Evaluation and Development of a University Visitor Parking Management Framework

Sashikanth Gurram

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Civil Engineering

Dr. Hesham Rakha, Chair
Dr. Antoine Hobeika
Dr. Montasir Abbas

November 16, 2009
Blacksburg, Virginia

Keywords: Visitor center survey, Follow-up survey, Parking lot survey, Web based interactive system, Database management

`

Virginia Tech Visitor Parking Management

Sashikanth Gurram

ABSTRACT

The main campus of Virginia Polytechnic Institute and State University (Virginia Tech) has a current parking inventory of approximately 14,310 parking spaces in 88 surface lots. Of the available parking spaces 92% are designated for the use of Virginia Tech students, faculty and staff members. The share of visitor usage of these parking spaces is approximately 2%. The objective of this thesis is to evaluate the current visitor parking system and develop techniques to enhance the parking facility operations. The current visitor parking demand is determined by conducting a field evaluation of the visitor parking in five parking lots. Personal and follow-up interviews were conducted with university visitors to determine their satisfaction levels with the existing parking facilities. While the survey results demonstrated that 52% of visitors rated the parking services to be good or very good, the study also showed that approximately 28% of the visitors did not know the location of the most convenient parking lot to access their destination and almost 6% were not satisfied with the parking services offered by Virginia Tech. Apart from this, about 32% of the visitors had to visit at least 2 parking lots before finding a parking space. These figures clearly demonstrate a need for a better management of the parking lots to serve the visitors in an efficient way. In an attempt to enhance the visitor parking system a web-based interactive framework is proposed. This framework identifies the best possible parking lot(s) for a selected destination considering various factors like - distance of the parking lot to the destination and typical occupancy of the parking lot at specific times of the day and other relevant factors. The web-based interactive system is designed to provide the visitor with two or more choices for the parking in order to arrive at their desired destination.

`

ACKNOWLEDGEMENTS

I sincerely acknowledge the valuable guidance and support that I received from my advisor, Dr. Hesham Rakha throughout this research work and my graduate study. He is an exceptionally kind and knowledgeable person. I respect and adore him unequivocally. I am grateful to him for funding me and keeping me financially stable throughout my study at Virginia Tech. I would also like to thank my committee members, Dr. Montasir Abbas and Dr. Antoine Hobeika, for their inputs and timely guidance.

I would like to acknowledge Virginia Tech Transportation Office for providing support and funding for this research. I gratefully acknowledge Alejandra Medina for her advice and help during the phase III of the project.

I would also like to acknowledge the work conducted by the graduate students at the Center for Sustainable Mobility, who helped in data collection efforts. My special thanks go to all those unknown PHP and MySQL Gurus out there on the internet who helped me a lot during the development of the web based system. I would like to specially mention Aly Tawfik, one of my fellow researchers and a great friend, for his suggestions and constant guidance throughout my graduate studies. I would also like to thank my friend Yatish Kasaraneni who helped me in designing the website for the interactive system.

I am grateful for my father who is my first inspiration, my mother for her support and my sister for her encouragement.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 PROBLEM OVERVIEW

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 2002 Survey of the 55 parking lots (95% of the supply) showed an average occupancy rate of 85%. Twenty six of these parking lots (76% of the supply) exceeded their effective capacity of 90%. These results demonstrated a need to characterize both the visitor parking demand and supply of the Virginia Tech facilities so as to serve the visitors more efficiently. There was a lack of information about the level of satisfaction of the visitors with the existing parking facilities. It would be impossible to make any improvements to the existing system or the existing infrastructure without any prior knowledge about the above mentioned factors.

According to the Parking Management study - Phase III, Virginia Tech receives an average of 100 parking pass requests per day during the weekdays. 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. Because of the large expanse of the Virginia Tech campus, visitors have a very limited knowledge about the geographic location of the university parking facilities and services. The high occupancy rate poses a challenge in finding a parking space in the parking lot near the visitor’s destination.

The visitor survey conducted as part of the parking management study - Phase III showed that about 32% of the visitors did not know the location of the most convenient parking lot to access their destination and almost 8% were not satisfied with the parking services offered by Virginia Tech. These figures clearly demonstrate a need for a better management of the parking lots to serve the visitors in an efficient way.

1.2 THESIS OBJECTIVES

There are two major objectives to this thesis. The first objective is to determine - the current visitor parking demand and supply, the adequacy of the visitor parking at Virginia tech, the level of satisfaction with the visitor parking, and the willingness of visitors to pay for better parking facilities. The second objective is to - develop a web based interactive system, which uses historical occupancy data and distances from the buildings on the campus to the parking lots, and provides the visitor with the most convenient parking lot choices.

1.3 RESEARCH APPROACH

In addressing the first objective two visitor interviews were conducted. The first interview was conducted at the VT Visitor Center. Visitors were surveyed as they stopped to get their parking permits. At the end of this interview the visitors were asked if they could be contacted by e-mail, phone or cell phone in order to conduct a follow-up interview. A web-based survey was created and the link to the survey was sent in an e-mail to the visitors who agreed to participate in the follow-up survey. The e-mail requested the visitors to participate in the survey, explaining the benefits that will result due to their participation in the survey. A contact address was also provided in the e-mail in case the participants had some concerns or doubts regarding the survey.

A field survey was conducted to characterize the use of five parking lots. The five parking lots that were selected were Cassell Coliseum, Drill Field, Litton Reaves, Squires and Perry Street II. In order to characterize visitor and violator patterns, data was gathered on Thursday and Friday, as Thursdays and Fridays were the two days which represent a typical weekday and the day-of-the-week with the highest demand respectively. The license plate information of a car displaying a visitor parking permit or with no parking permit was recorded every half hour. The survey was conducted for two days in each parking lot.

Parking Management refers to policies and programs that result in more efficient use of parking resources [2]. There are several parking management principles which help in the planning to support parking management. Some of the widely used parking management principles are consumer choice, user information, sharing, efficient utilization etc. In this thesis, the user information principle of parking management is used to manage the parking services for the visitors. 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 the campus. Thus, user information principle comes into play through the means of the web based interactive system.

PHP, which stands for *PHP: Hypertext Preprocessor,* is used to code the logic, which will give the output in the form of a list of the most convenient parking lots based on the given input. The variables used to execute the code are historical parking lot occupancy data, parking lots in the vicinity of the destination and distance of each parking lot to the destination. These variables are stored in a database objects called tables in MySQL. A query is generated using the PHP code which performs a search through the stored variables in MySQL database and returns an output.

CHAPTER TWO: VIRGINIA TECH PARKING MANAGEMENT STUDY

2.1 Introduction

Virginia Polytechnic Institute and State University (Virginia Tech), has evolved into a comprehensive university of national and international prominence with more than 29,000 full-time students. The mission of Virginia Tech as a land-grant university is to serve the Commonwealth of Virginia, the nation and the world through the discovery and dissemination of new knowledge. The institution's focus on teaching, learning, research and outreach are enhanced by effective transportation and parking at its main campus in Blacksburg, Virginia. Convenient parking, efficient traffic flow and effective alternative transportation systems including transit services are crucial elements in meeting the University's mission as well as maintaining the continued growth of the institution (Virginia Tech Transportation Master Plan). The main campus of Virginia Tech has a current parking inventory of approximately 14,310 parking spaces in 88 surface lots.

The first phase (Phase I) of the Virginia Tech Parking Study conducted a detailed investigation of the various parking surveillance and enforcement systems that are commercially available. The second phase of this project expanded on the Phase I study by conducting a field evaluation of the in-field technologies, characterized the service rates associated with the various technologies, and conducted a modeling study of the queuing associated with each of the technologies on three prototype parking lots.

The third Phase of this study expands on the Phase I and II studies by focusing on visitor parking, a very important component of the University Business. Phase III primarily determines the current visitor parking demand and supply by conducting a field evaluation of the visitor parking in five parking lots, characterizes the occupancy for each type of visitor, and also conducts personal interviews and follow-up interviews with university visitors.

2.2 Background

In the year 2002, the main campus of Virginia Tech had a parking inventory of approximately 14,310 parking spaces (including motorcycle spaces) in 88 surface lots as shown in Figure 1 and is also denoted in the Virginia Tech Parking and Transportation Master Plan.

These parking lots are mainly classified into three categories (a) faculty, staff, and visitor; (b) commuter and graduate; and (c) resident. Apart from these, the parking lots have been further classified into motor cycle, general, service & loading, handicap and metered. In most of the cases this classification is in effect for the five work week days from 7:30 AM to 5:00 PM, with the exception of a few faculty/staff lots that retain their classification for 24 hours, seven days a week. For instance, after 5:00 PM and on weekends, the parking lots for commuter/graduate are open to the general public. The parking permit required for each parking area is designated by signage. Any student, staff, or faculty affiliated with the university must purchase a parking permit to be able to park on campus. Virginia Tech does not restrict the number of parking permits sold to any user group. Vehicle registration is valid until the registrant is no longer affiliated with the university as a student, faculty, or staff member, or until the permit expires. Visitor parking in Virginia Tech is free of cost. Visitors can park in any faculty, staff, student or visitor parking space if they have a valid visitor parking permit and if the space is not restricted by signs. Some restricted areas include handicap spaces, carpool spaces, loading zones, service vehicle zones, timed areas, sidewalks, grass, admission areas, teaching assistant spaces, Owens and Dietrick dining hall lots [3]. Visitor parking permits are available at the visitor information center on the Southgate Drive (8:00 a.m. – 5:00 p.m., Monday through Friday) or at the University Police Department in the Maintenance Complex off Southgate Drive at all times.

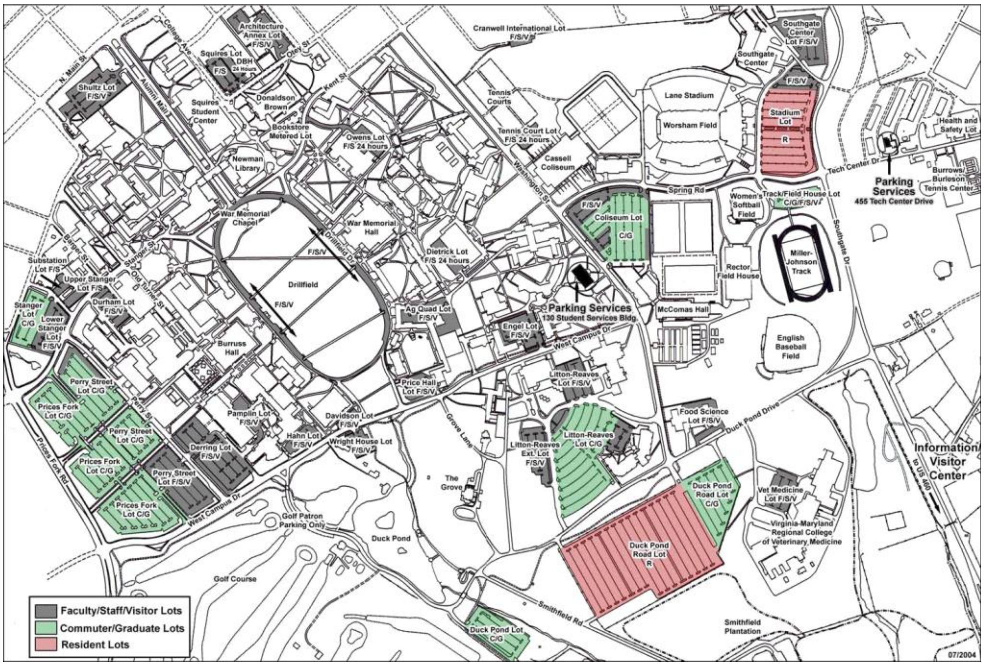


Figure 1: Virginia Tech Parking Lots

Of the 14,310 parking spaces, approximately 92% are designated for the use of Virginia Tech students, faculty, and staff members and follows a distribution as shown in Figure 2.

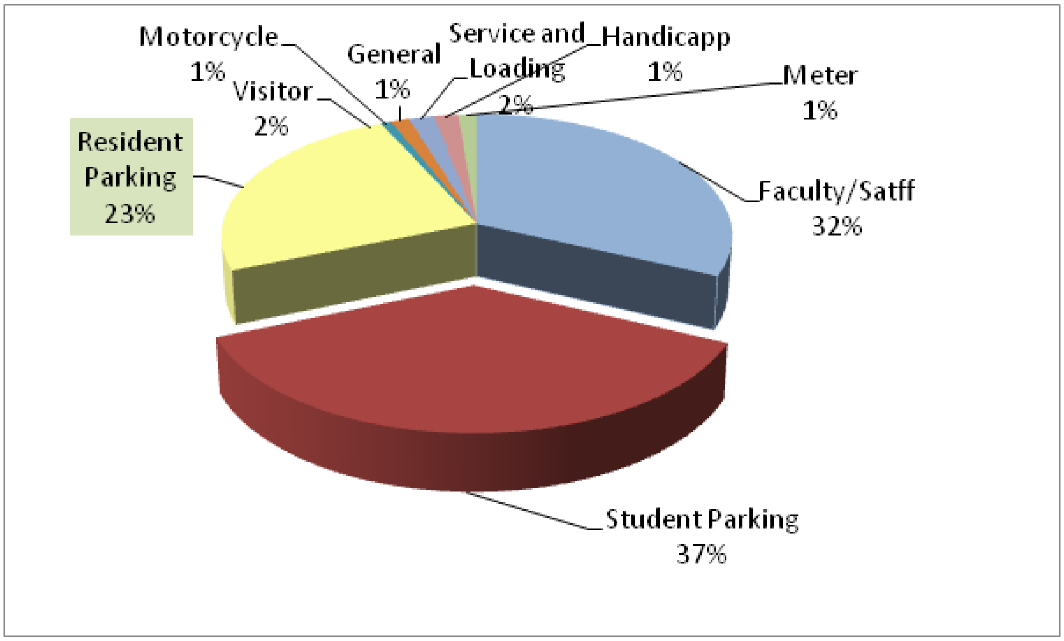


Figure 2: User Parking Permit Assignment Distribution at Virginia Tech

An individual may register more than one vehicle since the hangtag style permit can be moved from one vehicle to another. Individuals having two vehicles parked on campus at the same time must have each vehicle registered apart from displaying a permit. However, a vehicle can be registered to only one person and only one permit type per vehicle is allowed. The permit must be displayed so that it is readable through the windshield by parking enforcement officers.

A 2002 Survey of the 55 parking lots (95% of the supply) showed an average occupancy rate of 85%. Twenty six of these parking lots (76% of the supply) exceeded their effective capacity of 90%. However, parking lot occupancy level is not necessarily synonymous with the parking demand. Parking occupancy is used as an indicator of utilization of the existing parking supply. Parking demand on the other hand, provides an indication of the number of users from different categories willing to park at a given location and time when supply is adequate. This study attempts to characterize both the visitor parking demand and supply of the Virginia Tech facilities.

Most of the university parking lots do not have any type of access control. Virginia Tech Parking Services has parking control officers who issue tickets when a parked vehicle does not have the correct parking permit for the occupied lot. Despite the presence of numerous parking patrol officers, Virginia Tech has a high violation rate.

Virginia Tech does not regularly collect data on:

- Visitor parking occupancy.
- Occupancy by type of visitor.
- Stay duration for different visitor categories.
- Percentage of parking violators.

This information is critical in order to improve the quality of service, maximize the use of facilities and determine parking trends that would further identify parking space deficits for specific users or identify parking supply adjacent to specific buildings.

With a high parking lot occupancy rate on campus, finding a parking space has really become a tedious job. At Virginia Tech a visitor with a parking permit can park the vehicle on-campus. Visitors usually belong to one of the following categories:

1. A prospective student or a family member of a prospective or current student, or
2. A person having an appointment with one of the University branches.

Visitor parking permits are available at the Visitor Information Center, Parking Services, or at the University Police Department. Visitors may park in any faculty, staff, visitor or student parking space if they have a valid visitor parking permit and if the space is not restricted by signs**.** Some restricted areas include handicap spaces, carpool spaces, loading zones, service vehicle zones, timed areas, sidewalks, grass, admission areas, teaching assistant spaces, Owens and the Dietrich Dining Hall Lots. Visitors are recommended to use the Faculty/Staff section of Perry Street, Cassell Coliseum, Washington Street, Litton-Reaves Extension, and the Squires lots as shown in Figure 3. Visitors may also park around the Drill Field which is surrounded by several major university buildings, including the Undergraduate Admissions and the University Administration offices.

2.3 Virginia Tech Parking Study Overview

2.3.1 Study Objectives

The specific objectives of this study are to:

1. Determine the current visitor parking occupancy of selected parking lots at VT.
2. Determine the current violator parking occupancy of selected parking lots at VT.
3. Determine the current visitor parking demand and adequacy of VT parking lots.
4. Determine the level of satisfaction from visitors with regards to parking at VT.
5. Evaluate the possibility of charging visitors for parking.

2.3.2 Study Tasks

The main tasks accomplished by the project are:

1. A thorough field survey of the operations of the five parking lots.
2. Visitor personal interviews.
3. Follow-up interviews.
4. Analysis of historical visitor parking records.

2.4 Interviews

In order to realize the objectives of the research two visitor interviews were conducted. The first interview was conducted at the VT Visitor Center. Visitors were surveyed as they stopped to get their parking permits. At the end of this interview the visitors were asked if they could be contacted by e-mail, phone or cell phone in order to conduct a follow-up interview.

2.4.1 Visitor Interviews

Visitors were surveyed as they stopped to get their parking permits. The purpose of this interview was to know the reason for their visit, buildings planned to visit on campus, expected duration of stay, and their likely place of parking. To protect the identity of the visitor, the contact information was stored independently of the survey. The Center for Sustainable Mobility at VTTI has worked with the Virginia Tech Office of Transportation to ensure that the survey instrument met the research objectives specified for the project. The survey instrument was also pretested by VTTI to establish an optimal cooperation rate and also to ensure that the survey length is within the expected time. The interview only required a couple of minutes to be completed. The visitors were also asked if they could be contacted by e-mail, phone, or cell phone in order to conduct a follow-up interview.

The survey pre-test revealed the need for the clarification in the description (wording) of some survey items. After the approval of the Transportation Office to the wording changes on the survey, VTTI conducted the survey.

Permission to conduct the survey was obtained from the Virginia Tech Institution Review Board before the commencement of the survey, which is a prerequisite for any project involving human subjects.

To better serve the visitors in the future and to increase the number of respondents, it is recommended that the visitor center request visitors to fill a short form that could be completed when a visitor pass is requested, so that more follow-up interviews can be conducted. A copy of the visitor center survey form is provided in Appendix A: Visitor Personal Survey of the thesis.

2.4.2 Results

Summary of the Visitor center interview results		
QUESTION	RESPONSE	PERCENTAGE
Reason for Visit	Prospective student or family member	25%
	Current family member or friend of student	26%
	Academic activities	10%
	Local resident	3%
	Football Game	4%
	Other	24%
	Returning Student	1%
	No response	7%
Scheduled Meetings	Meeting Scheduled	20%
	No Meeting Scheduled	9%
	No Response	71%
Do you know theLocation of the offices you need to visit	Yes	53%
	No	20%
	No Response	27%
Do you know the most convenient parking lot to access your destination	Yes	61%
	No	28%
	No Response	11%
How long are you planning to stay at the location of your destination	Less than 1 Hr	11%
	1 to 2 Hrs	20%
	2 to 3 Hrs	20%
	4 Hrs	10%
	More than 4 Hrs	30%
	No response	9%
How do you rate your previous parking experiences	Very Good	17%
	Good	35%
	Fair	14%
	Poor	6%
	No response	28%
Are you willing to pay for designated visitor only parking spaces	Yes	33%
	No	55%
	No Response	12%

Table 1: Summary of the visitor center survey results

Almost 20% of the respondents were visiting to attend a scheduled meeting and 9% did not have any scheduled meetings whereas 71% of the people did not respond to this question. The most common buildings that the visitors visited included Burruss Hall, Randolph Hall, the Squires Student Center, the Newman Library, and Cassell Stadium. More than half of the visitors (53%) knew the location of the offices they were visiting while 20% of the visitors did not know the location of the offices they were visiting and 27% did not respond to the question. These findings demonstrate that a relatively large percentage of visitors could benefit from specific directions to their destination (this includes people who were not aware of the location of the building they were visiting or did not respond to the question). Surprisingly, 61% of the visitors indicated that they knew the most convenient parking lot to access their destination and 28% responded saying that they did not know the most convenient parking lot to their destination.

Only 11% of the respondents had planned to stay at their destination for less than 1 hour. 20% planned to stay between 1 to 2 hours and the same percentage expected to stay for 2 to 3 hours. 30% of the respondents expected to park for more than 4 hours. 22% of the respondents had visited the campus only once. Almost the same percentage (21%) of the respondents were yearly visitors to the campus visiting at various frequencies per year. 14% of the respondents visited the campus at various frequencies during the semester. 16% of the respondents were monthly visitors to the campus and 9% were weekly visitors. 5% of the respondents visited the campus at some other frequency not given in the questionnaire. 10% of the respondents did not respond to the question.

When asked how they rated the previous experience with the parking services 67% percent of the people that responded rated the services as very good, good or fair. Only 6% rated the service as poor. Most of the respondents who rated the services as poor were issued a ticket or were not able to find a parking stall. However, 28% of the people have not at all responded to this question. A majority of the respondents, about 55%, were not willing to pay for designated visitor-only parking spaces.33% responded positively saying that they would be willing to pay for designated visitor-only parking spaces.

Of the respondents who indicated they would be willing to pay for a dedicated visitor parking spot, 50% were willing to pay only $3 and almost 40% were willing to pay $5. The respondents willing to pay $7 or more constituted only 11% of the respondents.

2.4.3 Follow-up Interviews

Visitors who completed the initial survey were also asked if they could be contacted via e-mail, phone, or cell phone in order to conduct a follow-up interview. The purpose of this interview was to characterize the time taken to find a parking spot, the destination location on campus, the duration of time that the vehicle was parked, and the parking experience of the visitor. The visitors were further asked if after their parking experience they were willing to pay to ensure a guaranteed parking space and the opportunity cost that they were willing to pay for this parking stall. The Center for Sustainable Mobility (VTTI) worked with the Transportation office to ensure that the survey instrument met the research objectives specified for the project. VTTI also pre-tested the survey instrument in order to ensure an optimal cooperation rate for the survey and to ensure that the survey length did not exceed the expected time. The Survey only required a couple of minutes to be completed. After approval of the required verbal changes to the survey instrument by the Transportation Office, VTTI conducted the survey. Of the initial survey respondents, almost 30% agreed to participate in a follow-up interview. As was expected, some visitors did not provide adequate contact information which reduced the number of follow-up respondents to 25% (100 people). 44% of those contacted responded to the follow-up survey. The respondents were asked for their preferred choices of the follow-up interview i.e., via phone or e-mail and most of them selected the e-mail option. A personal e-mail was sent to all the participants who agreed to participate in the follow-up interview and reminder e-mails were sent later. A copy of the final survey is presented in Appendix B: Visitor Follow up Survey (Web-based or by mail) of the thesis. Permission to conduct the survey was obtained from the Virginia Tech Institution Review Board before beginning it.

