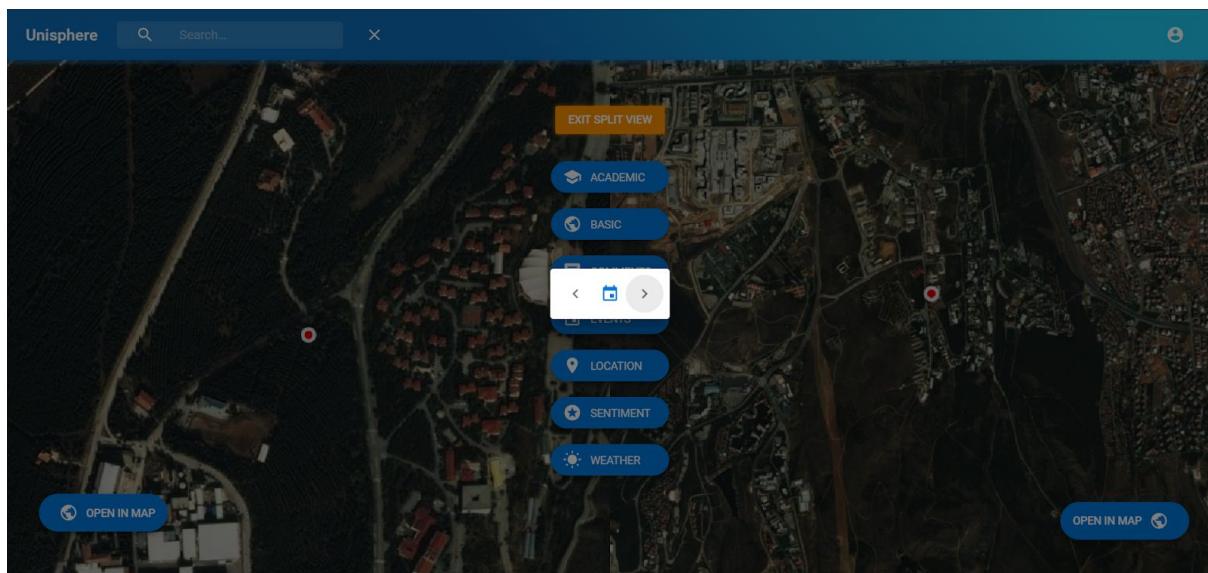
By clicking on the “basic” button while on the split view screen users can view the general information for both the universities and compare them.

Comment Detail



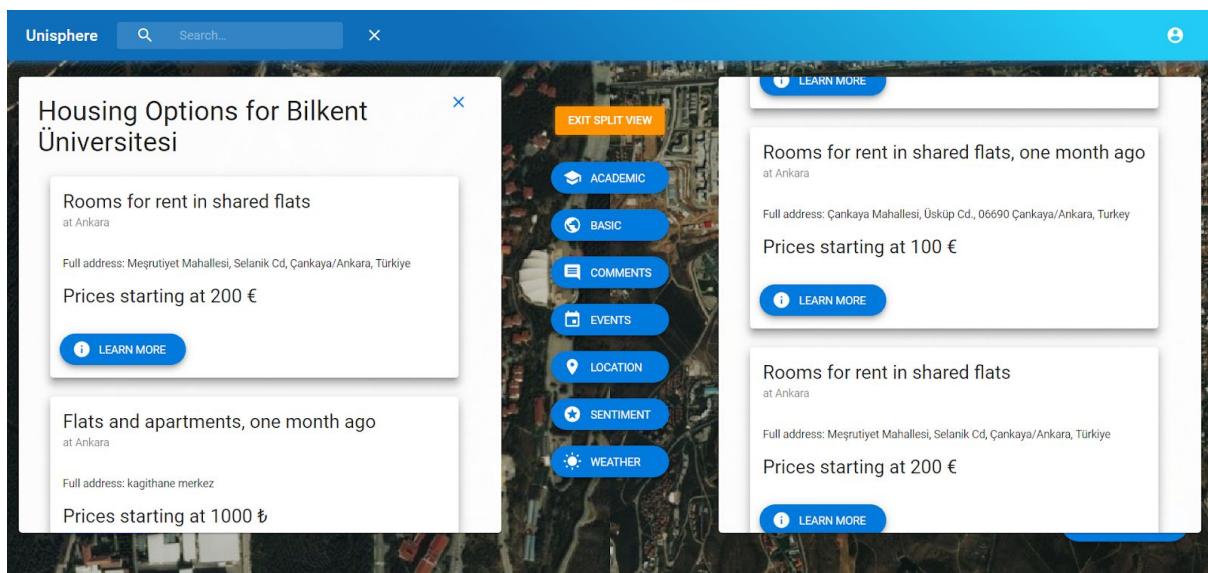
By clicking on the “comments” button while on the split view screen users can view the comments made for both the universities, compare them and comment on them.

