Carleton University
Carpool System

By

Robert Holwell - 287049
Linton DonBosco - 231721

Supervisor: Professor Gabriel Wainer

A report submitted in partial fulfilment of the requirements
of 94.498 Engineering Project

Document Status:	Final
Version:	1.0
Issue Date:	April 4, 2003
Department:	Systems and Computer Engineering Facility of Engineering Carleton university

ABSTRACT

There are many problems of great concern that exist in the world today. One major problem affecting our society is air pollution, caused by automobiles. Solutions are required to help reduce and control the emissions caused by gas-fuelled automobiles.

An effective way to decrease such emission is through developing a carpool system. This enables a group of people to share daily driving times and expenses and inturn help reduce air pollution, by decreasing the number of automobiles travelling on roadways. Another way of reducing air pollutants is to eliminate traffic jams. If the travel paths of carpool members were known then this information could indicate congestion points and thus be used to better plan new roadways or even improve current roadways.

To implement both solutions a web based carpool system was designed. This system allows registered members to perform a search that enables them to create a carpool group. The carpool system also enables the administrator to access database informatio n belonging to each member and perform an analysis that provides traffic flow details about all carpool members.

ACKNOWLEDGEMENTS

At this time we would like to thank Professor Gabriel Wainer for his advice and feedback throughout

the duration of this project.

AUTHORS

Name	Number	Course	Contact Information
Robert Holwell	287049	SYSC 4908	robertholwellis@msn.com
Linton Franklin DonBosco	231721	SYSC 4908	linton2001@hotmail.com

REVISION HISTORY

Version	Date	Author	Reason for Re-Issue
0.1	2003-03-03	Robert Holwell	Initial draft of final report
0.2	2003-03-10	Linton DonBosco	Initial draft of final report
0.3	2003-03-20	Robert Holwell	Updated initial draft
0.4	2003-03-25	Linton DonBosco	Review and Update Version 0.3
1.0	2003-04-02	Robert Holwell	Final Updates

WRITTEN DETAILS

Name	Sections
Robert Holwell	1, 3.2, 3.4, 4.1.1, 4.2.3, 4.2.4, 4.2.4.1, 4.2.5, 4.2.6.1, 4.2.6.2, 4.2.6.4, 4.2.6.6, 6, 7
Linton Franklin DonBosco	2, 3.1, 3.3, 3.4, 4.1.2, 4.2.1, 4.2.1.1, 4.2.2, 4.2.6.5, 4.6.2.7, 5, 6, 7

APPROVALS

Function	Name	Dept	Contact Info
Project Coordinator	Cheryl Schramm	SCE	schramm@sce.carleton.ca
Project Supervisor	Gabriel Wainer	SCE	Gabriel.Wainer@sce.carleton.ca

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In today's society, people are concerned about environmental issues. One such issue is air pollution but more specifically the causes of air pollution. For this reason, many environmentally friendly designs are being developed around the world to help reduce pollutions generated by every day living. A main source of pollution is automobiles that operate by gasoline. To help reduce the emissions caused by automobiles, many cities, companies and universities are taking the challenge of developing carpool systems.

Air pollution can be reduced by decreasing the number of cars travelling on roads and highways and also by limiting the amount of congestion caused by poorly designed roadways. By reducing both aspects of automobile travel previously mentioned the amount of air pollutants being emitted into the atmosphere will be decreased.

This Carleton University Carpool System will not only help to reduce the number of cars currently travelling to Carleton University but will also determine what roadways within the Ottawa area are being congested by those travelling to Carleton University.

The objective of this project was to build a web-based carpool system that provides an automated carpool search and traffic flow analysis. The carpool system will allow the student and staff population of Carleton University to create a carpool involving other students and/or staff who live in similar areas. This will allow members of the University community to share daily driving times and expenses, plus help reduce air pollution. By registering as a carpool member, he/she will provide some personal information such as current address. This information, plus other knowledge about main street

intersections and routes will enable the system administrator to create a chart or local map detailing the traffic flow of Carleton's carpool members.   Hence, providing information that can help future roadways to be specially engineered to reduce traffic congestion.

Making the carpool system operate, as was purposed at the beginning of the project, involved learning many different areas of Java and system networking. Before implementing our design a complete network was constructed and before a network was constructed a full design approach was taken. Below are detailed accomplishments that were made towards the solution of Carleton Universities Carpool System.

- **Design Approach** – The following steps were taken to successfully design the carpool system. All steps below were captured in separate documents and can be viewed in the corresponding Appendix.

    - *Current Carpool Systems* – researched other systems currently on the web to see what other systems had to offer. (**Appendix A**)

    - *Use Case Diagrams and Descriptions* (**Appendix B**)

    - *Class Diagrams* (**Appendix C**)

    - *Sequence Diagrams* (**Appendix D**)

    - *State Charts* (**Appendix E**)

    - *System Design* (**Appendix F**)

- *System Setup Procedures* – a list if steps that need to be executed in order to install, configure and verify the operation of the Web Server, Servlets, Database, JDBC, SSL and Java Mail. (**Appendix G**)

- *Database Initialisation Procedures* – script used to create the required tables and data structures in order to successfully write to and read from the database. (**Appendix H**)

- *Design and Verification Test Cases* (**Appendix I**)

- **Web Server** - A Tomcat-Apache Web Server was researched, configured and tested successfully in a Windows environment.

- **Database** – MySQL database was also researched, configured and tested successfully in a Windows environment.

- **Java Database Connectivity** (JDBC) – a java JDBC library was required. This was installed, configured and successfully testing in a Windows environment.

- **Secure Socket Layer** (SSL) – Secure Socket Layer functionality was imported and activate on the Tomcat web server.

- **Java Mail** – Java Mail functionality used.

Background information is mentioned at the beginning of this report to establish a foundation in which the report will build on. Secondly, a quick overview of the design details is mentioned. The report then proceeds to explain the vital network and database conditions that must be met in order for the system to work correctly. Fourthly, in depth detail about the accomplishments and rational behind our

solution is discussed. Finally, an explanation about other features that bring the system together as a package is reported. More details can be found in the Appendices, which are referenced throughout the report.

2. BACKGROUND

Due to the environmental concerns of the world today many solutions are being developed to help reduce the amount of pollutants emitted into the air on a day-to-day basis. One such source of pollution is gasoline-fuelled automobiles. To help reduce the emissions, cities, companies and universities are developing carpool systems.

Several carpool systems that are currently operational within Canada and the United States have been developed using similar user and system approaches. Slight differences are apparent but the overall goals are the same. Each carpool system can be separated into two main categories. The first being the administrative aspects. This includes:

- Describing benefits of becoming a Carpool member.

- Carpool System Etiquette.

- Signing of a Carpool Waiver.

- Describing who is eligible to become a carpool member.

- Parking regulations.

- Frequently asked questions.

The second being the technical aspects. This includes:

- The sign-in process.

- Carpool Searching.

- The ability to Change personal information.

Most systems provide a friendly and welcoming atmosphere, allowing a quick sign-up process, fast search results and easy navigation throughout the website.

The Carleton University Carpool System has taken a similar approach during design decisions, but because of the project objectives an emphasis is placed on the registration, carpool search and statistical analysis processes. More precisely the Carleton University Carpool System focuses on the storage of information during the registration process and how the database can be structured to provide quick carpool search results and statistical/graphical analysis results. Also, there is a focus on how the information is manipulated to display effective results to the member and administrator.

3. SYSTEM DESIGN

The following sections contain detailed information about non-functional requirements, Design Decisions, software tools used and examples of the carpool system design.

3.1 Non Functional Requirements

- **Response Time:** The Carleton University Car Pool System (CUCPS) should be able to confirm a car pool match within 3 sec.

- **Throughput:** The CUCPS should handle up to 50 requests concurrently.

- **Memory:** On the server, CUCPS should require less than 64 MB to run and about 2.5 GB of storage space for both code and data.

- **Security:** The CUCPS should be secure, not allow non members.

- **Availability:** The CUCPS should be available at least 99.999 percent of the time.

- **Usability:** The CUCPS should be very user friendly.

3.2 Design Decisions

During the design phase of the carpool system many decisions were made. Decisions relating to several of the Non-Functional Requirements listed in Section 3.1 above, such as Response Time, Security and Usability. Other important decisions involved what software packages are required and which software language should be used.

The first decision made was to use *java* and *java servlets*. This will provide the ability to compile code with html embedded. During execution the users will only see the html code not the java code, which helps remove certain security issues. Java also provides a communication link between a web

server and a database (*JDBC*). Secondly, in order to use Java Servlets, a *Web Server* that provides servlet capabilities is required. Open source software is widely available on the web, which provides several Web Server options. Details about the web server used are found in the following section. Similarly to Web Servers, *Database software* is widely available on the web as open source software. Since user information needs to be store and retrieved then a database is required for the carpool system. The database used is described in the following section. As well, a network diagram will be displayed later in the report.

Implementing a correct database structure was the key to minimizing the "*Response time*" during carpool searches. By using each Postal Code in the Ottawa area as database tables the time for searches was reduced. Refer to Section 4.2.3 for more information on how database structures were used in the carpool system. With respect to "*Security*", since the user requires access to personal information during a carpool search a decision was made for every carpool member to have a username and password. This eliminates members accessing other member's information. As well, since the user is accessing personal information through a TCP/IP connection a *Secure Socket Layer* (SSL) connect was decided to be used to protect information being transported from an internet browser to the carpool systems database.

Usability of the system is also a main decision maker. It was decided to have a simple user interface with limited links, error messages that are easy to understand and details about different aspects of carpool systems.

3.3 Software Tools Used

This section contains a list of software tools used to complete the carpool system network and functionality. The References and **Error! Reference source not found.** located at the end of this report contains set-up procedures and web information about all software tools used by the system.

- **Tomcat-Apache Web Server** – A group decision was made to use java servlets, which would provide the functionality needed to produce web pages and at the same time secure the code. In order to have servlet capabilities, a Tomcat-Apache Web Server was researched, configured and tested successfully in a Windows environment.

- **MySQL Database** – In order to store and manipulate user data a database is needed. To provide such functionality the MySQL database was also researched, configured and tested successfully in a Windows environment.

- **Java Database Connectivity** (JDBC) – In order for the java servlets, located in the Tomcat web server, to be able to access the MySQL database, a java JDBC library was required. This was installed, configured and successfully testing in a Windows environment. Both the web server and the database had to be 100% completed before the JDBC could function properly.

- **Secure Socket Layer** (SSL) – When sensitive information is being sent from the users browser to the system database, it becomes a security issue. To handle this situation the Secure Socket Layer functionality was imported and activated on the Tomcat web server. The SSL feature encrypts the information sent from the browser to Web Server and then the Web Server decrypts the information so the system can then manipulate or store it.

- **Java Mail** – To make the web site more interesting and to create a personal atmosphere we incorporated a couple of features into our design that require Java Mail. This allows the carpool system to use java and the SMTP (outgoing mail) of your ISP to automate the sending of email to carpool members.