2.4.4 Results

Summary of the Follow-Up survey results		
QUESTION	RESPONSE	PERCENTAGE
Reason for Visit	Prospective student or family member	25%
	current student family member or friend	26%
	Academic activities	12%
	Local resident using VT services	7%
	Other	30%
How long did it take you to find a parking space	Between 1 and 2 min	39%
	Between 2 and 5 min	41%
	Between 5 and 10 min	13%
	More than 10 min	7%
How many parking lots did you visit until you found a parking space	1 Lot	66%
	2 Lots	21%
	3 Lots	9%
	More than 3 lots	2%
	No response	2%
How long did you park	Less than 1 hour	14%
	1 hour to 2 hours	39%
	2 hours to 3 hours	29%
	4 hours	9%
	More than 4 hours	9%
How long were you planning to park	Less than 1 hour	14%
	1 hour to 2 hours	39%
	2 hours to 3 hours	25%
	4 hours	9%
	More than 4 hours	11%
	No response	2%
Would you say that your parking experience at Virginia Tech was…	Very Good	41%
	Good	41%
	Fair	9%
	Poor	9%
Are you willing to pay for designated parking spaces	YES	34%
	NO	66%

Table 2: Summary of the follow-up survey results

Of the respondents who answered the follow-up survey 51% were visiting either as a prospective student, a family member of a prospective student, a family member of an existing student, or a friend of an existing student. Visitors for various other reasons constituted approximately 24% of the total respondents. The sample categorization was very similar to the initial survey respondent breakdown.

The visitors visited multiple buildings, however, Burruss Hall, Cassell Stadium, the Squires Student Center, the Newman Library, Derring Hall, VMRC, and the Military Building constituted the most popular buildings visited. 80% of the respondents required less than 5 minutes to find a parking space and 39% of the respondents spent 1 to 2 min to find a parking space. Approximately, 7% of the respondents required more than 10 minutes to find a parking space. 66% percent of the people visited only 1 parking lot before finding a parking space and no one walked for more than 10 minutes to reach their destination. 71% of the respondents required less than five minutes to walk from their vehicle to their final destination point.

68% of the respondents parked between 1 to 3 hours and only 9% parked for more than four hours (compared with 30% in the initial survey). This could be explained by the fact that the participants provided, estimated stay duration in the initial survey. When asked about the time for which they were planning to park, 39% of the respondents indicated that they planned to park for 1 to 2 hours. 25% of the respondents planned to stay for 2 to 3 hours. 14% of the participants planned to stay for less than 1 hour. Almost 11% of the respondents indicated that they were planning to park for 4 hours or more. Only 23% of the respondents parked in more than one parking lot. After visiting the campus 82% of the respondents rated the parking services at VT to be very good or good. This percentage is distributed evenly between very good and good and only 34% of the respondents were willing to pay for parking spaces. This can be seen as an indication that the parking supply is good and visitors do not see any reason to pay for designated parking spaces. Of the people who were willing to pay, 46% opted to pay $3 only and 40% were willing to pay $5.

27% of the respondents had visited the campus only once while 34% of the respondents were yearly visitors and visit the campus at various frequencies during the year. 16% of the respondents visit the campus in a semester at varying frequencies. 14% of the respondents were monthly visitors while 11% of the respondents were weekly visitors to the campus and visited the campus at varying frequencies.

2.5 Field Survey

Field visitor parking data was gathered at five parking lots that were selected in agreement with the VT Transportation Office. The five parking lots that were selected are Cassell Coliseum, Drill Field, Litton Reaves, Squires and Perry Street II. Of the five Parking lots only the Drill Field had visitor dedicated spaces at the time of the study.

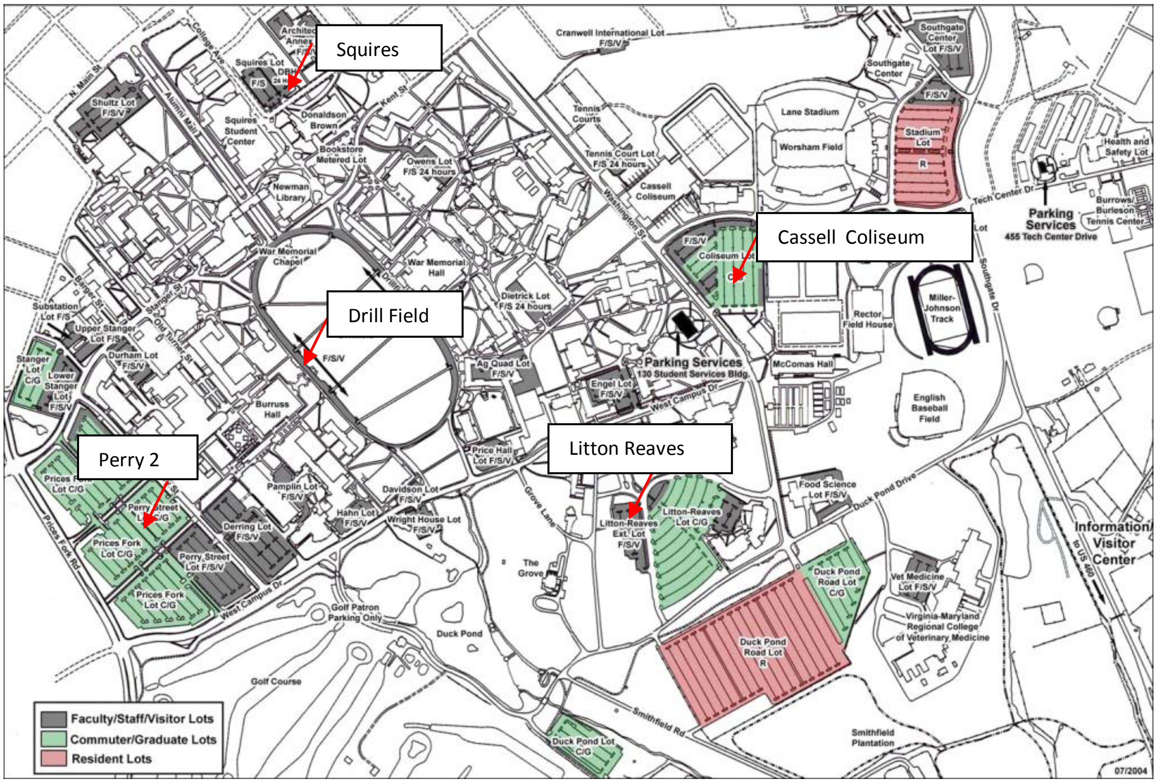


Figure 3: Surveyed Virginia Tech parking facilities

In order to characterize visitor and violator patterns, data was gathered on Thursday and Friday as shown in Table 3. The reason for selecting Thursdays and Fridays was because according to the historical data these two days represent a typical weekday and the day-of-the-week with the highest demand. As the visitor permits must be displayed from 8:00 AM - 5:00 PM Monday through Friday, the visitor parking data collection effort began at 8:00 AM except for the Drill

Field. The data for the Drill Field were collected from 6:00 AM to 8:00 AM and from 5:00 PM to 7:00 PM.

Cassell Coliseum	Drill Field South	Litton-Reeves	Perry Street 2	Squires
10/11 Thursday 10/12 Friday	10/18 Thursday 10/19 Friday	11/1 Thursday 11/2 Friday	10/18 Thursday 10/19 Friday	11/1 Thursday 11/2 Friday

Table 3: Visitor parking data collection schedule

2.5.1 Methodology

The University Transportation office provided the researchers with AUTOCAD maps of the parking lots. These maps were used as a base for the data collection process. The maps were used to input a space-specific row letter and a number as shown in Figure 4.

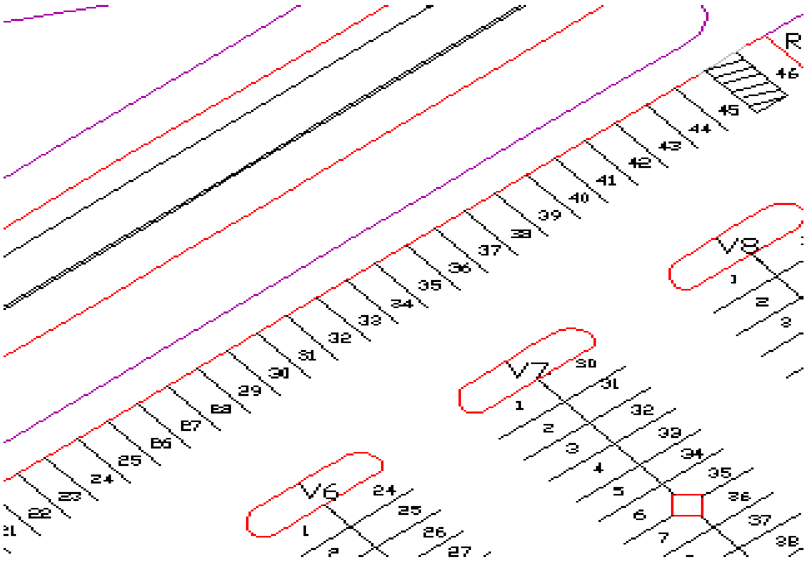


Figure 4: Parking spaces

This step was done to ensure a complete and orderly process of data collection. Depending on the size of the parking lot and the estimated number of visitors, 2 to 5 persons were assigned to each parking lot to collect the data. The entire parking lot was surveyed every half an hour. It was expected that in this way every visitor could be accounted for. Owing to the fact that nobody could park in the University parking lot without a permit, the data collector approached each vehicle and identified the type of permit and the following information:

- Time
- Parking Space number
- Visitor type
- The first and the last digit of the license plate number
- Parking violator type
- Occupancy
- Duration of stay

Type of Visitors

The type of visitor was defined by the type of parking pass issued by the University

- *Visitor (V)*

This permit is issued by the University for Short Term parking. Visitors (non-students or non-faculty/staff) may park in any faculty, staff, visitor or student parking space if they have a valid visitor parking permit and if the space is not restricted by signs. Some restricted areas include handicap spaces, carpool spaces, loading zones, service vehicle zones, timed areas, sidewalks, grass, admission areas and teaching assistant spaces.

- *Long Term Visitor (LV)*

This permit is available for long-term visitors and special event visitors who visit the campus frequently. Students and faculty/staff are not eligible for courtesy permits.

- *Retiree (R)*

This permit is issued by Parking Services for retirees of Virginia Tech.

- *CRC. Corporate Research Center (CRC)*

This permit is issued by Parking Services for CRC employees of private companies.

- *Contractor (C)*

Parking Services requires the purchase and display of a parking permit for all vendor/business/contractor vehicles that are owned privately and by companies that park on campus.

- *Temporary(T)*

This permit is available for faculty, staff and students who need special parking arrangements. This permit includes the permit types of new employee, departmental loading and temporary medical disability.

Parking Violator Types

Two types of parking Violators were considered:

- *Parking Violators Type 1. (V1).*

The vehicle has a parking permit, but was parked in an unauthorized parking space for that particular type of permit. For example, resident, commuter, and graduate student permit parking is prohibited on campus streets and in faculty/staff parking areas (except where signs designate otherwise) from 7:00 a.m. to 5:00 p.m., Monday through Friday.

- *Parking Violators Type 2. (V2).*

The vehicle does not have a parking permit at all.

Occupancy rate

Occupancy rate[4] is defined as the number of the vehicles in the parking lot, divided by the lot capacity. When analyzing the occupancy data, it is important to take into account the "effective supply" factor. This factor essentially is a cushion of spaces in excess of the calculated demand to reduce the search time for the last available parking stalls, and to account for the temporary loss of spaces due to improperly parked vehicles. Typically, for permit parking areas, the effective supply factor recommended is 90%. However this factor is reduced to 85% in the case of visitor parking areas, because it assumes that visitors are less familiar with the area. For this study the 85% effective supply factor was used.

Duration of stay

The duration of stay is defined as the time spent in the parking lot by a specific vehicle. Only partial information from the license plate was collected to protect the privacy of the visitors. In this way the researchers were able to identify how long each vehicle was parked. The number of visitors at any given time is defined as the total number of parking spaces that are occupied by visitor cars at any given time.

2.5.2 Results

Parking Lots			Cassell Coliseum	Drill Field	Litton-Reaves	Perry-II	Squires
Visitors	Range	Thursday	27-41	1-17	0-5	9-13	3-7
		Friday	27-46	17-33	0-1	4-13	0-16
	Average	Thursday	34	9	3	12	6
		Friday	39	23	1	9	7
	Total	Thursday	49	30	5	16	12
		Friday	57	77	1	19	18
Violators	Range	Thursday	7-14	2-29	2-3	1-3	2-7
		Friday	7-11	2-28	1-2	0-2	0-5
	Average	Thursday	11	11	2	2	6
		Friday	8	14	1	2	2
	Total	Thursday	23	60	3	3	9
		Friday	22	54	2	3	7
Maximum occupancy reached during			9:30 AM-10:30 AM	5:30 PM-6:30 PM	10:30 AM-11:30 AM	11:00 AM-11:30 AM	11:00 AM-11:30 AM

Table 4: Summary of the visitor counts in different parking lots

2.5.3 Cassell Coliseum Parking Lot

The Cassell Coliseum parking lot had a total of 806 parking spaces at the time of the study. These parking spaces included 211 parking spaces for F/S and 11 parking spaces for handicapped users and no dedicated spaces for visitors. The total number of visitors at a given time ranged from 27 to 46. On average 36 visitor vehicles were typically present in the parking lot at any given time. The total number of visitors was 49 for Thursday and 57 for Friday. The number visitors reached a peak between 9:30 AM and 10:30 AM and this peak was maintained after that. However the different visitor categories followed different trends. The short term visitors accounted for almost 75% of all the visitors on both the days. After the short term visitors the temporary visitors accounted for 13% of the total visitors on average. The other visitor types including CRC, Retiree and Long term visitors constituted 10% of the total visitors on both days. Most of the visitors stayed for more than 4 hours followed by visitors that stayed between half an hour and 1 hour. The average duration of stay was approximately 2 hrs and 45 minutes on both days. Similar patterns of frequency were repeated on both days.

The total number of violators at a given time ranged from 7 to 14 violators. On average 9 violator vehicles were typically present in the parking lot at any given time on both days. The total number of violators was almost equal for both days standing at 23 and 22 on Thursday and Friday, respectively. The violators without permits accounted for most of the violators on both days ranging from 68% to 83%. Most of the violators stayed between half an hour to 1 hour in the parking lot. The effective capacity of the parking lot was reached between 9:30 AM and 10:30 AM. Thursday in particular showed higher occupancy rates with values approaching 100%.

2.5.4 Drill Field Parking Lot

The data for this parking lot were collected on Thursday and Friday from 6:00 AM to 8:00 AM and from 5:00 PM to 7:00 PM.

The total number of visitors at a given time ranged from 1 to 17 on Thursday and 17 to 33 on Friday. On average 8 and 23 visitors were present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 30 on Thursday and 77 on Friday. The visitor volumes were higher during the evening hours in comparison to the morning hours. The number of visitors remained almost constant during the morning hours. During the evening hours there was a slight decrease in the number of visitors with time on Friday whereas a slight increase in the visitor number was observed on Thursday. However the different visitor categories followed different trends. Long term visitors accounted for approximately 47% of the total visitors on Thursday while short term visitors accounted for 27% and the remaining 26%

was shared by CRC and retiree visitors. On Friday, the short term visitors made up for most of the visitors by amounting to 47% of the total visitors. The retirees and long term visitors accounted for 25% and 21% of the total number of visitors, respectively. On Thursday most of the visitors stayed for 1 hour, followed by visitors that stayed for half an hour.

The total number of violators at a given time ranged from 2 to 29 on both days. On average 12 violator vehicles were typically present in the parking lot at any given time on both days. The total number of violators was almost equal for both days standing at 60 and 54 on Thursday and Friday, respectively. In the morning hours, the number of violators reached a peak at 7:00 AM on Thursday (with about 30 violators) whereas the number of violators reduced with time on Friday. In the evening hours the number of violators increased with time on both days and Friday had a higher rate of violation when compared to Thursday. The violators with a permit accounted for 57% of the violators on Thursday. On Friday almost all the violations did not have a permit. Most of the violators stayed for 30 minutes, followed by the violators that stayed for more than 1 hour. The average duration of stay was approximately 45 minutes on Thursday and 1 hour on Friday.

In the morning hours the parking lot operated well below the effective capacity. Even during the evening hours the parking lot operated below the effective capacity but the occupancy levels were high compared to that of the morning hours.

2.5.5 Litton-Reaves Parking Lot

The Litton Reaves parking lot is situated near the Duck pond drive and Washington Street. It has a total of 180 parking spaces out of which 90 are reserved for F/S and 90 spaces are reserved for COM.

The total number of visitors at a given time ranged from 0 to 5 on Thursday and 0 to 1 on Friday. On average 3 and 1 visitors were present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 5 on Thursday and 1 on Friday. The short term visitors accounted for about 80% of all the visitors on Thursday and the remaining 20% is accounted for by the retirees. On Friday all the visitors were long term visitors. On Thursday most of the visitors stayed for more than 2 hours, followed by visitors that stayed for more than 3 hours. The total number of violators at a given time ranged from 1 to 3 on both days. The total number of violators was almost equal for both days standing at 3 and 2 on Thursday and Friday, respectively.

The parking lot operated below the effective capacity for the entire period during which the survey was conducted. However occupancy values of almost 80% were reached on Thursday.

2.5.6 Perry-II parking Lot

The Perry-II parking lot is situated along Perry Street and had a total of 304 parking spaces at the time of the study. Of the 304 parking spaces, 292 were reserved for F/S and 12 spaces were metered parking spaces.

The total number of visitors at a given time ranged from 9 to 13 on Thursday and 4 to 13 on Friday. On average 12 and 9 visitors were present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 16 on Thursday and 19 on Friday. The number of visitors increased as time progressed on both days. The peak was reached between 10:30 AM and 11:00 AM on both days. The CON type of visitors accounted for about 70% of all the visitors followed by long term visitors with 13% and the remaining 18% was accounted for by the short term visitors, BUS, and CRC visitors equally. On Friday 26% of the total visitors were short term visitors followed by temporary and long term visitors accounting for 16% each. The remaining 32% was accounted for by the CRC, BUS, and retiree visitors equally.

On Thursday most of the visitors stayed for more than 4 hours, followed by visitors that stayed for more than 2 hours and 30 minutes. On Friday most of the visitors stayed for more than one and a half hours. The average duration of stay was approximately 3 hours on Thursday and 2 hours on Friday. On Thursday all violators had a parking permit. However on Friday the violators without permits accounted for 67% of the total violators whereas the remaining 33% was accounted for by violators with permits. Most of the violators stayed for 1.5 hours, followed by the violators that stayed for more than 4 hours on Thursday. The parking lot operated below the effective capacity for the entire period during which the survey was conducted.

2.5.7 Squires Parking Lot

The Squires parking lot borders on College Avenue and Otey Street. It has a total of 178 parking spaces. The total number of visitors at a given time range from 3 to 7 on Thursday and 0 to 16 on Friday. On average 6 and 7 visitors are present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 12 on Thursday and 18 on Friday. The number of visitors was almost constant on Thursday. On Friday the number of visitors increased with time. On both the days, long term visitors accounted for the highest number of visitors to the parking lot.

On Thursday the long term visitors accounts for 50% of all the visitors followed by short term visitors with 34% and the remaining 16% was equally accounted for by the retiree and CRC visitors. On Friday 61% of the total visitors were long term visitors followed by 28% short term visitors. The remaining 11% was accounted for by the retiree and CRC visitors. Most of the visitors stayed for less than 2 hours and some of the visitors stayed for more than 2 hours. On Friday most of the visitors stayed for 30 minutes followed by visitors who stayed for 1 hour and 3 hours. The average duration of stay was approximately 1 hour 35 minutes on Thursday and 1 hour 50 minutes on Friday.

The total number of violators at a given time ranged from 2 to 7 vehicles on Thursday and 0 to 5 vehicles on Friday. On average 6 violators were present in the parking lot at any given time on Thursday and 2 violators were present in the lot at any given time on Friday. The total number of violators was 9 on Thursday and 7 on Friday. The number of violators peaked between 10:00 AM and 10:30 AM and then decreased with time for both days. The percentage of violations without a permit constituted 89% and 71% of the total violations on Thursday and Friday, respectively. On Friday the frequency of violators staying for 30 minutes was the highest followed by violators staying for 1 hour. The parking lot operated above the effective capacity for the entire period during which the survey was conducted. The average occupancy of the parking lot for the entire period of time was approximately 92%.

2.6 Analysis of Historical Visitor Records

Analysis of the historical visitor parking records was one of the initial tasks of the study. Historical records regarding visitor registration and parking surveys were analyzed with a threefold objective: Firstly to determine the visitor's parking demand, secondly to study the statistical differences present among the different days of a week and thirdly to study the presence of statistical differences among the different months.

As was mentioned earlier, visitor parking permits are issued by the Visitor Center, the Virginia Tech Visitors Parking Center, and the Police Department at Virginia Tech. In addition, individual departments can contact Parking services in advance and request a specific number of parking permits. When a visitor arrives at the parking Center he/she needs to provide the following information:

- Name
- License Plate Information
- Time
- Number of visitor passes requested
- Number of visitors

Data, from January 1 to December 31 2007, pertaining to the total number of visitor parking passes issued and number of visitors (people) per day, time-of-the-day, day-of-the week, week, month, and year was analyzed.

There were a significant number of days for which no visitor parking data is available. These are the days which coincided with a special university event or a holiday. Engineering judgment was used and these days were excluded when computing the average and conducting the statistical analysis.

The number of passes and the number of visitors for the year are shown in Figure 5. The University does not issue passes on certain days and as a result some of the months may seem to have a less number of visitors, which might not be the actual case. The readers must exercise caution towards this fact.

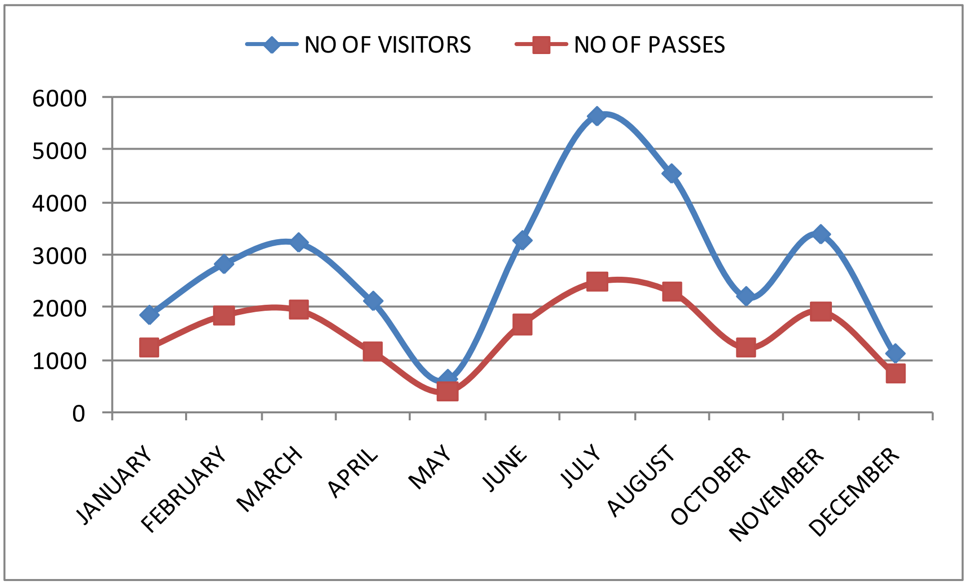


Figure 5: Number of visitors and visitor parking passes per month

The relationship between the number of visitors (people) and visitor parking passes was observed to vary from 1.5 to 1.8 for most of the months. The same relationship for the month of July was observed to be 2.27 and the average number of passengers per car was 1.8.