Event Select Side Dialog



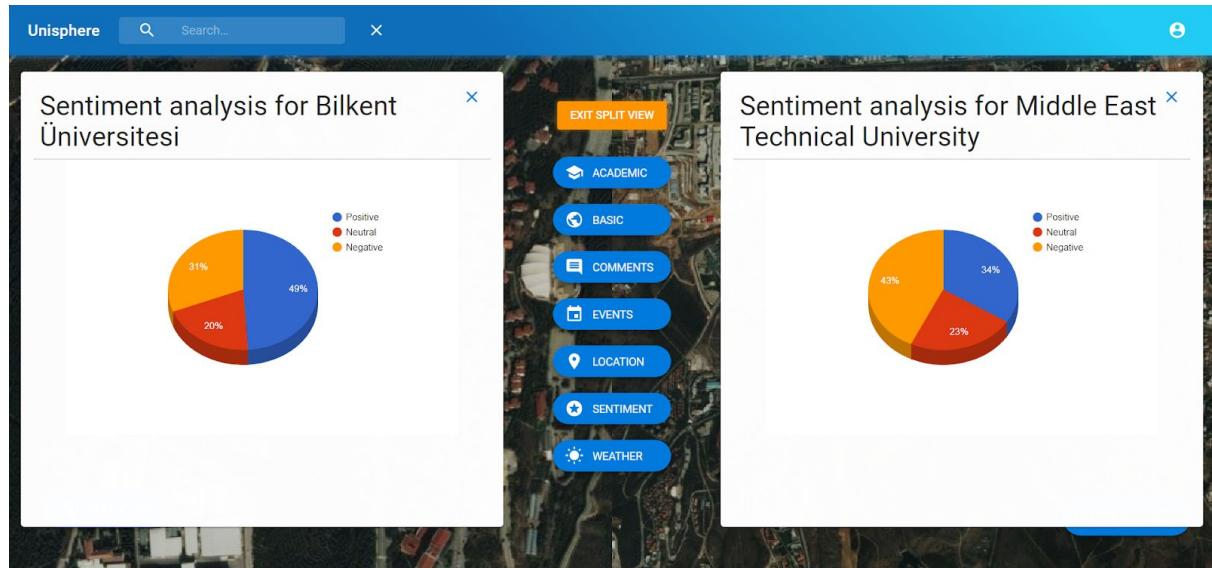
By clicking on the “events” button while on the split view screen users are presented with the option to view the events of the university on the left or the university on the right. If the user clicks on the right arrow the events screen for the university on the right opens and in the same way if the user clicks on the left arrow the events screen for the university on the left opens.

Housing Options Detail



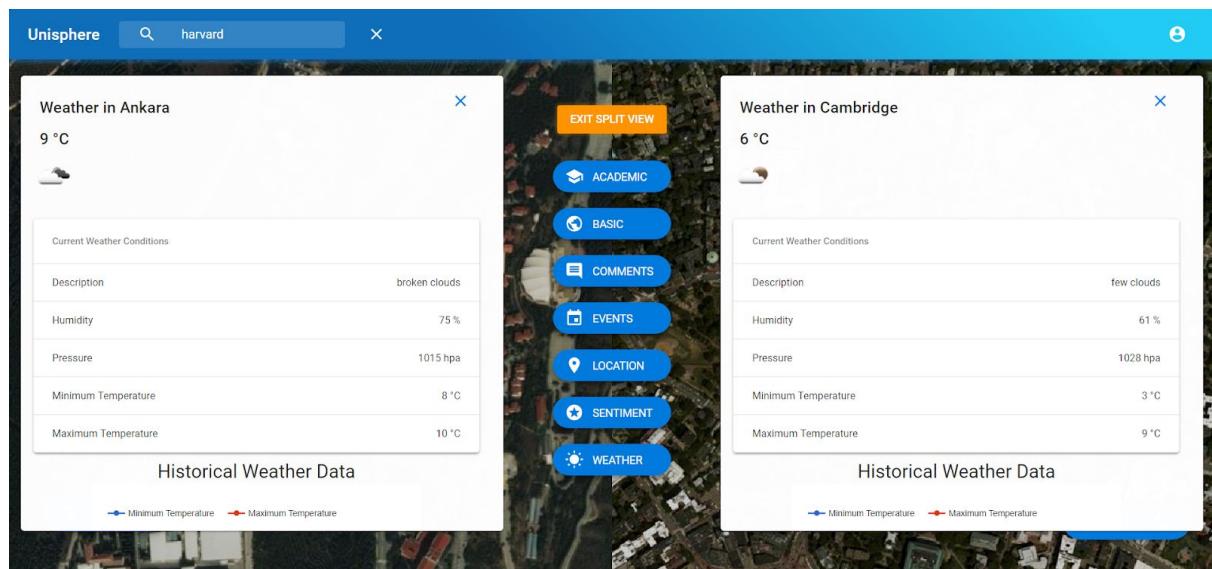
By clicking on the “location” button while on the split view screen users can view the housing options for both the universities and compare them. Users can also click on “learn more” button in the housing options view to open a link to the related advert of the selected university.

