EVALUTION AND DEVELOPMENT OF A UNIVERSITY VISITOR PARKING MANAGEMENT FRAMEWORK

FINAL REPORT

PennDOT/MAUTC Agreement Contract No. 510401
VT-2008-06
DTRS99-G-0003

Prepared for

Virginia Transportation Research Council

By

Sashikanth Gurram and Hesham Rakha
Virginia Tech Transportation Institute

May 2010

This work was sponsored by the Virginia Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
EVALUATION AND DEVELOPMENT OF A UNIVERSITY VISITOR PARKING MANAGEMENT FRAMEWORK	May 2010
	6. Performing Organization Code
7. Author(s)	**8. Performing Organization Report No.**
Sashikanth Gurram and Hesham Rakha	VT-2008-06
9. Performing Organization Name and Address	**10. Work Unit No. (TRAIS)**
Virginia Tech Transportation Institute 3500 Transportation Research Plaza Blacksburg, VA 24061	**11. Contract or Grant No.** DTRS99-G-003
12. Sponsoring Agency Name and Address Virginia Department of Transportation ****** U.S. Department of Transportation Research and Innovative Technology Administration UTC Program, RDT-30 1200 New Jersey Ave., SE Washington, DC 20590	**13. Type of Report and Period Covered** Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Visitors constitute an important component of a university business. Given that visitors are typically unfamiliar with university campus layouts, special assistance may be needed to assist them with their parking needs. For example, personal and follow-up interviews conducted with Virginia Tech visitors revealed that approximately 28% of the visitors did not know the location of the most convenient parking lot to access their destination and approximately 32% of the visitors had to visit at least two parking lots before finding a parking space. This paper attempts to develop a simple web-based interactive parking management framework that uses the PHP Hypertext Preprocessor (PHP) and My Structured Query Language (MySQL) platform. This framework identifies the best possible parking lot(s) for a selected destination considering various factors including – the distance of the parking lot to the visitor's destination and historical occupancy of the parking lot at specific times-of-the-day and days-of-the-week. The web-based interactive system is designed to provide the visitor with two or more choices for parking that meet the visitors search criteria.

17. Key Words	18. Distribution Statement
Database management systems, Parking management framework, University parking management, Intelligent transportation systems	No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	6	

TABLE OF CONTENTS

LIST OF FIGURES

Evaluation and Development of a University Visitor Parking Management Framework

Sashikanth Gurram and Hesham Rakha

***Abstract*— Visitors constitute an important component of a university business. Given that visitors are typically unfamiliar with university campus layouts, special assistance may be needed to assist them with their parking needs. For example, personal and follow-up interviews conducted with Virginia Tech visitors revealed that approximately 28% of the visitors did not know the location of the most convenient parking lot to access their destination and approximately 32% of the visitors had to visit at least two parking lots before finding a parking space. This paper attempts to develop a simple web-based interactive parking management framework that uses the PHP Hypertext Preprocessor (PHP) and My Structured Query Language (MySQL) platform. This framework identifies the best possible parking lot(s) for a selected destination considering various factors including – the distance of the parking lot to the visitor's destination and historical occupancy of the parking lot at specific times-of-the-day and days-of-the-week. The web-based interactive system is designed to provide the visitor with two or more choices for parking that meet the visitors search criteria.**

***Index Terms*— Database management systems, Parking management framework, University parking management, Intelligent transportation systems**

I. Introduction

A Visitor [1] is defined as any person who owns, operates and parks a vehicle on the university campus and is neither a student nor an employee of the university. Visitors are an important component of a University business. Due to the high parking occupancy rate at Virginia Tech, finding a parking space can be a difficult task. A 2007 Visitor Center Survey estimated that Virginia Tech receives an average of 100 visitor parking pass requests per business day.

