UAtilize: Interactive Visualization & Analysis of Campus Building Utilization

Suphanut Jamonnak, Bharani Anne, Nikhil Preeth, En Cheng

Department of Computer Science,
The University of Akron, Akron, OH

Abstract- With the increasing popularity of Google Maps, the integration of web services with Google Maps has recently attracted considerable attention. Using Google Maps JavaScript API v3, developers can build highly customizable maps with their own content and imagery. In this paper, UAtilize -- an Interactive Visualization & Analysis of Campus Building Utilization application using Google Maps and Google Fusion Tables is presented. UAtilize provides multiple functions and dynamic visualizations of campus building utilization data and Zipcard transaction data, leveraging the comprehensiveness, accuracy, and usability of Google Maps. UAtilize is capable to integrate and transform geographical data into a map. The primary goal of UAtilize is to assist several departments at The University of Akron, including Registrar Office, Parking and Transportation Services, Police Department, and Auxiliary Business Operations. With UAtilize, users can directly and interactively analyze the data using Google Charts and Google Visualization, instead of querying relational databases.

Keywords: Data Integration, Visualization & Analysis, Google Maps, Google Fusion Tables

I. Introduction

The invention of the Internet and the emergence of the World Wide Web revolutionized our daily lives. Thanks to advanced technologies such as computers, satellites, wireless sensors, tablets, smart phones, we have been collecting tremendous amounts of data on a daily basis, because we believe that information leads to success. More data has been created in the last three years than in all the past 40,000 years, the total amount data will quadruple in the next two years, said Stephen A. Brobst, chief technology officer of Teradata Corporation [1], at The Data Warehouse Institute (TDWI) [2] World Conference in 2012. The explosion of data requires the ability to store, secure, and manage the physical data, DAMA International [3] president John Schley said, but it also demands that the stored data be useful and meaningful. Efficient database management systems have been very important assets for storing and managing a large corpus of data and especially for efficient retrieval of particular information from a large collection whenever needed. Nowadays, we have far more data than we can handle: from scientific data and business transactions, to satellite pictures, electronic patient records and clinical reports. Information retrieval is simply not sufficient anymore for decision-making. Such a situation has given rise to the emergence of new needs including automatic summarization of data, extraction of the "essence" of information stored, discovery of patterns in raw data, and interactive visualization of data.

Recently, cloud-based services have been playing an important role in large-scale Web-based applications. Google Fusion Tables [4] is a cloud-based service for data management and integration. Launched in 2009, Fusion Tables has received considerable use. Fusion Tables enables users to upload tabular data files (spreadsheets, CSV, KML), currently of up to 250 MB space per table. It supports the integration of data from multiple sources by performing joins across tables that may belong to different users. Also, Fusion Tables has the ability to filter and aggregate the data and provides multiple ways of visualizing the data (e.g., charts, maps, and timelines). Google Maps [5] is one of the increasingly popular web mapping services which have been used in a wide range of areas including real estate, tourism, and weather forecast. Using Google Maps JavaScript API v3 [6], developers can build highly customizable maps with their own content and imagery. Google Maps provides geospatial visualization of information so that users can analyze and

understand the relationship between data and geographic location. It provides a novel option to visualize and analyze data.

In this paper, we present an interactive visualization & analysis of UA campus building Utilization application, named UAtilize, using Google Fusion Tables and Google Maps. The primary goal of UAtilize is to assist several departments at The University of Akron (UA) [7], including Registrar Office, Parking and Transportation Services, Police Department, and Auxiliary Business Operations. UAtilize provides multiple functions and dynamic visualizations of campus building utilization data and Zip Card transaction data, leveraging the comprehensiveness, accuracy, and usability of Google Maps. UAtilize is capable to integrate and transform geographical data into a map. With UAtilize, users can directly and interactively analyze the data using Google Charts and Google Visualization, instead of querying relational databases. For example, Parking and Transportation Services can easily estimate the campus traffic and schedule parking space accordingly with the dynamic visualizations of campus building utilization provided by UAtilize. Another example, in case of campus emergencies, policy department can use UAtilize to analyze each building's priority and take actions effectively.

The remainder of this paper is structured as follows: Section II presents an overview of UAtilize; Section III describes the system architecture of UAtilize; Section IV discusses each component in detail; Section V provides technical implementations, and Section VI concludes with discussion and some future work.

