ParkView


Manzanita Lake, Lassen National Park, courtesy of nps.gov

Final Project Report

Aisha Kigongo

Julia Kosheleva-Coats

Colin MacArthur

Sasaki Masanori

May 8, 2014

UC Berkeley, School of Information

Table of Contents

Executive Summary

Our client, the National Park Service (NPS), is constantly striving to improve park programs and services at all 401 national parks across the U.S. through a culture of reflection and evaluation. Currently, many traditional evaluation approaches require enormous data collection efforts hampered by continuous budget cuts and onerous survey research requirements. Although much of the data needed for park programs and services are available to park staff, the datasets are dispersed across multiple systems, platforms, and formats, making data retrieval resource-intensive and inefficient.

To help the National Park Service develop a solution, we partnered with 9 national parks and conducted a rigorous user research and needs assessment that led us to the following key discoveries:

- Although national parks do collect data about park program attendance, most use the data in limited ways such as scheduling and selecting topics for guided walks.

- Most parks do not utilize social media or reputation sites to evaluate their programs since obtaining, managing, and analyzing information from a variety of websites is costly.

- There is significant interest among park staff in using social media platform data for park program evaluation.

To address these needs we undertook an iterative development approach and focused on creating a modular, extensible, and re-deployable system to ensure cost efficiency of implementation and maintenance. As a result, we built the ParkView system (http://www.usparkview.com) – an online analytical processing (OLAP) tool that consists of analytical pre-processing algorithms, a data warehouse, and a web-based dashboard front end. This is a cost-effective approach to consolidating various data sources and to providing park staff easy access to the information.

Based on our conversations with partner parks and user feedback, the ParkView project is being considered for implementation in at least one national park. This fact alone makes our project a significant accomplishment.

Acknowledgements

We thank Dr. Ray Larson and Dr. Tapan Parikh for their guidance and many helpful suggestions. We also thank our partners at Lassen National Park, Saguano National Park, Bandelier National Monument, Death Valley National Park, Cuyahoga Valley National Park, Petrified Forest National Park, Glacier National Park, Badlands National Park, and Acadia National Park for sharing and providing insight into the problem and their feedback during our prototyping and development process.

Background

America's national parks are some of its greatest natural treasures. Largely funded by taxpayers' money, the national parks are established not only to preserve the country's wildlife and natural beauty, but also to provide access and enrichment to all citizens and visitors. Nonetheless, it is rare that visitors experience our national parks unmediated. The brochures they read, the websites they visit, and the park staff they interact with are the conduits to the national park experience.

The National Park Service (NPS), a bureau of the U.S. Department of the Interior, provides these conduits, or 'interpretive services', as an integral part of its charter. The NPS manages 401 parks across the country and maintains 909 visitor centers (The National Park Service, n.d.). Each park employs a Chief of Interpretation who deliberately designs park-specific interpretive programs, services, and events to build connections between the public and the park.

To improve these services, the National Park Service is building a culture of reflection and evaluation. In other words, park Chiefs of Interpretation are encouraged by the NPS national leadership to evaluate the effectiveness of their interpretive services and use the results of these evaluations to improve the quality of programs and services (The National Park Service, n.d.). However, many traditional evaluation approaches require enormous data gathering efforts hampered by continuous budget cuts and onerous survey research requirements.

Although many parks believe that they are 'data poor', numerous data sources that could provide wealth of information and insight into the quality of parks' interpretive services are readily available. They include:

- The results of annual short visitor surveys mandated by the Government Results and Progress Act (GRPA);
- The results of annual extended surveys conducted by the University of Idaho Department of Park Studies;
- Traffic sensor data collected to estimate the number of visitors at various locations within many parks;
- The results of the qualitative expert evaluations from interpretive coaches based on their reviews of guided walks and evening park ranger programs;
- Park-related qualitative data from social media sites, such as Twitter, Facebook, and Flckr;

- Information from reputation and experience sharing sites, such as TripAdvisor and Yelp.

We posit that park staff does not utilize these data assets because of the effort associated with collecting, consolidating, and analyzing this data. Although available, these datasets are housed on multiple unrelated systems. Accessing individual systems and consolidating these data sources into a cohesive set requires much time, effort, and technical expertise. Even once the data is consolidated, most of the analysis must be performed manually.

Problem Statement

Current approaches to evaluating park interpretive programs and services are time-consuming and costly. Although much data is available to park staff, the existing datasets are dispersed across multiples systems, platforms, and formats, making data retrieval resource-intensive and inefficient. As the result, much of NPS' data assets are rarely accessed and used for evaluation and decision-making purposes.

As a team of nature devotees and national park enthusiasts, we took on the project of developing an online analytical processing (OLAP) tool that consists of analytical pre-processing algorithms, a data warehouse, and a web-based dashboard front end as a cost-effective approach to consolidating various data sources and to providing park staff easy access to the information. We believe that our solution will not only improve current service and program evaluation efforts, but will be utilized for tactical and strategic decision-making, problem identification and monitoring, and resource allocation optimization.

Project Scope and Constraints

To help the National Park Service solve the problem we partnered with 9 national parks. More specifically, we worked directly with the Chiefs of Interpretation from Lassen National Park, Saguano National Park, Bandelier National Monument, Death Valley National Park, Cuyahoga Valley National Park, Petrified Forest National Park, Glacier National Park, Badlands National Park, and Acadia National Park. This collaboration helped us gain insight and clearly define the problem, determine project objectives and scope, as well as formalize the expectations around project deliverables.

Overall, the scope of our project included designing and building a proof of concept OLAP system. Specifically:

- Import and pre-process structured, semi-structured, and unstructured data from multiple social media platforms and other publically available data sources;
- Design and implement a data warehouse;
- Design and implement a web-based dashboard application that analyzes and summarizes information available in the data warehouse.

Considering time limitations of this semester-long project, the following aspects are considered out of scope:

- Exhaustively research all publically-available park-related data sources;
- Design and implement interactive dashboard features beyond hover-overs;
- Support and provide consulting in the case of system roll-out to multiple national parks;
- Prepare system training materials and additional documentation beyond final project requirements;
- Train the front end users and information technology (IT) staff.

At the start of the project, we also identified the following constraints:

Cost. Since the National Park Service and their partners do not have significant funds for system implementation and maintenance, we had to be cost-conscious as we proceeded with system design decisions, choice of software, and programming languages. For example, we ensured that all tools used for building the data warehouse and the dashboards are largely open-source like MySQL and Python.

Data availability. We expected that we will not load and use all data sources currently available to park staff due to privacy considerations and the time it would take to locate additional data. For instance, staff performance evaluations contain information that cannot be shared. Moreover, not all data from the social media platforms can be easily collected since APIs for many of these sites are limited and would require additional resources to obtain.

Technical skills limitations. Although experts in our respective fields, we do not have in-depth knowledge or significant experience developing and implementing front-end applications or devising natural language processing (NLP) algorithms.

User Research and Needs Assessment

Research Methodology

To better understand the problem and specific needs of our users, we started the design process by interviewing park superintendents and Chiefs of Interpretation – our potential user base. Our research was designed to answer the main two questions:

- How do Chiefs of Interpretation and park superintendents currently use data to evaluate their park programs and services?
- What are the challenges of the current approach?

We contacted 20 mid-size national parks (excluding national monuments) and were able to conduct in-depth semi-structured phone interviews with 13 park staff, including 9 Chiefs of Interpretation. Our interview protocol included the following questions:

- What quantitative and qualitative data do you currently collect to evaluate park programs?
- Where is this information currently stored?
- Who does access this information?
- How is it being accessed and how often?
- What are the major challenges you face when evaluating park programs?

User Research Results

Thorough analysis of the responses and comments collected during the interviews with Chiefs of Interpretation led us to the following discoveries:

- Generally, parks do collect data about park program attendance. However, most of the national parks in our sample indicated a well-established process for using the data in limited ways. For instance, most of the parks we interviewed use this data simply to adjust guided walk topics and scheduling.

- All of Chiefs of Interpretation, as well as other park staff, are very familiar with some park visitation and program utilization statistics. For example, park staff is well aware of the seasonal fluctuations in park visitation and typical increases in traffic counts and campground occupancy around peak seasons.

- Most of the parks, even the ones with more sophisticated knowledge about social media information available on the web, do not currently utilize social media or reputation sites to evaluate their programs.

- There is a significant interest in using social media and other reputation sites data for park program evaluation across the board. However, almost all interviewees expressed concerns around how much time access, collection, consolidation, and analysis of this information takes.

These insights helped us choose a specific project focus, narrow down the scope, and guide our ideation sessions. Thus, our project focused on the most obvious user needs: (1) collection and consolidation of data from social media platforms into a single data warehouse, (2) providing analytical summaries of the information available in the data warehouse.