In an attempt to characterize the familiarity of the visitors with the parking facility's layout and the efficiency of the visitor parking services, the Virginia Tech Parking Management Study – Phase III was undertaken in 2007. This study concluded that approximately 20% of the visitors did not know the location of the offices they needed to visit and 28% of the visitors did not know the location of the most convenient parking lot to access their destination. Approximately, 32% of the visitors had to visit at least two parking lots before finding a parking space. Approximately 6% of the visitors rated the parking services offered by Virginia Tech poorly and 28% of the sample did not respond to this question. Consequently, there is ample opportunity to enhance the existing parking services on campus. The survey also revealed that the majority of the visitors showed an unwillingness to pay for designated visitor-only parking spaces. Consequently, to serve the visitors in a more efficient manner without subjecting them to an additional financial burden, the Center for Sustainable Mobility (CSM) at the Virginia Tech Transportation Institute (VTTI) proposed to develop a Web-based Interactive Parking Guidance and Information Application which would provide the user with a list of the most convenient parking lots to access his/her destination.

Parking Management refers to policies and programs that result in more efficient use of parking resources [2]. There are several parking management principles that assist in parking management. Some of the widely used parking management principles are consumer choice, user information, sharing, and efficient utilization of the parking facilities. In this paper, the user information principle of parking management is used to manage the parking services for the visitors. The user information principle advocates that motorists should have information about the parking and travel options that are available. The web-based interactive system that is developed, aims at informing the visitors regarding the best possible parking lots and their locations on campus.

PHP, which stands for *PHP: Hypertext Preprocessor*, is used to code the system to generate output in the form of a list of the most convenient parking lots based on user input criteria. The variables used to execute the code are the time, day, month and the user input criteria including the destination location (name of the building), and criteria for selecting parking lots. These variables are used to retrieve data stored in database objects called tables in My Structured Query Language (MySQL). A query is generated using the PHP code which performs a search through the stored variables in the MySQL database and returns output.

Although there have been several attempts [3, 4] made at developing real-time parking information and guidance systems (PGIS) most of these attempts are restricted to small and large cities. For example, Chou et al. developed a PGIS named CanPAS [5] for university campuses but the infrastructure setup is cost prohibitive for most universities.

Manuscript received November 2, 2009. This work was supported in part by the Virginia Tech Transportation Office.

Sashikanth Gurram.is a graduate student with the Charles Via Jr. Dept. of Civil and Environmental Engineering at Virginia Tech and a graduate research assistant with the Center for Sustainable Mobility at the Virginia Tech Transportation Institute, Blacksburg, VA 24061. (e-mail: sashi34u@vt.edu).

Hesham Rakha is an Associate Professor with the Charles Via Jr. Dept. of Civil and Environmental Engineering at Virginia Tech and the Director of the Center for Sustainable Mobility at the Virginia Tech Transportation Institute, Blacksburg, VA 24061 phone: 540-231-1505 (e-mail: hrakha@vt.edu)

Furthermore, a review of response to dynamic parking information [6] showed that the driver response levels to real-time parking guidance and information systems are lower than expected. Most of the surveys reviewed by Thompson [6] showed that regular parkers trust their judgment over the PGIS's judgment in finding a parking space. It would be an operational and financial disaster for the parking services department in a university if the real-time PGIS of little use to the university community. It would be more logical to roll out a PGIS framework, test it on real users, obtain feedback from the users and make improvements to the PGIS framework. In order to address these concerns a cost effective PGIS which uses historical occupancy data is developed. This PGIS is intended for a kiosk setup at the welcome or visitor center in a university where visitors will stop by to get their parking passes. The PGIS system, described in the following sections, can also be hosted on the university website (with few modifications to the code) and visitors can take advantage of it from their homes. It should also be noted that the proposed system can also operate using real-time data if instrumentation are installed in the parking lots.

II. Study Objectives, TASKS, and Paper Layout

The objectives of this study are to: (a) develop a web-based interactive framework to assist visitors in the selection of convenient parking lots to access their destinations; and (b) design the system in a flexible fashion so as to accommodate any future updates in terms of the databases and real-time integration.