Sentiment Detail



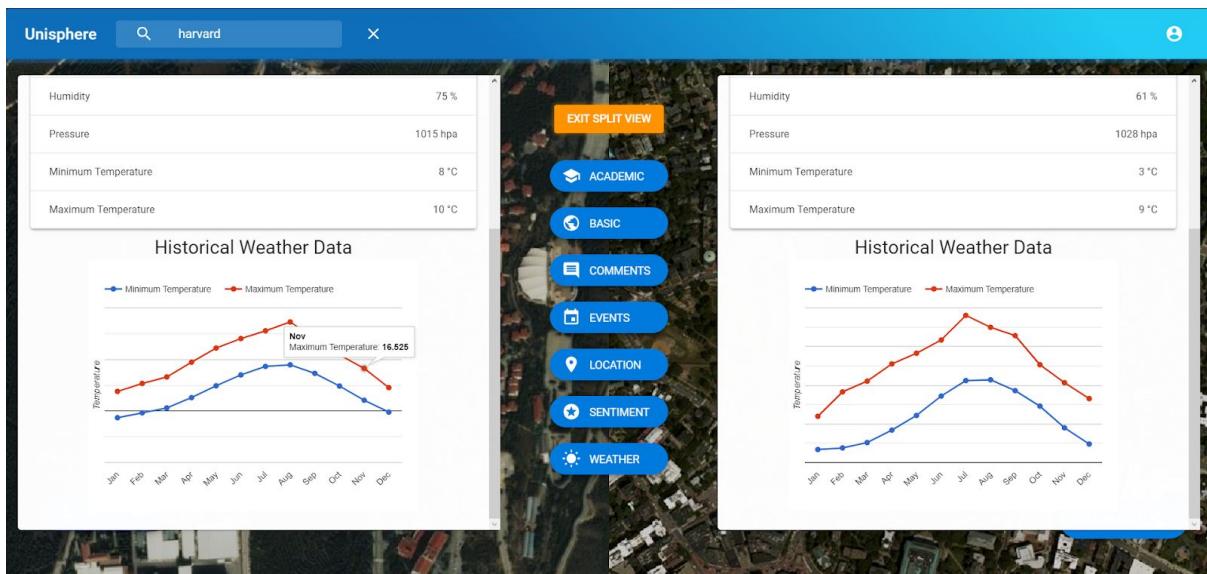
By clicking on the “sentiment” button while on the split view screen users can view the sentiment analysis results created by analyzing tweets for both the universities and compare them using the pie charts provided.

Current Weather Detail



By clicking on the “weather” button while on the split view screen users can view the weather information for both the cities that the universities are located in and compare them with respect to the weather conditions. Users are provided with textual data of the daily weather.

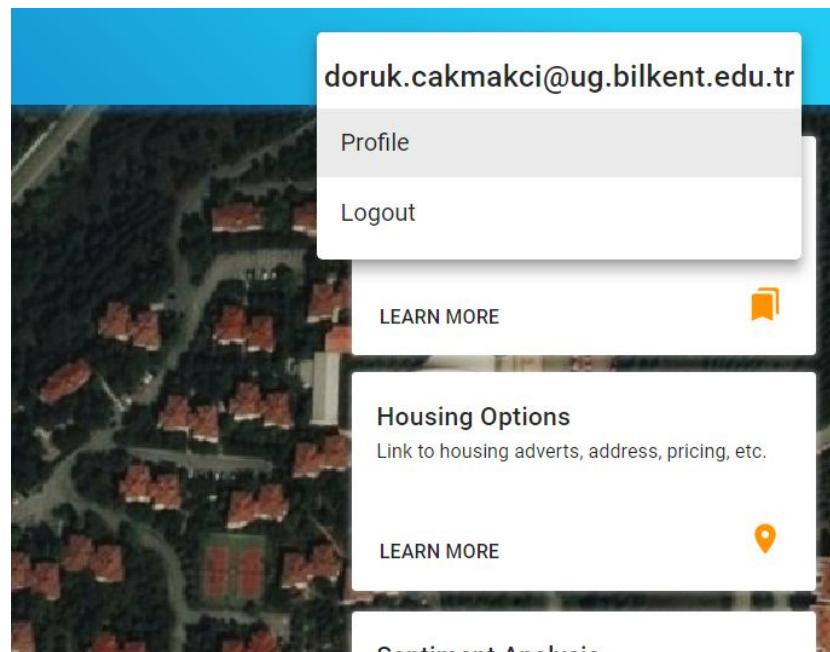
Historical Weather Detail



Users are provided with visual data in the form of a line graph for the historical weather which shows the weather conditions in the past for the city of the selected university. By hovering over the points for each month in the line graph users can see the exact temperature of that month.

