



Bilkent University

Department of Computer Engineering

CS 491- Senior Design Project I

Unisphere: Global University Catalog Analysis 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

Analysis Report

November 12,, 2018

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

Introduction	3
2. Current System	4
3. Proposed System	5
3.1 Overview	5
3.2 Functional Requirements	6
3.2.1 User Specific Requirements	6
3.2.2 Data Resources	7
3.3 Nonfunctional Requirements	8
3.3.1 Usability	8
3.3.2 Maintainability	8
3.3.3 Efficiency	9
3.3.4 Scalability	9
3.3.5 Response Time	9
3.3.6 Legal	10
3.4 Pseudo Requirements	10
3.5 System Models	11
3.5.1 Scenarios	11
3.5.1.1 Scenario 1	11
3.5.1.2 Scenario 2	11
3.5.1.3 Scenario 3	12
3.5.1.4 Scenario 4	12
3.5.1.5 Scenario 5	13
3.5.1.6 Scenario 6	13
3.5.1.7 Scenario 7	14
3.5.1.8 Scenario 8	15
3.5.1.9 Scenario 9	15
3.5.1.10 Scenario 10	15
3.5.1.11 Scenario 11	16
3.5.1.12 Scenario 12	16
3.5.1.13 Scenario 13	17
3.5.2 Use Case Model	18
3.5.3 Object and Class Model	19
3.5.4 Dynamic Models	22

3.5.4.1 Sequence Diagrams	22
3.5.4.1.1 Compare Universities Diagram	22
3.5.4.1.2 Search University Diagram	23
3.5.4.1.3 View Information Diagram	24
3.5.4.1.4 View Events Diagram	25
3.5.4.1.5 Select Department Diagram	26
3.5.4.1.6 Login Diagram	27
3.5.4.1.7 Sign-up Diagram	28
3.5.4.1.8 Comment on Universities Diagram	29
3.5.4.1.9 View Recommendations Diagram	30
3.5.4.1.10 View Nearby Attraction Diagram	31
3.5.4.1.11 Select Priority List Diagram	32
3.5.4.2 Activity Diagram	33
3.5.5 User Interface	34
3.5.5.1 Home Page	34
3.5.5.2 Register Page	35
3.5.5.3 Login Page	36
3.5.5.4 University Information Page	37
3.5.5.5 Compare Page	38
3.5.5.6 Compare Page - 2	39
3.5.5.7 Compare Page(Detailed View)	40
5 References	41

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 these universities in these several ways. Therefore, by following this urge, instead of searching all the universities in different tabs and then try to compare them by yourself 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 in and 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.

In this report, the current available systems will be demonstrated. Afterwards, functional, non functional and pseudo requirements are explained. System models including scenarios, use cases, class and dynamic models will be stated and clarified. Lastly, to illustrate Unisphere better, mockup user interfaces will be pointed out.

2. Current System

Though there are several applications which have some features of Unisphere, this project aims to gather all information from several websites. Besides, instead of focusing only on the academic information, this web application will combine both academic and social features about universities.

- Google Street View [1]

Google Street View allows 360 tour around the world.

By using Google Street View API, Unisphere will provide tour inside the campus.

- Google Maps [2]

Google Maps shows the location of universities and nearby places around the selected place. It provides routes from a selected point to another.

By using Google Maps API, Unisphere will provide information about nearby places around the university and show these places on the map.

- EventBrite [3]

EventBrite gathers several events around the world, including some events organized by universities.

The events near and inside the university will be fetched and listed in Unisphere.

- University Websites

Each university website has every information related to their university.

Though this application aims to gather several information including the general background information about the university, international student ratio, ranking of the university, departments, Ms and PhD programs of the university.

- Twitter [4]

Twitter has a huge database, which includes comments all around the world.

This application will fetch the comments related to the universities, and it will apply sentiment analysis on the comments and visualize them.

- Reddit [5]

By using Reddit, people leave comments on several topics. Reddit comments which are related to selected universities will be fetched, and sentiment analysis will be applied on these comments.

- OpenWeatherMap.org [6]

Up to date weather conditions of a certain area is provided hourly, weekly and monthly bases. Unisphere will collect the data according to the location of the university and visualize them on graphs.

3. Proposed System

3.1 Overview

Unisphere aims to gather various information related to universities all around the world. In this web application, undergraduate, master's and PhD. programs for different departments will be displayed for each selected university. Users will easily access all universities around the world and they will be able to compare the universities side by side on a user-friendly interface. Besides the educational aspects, by using Google Street view, users can travel inside the campus and they will be able to see the close by facilities such as hospitals, shopping malls, airport, subway. They can learn how to go nearby places by the color-coded fastest routes displayed on an interactive map.

Besides all the above features, this application will provide some additional data related to the location of the university such as the weather conditions in that city, dorms/university prices according to the stock market. Additionally, Unisphere will inform users about the social events (cinemas, concerts, exhibitions etc.) inside and outside of the university by filtering EventBrite events.

By combining both academic and social features of Unisphere, users can prioritize their needs and desires for the university in a priority list and they can get an instant result

which will fit their requests. For instance, they can select whether they give importance to academic success, sociability, whether the university is close to the city center etc. By using this list created by the user and by looking at the users search history, a list of recommended universities will be presented.

Users can utilize this application with or without registering. The registered users will be privileged to access the general public opinion about universities in a visually attractive way. By using Twitter and Reddit comments the thoughts and emotions about the universities will be illustrated. Secondly, these verified users can leave comments about the university.

3.2 Functional Requirements

3.2.1 User Specific Requirements

- All users should be able to view academic and non-academic information about universities. Academic information will include several different categories such as different programs offered by the university, tuition fees, etc. while non-academic information will include distance to the city center, housing prices, concerts, etc.
- Users can register to our application by providing their university email addresses.
- Verified users should be able to see the sentiment analysis results.
- Verified users should be able to comment on universities and programs.
- All users should be able to compare different universities and programs.
- All users should be able to get recommendations based on previous search results and the priority list they choose.
- All users should be able to view different visualizations of the information provided such as pie charts, map pins, textual data, etc..
- All users should be able to travel inside the campus by using Google Street View embedded to the system.

- All users can see the events inside and outside of the university.

3.2.2 Data Resources

- The data about the universities can be categorised to three distinct and mutually-exclusive categories: User-generated data, Manually-generated data and dynamically-generated data. User-generated data will be stored in our database for further usage and there will not be a manual operation to store these data. Manually-generated data will be stored statically in our database(in other words hardcoded data). Dynamically-generated data will be the data categorised as the data generated dynamically via using API functionality(e.g. Tweets, subreddits and EventBrite data), via scraping (e.g. dorm costs, meal plan costs, weather data) where it is not restricted, and via RSS feed parsing(news about universities).
- The data for sentiment analysis will be acquired from twitter and reddit comments.
- The application will take map and route data from Google Maps and 360 degree views of campus from Google Street View.
- The application will gather data for nearby activities and social events from Eventbrite.
- The social events inside the university will be gathered from Eventbrite and RSS feeds of the university web-pages if available.
- The application will have information about housing options near the university.
- The application will provide information about the weather conditions of the area that hosts the university.
- The application should be able to dynamically check current prices and stock market to provide university/dorm prices in a certain currency.
- The application will get a priority list from the user to recommend universities in an ordered list.

- Undergraduate, masters and PhD. programs of the universities will be gathered open source web pages, though some informations might be added manually to the database.
- Comments about the universities will be generated by the registered users.

3.3 Nonfunctional Requirements

3.3.1 Usability

- The application should be easy to use. Users should be able to just jump in and navigate through a user-friendly interface.
- Application should require minimum amount of information (such as university mail address) from users to classify them as verified.
- Different types of information should be available to different types of users (verified users and guest users). Only verified users will be able to comment to their own university and view sentiment analysis results for universities.
- Users should fluently select universities on the globe and should be able to compare the selected universities side by side in a visually attractive way.

3.3.2 Maintainability

Since the application is very heavily data driven, maintainability is very important for the correctness of the data at all times is vital. The data about universities will be collected and updated regularly. The user-generated data does not need any update since they are time stamped and not subject to change. However, the dynamically-generated and manually-generated data are subject to change. The dynamically-generated data is updated regularly and dynamically. On the other hand, in the case of manually-generated data, it is hard to detect when there is an update available for the data. Nevertheless, the

manually-generated data will consist of the data that are updated with a period, which will be held in our database and notified when an update is due.

3.3.3 Efficiency

For Unisphere, efficiency is of key value. The data retrieval, visualization and analysis algorithms should be efficient in order to make user experience better and more robust. For example if it would take less time for the user to find the data that Unisphere presents on other websites this would defeat the purpose of our application. For this reason, the layout and representation of the data must be efficient. The desirable data must be readily available and user must be able to reach such data quickly.

3.3.4 Scalability

- The application should be built so that information about new universities can be added and information about already existing universities can be updated without inconvenience.
- When there are several users, the application should stay available and responsive all the time.

3.3.5 Response Time

- The application should take minimal time while switching between different visualizations of data.
- A user shouldn't wait long when he/she selects a university. The data should be available to a user in a short time(no more than 2 seconds).
- Also, the time needed to visualize the data about a university will evidently be a bottleneck for this project. The runtime of visualization related operations shouldn't be more than 3 seconds.
- Loading Google Street View and Google Map should not take too much time.

3.3.6 Legal

- The data should be acquired by legal means, it shouldn't be acquired in an unauthorized manner from pages via methods like scraping.
- The data about the inter-university relations[THE][QS], will be used and properly-cited. Since the purpose of Unisphere is to collect various, important data about universities, this project does not show any rivalry to [THE] or [QS] but, use their data to enhance the abilities of the program.

3.4 Pseudo Requirements

This application will be a webapp and the front-end will be implemented using JavaScript and the ReactJS [5] framework. On the back-end side we will host the website on a DigitalOcean [6] droplet. Our database will be using AWS' DynamoDB [3] which is a non-relational / noSQL cloud database solution.

The application will have to use several different API's for data retrieval and other tools:

- SWOT is a Ruby based tool that allows domain name identification for universities. The application must use this tool in order to verify students are legitimate [7].
- Twitter API[4], specifically Keyword Insight tool will be used in order to retrieve tweets about a given university which will be used in sentiment analysis.
- Reddit API[5] will also be used in a similar fashion; the application must retrieve relevant reddit posts which will possibly be used in sentiment analysis and additional data regarding the university.
- A collection of Google APIs will be utilized such as the Google Maps API [2] in order to find the location of the university given the name. The maps API will also be used in order to find nearby interest points and social facets, and visualize the routes to these found places. Google's search API [8] will possibly be utilized in order to further retrieve and categorize data about a given university. Google Chart API [9] could be utilized in order to better visualize data. Google Cloud Translate API [10] will be used to translate non-english user generated data to English for them be used in sentiment analysis.

- OpenWeather API will be used to gather weather information both for short and long time frames [6].
- EventBrite API [3] will be used in order to find out events and event history around and inside a given university.