3.4 Design Details

After analysing current carpool systems, investigating non-functional requirements and software requirements, the next phase of the carpool system design is creating use case diagrams and descriptions. This process breaks down the carpool system into smaller manageable parts and helps identify what functionality is required, the actors in the system and the flow of events. Figure 1: Use Case Diagram - Carpool System shows the actors external to the system, each Use Case of the system and a high-level flow of events.
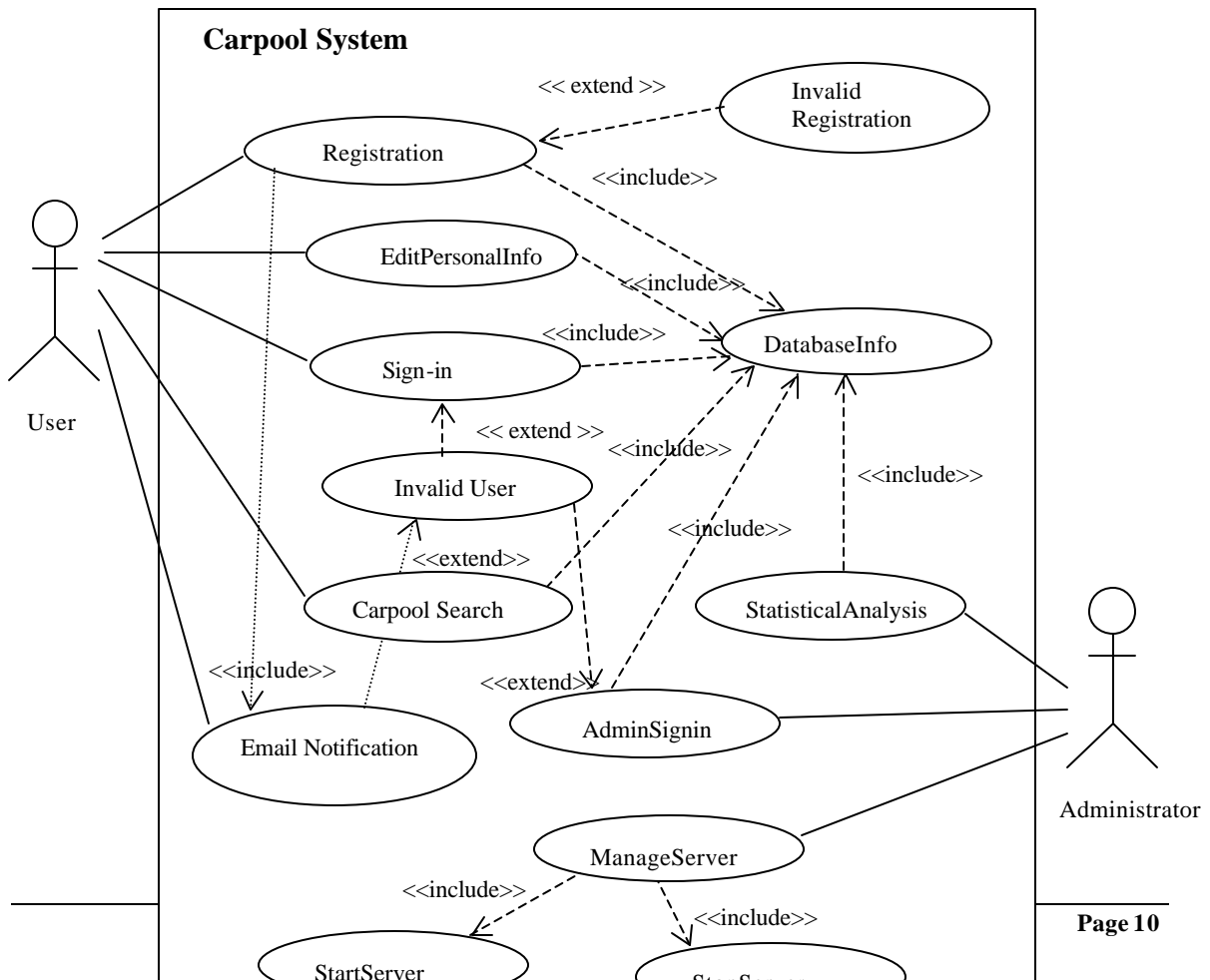
Figure 1: Use Case Diagram - Carpool System

With reference to Figure 1: Use Case Diagram - Carpool System the Registration Use Case includes Use Cases called Database Info and Email Notification and maybe Invalid Registration and Database Error. To further illustrate, a Use Case Description was developed for each Use Case or functionality in the Carleton University Carpool System. As an example, Figure 2: Use Case Description – Registration show the details involved during the Registration Use Case.

Use Case Name:	Registration
Extends:	**Invalid Registration**
Includes:	**StoreInfo**
Participating actor(s):	User
Entry conditions:	The user presses the "registration" button of the CU Carpool System Web Page.
Flow of events:	1. The user enters all required information into the empty fields.
	2. The user selects the Submit button. If the information given by the user is correct, registration is complete.
	3. The user information is stored in the database.
	4. An greeting **email** is sent to the new member with his/her username and password.
Exit condition:	5. A Successful message appears on the screen.
Exceptions:	**Invalid Registration**, **Database Error**
Special Requirements:	None

Figure 2: Use Case Description – Registration

A Use Case Description provides details about the entry conditions, flow of events, exit conditions

and exceptions. For more details about each Use Case in the carpool system please refer to **Error!**

Reference source not found. .

<<boundary>>
RegistrationForm

+ reqRegistration ( )
+ enterRecord ( )
+ allInfoEntered ( )

1..*   1

<<control>>
AddMemberControl

+ openAccount ( )
+ validAccount ( )
+ displayReqst ( )
+ reqEmail ( )
+ diplayMeg ( )

1

<<boundary>>
PasswordNotification

+ allInforEntered ()
+ enterEmail ( )

1..*   1

<<control>>
EmailControl

+ sendEmail ( )
+ verifyEmail ( )
+ emailInvalid ( )

1

<<boun dary>>
SignInForm

+ reqSignin ()
+ enterRecord ( )
+ allInfoEntered (
)

1..*   1

<<control>>
SignInControl

+ verifyLoginInfo ()
+ displayReqst ( )

1

<<boundary>>
CarpoolManagement

+ enterRecord ( )
+ reqEditInfo ( ) : void
+ reqSearchCarpool ( )
+ validEntry ( )
+ submit ( )
+ submitEdit ( )

1..*

1..*   1

<<control>>
SearchCarpoolControl

+ displayInfo ( )
+ searchCarpool ( ): Member

1..*   1

<<control>>
EditPersonalInfoControl

+ getInfo ( ) : Member
+ allInfoEntered ( ): void
+ updateInfo ( )
+ displayInfo ( )
+ diplayReqst ( )
+ confirmMsg ( )

<<boundary>>
AdminSignInForm

+ reqAdSignin ()
+ enterRecord ( )
+ allInfoEntered (
)

1

<<boundary>>
AdministratorManagement

<<control>>
AdministratorControl

1   + getTravelPaths ()
+ setTravelPaths ()

<<control>>
DatabaseControl

- totalDriver : int
- totalPassenger : int
- counter : Long

+ validAccount ( )
+ storeInfo ( )
+ retrieveInfo ( )
+ searchInfo ( )

1

1..*

<<entity>>
Member

-       timeOfDeparture:
String
- lastName: String
- firstName: String
- address: String
- phoneNumber: String
- email: String
- gender: String
- userName: String
- password: Sring
- destination: String
- origination: String
- passenger: boolean
- driver: boolean
- mainInt: string

<<entity>>
Passenger

- totalPassen: int

<<entity>>
Driver

-       carType:
String

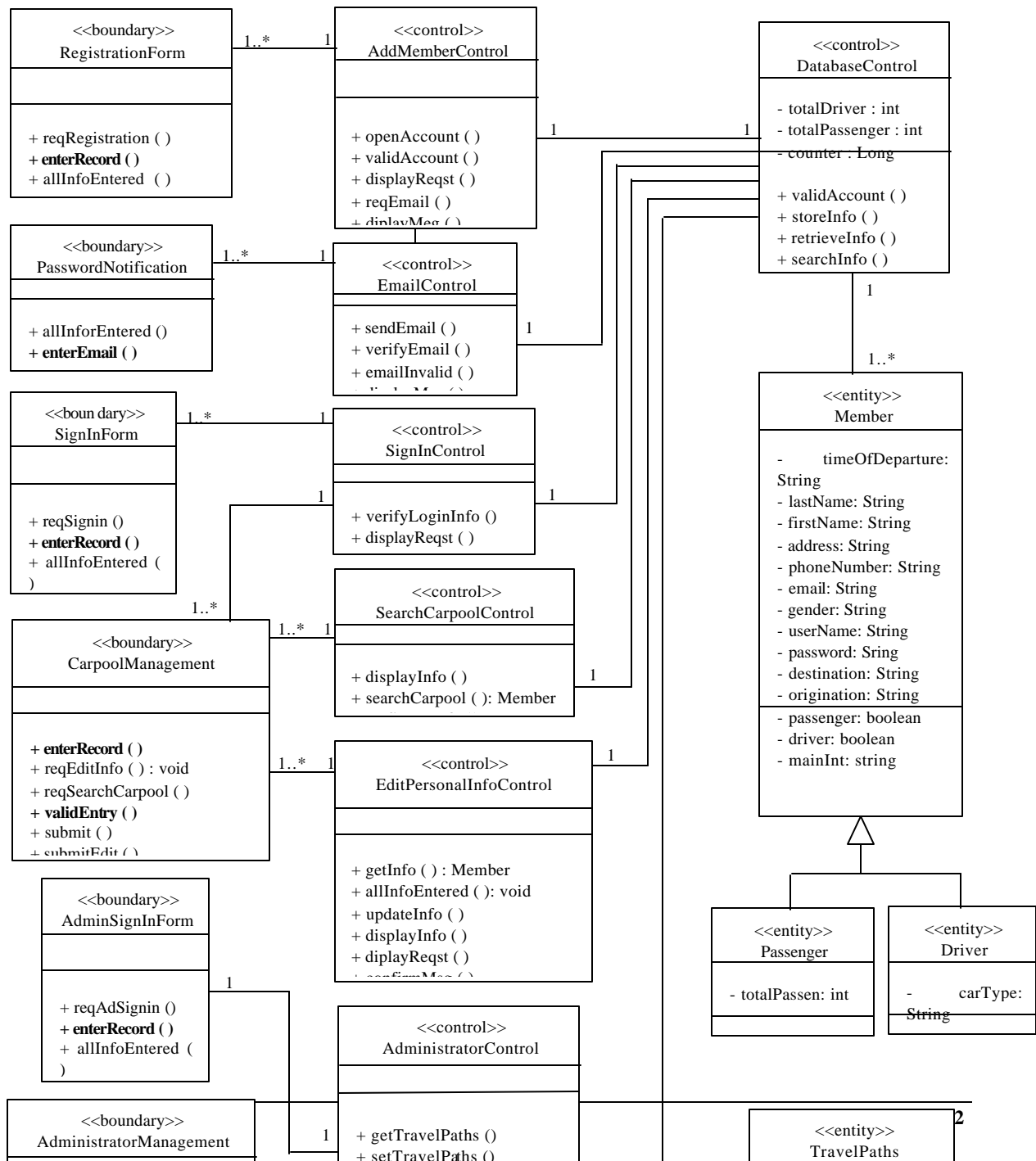**<<entity>>**
TravelPaths

Figure 3: Class Diagram

After completing the Use Case Analysis the next approach was to define the *boundary*, *control* and *entity* classes. Figure 3: Class Diagram above shows all defined classes relating to the carpool system. The class diagram also identifies the linkage between each class. For further details about the method and attributes used, refer to the data dictionary in **Error! Reference source not found.**.

