Spot and Park: Parking management system

Tandeep Sidhu
Computer Science Dept.
Univ. of Maryland, College Park
tsidhu@umd.edu

ABSTRACT

When visiting a new place for the first time or going to an event one must often deal with finding a car parking space. This can be a challenging and frustrating task for number of reasons. In most cases, location of parking spots is not known before hand. Even if it is known, the parking location is not ideally suited (might not be the closest or cheapest). Furthermore, if one is visiting a location for some event, one must try to find a parking space amongst the limited parking spaces that are available at events attended by large group of people. Parking spaces can also have restrictions attached to them. Some parking lots are available only during certain hours of the day or they may only allow parking by people with permits. To help navigate through all these issues, we present an application that provides relevant and context sensitive information to visitors who need to look up parking. This information is available in real time thus providing user the necessary information to make informed decisions.

1. INTRODUCTION

According to the last count, there are over 45 billion user generated documents on the Internet [5]. These documents have been a tremendous source of information and are used by billions of people daily to answer all sorts of questions. With the advent of mobile phones in recent years, people look to access the same information that is available on the Internet from their mobile phones. The difference between information retrieval tasks before mobile phones and after mobile phones is that since the mobile phones, people tend to access lot more information while they are on the go (walking, driving, traveling, flying, etc). This leads to lot of information retrieval tasks that are location specific. For example, while driving, I might need to look up the nearest gas station or the nearest place to eat.

In general, location specific information is available on the Internet these days. For example, on Google Maps (a popular maps application) [6] we can do location specific searches and find addresses, restaurants, gas stations, and many other things, all relative to the user's current location.

Unfortunately, the location specific information is only available for major or popular entities like gas stations, restaurants, addresses, etc. Information for things like parking is not available or is only sparsely available. For example, if we look up parking information for the University of Maryland, College Park (UMD) campus on Google Maps, we just see the location of the main parking lots on the campus (see Figure 1). No detailed information is available on any of the parking lots. We do not know the timings the parking lots are available or whether there are any restrictions in parking in the lots. If we search for the parking for UMD again on another maps application, Bing Maps [7], we do not even see any of the parking lots shown by Google Maps (see Figure 2). In lot of situations, traffic and parking information is available through specific website (in this case UMD's own transportation website [8]). The transportation site does provide a detailed map of the campus with all the parking lots highlighted. The only problem with the parking information through this site is that such information is not geo-tagged so as a user one cannot get driving directions to it.

The issue of sparsely available parking data at UMD is just one example. We found similar lack of data on other types of locations. The only places that tend to have detailed parking information are major hubs like airports, bus stations, train stations, etc. In such locations, not only location and availability of parking lots is available but in some locations one also get to see real-time availability of parking spots. For example, at parking garages next to the Baltimore Washington International airport (BWI) in Baltimore, Maryland, USA, one can exactly see the number of open parking spots on an electronic display board. At BWI, such information is not available online but at other airports it is available (for example, at the Minneapolist International Airport [9]).

In order to address the issue of sparsely available parking information, we propose a system called Spot and Park. This system will allow an organization like UMD to easily create and share their parking information with their users/visitors by making the information available online. The users then could easily access the information via their computers or mobile devices. The information will also be available in real-time thus allowing the users to make informed decisions quickly.