3.5 System Models

3.5.1 Scenarios

3.5.1.1 Scenario 1

Use Case Name: Compare

Actors: Umut

Entry Conditions:

- User Umut is on Home Page

Exit Conditions:

- User Umut is on Compare Screen

Main Flow of Events:

1. User Umut clicks the “Compare” button on Home Page
2. Unisphere displays 2 search bars that return a list of universities
3. User Umut selects the universities that he wishes to compare
4. Unisphere displays the information of both universities and compares their features individually.

3.5.1.2 Scenario 2

Use Case Name: Search

Actors: Umut

Entry Conditions:

- User umut is on Home Page

Exit Conditions:

- User Umut is on Search Results Page

Main Flow of Events:

1. User Umut searches a university/ department/ location using the search bar.
2. Unisphere checks the database for similar results to searched key.
3. Unisphere displays the universities similar to the searched key.
4. User Umut clicks on the university that he searched for.

Alternative Flow of Events:

1. User Umut searches a university that he wants to get information about using the search bar.
2. Unisphere checks the database for similar results to searched key.
3. Unisphere displays the universities similar to the searched key.
4. User Umut can't find the university that he searched for in the search results.

3.5.1.3 Scenario 3

Use Case Name: ViewInfo

Actors: Umut

Entry Conditions:

- User Umut selects a university from search results or he selects a university from the globe model

Exit Conditions:

- User Umut is on Information Screen

Main Flow of Events:

1. User Umut selects a university from the search results.
2. Unisphere displays the related information about the selected university.

Alternative Flow of Events:

1. User Umut selects the university that he is looking for on the globe model.
2. User Umut clicks on the pin of the related university.
3. Unisphere displays the related information about the selected university.

3.5.1.4 Scenario 4

Use Case Name: ViewEvents

Actors: Umut

Entry Conditions:

- User Umut is on the Information Screen

Exit Conditions:

- User Umut is on the Events Screen

Main Flow of Events:

1. User Umut clicks on the "View Events" button.
2. Unisphere displays events related to the university that Umut is in the information page of.

3.5.1.5 Scenario 5

Use Case Name: ViewDetails

Actors: Umut

Entry Conditions:

- User Umut is on the Information Screen

Exit Conditions:

- User Umut is on the Details Screen

Main Flow of Events:

1. User Umut clicks on the “View Details” button.
2. Unisphere navigates to the Details Screen.
3. Unisphere displays more detailed information about different features of the university.
 - a. Unisphere displays data visualizations related to the university.
 - b. Unisphere displays detailed information about non-academic features of the university.
 - c. Unisphere displays detailed information about academic features of the university.
 - d. Unisphere displays Google Street View of the university.

3.5.1.6 Scenario 6

Use Case Name: Register

Actors: Umut

Entry Conditions:

- User Umut is on the Home Screen

Exit Conditions:

- User Umut is on the Home Screen

Main Flow of Events:

1. User Umut clicks the “Register” button.
2. Unisphere navigates to Register Screen.
3. User Umut enters his school mail address.
4. User Umut enters his choice of password.
5. User Umut clicks the “Register” button.
6. Unisphere checks if the school mail address is legitimate.
7. Unisphere checks if the entered password complies to the password rules.
8. Unisphere creates a new account with the given email address and password.
9. Unisphere navigates to the Home Screen.

Alternative Flow of Events:

1. User Umut clicks the “Register” button.
2. Unisphere navigates to Register Screen.
3. User Umut enters his school mail address.
4. User Umut enters his choice of password.
5. User Umut clicks the “Register” button.
6. Unisphere checks if the school mail address is legitimate and if the password complies to the password rules.
7. Unisphere displays an error message saying that the password or the mail address is incorrect.
8. User Umut clicks on the “Return to Home Screen” button.
9. Unisphere navigates to the Home Screen.

3.5.1.7 Scenario 7

Use Case Name: Login

Actors: Doruk

Entry Conditions:

- Verified user Doruk is on the Home Screen

Exit Conditions:

- Verified user Doruk is on the Home Screen

Main Flow of Events:

1. Verified user Doruk fills in the mail address and password sections of the login box.
2. Verified user Doruk clicks on the “Login” button.
3. Unisphere navigates user to Login Screen.
4. Unisphere checks if the entered e-mail address exists in the database.
5. Unisphere checks if the entered password is in the database and if the password matches the entered e-mail.
6. Unisphere confirms that the credentials are correct.
7. Unisphere navigates to the Home Screen with Doruk used in.

Alternative Flow of Events:

1. Verified user Doruk fills in the mail address and password sections of the login box.
2. Verified user Doruk clicks on the “Login” button.
3. Unisphere checks if the entered e-mail address exists in the database.
4. Unisphere checks if the entered password is in the database and if the password matches the entered e-mail.
5. Unisphere displays a message that says the credentials are incorrect.

3.5.1.8 Scenario 8

Use Case Name: Logout

Actors: Doruk

Entry Conditions:

- Verified user Doruk is logged in to the system

Exit Conditions:

- Verified user Doruk is on the Home Screen

Main Flow of Events:

1. Verified user Doruk clicks on the drop down menu at the top.
2. Verified user Doruk clicks on the “Logout” button from the drop down menu.
3. Unisphere logs Doruk out of the system.
4. Unisphere navigates Doruk to the Home Screen.

3.5.1.9 Scenario 9

Use Case Name: ViewSentimentAnalysis

Actors: Doruk

Entry Conditions:

- Verified user Doruk is on the Home Screen

Exit Conditions:

- Verified user Doruk is on the Information Screen

Main Flow of Events:

1. Verified user Doruk clicks on the “View Info” button
2. Unisphere navigates Doruk to the Information Screen
3. Unisphere displays sentiment analysis data along with other data
4. Doruk views the sentiment analysis data

3.5.1.10 Scenario 10

Use Case Name: CommentOnUniversities&Programs

Actors: Doruk

Entry Conditions:

- Verified user Doruk is on the Information Screen

Exit Conditions:

- Verified user Doruk is on the Comments Screen

Main Flow of Events:

1. Verified user Doruk clicks on the “Comments” button
2. Unisphere navigates to the Comments Page

3. Unisphere fetches and displays comments
4. Verified user Doruk clicks on the comment box
5. Verified user Doruk writes his comment
6. Verified user Doruk clicks on the “Comment” button
7. Unisphere updates its comment database
8. Unisphere adds Doruk’s comment on the Comments Screen

3.5.1.11 Scenario 11

Use Case Name: ViewRecommendations

Actors: Doruk

Entry Conditions:

- User Doruk is on the Home Screen

Exit Conditions:

- User Doruk is on the Recommendations Screen

Main Flow of Events:

1. User Doruk clicks on the “Get Recommendation” button.
2. Unisphere navigates to the Recommendations Screen.
3. Unisphere collects relevant data from Doruk’s priority and his browsing history.
4. Unisphere lists the recommendations that it derived from relevant data

Alternative Flow of Events:

1. Verified user Doruk clicks on the “Get Recommendation” button.
2. Unisphere navigates to the Recommendations Screen.
3. Unisphere collects relevant data from Doruk’s browsing history.
4. Unisphere lists the recommendations that it derived from relevant data

3.5.1.12 Scenario 12

Use Case Name: ViewGoogleStreetView

Actors: Doruk

Entry Conditions:

- User Doruk is on the Information screen.

Exit Conditions:

- User Doruk is on the Home screen.

Main Flow of Events:

1. User Doruk click the Street View button.
2. Unisphere fetches the Street View data using the Google Maps API.
3. Unisphere embeds the view on top of the screen.
4. Unisphere renders the street view for Doruk’s viewing.

3.5.1.13 Scenario 13

Use Case Name: ChoosePriorityList

Actors: Doruk

Entry Conditions:

- Verified user Doruk is on the Preferences screen.

Exit Conditions:

- Verified user Doruk is on the Home screen.

Main Flow of Events:

1. Verified user Doruk clicks on “Set Priority List” button.
2. Unisphere displays an editable list of tags on the screen.
3. Doruk uses the priority list selector UI element in order to create a priority list.

Alternative Flow of Events:

1. Verified user Doruk clicks on “Set Priority List” button.
2. Unisphere displays an editable list of tags on the screen.
3. Doruk uses the priority list selector UI element in order to change his previous priority list elements’ order.

