Bilkent University

Department of Computer Engineering

CS 491- Senior Design Project I

Unisphere: Global University Catalog

Project Specifications 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

Project Specifications Report

October 15, 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

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 brief description of the project will be explained in detail. Afterwards, project constraints specifically economic, maintainability, social, implementation and time will be clarified. The professional and ethical issues will be discussed. Lastly, functional and nonfunctional requirements will be pointed out.

1.1. Description

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.

1.2. Constraints

1.2.1. Economic Constraints

- The API's, frameworks and libraries needed to develop this project are free(e.g. Google Maps API [1] for showing the universities, Google Street View API [2] for showing universities via Street View).
- The domain for the website of this project is 12.98$/year
- The web hosting for the domain is based on usage.
- Since there is a login system(even though login operation is not a necessity), the user credentials must be stored. To serve this purpose, Amazon DynamoDB [3] will be used. The pricing is dependent to the usage amount.
- The sentiment analysis of tweets and subreddits about a university will be done on cloud. To serve this purpose, Amazon Lambda [4] will be used. As mentioned above, the pricing is dependent to usage.
- As a future work, this web application can be implemented for Android platform and in order to publish this application on Google Play, 25$ fee need to be paid one time.

1.2.2. Maintainability Constraints

- User generated data and the data which is made available manually need to be updated time to time.
- The domain(s) for project website & documentation website need to be re-registered yearly.
- Other than data points per university that are not updated dynamically, there are some data points that have to be updated on a yearly basis such as rankings, tuition fee, dorm and meal fees etc.

1.2.3. Social Constraints

This application will allow some user generated content (text based and possibly image content in the future). There will be no system to check for explicit or Not-safe-for-work (NSFW) content automatically however there will be a “Report” functionality available to all users of the platform wherever there is content to do so; to report possibly offensive content. The report functionality can also be used in places were content that would interfere with the natural interaction of users such as advertisements, spam and re-posts.

1.2.4. Implementation Constraints

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[8], 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[9] 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 [10] will possibly be utilized in order to further retrieve and categorize data about a given university. Google Chart API [11] could be utilized in order to better visualize data. Google Cloud Translate API [12] 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 [13].

- EventBrite API [14] will be used in order to find out events and event history around and inside a given university.

1.3. Professional and Ethical Issues

The professional and ethical issues related to Unisphere is discussed regarding the data retrieval from websites and user specific information. Some of the data needed for our project are not readily available as a database. To use these data, either the data must be crawled or copied manually. For instance, the data for university rankings are not publicly distributed therefore, they need to be copied manually to our database. This operation is not restricted under laws if the data are appropriately referenced. Users private information will not be shared or distributed. The email - domain will be used to check whether it exists or not by using SWOT API. Confirmation email will be send by our application.

2. Requirements

2.1. Functional Requirements

2.1.1. 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.

2.1.2. 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.

2.2. Non-Functional Requirements

2.2.1. 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.

2.2.2. 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.

2.2.3. 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.

2.2.4. 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.

2.2.5. 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.

2.2.6. 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.

REFERENCES

[1]“Maps JavaScript API,” Google, 25-Sep-2018. [Online]. Available: https://developers.google.com/maps/documentation/javascript/tutorial. [Accessed: 14-Oct-2018].

[2] “Street View Service,” Google, 25-Sep-2018. [Online]. Available: https://developers.google.com/maps/documentation/javascript/streetview. [Accessed: 14-Oct-2018].

[3] “Amazon DynamoDB - Overview,” Amazon. [Online]. Available: https://aws.amazon.com/dynamodb/. [Accessed: 14-Oct-2018].

[4] “AWS Lambda – Sunucusuz Bilgi İşlem – Amazon Web Services,” Amazon. [Online]. Available: https://aws.amazon.com/tr/lambda/. [Accessed: 14-Oct-2018].

[5] “React – A JavaScript library for building user interfaces,” – A JavaScript library for building user interfaces. [Online]. Available: https://reactjs.org/. [Accessed: 14-Oct-2018].

[6] “Droplets on DigitalOcean - More than just virtual machines,” DigitalOcean. [Online]. Available: https://www.digitalocean.com/products/droplets/. [Accessed: 14-Oct-2018].

[7] Leereilly, “leereilly/swot,” GitHub, 02-Apr-2018. [Online]. Available: https://github.com/leereilly/swot. [Accessed: 14-Oct-2018].

[8] “Keyword Insights - Twitter Developers,” Twitter. [Online]. Available: https://developer.twitter.com/en/docs/ads/audiences/api-reference/keyword-insights.html. [Accessed: 14-Oct-2018].

[9] reddit.com: api documentation. [Online]. Available: https://www.reddit.com/dev/api/. [Accessed: 14-Oct-2018].

[10] “Places | Google Maps Platform | Google Maps Platform | Google Cloud,” Google. [Online]. Available: https://cloud.google.com/maps-platform/places/. [Accessed: 14-Oct-2018].

[10] “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].

[11] “Charts | Google Developers,” Google. [Online]. Available: https://developers.google.com/chart/. [Accessed: 14-Oct-2018].

[12] “Cloud Translation API Documentation | Cloud Translation API | Google Cloud,” Google. [Online]. Available: https://cloud.google.com/translate/docs/. [Accessed: 14-Oct-2018].

[13] OpenWeatherMap.org, “Weather API,” openweathermap. [Online]. Available: https://openweathermap.org/api. [Accessed: 14-Oct-2018].

[14] “Eventbrite APIv3 Developer Documentation¶,” Eventbrite APIv3 Developer Documentation - Eventbrite Developer Center. [Online]. Available: https://www.eventbrite.com/developer/v3/. [Accessed: 14-Oct-2018].