The flow of control that occurs during execution of the carpool system also needs to be broken down into its defined parts. From the Carpool System Class Diagram each flow sequence was defined, establishing which classes and methods were required to successfully complete a Use case. Continuing to use the Registration Use Case, defined at the beginning of the design, as an example, Figure 4: Registration Sequence Diagram below shows the flow of execution from one method to another and the interaction of the user. For more details on execution flow please refer to the Sequence Diagrams located in **Error! Reference source not found.**.

Figure 4: Registration Sequence Diagram
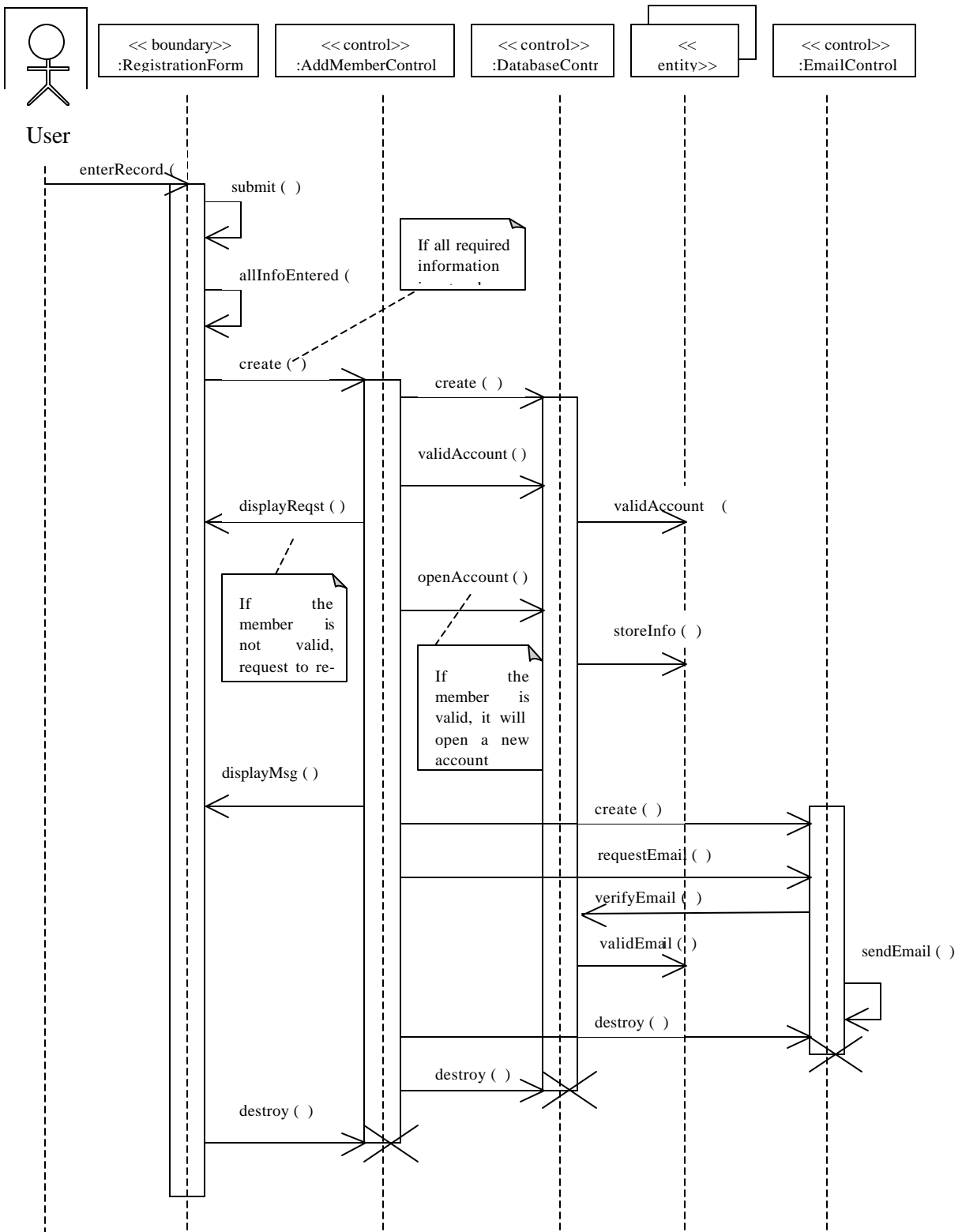
To further understand the carpool systems, states charts were created for each Use Case and its components. This will allow the user to understand the system, as it reacts to triggers. The following figure shows the state chart of the Registration Use Case.
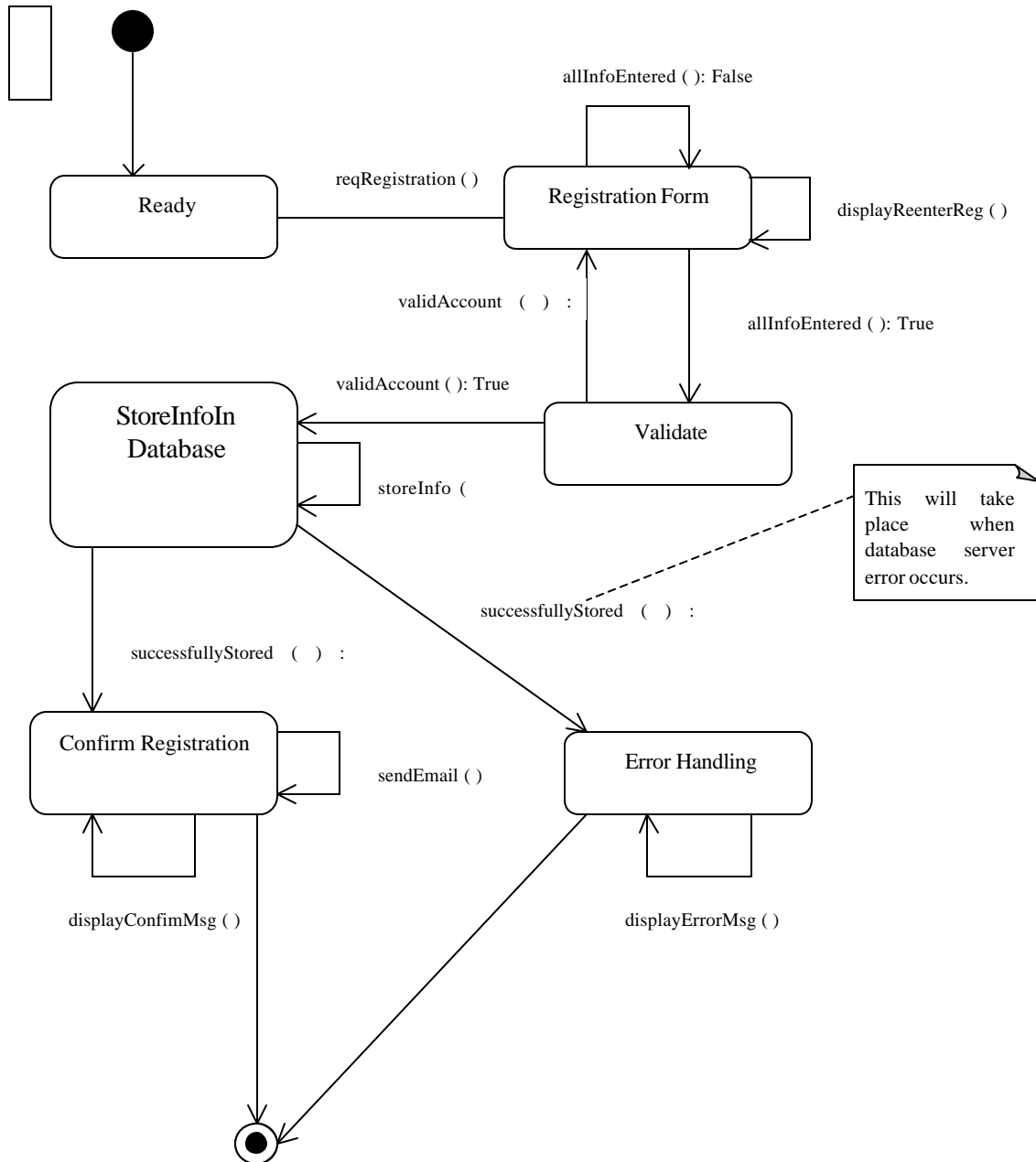

Figure 5: State Chart - Registration

Each rectangle represents a state of the system and the arrows and methods represent the triggers. The diagram starts at the solid circle and finishes at the double circle. The path it takes to get there depends on the triggers and the systems state. For further information on State Diagrams relating to the carpool system and its Use Cases please refer to **Error! Reference source not found.**.

Another important aspect of system design is System Decomposition and the Deployment Diagram. For information on both of these topics please refer to **Error! Reference source not found.**.

3.5 Testing

Testing is a very important and critical aspect of the software development process. In order to assure that the Carleton University Carpool System functions properly, unit testing, integration testing and regression testing were fulfilled throughout all parts of the system. This approach will generate stable and bug free software.

A test plan document has been created illustrating the functions tested. This document can be found in APPENDIX I.

4. TECHNICAL DETAILS

The following sections contain detailed information about the network used by the carpool system and system functionality.

4.1 System Implementation

This section provides details about the network and database set-up required in order for the system to function correctly.

4.1.1 Network Configuration

Prior to any implementation or coding a complete network needed to be configured. The approach taken was based on several decisions made during the design process.

1. Use of Java Servlets

2. Need to store and retrieve user information

3. Member information Security

From these three decisions the following details were finalized.