II. An Overview of UAatilize

UAtilize is primarily designed to visualize class enrollment data and Zip Card transaction data. UAtilize is a Web-based application which has two main components: 1) interactive visualization & analysis of campus building utilization; 2) interactive visualization & analysis of Zip Card utilization. Users can access UAtilize on their computers, tablets, and smart phones. UAtilize supports both traditional web browsers and mobile web browsers. For mobile web browsers, it automatically resizes its interface for the browser window using responsive actions of web application.

For the campus building utilization component, UAtilize visualizes the student intensity on each location in UA. Registrar Office at UA provided the class enrollment data for the semester of Spring 2014 to assist this project. A snapshot of this component is shown in Figure 1. Users can easily obtain the information about the number of students in each building at a certain time period. UAtilize provides options for users to choose the time period by selecting hour of day, day of week, and time of day. UAtilize visualizes the student intensity by creating a marker on each location. Users can click on each location to see the total number. In addition, UAtilize also provides building images for easy recognition.

Figure 1. A snapshot of Campus Building Utilization

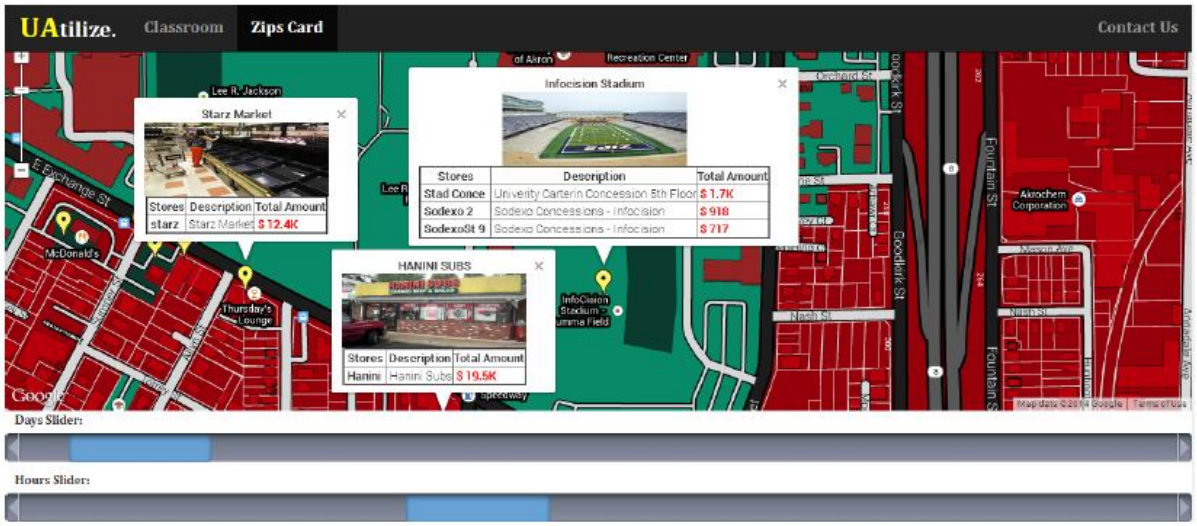

Figure 2 A snapshot of Zip Card Utilization by locations

The Zip Card is the official UA identification card. It provides easy access to UA resources and packs a multitude of campus community applications in one convenient card. The Zip Card has multiple uses: Library services; Entrance to campus buildings; fitness facilities and labs; Admittance to University athletic events; Dining plan spending; Dining and All Campus account spending; Zip Print at campus computer labs. The Auxiliary Services at UA provided de-identified Zip Card transaction data in the year of 2012 to support this project. Each card holder ID has been encrypted when the data is provided for UAtilize.

For Zip Card utilization component, UAtilize first integrates transaction data and building location data using Google Fusion Tables. Then, it provides interactive visualizations of the total transaction amount for all applicable locations, such as Library, student union, and recreation center, using Google Maps and Google Chart APIs. A snapshot of this component is shown in Figure 2. The outcome of the Zip Card Analytics system can help end users know about the usage patterns of Zip Card at vendors/events accepting zip cards. The vendors and the event managers can plan their business or events accordingly.