The main tasks accomplished by the project are: (a) create a database; (b) generate the code to perform a search in the database; and (c) create a web-based system to receive input and display output to the user.

In terms of the paper layout, initially an overview of the Web-based interactive system is provided followed by a description of WAMP and WAMPSERVER. Subsequently, the advantages of the MySQL-PHP platform are presented followed by an overview of the MySQL database management system. The coding done in PHP is then discussed along with some illustration of the system. Finally, the conclusions of the paper are presented together with recommendations for further research.

III. Overview of the Web-based Interactive System

In order to provide the user with information of various parking lots, so that he/she may access their destination, a web-based interactive system is created. This system receives the user input with regards to their final destination, performs a search based on the given criteria, and finally retrieves and displays the results in a tabulated format. The results consist of a list of the most suitable parking lots to access the user's destination. User can input the details pertaining to the destination and some other relevant criteria using HTML forms. The data pertaining to the parking lots and buildings is stored in a database using a database management system. When the user requests for the parking lot information near his/her destination, the data in the database has to be retrieved and displayed to the web browser. For this purpose, a code is written which performs the retrieving and printing actions. Scripting languages are generally used to generate the code which performs these actions. A brief introduction to the WAMP system which is necessary to deliver the above tasks is provided in the next section.

IV. WAMP System and WAMPSERVER

The WAMP system refers to a package of programs running on the Windows operating system. WAMP is an acronym formed by using the initials of Windows, Apache, MySQL and PHP. To realize the goals and objectives of this research it was necessary to store the parking data in some location (usually a server) and retrieve the data when needed by the user. MySQL, which stands for My Structured Query Language is used as a database management system to create and store the data needed by the application. PHP is used as the scripting language needed for the dynamic web page design. Apache is the web server that forms a connection between the user computer and the main server computer where the database is stored. Apache transfers the user input to the main server and then transfers the results generated by the server to the user computer. The flow of data between a local computer (user computer) and the server computer where the data is stored is shown in Fig 1.

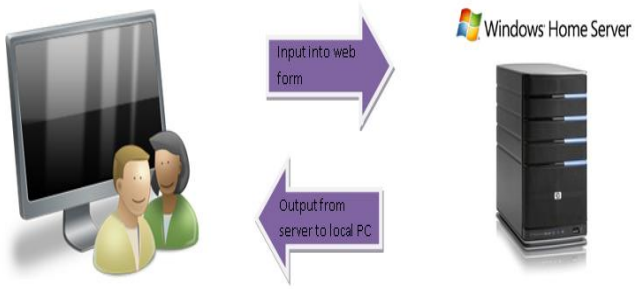


Fig. 1. Data Flow between a Local PC and a Web Server

For the purposes of developing this web-based interactive system, WAMPSERVER, which is a windows-based web development environment, was chosen. WAMPSERVER is an open source software and is free to download.

V. MySQL-PHP Platform

Any interactive database application will have what is known as a back-end and a front-end. The back-end generally refers to the database management system while the front-end refers to the external application used by the user to access the database. The front-end provides the user with an interface to access the back-end. In this application, MySQL is the back-end and PHP is the front-end. MySQL and PHP are frequently used together and are often referred to as the *dynamic duo* [3]. A MySQL-PHP platform has many advantages that include speed, ease of use, and cost effectiveness. In fact both MySQL and PHP can be downloaded and used for free.

All these advantages have led to the adoption of the PHP and MySQL platform for the development of the web-based

interactive parking information system. The flow of information or data using the proposed PHP-MySQL platform is shown in Fig 2.

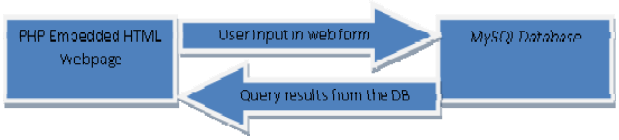


Fig. 2. Information Transfer between PHP and MySQL

VI. Database Management

MySQL is a relational database management system (RDBMS) that serves as an electronic filing cabinet that stores information in an organized fashion. Technically, a database refers to a structured collection of records or data that are stored electronically.