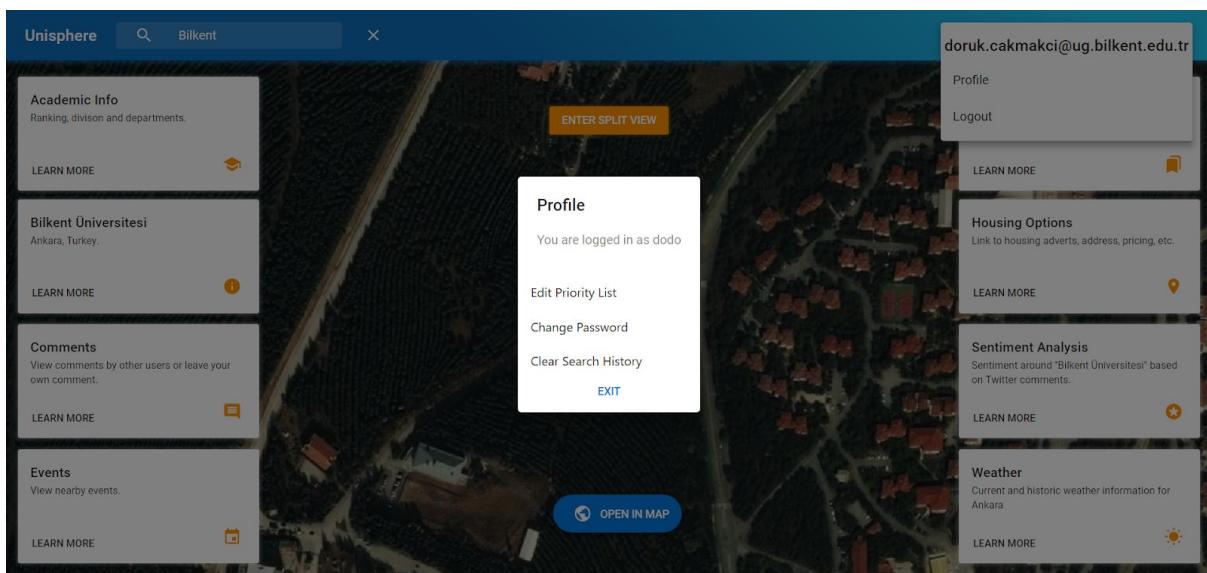
Profile and Options

Profile Menu Main View



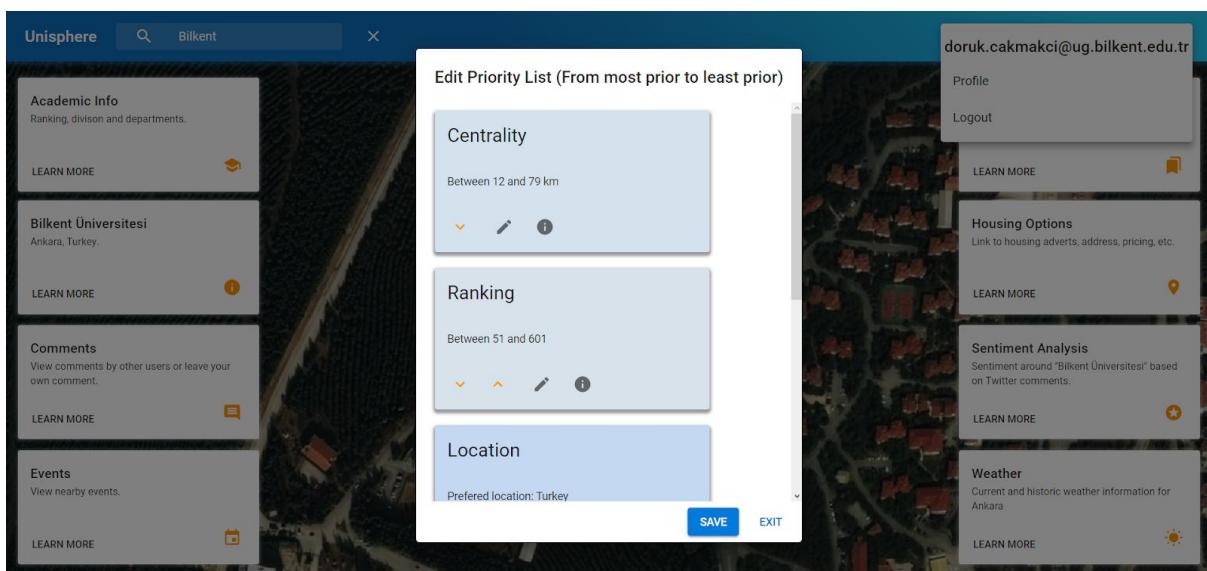
In the main view users can click on the icon on the upper right corner to see which email they are logged in under as and have the option to view the options view by clicking profile and also logout of their accounts using the logout button.

Profile Menu, Options Display



In the profile menu users can see which profile name they are logged in under as and have the options to edit their priority list, change their password and clear their search history.