1. Apache Tomcat Web Server that provides Java Servlets capabilities.

2. MySQL database to store and retrieve member information

3. Java Database Connectivity (JDBC) to connect the Web server to the Database

4. Secure Socket Layer (SSL) to encrypt user information when sent over the web.

The above details lead to the network diagram shown in Figure 6: Network Diagram below:
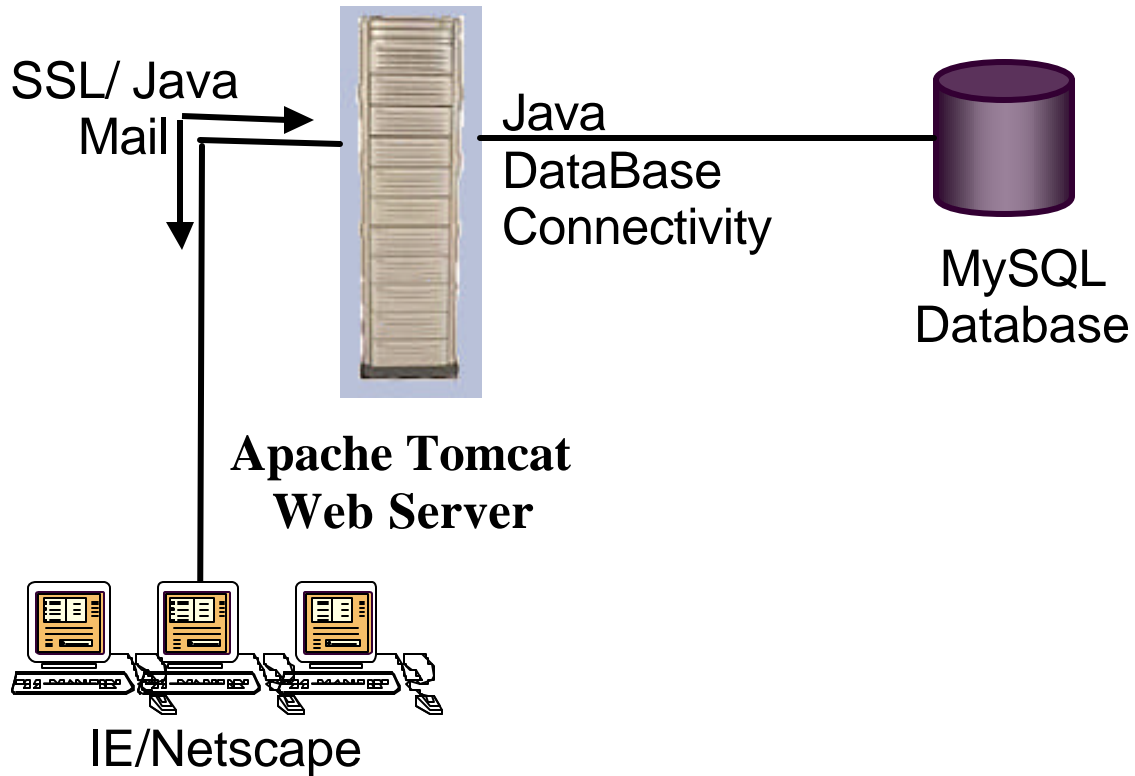

SSL/ Java Mail

Java DataBase Connectivity

MySQL Database

Apache Tomcat Web Server

IE/Netscape

Figure 6: Network Diagram

It is important to note that all nodes in the diagram above are configured on one computer running Windows 2000. Please refer to **Error! Reference source not found.** for full step-by-step instructions on how to set-up the above configuration on a Windows 2000/NT system. With a complete network working the implementation and testing phase was successful.

4.1.2 Database Management

An important part of the network consists of the MySQL database. This enables members to read information from and write information to the database with no effort or know how on the part of the carpool members. With just a click of a button, writing to the database occurs during the registration process, changing password, and also during the edit personal information process where reading from the database occurs during most web operations.

In order to successfully store information in the database, the database itself must be formatted correctly. To do this a script was created that, when executed on the mysql command prompt, will automatically create the desired tables and data structures. Table 1: DATABASE INITIALIZATION below is an exert from the db_initialization file showing a sample of the commands used, naming convention and data types.

```
CREATE TABLE K1Y(POSTALCODE VARCHAR(7) NULL,
                DEPARTHOME VARCHAR(8) NULL,
                DEPARTDEST VARCHAR(8) NULL,
                LASTNAME VARCHAR(25) NULL,
                FIRSTNAME VARCHAR(25) NULL,
                ADDRESS VARCHAR(30) NULL,
                PHONENUMBER VARCHAR(12) NULL,
                EMAIL VARCHAR(35) NULL,
                GENDER VARCHAR(6) NULL,
                USERNAME VARCHAR(20) NULL,
                PASSWORD VARCHAR(12) NULL,
                DESTINATION VARCHAR(30) NULL,
```

```
                    ORIGINATION VARCHAR(30) NULL,

                    PREFERENCE CHAR(20) NULL,

                    MAININT VARCHAR(50) NULL,

                    PARTTYPE VARCHAR(10) NULL,

                    CITY VARCHAR(20) NULL,

                    PROVINCE VARCHAR(20) NULL,

                    COMMENTS VARCHAR(100) NULL,

                    REGISTRATIONDATE VARCHAR(20) NULL);

CREATE TABLE MEMBERS(USERNAME VARCHAR(20) NULL,

                    PASSWORD VARCHAR(12) NULL,

                    FIRSTNAME VARCHAR(25) NULL,

                    EMAIL VARCHAR(35) NULL,

                    POSTALCODE VARCHAR(7) NULL);

CREATE TABLE ADMIN(USERNAME VARCHAR(20) NULL,

                    PASSWORD VARCHAR(12) NULL);

INSERT INTO ADMIN (USERNAME, PASSWORD) VALUES('passadmin','passadmin');

CREATE TABLE POSTAL(POSTALCODE VARCHAR(3) NULL);
```

Table 1: DATABASE INITIALIZATION

For more details about the db_initialization file please view **Error! Reference source not found.** .

4.2 System Features

This section illustrates the use of the many functions offered by the Carleton University Carpool System.

4.2.1 Registration

The registration functionality is the most important in the carpool system. In order to become a carpool member, the member must register in the carpool system. Without the member information being stored during this registration process, the carpool member will not be able to access the carpool system. The goal of the registration is to create a member account (store member' personal information in the database) so that the members can sign-in to the carpool system using his/her registered user-name and password. More details about the actual information that is required to complete a carpool registration will be explained later in this section. There is a registration link on the carpool system's main web page that a user can select to register in the carpool system. Registration is a web form that the member has to input his/her personal information such as name, address, postal code, near intersection, and departure time etc.

In the carpool database there is a table called INTERSECTION, which contains all the main intersections in the Ottawa area. Located in the registration form is a drop down menu, which provides a list of near intersections for the carpool member to select. This list of near intersections is retrieved

from the database table called INTERSECTION. Please view Figure 7: REGISTRATION FORM

for exact information that the user has to input during the registration.

Figure 7: REGISTRATION FORM

There are some required fields in the registration form marked by ∗ as can be seen in Figure 7 The user must complete these fields in order for the registration to be successful. After all required information has been entered, the user must then select the "Submit" button in the registration form to submit his/her information.

The criteria which must be satisfied during the registration process can be seen in Table 2:VALIDATION IN REGISTRATION FORM below. If any of these fail, the registration is not complete and the appropriate error message will be displayed for the user to correct.

|	**Criteria**	**Description**
1.	All the required fields in the registration form must be filled.	
2.	Validate user name	• User name must either be letters, digits, '_' or a combination of all. • User name is unique, it should not already exist in the datbase. (should't be already selected by another carpool member.) • No space allowed between the characters.
3.	Validate password	• Password must be 6-12 characters long and either letters, digits or a combination of both. • Password is unique, it should not already exist in the database. (should't be already selected by another carpool member.) • No space allowed between the

		characters.
4.	Valdiate phone number	• Phone number must be a 10 digit number only.

5.	Validate email address	Email address must have the follwing:
		
		• Must have only one "@" character in the email address.
		
		• At least one "." character exists in the email address.
		
		• Must be either letters, digits, '_' or either combination of all , and also allow the ".", "@" characters only.
		
		• First character must either letter or digit only allowed.
		
		• No space allowed between the characters.
		
		• Not cotinues "." Character allowd.
		
		  Eg. nick@hotmail..com.
		
		• Not allow the "." Character, right after the "@" character.
		
		  Eg. nick@.hotmail.com

6.	Validate postal code	• Postal code must be the format as follows: **K2E 7P8 or K2E7P8.** • First 3 characters of the postal code must be in Ottawa or Quebec (Hull-Gatineau) area.

Table 2:VALIDATION IN REGISTRATION FORM

Please view Figure 8: SUCCESSFUL REGISTRATION for the correct output that will be seen by a carpool member after completing a successful registration. The information displayed in Figure 8 below provides the exact member information stored in the database.
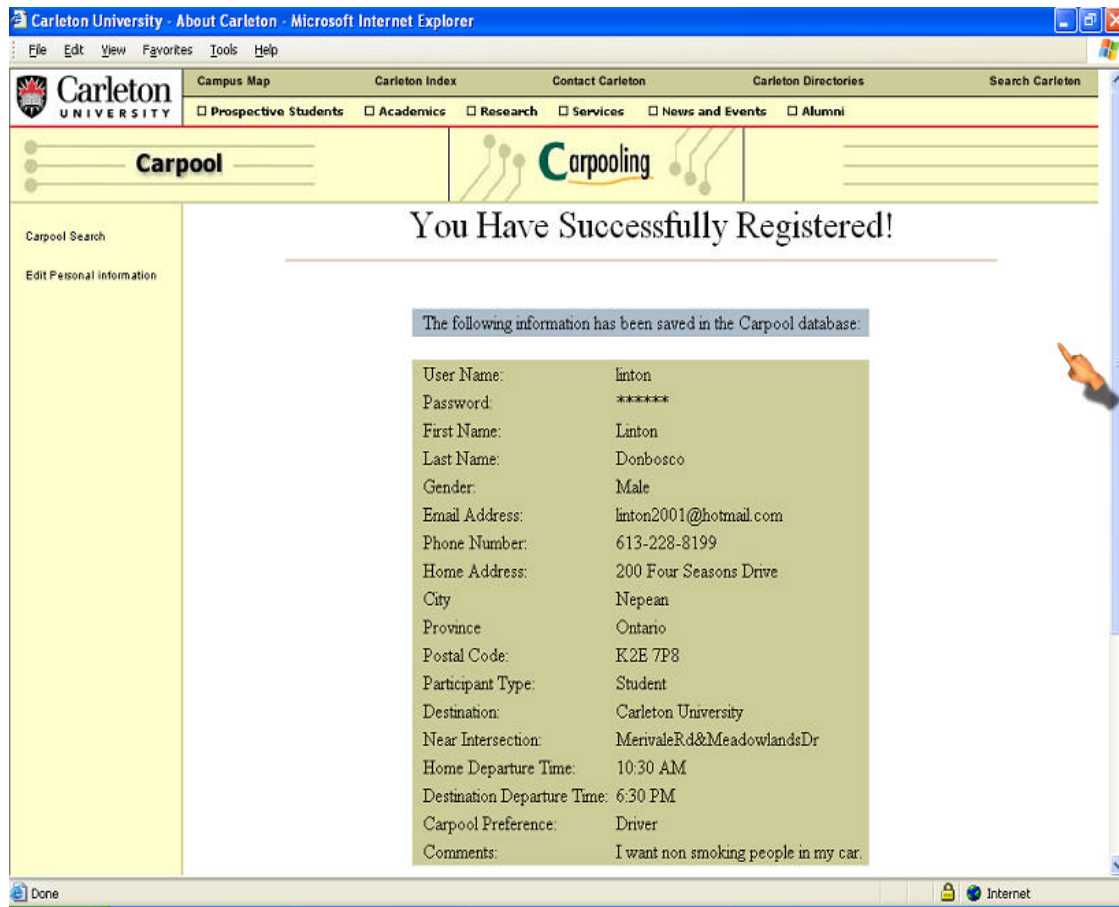
Figure 8: SUCCESSFUL REGISTRATION

At the end of the registration process, if the user successfully registered in the carpool system, then he/she will receive an email with their user name and password.