In order to study the statistical differences among the various days the average visitor parking passes for each day of the week for every month was computed as shown in Figure 7. In general May has the least number of visitor passes issued. When each day of the week is analyzed differences among months are more noticeable on Friday. This can be explained, for example by the fact that on Fridays, during football seasons usually more parking passes are issued.

A statistical t-test was done to check if there were significant differences between the various days. The results shown in Table 5 conclude that in general Mondays, Tuesdays, Wednesdays, and Thursdays are not statistically different, and that there is a statistically significant difference between Fridays and other days. These findings seem to demonstrate that the days selected to conduct the study Thursday and Fridays are a good representation of what happens during the rest of the week. Thursday was selected because it is a good representation of a typical weekday and Friday because it is statistically different for the rest of the days, showing a higher demand for the visitor parking.

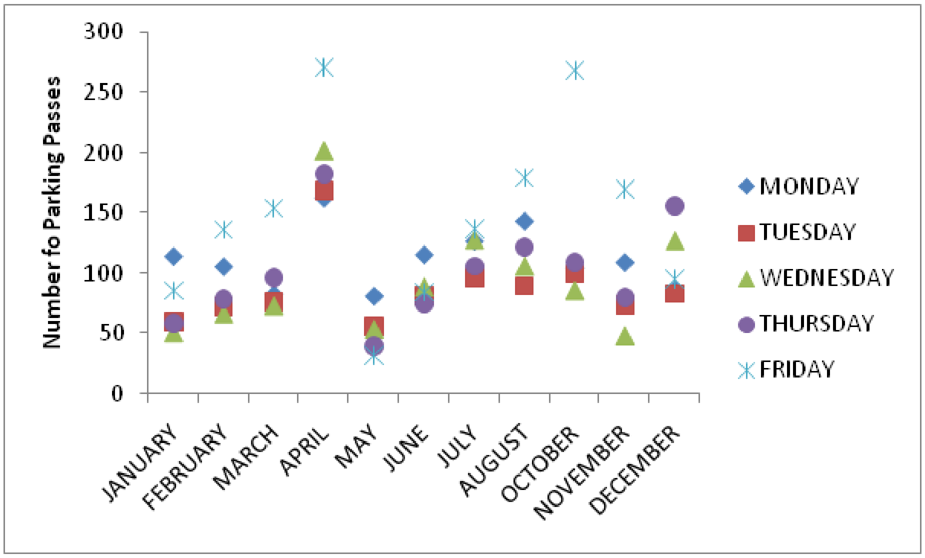


Figure 6: Number of parking passes per month per day of the week

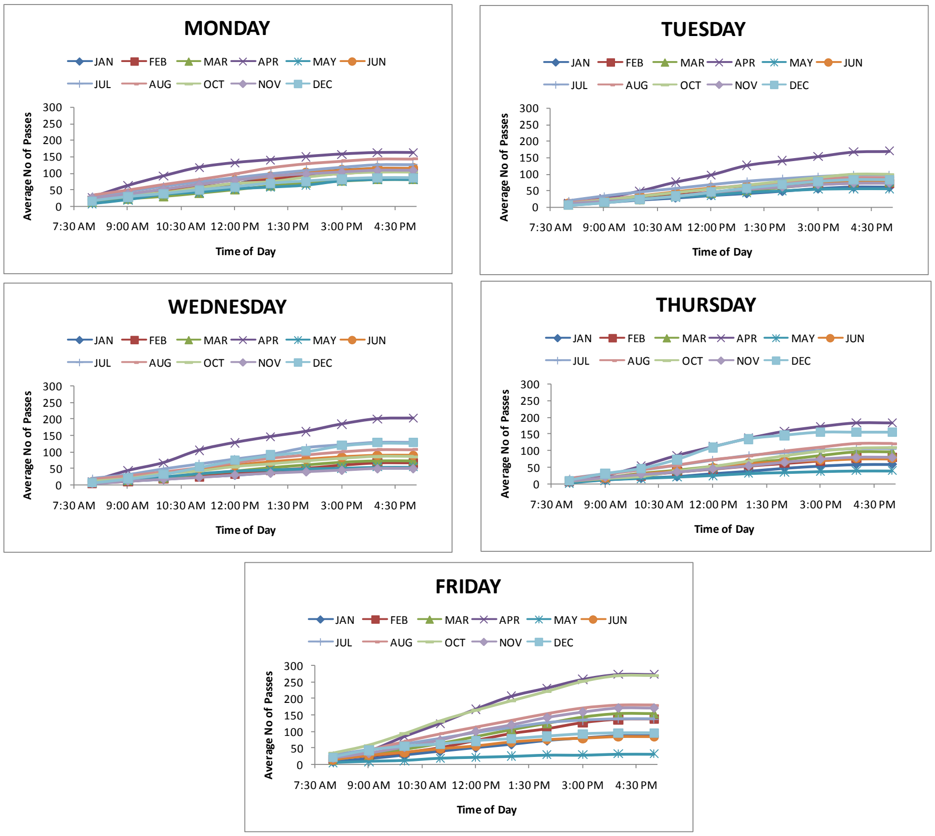


Figure 7: Average no of passes issued on different days in different months

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
MONDAY	1	0.052011	0.260526	0.429728	0.160238
TUESDAY		1	0.69749	0.419595	0.024165
WEDNESDAY			1	0.737527	0.058796
THURSDAY				1	0.086318
FRIDAY					1

Table 5: T-test for historical data for visitor parking passes

The statistical t-test was done to check if there were significant differences between the various days-of-the-week for the number of visitors to campus and the results are summarized in Table

6. The results, similar to the number of parking pass results, show that in general Mondays, Tuesdays, Wednesdays, and Thursdays are not statistically different, and that there is a statistically significant difference between Fridays and other weekdays. The results also show that the number of visitors for Mondays and Tuesdays are marginally statistically different.

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
MONDAY	1	0.0469	0.302751	0.49152	0.108742
TUESDAY		1	0.532444	0.275216	0.008162
WEDNESDAY			1	0.718549	0.034578
THURSDAY				1	0.053194
FRIDAY					1

Table 6: T-test for historical data pertaining to number of visitors on different days

The number of parking passes were also plotted for each month. In particular Figure 8, Figure 9, and Figure 10 show the number of passes issued per day for February, July, and November as representatives of the spring, summer and fall seasons. On average there is a request of 100 parking passes per day.

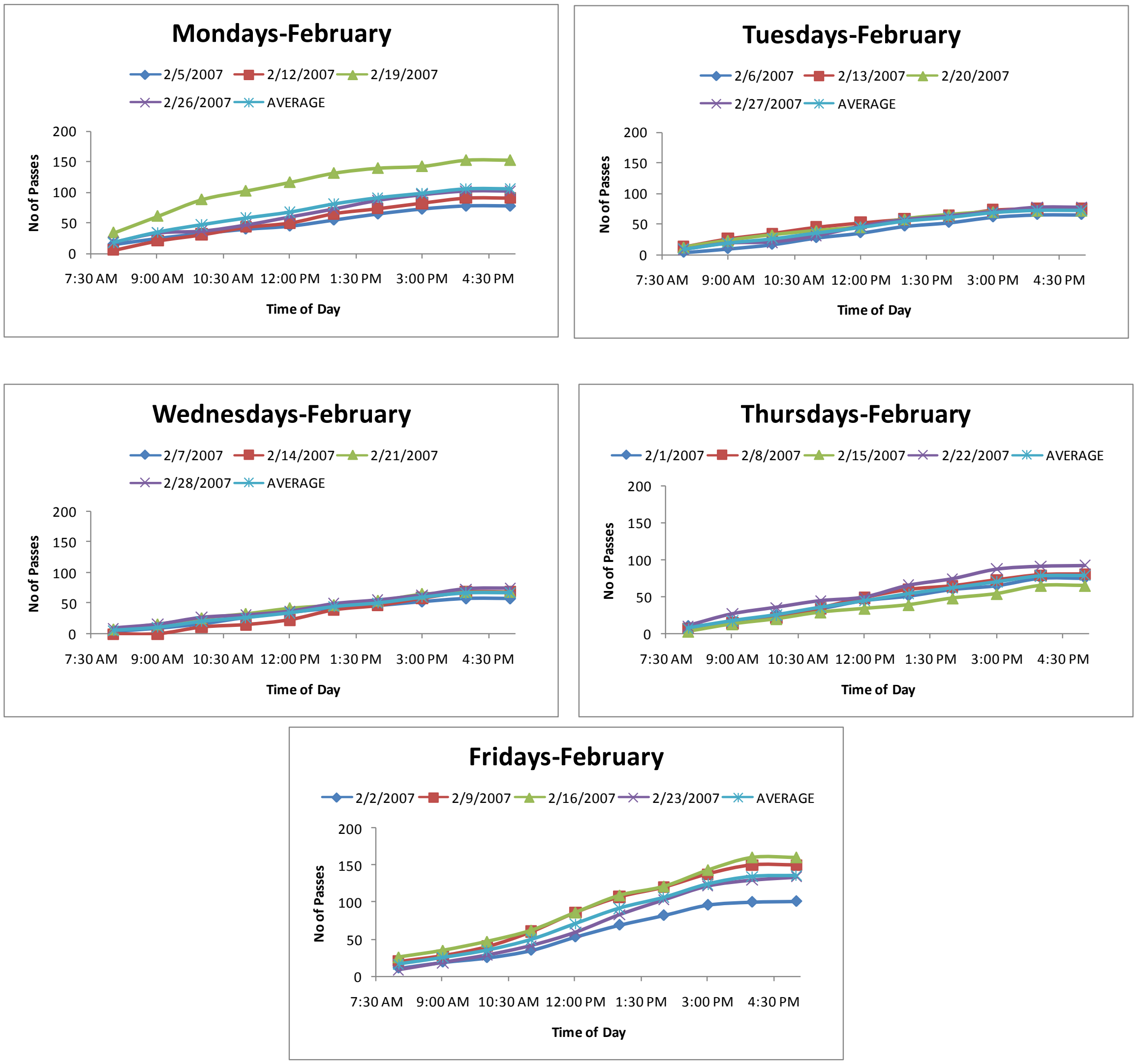


Figure 8: Distribution of number of visitor parking passes per day for February

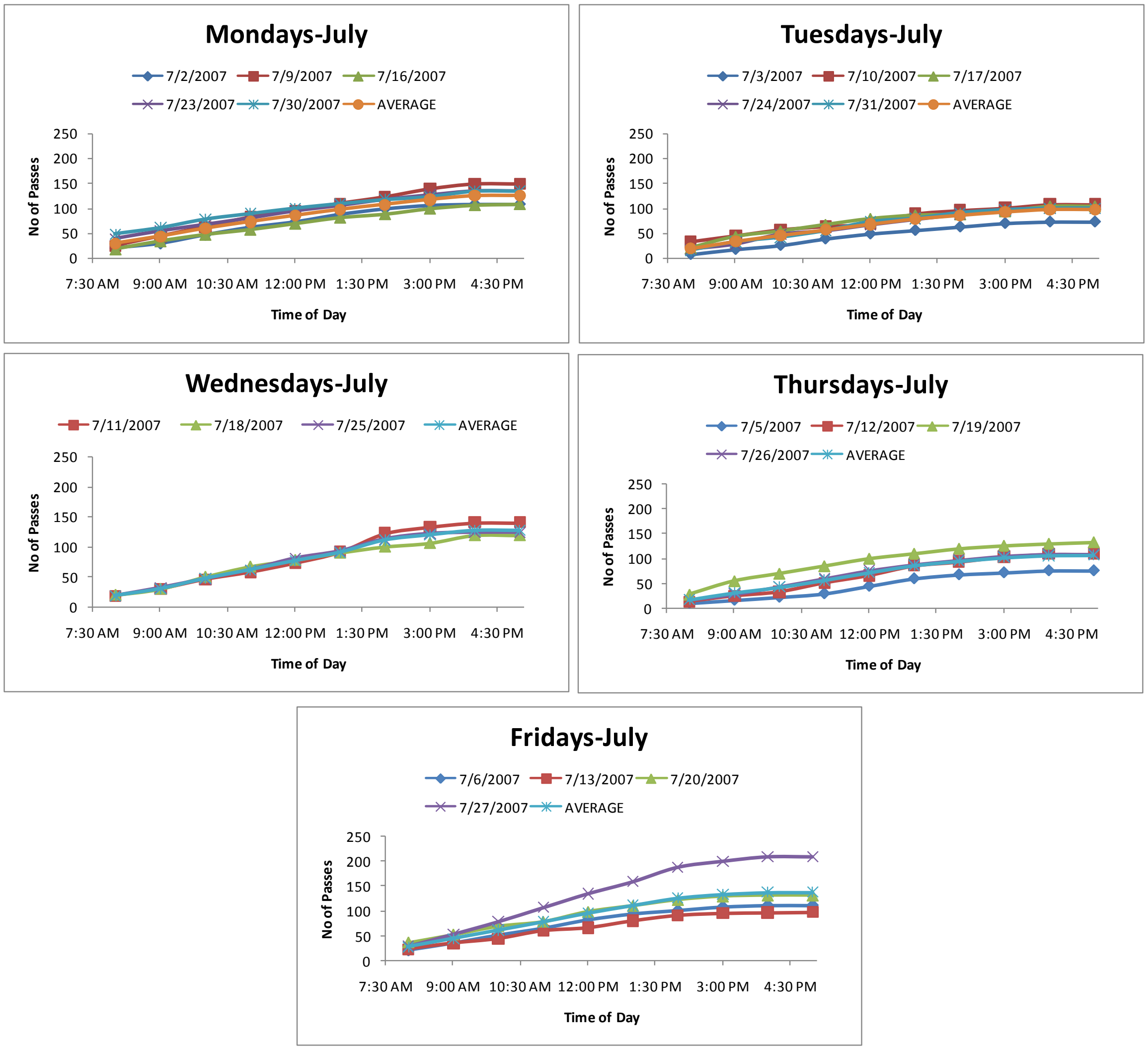


Figure 9: Distribution of number of visitor parking passes per day for July

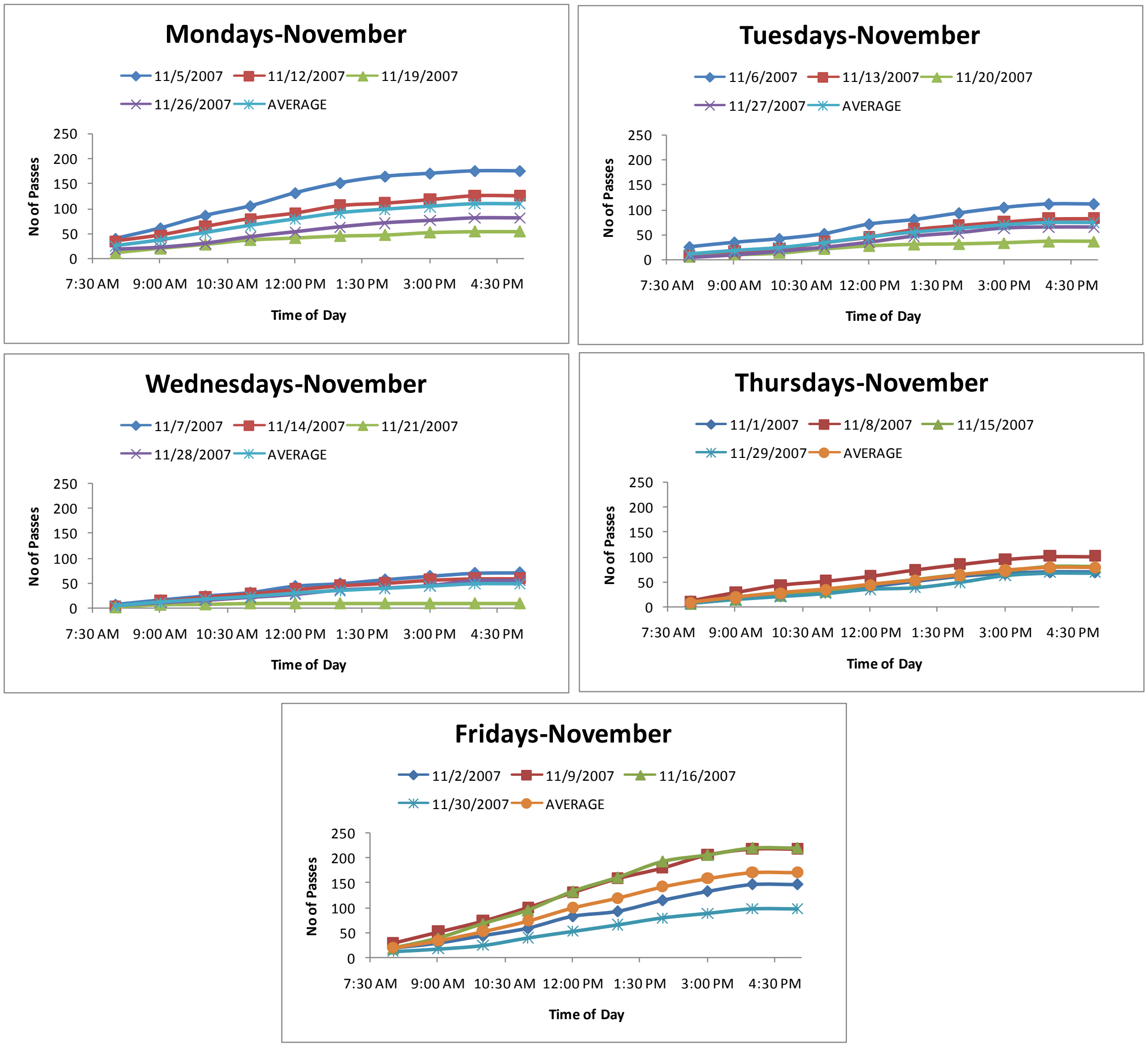


Figure 10: Distribution of number of visitor parking passes per day for November

2.6.1 Parking Lots

The second set of data is obtained through the Office of Transportation and the Virginia Tech Visitor Parking Center. The survey collected data on the number of parking spaces that were occupied by visitors at 8:00 am, 10:00 am, 1:00 pm and 3:00pm. The results are shown in Figure 11. Friday has the largest number of visitors for the first week. For the second week apart from Monday, all the other days have a relatively similar pattern. This fact clearly accounts to the fluctuation in the number of visitors from one week to another.

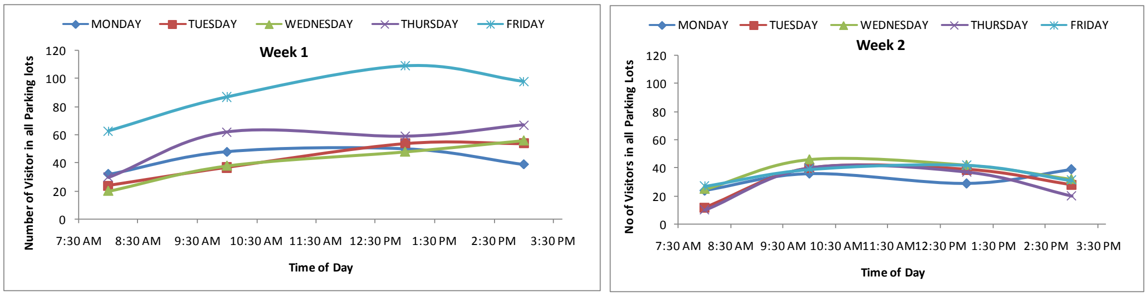


Figure 11: Number of visitors for all the parking lots

On average at any given time in the University there are 53 and 30 parking spaces occupied by visitors for week 1 and week 2 respectively.

A statistical analysis of the data was performed in order to determine if there were significant statistical differences between days of the week for the two weeks (Monday week 1 to Monday week 2). The results concluded that for a level of significance of 5%, there were significant statistical differences for each of the days of the week.

The results for only the five parking lots, where the field test was conducted are shown in Figure 12. On average at any given time in the University there are 27 and 16 parking spaces occupied by visitors for week 1 and 2, respectively. Furthermore these five parking lots represent 48% and 52% of the parking capacity, respectively, if all the parking lots are considered. So, it can be concluded that the field survey conducted would account for about 50% of the visitor occupancy. For week 1, Friday and Thursday have more visitors for the entire day. For week 2, almost all the days in the week follow a similar pattern.