3.5.2 Use Case Model

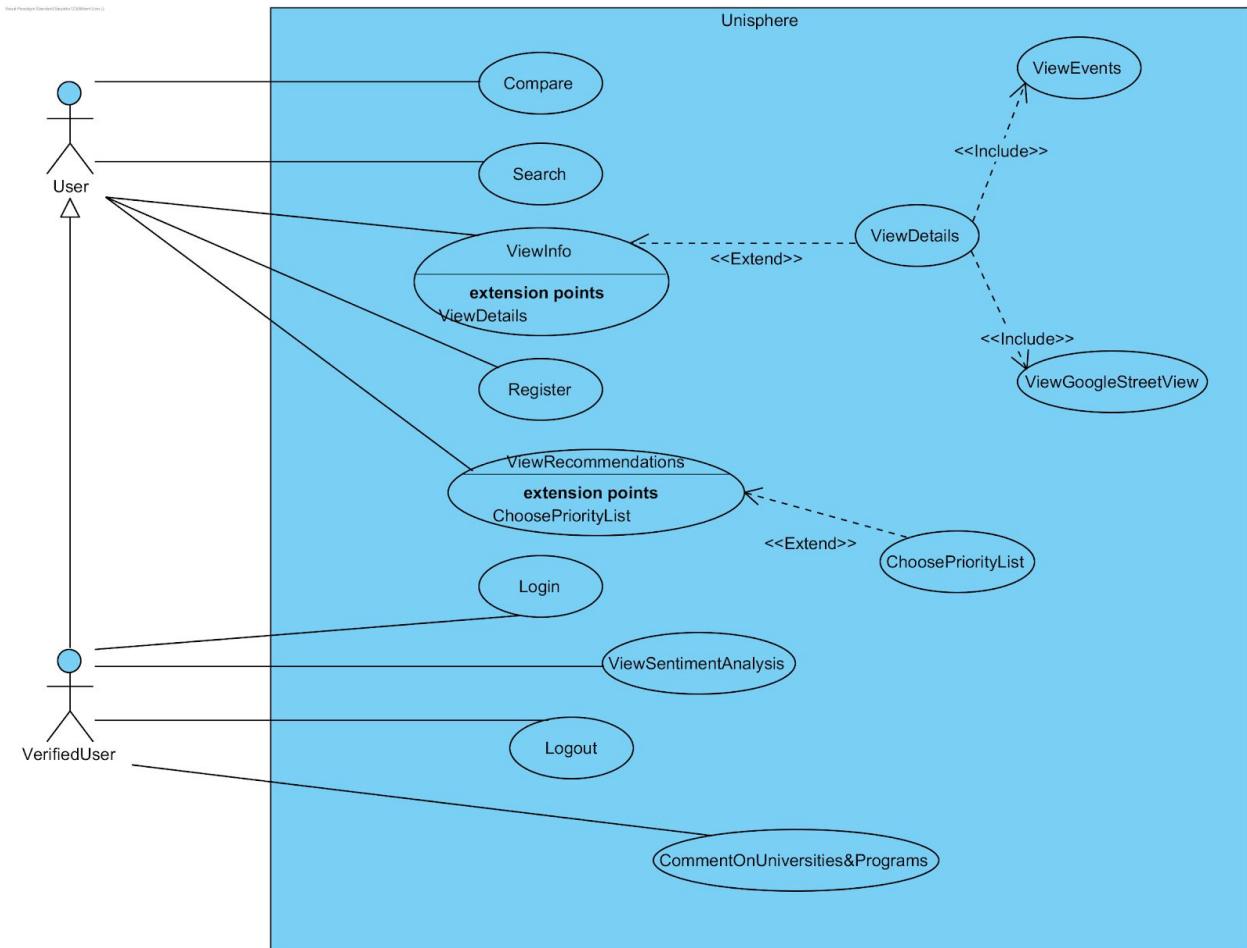


Figure 1: Use Case Diagram

3.5.3 Object and Class Model

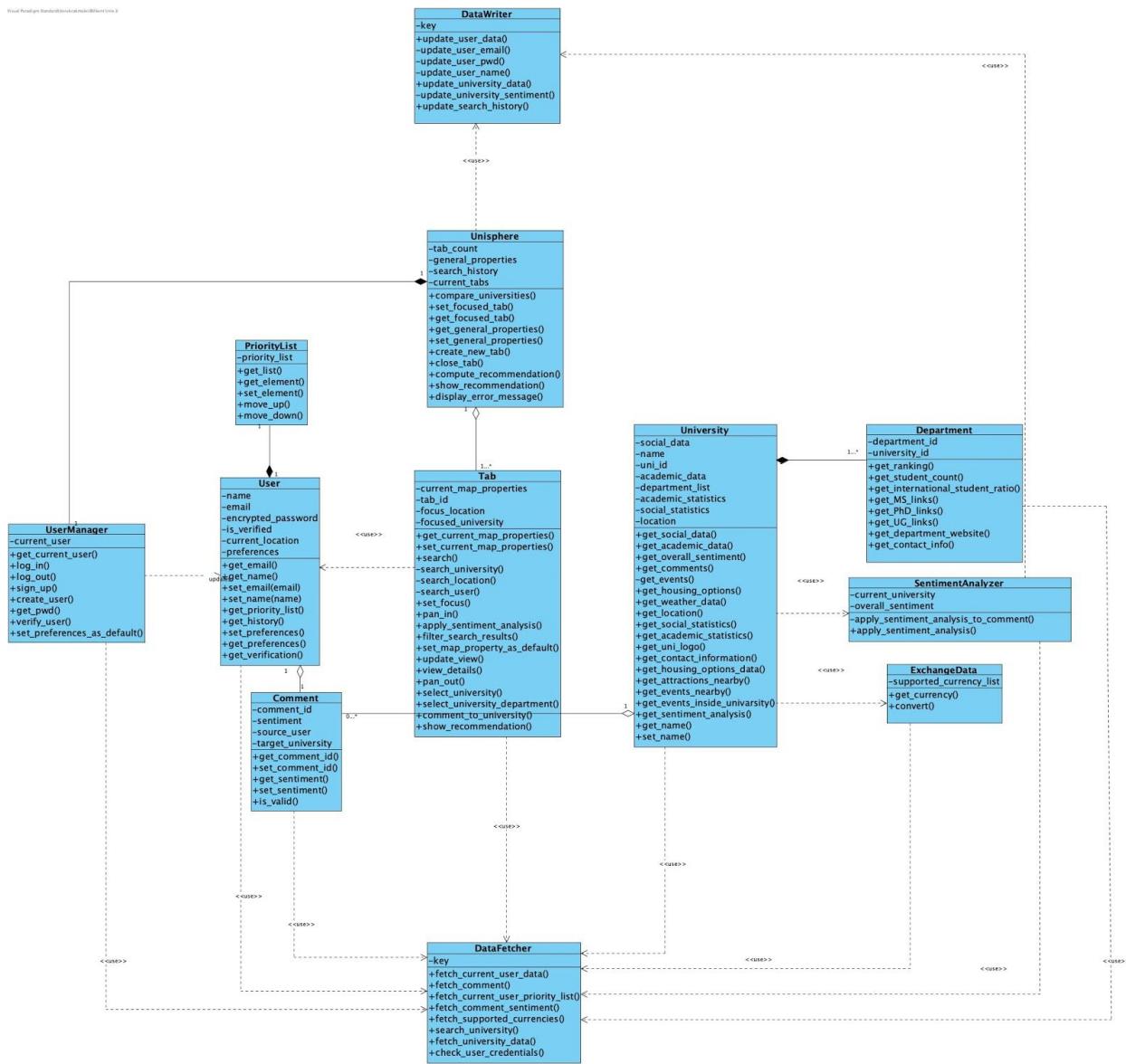


Figure 2: Class Diagram

Class Descriptions

Unisphere

Unisphere class is the main class of the project. All opera

User

Holds information and methods that let access to these information. The information held by this class is not directly requested from and used by objects of this class but rather the UserManager class is used for this purpose. In other words; this class acts as a gateway for any operation regarding the user's information.

PriorityList

Holds the priority information per user. These priority information can be weather conditions of the location of the university, academic reputation of the university, the attractions near the university, the average price of the university programs and much more.

UserManager

This class is the controller class for the User instances. The alteration of any login credentials will be done by UserManager.

Comment

This class is responsible for representing comments and holding information regarding these comments made by users about universities. Although a given comment object is "aware" of its author it has no knowledge of which university it is listed under. Universities will rather hold a list of comment ID's.

Tab

Tab class adds the multi-processing functionality for a user. User can search different universities in different tabs. This class also holds the information to visualize map. The operations regarding map usage and navigation are done on a tab so the Tab class is also responsible for shifts in focus on the map and other operations such as yaw, pitch, roll, rotate and zoom.

University & Department

University and_Department classes,together, are the Unisphere representation of a university.

SentimentAnalyzer

This class does sentiment analysis to the tweets and comments related to the university under consideration.

ExchangeData

This class contains the needed data for converting currencies to the currency under consideration for housing prices, school fees or any other attraction on the map that may require additional fees and purchases to be made such as transportation, events, food, etc.

DataFetcher & DataWriter

These classes are the façade for database access. Every database access is undertaken by DataFetcher and DataWriter.

3.5.4 Dynamic Models

3.5.4.1 Sequence Diagrams

3.5.4.1.1 Compare Universities Diagram

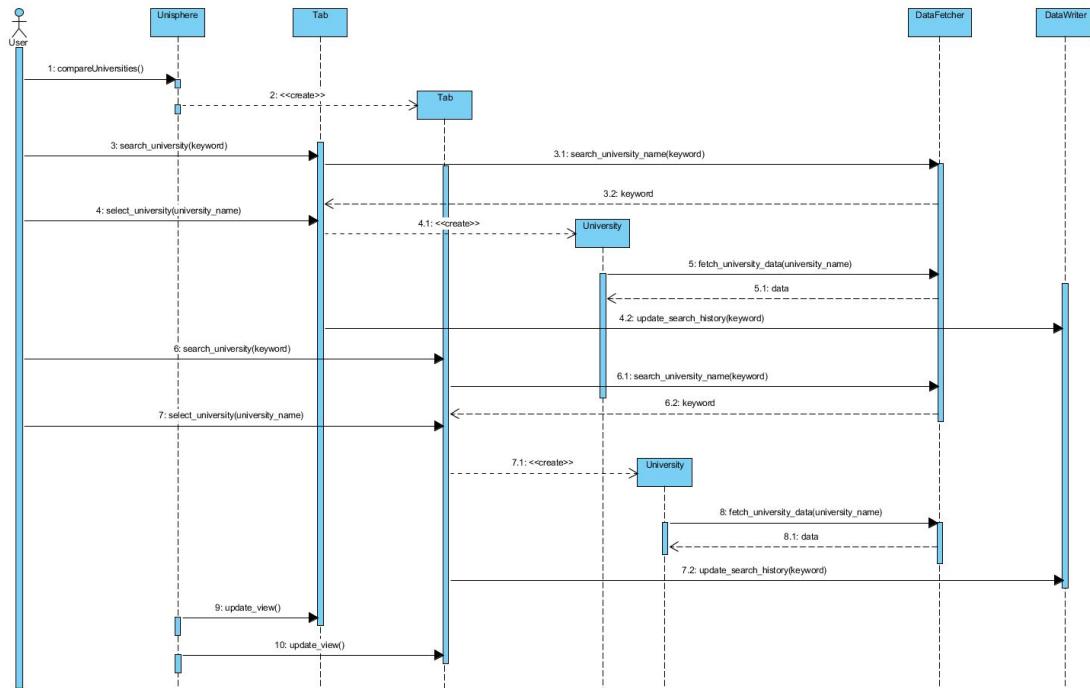


Figure 3: Compare University Sequence Diagram

User clicks compare button on the current tab. The current screen splits into two. There are two tabs and two search bars on the webpage. User selects a university on each bar, a new University object is created for each tab. The data is fetched from the database. User search history is updated in the database, the last searches are added. The view on each tab is updated.

3.5.4.1.2 Search University Diagram

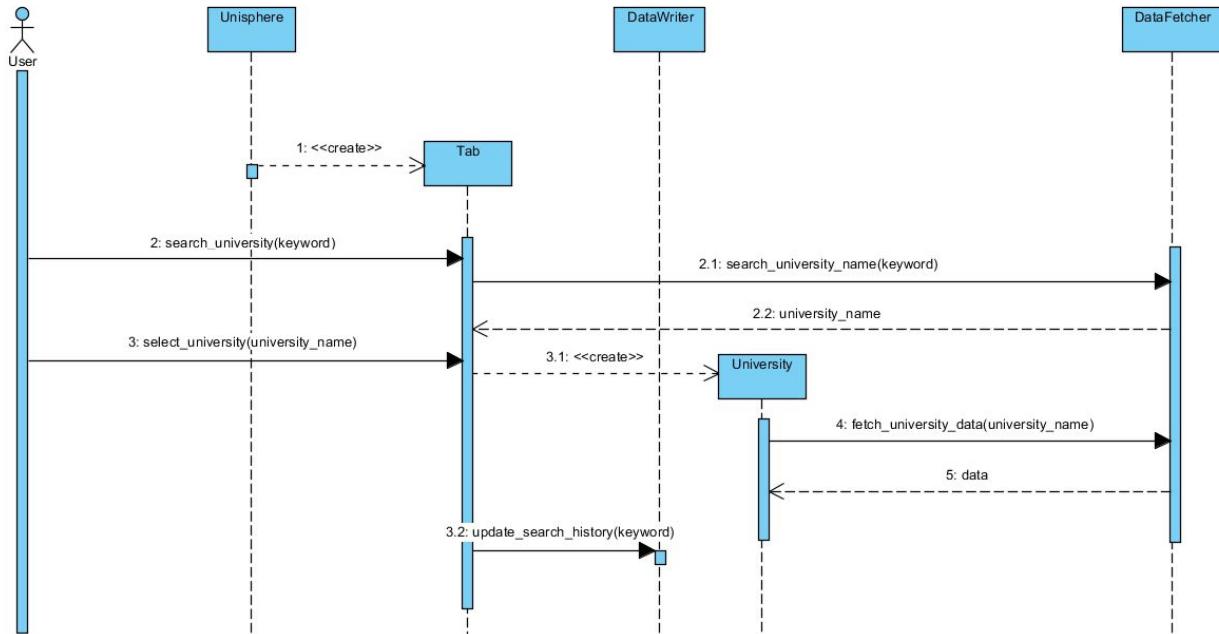


Figure 4: Search University Sequence Diagram

User searches for a university by using the search bar on the home page. The general information of the university, which includes, university logo, name, location, general academic info, is created in University object. The matching universities with the keyword are listed.