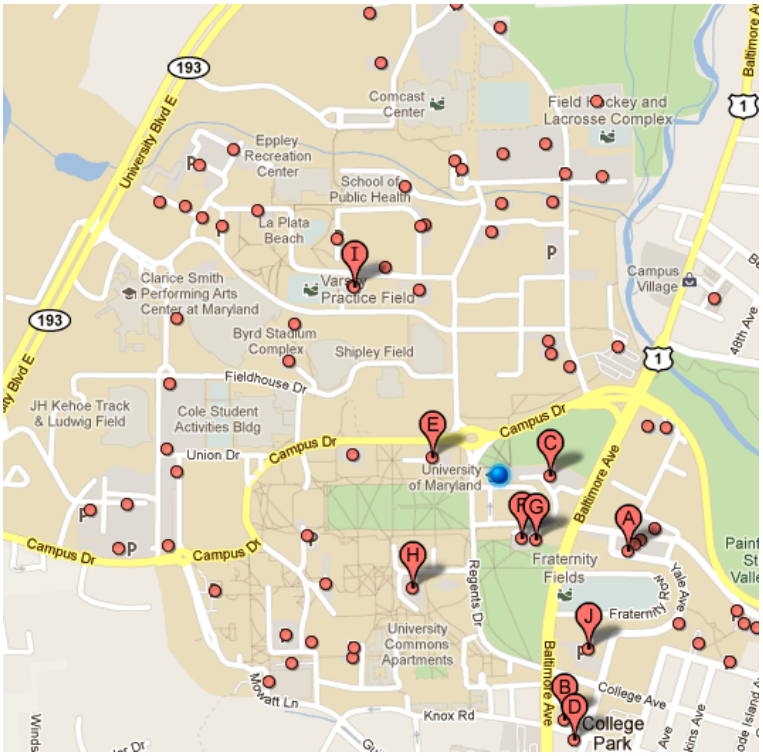

Figure 1: Search results for parking at UMD in Google Maps

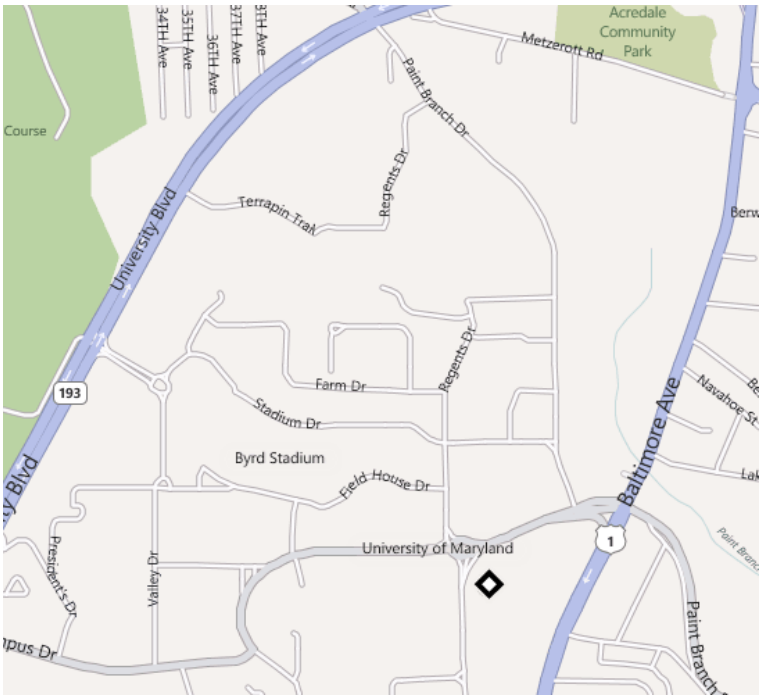

Figure 2: Search results for parking at UMD in Bing Maps

Spot and Park has been architected to work in variety of environments but we demonstrate its capabilities by applying it to the University of Maryland, College Park campus. The UMD campus is a community of about 50,000 students, faculty, and staff who come in to campus every day to study, teach, and work. In addition to that, there are lot of visitors who come in to campus everyday for meetings, sports, and other cultural or social events. Spot and Park has been

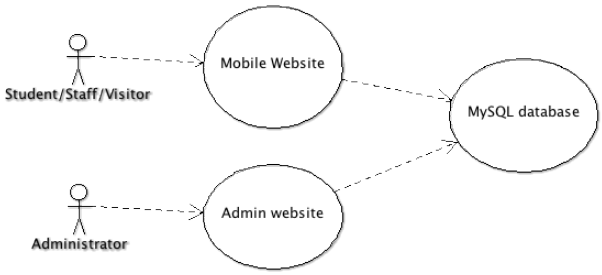

Figure 3: Architecture of Spot and Park

designed to address the parking needs of all these people. Using this application, parking administrators can easily associate parking lots with an event going on campus. Then when a visitor needs to lookup parking for an event, they can see all the parking lots associated with the event. The application is smart enough to show the parking lots sorted in distance (closest first) so that the user sees the most appropriate parking lot on top of the list. Event visitors can also get driving directions to any of the parking lots and they can also see real-time parking availability.