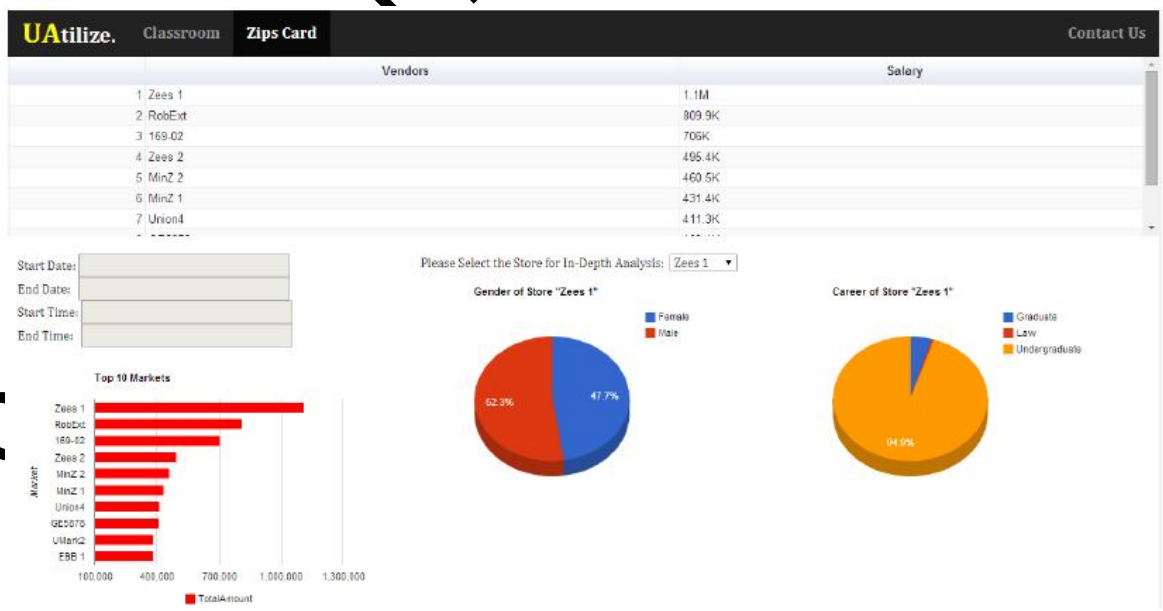

Figure 3. A snapshot of Zip Card Transaction Data Analysis

III. System Architecture of Uatilize

UAtilize is a web-based application, which supports both traditional web browsers and mobile web browsers. For mobile web browsers, it automatically resizes its interface for the browser window using responsive actions of web application. Users can access UAtilize on their computers, tablets, and smart phones. The system architecture of UAtilize is shown in Figure 4. Several frameworks (including Google Maps, Google Fusion Tables, Google APIs, and Bootstrap) and programming languages including HTML5, JavaScript, CSS, and JQuery are used to implement UAtilize.

Figure 4 shows the system architecture of UAtilize. Class enrollment data and Zip Card transaction data are originally stored in relational databases which are controlled by Registrar Office and Auxiliary Services at UA. We received class enrollment data and Zip Card transaction data in CSV files format and uploaded them to Google Fusion Tables. UAtilize is built upon four data tables which are listed in Table 1. Specifications for each data table are provided in Table 2.