3.5.4.1.3 View Information Diagram

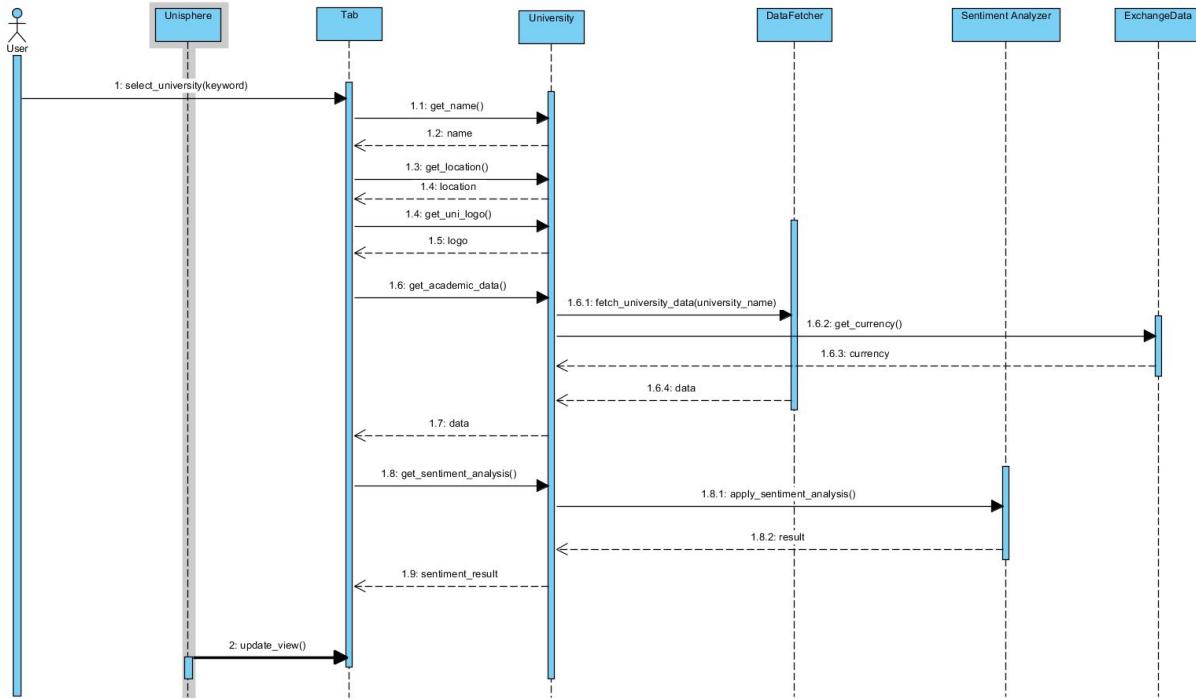


Figure 5: View Information Sequence Diagram

User selects the university from search results. University information is viewed on the same tab, that is created before. The name, location, logo and general information is viewed. The tuition fee of the university is calculated according to the up to date currency. Sentiment analysis results created and viewed. Lastly, the view of the tab is updated, detailed academic information of the university is displayed.

3.5.4.1.4 View Events Diagram

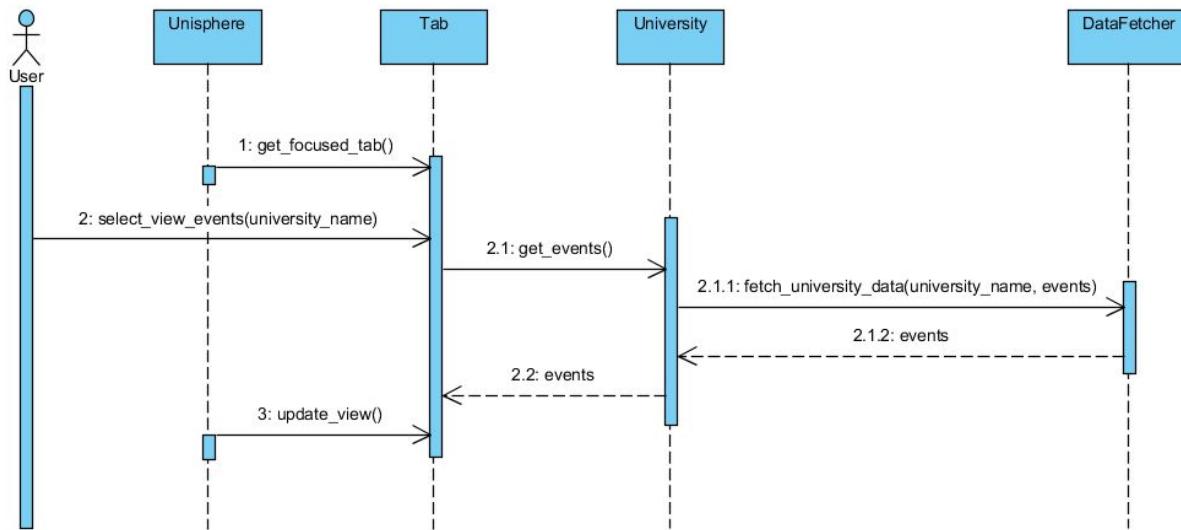


Figure 6: View Events Sequence Diagram

User clicks on the “View Events” button. The social events, which includes both the social events near the university and inside of the university, are viewed on the same tab, that is created before.

3.5.4.1.5 Select Department Diagram

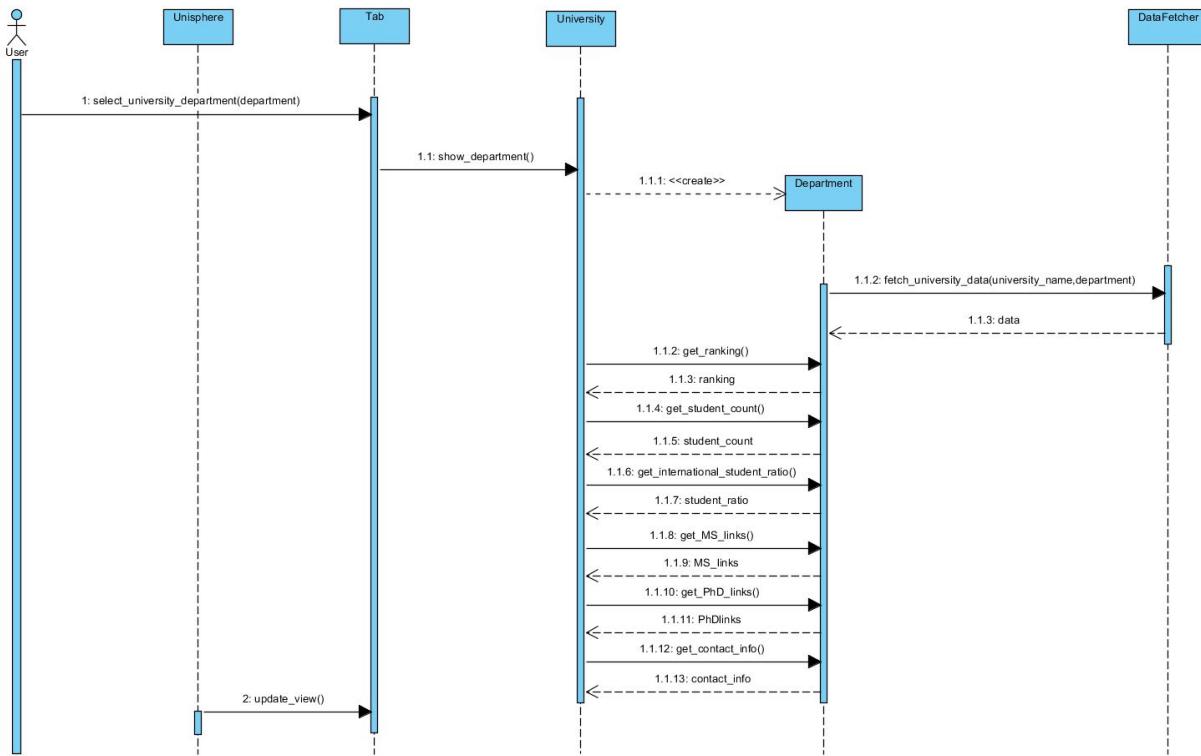


Figure 7: Select Department Sequence Diagram

User selects the department of the university through the list of universities on the information screen. The detailed information of the department, such as MS and PhD programs of the university, is displayed.

3.5.4.1.6 Login Diagram

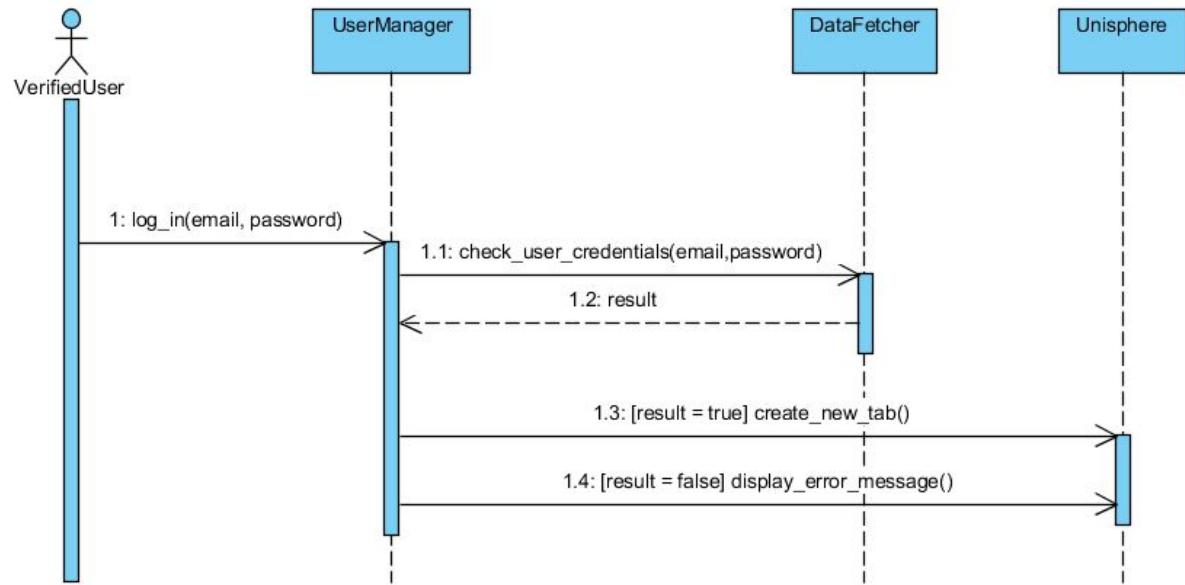


Figure 8: Login Sequence Diagram

Verified User fills in the mail address and password sections of the login box and then clicks on the “Login” button. The existence of user credentials is checked. If the credentials are correct, then the Verified User navigated to the home screen. Otherwise, an error message is displayed.