4.2.1.1 Storing Data

The following section will explore how a members' information is stored in the database during the registration process. The goal is to access the database as quickly as possible, so the carpool members are not waiting for results.

All data will be stored in a Database called **Carpool**. The database will contain many tables; many of which are named using the first three characters of each postal code in the city of Ottawa and surrounding areas. Within each "Postal Code" table (e.g. K2E) there are fields pertaining to each member, such as name, address, phone number, password, username, origination, email address, etc. All information will be store in a Postal Code table depending on the postal code used by the member during the registration process. There is a MEMBERS table in the database as well, which has fields pertaining to each member, such as username, password, firstname, email, and postal. Postal field in the MEMBERS table contains the first three characters of the Postal Code of the user, which is needed to identify the table that contains the user information. A visual look at the structure of the database can be seen in Figure 9 below.

Carpool

K2C

username	password	name	email	address	Preference	● ● ●
linton	lin2003	Linton	hotmail	2 Mill St.	Driver	● ● ●

K2E

username	password	name	email	address	Preference	● ● ●
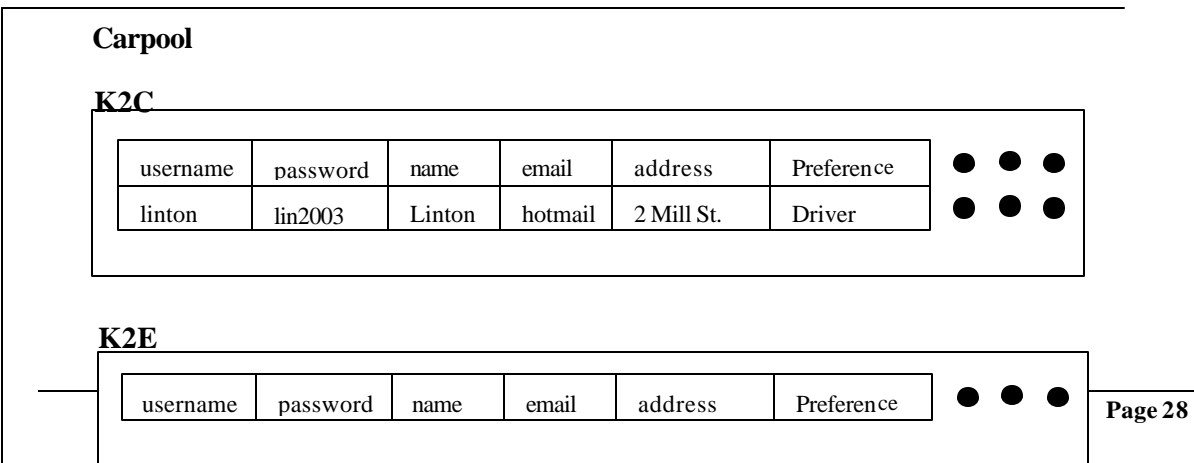
K3K

username	password	name	email	address	Preference

MEMBERS

username	password	firstname	email	postal
linton	lin2003	Linton	hotmail	K2E

Figure 9: STRUCTURE OF DATABASE

The exact operation of storing data during the registration is explained below with an example using the data in the Figure 8: SUCCESSFUL REGISTRATION.

In this example, the postal code retrieved is "K2E" and therefore all the information is stored in the "K2E" database table. Please view Figure 10: EXAMPLE OF K2E DATABASE TABLE for an example of the K2E table.

K2E

USERNAME	linton
PASSWORD	******
LASTNAME	Donbosco
FIRSTNAME	Linton
POSTALCODE	K2E 7P8

GENDER	Male
PHONENUMBER	613-228-8199
EMAIL	linton2001@hotmail.com
PARTTYPE	Student
ADDRESS	200 Four Seasons Drive
CITY	Nepean
PROVINCE	Ontario
DESTINATION	Carleton University
DEPARTHOME	10:30 AM
DEPARTDEST	6:30 PM
PREFERENCE	Driver
ORIGINATION	200 Four Seasons Drive
MAININT	MerivaleRd&MeadowlandsRd
COMMENTS	I want non smoking people in my car.
REGISTRATIONDATE	March 18, 2003 8:55:05

Figure 10: EXAMPLE OF K2E DATABASE TABLE

MEMBERS

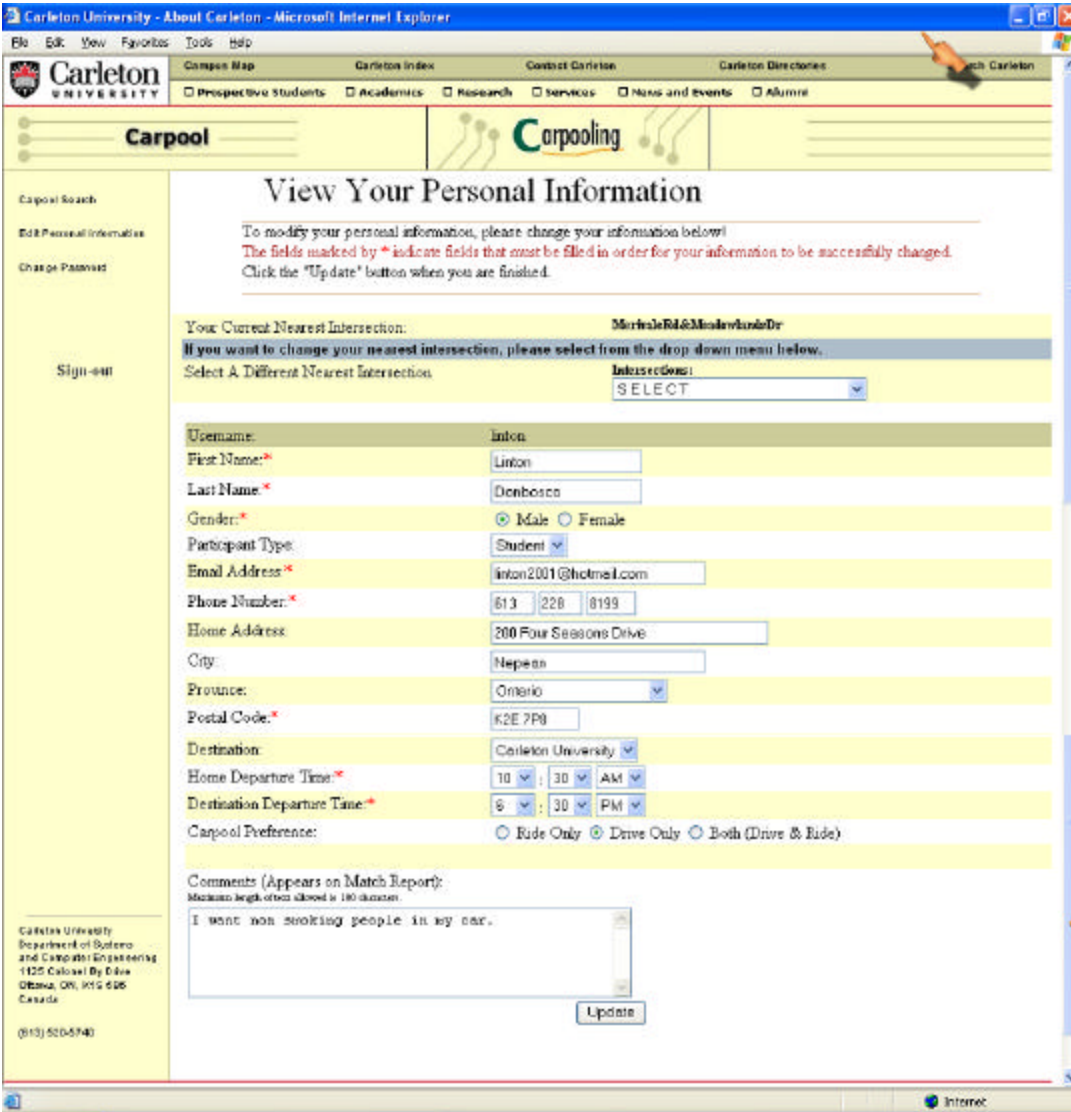| USERNAME	linton
PASSWORD	******
FIRSTNAME	Linton
EMAIL	linton2001@hotmail.com
POSTALCODE	K2E

● ● ●

Figure 11: EXAMPLE OF MEMBERS DATABASE TABLE

4.2.2 Edit Personal Information

The Edit Personal Information functionality allows the carpool members to view or edit their registered personal information. When a member moves, wants to change his/her phone number or wants to change his/her carpool search criteria then the member can do so by using this edit personal information feature. The carpool member can access this functionality after he/she has signed in to the carpool system using his/her username and password. This feature retrieves the user information from the database and displays it in a web-based form. Then, it allows the user to change the information and re-submit it to the database. Using this feature the user can re-select his/her nearest intersection from the drop down menu located in the form. Please view Figure 12: EDIT/VIEW PERSONAL INFORMATION for exact information that the user can view or edit using this feature.

Figure 12: EDIT/VIEW PERSONAL INFORMATION

A member is able to change all of their information except the username and password. The username is chosen during the registration process and cannot be changed by the member. This can be seen that from Figure 12. The member can only change their password by clicking the "Change Password" functionality link.

There are some required fields in the edit personal information form marked by * as can be seen in the Figure 12, that a user must complete to update information successfully. After, all the required information is changed the user has to select the "Update" button in the edit personal information form to submit his/her information.

All the necessary validation criteria that is required during the edit personal information process is described in Table 3: VALIDATION IN EDIT PERSONAL INFORMATION FORM below. If any of the criteria fail then the user cannot update the information and an appropriate error message will be displayed for the user to correct.

	Criteria	**Description**
1.	All the required fields in the edit personal information form must be filled.	
2.	Valdiate phone number	Phone number must be a 10 digit number only.
3.	Validate email address	Email address must have the follwing: - Must have only one "@" character in the email address. - At least one "." character exists in the email address. - Must be either letters, digits, '_' or either combination of all , and also allow the ".", "@" characters only. - First character must either be a letter or digit only allowed.

		- No space allowed between the characters. - Not continues "**.**" Character allowd. Eg. nick@hotmail**..**com. - Not allow the "**.**" Character, right after the "**@**" character. Eg. nick**@**.hotmail.com
4.	Validate postal code	- Postal code must be the format as follows: **K2E 7P8 or K2E7P8.** - First 3 characters of the postal code must be in Ottawa or Quebec (Hull-Gatineau) area.