The first step in the creation of any database application is the creation of the database and the selection of the database platform. After selecting a database platform, the next step is to create tables. The structure of a table has to be planned in advance. After the creation of a table, the next logical step is to add, modify, update or delete data in the table. A database might consist of several tables. If the tables are not related to one another there is no need to create several tables. In a relational database, the individual tables which store the data are related to one another. For example, a entry in one table might be related to several entries in another table, based on the column attributes.

In creating the parking information system, data pertaining to the buildings on the Virginia Tech campus, Parking lots, typical number of empty spaces in each parking lot based to the time-of-the-day, day-of-the-week and month-of-the-year and distances between the parking lots and the buildings were constructed. In addition to these data a campus map was displayed to the user highlighting his/her destination building along with the parking lots located near his/her destination.

For the purpose of developing this parking information application, the database VT_VPM, which stands for Virginia Tech Visitor Parking Management, was constructed. Five tables were created to manage the data in an effective manner, as follows:

- Buildings
- ParkingLots
- Buildings_Lots
- Occupancy
- Image

A brief description of the contents of the columns in each table is described.

BUILDINGS: This table contains a total of two fields, as follows:

1. BUILDINGNAME: This field contains the names of buildings on the Virginia Tech Blacksburg campus. The field houses a total of 124 building names.
2. BUILDINGCODE: This field contains a unique code that is assigned to each building on campus. The code that was assigned to each building was taken from the Virginia Tech campus map.

PARKINGLOTS: This table contains a total of 16 fields, as follows:

1. PARKING_LOT_NAME: This field contains names of the parking lots on the Virginia Tech Campus.
2. PARKING_LOT_CODE: This field contains the code numbers assigned to the on-campus parking lots. The code numbers applied to the parking lots are taken from the Virginia Tech Parking Services.
3. FS: This field contains the number of parking spaces allocated to Faculty/Staff in the parking lot.
4. TA: This field contains the number of parking spaces allocated to Teaching Assistants.
5. COM: This field contains the number of parking spaces allocated to Student Commuters.
6. RES: This field contains the number of parking spaces allocated to Student Residents.
7. FS_COM: This field contains the number of parking spaces allocated for Faculty/Staff Commuters.
8. Visitor: This field contains the number of parking spaces allocated to Visitors.
9. CP: This field contains the number of parking spaces allocated to carpooling.
10. MC: This field contains the number of parking spaces allocated for Motorcycle.
11. Timed: This field contains the number of limited time parking spaces.
12. GEN: This field contains the number of parking spaces allocated to the general public.
13. SV: This field contains the number of parking spaces allocated to service vehicles.
14. HC: This field contains the number of parking spaces allocated to handicapped people.
15. Meter: This field contains the number of metered parking spaces.
16. Total: This field contains the total number of parking spaces in a parking lot.

BUILDING_LOTS: This table contains a total of three fields. In this table, all buildings are linked to all parking lots. Consequently, the combination of the parking lot code and building code creates a primary key. The fields in this table are named as follows:

1. BUILDING_CODE: This field contains the code numbers that are assigned to the buildings on the Virginia Tech campus. Since, there are more than one parking lot associated with a building the building code is repeated for all the parking lots associated with it.
2. PARKING_LOT_CODE: This field contains the code numbers assigned to the on-campus parking lots. As with the case with a building served by two lots, a parking lot may serve two different buildings and hence a repetition of parking lot code for the buildings being served by it is possible. Thus, the combination of the building code and the parking lot code forms the primary key.
3. DISTANCE: This field contains the approximate distance in feet from a central point in the parking lot to a specific building entrance.