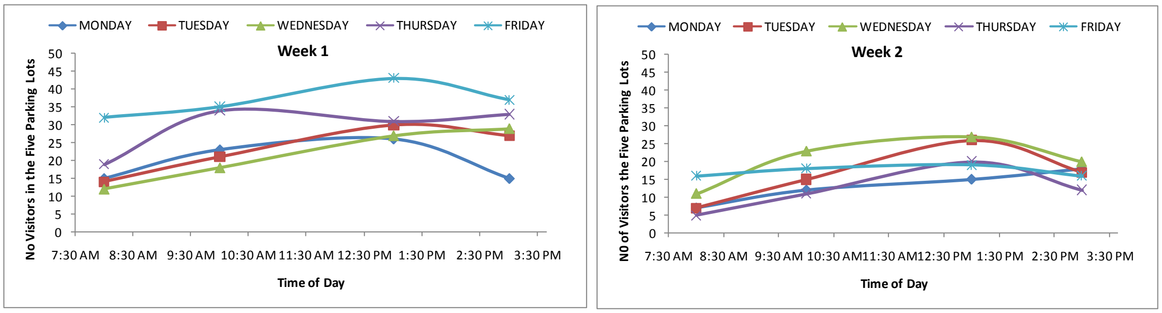


Figure 12: Number of visitors per day for the five parking lots

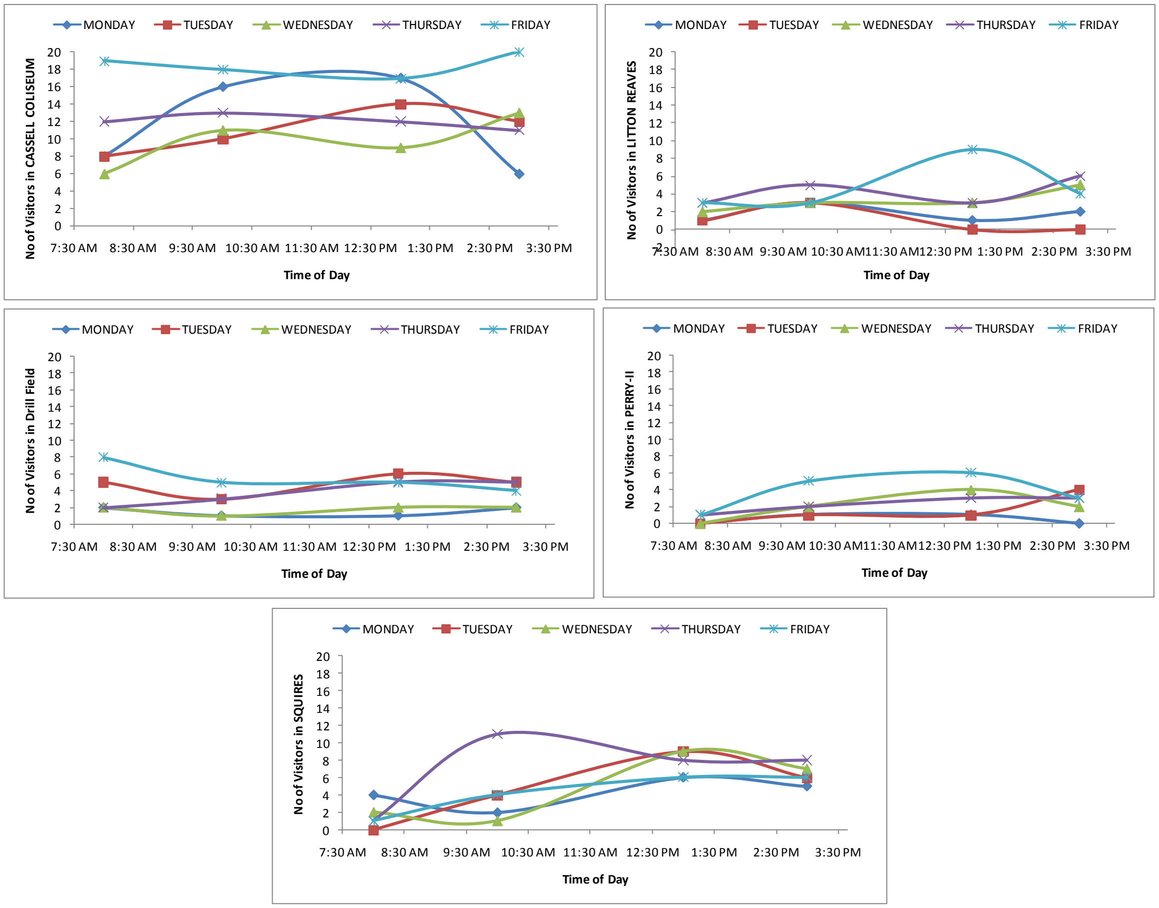


Figure 13: Number of visitors per day for each parking lot-week 1

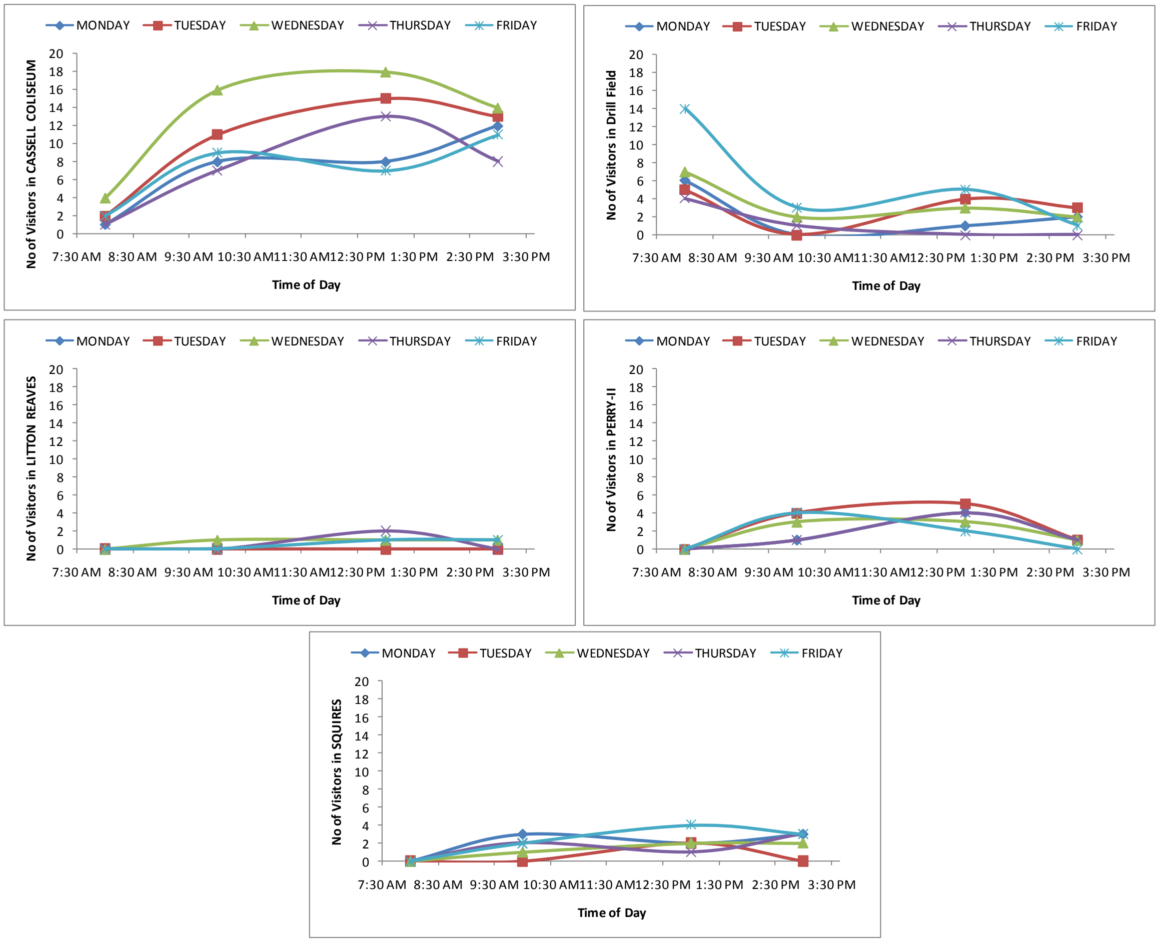


Figure 14: Number of visitors per day for each parking lot-week 2

These results are consistent with the ones obtained from the data collection effort. The Cassell Coliseum parking lot has the largest number of visitors and the one that presents more fluctuations for the different days of the week. However Thursdays and Fridays present patterns that were consistent with the field data collection efforts. If not for the Drill Field the number of visitors present at any given time increases during the day until it reaches a peak.

2.6 Study Conclusions

In this section, the conclusions drawn from the Virginia Tech Parking Management Study Phase III are provided as shown below

- Majority of the trips to the campus are undertaken by prospective students, family members of prospective students and family members or friends of current students
- About 20% of the people did not know the location of the offices they needed to visit and almost 27% of the people have not responded to this question
- Almost 28% of the visitors did not know the location of the most convenient parking lot to access their destination and about 11% of the visitors have not responded to this question
- Approximately, 32% of the visitors had to visit at least 2 parking lots before finding a parking space
- About 6% of the people have rated the parking services offered by Virginia Tech poorly but 28% of the people did not respond to this question. So, it cannot be said that the parking services are doing good with complete confidence
- Majority of the visitors showed their unwillingness to pay for designated visitor-only parking spaces
- Cassell Coliseum gets a lot of visitors and the effective capacity of the lot is reached around 10:00 AM. To assist the visitors, some of the Faculty/Staff parking spaces can be designated to the visitors from 8:00 AM till 5:00 PM. Though Squires lot operates above capacity throughout the day it is not recommended to create visitor only spaces as it is a very small parking lot
- For most of the parking lots, the number of Violators is quite high. To reduce the number of Type I violators, parking services has to disseminate the information more effectively through the use of web pages and pamphlets. To reduce the number of Type II violators, parking services must continue with an aggressive control policy

CHAPTER THREE: WEB BASED INTERACTIVE VISITOR PARKING SYSTEM

3.1 Problem Statement

Visitors constitute an important portion of the users of the parking lot facilities on the University campus. Based on the results of the previous chapter, Virginia Tech receives an average of 100 parking pass requests per day during the weekdays at the visitor information center, located on the Southgate Drive. A Visitor [5] 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. Virginia Tech's main campus in Blacksburg has a campus area of about 2600 acres and has about 93 surface parking lots. As such, visitors have very limited knowledge about the geographic location of the university parking facilities and services. Furthermore, there is no guarantee that the visitor finds a parking space even after locating a parking lot successfully near his/her destination since most of the parking lots operate at capacity during the week days.

The visitor survey conducted as part of the previous chapter showed that 20% of the visitors did not know the location of the office they were heading to and 28% of the visitors had no knowledge about the location of the most convenient parking lot to access their destination. Approximately 6% of the visitors rated the parking services provided by Virginia Tech to be poor. 14% of the visitors considered the parking services fair enough. The follow-up survey showed that for 7% of the visitors took more than 10 minutes to find a parking space. 14% of the visitors took 5 to 10 minutes to find a parking space. About 32% of the visitors had to visit more than 2 parking lots before finding a parking space. 9% of the visitors rated the parking services poorly after their visit. These figures demonstrate the potential for improving the visitor parking services.

3.2 Research Goals and Objectives

The main goal of this chapter is to develop a web-based interactive framework which uses historical parking lot occupancy data together with distances from various parking lots to buildings on campus to identify candidate parking lots for the visitor to select from. The specific objectives of this research effort are summarized as follows:

- Develop a web based interactive framework to assist visitors in the selection of convenient parking lots to access their destinations
- Design the system in a flexible way so as to accommodate any future updates in terms of the database

3.3 Web Based Interactive System

In order to provide the user with details of different parking lots, to access his/her destination, a Web based interactive system is created. This system helps the user to input the details pertaining to his/her destination, performs a search based on the given criteria and finally retrieves and prints the result in a tabulated format. The result consists of a list of 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 the HTML forms provided. The data pertaining to the parking lots and buildings is stored in a database using a database management system. When the user requests the parking lot information near his/her destination, the data existing in the database has to be retrieved and printed 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 about WAMP system which is necessary to deliver the above tasks is provided in the next section.

3.4 WAMP System

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 is absolutely necessary to store the parking data at some location (usually a Computer) and retrieve it as and 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, which stands for PHP: Hypertext Preprocessor 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 occurs as shown in Figure 15.

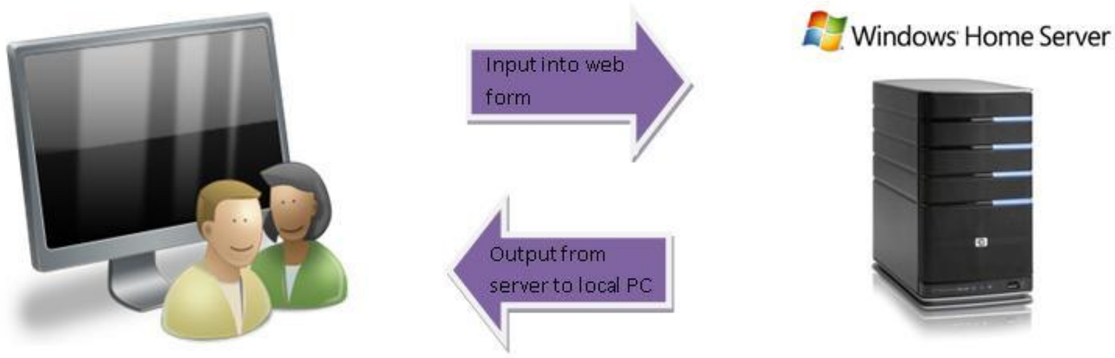


Figure 15: Data flow between a Local PC and a Web Server

3.4.1 WAMPSERVER

To have a WAMP system running on the local windows based PC, one can either install separate versions of MySQL, PHP and Apache Software or install commercially available software which packs all the three required programs into a single installation file. For the purpose 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. After installing WAMPSERVER, it should be started from the start menu and an icon will be displayed in the system taskbar tray. This icon will be displayed in different colors according to the working status of the WAMP system. The icon will appear as

- Red and white if the WAMP system is started and is running but all the services are stopped or disabled,
- Yellow and white if the WAMP system is started and is running but one of the two services i.e. Apache or MySQL did not start, or
- White if the WAMP system is started and is running and all the services are running fine.

There is also an option to put the server offline by clicking on the icon in the system tray.

3.5 MySQL-PHP Platform

Any interactive database application will have what is known as a back end and a front end. 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 will provide 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 called *dynamic duo* [6]. A MySQL-PHP platform, when used together, has many advantages. Some of the advantages are summarized below

- **Cost:** Both MySQL and PHP are freeware (free software) and this would lessen the financial burden on small application developers.
- **Ease of Use:** PHP and MySQL are quite easy to learn and both software work with simple English commands which makes the learning process simple.
- **Interoperability:** Another big advantage of using PHP and MySQL platform is its interoperability. Interoperability refers to the ability of software to work on different platforms. For example, MySQL and PHP can work on Windows, Unix or MAC operating systems.
- **Web-Oriented:** Both software are designed to specifically serve the purpose of usage on the internet. They have strong features which enable them to focus on the development of dynamic websites.

- **Speed:** One of the main objectives of the developers of PHP and MySQL was speed. Both software were designed with speed as a major goal and provides one of the fastest platforms to deliver dynamic web pages.
- **Communication:** Both PHP and MySQL communicate well with one another. PHP has some built-in features which improve its communication efficiency with MySQL. There is no necessity for the user to know the technical details unless he/she is interested in getting to the core of the relationship between PHP and MySQL.
- **Support:** Both software are popular and have a large base of support. There are several forums and listservs on the internet which help in troubleshooting the software and the code developed using the software. This thesis would not have been a reality without the invaluable support given by several PHP and MySQL Gurus out there on such forums and Listservs.
- **Open Source:** Both software are open source. Open source software refers to a computer software whose source code is available to everyone under a software license that meets the Open Source Definition [7]. Open source software can be modified by the programmers to suit their working environments.
- **Stability:** Stability is another main feature incorporated into PHP and MySQL. Stability refers to the ability of server to run without being rebooted often and uniformity in the software without radical and incompatible changes from release to release [8]. This claim is substantiated by the fact that in a two and a half month head to head test conducted by Network computing labs in October 1999, Apache server powered PHP outplayed IIS/Visual Studio and Netscape Enterprise server/Java for stability of environment [8].

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 path of the flow of information or data using a PHP-MySQL platform is shown in Figure 16.

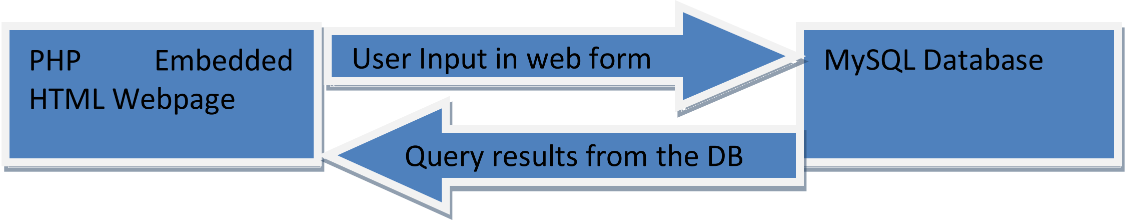


Figure 16: Information Transfer between PHP and MySQL

3.6 MySQL

MySQL is named after its co-founder Monty Widenius's daughter, My [9]. The SQL in MySQL stands for Structured Query Language. MySQL is an open source software and its source code is available under the license agreement of GPL (GNU General Public License). Under this license, anyone can download, use and modify MySQL software provided that the license terms are not violated. MySQL AB, a commercial company, founded by MySQL developers looks after the development, distribution and support of MySQL.

MySQL is a relational database management system (RDBMS). A database works as an electronic file cabinet and stores information in an organized manner. Technically, *database* refers to a structured collection of records or data that is stored in a computer system [7]. To add, modify, access, process and retrieve the data stored in a database the use of a database management system is often indisputable. A database management system can follow any of a variety of database models that are available. The most important of the database models are the network model and the relational model. In a network model, the data is arranged in a hierarchical structure and each record has a parent record and may have many children records. In a relational model, data is organized and stored in a set of related tables. The tables are related to each other with the help of a key, which is a unique identifier to every cell in a table.

MySQL is an RDBMS, which is very easy to use and which is quite fast. MySQL developers have focused more on speed and to attain this goal they have decided to offer fewer features when compared with other RDBMS software.

3.6.1 MySQL Server

One of the main components of the MySQL software is the MySQL server, which is the heart of the system. MySQL server manages the entire database system and processes all the database instructions or commands given by the user. If a new row of data is to be added to an existing table, a command is to be sent to the MySQL server to that effect and MySQL server adds a new row of data. However, for this operation to be successful, the command being sent to the server should follow the MySQL syntax. To aid in the design of the interactive parking information system, the MySQL server installed, as part of the WAMPSERVER package, is named *mysqld* which stands for *mysql daemon.* This is the default server type chosen by many MySQL users.

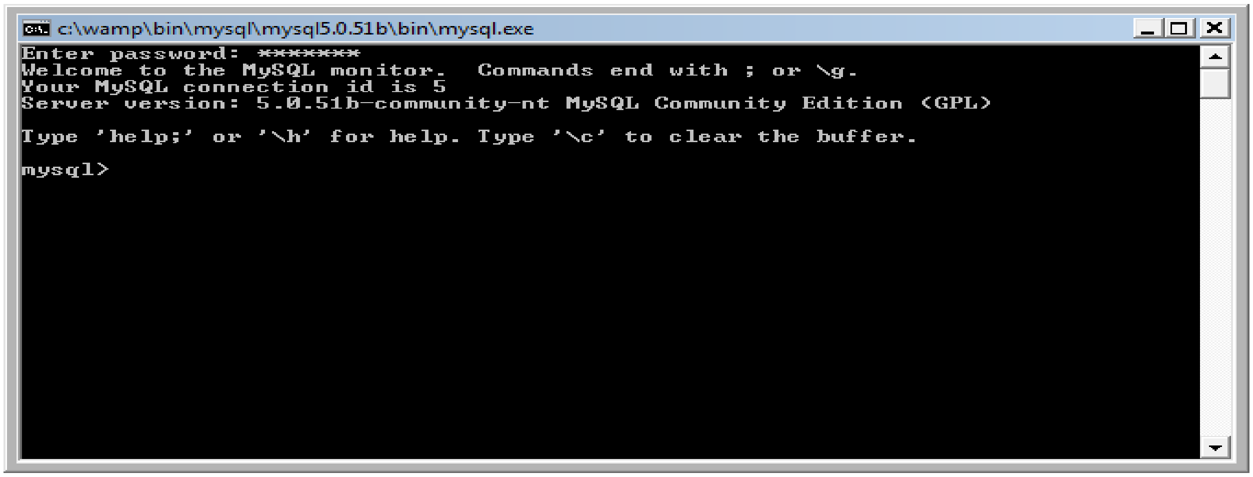
As already mentioned, MySQL can be installed as an individual software or collectively within the WAMPSERVER system. WAMPSERVER version 2.0 is installed onto the system and the MySQL version provided by this package is 5.0.51b. WAMPSERVER installation is quite simple. During the installation, a particular pop window asks for a username and a password to be setup to access the MySQL server. It is very important to remember this username and password as the connection to MySQL server will not be established without either of them.

3.6.2 Connecting to the MySQL Server

There are several ways to communicate with a MySQL server. This thesis mainly deals with two ways of communication with the MySQL server. For both ways to work properly, WAMPSERVER has to be running in the system taskbar tray. The color of the WAMPSERVER icon should be white.

In the first method, place the cursor on the icon and left click on the WAMPSERVER icon. This brings up a popup menu. Then, place the cursor on the MySQL menu option which brings up some more menu options. Click on the *MySQL Console* option and a command processor (CMD) window will pop-up asking for the password. If the password is typed incorrectly, an error message will appear and the command processor window will close down. If the password is typed correctly, a welcome message will appear on the command processor window along with some background information about the MySQL server version. Once the connection to the server is established, the user can create, update, modify, process and retrieve data. The other way to establish a connection to MySQL server is by using the PHP query language. A connect command will be used to connect to the MySQL server. By default, this connect command will need the username and password to access the MySQL server.

In the design of the parking information system for this thesis, the data entry and data modification was applied using the MySQL Console. All the commands pertaining to the creation and modification of the database are typed in the command processor window. Once the database is created successfully, the data retrieval is done by directing the commands to the MySQL server through the PHP query language. The data retrieved from the database is provided to the system's user on an html webpage. A screenshot of the MySQL command processor (CMD) working environment is shown in the Figure 17. As can be clearly seen, the first requirement to access the MySQL server is to enter the password. Once a correct password is entered, the connection is established to the MySQL server. The prompt, *mysql>* indicates a statement that one can execute from the mysql client program [9].



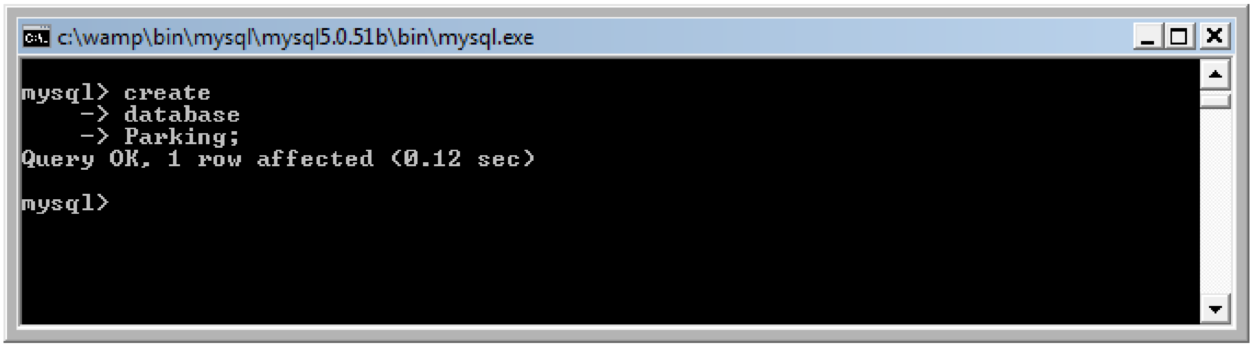
```
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.0.51b-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Figure 17: MySQL Command Prompt Working Environment

3.6.3 MySQL Queries

After connecting to the server, there is a need to issue commands or queries to create, select or update a database. When a command or query is issued, mysql sends it to the server and the command is executed. If any error is encountered during the execution of the command, MySQL CMD displays the error message with a brief explanation about the error. The *mysql>* prompt, shown in Figure 17 indicates that MySQL is ready for new commands.

Generally, a command typed in the CMD consists of an SQL statement with a semicolon in the end. The semicolon indicates the end of a statement and MySQL will send the statement to the server after it finds the semicolon. As such, an SQL statement can be written in either as a single line or multiple lines. When entering a multiple-line command, the *mysql>* prompt changes to -> after entering the first line of the command as shown in the Figure 18. However, the semicolon is not needed for some statements like *Quit* or *End* used to exit the MySQL CMD.