Table 3: VALIDATION IN EDIT PERSONAL INFORMATION FORM

Please view Figure 13: UPDATE SUCCESSFUL RESULTS for exact output as seen by a carpool member after completing a successful update. The information displayed is the exact member' information stored in the database.
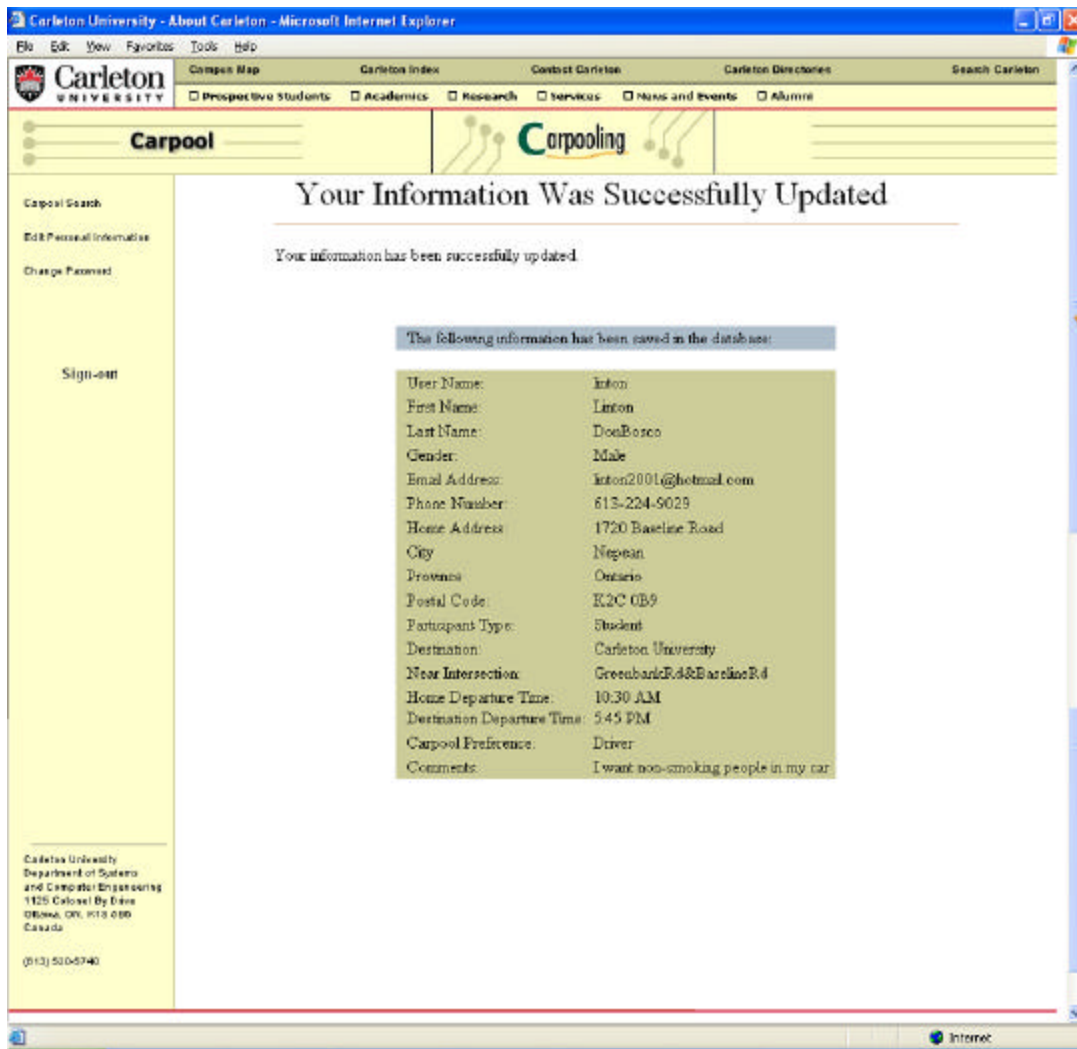
Figure 13: UPDATE SUCCESSFUL RESULTS

During the edit personal information process, if the user enters a different postal code then the original one, the information will be stored in the new postal code table and all the previous information will be deleted from the old postal code table.

4.2.3 Carpool Search

The carpool search functionality is directly dependent on the information stored in the carpool database. Without a member's information stored during the registration process, the carpool search would not work correctly. More details about what information is needed to complete a carpool search will be explained later in this section. The goal of the carpool search is to provide a carpool member with information pertaining to other carpool members that live within close proximity. This search should be completed in a timely manner so not to leave the member waiting for results. A long search time would be more apparent if the execution had to proceed through a large database. For example, to complete a carpool search of 10,000 members could take several seconds. To minimise the search time the database has been reduced from one large table to 38 tables. Each table represents a different postal code located in the Ottawa area. Please refer to Appendix H for the exact postal codes used. By doing this, the search can jump directly to the postal code it is concerned with and provide quick results. Instead of searching through 10,000 carpool members it may now only have to search through approximately 263 carpool members. This approach will make the search seem almost instant for the user.

When the carpool search is activated it uses specific information located in the carpool database, information pertaining to the member completing the carpool search.
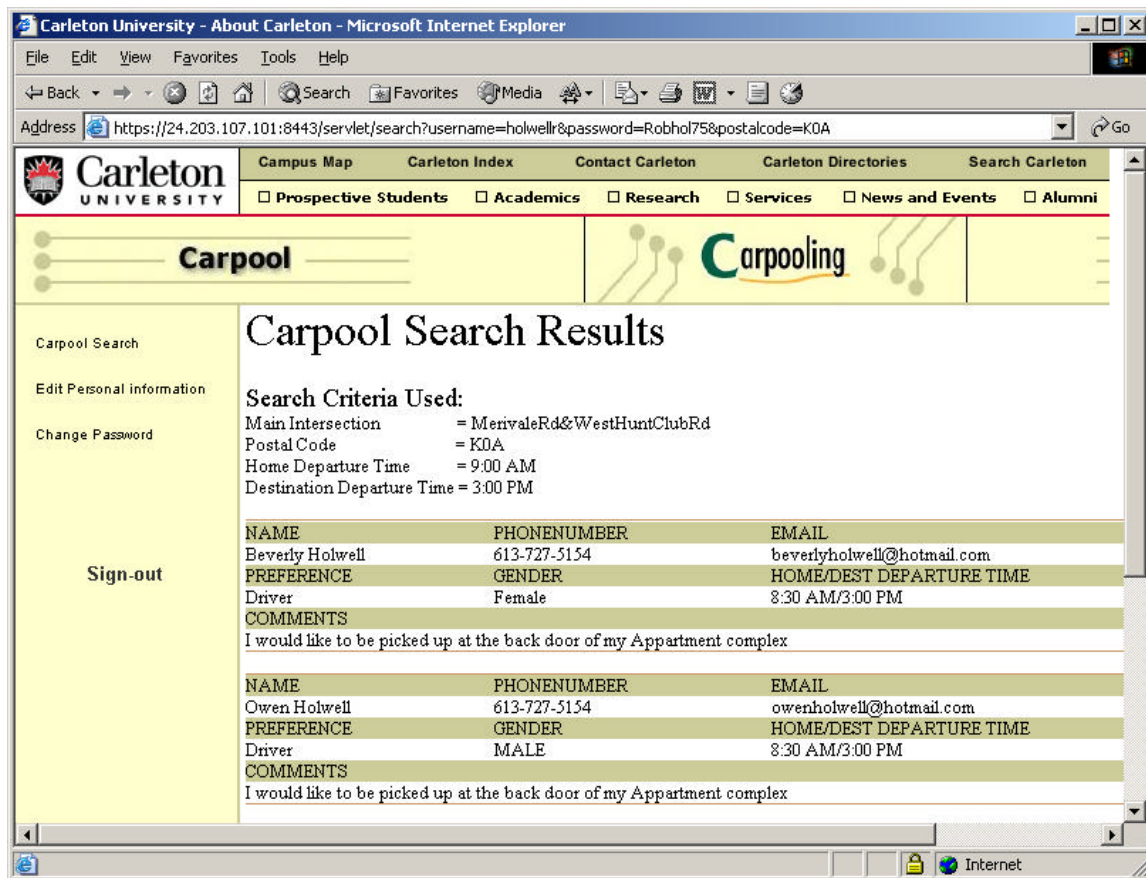
Figure 14: Carpool Search Results

From Figure 14: Carpool Search Results above, one can view the search criteria used to perform the carpool search. The Postal Code, Main Intersection, Home departure Time and the Destination Departure Time of the member completing the search are the 4 parameters used. Another parameter that is not shown on the screen but is used when completing a carpool search is the "Preference Type" (Driver, Passenger or 'Drive and Ride'). The exact operation of how the carpool search works is explained below.

Using the username and password collected during the sign-in process (e.g. username = holwellr, password = ******), the members table is accessed and the Postal Code is retrieved. Please view Figure 15: MEMBERS DATABASE TABLE for an example of the Members table.

MEMBERS

USERNAME	holwellr	linton	
PASSWORD	******	********	
FIRSTNAME	Robert	Linton	● ● ●
EMAIL	robert@hotmail.com	linton@hotmail.com	
POSTALCODE	K0A	K2E	

Figure 15: MEMBERS DATABASE TABLE

Once the Postal Code is located and retrieved from the members table execution proceeds to that specific postal code table.   In this example, the postal code retrieved is "K0A" and therefore the execution continues to the K0A database table.

From the K0A Table as indicated in Figure 16: K0A DATABASE TABLE below, the DEPARTDEST, DEPARTHOME, PREFERENCE, and MAININT, which belong to the carpool member completing the carpool search, are retrieved. This criteria is compared with all other members located in the "K0A" database table and if a successful match exists it will be displayed on the screen.

K0A

POSTALCODE	K0A 3K3	K0A 2k6
DEPARTDEST	3:00 PM	3:00 PM
DEPARTHOME	8:30 AM	8:30AM
LASTNAME	Holwell	Holwell
FIRSTNAME	Beverly	Owen
ADDRESS	905 Meadowlands Dr.	1165 Merivale Rd.
PHONENUMBER	613-727-51514	613-727-1234
EMAIL	beveryl@hotmail.com	owen@hotmail.com
GENDER	Female	Male
USERNAME	Holwellb	holwello
PASSWORD	******	********
DESTINATION	Carleton	Carleton
ORIGINATION	905 Meadowlands Dr.	1165 Merivale Rd.
PREFERENCE	Driver	Driver
MAININT	MerivaleRd&WestHuntClubRd	MerivaleRd&WestHuntClubRd
PARTTYPE	Student	Professor
CITY	Ottawa	Ottawa
PROVINCE	Ontario	Ontario
COMMENTS	No Smoking	Don't be Late
REGISTRATIONDATE	March 17, 2003	March 17, 2003

Figure 16: K0A DATABASE TABLE

A successfully match occurs only when all criteria are satisfied.