3.5.4.1.7 Sign-up Diagram

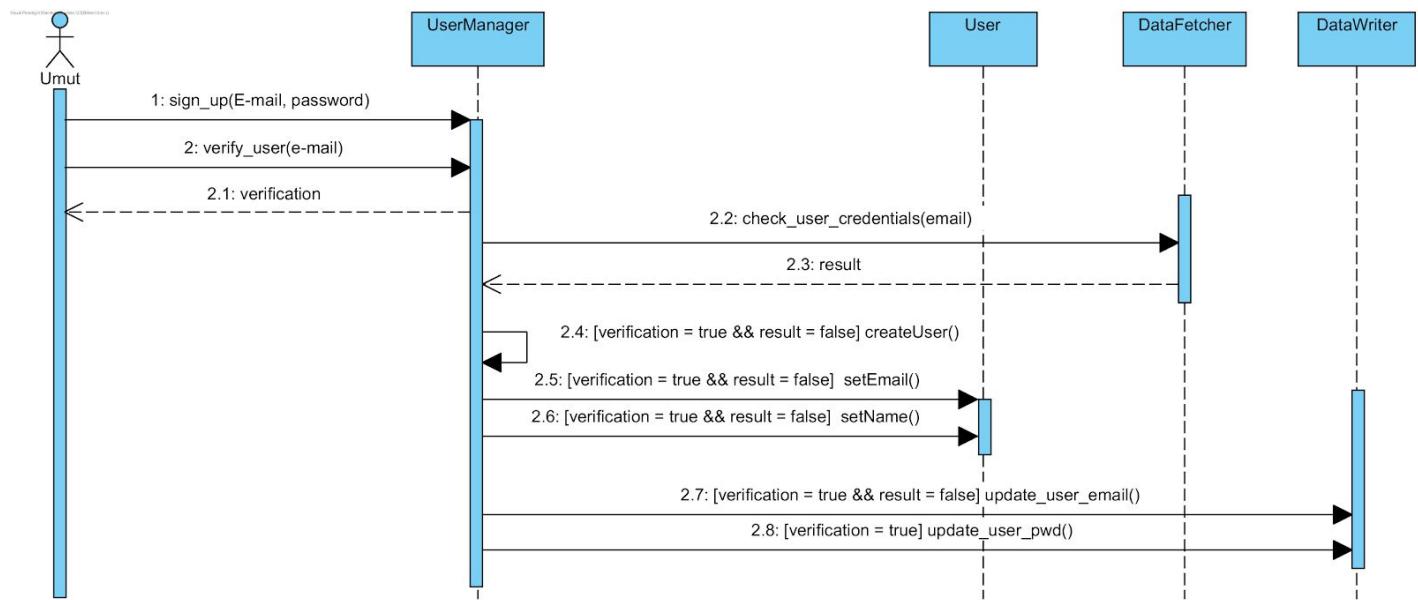


Figure 9: Sign-up Sequence Diagram

User initially clicks on the “Register” button and then enters his/her school mail address and password. If his/her school mail address is legitimate and unique, and password complies to the rules, then the account is created. User is navigated to the home screen.

3.5.4.1.8 Comment on Universities Diagram

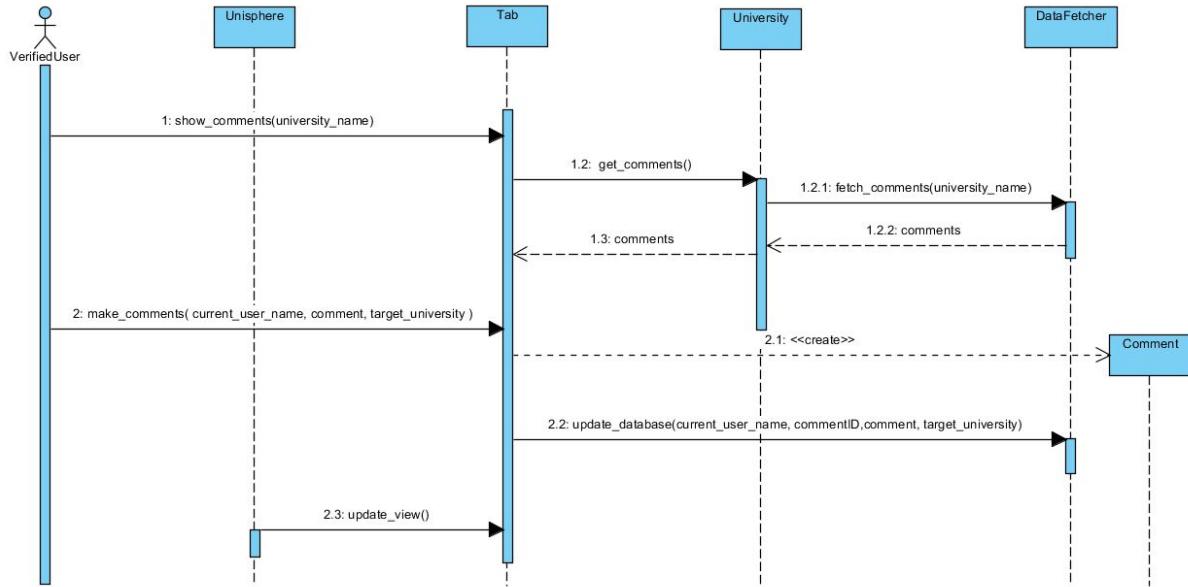


Figure 10: Comment on Universities Sequence Diagram

Verified user, who has a legitimate email address and signs up to the system, can make comments to universities and Ms and PhD programs. User writes comment in comment box and clicks the make comments button on the Information screen of selected university. A new comment is generated by the system and added to the database. Lastly, the view is updated.

3.5.4.1.9 View Recommendations Diagram

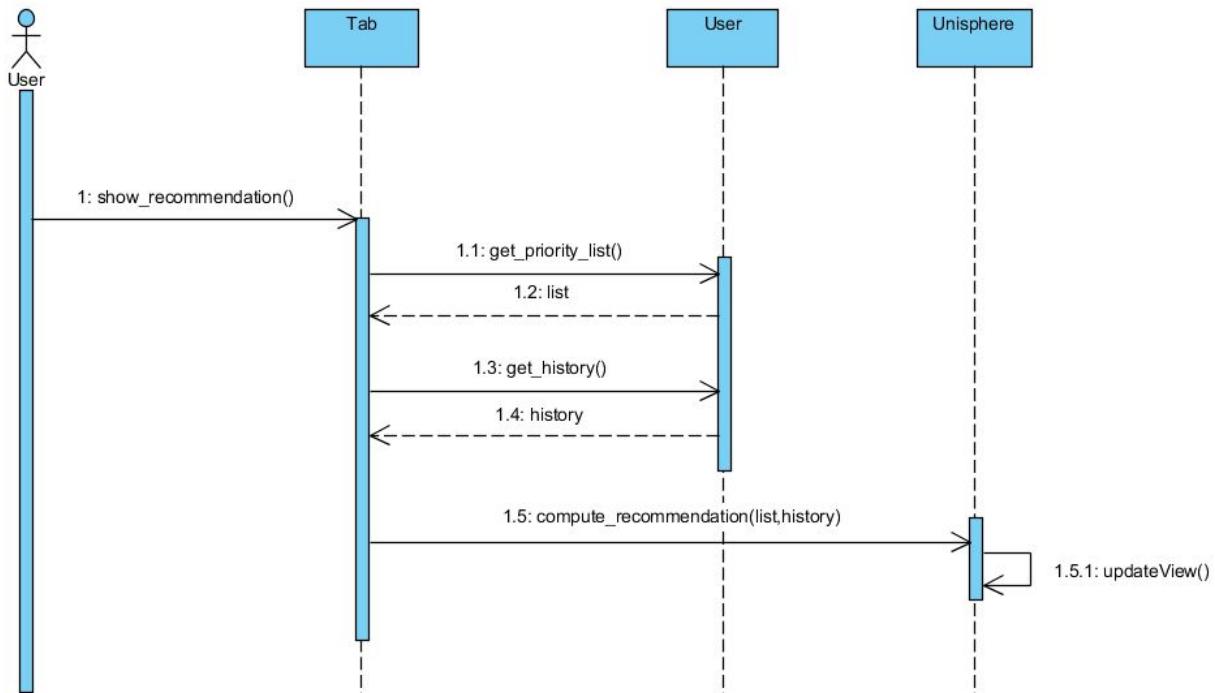


Figure 11: View Recommendations Sequence Diagram

User clicks on the “Get Recommendations” button. User is navigated to the recommendations screen. By looking the user’s priority list and his browsing history, Unisphere creates a recommendation list and show it to the user.