The organization of this paper is as follows. In the next section, we discuss work that is closely related to Spot and Park. In the following sections, we present the design and architecture of Spot and Park.

2. RELATED WORK

There are number of applications and solutions that are trying to provide parking data to users. In addition to sites like Google Maps and Bing Maps, there are other websites like Parkopedia[1], ParkingSpotter[3], and Primo Spot[2]. All these sites are specifically designed to help the user find parking at any location in US. They all feature mobile apps that users can use while on the go to lookup nearest parking. Unfortunately, all these services suffer from lack of data. For example, search for parking in the College Park, MD area shows no close available parking in any of the 3 websites (which is not true). The closest available parking is at a train station 3 miles away. This global approach of providing information for every town in this case is not working. A more targeted solution needs to be used like Spot and Park where individual cities or organizations can publish their parking data easily and share with their users.

There have also been efforts in providing real-time parking availability information to users. City of San Francisco is in the process of testing a technology called SFPark[4] which they have applied to their 28,800 parking meters and 12,250 parking spaces in garages that provides real-time parking information. The technology uses sensors in each parking spot to figure out if the parking spot is available or not. Users can use a mobile app to access this parking information and to also pay for their parking. The SFPark technology is designed to increase (or decrease) the cost of parking based on demand. In areas or during times of high demand, parking fees is higher compared to areas of less demand. This sort of technology seems as the ideal solution to solve the parking availability issues but its high cost of deployment (each parking meter needs to have a sensor), makes it less practical. Spot and Park doesn't completely replace the need of parking sensors to collect real-time parking availability information. Instead, we provide an architecture in which real-time information from automated sensors, attendants at each parking lot, or by some other means could be easily fed into Spot and Park.

3. SYSTEM ARCHITECTURE

Spot and Park software architecture consists of 3 components: a user facing mobile optimized website, an admin facing website, and a MySQL database (see Figure 3). We go over these 3 components in detail now.

3.1 The mobile optimized website

The mobile optimized website is the main user facing interface of Spot And Park. Its purpose is to help students/staff and visitors find parking at UMD campus. The application is also capable of providing driving directions to any of the parking lots. The application is designed using Web technologies (JSP, HTML, Javascript, JQuery) which has the benefit of making the application cross-platform so that it can be accessed on any mobile phone with a modern browser (that supports HTML5).

The mobile website is designed keeping in mind the mobile nature of mobile phones. Since these devices might operate in area with slow wireless networks, the mobile website has been designed to use minimum bandwidth to load and operate the website. Note that since this application is only available over the web, it always requires an active internet connection to operate. In the future, we will look to add certain offline capabilities that will allow the user to operate certain features of the application when there is no internet connection available.

We will go over the actual user interface design of the mobile website later in this paper.

3.2 Admin website

The admin website serves as a tool that parking administrators will use to manage the events that happen on the UMD campus. Using this application, they could create new events, attach available parking lots to a given event, manage the amount of available parking for an event, and control the availability of parking lots. Thus, this application serves as the main control module for managing all available parking on UMD campus. The data that parking administrators enter through this application will be saved in a MySQL database (discussed below) from where it will be access by the mobile website used by the students/staff and visitors. This admin interface is designed using PHP/HTML.

We will go over the actual user interface design of the admin website later in this paper.

3.3 MySQL database

The MySQL database is used to house all the parking, event, building, and parking availability information. Some of this information will be static information like parking lot information (name, location, geo-coordinates, description) which has to be entered on-time manually into the database. There will also be lot of dynamic information like calendar of events which will be entered through the admin interface. In the database, we will also track the number of open spots for each parking lot. This data can be entered through the mobile website (admin only function) but because of the open nature of the database, could also be fed through other sources. For example, if we have a source that feeds us real-time availability of a parking spot (perhaps through sensors embedded on each parking spot), we can feed that information directly to this database. Once that information is in the database, it can be easily used by the mobile interface to show all users availability of parking lots.