OCCUPANCY: This table contains a total of five fields and contains information regarding the typical number of empty

spaces in all the parking lots in a given month on a given day and at a specific time. Thus, the combination of the parking lot code, month, day and time forms a primary key. The columns in this table are named as follows:

1. PARKING_LOT: This field contains the code numbers assigned to the on-campus parking lots. The parking lot codes repeat themselves for every different combination of month, day and time-of-the-day.
2. MONTH: In this field, the 12 months of the year are stored. The months are repeated for each combination of parking lot code, day and time.
3. DAY: This includes the days of the week. Though there are only five working days in a week, the names are repeated for each combination of parking lot code, month and time.
4. TIME: This field contains the time in the 24-hour clock format. Each hour is broken into four 15-minute intervals and these times are loaded into the column. The time span ranges from 8:00 in the morning until 18:00 in the evening. A particular time will repeat itself in the column for a different combination of parking lot code, month and day.
5. EMPTY_SPACES: This field contains the number of empty spaces in all the parking lots for each day, time-of-day, and month-of-the-year combination.

IMAGES: This table contains information regarding the location of an image file for each building, which is stored on a local computer. In this table the columns are named as follows:

1. BLDCODE: This field contains the code numbers that are assigned to the buildings on the Virginia tech campus. This column acts as the primary key for this table.
2. LOCATION: This field contains the paths to the various images.

VII. PHP CODE FUNCTIONALITY

The application requires the user to submit a query to the system. The system then displays the results a response to the query. In assisting the user with his/her interaction with the database, an HTML form was created that provided user-friendly drop down lists to input search criteria, the number of records to be displayed, and the destination building name, as shown in Fig 3. Users may prefer parking lots that are closer to their destination building even though there is a lower probability of finding an empty space. These users are clearly guided by the criteria of distance. Alternatively, other users may prefer to park in parking lots with a higher probability of finding empty spaces even though they may be slightly farther away from their desired destination. These users are influenced by the number of empty spaces. The criteria form allows the user to input their criteria and the results are displayed according to the preference of the user.

The Virginia Tech campus has a total of 124 buildings and 93 parking lots that serve the public. Each building on the campus is linked with all the parking lots and the combinations are uploaded to the BUILDINGS_LOTS table of the database. The user can select the number of records to be displayed from the database using the records form. The minimum number of records that can be displayed is 5 and the maximum number of records is 93. Once the user selects the search criterion (distance or expected number of empty spaces), number of records to be displayed, and the target building name and clicks the submit button, the user input data is sent to a particular file for further action. The query response data are assembled and then stored for future use. Information regarding the month, day, hour-of-the-day, and minute-of-the-hour are obtained from the actual time of the request. All this information is stored in individual variables. Data pertaining to the estimated number of empty spaces is discrete are stored at 15-minute increments. The system time is recorded for the hours and minutes individually. The minutes time is changed rounded down to the nearest 15 minutes as

- 00 if the local system time in minutes is equal to or greater than 00 and less than or equal to 14
- 15 if the local system time in minutes is equal to or greater than 15 and less than or equal to 29
- 30 if the local system time in minutes is equal to or greater than 30 and less than or equal to 44
- 45 if none of the above conditions are satisfied

Finally, the modified time is assigned to another variable to be able to run a query in the database. Using all the variables shown above a query is run to obtain the information regarding the estimate of number of empty spaces in different parking lots near a building and the estimated walking time from each parking lot to the building. If the user submits distance as the criteria, the SQL query retrieves the data and displays the parking lots in ascending order based on distance. If the user submits empty spaces as the criteria, the SQL query retrieves the data and displays the parking lots in ascending order with regards to the number of empty spaces.

The final task of this application is to display the image of the building the visitor intends to visit together with the adjacent parking lots. For technical reasons it is not possible to display the image directly using code in the first PHP file. Consequently, another file is created using a command in the first file to go to the second file when there is a need to display the image. The name of the building to which the user intended to go will be uploaded to the second PHP file from the first PHP file. Then, the path of the file stored on the local computer will be retrieved for that building name and finally the image is displayed to the browser.