3.5.4.1.10 View Nearby Attraction Diagram

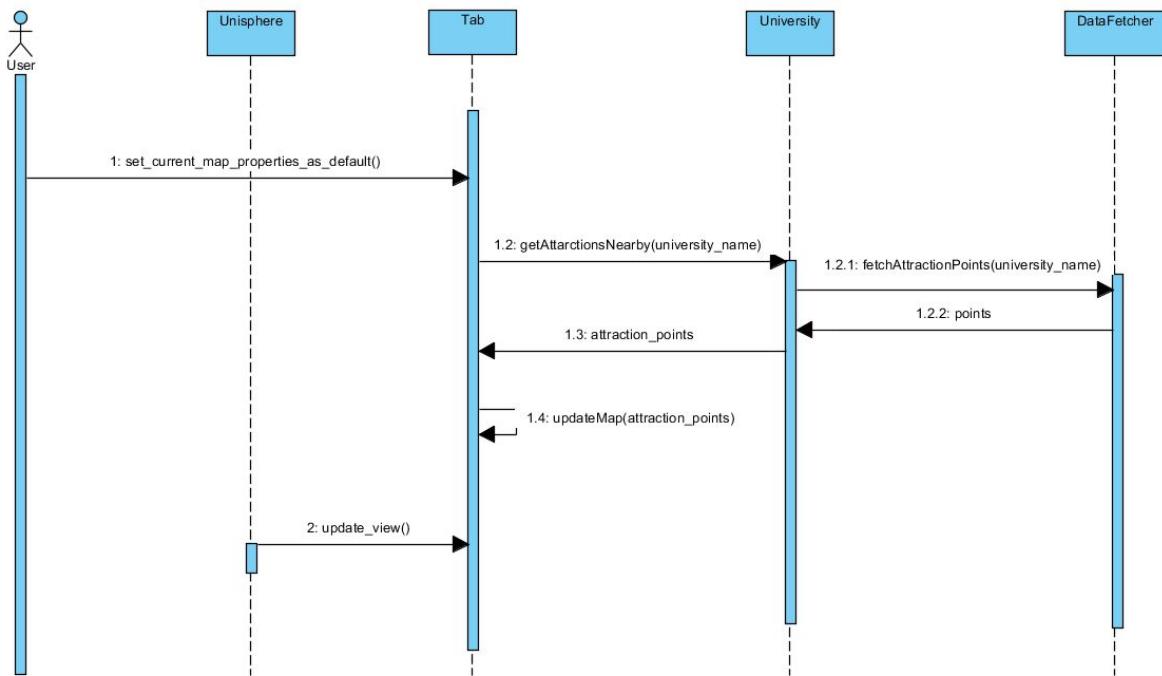


Figure 12: View Nearby Attraction Sequence Diagram

User previously selects the university from the search results. The detailed information of the university is provided. The nearby attractions are viewed on the map, such as the museums, shopping centers etc. The data points on the map is fetched from the database, which is initialized by using Google Maps API.

3.5.4.1.11 Select Priority List Diagram

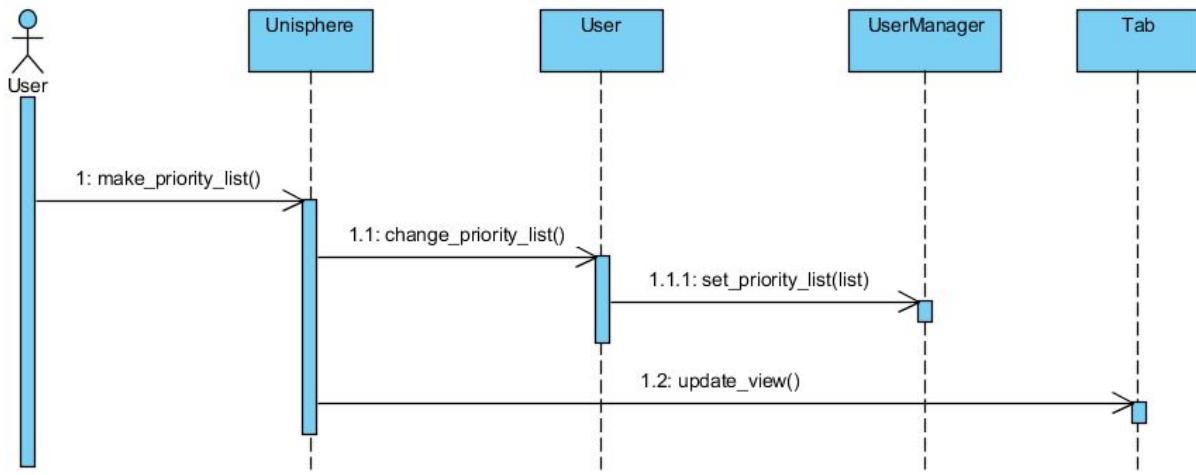


Figure 13: Select Priority List Sequence Diagram

User selects make priority list button from home page of the site. Then he/she puts in order his/her priorities from a predefined list and saves it.

3.5.4.2 Activity Diagram

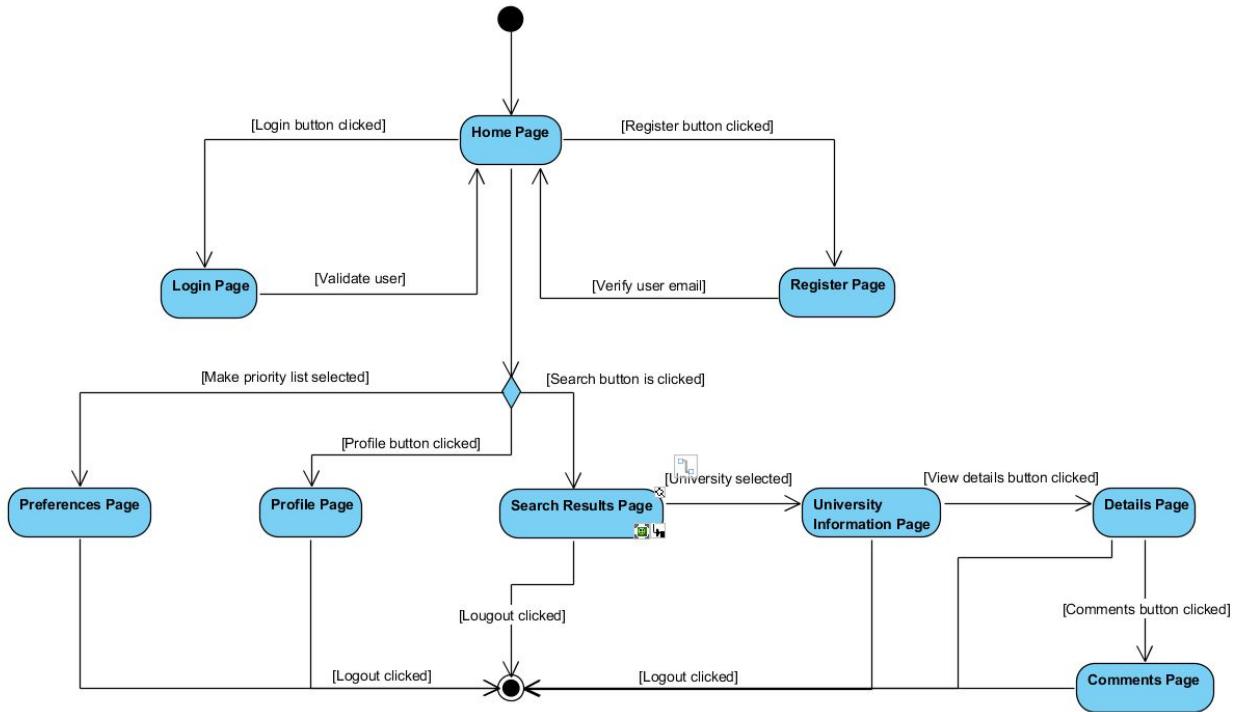


Figure 14: Activity Diagram

The general flow through the pages is demonstrated in the activity diagram. User can access the general properties of the application. Therefore, any user can access the home page and navigate through the pages. If they click to the sign up button on the home page, they can register to Unisphere by using their university email accounts. If they had already registered, they can login in using the login button. Afterwards, user can create priority list by pressing “makePriorityList” button. Besides, they can click search button and they will be navigated to the Search Results Page. Users can select a university and get further information about the university on University Information Page. They can click to view details button and access both academic and non academic information on the Details Page. They will see the visual results on maps and charts. Furthermore, they can access variety of comments about the selected university, which gathers the comments from Twitter, Reddit and as well as the comments left in Unisphere. From each page, user can click to log out and leave the webpage.

3.5.5 User Interface

In this section of the report, early sketches of what the UI might look like will be presented. A brief description for each image will provide context as to how these UI sketches relate to the overall flow of the application.

3.5.5.1 Home Page

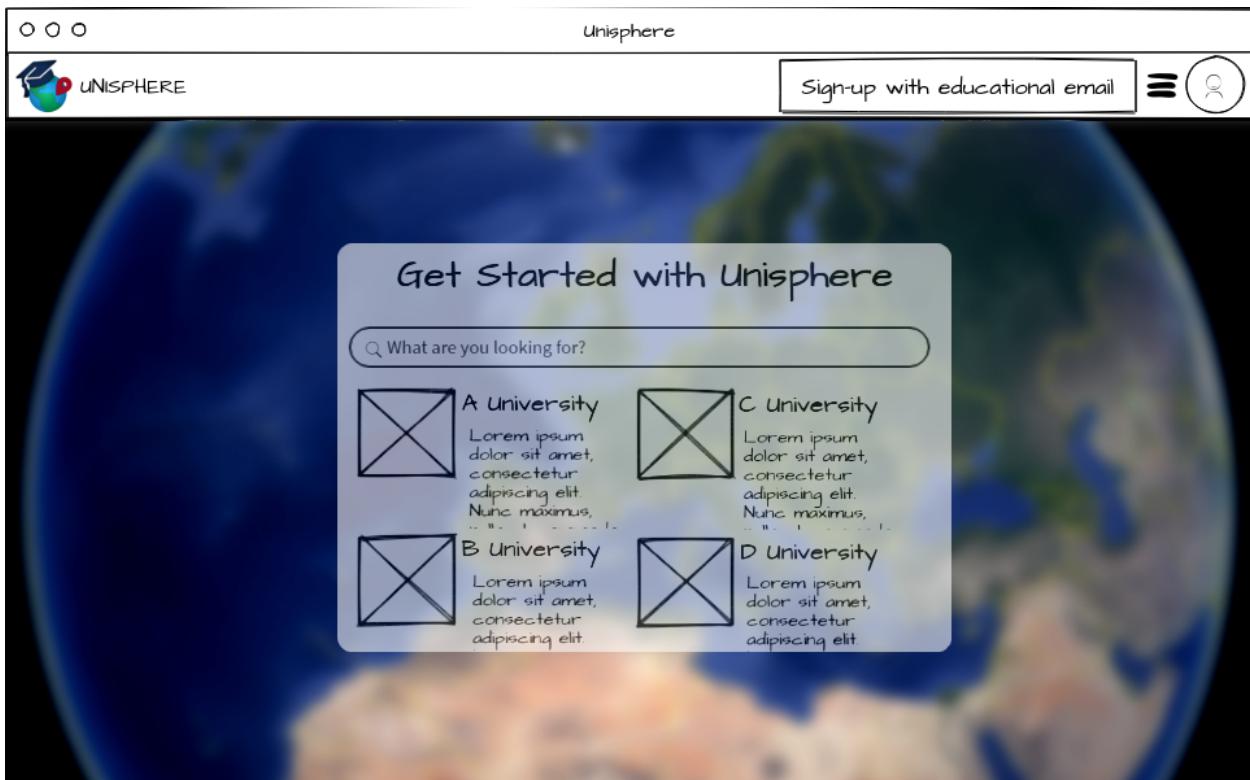


Figure 15: Home Page

This is the first screen the user sees when they first enter the Unisphere app. It provides a simple UI elements that will help the uninitiated user get going. It suggests some universities to check out for both undergrad and postgrad. It also hosts a search bar so that the user can start searching for any university, place, department etc.