1. The MAININT of the searcher must match the MAININT of other carpool members in the

    K0A table (MerivaleRd&WestHuntClubRd).

2. The DEPARTHOME and DEPARTDEST times of other carpool member must be within a 1-

   hour range of the searchers DEPARTHOME and DEPARTDEST.

3. If the searcher is a driver all matches will be displayed on the screen, but if the searcher is just a passenger then only matches with a PREFERENCE of "Driver" or "Ride and Drive" will be displayed on the screen.

Please view Figure 14: Carpool Search Results for exact output as seen by a carpool member after completing a carpool search. The information displayed provides enough details for the carpool searcher to contact those individuals, who were a successful match.

4.2.4 Statistical Analysis

The statistical analysis functionality is also directly dependent on the member information stored in the carpool database and also by predefined paths, which are stored in the database by the system administrator. Without the member information stored during the registration process and the predefined paths, the statistical analysis will not work correctly. More details about what information is needed to complete a statistical analysis will be explained later in this section. The goal of the statistical analysis is to use database information relating to a members origination and destination, plus other knowledge about main street intersections and routes, to develop a chart on traffic flows of all carpool members. From Figure 18: Statistical Analysis Results below, the output displayed when a statistical analysis is completed can be viewed. The chart indicates a specific intersection and the number of carpool members that travel through that intersection, during the time interval defined by the administrator before the statistical analysis was activated.
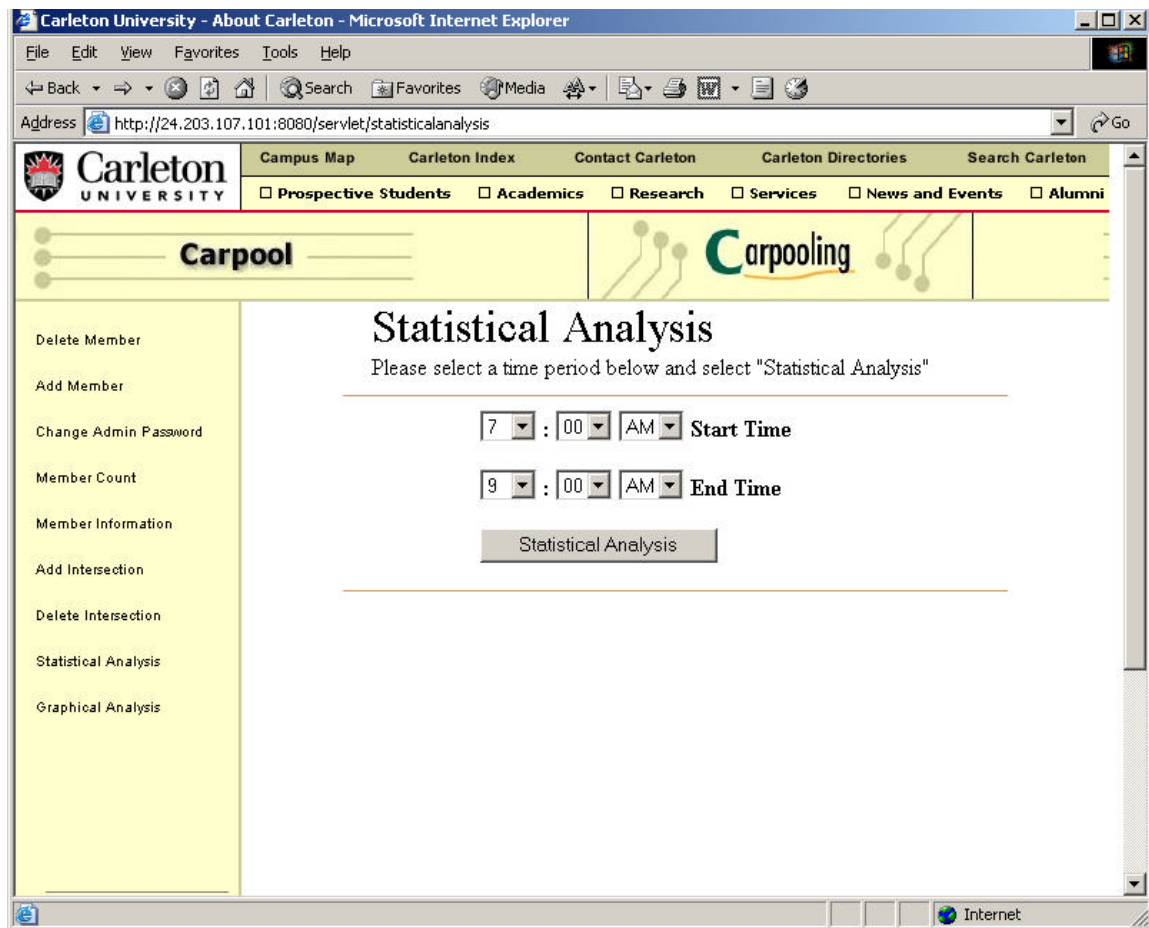
Figure 17: Statistical Analysis

When the start and end times (Figure 17: Statistical Analysis) are selected by the administrator and the statistical analysis is activated, it uses specific information located in the carpool database. This information comes from 4 different database tables.

1. Members

2. "Postal" e.g K0A, K2E, …

3. Routes

4. Intersections

For in depth details about each table listed above please refer to **Error! Reference source not found.**
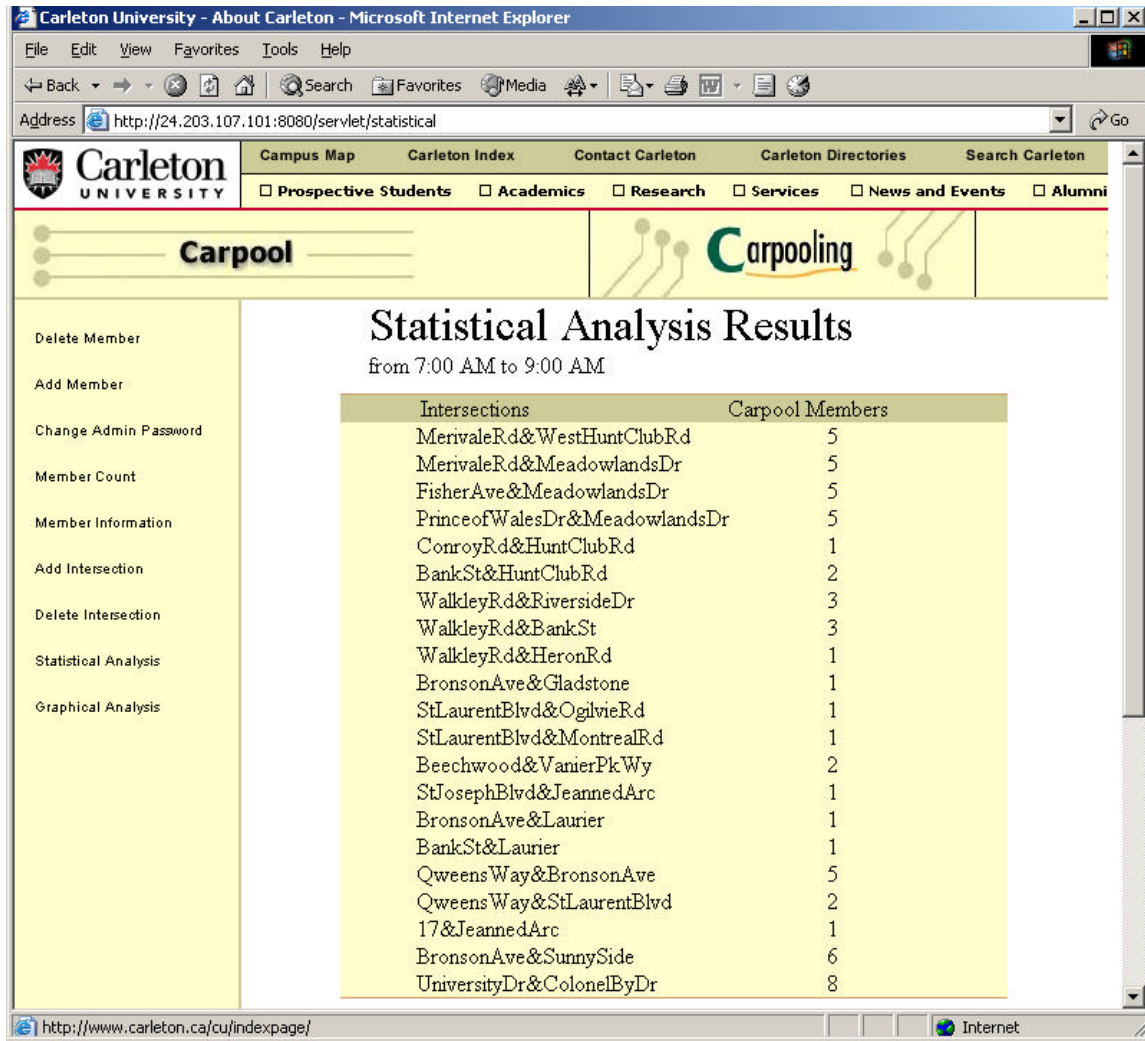
Figure 18: Statistical Analysis Results

It is important to note that the following procedure will be explained for one carpool member, but the process continues through all carpool members stored in the database.

The execution will access the first carpool member in the MEMBERS table and retrieve the username, password and postal code. Please refer to Figure 15: MEMBERS DATABASE TABLE for an example of information stored in the members table. Depending on what postal code is retrieved from

the MEMBERS table depends on which "Postal Code" table is accessed. In the example being discussed the K0A table will be accessed. Please refer to Figure 16: K0A DATABASE TABLE for an example of information stored in the K0A (any postal code table) table. From this table the DEPARTDEST, DEPARTHOME and MAININT are retrieved. If the DEPARTDEST and/or DEPARTHOME are equal to or between the start and end times selected by the administrator then the execution will continue using the same carpool member, otherwise the execution is brought back to the members table to access the next carpool member. If the times are a successful match then the ROUTES table is accessed. Here it uses the MAININT of the K0A table and finds an exact match in the first column of the ROUTES table. Once a match occurs all associated intersections belonging to the MAININT are retrieved. Please refer to Figure 19: ROUTES DATABASE TABLE below for a view of the ROUTES table. Continuing with the example, the first column matches with 'MerivaleRd&WestHuntClubRd", and therefore retrieves all associated intersections indicated here in a grey backgound.

ROUTES

MAININT	**MerivaleRd&WestHuntClubRd**	FallowfieldRd&MainSt
MAININT2	MerivaleRd&MeadowlandsDr	MainSt&FernbankRd
MAININT3	FisherAve&MeadowlandsDr	HazeldeanRd&MainSt
MAININT4	PrinceofWalesDr&meadowlandsDr	QweensWay&CarpRd
MAININT5	UniversityDr&colonelByDr	QweensWay&EagelsonRd
MAININT6	0	QweensWay&MoodieDr
MAININT7	0	QweensWay&WoodroffeAve
MAININT8	0	Qweensway&BronsonAve
MAININT9	0	BronsonAve&SunnySide
MAININT10	0	0
MAININT11	0	0
MAININT12	0	0
MAININT13	0	0
MAININT14	0	0
MAININT15	0	0

● ● ●

Figure 19: ROUTES DATABASE TABLE

Using the intersections retrieved from the ROUTES table, execution then proceeds to the INTERSECTION table. Each intersection is then used to search through the INTERSECTION table for an exact match. Once a match occurs then the associated count for that intersection is incremented. Please refer to Figure 20: INTERSECTIONS DATABASE TABLE for details.