VIII. MODEL APPLICATION

In this section some of the working configurations of the Web-based interactive parking lot information system are provided. The parking lot information system web page displays an HTML form that retrieves user input. The HTML drop down menus by default show the following messages: (a) Select the Criteria, (b) Select the Number of records you want to search, and (c) Select a Building. There is also a Submit button used to submit the user input.

Now the user can input the criteria of his/her selection using the drop down menus. If the user selects the Criteria as Distance, the number of records to be displayed as 5 and the building name as Burruss Hall, as shown in Fig. 4, the results are ordered by distance in an ascending order with the nearest

parking lot at the top. Now, if the same number of records is chosen along with the same building name but the criteria is changed to Empty Spaces as shown in Fig 5 the results are ordered by empty spaces in a descending order with the parking lots having the highest number of expected empty spaces at the top. In both cases, an image of the building with the closest parking lots around it is also shown.

IX. CONCLUSIONS AND FUTURE RESEARCH

The research presented in this paper presents a framework for a web-based parking management system that assists visitors who may not be familiar with the parking facilities on university campuses. The system is most beneficial in the case that the university has many parking lots to serve the university community. The system identifies the best parking lots available for a visitor depending upon his/her input criteria. The parking information system designed here is a prototype model which helps in the realization of the User Information Principle and thus paves the way for better parking management. This system is cost-effective because is does not require parking surveillance equipment, however the system is capable of operating using real-time parking information if surveillance instrumentation are installed. It is not sure if a real-time PGIS is a financially feasible solution to address the parking problems on a university campus. As such the PGIS developed at VTTI which uses historical data can be deployed free of charge on a campus and its operation can be gauged on a regular basis. Based on user feedback and system performance a decision to upgrade to a real-time PGIS is possible.

The model in its current form displays the image of the buildings and the surrounding parking lots. However, a connection can be established between a GIS server of the campus and the web application to retrieve and display the real-time GIS maps of the campus as and when needed. These GIS maps will be capable of showing the shortest walking routes from the parking lots to the buildings along the sidewalks.

It should be mentioned that the application is currently developed considering visitors as the main users of the system. However, it can also be used by faculty and students at the campus to assist in making better parking decisions. The parking application, in its current format, can give an estimate of the number of empty spaces based on the previous year's data. No randomness is introduced into the system. However, to make the application more probabilistic in nature, as part of the future research randomness may be introduced into the system so that the application can estimate the number of empty spaces in a parking lot based on a stochastic model.