```
mysql> create
    -> database
    -> Parking;
Query OK, 1 row affected (0.12 sec)

mysql>
```

Figure 18: Multiple-Line Query

3.6.4 MySQL Database

Database refers to a structured collection of files or groups of files that hold the data which is of interest to the user. It is very important to plan the structure of a database before creating it. A database should be planned thoroughly, keeping in mind, the need for the application and the requirements of the user. A database can store large amounts of data, but storing unnecessary data will seriously affect the speed of the application. To overcome this constraint, it is quite important to plan the database ahead and load only the data that is of interest to it. Each database will have a unique name assigned to it.

3.6.4.1 MySQL Data Organization-Tables

MySQL which is an RDBMS (Relational Database Management System) stores the data in tables. Each table is comprised of a combination of rows and columns. The intersection point of a particular column with a row is called, *field*. The data is uploaded into these fields. Each table will have a specific name assigned to it. Linux and Unix systems are case-sensitive but Windows is not a case-sensitive operating system as far as MySQL is concerned. For example, a table named *PARKING* is different from a table named *Parking* in a Linux or Unix system. However in Windows, PARKING, Parking and PaRkInG refer to the same table. Generally, the first row in any table represents the labels for the columns. Labels refer to the names assigned to the columns in a table. Technically, a data element stored in a row is referred to as an *entity.* Each *entity* has some *attributes* assigned to it. It can be generally said that each row represents an *entity* and the columns contain the *attributes* for each *entity.* For example, in a database containing the information of students in a university, each row contains the information for a single student (entity) and each column contains an attribute of the entity, like the first name, last name, ID number and date of birth of a student.

It is a good practice to name a column such that the name identifies the information in that column. The column name should be created from a single word. Spaces are not allowed in the name of a column or table. Some words are reserved by MySQL for its own use and they cannot be used as column names. ADD, SELECT, TABLE, ORDER and USE are some of the words that MySQL has reserved for its use.

3.6.4.2 Primary Key

Generally, there will be repetition of data in almost all the database tables. For example, a database containing the first names of the people may have a particular name in more than one row as shown in Table 7. In this example, if the records pertaining to John residing in VA are required, then a query has to be sent to the server quoting the ID number 1234. However, if a query is sent to the server quoting the first name only, then the records pertaining to John residing in VA and John residing in TX will be returned to the user which is not what was asked for in the first place. If a query is sent to the server quoting the state name only, then the records pertaining to John residing in VA and Jim residing in VA will be returned to the user. It can be seen that a correct record is returned when the ID number is quoted in the query. It can also be seen that there is no repetition of the ID number in the table shown below. This unique identifier (ID number, in this case) which identifies each row in the table is called a *Primary Key.*

A primary key can be made of a single column or a combination of columns. If a primary key is made of a single column, then no two rows can have the same value in that column. Alternatively, if the primary key is made of a combination of columns, the combination should not repeat itself in any other row. While creating a table, the column(s) which form the primary key have to be identified.

ID	First Name	State
1234	John	VA
1235	Joseph	TX
1236	John	TX
1237	Jim	CA
1238	Jim	VA

Table 7: MySQL Table

3.6.4.3 Default and Null Values

During the creation of database tables, it is often required to define default values for some columns. These default values will be assigned to the field if no data is entered into the field. For example, in a database containing information about taxpayers, VA can be assigned as a default value for the state. So, if no value is entered for a particular record in the state column, the state name VA will be assigned to it. But, if for example, TX is assigned to the state name column then the default value of VA will be overwritten with TX.

In a database application, it is required to define if a column can be empty (NULL) or not. If a column is defined *NOT NULL*, then all the data in the column must be entered by the user. If some fields are left empty by the user, then the MySQL server returns an error message. For example, the primary key column cannot generally have an empty field. All the fields must be filled by the user in a primary key column. In this manner, this feature can be used as an error check by defining the primary key column, *NOT NULL*. If the *NOT NULL,* attribute is not applied to a column and if the user does not fill any data into a field in that particular column, then MySQL automatically creates a field but leaves it empty.

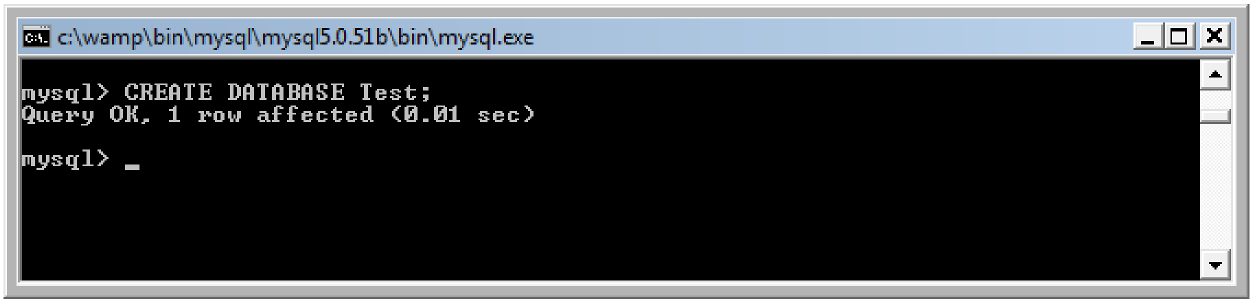
3.6.4.4 Data Types in MySQL

A data type refers to the attribute or characteristic of data together with information regarding the type of data [7]. MySQL supports a variety of data types including numeric, date and time and string (character) types.

In the string data type, *CHAR* and *VARCHAR* are the two types of data commonly used. Both are declared by indicating the maximum string length (number of characters) one would want to use. For example, *CHAR (25)* indicates that the particular element can hold up to 25 characters. In the development of this application, the most frequently used data type was *VARCHAR* which corresponds to a variable length character string. The difference between *VARCHAR* and *CHAR* is that, if an attribute is declared as having a maximum string length of 60 characters and if a string of 45 characters is used, then *VARCHAR* stores the 45 characters and discards the remaining 15 empty spaces whereas *CHAR* stores the 45 characters and will leave the extra 15 spaces empty. Thus, there is wastage of space when a *CHAR* data type is used, when compared to the *VARCHAR* data type. One more data type that is used in this application is *INT* which corresponds to integer type data. This data type stores numerical data. An attribute can be defined as an integer type data by using, *INT (length)* command. The length corresponds to the maximum number of digits allowed in the entity.

3.6.5 Database Management

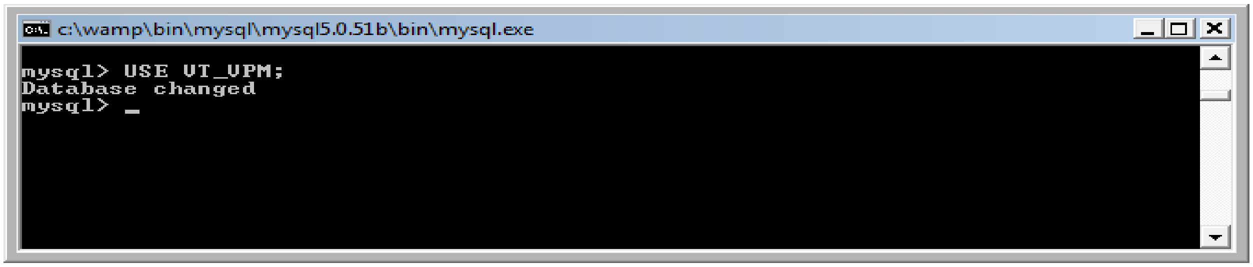
The first step in the creation of any database management application would be to create the database itself. This can be done through a simple *CREATE DATABASE* command as shown in Figure 19. As can be seen, the command creates a new database with the name Test. For the purpose of this parking application the database is named ***VT_VPM*** which stands for ***Virginia Tech_Visitor Parking Management***.



```
mysql> CREATE DATABASE Test;
Query OK, 1 row affected (0.01 sec)

mysql> _
```

Figure 19: Creating a Database in MySQL

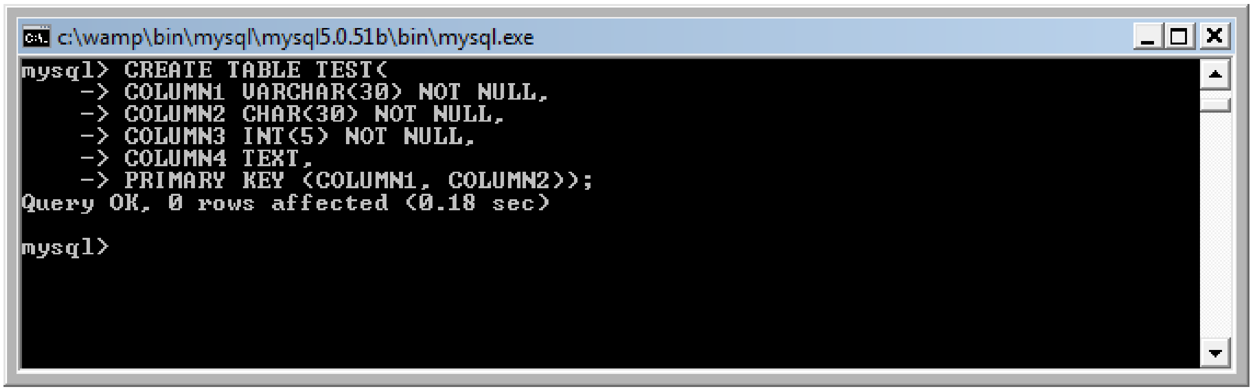
After the creation of the database, the next step would be to select the database to use it. Creating a database will not automatically make it available for use. A database can be made available through a *USE* command as shown in Figure 20. Once a *USE* command is used, a database will be selected and this can be confirmed by the *Database Changed* message that appears in the CDM. If an error is encountered during the database selection process, the MySQL server displays a suitable error message in the Command Processor Window.



```
mysql> USE VT_VPM;
Database changed
mysql> _
```

Figure 20: Selecting a Database for use

After selecting a database, the next step is to create a table. The structure of a table has to be planned well in advance of its creation. A table can be created using a *CREATE TABLE* command as shown in Figure 21. In this case, the table is named TEST. If the MySQL server creates a table successfully, a message will appear in the CDM which says *Query OK.* An approximate processing time taken to complete the query is also displayed in the CDM. In this example, the table is created with four columns and the combination of column1 and column2 forms the primary key. It can also be seen that entities falling under Columns 1, 2 and 3 cannot have a null value. But, entities falling under column4 can take a null value.

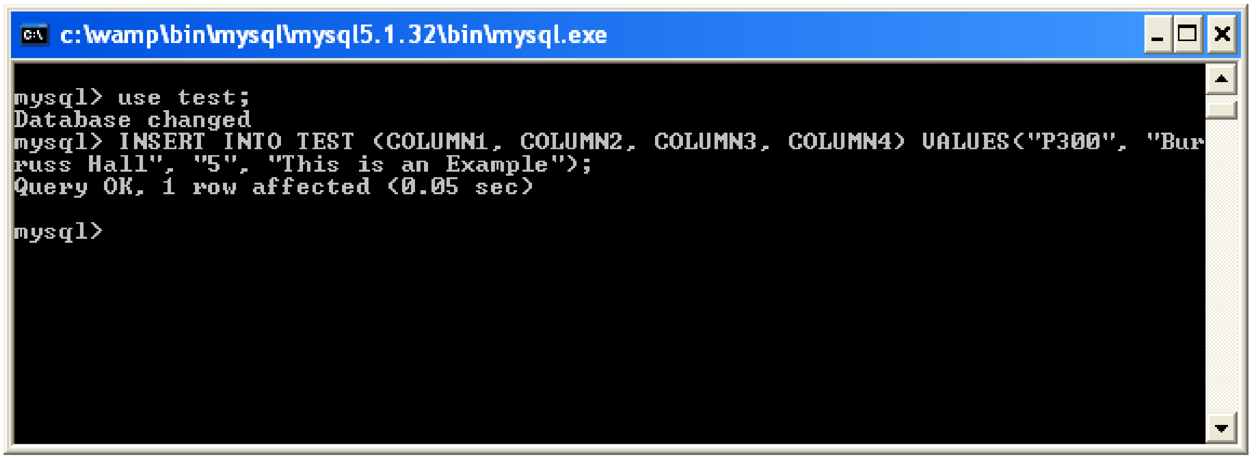


```
mysql> CREATE TABLE TEST(
    -> COLUMN1 VARCHAR(30) NOT NULL,
    -> COLUMN2 CHAR(30) NOT NULL,
    -> COLUMN3 INT(5) NOT NULL,
    -> COLUMN4 TEXT,
    -> PRIMARY KEY (COLUMN1, COLUMN2));
Query OK, 0 rows affected (0.18 sec)

mysql>
```

Figure 21: Creating a Table in MySQL

After the creation of a table, the next logical step would be to add data into the table. Data can be added into the database in two ways. In the first method, data can be added to the table one row at a time and in the second method, all the data can be loaded into the table from some other supported file format.

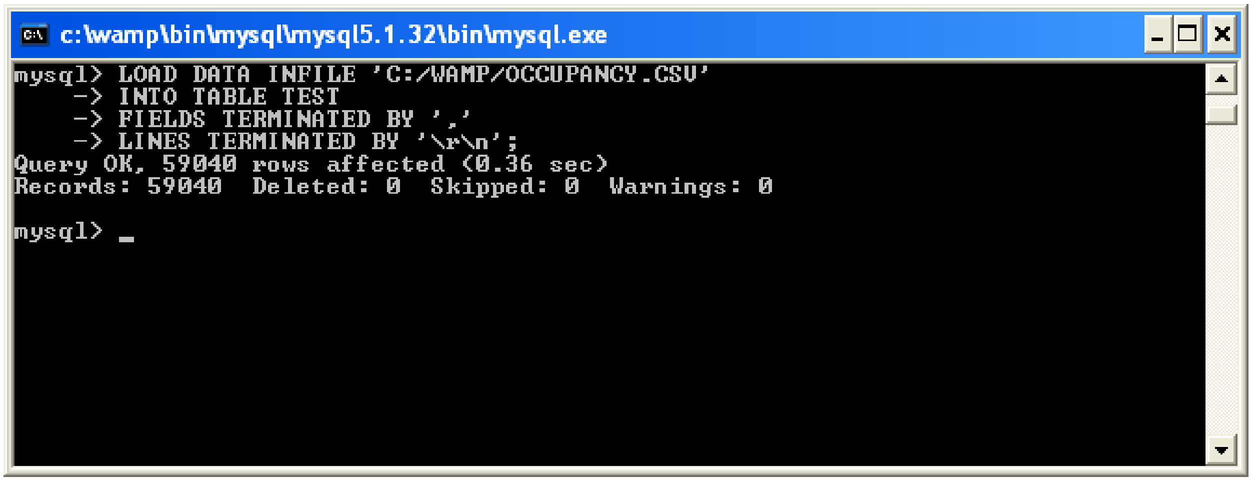
Data can be added to the database one row at a time using the *INSERT* query as shown in Figure 22. Here, the data is entered into the table TEST. The columns into which the data is entered have to be listed in the query. The first value in the values list will be added to the first column in the columns list, second value in the values list will be added to the second column in the columns list and so on. If the data is entered into all the columns in a table, then there is no need to list the columns in the *INSERT* query. When typing the values for entering into the table, they should be enclosed in double quotes, as that is the standard syntax requirement for MySQL. This method of data entry will consume a lot of time for typing the queries. Furthermore, if a huge amount of data is to be loaded into the database, this method is time consuming as it might involve some hundreds of hours of typing.



```
mysql> use test;
Database changed
mysql> INSERT INTO TEST (COLUMN1, COLUMN2, COLUMN3, COLUMN4) VALUES("P300", "Bur
russ Hall", "5", "This is an Example");
Query OK, 1 row affected (0.05 sec)

mysql>
```

Figure 22: Inserting Data One Row at a Time

Instead of using the *INSERT* query to load the data, one can use the *LOAD* query to load huge amounts of data into a table without consuming much time as shown in Figure 23. To be able to use the *LOAD* query, the data has to be available in a text file or a Comma-separated values (CSV) file. The *LOAD* query has to specify the file path. In this example, the file is stored at the location C:\WAMP\OCCUPANCY.CSV. Though the windows system uses a backward slash to indicate the file path, it has to be replaced by a forward slash while giving the file path to the MySQL server. The query may also use some optional phrases like *FIELDS TERMINATED BY* and *LINES TERMINATED BY* to load the data into the fields correctly. All the characters, either indicating the file path or the field and line termination should be enclosed in single quotes. If the query is successful in uploading the data into the table, a *Query OK* message is returned by the server. The number of updated rows in the table (or the number of rows in the uploaded file) is also displayed in the CMD. In this example, a total 0f 59,040 records (or rows of data) are uploaded into the table. A summary of skipped and deleted records will also be displayed in the CMD after the query is sent successfully to the server.



```
mysql> LOAD DATA INFILE 'C:/WAMP/OCCUPANCY.CSV'
    -> INTO TABLE TEST
    -> FIELDS TERMINATED BY ','
    -> LINES TERMINATED BY '\r\n';
Query OK, 59040 rows affected (0.36 sec)
Records: 59040  Deleted: 0  Skipped: 0  Warnings: 0

mysql> _
```

Figure 23: Loading the Data From a File

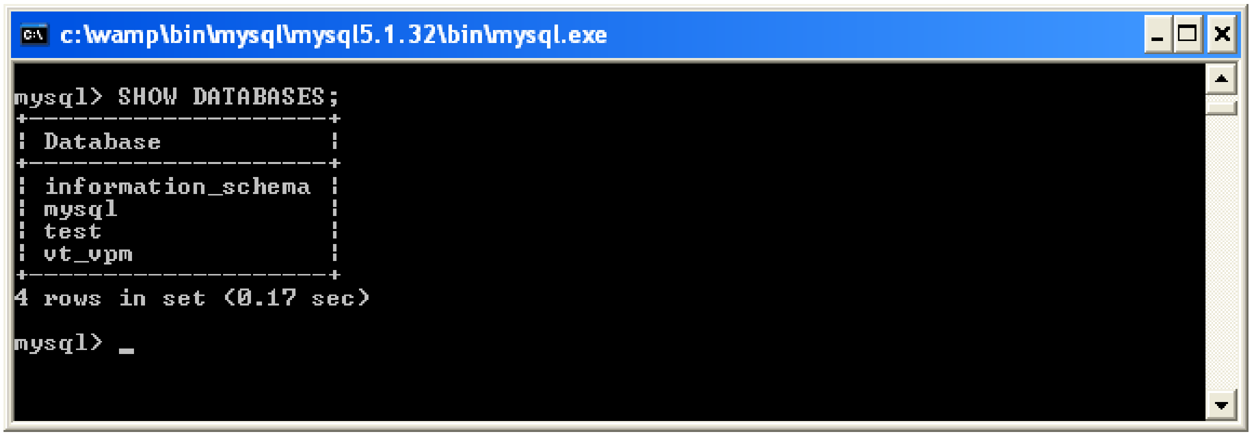
A database might consist of several tables. If the tables are not related to one another there is no use of creating several tables. In a relational database, the individual tables which store the data are related to one another. For example, a row in one table might be related to several rows in another table, based on the column attributes. Columns are needed to relate the tables with one another. Generally, the data in the column of a table must match the data in the column of a second table with the column in question in the second table being a primary key column, to relate two tables together. A detailed discussion about relating tables is provided later.

3.6.6 Parking Database Structure

To create the parking information system, data pertaining to the buildings on the Virginia Tech campus, Parking lots, number of empty spaces in each parking lot according to the time of the day, day of the week and month of the year and distances between the parking lots and the buildings was found to be necessary. Apart from this data, to assist the user in a more extensive way, it was decided to provide the user with a campus map highlighting his/her destination building along with the parking lots located near his/her destination.

As already mentioned, a database is a collection of files (tables) that hold the data in an organized manner. In the development of this parking application, 5 tables were created in total to manage the data in an effective way. Each table consists of some columns, which store the required data. Since there are a combination of tables and columns, it can get quite confusing accessing different data elements and it may not be possible to remember all the details regarding the tables and their constituents. However, MySQL provides the user with some queries and commands which help in determining the elements in a database.

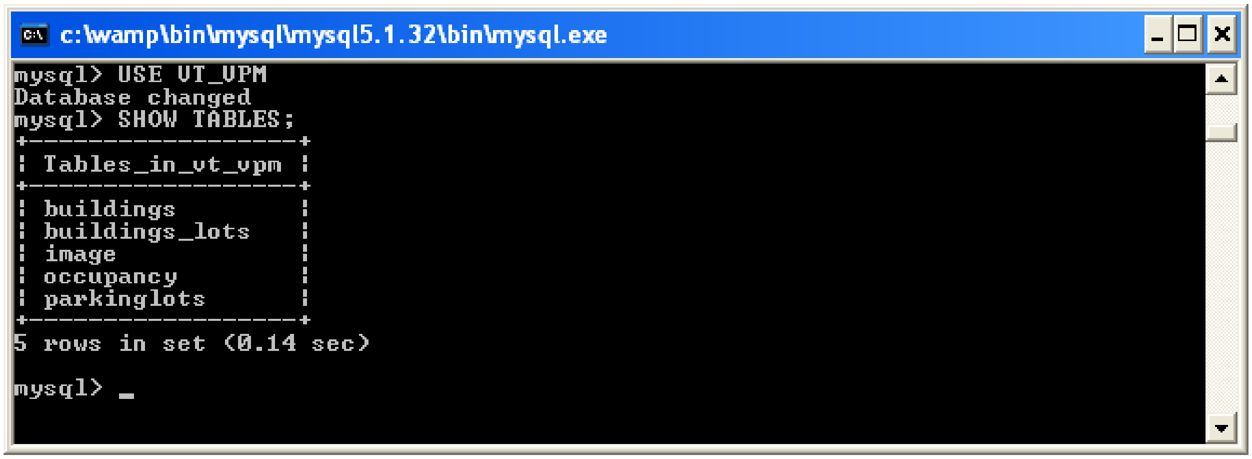
The first step in acquainting oneself with the database would be to know the name of the database. If the user does not know the name of the database, a *SHOW* query can be used to generate a list of databases existing on the server as shown in the Figure 24. Once the query is sent to the server, it delivers a list of the existing databases on the server, to the command processor window.