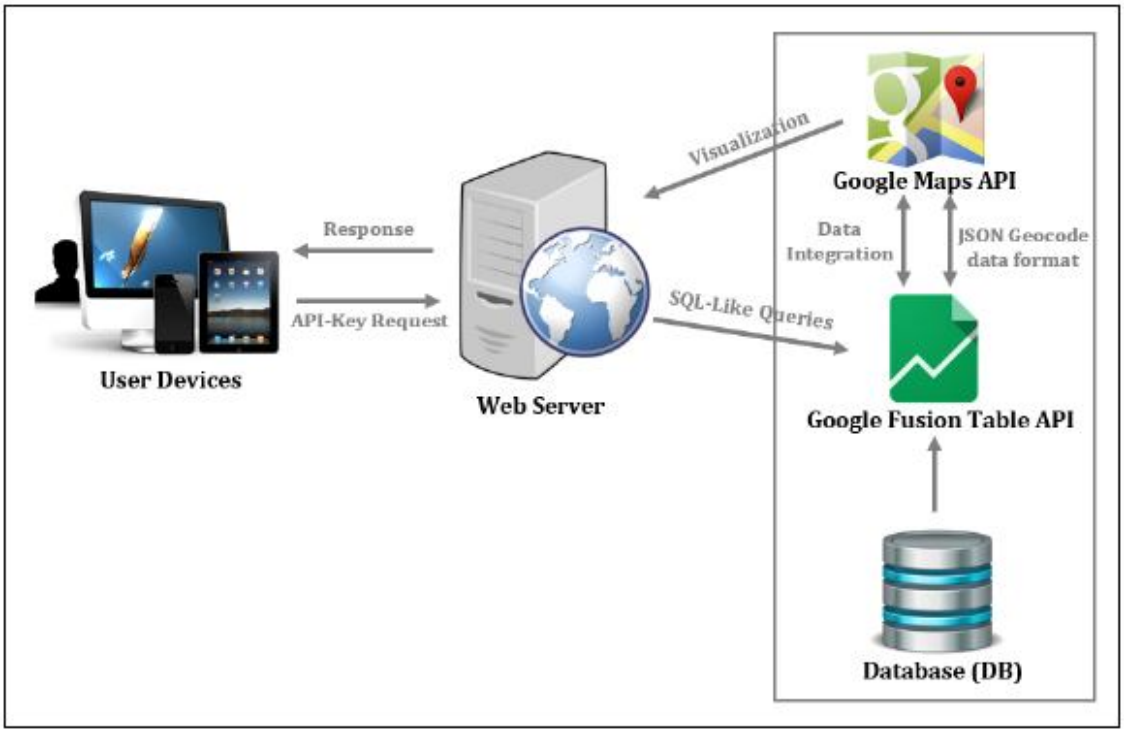

Figure 4.The System Architecture of UAtilize

Table 1 The four tables used in UAtilize

Table Name	Records (# of rows)	File Size
Class Enrollment	1,873	291 KB
Zip Card Transaction	2,405,647	124 MB
Student Information	130,768	17.4 MB
Building Location	397	30.2 KB

Table 2 Specifications of four data tables listed in Table 1

Table Name & Description	Attributes & Description
Class Enrollment – consists of all classes in Spring 2014 semester	• Building: codes, names, latitude, and longitude • Hours: provides start and end time for each class in HH: MM: 00 AM/PM format • Days: including all academic days running from Monday to Sunday • Student amount: shows the total amount of students of each class in numerical format
Zip Card Transaction – consists of all transactions in 2012	• Date and time: time which transaction has occurred • Location and Name: Identify where transaction has occurred • ID Number: this column is encrypted when we receive the data • Amount: the transaction amount
Student Information – consists of students who used Zip Card in 2012	Gender, career, major, and ID (note: the ID column is encrypted when we receive the data)
Location and Building – collections of campus buildings and markets locations	Latitude, longitude, building code, market code, and building name

According to the system architecture shown in Figure 4, the following four steps illustrate the data flows for UAtilize:

Step 1. To use the Google Fusion Table service, users send HTTP requests followed by Google APIs encrypted-key and SQL-like queries to the Google web server.
Step 2. Google Web Server identifies which services user would like to use. And then send the further additional request such as SQL-like queries to the Google Fusion table services.
Step 3. Google Fusion Table interprets the information we requested, and send all records to the user in JSON format.
Step 4. All of the records in JSON data will be translated into geographical data by Google Maps API. Moreover, Google Maps repeatedly process the JSON data with Google APIs service, and create a stunning and interactive visualization map which will be sent back to the user devices (including laptops, tablets, and smart phones).

IV. Classroom Utilization & Zip Card Utilization

We built our application on the top of the Bootstrap [8] which contains HTML and CSS-based design templates for creating web applications. All functions for serving as user interaction purposes are implemented in JavaScript, where visualizations are interactive and dynamic based upon user's requests. Moreover, UAtilize is compatible with the latest version of multiple major browsers such as Internet Explorer, Chrome, Firefox, and Safari. For the server-side, we used several Google services and Google mechanisms powered by Google Inc. In the following sections, we describe the implementation details for campus building utilization and Zip Card utilization.