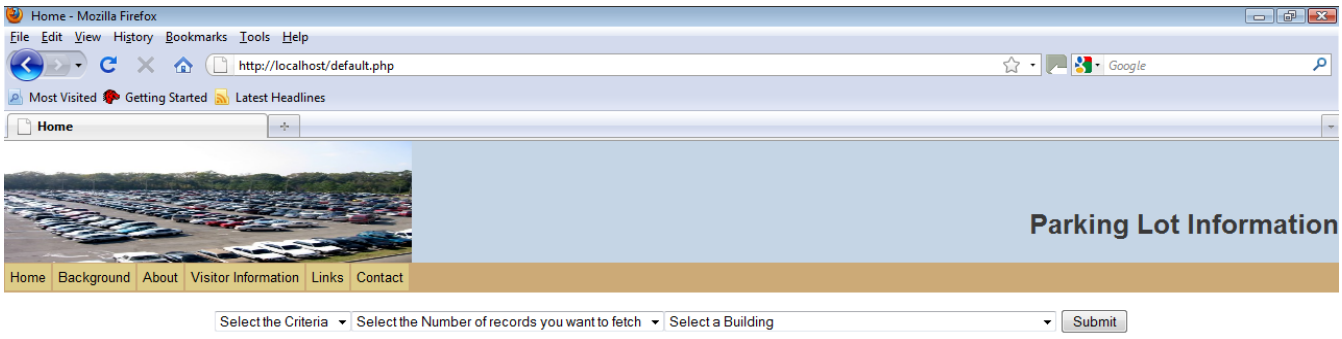
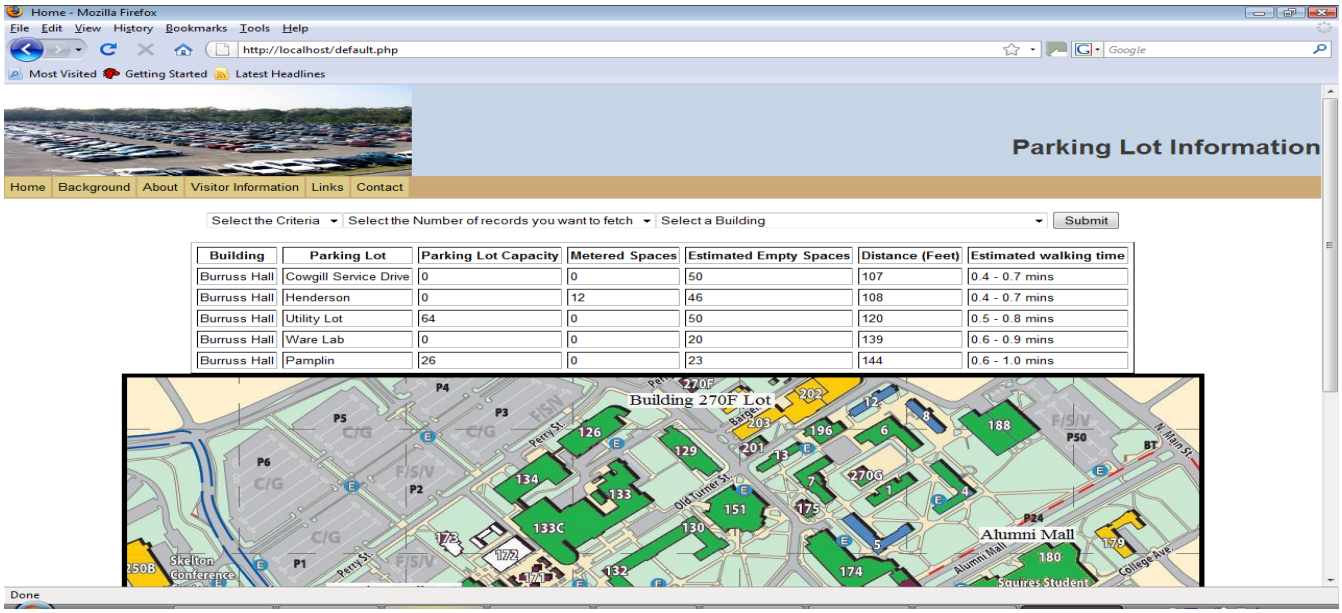


Fig. 3. Screen Shot Depicting the HTML Form Used in the Application



Building	Parking Lot	Parking Lot Capacity	Metered Spaces	Estimated Empty Spaces	Distance (Feet)	Estimated walking time
Burruss Hall	Cowgill Service Drive	0	0	50	107	0.4 - 0.7 mins
Burruss Hall	Henderson	0	12	46	108	0.4 - 0.7 mins
Burruss Hall	Utility Lot	64	0	50	120	0.5 - 0.8 mins
Burruss Hall	Ware Lab	0	0	20	139	0.6 - 0.9 mins
Burruss Hall	Pamplin	26	0	23	144	0.6 - 1.0 mins