4. USER INTERFACE

4.1 Mobile Website

The user interface of the mobile website is designed from the perspective of 3 types of users: student/staff who go to UMD, visitors who frequently have to come to campus for meetings and events, and administrators who need to manage the parking lots and events (see Figure 4).

When a user loads the mobile website for the first time, they will be asked to specify what type of user they are. Based on their selection, they would be shown the appropriate functions. Note that once a user has specified their role in the system (either student, visitor, or admin), they do not have to specify the role again. So, next time the user loads this application, they will be automatically assigned their previous role. We also provide the feature of allowing the user to change their role if needed.

The whole application is built using mobile optimized interface which easily scales the size of the interface to match the screen dimensions of the mobile browser. It uses easy navigation controls. Users can click on a link in any page and that page will be rendered using Web 2.0 style page transitions. Users can also navigate back to their previous page by pressing the back button in the application or the back button in their browser. They also have the option to press the home button to go back to the home screen.

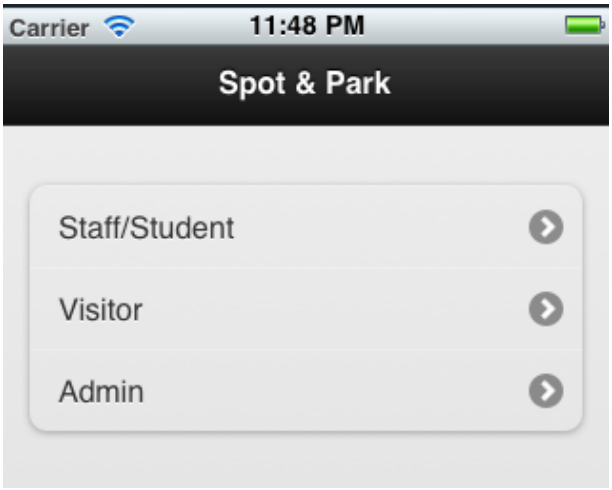

Figure 4: Spot and Park home screen

Next we go over the interface design from the perspective of each user type.

4.1.1 Student/Staff

Students and staff typically have assigned parking area on campus. They come regularly to campus so they are aware of the roads and the general structure and location of the parking lots. They regularly attend events on-campus and sometimes these events are not close to their designated parking areas so they also have to find nearest parking. Because certain events on campus (like football or basketball game) tend to take over parking lots allotted to people, students/staff are often asked to move their car before an event starts. So, this set of users could benefit from being notified if there is an event happening on campus that would affect their parking situation on a certain day. We have added this feature in the interface and is discussed below in detail.

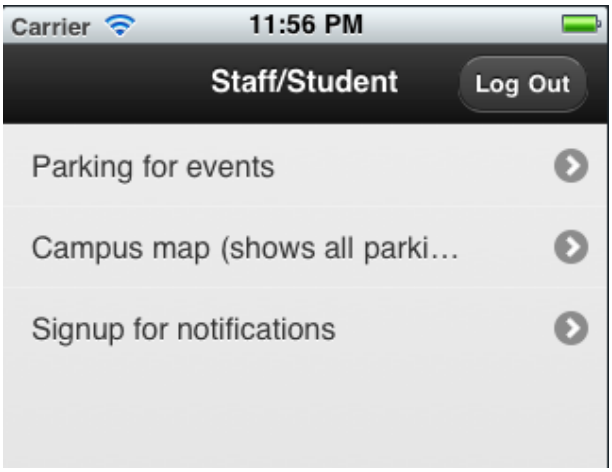

Figure 5: Student/Staff home screen

The mobile website for students/staff provides 3 main functions :

1. **Parking for events:** This shows events that are up-

coming (see Figure 6). The events are ordered in chronological order and separated by categories. User can click on an event which will show the parking that is available for that event (see Figure 7). The list of parking lots is ordered by distance from the event so closest parking is shown first. If the parking lot has some restrictions or is a paid parking lot that information is also shown. The availability of the parking lots is also shown including the number of available parking spots for each of the lots. The row displaying each parking lot is also color coded (green = available, red = unavailable) so that the user can easily spot the parking lots that are available.