System, Data Warehouse, and Dashboard Design

System Design Objectives

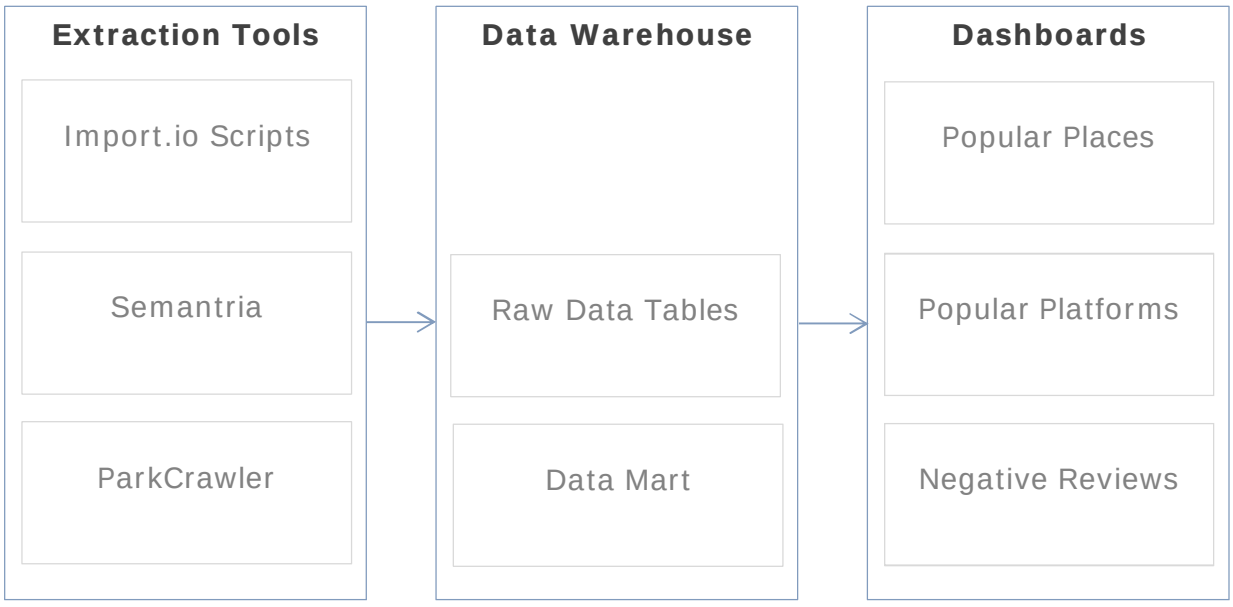
Our conversations with Chiefs of Interpretation at our partner parks hinted at the possibility of a larger scale system roll-out across multiple parks. We designed this proof of concept system accordingly. Thus, the primary system design objectives were:

- Ease of system implementation and maintenance to ensure cost efficiency;
- Modular and extensible system design to ensure that new functionalities can be added with minimum effort;
- Re-deployable design for easy installation on other servers.

Final System Architecture and Technology

The ParkView OLAP system is comprised of the three main components: a set of data extraction and analytical pre-processing tools, the data warehouse, and the front-end summary dashboards. Figure 1 displays the system architecture diagram.

Figure 1. Final system architecture



Most of the technologies we used to build the system are open-source. Each of the system components is described in more detail in the subsequent sections of the report.

Analytical pre-processing. A combination of custom-built and existing algorithms collects, transforms, and augments data from social media sites as part of the extract, transform, and load (ETL) process. We built a set of Python programs that collect data by scraping or, when possible, connecting directly to site's API. We are still arranging access to some of the sites' APIs. For those, we used import.io to scrape the data. We also use Semantria, open-source sentiment analysis software, to analyze text of all social media posts and augment each record with semantic knowledge – a sentiment score. Furthermore, we developed ParkCrawler, a Python algorithm which uses regular expressions to search for the mentions of the specific park locations in reviews and comments.

Data warehouse. The ParkView data warehouse is hosted on a Debian server, running the MySQL database management system. SQL Workbench and phpMyAdmin are used to administer the database.

Dashboards. The ParkView dashboards are implemented in CSS, HTML, D3, and JavaScript. For the basic web page layout, we used Twitter Bootstrap, extended with Flat UI. We used PHP, JavaScript, and the object-relational database mapper Propel to retrieve data from the database on the server side. Dashboard charts are powered by Google Maps API (https://developers.google.com/maps), HighCharts.js (http://www.highcharts.com), and Google Charts libraries. The files behind the front-end dashboards are hosted on the Apache web server, running Debian operating system.

Data Architecture Design Objectives

The ParkView back end is a comprehensive data repository that will support changes in park programs and services evaluation measurements and integration of social media data into program evaluation process.

To accommodate ongoing changes, the following objectives were established for ParkView data architecture design:

Data integration. The ParkView data warehouse seeks to integrate social media data from multiple platforms and across various dimensions – time, specific locations within a park, comments and reviews, ratings, etc. Thus, the data had to be integrated in such a way that park staff can easily obtain facts about each park's presence across all social media platforms. This integration is key to making park staff more fact-driven and more responsive to visitor feedback. Many subsequent objectives for our database design stem from this integration concept of a data warehouse.

Data standardization. Standardization and normalization of data collected from various social media platforms are fundamental to ensure usefulness of the data warehouse. Therefore, it was important to integrate data across time and subject areas, as well as simplify design and maintenance of the data warehouse.

Extraction, Transformation, and Load (ETL) Process

Data sources. We collected and imported unstructured and semi-structured data from the following social media platforms: Yelp, Facebook, TripAdvisor, Flickr (comments only), and Twitter. We also collected publically available visitor survey results, visitor and traffic counts from the NPS' website, and weather data from the National Oceanic and Atmospheric Administration website. Initially, we also included this data in the warehouse. However, we learned early on that park staff's priority was obtaining social media information. This feedback allowed us to limit the scope of the project and to focus on collecting and working with social media data.

Data extraction. We devised a set of Python algorithms that collect data in .csv or .json file format. When possible, Python programs connect directly to the site's API. Otherwise, we use import.io (http://import.io) and kimono (http://www.kimonolabs.com) which transform websites to structured datasets. Upon parsing the files we retained only the fields relevant to the goals and the scope of the project. Our approach was to collect, clean, and organize data as a separate process before running any analytical pre-processing on it. This approach ensured easier data warehouse development since all data manipulation is documented in the first part of the program. This would allow efficient database maintenance in the future. For example, if users decide to change data sources or modify any data cleaning steps performed on the existing data sources, all code modifications could be done in just one place.

Data integration and transformation. Integrating social media data from different platforms and time dimensions is difficult because of the variations in file format and data structure. There are two main reasons for difficulty of integrating data across various dimensions (e.g., time, comments and reviews, ratings):

- There are inherent conceptual differences between dimensions. For example, the reviews from reputation sites like Yelp and comments from social networking sites like Facebook are conceptually different. Integration across these dimensions requires a concerted effort at the data modeling stage to look below the surface and find common concepts and common data. There are substantial warehouse design implications stemming from the fact that the data comes from many different sources. For instance, the modeling process leads to discovery of data elements that appear different on the surface, but in reality are the same.

- There is the requirement that data values are sufficiently understood so they could be transformed into common terms in the data warehouse. This data transformation, aside from any other inherent advantages, allows us to summarize disparate social media activity. For example, some of the data we collected (e.g., Google Plus) utilizes a 1-10 rating scale while most of other sources utilize a 1-5 scale. This required rating normalization so ratings could be summarized and compared across multiple sources. In this situation we used the 'min-max' normalization technique (Han, 2012) to re-scale all ratings to 1-10 scale. (Note: we later removed Google Plus data from the database and rating normalization was no longer necessary; therefore, it was removed.)

By cleaning the data from the different sources, we ensured that it was well formatted and could be used correctly within the application. Part of formatting the data was to clearly define and map similar attributes to a single entity. For example, TripAdvisor data had the date field split into year, month, and day while other files contained a W3C date. We transformed the TripAdvisor date format into a new date field.

Formatting addresses or user location was challenging within our system because there was no consistency in the address format across datasets. In some data sources the address contained only the country of user residence, in others it was a city and an abbreviated state, or no city and a full state name. In the end we decided to retain addresses as a separate field, and added the city and state for those data sources for which they exist.

Analytical pre-processing. Our data transformation process not only cleans and standardizes data elements, but also conducts analytical processing and augmentation. For instance, it adds semantic knowledge to each record that contains free post text data. We utilize Semantria, an open-source sentiment analysis tool, to process each comment or review and assign a sentiment score. ParkView's Negative Reviews dashboard is designed to surface reviews that could be used to improve park services. Originally we used review ratings between 1 and 3 to identify a negative review. However, in-depth data analysis revealed that star ratings are unreliable in identification of critical feedback in social media reviews because many visitors rate parks high while describing negative experiences within their posts. Thus, Semantria allows us to go beyond misleading ratings and recognize negative feedback. Furthermore, a Python algorithm, ParkCrawler, uses regular expressions to search for the mentions of the specific park locations in reviews and comments.