A. Campus Building Utilization

UAtilize provides classroom utilization, which visualize the student intensity on each location in the university. Users can get benefit by knowing the amount of student in each building. For instance about the emergency scenario, we can manage to evacuate students in the nearest location to avoid the emergency issues such as fire accidents, snow storm, and so on. UAtilize provides users to adapting the time with hours of days by using the class enrollment information. UAtilize visualize the student intensity by create a marker on each location. Users can click on each location to see the amount. Moreover, UAtilize also

provides building image for easily recognition. Figure 5 shows an Example of student intensity in the College of Arts and Sciences building.

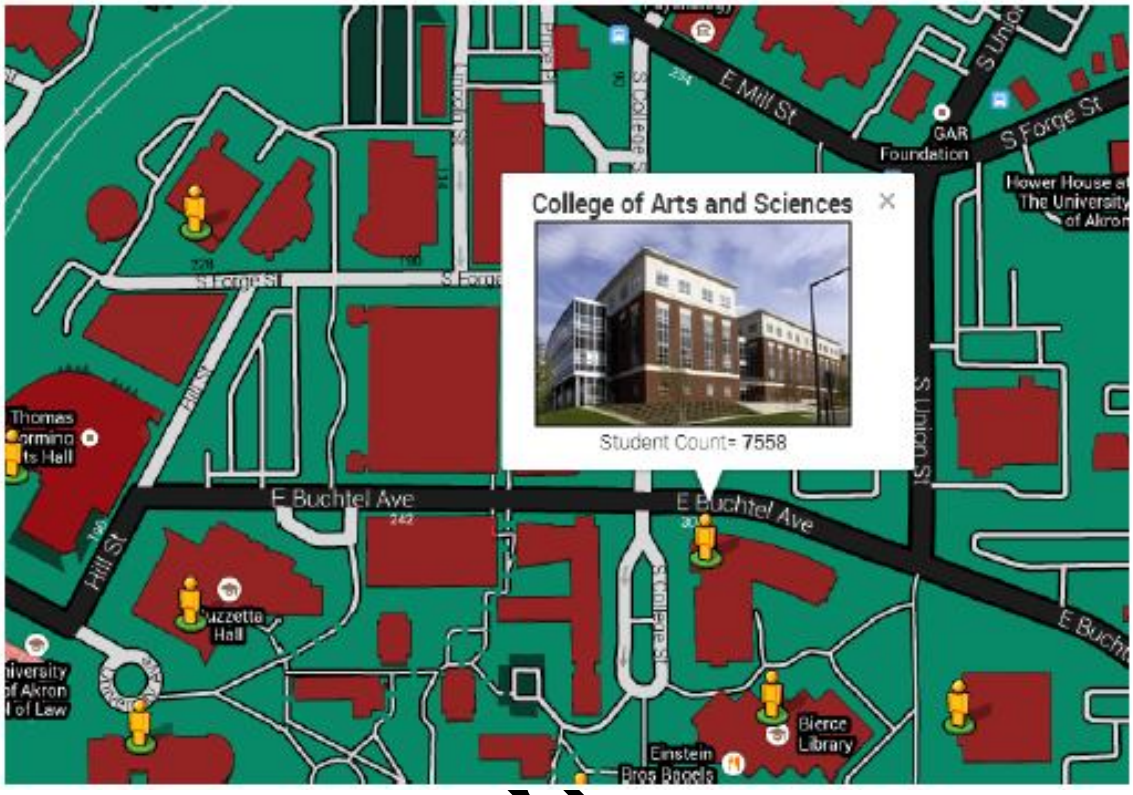

Figure 5 An Example of student intensity in the College of Arts and Sciences building

In addition, in order to obtain specific results, UAtilize provides several components (e.g. building name, weekdays) for filtering the result. UAtilize uses a JQRangeSlider [9] component to provide a time slider bar, which allows users to easily choose a certain time period. Moreover, in classroom utilization, we also provide charts for further data analysis by the components shown in Figure 6.