```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
| vt_vpm             |
+--------------------+
4 rows in set (0.17 sec)

mysql> _
```

Figure 24: Query to Generate a List of Databases on the Server

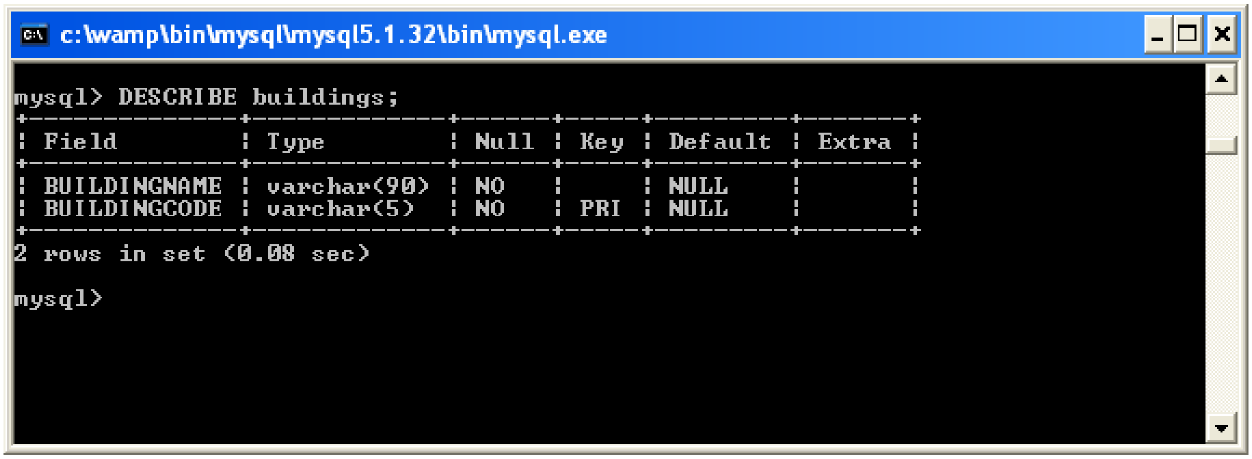
Once the list of databases is generated, the user will have to select a database to use it. This can be done by using the *USE* query as shown in the Figure 20. After selecting a database, the user can either select a table to know the properties of the table or use a *SHOW TABLES* query as shown in Figure 25. Here, the VT_VPM database is chosen to enlist the tables. As already mentioned, VT_VPM is the database that is being used in the development of the parking information application to assist the visitors to the Virginia Tech campus. It can be seen that, VT_VPM has a total of 5 tables in it and all the data that is relevant to the application is stored in these 5 tables.



```
mysql> USE VT_VPM
Database changed
mysql> SHOW TABLES;
+------------------+
| Tables_in_vt_vpm |
+------------------+
| buildings        |
| buildings_lots   |
| image            |
| occupancy        |
| parkinglots      |
+------------------+
5 rows in set (0.14 sec)

mysql> _
```

Figure 25: Query to Generate a List of Tables on a Database

To know the properties of any table, the user can select a table and use the *DESCRIBE* query to get a complete description of the table properties, as shown in the Figure 26. In this example, the table **buildings** is used to demonstrate the use of *DESCRIBE* query. When a *DESCRIBE* query is sent to the server, the server returns a table of information to the command processor window as shown in the Figure 26. In the displayed table, the column “Field” corresponds to the names of the columns in the table, **buildings**. It can be seen that there are two columns in this table and they are named BUILDINGNAME and BUILDINGCODE. The column, “Type”, corresponds to the data type employed by each column in the table, **buildings**. It can be seen that, BUILDINGNAME uses a VARCHAR data type with a maximum string length of 90 characters and BUILDINGCODE uses a VARCHAR data type with a maximum string length of 5 characters. The column, “Null”, gives the information about NULL values in columns of the table, **buildings**. Here, both the columns BUILDINGNAME and BUILDINGCODE do not support null values. The column, “Key”, gives the information about the PRIMARY KEY columns of the table, **buildings.** In this case, the column BUILDINGCODE acts as the PRIMARY KEY for the table. The column, “Default”, gives the information about the DEFAULT values assigned to columns of the table, **buildings.** In this example, both the columns, BUILDINGNAME and BUILDINGCODE do not use any DEFAULT values.



```
mysql> DESCRIBE buildings;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| BUILDINGNAME | varchar(90) | NO   |     | NULL    |       |
| BUILDINGCODE | varchar(5)  | NO   | PRI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.08 sec)

mysql>
```

Figure 26: Query to get a Description of Table Properties

3.6.6.1 Table Structure in the Database

The VT_VPM (Virginia Tech_Visitor Parking Management) database is made of a total of 5 tables and they are named

- Buildings
- ParkingLots
- Buildings_Lots
- Occupancy
- Image

A brief description of the contents of the columns in each table is presented in this section.

Buildings: In this table, a total of two columns are present and they are named

1. BuildingName: This column contains the names of the buildings on the Virginia Tech Blacksburg campus. This column houses a total of 124 building names.
2. BuildingCode: This column contains the code numbers that are assigned to the buildings on the campus. The code that is assigned to each building is taken from the Virginia Tech campus map.

ParkingLots: This table contains a total of sixteen columns and they are named

1. Parking_Lot_Name: This column contains the names of the parking lots on the Virginia Tech Campus.
2. Parking_Lot_Code: This column contains the code numbers assigned to the on-campus parking lots. The code numbers applied to the parking lots are taken from the data given by the Virginia Tech Parking Services
3. FS: This column contains the number of parking spaces allocated for Faculty/Staff
4. TA: This column contains the number of parking spaces allocated for Teaching Assistants
5. COM: This column contains the number of parking spaces allocated for Student Commuters
6. RES: This column contains the number of parking spaces allocated for Student Residents
7. FS_COM: This column contains the number of parking spaces allocated for Faculty/Staff Commuters
8. Visitor: This column contains the number of parking spaces allocated for Visitors
9. CP: This column contains the number of parking spaces allocated for carpool
10. MC: This column contains the number of parking spaces allocated for Motorcycle
11. Timed: This column contains the number of timed parking spaces
12. GEN: This column contains the number of parking spaces allocated Generally
13. SV: This column contains the number of parking spaces allocated for Service Vehicles
14. HC: This column contains the number of parking spaces allocated for Handicapped people
15. Meter: This column contains the number of Metered parking spaces
16. Total: This column contains the total number of parking spaces in a parking lot

Buildings_lots: This table contains a total of 3 columns. In this table, a combination of buildings and parking lots is made by choosing all the parking lots near a building. Thus, in this table, the combination of the parking lot code and building code creates a primary key. The columns in this building are named as follows

1. Building_Code: This column contains the code numbers that are assigned to the buildings on the Virginia tech campus. Since, there may be more than one parking lot available near a building the building code has to be repeated for all the parking lots near it.
2. Parking_Lot_Code: This column contains the code numbers assigned to the on-campus parking lots. As with the case of 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 column contains the approximate distance in feet from a parking lot to a building.

Occupancy: This table contains a total of 5 columns and contains information regarding the number of empty spaces in all the parking lots in a given month on a given day and at a specific time. Thus, in this table, 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 column 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 column, names of the months in a year are stored. The names of the 12 months are used in this column, but these names are repeated for every different combination of parking lot code, day and time.
3. DAY: This column houses the names of the days in a week. Though there are only 5 working days in a week, the names are repeated for every different combination of parking lot code, month and time.
4. TIME: This column 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 till 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 column contains the number of empty spaces in all the parking lots for all the working times in a day, for all the working days in all the months of a year.

IMAGES: This table contains information regarding the path of an image file for each building. Though, there are ways to store an image file directly onto the database instead of storing the file path to the image file, the latter option is chosen. Storing the image directly onto the database uses a lot of space on the server and when the image is called, a lot of data has to be transferred from the server to the local system and this might slow the performance of the interactive system. Furthermore if the images are to be changed in the future, they have to be deleted, modified and re-uploaded onto the database. But if the image location is stored instead of the image, there is no need worry about the database as long as the image path remains the same. So, in this table the columns are named as follows

1. BLDCODE: This column 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 column contains the information regarding the location of the images on the file system. As already mentioned, the path of a file in windows system must use forward slashes instead of the backward slashes for the MySQL server to read the file path correctly. However, back slashes can also be used in the file path instead of the forward slashes but they have to be doubled at every place in the file path. For example, if the file path *C:\wamp\bin\apache\apache2.2.8\htdocs\Bldgs_lots* has to be uploaded into a mysql table then either the path *C:/wamp/bin/apache/apache2.2.8/htdocs/Bldgs_lots* can be used or the path *C:\\wamp\\bin\\apache\\apache2.2.8\\htdocs\\Bldgs_lots* can be used. For this table, the second format for file path is used.

A table summarizing the properties of each table in the database is shown below.

Table Name	Field	Type	Null	Key	Default	Extra
Buildings	BuildingName	Varchar(60)	NO		NULL	
	BuildingCode	Varchar(5)	NO	PRI	NULL	
ParkingLots	Parking_Lot_Name	Varchar(50)	NO		NULL	
	Parking_Lot_Code	Varchar(5)	NO	PRI	NULL	
	FS	Int(4)				
	TS	Int(4)				
	COM	Int(4)				
	RES	Int(4)				
	FS_COM	Int(4)				
	Visitor	Int(4)				
	CP	Int(4)				
	MC	Int(4)				
	TIMED	Int(4)				
	GEN	Int(4)				
	SV	Int(4)				
	HC	Int(4)				
	Meter	Int(4)				
	Total	Int(4)				
Buildings_Lots	Building_Code	Varchar(5)	NO	PRI	NULL	
	Parking_Lot_Code	Varchar(5)	NO	PRI	NULL	
	Distance	Varchar(5)	NO		NULL	
Occupancy	PARKING_LOT	Varchar(5)	NO	PRI	NULL	
	MONTH	Varchar(9)	NO	PRI	NULL	
	DAY	Varchar(9)	NO	PRI	NULL	
	Time	Varchar(5)	NO	PRI	NULL	
	EMPTY_SPACES	Varchar(3)	YES		NULL	
Image	Bldcode	Varchar(5)	NO	PRI		
	Location	Varchar(100)	Yes			

Table 8: MySQL Database Properties

The relationships between the different tables of the database are shown in the Figure 27.

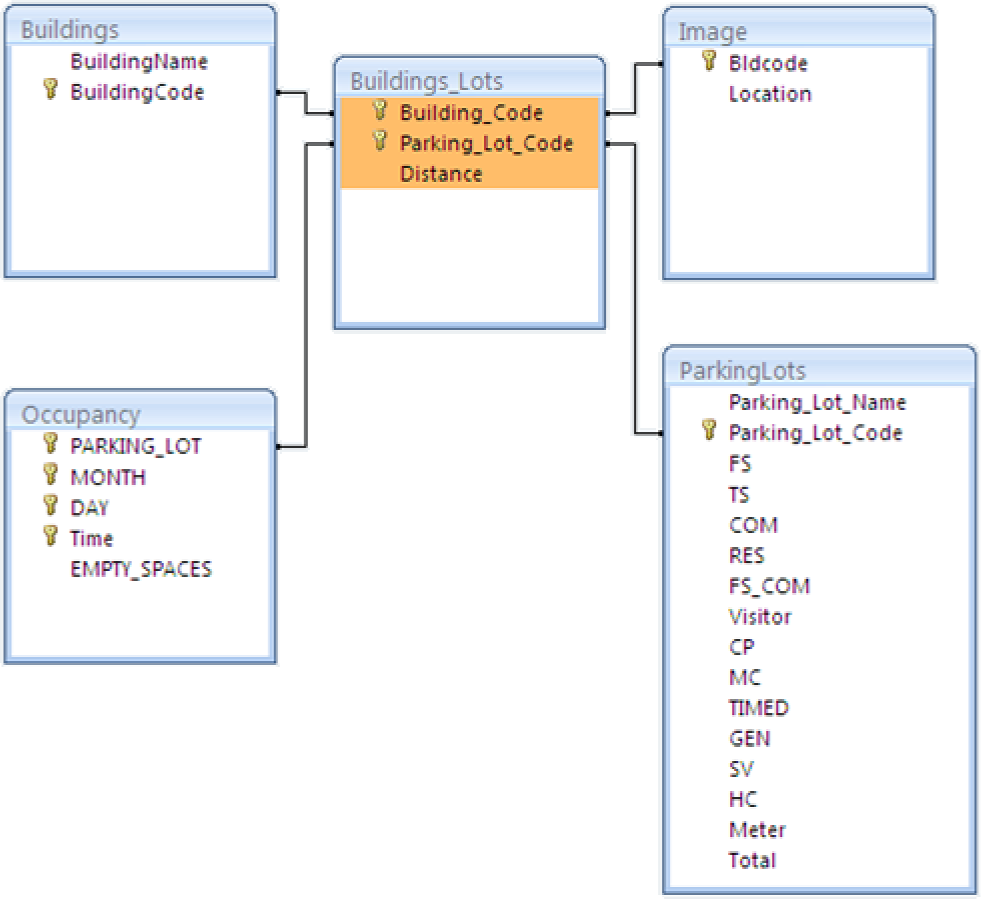


Figure 27: Relationships between the tables in the database

3.7 PHP

PHP which stands for PHP: Hypertext Preprocessor is a scripting language used to create dynamic web pages. PHP was originally called *Personal Homepage Tools* and was created by Rasmus Lerdorf in 1995. PHP is a free software and is released under the PHP license instead of the GNU General Public License due to restrictions on the usage of the term PHP [7]. PHP is a very popular scripting language in use and according to Netcraft, approximately 20 million domains have PHP installed on them.

PHP has a syntax which is somewhat similar to that of "C" language but unlike "C", there is no need to include the libraries for PHP to work. Although it is always a good practice to create, typecast and destroy variables while using a programming language, PHP works just fine for the people who do not follow this practice.PHP is a loosely typed language which means, there is no need to create or destroy a variable. It can create variables on the fly as it comes across them in the script [10].

3.7.1 Server-Side Scripting

Static web pages are web pages that are created using HTML language. As the name suggests, in a static webpage, the information does not change or the information is static. Even though the web page is accessed by different people at different times, the information that is displayed remains the same. A dynamic web page on the other hand provides interactive experience to the user. The content displayed on a web page can change in response to different contexts or conditions [11]. A web page can behave as a dynamic web page by using either client-side scripting or server-side scripting. In a client-side scripting, a specific web page can become dynamic by responding to mouse or keyboard actions. In a server-side scripting the web page becomes dynamic as a result of the actions performed by the server. The server executes the script in a file (like an html file) before returning it to the browser. PHP, used in the development of this application is a server-side scripting language.

Server-side scripting is generally used to connect web pages to the servers at the back end [8]. Most often, the servers at the back end are databases. The server side scripting enables a two-way communication as follows:

- **Client to server:** The user entered information is sent to the server so that necessary action can be taken.
- **Server to Client:** After processing the user entered information, the server sends the requested information to the front end.

3.7.2 PHP-Data Mover

PHP, being a server side scripting language, facilitates the two-way communication. It enables the user to send information from the local computer to the server and then collects information that is sent from the server to display in the web browser of the local computer. In this fashion, PHP performs the function of a data mover.

3.7.3 PHP Syntax

In a PHP file, the opening tag generally is *<?PHP* and the closing tag is *?>.* But, this does not mean that these are the default opening and closing tags in a file. These tags must bind the PHP statements or the PHP scripting block in a file. Each code line written in the PHP scripting block must be ended with a semicolon. A semicolon acts as a separator and distinguishes between one set of instructions and another. Typically, a PHP file is saved using a .php file extension. To print text in PHP, two statements, *echo* and *print* are commonly used. In the process of development of this application, *echo* is chosen to print the data wherever necessary. An example of a PHP file printing some text in the browser is shown below. For the text (or data) to be displayed in the browser, it has to be enclosed in double quotes.

```
<?PHP
echo "Hello world!";
?>
```

Comments can be added to a PHP file to facilitate the users of the program. A multi line comment can be added to PHP by starting and ending the comment with the character pairs */** and **/* respectively as shown below.

```
/* This is
a multi line
comment */
```

A single line comment can be added to PHP by starting the comment with the characters # or //.

3.7.4 Embedding PHP into HTML

This is one of the greatest advantages of the PHP language. PHP can be embedded into an HTML page. The web server recognizes that there are some PHP lines embedded into the file by looking at the .php extension. The web server does not process every line in the PHP file. It processes only those lines of code written in the PHP scripting block (enclosed by *<?PHP* and *?>*) by switching to PHP mode. The code lines that are not enclosed within the PHP tags are sent by the web server to the browser and the browser processes those lines. Thus, the HTML code will be processed by the browser and the PHP code will be processed by the web server. The code shown below provides an example of PHP's capability to be embedded in html.

```
<HTML>
<HEAD>
<TITLE>PHP Embedded HTML</TITLE>
</HEAD>
<BODY>
<P>This is an HTML line
<br>
<?PHP
echo "This is a PHP line";
?>
```

The output to the above code will be

```
This is an HTML Line
This is a PHP line
```

The first line in the output is processed by the browser and the second line is processed by the web server by switching to PHP mode. Once the web server hits the *?>* tag, it recognizes that the PHP scripting block has ended and returns to the normal HTML mode. This process repeats itself for all the PHP scripting blocks found in a file. Instead of placing PHP tags at the start and end of the PHP file, it would be beneficial to create several PHP scripting blocks, as this procedure will limit only the PHP code to be executed by the browser.

3.7.5 PHP Variables

Variable are used to hold the information or data during the execution of a PHP program. By default, any variable name should start with a dollar sign ($). This enables PHP to identify it as a variable. Variable names can be of any length and can use letters, numbers and underscores. Variable names can begin with an underscore or a letter but cannot begin with a number. Variable names are case sensitive. For example, $Emptyspaces is different from $emptyspaces. Values can be assigned to the variables multiple times and the value in a variable will be the value of its most recent assignment. Furthermore, there is no need to define the data type for a variable.

The variables in PHP can store numbers or strings of characters. Values can be stored in a variable using the equal (=) sign. A string of characters has to be enclosed in double quotes to assign it to a variable. An example of assigning variables with values is shown below

```
$id_no=34;
$Time=18:15;
$name=”John”;
```

3.7.6 Quotes in PHP

Single quotes and double quotes behave quite different from each other in PHP. A variable enclosed in double quotes is interpreted as a variable but if the same variable is enclosed within single quotes it is simply interpreted as the variable name. This can be demonstrated by the following statements:

```
$id=34;
echo “John’s id is $id”;
echo ‘John\’s id is $id’;
```

The output to the above two statements will be

```
John’s id is 34
John’s id is $id
```

The reason to use a backslash (\) in the statement enclosed in single quotes is to escape the single quote in the text John’s. If the backslash is not used a syntax error will be the result of the execution of this code.

To start a new line in the output, the newline character *\n* should be used at every point in the string where a new line is desired. Furthermore, the entire string must be enclosed in double quotes. If the string is enclosed in single quotes, the new line character will be printed as it is.

One more interesting aspect with respect to quotes is, the type of quotes that enclose the complete string determine the treatment of the variables and special characters, even if other sets of quotes are inside the string. This can be well demonstrated by the following statements:

```
$id=34;
echo “ The id given to John is ‘$id’.”;
echo ‘The id given to John is “$id”.’;
```

The output for the above statements will be

```
The id given to John is ‘34’.
The id given to John is “$id”.
```

It can be observed that in the second statement which is enclosed by single quotes the variable is treated as variable name even though it is enclosed by double quotes.

To concatenate (join) two strings in PHP a dot (.) is used. This is demonstrated by the following statements:

```
$string1=’This is’;
$string2=’concatenation’;
$stringct=$string1.” “.$string2;
Echo $stringct;
```

The empty space enclosed in double quotes is used in the concatenation so as to introduce an empty space between the words is and concatenation. The output for the above statements will be

```
This is concatenation
```

3.7.7 PHP Date and Time

The dates and times form important elements in the development of this parking application since the application has to search for the estimated number of empty spaces in the parking lots depending upon the time and day on which the request is made. PHP cannot recognize date and time as we normally would. Instead, PHP recognizes date and time in a format called Timestamp which stands for an integer representing the number of seconds elapsed from January 1, 1970, 00:00:00 GMT (Greenwich Mean Time). Since we cannot understand the Timestamp format, PHP converts the time stamp into a format which we can understand. Thus, PHP converts date and time into Timestamp format so that the computer can understand the time and then convert the date and time in Timestamp format to the general format to display to the user.

As already mentioned this application requires date and time to be setup on the local system to run queries with reference to those dates and times. The following steps are needed to setup a default time zone for the local system:

- Open the php.ini (php initialization) file in a text editor
- Go to the section under the header [Date]
- Look at the setting date.timezone =
- If the above setting begins with a semicolon, remove the semicolon
- Look for the definitions of time zones at http://www.php.net/manual/en/timezones.php and find a time zone that fits the application needs. Copy the time zone code and paste it after the equal sign in date.timezone setting. For this application the eastern time zone which has a time zone code America/New_York is chosen

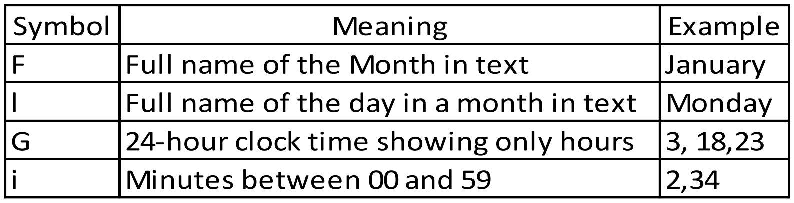
Once the time zone is setup, the application gets synchronized with that time zone and works irrespective of the time on the local system. In PHP the function that is used to convert a Timestamp into a specified format is *date.* An example of the *date* function is shown below

$today=date (“m/d/y”);

If the date today is April 20, 2009, the above statement will return

04/20/2009

In the development of this application, full names of month and day were loaded into the database. So, to run a query citing the month and day, it would be necessary to first create some variables which store the full names of current months and days. Time of the day is one more important search criteria that is required in this application. The 24-hour clock time in the hour:minute format is used as the search criteria in this application. Fortunately, PHP can return all these types of time and date formats. Some of the important time and date formats used in this application are shown in the Table 9.



Symbol	Meaning	Example
F	Full name of the Month in text	January
l	Full name of the day in a month in text	Monday
G	24-hour clock time showing only hours	3, 18,23
i	Minutes between 00 and 59	2,34

Table 9: Time and Date Format Symbols

The following set of statements demonstrates the use of the date function in this application.