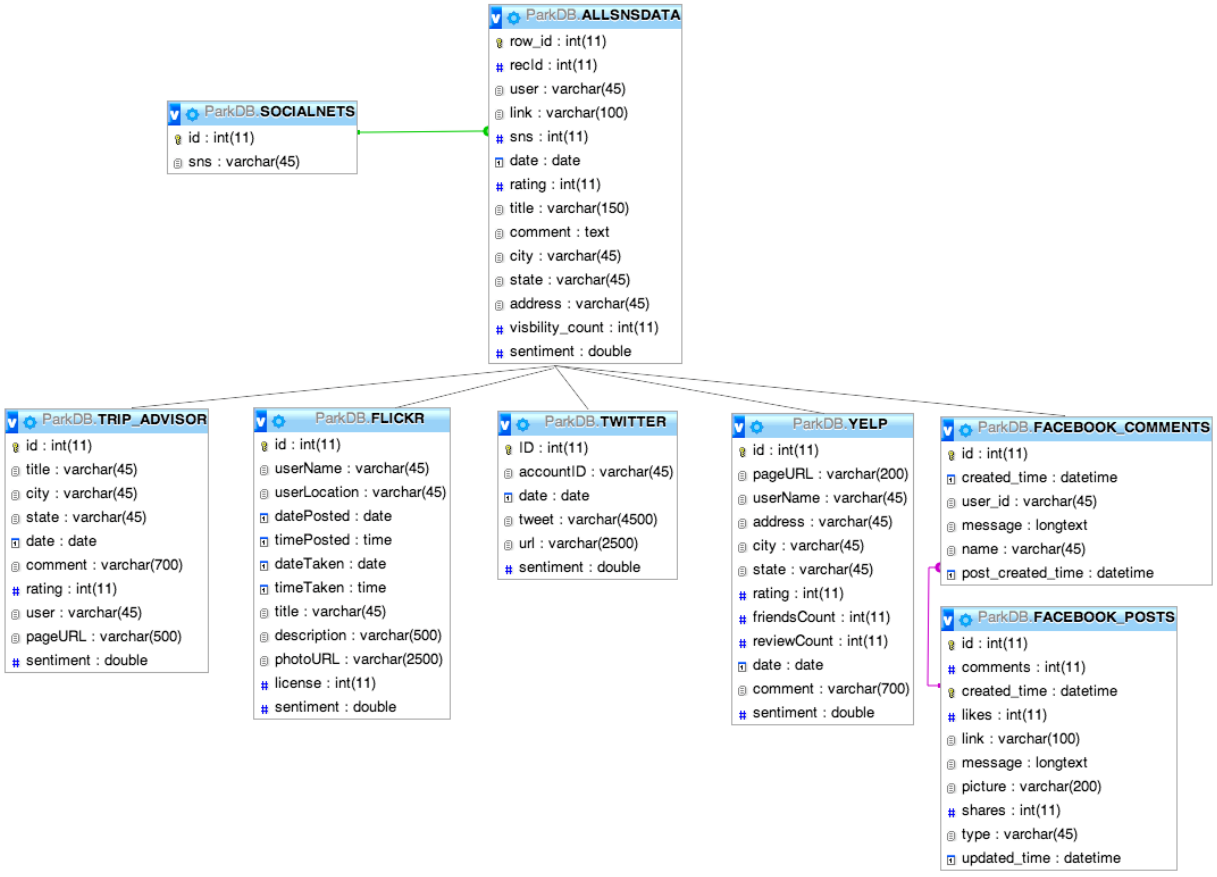
Data load. After the data preparation stage, which results in generating comma-delimited files, our process loads the data into staging tables. A separate table gets loaded for each data source. This technique makes it easier to identify dimensions of interest and to parse the records, since at this stage they are structured and organized.

Initially we created five staging tables (Figure 2) for loading with comma-delimited files (e.g., TWITTER, TRIP_ADVISOR, YELP, GOOGLE_PLUS, FLICKR, FACEBOOK). Later in the process we decided to drop GOOGLE_PLUS data because there were not enough meaningful records that could be of interest to park staff. After reviewing the data in separate tables, we isolated entities in each dataset and across datasets that would be interesting to analyze, then we merged the data into

one table (i.e., ALLSNSDATA). There is no foreign key or internal relation defined between ALLSNSDATA and the staging tables. Instead, we retain the ID and attach a platform ID, which the social media staging table record in ALLSNSDATA came from.

ALLSNSDATA table is the core table of our project. Data from this table is loaded into the analytical cube (Data Mart) and is analyzed at varying levels of granularity and at different dimensions.

Figure 2. Relation view of staging tables



Data Mart

The ParkView architecture is an OLAP cube that enables users to analyze multidimensional data interactively, from multiple perspectives. For example, users can roll-up to view average yearly ratings or drill-down to navigate through the details such as monthly ratings. We decided to design a data mart because it improved end-user response time by allowing users to access specific type of data more efficiently (e.g., ratings they needed to view most often).

To load data into the OLAP cube we made use of the Python notebook Pandas library that enables manipulation of relation database management objects in a dataFrame and exported the resulting data files for import into the database. SQL commands were also used to update and maintain the database with incoming data.

The ParkView OLAP cube uses a 'star' schema (Figure 3), which contains the following components:

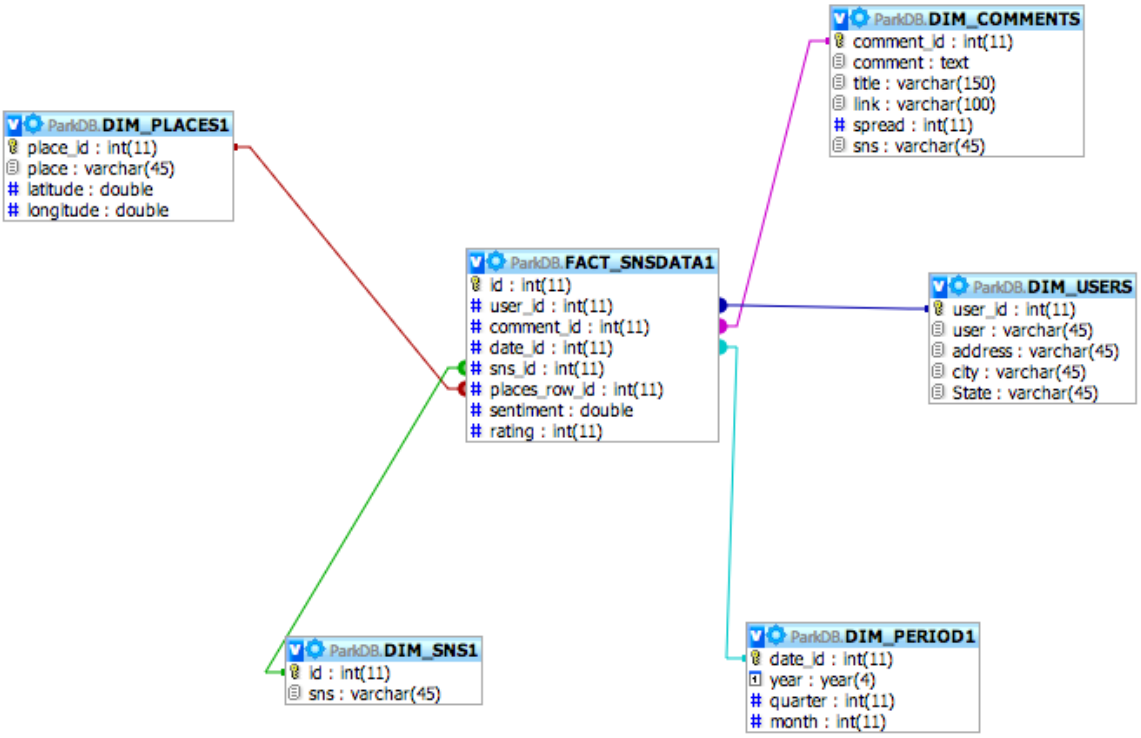
Fact tables

- FACT_SNSDATA contains ratings, sentiment scores, and foreign keys to dimensional tables where description information is stored.

Dimensional tables

- DIM_USER contains user_id (primary key), row_id (unique ID of record in ALLSNSDATA table), address, city, state, and user whose social media post and rating is read;
- DIM_PERIOD contains the data on when a social media review or comment was posted; the date is parsed into years, quarters, months and day for entries recorded in the star schema;
- DIM_SNS describes a social media platform;
- DIM_PLACES describes the places mentioned in any of the social media posts;
- DIM_COMMENTS describes each comment, review, tweet, or other post.

Figure 3. Star schema with Fact and Dimension tables



Dashboard Design Process

During dashboard design and development we followed an agile development process to quickly iterate through prototypes, gather user feedback, and implement changes. As a result, we built three dashboards that users could interact with via standard web browsers. Below is the overview of our dashboard design process that summarizes how we translated user research findings into design decisions and features reflected in the final front-end application.

User Description

The main users of our platform are national park Chiefs of Interpretation and other park staff tasked with administration of programs, interpretive staff supervision, resource allocation, and strategic planning. They are busy administrators that need to access information quickly and efficiently to help them effectively manage park programs. Although experts in park administration, typically, these park employees

do not have technical skills in database management, data retrieval, and data analysis. Nonetheless, the strategic questions and challenges of their daily jobs require them to turn to visitor metrics and insights to better serve the public. These users are already familiar with some key park visitation trends and patterns, such as summer peaks in number of visitors and winter visitation slowdown. They also have additional insight about park visitors that they retrieve from annual visitor surveys (e.g., visitor demographics, facilities and programs used, and visit preparation approaches).