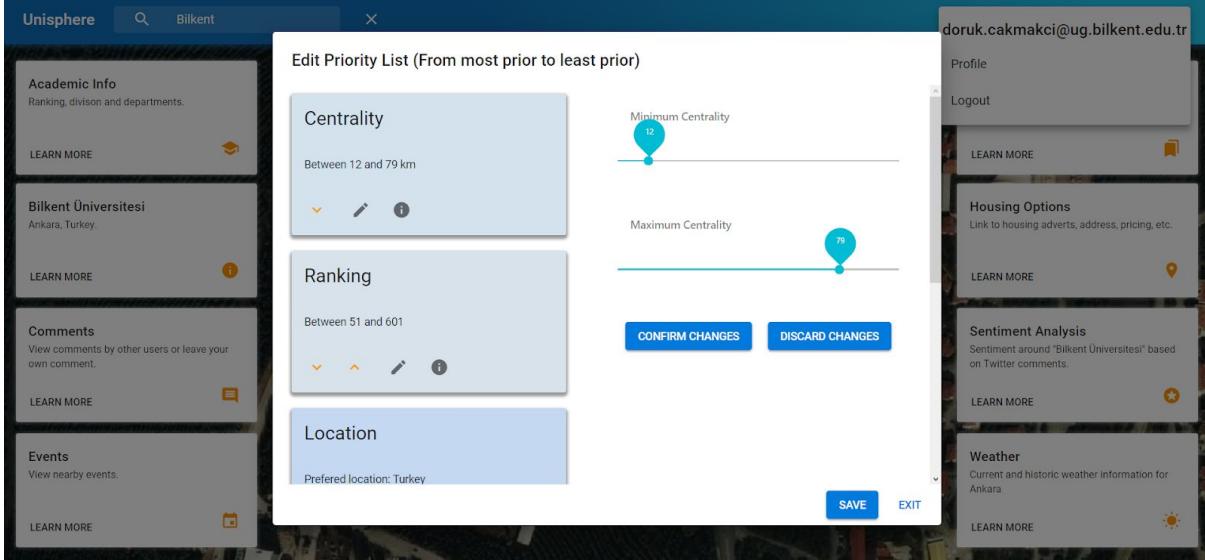
Priority List Main View



Users can change the order of the priorities by using the down arrow to switch places with the next bottom priority and the up arrow to

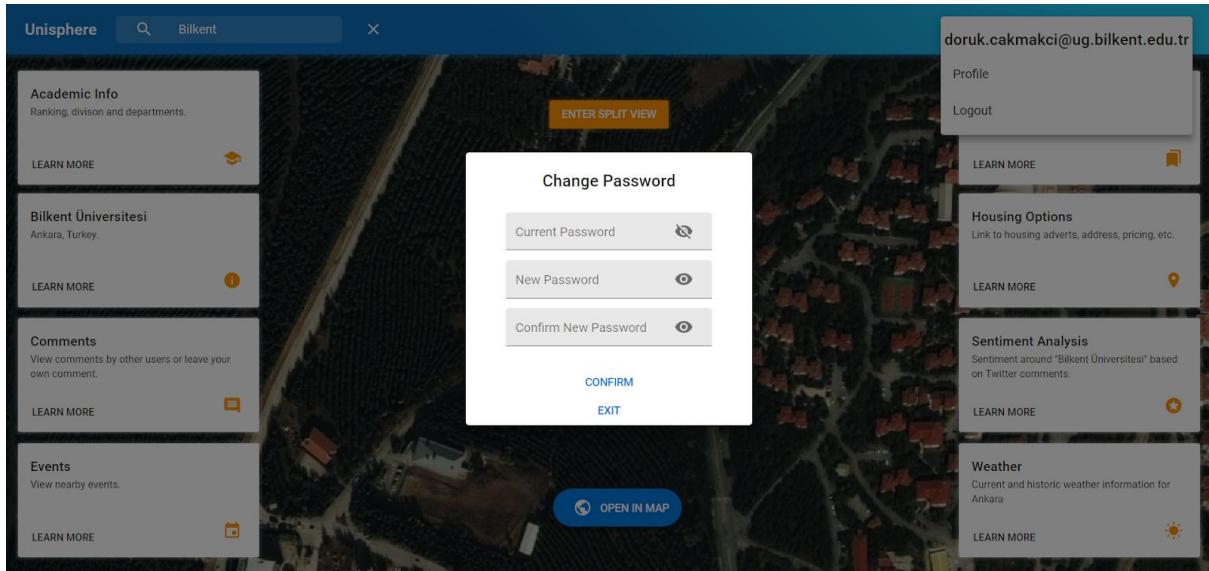
switch places with the next top priority. This swap operation is done so that when the system recommends a university to the user, it does so by considering the topmost priority the most and the bottommost priority the least.

Priority List Edit View



Users can change the magnitude of priority metrics to be used in university recommendations in the edit priority list screen. By clicking the edit icon users can alter the properties of the selected priority with the use of sliders and using the confirm changes button.