3.5.5.2 Register Page

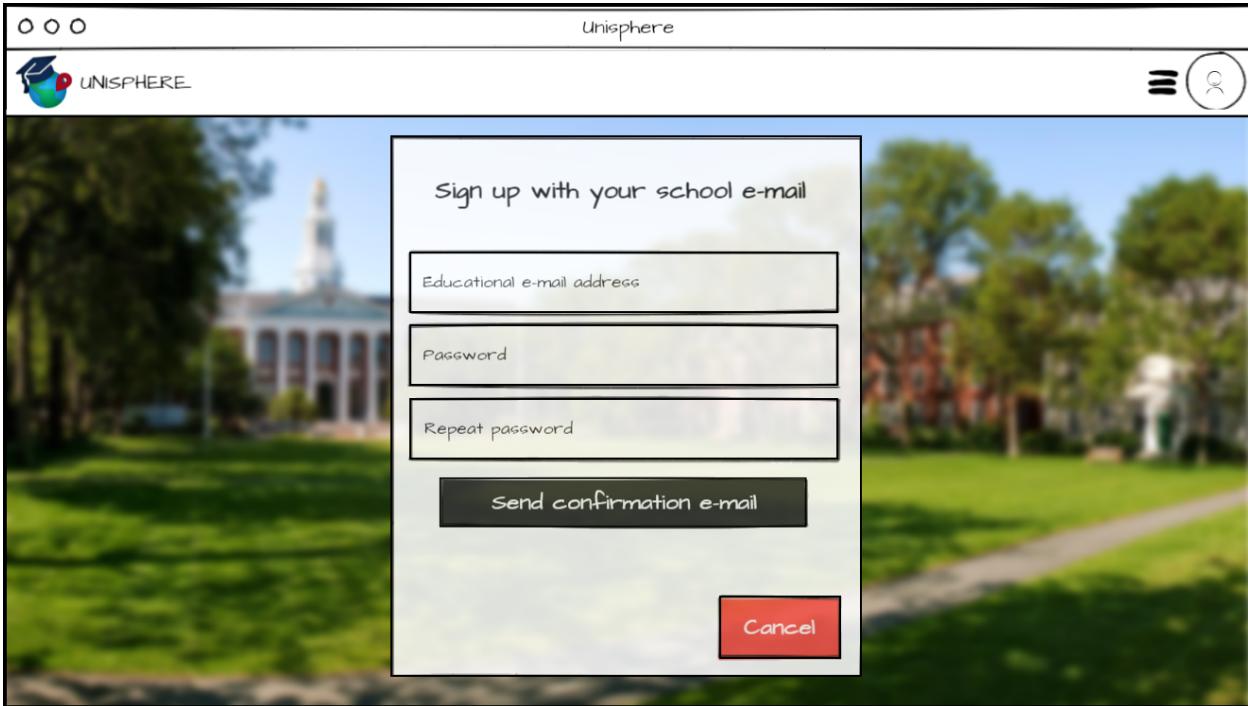


Figure 16: Register Page

Users can sign-up using their educational email addresses to be able to do certain exclusive actions within the Unisphere app. Unisphere will send a confirmation email to their educational email address and will make sure that the email entered is off a legitimate higher educational institute. Once the confirmation link is clicked by the user, Unisphere will verify the user on the backend and the user can now log-in using their credentials.

3.5.5.3 Login Page

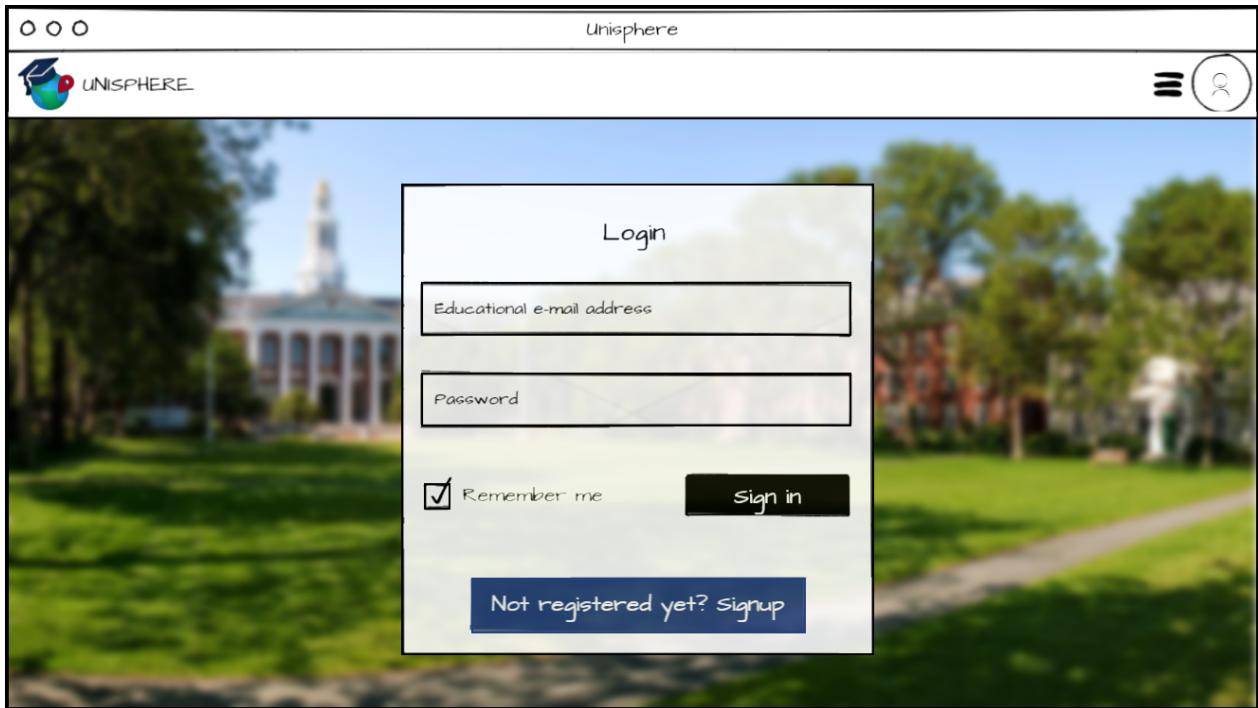


Figure 17: Login Page

After registering or logging out users can log-in using the credentials of a verified user. This screen also provides a button to register if the user has not done so already. Users can use the remember me checkbox to enable cookie usage, so that they won't have to log in everytime they want to access the exclusive actions and features.

3.5.5.4 University Information Page

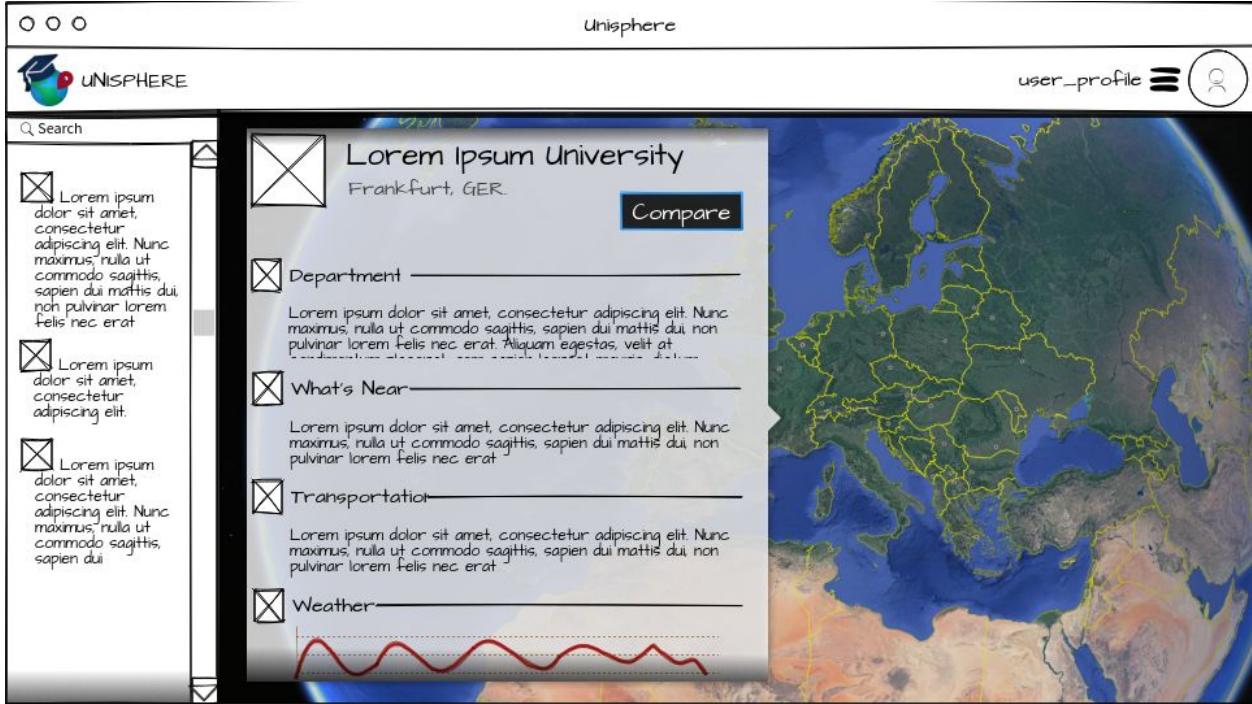


Figure 18: University Information Page

In the main information screen, users can simultaneously search for tags or university names and drag and drop them on top of tags to navigate to them on the map. The initial displayed information will be a pop-up and will include brief descriptions for certain data-points (of which are yet to be determined). The pop-up has a compare button which will open a second tab in split screen view so that another university can be viewed with the current one side-by-side. The UI elements will shift towards the left side of the screen smoothly and make place for the new tab that includes its own globe. The tabs share the search bar.