Assumptions

Although we greatly relied on user research findings throughout the dashboard design and implementation process, we made a few assumptions to better frame the problem:

- Our current and future users have some familiarity with basic concepts behind social media sites such as Facebook, Twitter, and TripAdvisor. For example, the users understand how rating systems of social media sites work.

- The users understand major similarities and differences in the type of posts specific to various social media platforms (e.g., reputation sites vs. social networking sites). For example, the users understand the difference between a TripAdvisor review and a tweet.

- The users are familiar with the concept of online interactive graphics. More specifically, they are aware of the interactive functions such as 'hover-overs' for obtaining additional information, 'drill-downs' for getting underlining detailed data, etc.

Design Objectives

The process of user research revealed valuable insight into user needs, wishes, and current patterns of information retrieval and utilization process. Our findings provided guidance for the dashboard design process and helped us structure our design objectives outlined below:

- The administrators of multiple parks emphasized the importance of being able answer three key questions:

    1. What are the most popular and busy places in the park?

    2. What are the most popular platforms that visitors choose to talk about the park?

    3. What are the key problems that the visitors bring up?

- The dashboards need to provide insight into visitor experiences in a park;

- The dashboards should incorporate the ‘voice of a visitor’ and incentivize the users to go to the original source to read this and other visitor reviews;

- The dashboards should integrate information from multiple data sources and present an ‘at-a-glance’ view of visitor activity across multiple social media sites;

- Dashboard design should ensure a balanced approach between the amount of presented information and a clean, simple, easy-to-follow visual design.

Competitive Research

We started our design process by conducting competitive research. We explored a multitude of publically-available examples of social media dashboards that summarize information from a social media sites. Examples of dashboard designs by Gregoire Vella that inspired us can be found in the Appendix (section A). We also consulted information visualization literature such as 'The Functional Art' by Alberto Cairo, 'The Visual Display of Quantitative Information' by Edward Tufte, and 'Dashboard Design' by Stephen Few to explore some additional ideas and techniques. This research not only helped us find inspiration, but also identify specific elements of design we wanted to incorporate and the elements that we would prefer to avoid.

We identified the following dashboard design elements that we wanted to incorporate into our design:

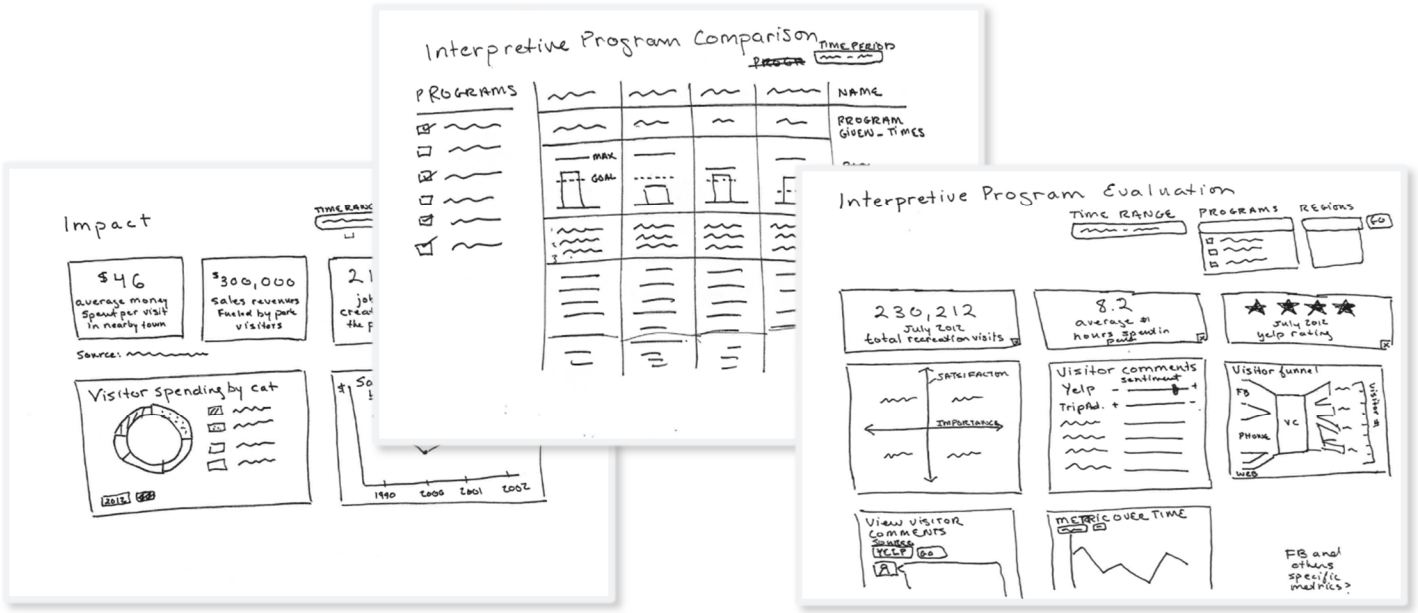
- Key fact statistics displayed in large font (e.g., number of reviews, unique users, etc.);
- Adequate context information for key facts (e.g., growth statistics, other comparisons) and historical trends for key facts;
- Verbatim visitor reviews to incorporate the 'voice of a visitor' and links to the original social media sites for seamless access to the source data;
- Maps to visually and intuitively highlight specific places in a park;
- Logical organization of graphs, facts, and qualitative information within the dashboard;
- Clean, simple layout with plenty of white space for easy navigation.

Dashboard Design Iterations

We have gone through multiple iterations before finalizing the design of the dashboards. In fact, our original plan was to utilize existing open-source Business Intelligence (BI) software for dashboard implementation. We thought that using an 'of-the-shelf' tool will ensure easy implementation, utilization, and maintenance for the Park Service. However, the inflexibility of open-source BI tools and the prohibitive cost of commercial BI software steered our dashboard development in JavaScript, D3, PHP, and HTML.

Lo-Fi Prototyping. The team prepared several low-fidelity sketches to solidify and visualize a few key design ideas that came up during our competitive research and ideation sessions. Figure 4 displays the first set of sketches. They were focused on addressing key user needs by conducting data analysis, searching for the best ways to visualize and present our data, and constructing a logical and cohesive layout of the information. Additional Lo-Fi sketches can be found in the Appendix (section B).

Figure 4. Low-Fi prototypes



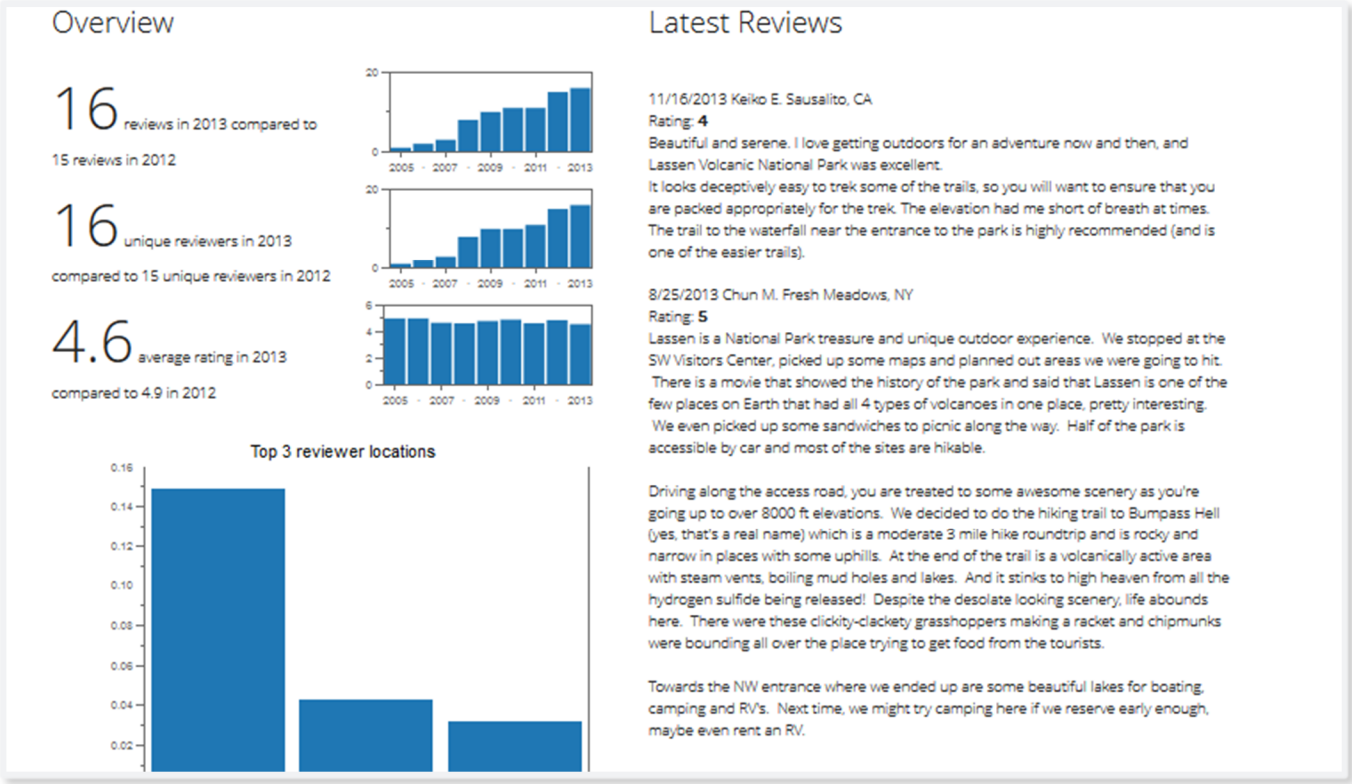
BI Tool Search. Originally, our plan was to build the dashboards using existing BI software. We conducted extensive research of various tools currently available on the market. First, based on our paper prototypes we identified key capabilities that served as main requirements for the software. Second, we consulted Gartner's published BI tools evaluations and recommendations to identify a list of acceptable tools. Finally, we conducted in-depth research and analysis of each tool and narrowed down our list to top three systems: SiSense, Pentaho, and Tableau. Figure 5 displays a sample from our informal documentation of all explored business intelligence platforms with notes on key capabilities and sample dashboards images. A sample of a detailed comparison matrix for the selected BI tools can be found in the Appendix (section D).