INTERSECTIONS

INTERSECTIONS	COUNT
MerivaleRd&MeadowlandsDr	1
FisherAve&MeadowlandsDr	1
PrinceofWalesDr&meadowlandsDr	1
QweensWay&CarpRd	0
BronsonAve&SunnySide	0
MerivaleRd&MeadowlandsDr	1
Qweensway&BronsonAve	0
UniversityDr&colonelByDr	1
QweensWay&WoodroffeAve	0
HazeldeanRd&MainSt	0
MainSt&FernbankRd	0
FallowfieldRd&MainSt	0
QweensWay&MoodieDr	0

●
●
●

Figure 20: INTERSECTIONS DATABASE TABLE

In the example, all 5 intersection (MerivaleRd&WestHuntClubRd, MerivaleRd& MeadowlandsDr, FisherAve&MeadowlandsDr, PrinceofWalesDr&MeadowlandsDr, UniversityDr&colonelByDr) are matched and the associated count is incremented. When all members have been analysed the final output will contain all used intersections and the number of carpool members that travel through that

intersection at the specific time selected by the administrator. Please view Figure 18: Statistical Analysis Results for the output of a statistical analysis.

4.2.4.1 *Redefined Paths*

In order to accurately predict what paths carpool members take, 105 main intersections were selected throughout the area of Ottawa. For each intersection a best guess analysis was done to determine what path would be taken to get from that intersection (the members "Nearest Main Intersection" selected during the registration process) to the destination address (Carleton). This path includes a combination of one too many intersections from the list of 105. Please refer to Figure 19: ROUTES DATABASE TABLE for an example of the predefined path of a carpool member whose "Nearest Main Intersection" is "MerivalRd&WestHuntClubRd". There is a total of 105 predefined paths located within the carpool database.

4.2.5 Graphical Analysis

The Graphical Analysis functionality operates the same way as the statistical analysis which was explained in Section 4.2.4 above. The only difference is the output of the information which the administrator can view. In the statistical analysis, the intersections and the number of carpool members which drove through that intersection were displayed in a chart like fashion. The graphical analysis still takes the data located in the INTERSECTIONS table but now displays the results on a map of Ottawa. The concept of displaying the information is basic HTML. The map of Ottawa is the background of the table and the coloured blocks are cells within the table. When the analysis is finished the final count for each intersection is linked to the appropriate cell and displayed. From the

Figure below the numbers associated with each intersection are located on or near the actual intersection location.
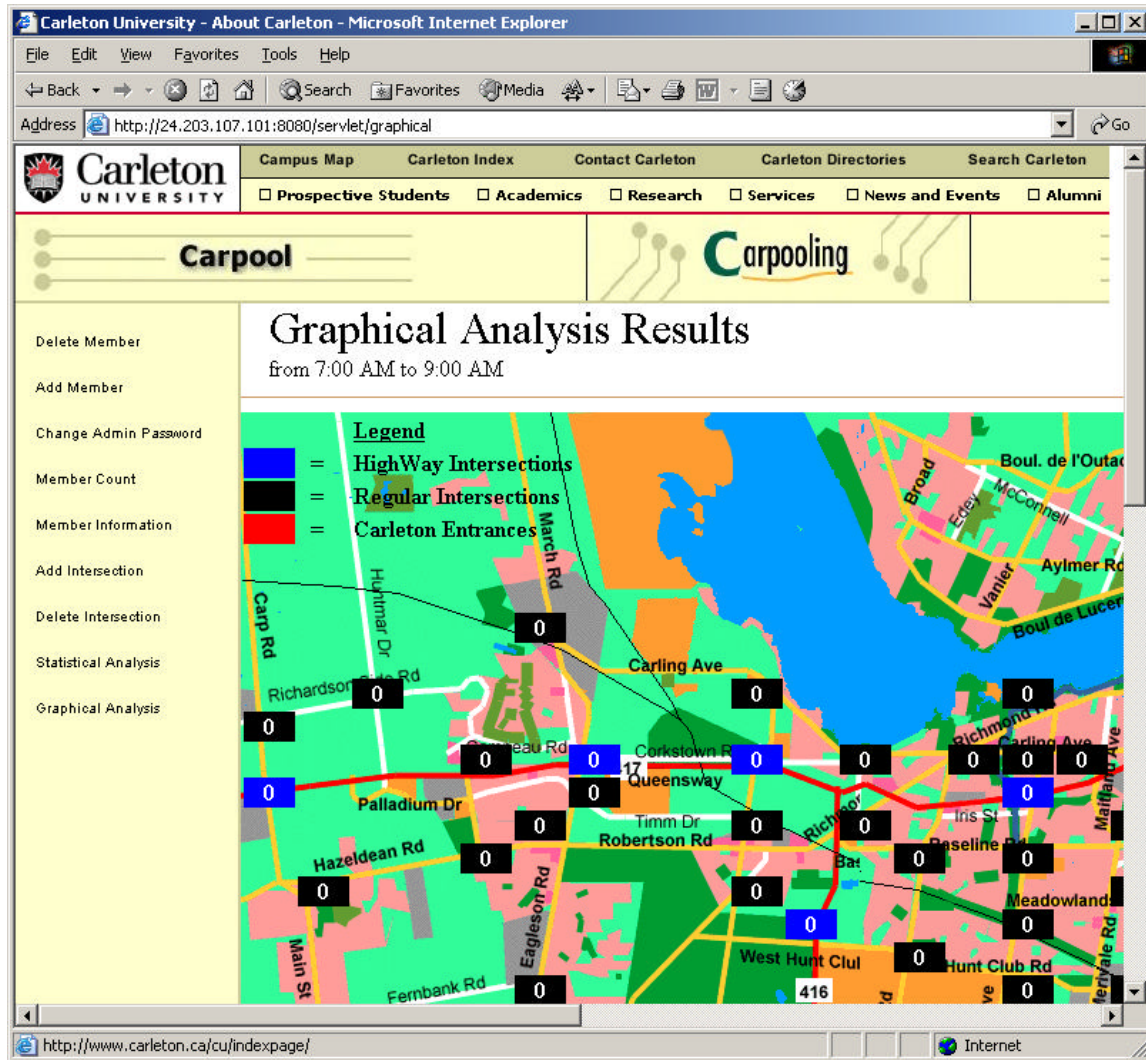

Figure 21: Graphical Analysis Results

This allows the administrator to obtain a better view or understanding as to where congestion occurs and/or the driving habits of all carpool members.

4.2.6 Additional Features

The following sections are additional features that do not directly relate to the overall problem but will help tie all aspects of the solution together to make a complete package. Please view **Error! Reference source not found.** ‐ **Error! Reference source not found.** for design details on the following features.

4.2.6.1 *Sign-in*

In order for a carpool member to access his/her account, he/she must provide the system with a valid username and password. These parameters were set during the registration process. In order for a carpool search to be completed the member must access and use personal information during the search. Therefore, to make this as safe as possible a username and password is used to secure the information from others.

4.2.6.2 *Change Admin Password/Change Member password*

In case the system administrator would like to change the admin password, he/she must provide the current password and a new password to the system interface. If the old password is valid then the new password is updated in the database for the next time the admin signs into the system. If the old password is invalid then the admin has the opportunity to try again.

4.2.6.3 *Add Intersection*

This feature allows the administrator to add an intersection to the INTERSECTIONS table. This will

give the potential carpool members more selection for "Nearest Main Intersections" to choose from.

It is important to note that when adding an intersection the admin must also add the associated path to

the ROUTES table manua**l**y. It is also important to keep the same naming convection when adding

new intersections. For example, **M**erivale **Rd&W**est**H**unt**C**lub**R**d (no spaces) not Merivale Road &

West Hunt Club Road.

4.2.6.4 *List Intersections*

This feature allows the administrator to view a list of "Main Intersections" that currently exist in the

carpool database. Which will help the administrator manage the intersections better.

4.2.6.5 *Add Member*

This feature is the same as the Registration feature, which is explained in detail in Section 4.2.1   It is

only the administrator, that can use the Add Member feature.

4.2.6.6 *View/Modify/Contact Member*

This feature allows the administrator to view how many members are in the database, view contact

information for each member and also enables the administrator to delete a member completely from

the system.

4.2.6.7 *Email Notification*

This feature sends an email to the carpool member after the registration process is complete. The

email contains a welcome message and the username and password of that member. Also whenever

carpool member change their password, this feature will send an email containing the username and

password of that carpool member. If a carpool member forgets his/her password then, he/she can

provide the registered email address to receive the username and password via email notification.

5. CONCLUSION

In the world today there are many problems that affect our environment and inturn our society. One such problem is air pollution caused by gasoline-operated automobiles. Has the world's population gets bigger, more ways are needed to help reduce and control the emissions from gas-fuelled automobiles.

An effective way to help control this problem is to create a carpool system. This enables a group of people to share daily driving times and expenses and inturn help reduce air pollution by decreasing the number of automobiles travelling on roadways. Another way of reducing air pollutants is to eliminate traffic jams. If the travel paths of carpool members were known then this information could indicate congestion points and thus be used to better plan new roadways or even improve current roadways.

This report discussed in detail both solutions as it pertain to a web based carpool system. In summary, the system allows registered members to perform a carpool search to help create a carpool group. The carpool system also enables the administrator to access the database information of each member and perform an analysis that provides traffic flow details about all carpool members.

Along with achieving solutions to the problem, other accomplishments have been made. Knowledge of system networking and database management has been gained and used to aide in the successful completion of the carpool system project.

When the carpool system is implemented into Carleton's computer system, it will indeed help in the reduction of automobiles on Ottawa roadways. As well as provide adequate information about traffic flow.

6. REFERENCES

Carleton University, Systems and Computer Engineering Department
- http://www.sce.carleton.ca/courses/sysc-4907/index.html

University of Waterloo
- http://www.carpooling.ca

Nova Scotia
- http://www.trax.ns.ca/options/carpool-f.html

Commuter Connections
- http://www.carpool.ca

Carpool Canada
- http://www.carpool-canada.com/index.html

Berkeley
- http://public-safety.berkeley.edu/trip/carpool.html

University of Georgia
- http://www.ugasmis.org/carpool/

Nortel Networks
- Intranet

The Apache Jakarta Project
- http://jakarta.apache.org/tomcat/

Java Programming Resources
- http://www.apl.jhu.edu/~hall/java/

MySQL
- http://www.mysql.com/

SQL Commands
- http://www.w3schools.com/sql/sql_delete.asp