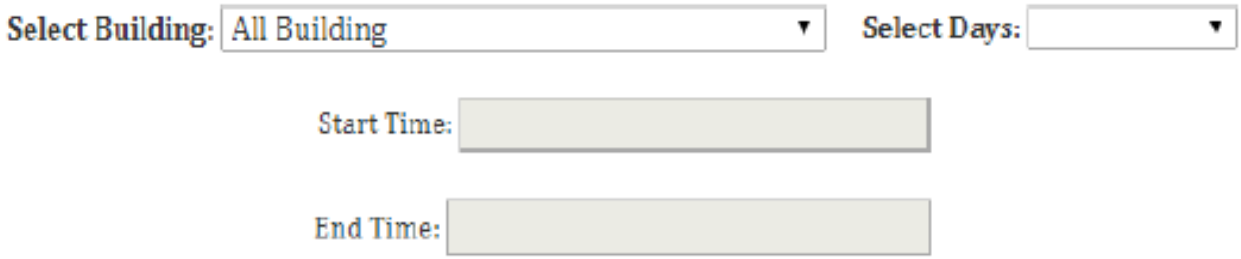

Figure 6. Components (including building and day time) for analyzing building utitlization

a. Analysis for Classroom Utilization

UAtilize also provides analysis results in tabular and pie chart format. Users can easily access the data, which represented in numerical percentages for the further analysis. Figure 7 shows an example of the analysis result in percentage following in the pie chart and tabular format.

	Building_Name	Student_Count
1	AYER HALL	1926
2	Auburn Science and Engineering Ctr.	5579
3	Bierce Library	549
4	Buckingham Building	960
5	Business Administration Building	11065
6	Center For Child Development	85
7	College of Arts and Sciences	7558
8	Crouse Hall	7428
9	E.J. Thomas Performing Arts Hall	78
10	Folk Hall	2184
11	Goodyear Polymer Center	1368
12	Guzzetta Hall	8055
13	James A. Rhodes Health and PE	1533
14	Knight Chemical Laboratory	2722
15	Kolbe Hall	9071
16	Leigh Hall	10099

Figure 7. Showing student intensity in percentage following in the pie chart and tabular format

B. Zip Card Utilization

The second component of UAtilize is the interactive visualization and analysis of Zip Card utilization. This component is built upon Google Maps and Google Chart APIs. Figure 8 shows an example of Zip Card Utilization. According to Figure 8, UAtilize provides a clear representation for Zip Card Utilization analysis. The output is created on each marker in tabular format Users can know where the transaction has occurred on the specific building, and how much money they have been made in the periods of time.

Similar to class utilization page, Zip Card Utilization also provides the time slider bar for filtering the results. Moreover, this page has go further than the class utilization page by integrating transactions results and analyzing with the information on the student table, which help users to identify total amount of student with genders, and career filters. Figure 9 shows an example of top 10 vendors and student distribution percentage. In summary, UAtilize provides an easy and effective way for users to visualize and analyze all of the data for further analysis.