```
$Today=date("l");
$Month=date("F");
$hour=date("G");
$minute=date("i");
echo " The current day is $Today";
echo " The current month is $Month";
echo " The current time in hours is $hour";
echo " The current time in minutes is$minute";
echo " The current time is $hour:$minute";
```

For example, if today is April 30th, 2009 and the current time is 15:05 the output for the above statements will be

```
 The current day is Thursday
The current month is April
The current time in hours is 15
The current time is minutes is 05
The current time is 15:05
```

3.7.8 Conditional Statements

The time stored in the database is not a continuous variable. As such, the time in the local time zone will not always match the times stored in the database. Therefore it would be necessary to convert the local time to one of the times stored in the database before running a search query. Each hour is divided into 4 intervals of 15 minutes each and stored in the database. So, the time for every hour from 8:00 till 18:00 is stored at 00, 15, 30 and 45 minutes only. Thus, in the time slot between 9:00 and 10:00, only the times 9:00, 9:15, 9:30 and 9:45 are stored in the database. But the local time for the system could be anything between 9:00 and 9:59 for the same time slot. To avoid this problem, conditional statements are used to convert the minutes in the local time to one of the 4 minute values stored in the database.

To accomplish this task, *If* statements are used in this application. *If* statement sets up a condition and tests it. If the condition is true one block of statements is executed and if it is false a different block of statements is executed. The general format of an *If* statement looks like

```
If (Condition…….)
        {
                set of statements
        }
elseif (Condition…….)
        {
                Set of statements
        }
Else
        {
                Set of statements
        }
```

If the condition is true at the *If* section, then the set of statements after the *If* section are executed and the program moves to the next block of statements skipping over all the *elseif* and *else* sections. If the condition is false at the *If* section, the set of statements following the *If* section are not executed and the program moves to the check the *elseif* or *else* conditions, if any, or moves to the next block of statements. The *elseif* section is optional and may not be present in an *if* statement. If the condition at the *elseif* section is true, then the program will execute the block of statements after the *elseif* section and will skip any additional *elseif* sections or *else* section. If the condition at the *elseif* section is false, then the set of statements following the *elseif* section is not executed and the program moves to the next *elseif* section or *else* section, if any, or moves to execute the next block of statements. The *else* section is also an optional section in an *if* statement. There can only be one *else* section present in an *if* statement. This section does not test any condition and directly executes the block of statements following the *else* section.

3.7.9 SQL Queries in PHP

PHP has some strong built in features to communicate with MySQL. PHP uses some built functions to communicate with MySQL server. It is quite simple to send queries to MySQL server if the appropriate PHP functions are known. There is no need to know the details of the communicating with MySQL as PHP handles the details.

The SQL query that is generally used to retrieve information from a database is the *SELECT* query. The general syntax of a *SELECT* query is

SELECT coulmnname1, columnname2, columnname3……FROM tablename1, tablename2, tablename3

In the above query, columnname1, columnname2 and columnname3 represent the names of the columns in different tables from which the information has to be retrieved and tablename1, tablename2 and tablename3 represent the names of all the tables from which the data has to be retrieved. The above query returns all the data in all the columns listed in the above query. But generally it is required to retrieve data with specific characteristics from any database. To filter the data, it is required to add another SQL word to the query. For this application, the word WHERE is added to the above query as shown below

SELECT coulmnname1, columnname2, columnname3……FROM tablename1, tablename2, tablename3 WHERE expression1 AND|OR expression2 AND|OR expression3

In the above query, the data in the columns listed will be retrieved only if it possesses the characteristics listed in the expression1 AND|OR expression AND|OR expression3. In the above query either of AND or OR should be used between each expression. If AND is used between two expressions, then the data which satisfies both the expressions will be chosen. If OR is used between two expressions, then the data which satisfies either of the expressions will be chosen.

When there are multiple tables from which data has to be retrieved it would generally be required to relate the tables with one another using the primary keys or the columns which hold similar data across different tables. Suppose there are two tables named table1 and table2. Let the columns in table1 be named col1, col2, col3 and columns in the table2 be named colA, colB, colC. If the data in col1 and colB is same then both the tables can be related using the following expression

table1.col1=table2.colB

In the above expression, the table names are used to remove the ambiguity in case the column names in different tables have same names and also to indicate that the column after dot belongs to the table before dot.

3.7.10 Code

The MySQL version used in the WAMPSERVER is an improved version and the MySQL functions used in the PHP code will be of the form *mysqli_function (value. Value,…)* where the *i* in *mysqli* stands for improved. This application needs the input from users to send a query to the database and display the results to the user finally. So, to assist the user to interact with the database, a HTML form is created with drop down lists for criteria, number of records to be fetched and building names. Some of the users may prefer parking lots which are nearer to the destinations even though there is a risk of not finding empty spaces in it. These users are clearly guided by the criteria of distance. Some other users may prefer parking lots with more number of empty spaces even though they are not located close enough to the destination. These users are guided by the criteria of empty spaces. The criteria form allows the user to input their criteria and the results are printed out according to the preference of the user. There are total of 124 buildings on the Virginia Tech campus and about 93 parking lots to cater for these buildings. 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 fetched from the database using the records form. The minimum number of records that can be fetched is 5 and the maximum number of records that can be fetched is 93. Once the user selects the criteria, number of records to be fetched and the building name and clicks the submit button, the user input data will be sent to a particular file for further action. The two methods that are used to submit a form are GET and POST. For this application, POST is used to submit the form as shown below

```
<form name= "Criteria_select" action="default.php" method="post">

<form name= "Records" action="default.php" method="post">

<form name= "building_select" action="default.php" method="post">
```

In the above statement, the form is named as building_select. The action directs the submitted form entry to the file *default.php*.

Now, once the form value is sent to the file, it is necessary to collect the value and also to connect to the database and collect some more necessary information to run a query in the database. To communicate with the database, the first step would be to connect to the MySQL server. To connect to the server, the name of the computer onto which the database is stored is required along with the name of the MySQL user account and password to the account. The name of the database is also required, to connect to the database. The connection to the database can be established by using the following function

```
$cxn=mysqli_connect($host, $user, $password, $dbname) ;
```

In the above function the variable $host stores the address of the computer, onto which MySQL is installed. In WAMPSERVER, the variable $host can take LOCALHOST as its value. The variables $user and $password take the values assigned by the user during the installation of WAMPSERVER. The variable $dbname should be assigned with the name of the database. For this application, the name of the database is VT_VPM.

It can be checked if any error is encountered while establishing a connection to the database using the following *If* statement

```
if (!$cxn=mysqli_connect($host, $user, $password, $dbname))
        {
                $error=mysqli_error($cxn);
                echo "$error";
                die();
        }
```

In the above statement the symbol “!” in front of the variable $cxn represents not equal to. So, if the variable does not equal the *mysqli_connect* function it leads to an error. The error information is stored to the variable *$error* and it is printed using the *echo* function as shown above.

Now, the form entries submitted by the user have to be retrieved into the file. For this purpose, the function shown below is used in which the values (criteria, number of records and building name here) are stored into the variables *$criteria*, *$records* and *$build* respectively

```
$criteria=$_POST["criteria"];

$records=$_POST["records"];

$build=$_POST["name"];
```

The other necessary information required to generate and run a query is information about the current day and current month. The information about current day and month can be obtained using the *date* function. Furthermore, to help the users come up with an approximate walking time from the parking lot to the building, to speeds of 3 ft/sec and 5 ft/sec are used. All these values are assigned to the variables as shown below

```
$Today=date("l");
$Month=date("F");
$low_speed=3;
$high_speed=5;
```

In this application, Conditional Statements are used to convert the local system time to match the times stored in database as shown in the following statements:

```
$hour=date ("G");
$minute=date ("i");
If ($minute>=00 && $minute<=14)
        {
                $minute=00;
        }
elseif ($minute>=15 && $minute<=29)
        {
                $minute=15;
        }
elseif ($minute>=30 && $minute<=44)
```

```
        {
                $minute=30;
        }
else
        {
                $minute=45;
        }
$times="$hour:$minute";
```

In the above statements, the system time in hours and minutes is recorded individually. Then the minutes time is changed to one of the 4 values used in the database using the conditional statements as shown below

- 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 named *$times* to be able to run a query in the database. Using all the variables shown above a query is run to get 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 used to retrieve and store the required data into another variable is shown below

```
$sql="SELECT buildingname, parking_lot_name, (FS+COM+RES+FS_COM+VISITOR+GEN) AS total_spaces, (TIMED+METER) AS Metered, empty_spaces, distance, round(distance/($low_speed*60),1) AS high_time, round(distance/($high_speed*60),1) AS low_time FROM buildings, buildings_lots, parkinglots, occupancy where (buildings.buildingcode=buildings_lots.building_code AND parkinglots.parking_lot_code=buildings_lots.parking_lot_code AND
```

parkinglots.parking_lot_code=occupancy.parking_lot) AND buildingname='$build' AND month='$Month' AND day='$Today' AND Time='$times' ORDER BY distance";

$data = mysqli_query($cxn,$sql);

In the above SQL queries, the variable $sql stores the SQL query which retrieves the required data. It can be seen that the data is being selected from columns named buildingname, parking_lot_name, empty_spaces and distance. Further, data can also be modified and selected as a new column shown as *round(distance/($low_speed*60),1) AS low_time.* In this statement, the value *distance/($low_speed*60)* is rounded till 1 decimal using the *round* function. Then the value is retrieved into a new column named *low_time.* Similarly, another set values are retrieved into a newly created column named *high_time.* The data is being selected from the tables named *buildings, buildings_lots, parkinglot* and *occupancy.* The complete data being selected into the columns is of no interest to the user. The data pertaining to the building the visitor is willing to visit on the specific month, day and at the specific time is of interest to him/her. So, using the *WHERE* clause, the tables are related to each other and the search criteria is fed into the query as shown above. Here the data will be sorted in an ascending order of distance as that is the criteria required by the user. Finally, the data is stored to a variable named *$data* using the *mysqli_query* as shown above.

To check for any errors that may have come up during the data retrieval process, the following statements can be used

```
if (!$data=mysqli_query($cxn,$sql))
        {
                $error=mysqli_error($cxn);
                echo "$error";
                die();
        }
```

The above *If* statement tests the case of inequality between the variable $data and the data retrieval query and if inequality turns out to be true then the error information is printed to the browser.

If no errors are encountered till now, the next step would be to prepare a table for the data to be displayed in. This could be achieved by using the following statements

```
echo "<table border='1' cellspacing='5' cellpadding='2'>";
echo "<tr>\n
                <th>Building</th>\n
                <th>Parking Lot</th>\n
                <th>Parking Lot Capacity</th>\n
                <th>Metered Spaces</th>\n
                <th>Estimated Empty Spaces</th>\n
                <th>Distance (Feet)</th>\n
                <th>Estimated walking time   </th>\n
        </tr>\n";
```

The above statements are nothing but the statements used in HTML to create a table with the statements enclosed in double quotes. The function *echo* is used to print the table into the browser. The table used here has a table border of 1 with a cellspacing of 5 and a cellpading of 2. The header row consists of the names of the columns and the columns are named *Building, Parking Lot, Parking Lot Capacity, Metered Spaces, Estimated Empty Spaces, Distance (Feet)* and *Estimated walking time.*

After printing the table for displaying the data, the next step is to fill the table with the data retrieved from the database. To fill the table with the retrieved data from database a *WHILE* function is used as shown below. The data is fetched from the variable *$data* row by row using the function *mysqli_fetch_row($data)*, until the *WHILE* loop breaks out. For this purpose, the number of rows retrieved into the variable *$data* can be fetched using the function *mysqli_num_rows($data).* The coding for retrieving and printing the records from the variable *$data* onto the web browser is shown below

```
$nrows = mysqli_num_rows($data);
While ($i<$records)
        {
$row=mysqli_fetch_row($data);
        $building = $row[0];
        $parking_lot = $row[1];
        $total_spaces = $row[2];
        $Metered = $row[3];
        $Number_of_Empty_Spaces = $row[4];
        $Distance = $row[5];
        $time_h = $row[6];
        $time_l=$row[7];
echo "<tr>\n
                <td>$building</td>\n
                <td>$parking_lot</td>\n
                <td>$total_spaces</td>\n
                <td>$Metered</td>\n
                <td>$Number_of_Empty_Spaces</td>\n
                <td>$Distance</td>\n
                <td>$time_l - $time_h mins</td>\n
</tr>\n";
$i++;
If ( $i==$nrows)
{
break;
}
else
{
Continue;
}
}
```

As shown in the above statements, to fetch number of rows stored into the variable *$data* the function *mysqli_num_rows($data)* is used. This value is stored into a variable *$nrows.* Another variable *$i=0* is created and this variable is used as a counter. First the *WHILE* condition is tested using the function *While ($i<$records)*. If the condition is true that is to say if the variable *$i* is less than the variable *$records* the code within the loop brackets is executed. One row of data is fetched from the variable using the function *mysqli_fetch_row($data).* The extracted line of data is fed to a variable named *$row*. The values in the columns can be accessed using the variable format *$row[x].* The x in the square represents the position of the column from the extreme left. In arrays, the first column number is zero. As such, the values in the array are transferred into individual variables as shown below

```
   $building = $row[0];
        $parking_lot = $row[1];
        $total_spaces = $row[2];
        $Metered = $row[3];
        $Number_of_Empty_Spaces = $row[4];
        $Distance = $row[5];
        $time_h = $row[6];
        $time_l=$row[7];
```

After the transfer of the values into the variables, these variables are printed out using the *echo* function as shown below

```
echo "<tr>\n
                <td>$building</td>\n
                <td>$parking_lot</td>\n
                <td>$total_spaces</td>\n
                <td>$Metered</td>\n
                <td>$Number_of_Empty_Spaces</td>\n
                <td>$Distance</td>\n
                <td>$time_l - $time_h mins</td>\n
</tr>\n";
```

After this step, the counter variable *$i* is incremented using the function *$i++.* Then another condition is tested as to ascertain that the counter value is less than the total number of rows in the variable *$data.* If this test is not conducted then the loop continues till *WHILE* loop fails which might fill the table with empty fields. So, in order to cut off the printing of values once all the rows in the variable *$data* are printed the *IF* condition is used. As long as the *If* condition is false, the *While* loop continues but once the *If* condition holds true the *While* loop breaks away and the next line of code outside the *While* loop is executed.

If the user submits empty spaces as the criteria, the sql query used to retrieve and store the required data into another variable is shown below

```
$sql="SELECT buildingname, parking_lot_name, (FS+COM+RES+FS_COM+VISITOR+GEN) AS
total_spaces, (TIMED+METER) AS Metered, empty_spaces, distance,
round(distance/($low_speed*60),1) AS high_time, round(distance/($high_speed*60),1) AS
low_time FROM buildings, buildings_lots, parkinglots, occupancy where
(buildings.buildingcode=buildings_lots.building_code AND
parkinglots.parking_lot_code=buildings_lots.parking_lot_code AND
parkinglots.parking_lot_code=occupancy.parking_lot) AND buildingname='$build' AND
month='$Month' AND day='$Today' AND Time='$times' ORDER BY empty_spaces DESC";
```

Here the results are ordered in descending order of the empty spaces in a parking lot. The parking lots with highest number of empty spaces will be shown at the top.

The next step in this application is to print the image of the building the user has intended to visit and the parking lots surrounding the building. To print images in PHP *header* statements are to be used. But *header* statements can only be used before sending any output to the browser. A lot of information has been sent to the browser till now and therefore a *header* statement cannot be used in this file. So, to print an image to the browser it becomes absolutely necessary to create another file and use a command in the first file to go to the second file when there is a need to print the image. An image can be created using a simple HTML image tag <IMG>. But, in a dynamic webpage the image can vary depending upon the user's criteria. So, to print an image in the browser, the following statement has to be used

```
echo '<img src="imgtest.php?id=' . $building . ' " border="5" alt="' . $build . '">';
```

In the above statement, *imgtest.php* refers to the PHP file in which the code to print an image is written. After the question mark (?), there is a variable named *id*. The name of the building, input by the user, will be assigned to this variable. Further, the border for the image is assigned a thickness of 5 and the image is assigned the name of the building chosen by the user. Once the program reads this statement, it will go to the PHP file named *imgtest.php* and will carry the name of the building in the variable *id*.

The next step is to get the building name from the variable *id* in the PHP file named *imgtest.php.* This is done using the following statement

```
$build=$_GET['id'];
```

The *GET* command is used to get the name assigned to the variable *id* and store it to the variable named *$build.* Then, this variable is used to get the location of the image from the table named *IMAGE.* For this purpose, it is required to connect to the database using the procedure explained earlier. Once the connection to the database is established, a *SELECT* query is run to retrieve the image location from the database. After retrieving the location of the image to the variable, it has to be fetched into another variable, so that the location can be read by the browser, using the *mysqli_fetch_array* function. The execution of these statements is shown below

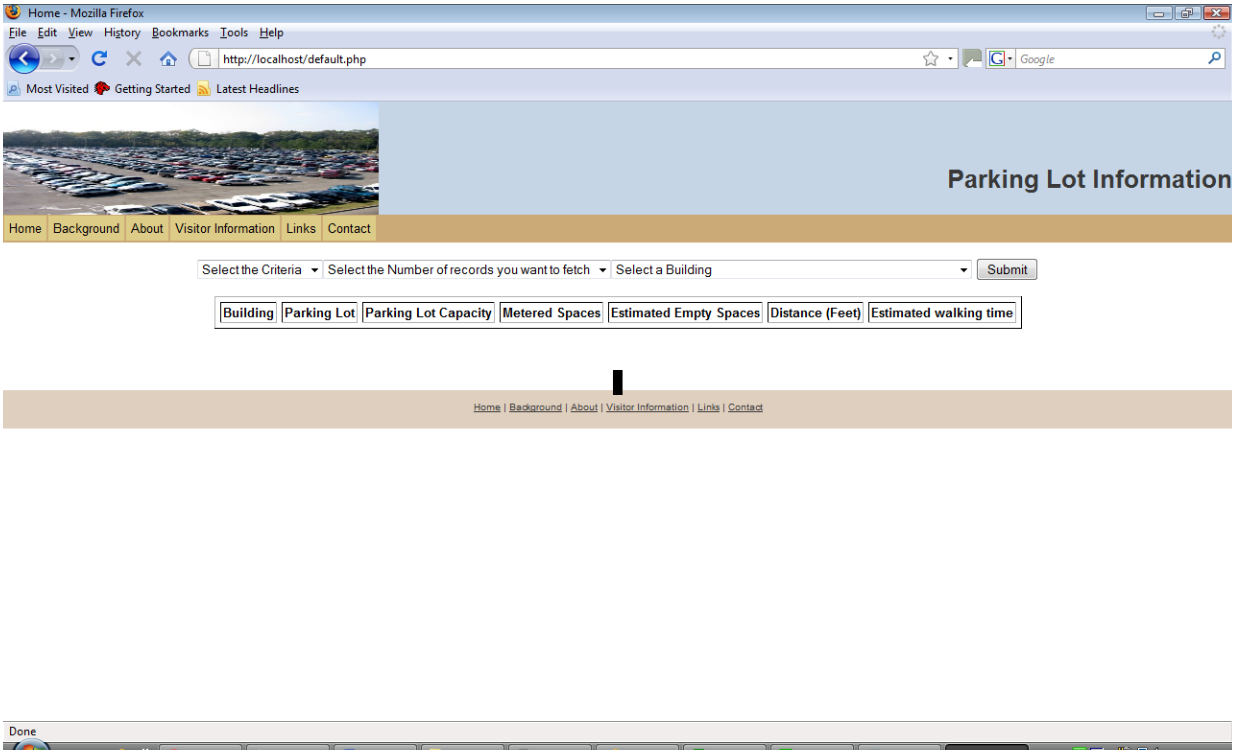
```
$sql="SELECT Location FROM buildings, Image where
(Buildings.BuildingCode=Image.BuildingCode) AND buildingname='$build' ";
$data = mysqli_query($cxn,$sql);
while ($row=mysqli_fetch_array($data))
        {
                extract($row);
$building = $row[0];
}
```

Once the location of the image is retrieved to the variable, the header function is used to print the image onto the browser and then the image is created from jpeg as shown in the following statements