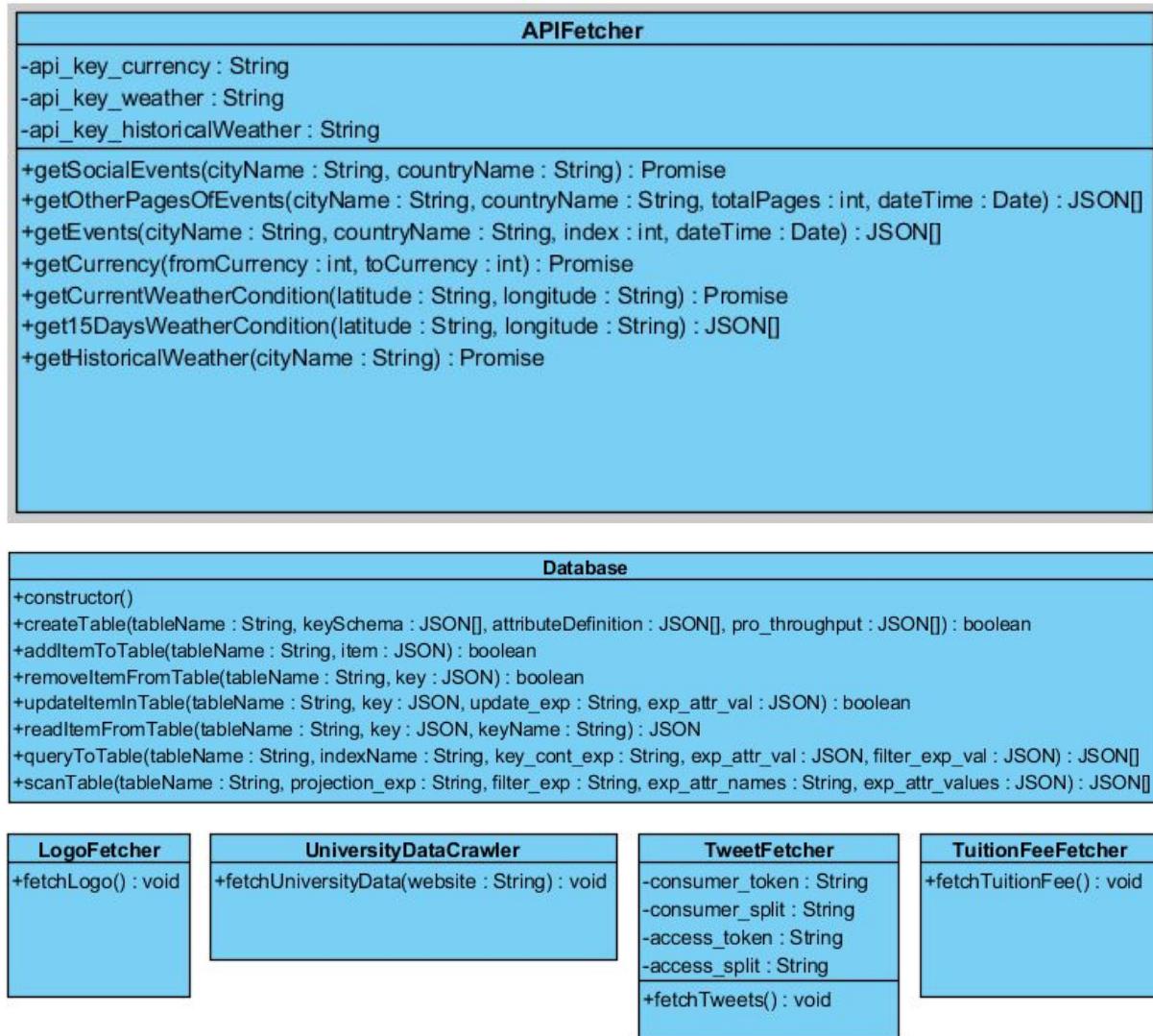
Change Password View



Users can change their passwords by entering their current passwords in the current password field and the desired new password into the new password and confirm new password areas and the clicking confirm button. Users can also exit from this screen by using the exit button.