Fig. 4. Parking Lot Information with Distance as Criteria

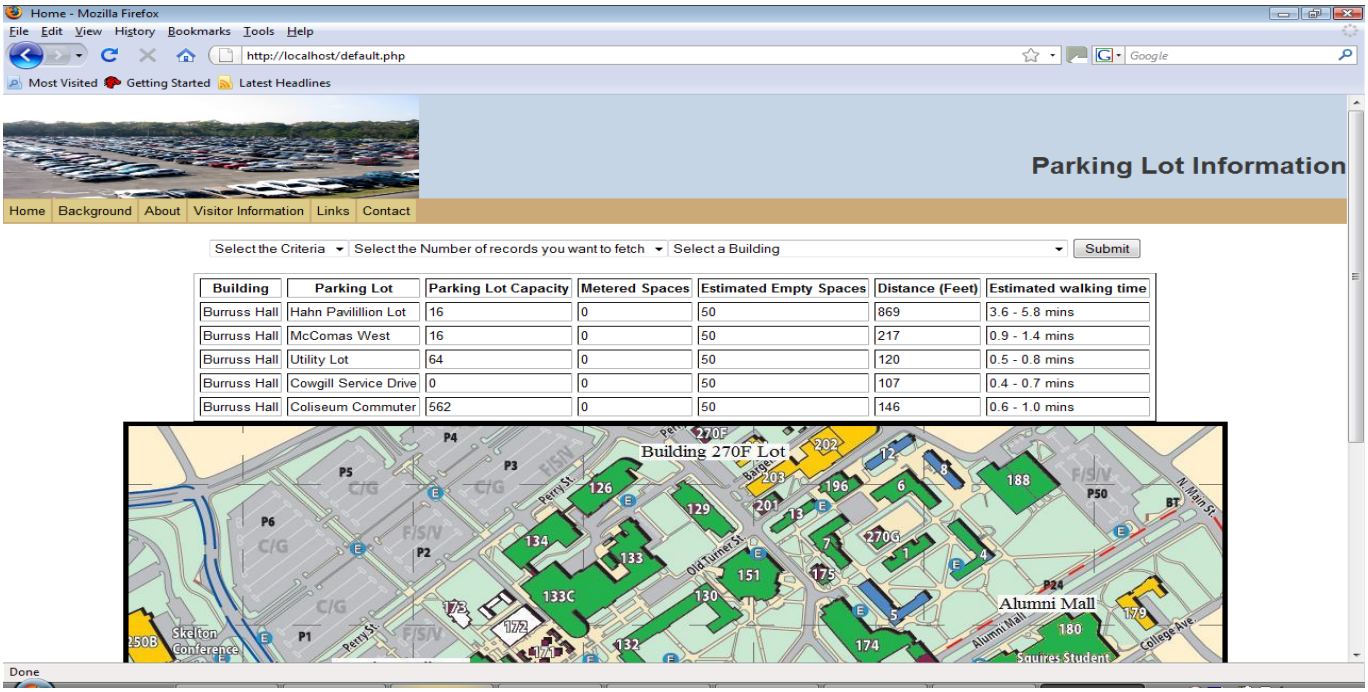


Fig. 5. Parking Lot Information with Empty Spaces as Criteria

X. ACKNOWLEDGEMENTS

The authors are grateful to the input and feedback received from Steven Mouras, Richard McCoy, Mike Dunn and Jason Shelton. Finally, the authors also acknowledge the financial support provided by the Virginia Tech Transportation Office in conducting this research effort.

REFERENCES

[1] The DPS Parking Division Office, I.S.U. (2003) The Iowa State University Parking Division Manual.

[2] Todd Litman, "Parking management strategies, evaluation and Planning," 2008.

[3] Qiang Liu, Huapu Lu, Bo Zou and Qiang Li "Design and development of parking guidance information system based on web and gis technology," IEEE ITS Telecommunications proceedings, pp. 1263-1266, 2006.

[4] Akihito Sakai, Takumi Sugimoto, Kozi Mizuno and Takeshi Okuda "Parking guidance and information systems," IEEE Vehicle Navigation & Information Systems, pp. 478-485, 1995

[5] T. S. Chou et al. "CaNPAs: a campus navigation and parking assistant system," IEEE Int. Conference on Systems, Man and Cybernetics, pp. 631-638, October 8, 2006.

[6] Russell G. Thompson, "Drivers' response to parking guidance and information systems," Transport Reviews, 1997, Vol.17, No.2, pp.89-104.