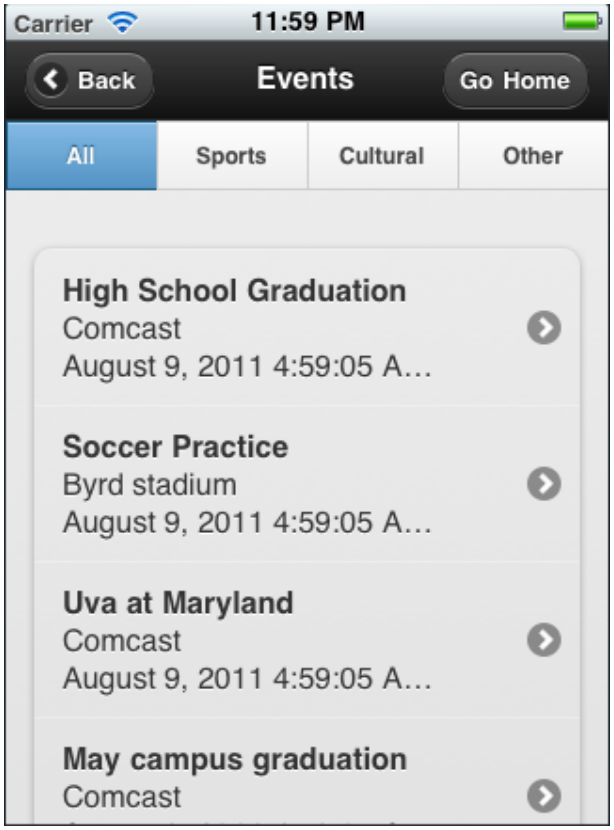

Figure 6: List of events

Users can look at the list of parking lots to decide which is most appropriate for them or they can look at the parking lots in a map by clicking the View on Map button. In either of the views (list or map), a user can click on a parking lot, get more information about it, or get driving directions to the parking lot from their current location or some other address (see Figure 8).

2. **Campus map:** The campus map function will show the location of all parking lots on campus. The parking lots can be viewed in a list or on a map. Same as the Parking for events feature, user can click on any of the parking lots to get more information about it or to get driving directions to it (see Figure 9).

3. **Signup for notifications:** As previously mentioned, this feature allows a student/staff to signup for notifi-