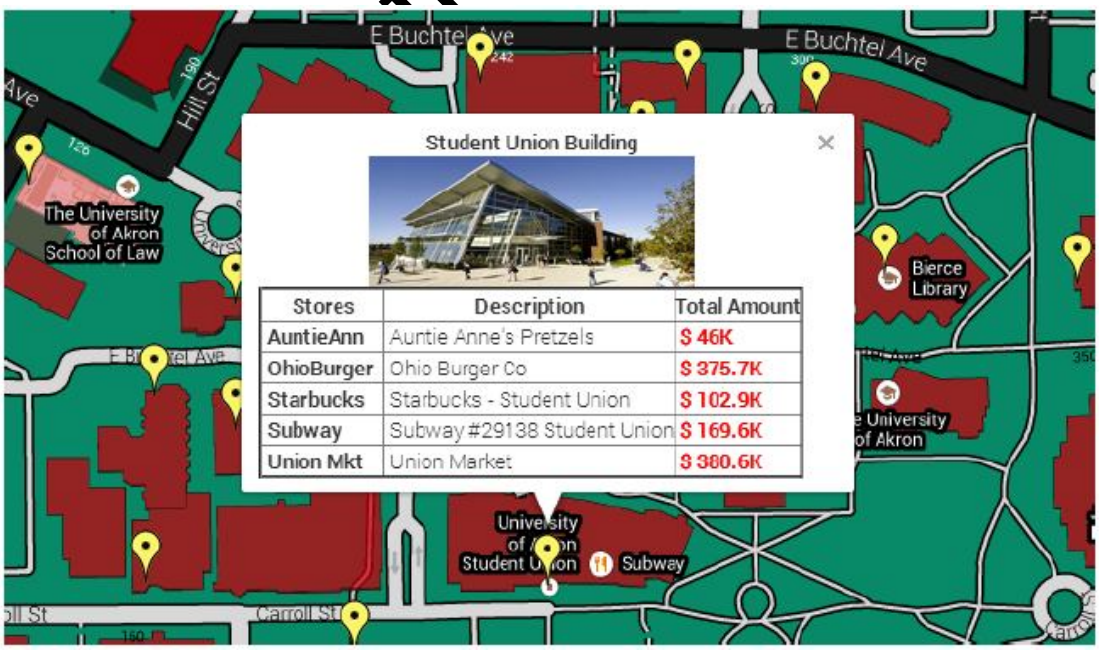

Figure 8. An Example of Zip Card Utilization analysis result

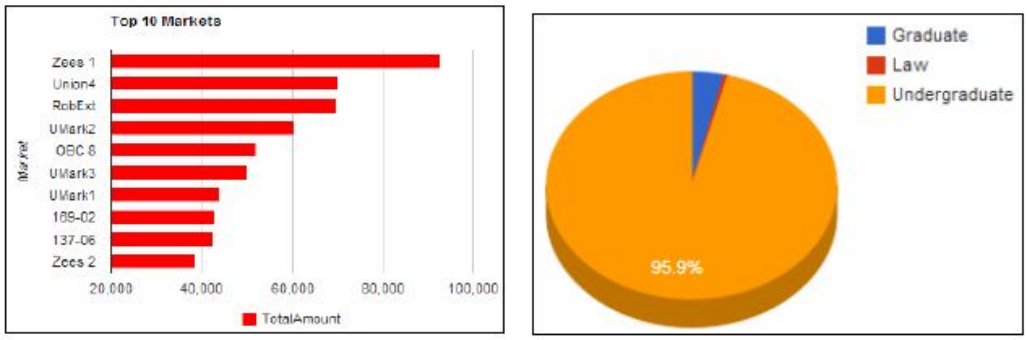

Figure 9. Top 10 vendors and student distribution percentage

C. Challenges

There were many challenges in the early development of UAtilize. The first one is to visualize the data on UA campus map. Secondly, we faced one challenge when we integrate the slider bar between the Google Maps API and Google Fusion Tables. To overcome this, we chose to use JQuery component on the web called JQRangeSlider [9]. Last but not least, we learned new knowledge about Google Fusion Tables. The new discovery includes: can only send 25,000 query requests/day; store 250MB data for each table; a cell of data in Fusion Tables supports a maximum of 1 million characters. In addition, regarding insertion records to Fusion Tables, we observed:

- Maximum data size for a single HTTP request is 1 MB
- Total number of table cells being added cannot exceed 10,000 cells (not rows!)
- A single request may contain up to a maximum of 500 INSERT statements
- Only the first 100,000 records are mapped or included in spatial query results
- When viewed on a map, map tiles may exceed the 500 feature-per-tile limit, and features will be dropped from the map.

V. Implementation

According to our case study along with several Google services such as Google Maps, Google Fusion Tables, we have selected Google Fusion Tables, Google Maps, and Google Visualization as the frameworks for UAtilize to store and visualize the data. Specifically, we use Google Fusion Tables to store four data tables which are listed in Table 1 and use Google Maps to indexes streets, displays satellite and street-level images. In addition, we also use Google Visualization APIs to provide graphs and charts to end users for further data analysis.

A. Data Collection

Google Fusion Tables allow developers to import local data from spread sheets in .CSV file format into Google cloud-base services. As a result, cloud-data can be easily integrated and linked to our website. Google Fusion Tables provides API services, which allow developers to query, gather, and process the data stored in Fusion Tables. In addition, Google Fusion Table APIs allow developers to send the HTTP requests in the form of SQL-like queries to programmatically perform all tasks such as insert, update, and delete from the table.

Step 1 – We import class enrollments, transactions, students, and location data in .CSV format from spreadsheets into Google Fusion Tables. In order to query the cloud-data, Google APIs support sending HTTP requests to Google web server:

- Using SQL-like query followed by Google account encrypted-key in form of URL link: https://www.googleapis.com/fusiontables/v1/query?sql = *{SQL query}* where {SQL query} is a valid SQL-like query provided by Google Fusion Table APIs 1.0.

Step 2 – Calling APIs from a browser by using JavaScript and query parameter. This technique allows us to write a rich application that displays Google Fusion Tables data without writing any server-side code.
Step 3 – Fusion Tables APIs allow us to specify data type formats. Thus, we can define the format for returning response data in .CSV or JSON file format. In this case, we have selected the default data format in JSON typed data.