Figure 5. BI tool search documentation

Pentaho BI. Upon narrowing down our search to the top three BI platforms, we conducted further examination of features and capabilities, looked into pricing options, and worked with free trial versions of each software to select the most suitable tool. While we determined that SiSense and Tableau to be the best tools for the project, the cost of these tools turned out to be prohibitive not only for a student project, but also for the National Park Service. Therefore, our best option was to adopt an open-source version of Pentaho BI.

After we installed and started working with Pentaho, we quickly realized many limitations of the software. The tool provided only the very basic choices of graphs such as bar charts, pie charts, and line graphs. Geographical maps feature – a key component for our design – was disabled in the open-source version of the software. Furthermore, we found that the software's inflexibility made it very difficult and time-consuming to implement simple modifications to the dashboards. For example, we discovered that the open-source version of Pentaho does not allow pivoting vertical bar graph into horizontal bar graph. Finally, we learned that Pentaho does not provide enough flexibility for displaying unstructured data. For instance, we found it very difficult to truncate the text of a review and provide a link to a full version of that review in a pop-up window or a different screen. Figure 6 displays the resulting Pentaho BI layout.

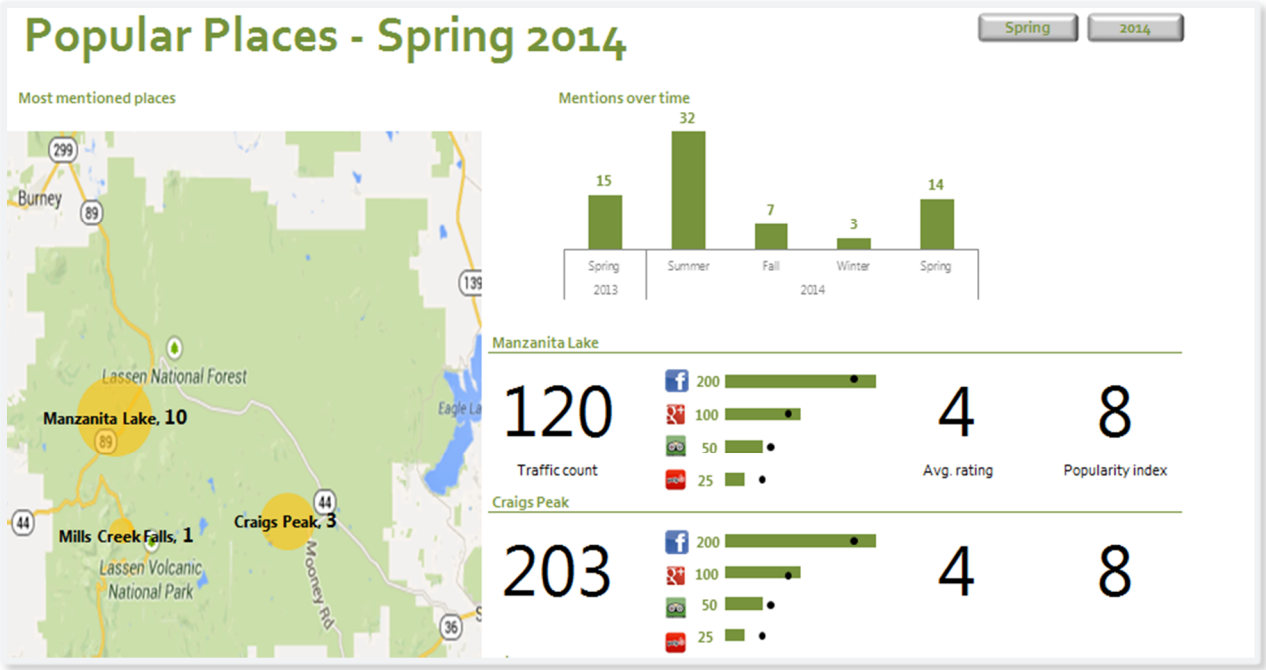
Figure 6. Pentaho BI implementation



Hi-Fi Prototyping. The disappointing experience and the unacceptable result of working with open-source Pentaho BI led us to re-consider our original plan to build the dashboards using 'off-the-shelf' business intelligence software. We agreed that developing the dashboards using tools like JavaScript, HTML, and PHP would yield better results. Thus, we quickly moved on to creating the next iteration of prototypes using MS Excel. Based on our deeper understanding of the data and the user feedback that we gathered based on the sketches and Pentaho dashboard, we took this opportunity to incorporate additional graphics, new data elements, and features. For example, we now had the flexibility to create maps and incorporate donut charts on the map to mark specific park locations mentioned in the reviews and comments. The final high fidelity prototypes are displayed and briefly described below. Additional iterations of these designs can be found in the Appendix (section C).

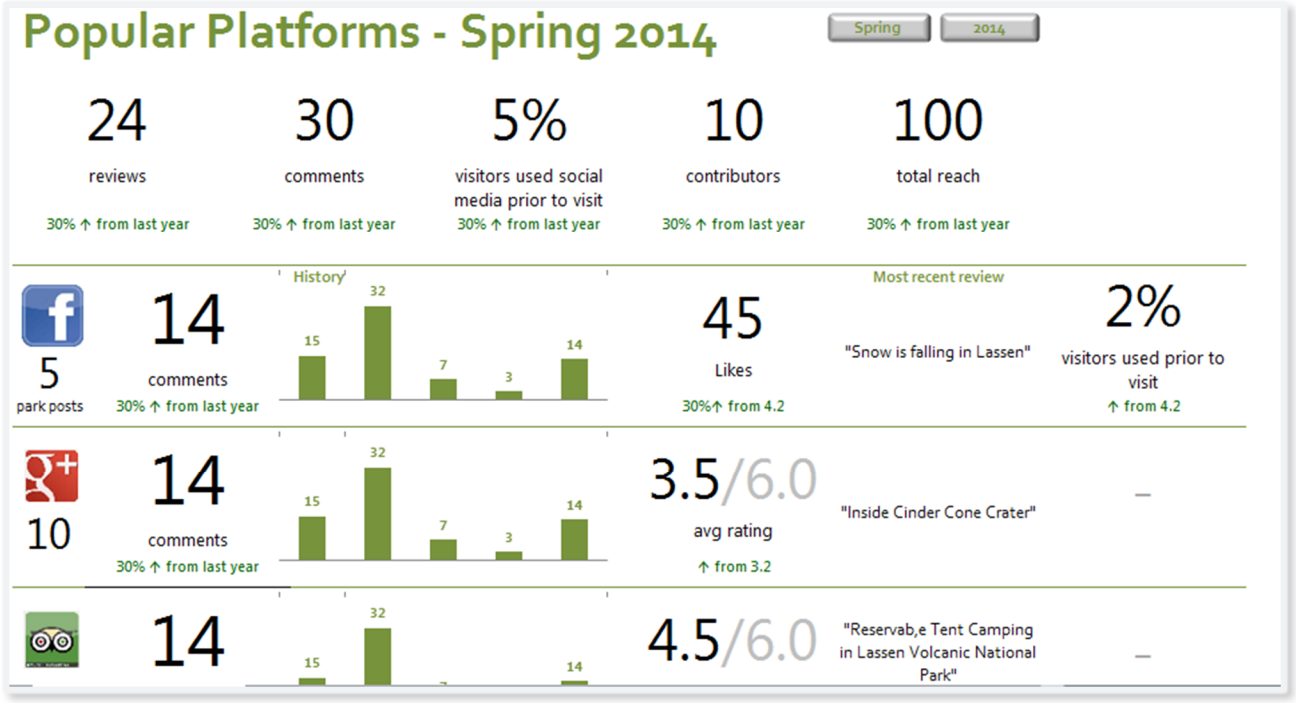
The Popular Places dashboard (Figure 7) was designed to answer the question: 'What are the most popular and busy places in the park?' In this original dashboard we decided to highlight these places on the map by counting how many times they appear in visitors' social media posts. The dashboard also displays the historical trend in the number of overall park mentions across all social media platforms and detailed place-specific statistics.