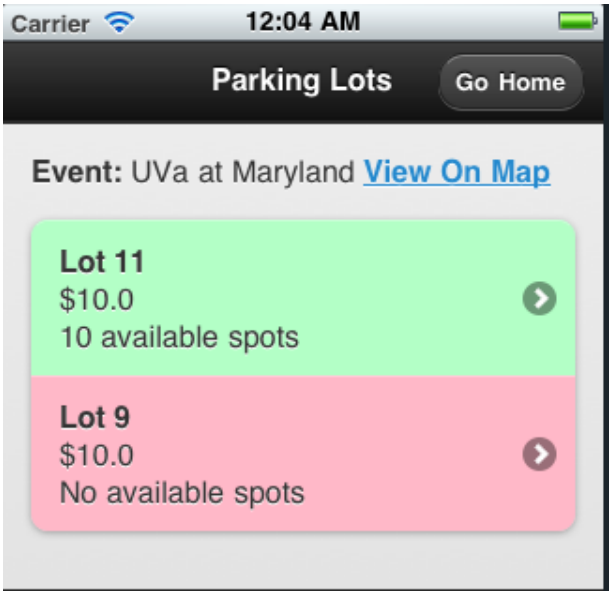

Figure 7: Parking locations for an event

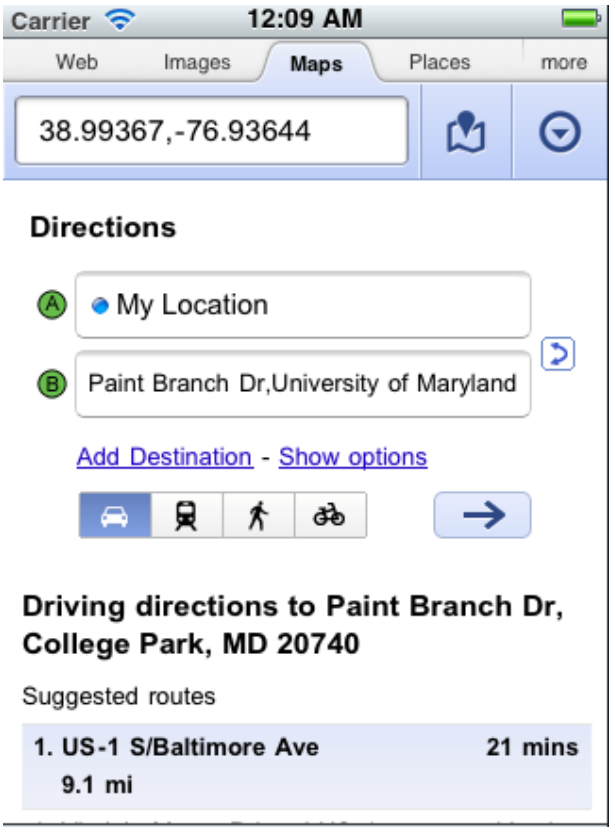

Figure 8: Driving directions to a parking lot

cations of any upcoming disruptions to their parking assignments (see Figure 10). Users need to specify their parking lot assignment and their email address. If in the future, there are any disruptions, we would email the user with the details of the event includ-

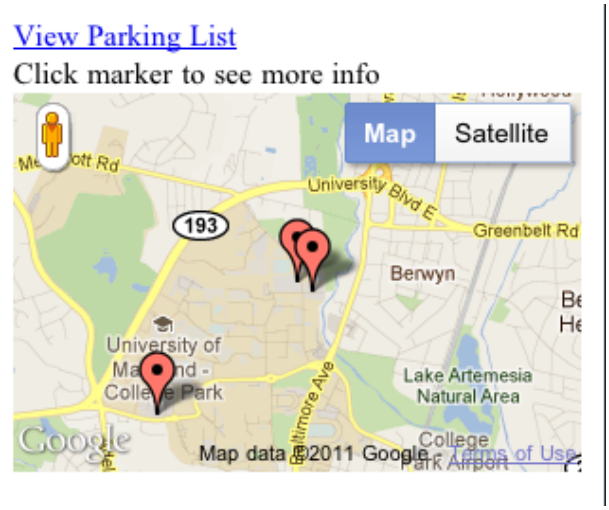

Figure 9: All parking lots shown on a map

ing instructions on what to do for alternative parking. Note, that this targeted notification is better than blasting the entire campus community with an email regarding an event that might not affect everyone on campus.

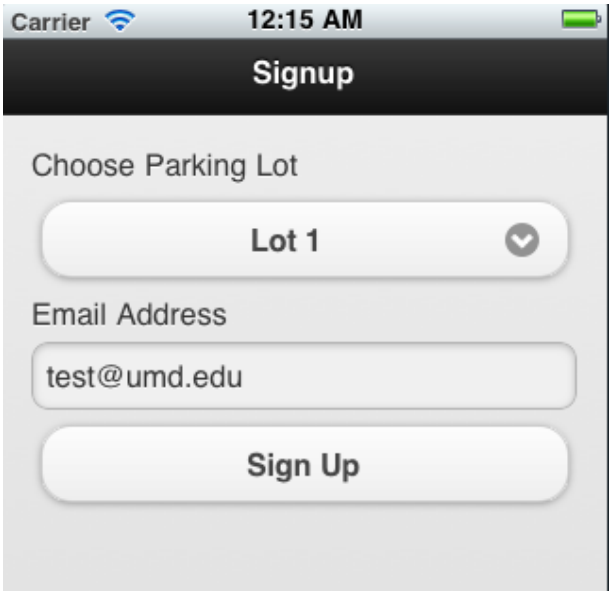

Figure 10: Allows a user to signup for notifications

4.1.2 Visitor

In most cases, visitors have similar needs as student/staff. They need to be able to find parking for events that they might come to campus for. Visitors might come to campus for also other reasons like attend meetings. In these situations, they are not coming in for some large event but are coming in to meet someone in a certain building. So, they just need to know where they can find closest parking for their meeting. So, we provide them the functionality that helps them find that information.