3.5.5.5 Compare Page

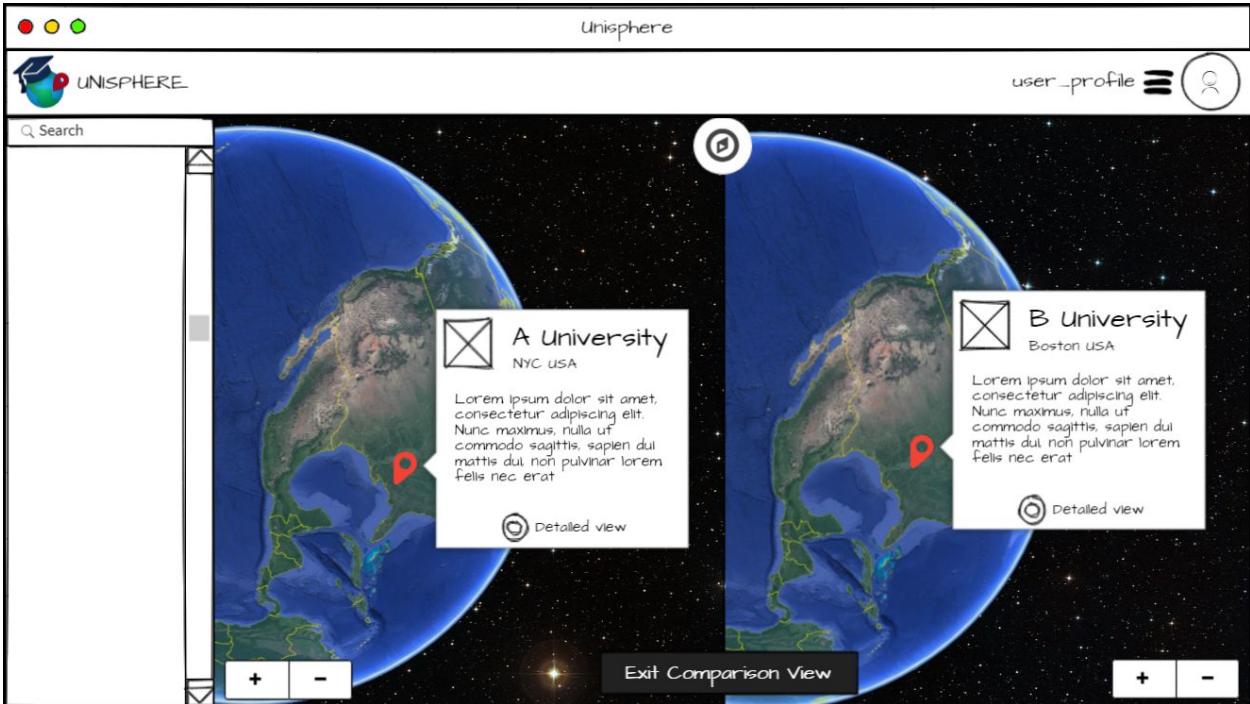


Figure 19: Compare Page

The dual-tab view provides a fast comparison opportunity for the user. Initially only brief descriptions of the universities that are being viewed are displayed. Upon user command these pop-up windows can be expanded, from which more detailed information will be present. The dual tab view has a zoom function where the user can zoom in and out of the globe. This screen will also have additional map control UI elements such as a scale (legend), pitch, yaw, rotate and a reset button to reset the view to the Unisphere default view.

3.5.5.6 Compare Page - 2

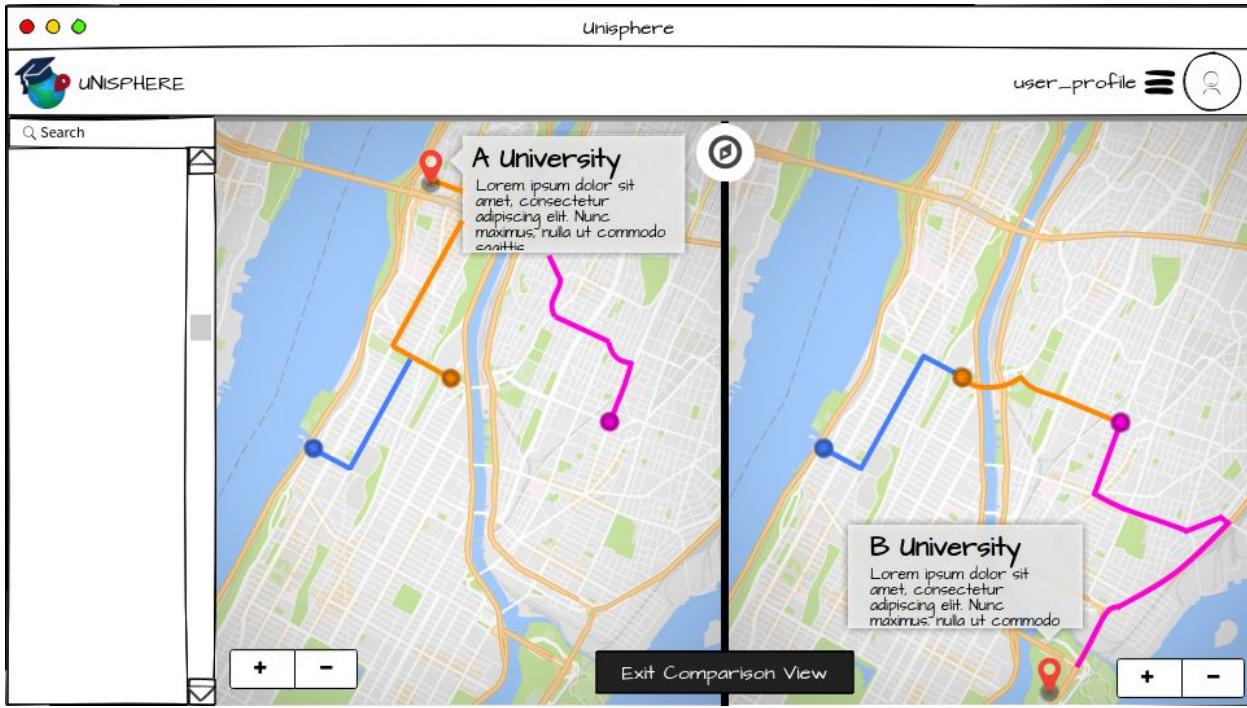


Figure 20: Compare 2 Page

The side-by-side map view screen allows users to see what is near the university's vicinity. These include but are not limited to food options, cinema, nightlife, hospital, transportation options or any other attraction that is tagged by the user in their priority list. The pop-ups still provide a way to see the university names and brief descriptions but the user has to further expand the view to be able to compare detailed information side-by-side.

3.5.5.7 Compare Page(Detailed View)

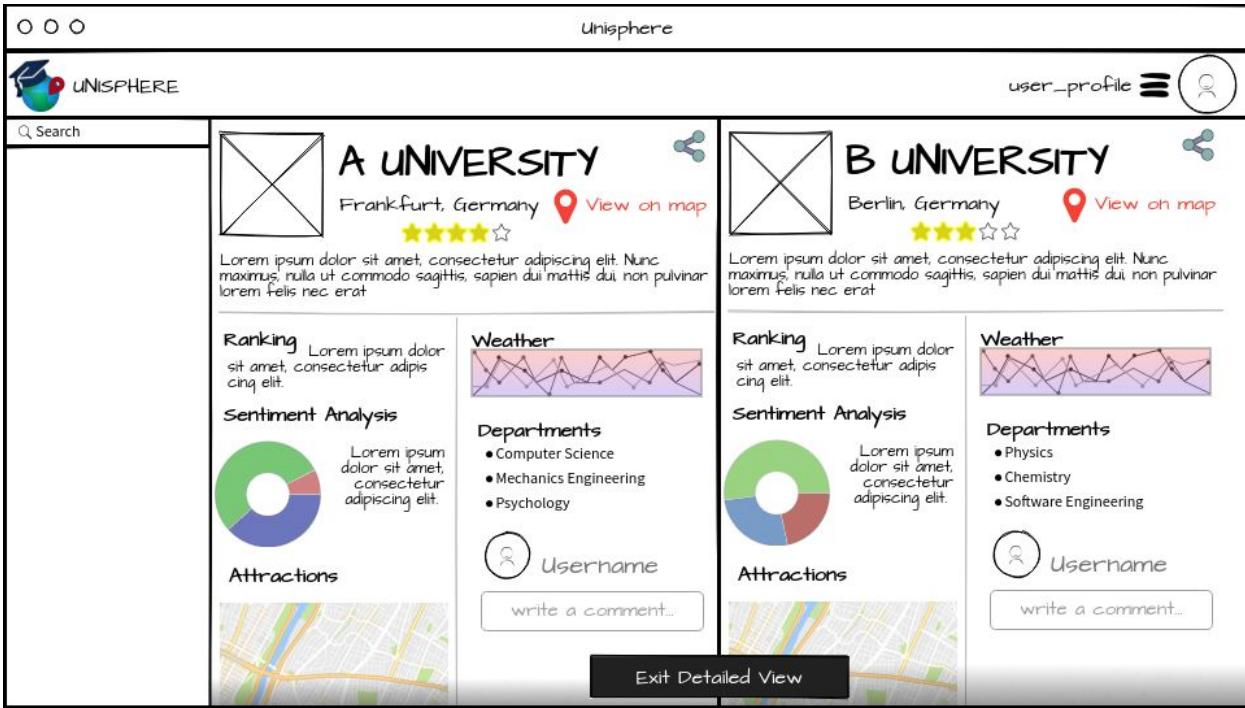


Figure 21: Compare Page(Detailed View)

The direct detailed comparison screen hosts the most amount of information in one place. Here university's name, location, logo and brief description is presented on the top header. Under this header is the main body which hosts the various data-points Unisphere deals with such as various ranking data, weather data, sentiment analysis summary done by Unisphere, department information, comment section, a preview of the map view and possibly more. Any of these elements are clickable and will expand to a full page upon being clicked which will then present a full page view of all the details and data available on Unisphere.

5 References

- [1] "Street View Service," Google, 25-Sep-2018. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/streetview>. [Accessed: 12-Nov-2018].
- [2] "Maps JavaScript API," Google, 25-Sep-2018. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/tutorial>. [Accessed: 12-Nov-2018].
- [3] "Eventbrite APIv3 Developer Documentation," Eventbrite APIv3 Developer Documentation - Eventbrite Developer Center. [Online]. Available: <https://www.eventbrite.com/developer/v3/>. [Accessed: 12-Nov-2018].
- [4] "Keyword Insights - Twitter Developers," Twitter. [Online]. Available: <https://developer.twitter.com/en/docs/ads/audiences/api-reference/keyword-insights.html>. [Accessed: 12-Nov-2018].
- [5] reddit.com: api documentation. [Online]. Available: <https://www.reddit.com/dev/api/>. [Accessed: 12-Nov-2018].
- [6] OpenWeatherMap.org, "Weather API," openweathermap. [Online]. Available: <https://openweathermap.org/api>. [Accessed: 12-Nov-2018].
- [7] Leereilly, "leereilly/swot," GitHub, 02-Apr-2018. [Online]. Available: <https://github.com/leereilly/swot>. [Accessed: 14-Oct-2018].
- [8] "Custom Search JSON API | Custom Search | Google Developers," Google. [Online]. Available: <https://developers.google.com/custom-search/json-api/v1/overview>. [Accessed: 14-Oct-2018].
- [9] "Charts | Google Developers," Google. [Online]. Available: <https://developers.google.com/chart/>. [Accessed: 14-Oct-2018].
- [10] "Cloud Translation API Documentation | Cloud Translation API | Google Cloud," Google. [Online]. Available: <https://cloud.google.com/translate/docs/>. [Accessed: 14-Oct-2018].