Figure 7. Popular Park Places dashboard prototype



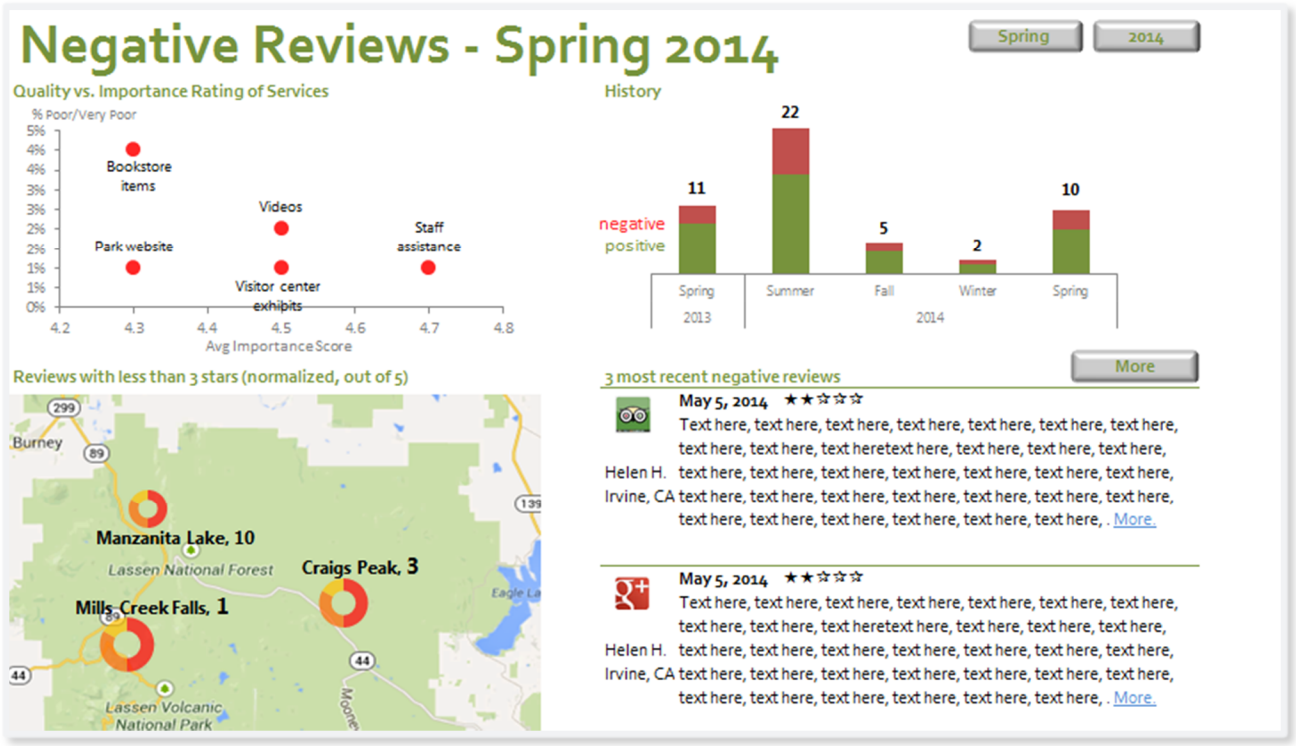
The Popular Platforms dashboard (Figure 8) was designed to answer the question: 'What are the most popular platforms that visitors choose to talk about the park?' To help answer this question we decided: (1) to summarize all social media activity related to a given park and (2) to provide detailed metrics for each individual platform.

Figure 8. Popular Platforms dashboard prototype



The Negative Reviews dashboard (Figure 9) was designed to answer the question: 'What are the key problems that the visitors bring up?' To help answer this question we wanted: (1) to summarize all negative reviews for a given park, (2) to highlight specific park places mentioned in those negative reviews, (3) to augment social media data with information from visitor surveys, and (4) display most recent negative reviews to incorporate 'voice of a visitor' and reveal concrete examples verbatim.

Figure 9. Negative Reviews dashboard prototype



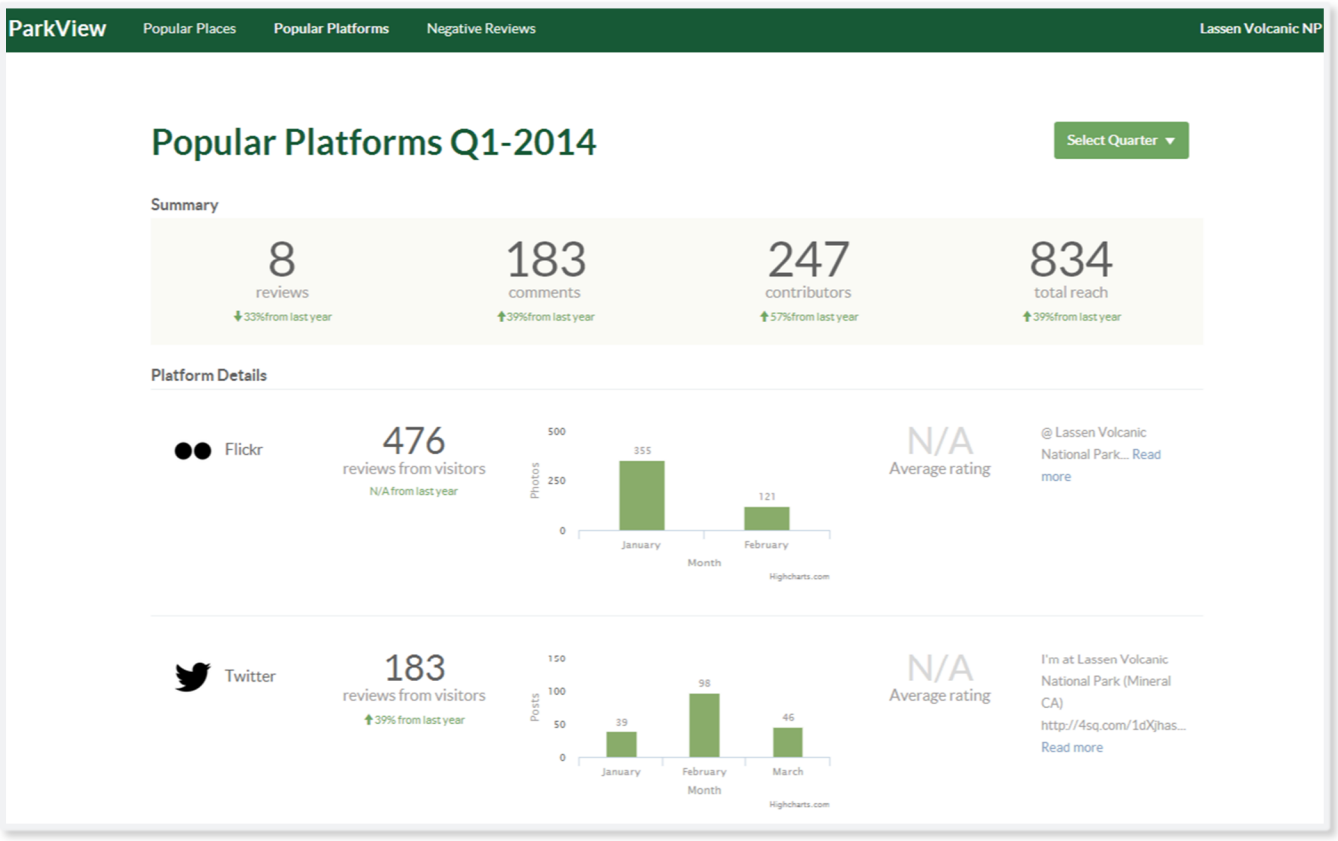
Final Dashboards Design

Popular Platforms dashboard. The final design of the ParkView Popular Platforms dashboard (Figure 10) is focused on an overview of all social media activity related to a given park and given time frame (i.e., quarter). It provides analysis to answer the question: 'What are the most popular platforms that visitors use to talk about the park?'

The Summary section at the top displays key facts. The first two measures are: (1) the number of reviews from the reputation sites like Yelp and TripAdvisor, and (2) the total number of comments from social media and networking sites like Twitter and Facebook. It, then, shows the total number of individual unique content contributors across all platforms and the total reach measured by count of unique users that contributed comments, likes, etc. A hover-over feature brings up a pop-up window with the bar graph that shows how each metric is distributed across social media sites.

Each row of the Platform Details section is dedicated to a specific social media platform. It displays the total number of park related posts, trend over the past few months, the average star rating, and a brief extract from the most recent post to bring the 'voice of a visitor' into the dashboard.