The mobile interface for visitor provides 2 main functions (see Figure 11):

1. **Parking for events and campus map**: This works exactly the same way as it does for student/staff (explained above).
2. **Parking for buildings**: This allows a user to select a building on campus and then look at the closest available parking. User can view the parking lots on a map or in a list view. User can also get to see if a parking lot has fees attached to it and the hours of operation. They can also look up driving directions to any of the available parking lots.

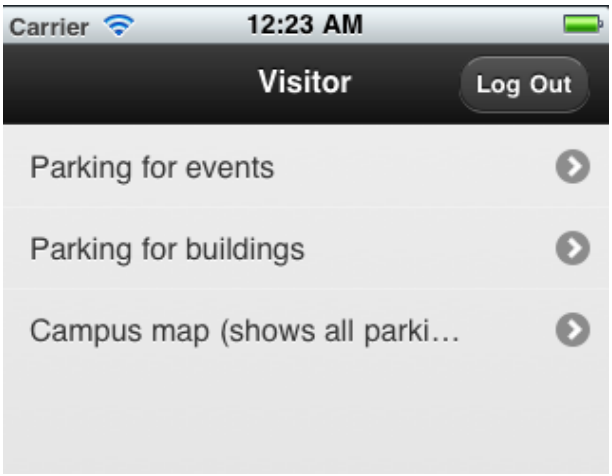

Figure 11: Visitor home screen

4.1.3 Admin

The main function parking administrators will need access to is the ability to manage the parking lots when an event is going on. They need to be able to look at the parking for any of the events, specify how many open spots are in each parking lot, and to mark each parking lot as full (thus making it unavailable for all users). We provide all this functionality through the mobile interface so that the parking administrators can easily mange the lots while they are in the field (see Figure 12).

4.2 Admin website

As previously mentioned, the admin website is used by the parking administrators to manage the events and parking. Administrators can create new events, associate parking lots with an event, change the availability of a parking lot, and do other administrative tasks through this interface (see Figure 13).

5. FUTURE WORK

Currently, to provide the real-time parking spot availability, in this application we rely on the parking administrators to manually update the parking availability. As previously mentioned, if the parking availability data is provided from some other automated systems (like parking sensors in the parking spots), then we can integrate that data with our