```
header('Content-Type: image/jpeg');
imagejpeg(imagecreatefromjpeg("$building"));
```

3.8 Model Application

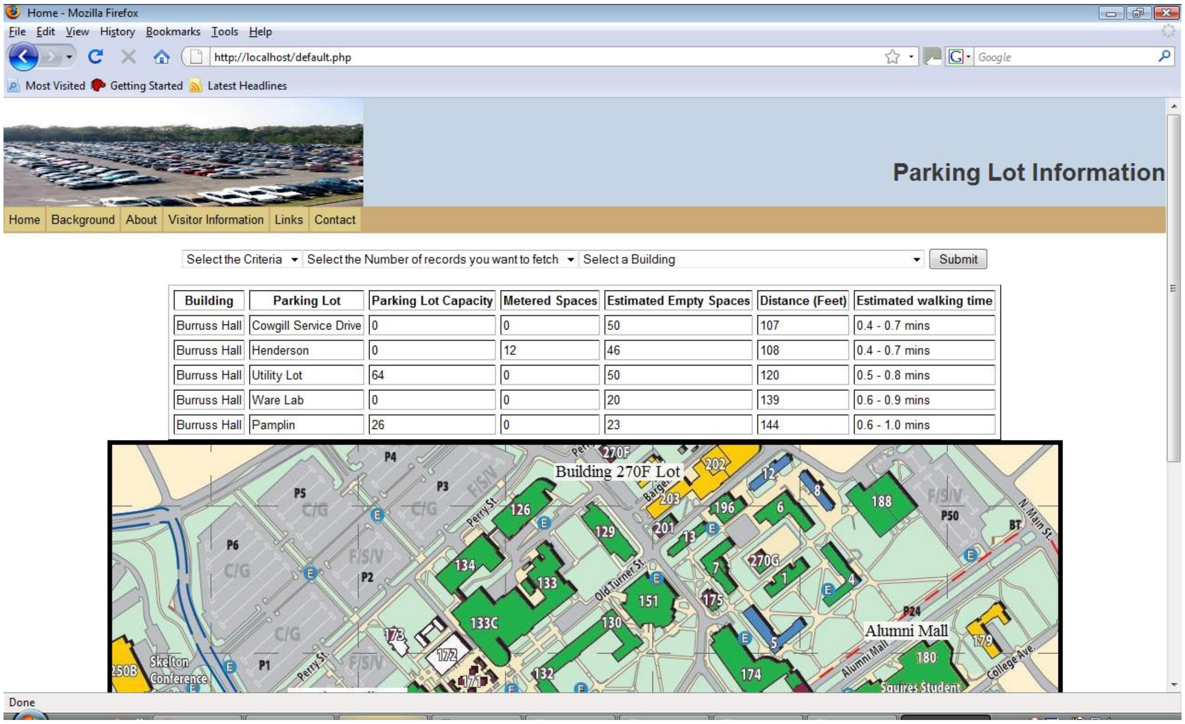
In this section some of the working configurations of the Web based interactive parking lot information system are provided. The parking lot information system web page displays an HTML form which collects the information from users as shown in Figure 28. The HTML drop down menus by default show the messages *Select the Criteria*, *Select the Number of records you want to fetch and select a Building* respectively from the left. There is also a *Submit* button used to submit the user input.


Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/default.php
Google
Most Visited Getting Started Latest Headlines
Parking Lot Information
Home Background About Visitor Information Links Contact
Select the Criteria
Select the Number of records you want to fetch
Select a Building
Submit
Home | Background | About | Visitor Information | Links | Contact
Done

Building	Parking Lot	Parking Lot Capacity	Metered Spaces	Estimated Empty Spaces	Distance (Feet)	Estimated walking time

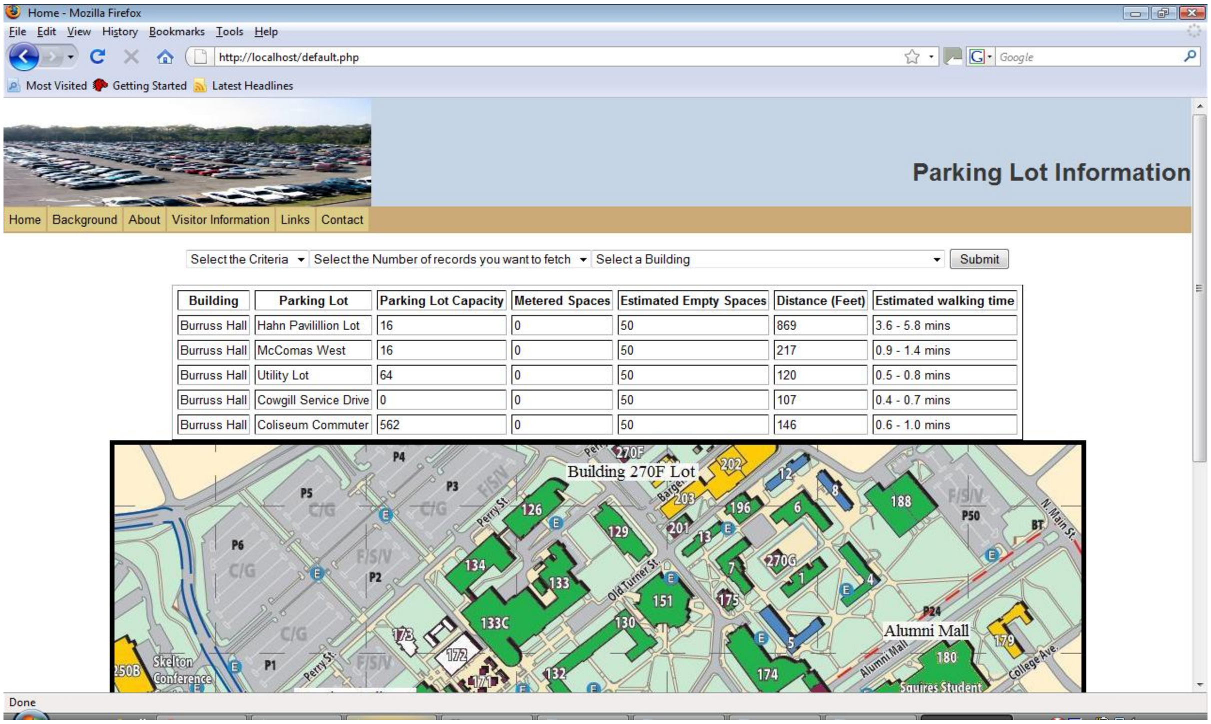
Figure 28: Parking Lot Information System

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 fetched as 5 and the building name as *Burruss Hall* as shown in Figure 29, we can see that 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 are chosen along with the same building name but the criteria is changed to Empty Spaces as shown in Figure 30 we see that the results are ordered by empty spaces in a descending order with the parking lot having highest number of empty spaces at the top. In both the cases, an image of the building with the closest parking lots around it is shown.


Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/default.php
Most Visited Getting Started Latest Headlines
Parking Lot Information
Home Background About Visitor Information Links Contact
Select the Criteria
Select the Number of records you want to fetch
Select a Building
Submit
Building 270F Lot
Alumni Mall
Done

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

Figure 29: Parking Lot Information with Distance as Criteria


Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/default.php
Most Visited Getting Started Latest Headlines
Parking Lot Information
Home Background About Visitor Information Links Contact
Select the Criteria
Select the Number of records you want to fetch
Select a Building
Submit
Building 270F Lot
Alumni Mall
Done

Building	Parking Lot	Parking Lot Capacity	Metered Spaces	Estimated Empty Spaces	Distance (Feet)	Estimated walking time
Burruss Hall	Hahn Pavilillion Lot	16	0	50	869	3.6 - 5.8 mins
Burruss Hall	McComas West	16	0	50	217	0.9 - 1.4 mins
Burruss Hall	Utility Lot	64	0	50	120	0.5 - 0.8 mins
Burruss Hall	Cowgill Service Drive	0	0	50	107	0.4 - 0.7 mins
Burruss Hall	Coliseum Commuter	562	0	50	146	0.6 - 1.0 mins

Figure 30: Parking Lot Information with Empty Spaces as the Criteria

3.9 Conclusions

Some of the important conclusions pertaining to the web based interactive parking information system are shown in this section

- The parking information system designed here is a prototype model which helps in the realization of the User Information Principle [2] and thus paves a way for better parking lot management
- The interactive system allows the user to provide specific information regarding his/her preferences and destination. This information is used by the system to generate a list of parking lots based on the user criteria
- Some important part of the data used in the application, like the estimated number of empty spaces in a parking lot, is randomly generated using MS Excel spreadsheets because of the lack of real time data. If real time data can be obtained and used by the parking services, this prototype model can be used to create a real time working model of the application
- The model in its current format fetches the image of the buildings and parking lots, stored on the local file system, to print the image onto the browser. However, a connection can be established between the GIS server of the campus and the web application to retrieve and print 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 side walks

CHAPTER FOUR: CONCLUSIONS AND FUTURE RESEARCH

4.1 Findings and Conclusions for the visitor and parking lot surveys

Visitors constitute an important component of any university's business. Due to the high occupancy rate of parking at Virginia Tech, it becomes quite a tedious task to find a parking space on campus. The objectives of this study were to determine the current visitor parking demand and supply, the adequacy of the visitor parking at Virginia Tech, the level of satisfaction with the visitor parking, and the willingness of visitors to pay for better parking facilities. These objectives were achieved by conducting visitor personal interviews, follow-up interviews, field surveys and analyzing historical data.

A total of 349 surveys were collected as part of the visitor center survey. Major findings of this survey are:

- Visiting prospective students, family member of a prospective student, family member of a current student, or a friend of VT students constitute the majority of the trips (51%). Apart from this, visitors for other reasons (bringing pets to the Vet school, football games, or sightseeing, etc.) constitute 24% of the total visitor respondents. Almost 20% of the respondents have scheduled meetings.
- About 20% of the people did not know the location of the offices they needed to visit and almost 28% of the visitors did not know the location of the most convenient parking lot to access their destination.
- 20% planned to stay between 1 to 2 hours and the same percentage expected to stay for 2 to 3 hours at their destination, while 30% of the people expected to park for more than 4 hours.
- When enquired about the rating of their previous experience with the parking services 73% of the respondents rated the services to be good or very good and 6% rated the service poorly with 28% of the people not responding at all.

- The majority of the visitors (55%) have showed their unwillingness to pay for designated visitor-only parking spaces, while only 33% responded positively saying that they would pay for designated visitor-only parking spaces. Of the people who have responded to pay for designated visitor-only spaces, the majority of them are willing to pay only $3 (50%) and almost 40% are willing to pay $5.

For the visitors that responded to the follow-up survey the following conclusions can be derived:

- Of the follow-up survey respondents, 25% visited VT as a prospective student or family member and 26% were either a current family member or a friend of a VT student. Together they constituted the major portion of the trips made. These percentages are very similar to the ones in the initial survey and thus demonstrate the consistency in both samples.
- For 79% of the respondents it only took less than 5 minutes to find a parking space. For 38% of the respondents the parking space was clearly available (1 to 2 min. to find a parking space). While for 7% it took more than 10 minutes to find a parking space.
- Approximately, 32% of the visitors had to visit at least 2 parking lots before finding a parking space.
- 68% of the respondents parked between 1 to 3 hours and only 9% parked for more than four hours (compared with the 30% obtained in the initial survey). This could be explained by the fact that in the initial survey the participant provided an estimate of their stay time whereas in the follow-up survey they provided their actual stay time. Only 23% of the respondents checked more than one parking lot to find a parking space.
- After their campus visit, 82% of the respondents rated the parking services at VT to be very good or good. This percentage is distributed evenly between very good and good. It is noted that 34% are willing to pay for parking spaces, out of which 46% were willing to pay $3 only while 40% were willing to pay $5 only.

The data collection efforts concentrated on five parking lots namely Cassell Coliseum, Drill Field, Litton Reaves, the Squires, and Perry Street. The data was gathered on Thursday and Friday because of the fact that the historical data accredits these two days to be representative of the average and the highest demand days of the week, respectively. With the exception for the Drill Field that has 5 visitor parking spaces, none of the other parking lots have dedicated visitor parking spaces.

For the Cassell Coliseum parking lot the total number of visitors at a given time ranged from 27 to 46 with an average of 36 visitor vehicles. The total number of visitors was 49 for Thursday and 57 for Friday. The short-term visitors account for almost 75% of all the visitors on both days. After the short-term visitors the Temporary visitors come second with an average of 13%. Other visitors including CRC, retiree, and long-term visitors make up for about 10% of the total visitors on both days. Most of the visitors stayed for more than 4 hours, followed by those who stayed between thirty minutes to an hour. The total number of violators at a given time ranged from 7 to 14. On average 9 violator vehicles were typically present in the parking lot at any given time on both days. The total number of violators was almost equal for both days standing at 23 and 22 on Thursday and Friday, respectively. The parking violators without a permit constituted the highest percentage of the violators on both days (68% to 83%). Most of the violators stayed between thirty minutes and an hour.

For the Drill Field parking lot, the total number of visitors at a given time ranged from 1 to 17 on Thursday and 17 to 33 on Friday. On average 9 and 23 visitors were present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 30 on Thursday and 77 on Friday. The parking lot operated well below the effective capacity but the occupancy levels were higher in the evenings. The long-term visitors constituted about 47% of all the visitors on Thursday. The short-term visitors accounted for 27% and the remaining 26% was shared by CRC and retiree visitors. On Friday, the short-term visitors constituted most of the visitors by amounting to 47% of the total visitors. The retirees and long-term visitors accounted for 25% and 21% of the total number of visitors, respectively. On Thursday most of the visitors stayed for an hour, followed by visitors who stayed for half an hour. The total number of violators at a given time ranged from 2 to 29 on both days. On average 12 violator vehicles were typically present in the parking lot at any given time on both days. The total number of violators was almost equal for both days standing at 60 and 54 on Thursday and Friday, respectively. The violators with a permit constituted 57% and the violators without a permit amounted to 43% of the total number of violators on Thursday. On Friday most of the violations did not have a permit. Most of the violators stayed for 30 minutes, followed by violators who stayed for an hour.

The Litton-Reaves parking lot operated below the effective capacity for the entire period during which the survey was conducted, however on Thursday the value was observed to have reached 80%. The total number of visitors at a given time ranged from 0 to 5 on Thursday and 0 to 1 on Friday. The total number of visitors was 5 on Thursday and 1 on Friday. The short-term visitors accounted for 80% of all the visitors on Thursday and the remaining 20% was accounted for by retirees. On Friday all the visitors were long-term visitors. The total number of violators was almost equal for both days standing at 3 and 2 on Thursday and Friday, respectively.

In the case of the Perry Street II parking lot the total number of visitors at a given time ranged from 9 to 13 on Thursday and 4 to 13 on Friday. On average 12 and 9 visitors are present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 16 on Thursday and 19 on Friday. The number of visitors increased with time on both days and peaked between 10:30 AM and 11:00 AM on both days. The parking lot operated below the effective capacity for the entire period during which the survey was conducted. The phase II study found that the parking lot operated beyond its capacity for Tuesday and Wednesday, reaching 100% occupancy. This could be considered as an indicator that Thursdays and Fridays are not one of the days with the highest demand for this specific parking lot. The CON type of visitors constituted approximately 70% of all the visitors followed by the long-term visitors with 13% and the short-term, BUS and CRC visitors with 18%. On Friday 26% of the visitors were short-term visitors followed by temporary and long-term visitors with 16% each. The remaining 32% was accounted for by the CRC, BUS and retiree visitors equally. On Thursday most of the visitors stayed for more than 4 hours. On Friday the frequencies for different durations of stay were almost identical to each other. The total number of parking violators at a given time ranged from 1 to 3 on Thursday and 0 to 2 on Friday.

The Squires parking lot operated above the effective capacity for the entire period during which the survey was conducted. The average occupancy of the parking lot for the entire period of time was about 92%. The total number of visitors at a given time ranged from 3 to 7 on Thursday and 0 to 16 on Friday. On average 6 and 7 visitors were present in the parking lot at any given time on Thursday and Friday, respectively. The total number of visitors was 12 on Thursday and 18 on Friday. Long-term visitors accounted for more than 50% of all the visitors, followed by short-term visitors with 34%, and the remaining 16% equally constituted retiree and CRC visitors. On Thursday most of the visitors stayed for less than 2 hours and on Friday most the visitors stayed for 30 minutes followed by the visitors who stayed for 1 hour and 3 hours. The total number of violators at a given time range from 2 to 7 on Thursday and 0 to 5 on Friday. The total number of violators is 9 on Thursday and 7 on Friday.

It could be concluded that very few people are willing to pay for parking spaces and the majority of those who are willing to pay consider a value of $3 to be a reasonable charge. The fact that very few people are willing to pay and that they selected the smallest amount, provides us with an indication of a good visitor parking system where the visitors do not see a reason to pay.

Due to the high number of visitors visiting the Cassell Coliseum parking lot and taking into account that the effective capacity is reached around 10:00 AM, it is recommended that a number of spaces be dedicated for visitors between 8:00 AM and 5:00 PM. Considering the number of parking spaces available for F/S and commuters, it is not expected that these users will notice any change even if some of the parking spaces were to be designated for visitors. Dedicating a reasonable number of spaces, in a visible way will allow the visitors to feel welcome and park more quickly. These parking spaces must be located near the entrances so that they can easily find them and also near the Stadium as it is one of the most frequently visited building.

Taking into account the small number of visitors of Litton Reaves and that it operates below the effective capacity, no specific action is recommended for this parking lot. Similarly, the Drill Field does not require any actions for improvement.

Dedicating parking spaces for visitors in Squires is not recommended due to the amount of visitors and also due to the fact that the parking lot operates above its effective capacity for the entire day. Visitors must be discouraged from using the Squires parking lot and must be advised to use the Coliseum Parking Lot in an effort to reduce their search time.

The number of violators is relatively high for most of the parking lots excluding Perry Street. It is recommended that Parking Services continue with the information campaign through the web pages and pamphlets in order to keep the costumers informed about the appropriate places they can use for parking in order to reduce the number of violations by those people who have a parking permit but park in an area not designated for that specific parking space. To reduce the number of type II violators, parking services must continue with an aggressive control policy that discourages such violators from using the parking lots.

It appears that in general the visitors were satisfied with the parking services on campus but there is an opportunity for improvement by providing more information on visitor parking locations and availability. In order to assist the visitors in finding a parking lot and space, a web based interactive system is proposed to be developed. This system will attempt to identify the optimum parking lot available (nearest or with a higher probability of finding a parking space) for a visitor depending upon his/her input criteria.

Finally, it is recommended that the university conduct a visitor survey on a continuous basis in an effort to improve service and its strategic parking management plan

4.2 Future Research

The parking application developed has a capability to search through the database to provide the user with an estimate of the approximate number of empty spaces and approximate walking time from the parking lot to the building. But, the application will be useful only if there is sufficient amount of supporting data. The dynamic nature of the application can be brought out only by providing the database with relevant data. For this purpose, the parking services at Virginia Tech must conduct parking lot surveys at regular intervals. They can install two cameras, one at the entry and the other at the exit of the lot, and can predict the number of empty spaces available in the parking lot by subtracting the difference between the cumulative number of arrivals and departures from the total number of empty spaces in the parking lots.

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 probabilistic approach.

The model in its current format fetches the image of the buildings and parking lots, stored on the local file system, to print the image onto the browser. However, a connection can be established between the GIS server of the campus and the web application to retrieve and print 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 and will also indicate any new developments on the campus such as the closure of a sidewalk near a building under construction.

Furthermore, the application is currently developed considering visitors as the main users of the system. But, it can also be used by the employees and students of Virginia Tech to come up with their parking plans by converting this into a real time working application. It would also be beneficial to create smart phone compatible software which can be downloaded onto the cell phone and use it on the move to come up with parking plans.

REFERENCES

1. ***The DPS Parking Division Office (2006) The Iowa State University Parking Division Manual.***

2. ***Todd Litman, Parking Management - Strategies, Evaluation and Planning. 2008.***

3. ***Parking Services at Virginia Tech. Visitor Parking. 2008; Available from: http://www.facilities.vt.edu/tcs/parking/visitors.asp.***

4. ***HSMM Inc., et al. (2002) VT Parking and Transportation Master Plan.***

5. ***University, I.S., Iowa State Law and Administrative Regulations. 2003.***

6. ***Janet Valade, PHP & MySQL For Dummies. 3rd edition ed. 2007: Wiley Publishing, Inc. 436.***

7. ***Wikipedia. Open Source Software. 2009 [cited 2009 April, 20th]; Available from: http://en.wikipedia.org/wiki/Open_source_software.***

8. ***Clark Morgan, Tim Converse, and Joyce Park, PHP5 and MySQL Bible. Third edition ed. April, 2004: Wiley Publishing, Inc. 1083.***

9. ***Sun Microsystems Inc. (2009) MySQL 3.23, 4.0, 4.1 Reference Manual.***

10. ***W. Jason Gilmore, Beginning PHP and MySQL. 3 edition ed. 2008: Apress. 1080.***

11. ***Wikipedia. Database. 2009 [cited 2009 4th April]; Available from: http://en.wikipedia.org/wiki/Database.***

Appendix A: Visitor Personal Survey

Code XXXX

The Office of Transportation at Virginia Tech is conducting a visitor parking survey to determine the current visitor parking demand and how we can better serve the visitor's parking needs. Would you be willing to complete this short survey?

1. Are you at least 18 years old?

□ Yes □ No

If your answer is no, please stop here and discard the survey.

2 The reason for your visit is:

□ Prospective student or family member
□ Current family member or friend of a student
□ Academic Activities
□ Local resident using VT facilities or services
□ Other____

3 If your answer is Academic Activities, do you already have a scheduled meeting?

□ Yes □ No

4. Which University office(s) are you planning to visit?

____

5. In what building(s) are they located?

____

6. Do you know the location of the office(s) that you need to visit?

□ Yes □ No

7. Do you know the most convenient parking lot to access your destination?

□ Yes □ No

8. How long are you planning to stay at your destination?

□ Less than 1 hour

□ 1 hour to 2 hours
□ 2 hours to 3 hours
□ 4 hours
□ More than 4 hours

9. How often do you visit Virginia Tech Blacksburg campus? Please specify number of times

□ Only once ______
□ Per a year ______
□ Per semester ______
□ Per month ______
□ Per week ______
□ Other (please specify)______

10. How do you rate your previous parking experiences?

□ Very good □ Good □ Fair □ Poor

a. If poor, please explain why.____

11. Would you be willing to pay for designated visitor-only parking spaces?

□ Yes □ No

12. If your answer is Yes, what is the most you would be willing to pay for all day parking?

□ $3 ______
□ $5 ______
□ $7 or more ______

License plate information (optional)
(Example License plate “ADF 1431”, enter First two letters “ADF”; Last Digit “1”)

First two letters ____; Last Digit____

Code XXXX
We would like to follow-up with you on your parking experience.

□ I am willing to complete an additional questionnaire regarding my parking experience. I prefer to (circle one):

1. Receive the questionnaire via email, and return it by email.
Email address: _________________________

2. Fill a web base survey.
Email address: _________________________

3. Have a researcher call me and ask me questions on the phone.
Phone Number:___________
First Name:____________

□ I do not wish to complete an additional questionnaire regarding my parking experience.

Appendix B: Visitor Follow up Survey (Web-based or by mail)

Introduction to Visitor Follow up Survey (Web-based or by email)

Mrs/Mr:

We are contacting you because when you visited Virginia Tech recently you indicated that you were willing to complete an additional questionnaire regarding your parking experience. To complete the survey it will only take you a couple of minutes. This survey will help the Virginia Tech Office of Transportation to identified strategies to better serve the visitors in the future such as especially dedicated visitor's parking spaces, more detailed parking maps, etc. These measures will translate into measurable benefits to the visitors, like taking shorter time to find a parking space and parking near the building they are planning to visit.
If you are completing the survey in the web please click in the following link ………..
If you are sending the complete questionnaire by email please send it to the following e-mail address ________

Code XXXX (please enter this code in the survey)

If you have any questions you can reply to this e-mail with your questions or call
Alejandra Medina at 540-231-1508
Your collaboration is greatly appreciated. Thank you very much for your time.

1 .The reason for your visit was:

□ Prospective student or family member
□ Current family member or friend of a student
□ Academic Activities
□ Local resident using VT facilities or services
□ Other____

2. Which University office(s) did you visit?

____

____

3. In what building(s) are they located?

____

____

4. How long did it take you to find a parking space?

□ Between 1 and 2 min
□ Between 2 and 5 min
□ Between 5 and 10 min
□ More than 10 min

5. How many parking lots did you visit until you found a parking space?

□ 1
□ 2
□ 3
□ More than 3

6. How long did it take you to walk from your vehicle to your point of destination?

□ Less than 1 min
□ 1 min to 2 min
□ 2 to 5 min
□ 5 to 10 min
□ More than 10 min

7. How long did you park?

□ Less than 1 hour
□ 1hour to 2 hours
□ 2 hours to 3 hours
□ 4 hours
□ More than 4 hours

8. How long were you planning to park?

□ Less than 1 hour
□ 1hour to 2 hours
□ 2 hours to 3 hours
□ 4 hours
□ More than 4 hours

9. Did you park in more than one parking lot?

□ Yes □ No

10. Would you say that your parking experience at Virginia Tech was...

□ Very good □ Good □ Fair OR □ Poor

a. If poor, please explain why.____

11. After your experience with the visitor parking at VT, would you be willing to pay for designated visitor only parking spaces?

□ Yes □ No

12. If your answer is Yes, what is the <u>most</u> you would be willing to pay for all day parking?

□ $3 ______
□ $5 ______
□ $7 or more ______

13. How often do you visit Virginia Tech Blacksburg campus? Please specify number of times

□ Only once ______
□ Per a year ______
□ Per semester ______
□ Per month ______
□ Per week ______
□ Other (please specify) ______

14. License plate information (optional)
(Example License plate "ADF 1431", enter First two letters "ADF"; Last Digit "1")
First two letters ____; Last Digit____

15. After your parking experience, how do you think Virginia Tech can improve visitor parking?
Additional Comments
____
____
____
____
____
____

Appendix C: Initial Survey to be administered by the Visitor Center

Code XXXX

The Office of Transportation at Virginia Tech is conducting a visitor parking survey to determine the current visitor parking demand and how we can better serve the visitor's parking needs. Would you be willing to complete this short survey?

1. Are you at least 18 years old

□ Yes □ No

If your answer is no, please stop here and discard the survey.

2. The reason for your visit is:

□ Prospective student or family member
□ Current family member or friend of a student
□ Academic Activities
□ Local resident using VT facilities or services
□ Other____

3. If your answer is Academic Activities, do you already have a scheduled meeting?

□ Yes □ No

4. Which University office(s) are you planning to visit?

____

5. In what building(s) are they located

____

6. Do you know the location of the office(s) that you need to visit?

□ Yes □ No

7. How long are you planning to stay at your destination?

□ Less than 1 hour
□ 1 hour to 2 hours
□ 2 hours to 3 hours

□ 4 hours
□ More than 4 hours

8. How often do you visit Virginia Tech Blacksburg Campus? Please specify number of times

□ Only once _____
□ Per a year _____
□ Per semester _____
□ Per month _____
□ Per week _____
□ Other (please specify)

License plate information (optional)
(Example License plate "ADF 1431", enter First two letters "AD"; Last Digit "1")
First two letters ____; Last Digit____

Code XXXX

We would like to follow-up with you on your parking experience.

□ I am willing to complete an additional questionnaire regarding my parking experience. I prefer to (circle one):

1. Receive the questionnaire via email, and return it by email.
Email address: ________________________

2. Fill a web base survey
Email address: ________________________

3. Have a researcher call me and ask me questions on the phone
Phone Number:___________
First Name:____________

□ I do not wish to complete an additional questionnaire regarding my parking experience.