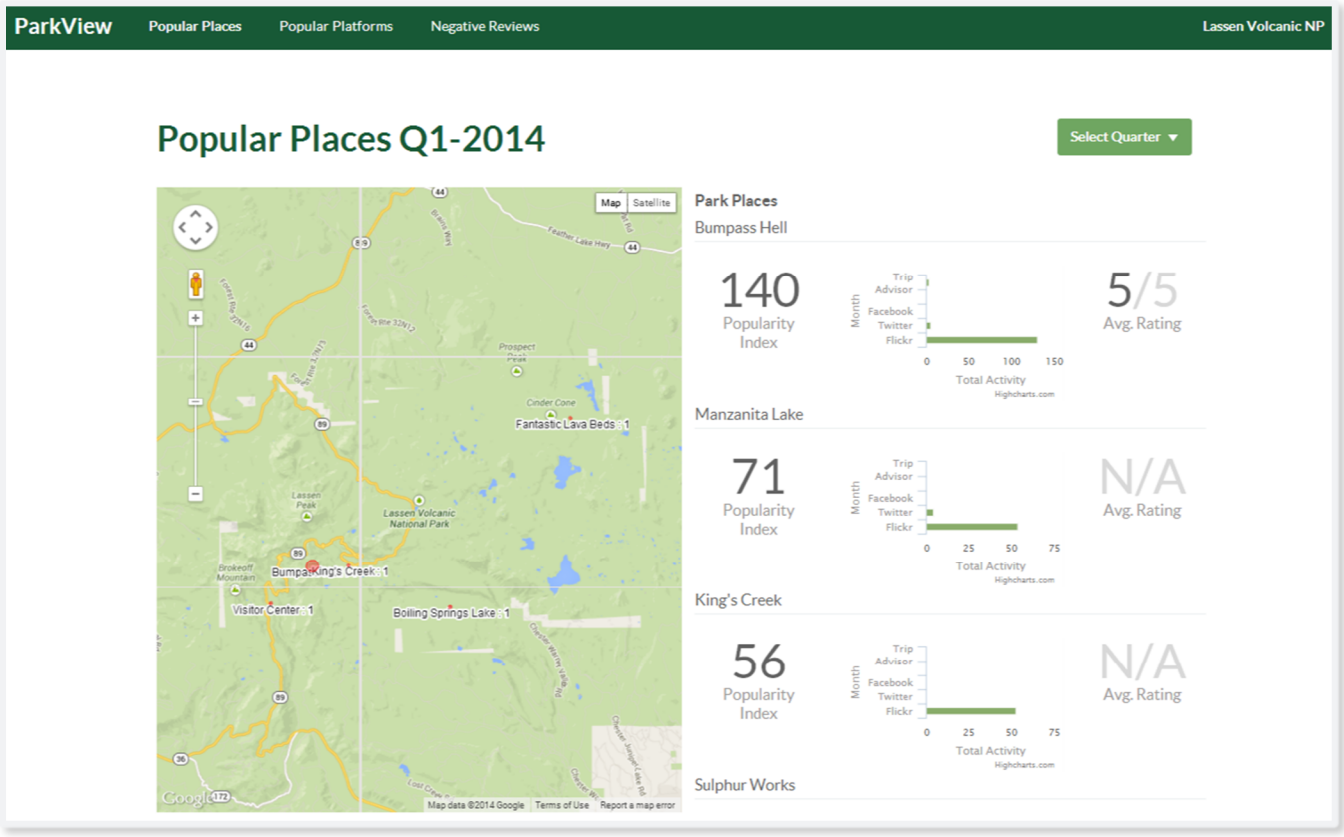
Figure 10. Final Popular Platforms dashboard



Popular Places dashboard. The final design of the ParkView Popular Places dashboard (Figure 11) is focused on an answering the following question: 'What are the most popular and busy places in the park?' It highlights specific locations within the park by counting the total number of social media posts that mention those places.

First of all, we mark each location with at least one mention on the park map with a circle, the diameter of which corresponds to the total number of social media posts that mention this location – the larger the circle, the greater the number of posts. The Park Places section of the dashboard allows side-by-side comparisons of additional statistics across all key park attractions. The popularity index conveys the total number of mentions across all posts. The bar graph displays visitor activity distribution across all social media platforms. Finally, we display the average rating for posts that mention this location.

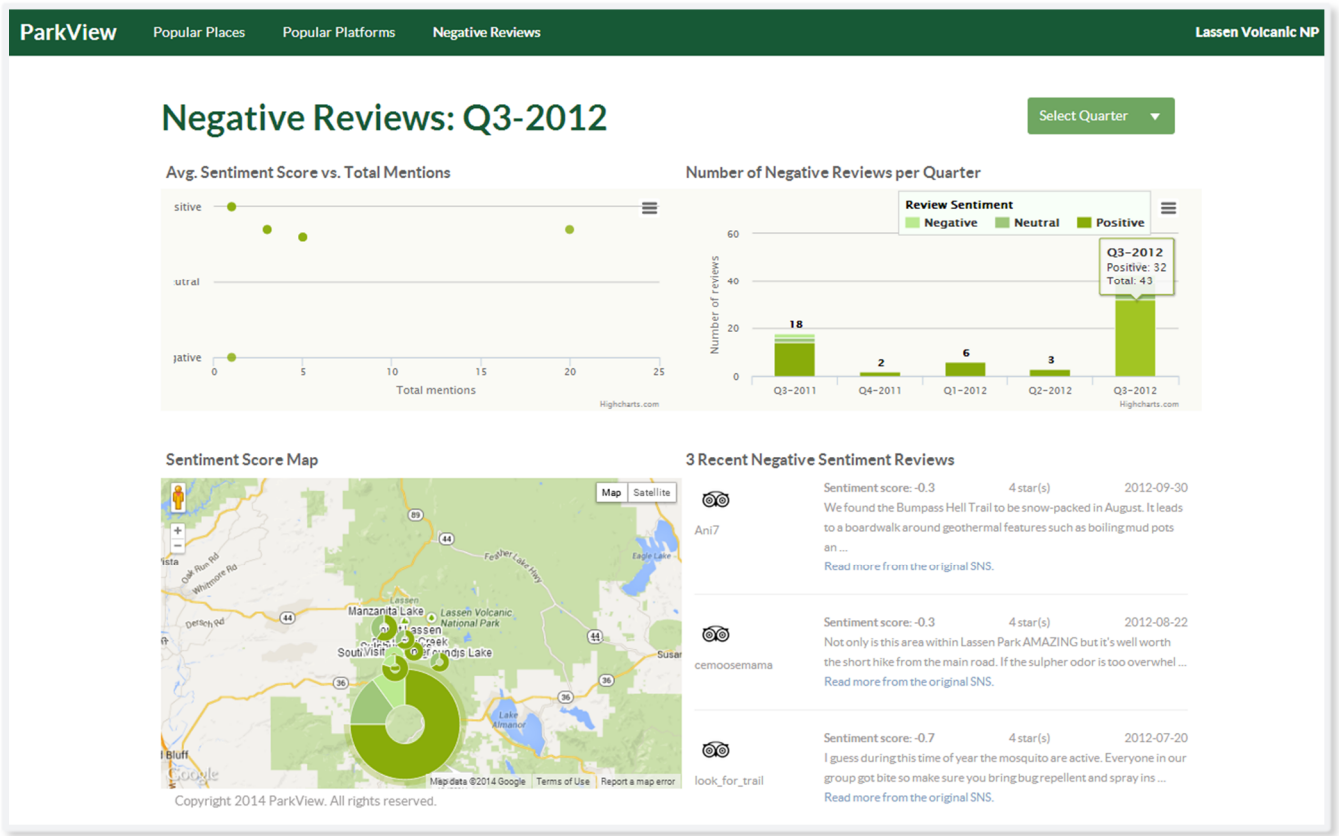
Figure 11. Final Popular Platforms dashboard



Negative Reviews dashboard. The final design of the Negative Reviews dashboard (Figure 12) is focused on an answering the following question: 'What are the key problems that the visitors bring up?' It provides an overview of the general sentiment about park services. It is also designed to help park staff surface problems and discover opportunities for service and facilities improvements.

The bar graph depicts the historical trend of visitor reviews across multiple social media platforms. In addition, it shows what proportion of the reviews expressed a positive, negative, or neutral sentiment. The scatter plot summarizes average sentiment scores associated with reviews that mention specific park places. The map marks park locations mentioned in the reviews with donut charts. The size of the donut charts indicates total number of reviews with place mentions. The slices convey what proportion of these reviews contained positive, negative, or neutral message. Finally, we are displaying excerpts from three recent negative reviews to incorporate 'voice of a visitor' into the dashboard. The links to the original source of the reviews is designed to encourage users connect to the original site and explore visitor feedback.

Figure 12. Final Negative Reviews dashboard



Final User Assessment

Research Methodology

Our final user assessment uses qualitative research design to answer the following questions:

- How do park Chiefs of Interpretation read and interpret the graphs on the dashboards?
- What elements of the graphs are easily understood and what elements could be easily misinterpreted?
- After gaining an understanding of what the graphs represent and what data the system provides, what do they perceive as the most useful?

Procedure. To answer these questions, we conducted user tests of the final dashboards. Because our partner parks are scattered throughout the country, tests were conducted over the phone. During each test, a park Chief of Interpretation received an email with a link to the dashboards (http://www.usparkview.com). Then, for each dashboard, we asked: 'What do you think each graph/table in this dashboard means? What is it showing?' After recording users' opinions and feedback, we clarified any misconceptions and asked: 'Which of these graphs/tables provides the most valuable information? How can they be even more valuable?'

Participants. Chiefs of Interpretation from all nine partner parks agreed to participate in user testing. As of the present time, two have participated and seven are scheduled to participate in the future. (Note: Park Chiefs of Interpretation have extremely busy schedules. Although we did our best to conduct interviews with each one of them before this report was due, several had to reschedule due to last minute conflicts.)