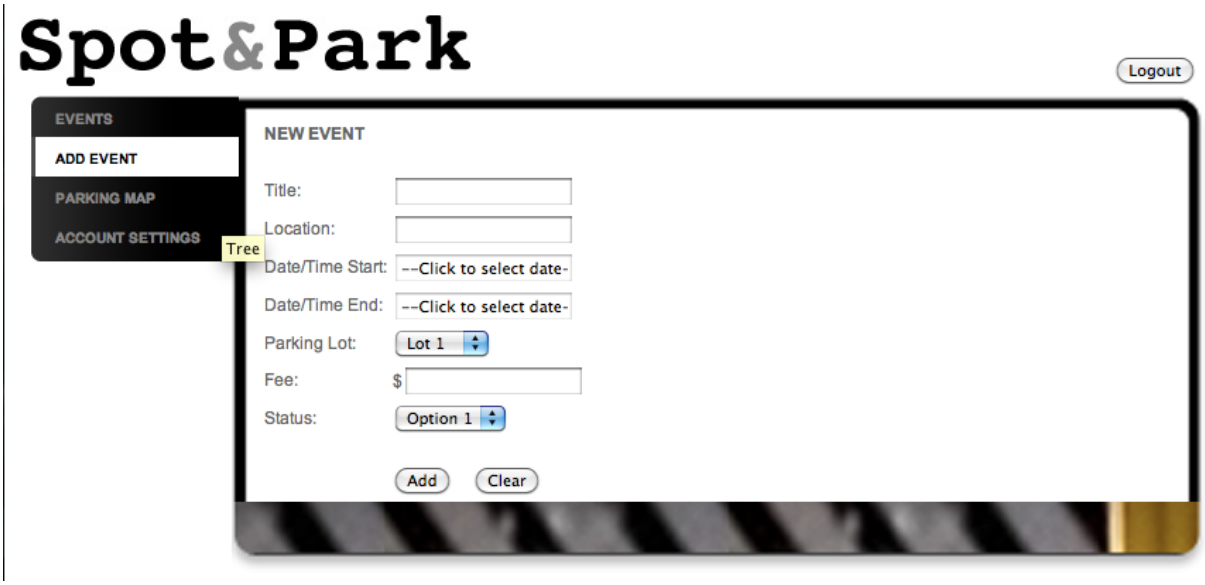

Figure 13: Admin website

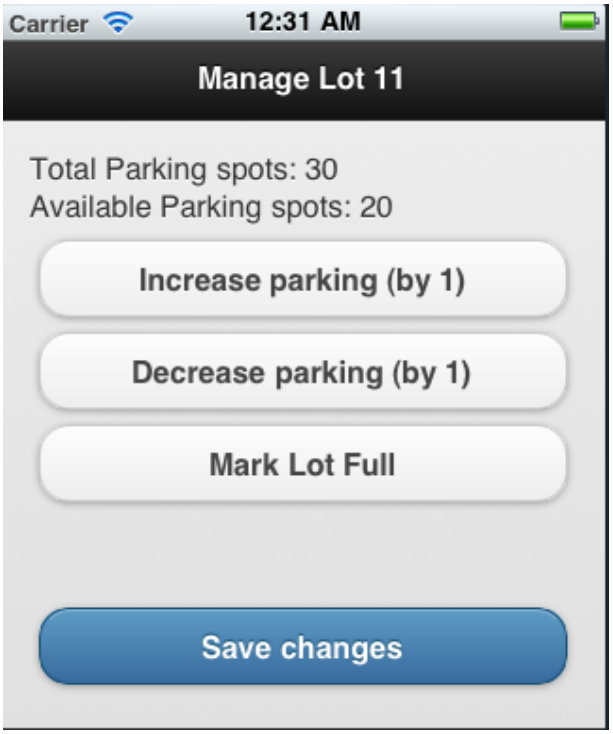

Figure 12: Management of a parking lot

application. Furthermore, there is also the possibility of involving the students and visitors in collecting the real time data. For example, if I go to a parking lot, and I notice that there is no availability, I could update that in Spot and Park. And this will be useful to anyone who uses the Spot and Park after I do. Using data collected from users will help build our information database much more quickly but we must watch out for the reliability of this data as it is not being collected from trusted sources.

Currently, in this application, the location of parking lots and buildings is fixed and is manually fed into the database. In future, we need to add more functionality to the mobile website (or the admin website) so that the building and lot information can be easily entered into the system. Since specifying buildings and parking lots also requires specifying geo-coordinates for these entities, this data collection is especially suited to be built in the mobile website where an administrator could use the GPS on a mobile device to find out the geo-coordinates of a building or a parking lot.

We have demonstrated the use of Spot and Park in a university environment. We should test it out in other environments as well which would help us make it a much more robust system.

6. CONCLUSIONS

In this paper we have described an application called Spot and Park that aims to solve the problem of finding parking. This application with its flexible architecture and easy to use interface will provide transportation or parking administrators easy tools to deploy a parking management system in their organization. This application also has an easy to use mobile interface that will help guide visitors in finding parking for their destination.

7. ACKNOWLEDGMENTS

I would like to thank Dr. Ashok Agrawala for his continuous guidance on this project.

8. REFERENCES

[1] Parkopedia `http://en.parkopedia.com/`.
[2] Primospot `http://primospot.com/`.
[3] Parking Spotter `http://www.parkingspotter.com/`.
[4] SFPark `http://sfpark.org/`.
[5] Documents on the web `http://www.searchenginejournal.com/`

yahoo-boasts-size-of-its-search-engine-index/2036/.

[6] Google Maps http://maps.google.com/.

[7] Bing Maps http://www.bing.com/maps/.

[8] Univ. of Maryland, College Park Transportation website http://www.transportation.umd.edu/maps.html

[9] Minneapolis International Airport http://www.mspairport.com/parking/surepark.aspx