Appendix B - Class Diagrams

Data Layer



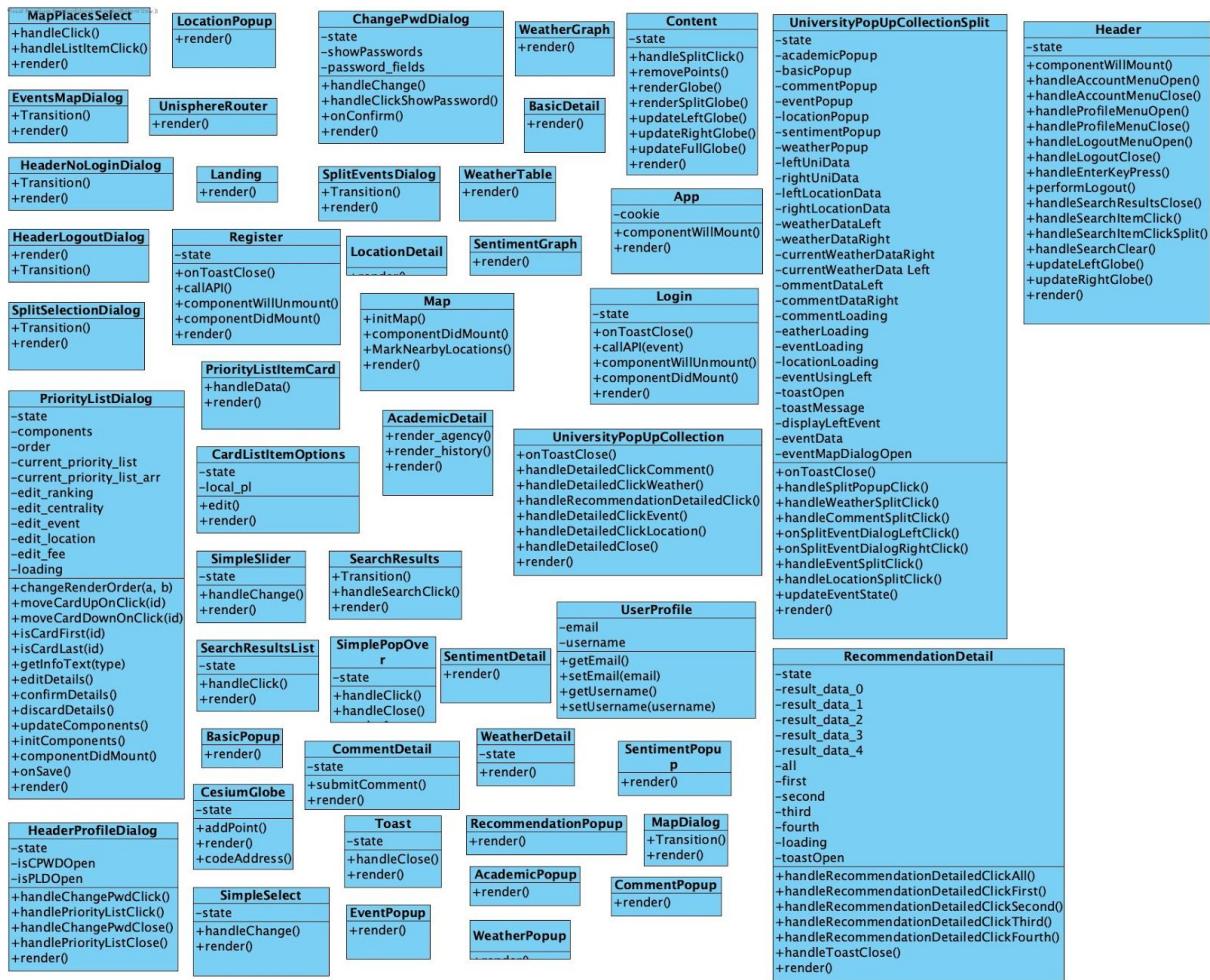
Logic View

MainController
-db : Database
+searchDatabase(key : String) : JSON[]
+search(req : JSON, res : JSON, data : JSON) : void
+makeComment(req : JSON, res : JSON, data : JSON) : void
+deleteComment(req : JSON, res : JSON, commentID : int) : void
+editComment(req : JSON, res : JSON, data : JSON) : void
+getComments(req : JSON, res : JSON, uniName : String) : void
+getHousingOptions(req : JSON, res : JSON, cityName : String) : void
+getRecommendations(req : JSON, res : JSON, email : String) : void
+requestRecommendationFirst(req : JSON, res : JSON, queries : JSON, tuition_fee_min : int, tuition_fee_max : int) : void
+requestRecommendationSecond(req : JSON, res : JSON, queries : JSON, tuition_fee_min : int, tuition_fee_max : int) : void
+requestRecommendationThird(req : JSON, res : JSON, queries : JSON, tuition_fee_min : int, tuition_fee_max : int) : void
+requestRecommendationFourth(req : JSON, res : JSON, queries : JSON, tuition_fee_min : int, tuition_fee_max : int) : void
+filterData(data : JSON, tuition_fee_min : int, tuition_fee_max : int) : void

FetcherController
-db : Database
+retrieveCurrency(req : JSON, res : JSON, fromCurrency : String, toCurrency : String) : void
+retrieveCurrentWeatherConditions(req : JSON, res : JSON, latitude : String, longitude : String) : void
+retrieveHistoricalWeather(req : JSON, res : JSON, cityName : String) : void
+retrieveSocialEvents(req : JSON, res : JSON, queries : JSON, cityName : String, countryName : String) : void

SearchResultsController	UserController
-db : Database	-db : Database
+uni_details_by_name(req : JSON, res : JSON, data : JSON)	+login(req : JSON, res : JSON, data : JSON) : void
	+register(req : JSON, res : JSON, data : JSON) : void
	+changePassword(req : JSON, res : JSON, data : JSON) : void
TweetAnalyzer	App
-db : Database	+post(request_keyword : String, callback : func()) : void
+sentiment_scores(sentence : String) : void	
+operation()	