Final User Assessment Results

Although final user testing is incomplete, much of the feedback we have received falls into one of the following categories:

- The data presented in the dashboards is new and interesting to Park Chiefs of Interpretation. The negative reviews dashboard is particularly interesting and useful.

- For every aggregate statistic, the users were interested in seeing the text of the reviews that composed them. They were pleased with the sections of the dashboards that already did this and suggested additional drill-down capabilities for the sections that did not.

- Various small aesthetic changes were suggested to make the dashboards easier to read and interpret without explanation. Although the particular recommended changes vary among users, upping the contrast between the colors that differentiate among negative, neutral, and positive reviews would be helpful to several.

- Choosing an appropriate time range for each dashboard to cover is difficult. Though calendar year quarters are meaningful to some Chiefs of Interpretation, they are hard to internalize for others. Similarly, using the title ‘Q3-2012’ instead of ‘Quarter 3 – 2013’ was confusing to some users.

- The Negative Reviews dashboard appears to be the most potential to enable true data-driven management decisions. It could be even further enhanced by breaking negative reviews into categories specific to the managers responsible for them.

Future Development

Most of the initial scope for our project has been completed. Final deliverables agreed upon have been presented to our partners. Nonetheless, we would like to spend more time on the following aspects of the project:

Extend dashboard interactivity. Currently, ParkView dashboards offer very little user interaction. Users are able to zoom-in map views, hover over graph elements to get detailed information, and use links to connect to the original source for visitor reviews. We would like to introduce interactive features to the graphs and charts that would allow users to drill-down to more detailed information behind each graph element. Also, we would like to build interactive analytical graphics that would allow users to manipulate graph inputs and explore the results (e.g., 'what-if' analyses).

Collect more data. We were able to collect data from a variety of sources including social media platforms, visitor survey results, and publically available visitor and traffic counts data. Our social media datasets include information from 5 social media sites including Yelp, Flckr, Twitter, TripAdvisor, and Facebook. We would like to further augment the database with data from YouTube, Google Plus, as well as internal data sources that, although not publically available, could be shared without any privacy concerns.

Build more dashboards. While current set of dashboards provides information needed to address users' immediate needs, there are many more strategic questions and concerns that could be addressed with the data we collected. Spending more time to select, design, and implement additional dashboards could further enhance our final product. For example, conducting further semantic analysis of negative reviews would allow us identify, categorize, and present reviews' main topics.

Database interaction. A great majority of our end users do not possess knowledge of SQL language and will not be able to query the database to get additional information. Thus, our users can benefit from a well-designed front-end tool that could help them easily retrieve additional data.

Training modules. Providing simple web-based training modules with well-designed tutorials on how to interact with the ParkView system and retrieve information from the database would be a great complement to our final product.

Conclusion

Our team has achieved its goal of building a functioning proof of concept system for our clients at the National Park Service. The ParkView system is comprised of three main components: the data warehouse, a set of data transformation and analytical pre-processing algorithms, and the front-end summary dashboards. Based on the final user testing and feedback, the ParkView project is being considered for implementation in at least one national park. This fact alone makes our project a significant accomplishment. And, more importantly, our project provides a tangible solution to an important U.S. government agency which might not have been able to develop it otherwise.

Furthermore, this project gave us an opportunity to draw on knowledge we gained from multiple disciplines across the ISchool curriculum and practice our skills while working on a real-world problem. During the data warehouse design phase we relied on our knowledge of relational database management systems acquired in the Database Management course. We drew on front-end design methods and tools learned in User Interface Design and Information Visualization courses during the dashboard design and implementation stage of the development process. Moreover, we acquired new skills, knowledge, and experience by trying new things, making mistakes, changing direction, and working with each other.

References

Cairo, A. (2012). The Functional Art: An Introduction to Information Graphics and Visualization. *New Riders*.

Few, S. (2006). Information Dashboard Design. *O'Reilly*.

Hagerty, J., Sallam, R., Richardson, J. (2012). Magic Quadrant for Business Intelligence Platforms. *Gartner for Business Leaders (February 6, 2012)*.

Han, J., Kamber, M., Pei, J. (2012). Data Mining: Concepts and Techniques. *Morgan Kaufmann.*

National Park Service. (n.d.). Retrieved May 2, 2014 from http://www.nps.gov/hfc/services/interp/FearlessEvaluations.cfm

Tufte, E., Graves-Morris, P. (1983). The Visual Display of Quantitative Information (Vol. 2). *Cheshire, CT: Graphics Press*.

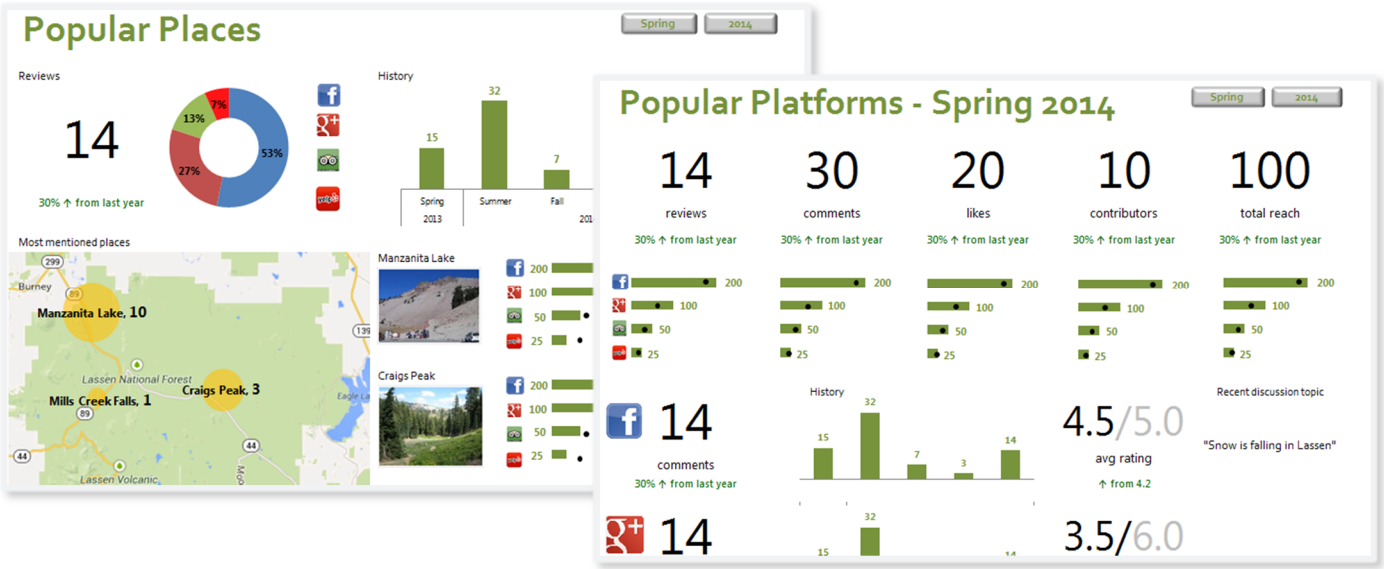
Vella, G. (n.d.). Retrieved February 5, 2014 from https://dribbble.com/gregoirevella

Appendix

A. Competitive Research:
Social Media Analysis Dashboard Designs by Gregoire Vella

B. Additional Lo-Fi Prototypes

C. Additional Hi-Fi Prototypes



D. Sample Business Intelligence Software Comparison Matrix

	SiSense	birst
AWS	Yes	Yes
Ad-hoc query	Formulate complex queries without SQL	Yes
Text mining	Yes	?
Data integration	High performance database, ETL – all-in-one solution	Integrate data from SAP BW, Salesforce.com, NetSuite, Marketo, Microsoft Analysis Services, and other operational systems; integrated ETL
MySQL	SQL Server, Oracle, ***MySQL***, PostgreSQL, Excel, CSV files, Google, SalesForce, ZenDesk; use API to connect to data	Oracle, DB2, SQLServer, ***MySQL***, Sybase, and PostgreSQL and any JDBC compliant data source
Share with others	Yes, also integrate with SharePoint, internal/external websites, and other content management systems	Annotate dashboards, export to Excel, PDF, and PPT; automated report distribution
Cost	$.60 - $2.00 per hour ($.96 - $10.00 per hour including AWS)	From $995 per month
Screenshots		
Other	Mash-up and manage data using drag-&-drop. Customize visualizations using JavaScript.	Drag-&-drop data management tools iPad capability Good for mid-size companies and government

	tableau	Yellowfin
AWS	No	No
Ad-hoc query	Not clear	Not clear
Text mining	?	?
Data integration	Yes, provides data management tools	Yes
Share with others	Publish dashboards/reports online	Integrate with websites, annotate dashboards, create alerts, create discussions
Cost	?	$3,000/year for 5 users
Screenshots		
MySQL	Yes	Oracle, PostgreSQL, MS SQL Server 2000+, ***MySQL 4+,*** ODBC, OpenEdge, Interbase, Firebird, Sybase, SQLite, Teradata, MS Access, Greenplum
Other	"Show me" feature that presents visualization that best fits data OS: Windows, web browser	Available for mobile BI Storyboard feature for presentations Designed to be easily scaled OS: Windows, Mac, web browser