JSON (JavaScript Object Notation) is a common, language-independent data format, which provides a simple test representation of arbitrary data structures. Figure 10 shows an example of response data we received by sending the HTTP request to query 10 records from *Zip Card Transactions* Table:

```
{       "kind": "fusiontables#sqlresponse",
        "columns": ["Store","TotalAmount"],
        "rows": [["131-01",645.25],
                 ["131-02",217.5],
                 ["131-03",47.97],
                 ["134-01","3285"],
                 ["134-03",2094.75],
                 ["134-05",45986.99999999794],
                 ["134-12",51459.68],
                 ["134-14","1326"],
                 ["135-03","414"],
                 ["135-05",418.5]]
}
```

Figure10. JSON Response data, *"131-01"* represented as a name of vendors in the Zip Card Transaction table and *"645.25"* are the dollars amount they made at a time.

B. Visualization

In order to visualize the data, Google Maps JavaScript API v3 [6] allows developers to create, style, color, and display a rich application and stunning visualization, which including Geocoding, Directions, and Street view.

Step 1 – In order to use Google Maps services, we load the Google Maps API by adding URL link to Google APIs followed by an API encrypted-key:

```
<script src="https://maps.googleapis.com/maps/api/js?key=API_KEY"
type="text/javascript"></script>
```

where *API_KEY* parameter contains a Google APIs key getting from our Google service account

Step 2 – Loading APIs on our application, we use a window.onload response command which was written in a map options into another <script> tag. Also create an "Initiate" function in JavaScript which creates the map and specific center: to allocate where latitude and longitude position we want to focus on. And putting a DOM elements, which displaying the map on our web page followed by tag. For instance:

```
<div id="map-canvas" style="width: 100%; height: 100%"></div>
```

Step 3 – we can call Map Objects by using provided services from Google APIs, which presented in a Map class to create a large-single map on our system by following this script format:

```
var map = new google.maps.Map(document.getElementById("map-canvas"),mapOptions);
```

Styling Map Approach – Styled maps allow us to customize the presentation of the standard Google base maps, changing the visual display of such elements as roads, parks, and built-up areas. In this case, instead of implement our styled map by our own hands. We use the Styled Map Wizard [10] to set the styled maps. Styled Map Wizard give us to export the JSON file format and simply put on our source code.

VI. Conclusion

In this paper, we presented UAtilize which is an Interactive Visualization & Analysis of Campus Building Utilization application using Google Maps and Google Fusion Tables. UAtilize is a web-based application, which supports both traditional web browsers and mobile web browsers. For mobile web browsers, it automatically resizes its interface for the browser window using responsive actions of web application. UAtilize is built upon several frameworks (including Google Maps, Google Fusion Tables, Google APIs, and Bootstrap) and programming languages including HTML5, JavaScript, CSS, and JQuery. UAtilize provides multiple functions and dynamic visualizations of campus building utilization data and Zip Card transaction data, leveraging the comprehensiveness, accuracy, and usability of Google Maps. UAtilize is capable to integrate and transform geographical data into a map. The primary goal of UAtilize is to assist several departments at UA, including Registrar Office, Parking and Transportation Services, Police Department, and Auxiliary Business Operations. With UAtilize, users can directly and interactively analyze the data using Google Charts and Google Visualization.

VII. Reference

1. TeraData, http://www.teradata.com/?LanType=1033
2. TDWI, http://tdwi.org/Home.aspx
3. DAMA International, http://www.dama.org/i4a/pages/index.cfm?pageid=1
4. Google Fusion Tables, https://support.google.com/fusiontables/answer/2571232
5. Google Maps. https://developers.google.com/maps/
6. Google Maps JavaScript API v3, https://developers.google.com/maps/documentation/javascript/
7. The University of Akron, http://www.uakron.edu/
8. Bootstrap Framework, http://getbootstrap.com/
9. JQRangeSlider, http://ghusse.github.io/jQRangeSlider/
10. Google Styled Map Wizard. http://gmaps-samples-v3.googlecode.com/svn/trunk/styledmaps/wizard/index.html?utm_medium=twitter
11. Google APIs, http://en.wikipedia.org/wiki/Google_APIs