Client Subsystem



References

- [1] N. Foundation, “About,” *Node.js*. [Online]. Available: <https://nodejs.org/en/about/>. [Accessed: 09-May-2019].
- [2] “Amazon DynamoDB,” *Amazon Web Services, Inc.* [Online]. Available: <https://www.amazonaws.cn/en/dynamodb/>. [Accessed: 09-May-2019].
- [3] “Build software better, together,” *GitHub*. [Online]. Available: <https://github.com/>. [Accessed: 09-May-2019].
- [4] “API Development Environment,” *Postman*. [Online]. Available: <https://www.getpostman.com/>. [Accessed: 09-May-2019].
- [5] “Browser Automation,” *Selenium Blog Posts*. [Online]. Available: <https://www.seleniumhq.org/>. [Accessed: 09-May-2019].
- [6] “Compute Engine - IaaS | Compute Engine | Google Cloud,” *Google*. [Online]. Available:
https://cloud.google.com/compute/?utm_source=google&utm_medium=cpc&utm_campaign=emea-tr-all-en-dr-bkws-all-all-trial-b-gcp-1003963&utm_content=text-ad-none-any-DEV_c-CRE_253512765103-ADGP_Hybrid | AW SEM | BKWS ~ BMM_M:1_TR_EN_Compute_Compute Engine_google engine compute-KWID_43700016300875859-kwd-205796560567-userloc_1012763&utm_term=KW_google engine compute-ST_google engine compute&ds_rl=1245734&gclid=CjwKCAjw_MnmBRAoEiwAPRRWWxfboeTKNTQ3DKwaOeKxKnQByoKV-TUQGS-6ZRINT2Kq5DJN832yxBoCLjEQAvD_BwE. [Accessed: 09-May-2019].
- [7] “Beautiful Soup Documentation¶,” *Beautiful Soup Documentation - Beautiful Soup 4.4.0 documentation*. [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [Accessed: 09-May-2019].
- [8] Hardikvasa, “hardikvasa/google-images-download,” *GitHub*, 01-May-2019. [Online]. Available: <https://github.com/hardikvasa/google-images-download>. [Accessed: 09-May-2019].
- [9] OpenWeatherMap.org, “Weather API,” *openweathermap*. [Online]. Available: <https://openweathermap.org/api>. [Accessed: 09-May-2019].

[10] “Eventbrite,” *Eventbrite*. [Online]. Available: <https://www.eventbrite.com/>. [Accessed: 09-May-2019].

[11]“What does our historical or past weather API offer?,” *Historical Weather API | Historical Weather | World Weather Online*. [Online]. Available: <https://www.worldweatheronline.com/developer/api/historical-weather-api.aspx>. [Accessed: 09-May-2019].

[12] M. Vergel, “The simplest API for currency conversion,” *Currency Converter API - Web Services for Currency Conversion Free or Premium*. [Online]. Available: <https://www.currencyconverterapi.com/>. [Accessed: 09-May-2019].

[13]Google.[Online].Available:<https://developers.google.com/maps/documentation/javascript/tutorial>. [Accessed: 09-May-2019].

[14] “Overview | Places API | Google Developers,” *Google*. [Online]. Available: <https://developers.google.com/places/web-service/intro>. [Accessed: 09-May-2019].

[15]Google.[Online].Available:<https://developers.google.com/maps/documentation/directions/start>. [Accessed: 09-May-2019].

[16]Google.[Online].Available:<https://developers.google.com/maps/documentation/geocoding/start>. [Accessed: 09-May-2019].

[17]Google.[Online].Available:<https://developers.google.com/maps/documentation/roads/intro>. [Accessed: 09-May-2019].

[18]“Cloud Translation API documentation | Cloud Translation API | Google Cloud,” *Google*. [Online]. Available: <https://cloud.google.com/translate/docs/>. [Accessed: 09-May-2019].

[19] “Twitter Developer Platform - Twitter Developers,” *Twitter*. [Online]. Available: <https://developer.twitter.com/>. [Accessed: 09-May-2019].

[20]“Tweepy,” *Tweepy*. [Online]. Available: <https://www.tweepy.org/>. [Accessed: 09-May-2019].

[21]“Source code for nltk.sentiment.vader,” *nltk.sentiment.vader - NLTK 3.4.1 documentation*. [Online]. Available: https://www.nltk.org/_modules/nltk/sentiment/vader.html. [Accessed: 09-May-2019].

[22]“Natural Language Toolkit¶,” *Natural Language Toolkit - NLTK 3.4.1 documentation*. [Online]. Available: <https://www.nltk.org/>. [Accessed: 09-May-2019].

[23]“langdetect,” *PyPI*. [Online]. Available: <https://pypi.org/project/langdetect/>. [Accessed: 09-May-2019].

[24]“Boto 3 Documentation¶.” *Boto 3 Documentation - Boto 3 Docs 1.9.144 documentation*. [Online]. Available:

https://boto3.amazonaws.com/v1/documentation/api/latest/index.html?id=docs_gateway. [Accessed: 09-May-2019].

[25]Max, “max/swot-api,” *GitHub*, 22-May-2018. [Online]. Available: <https://github.com/max/swot-api>. [Accessed: 09-May-2019].