Electronic Accident Reporting Form Report

by

David Kunkle
Computer Science Graduate

Western Transportation Institute
College of Engineering
Montana State University – Bozeman

Prepared for

Gary Harkin
Western Transportation Institute
Bozeman, MT 59717

April 30, 2003

ABSTRACT

On Montana Indian reservations, there are many vehicular accidents that are under recorded or under reported. Many reasons could explain this. It may be that the law enforcement officials see no incentive to record their accidents or that they do not have the time. Because accidents are infrequently reported to the state, the Indian tribes are not accessing funds that may help reduce the accident rate.

A proposed project is to automate the entire process of accident reporting. This automation would involve creating an electronic accident report form that would reside in a computer device such as a PDA, laptop or tablet computer that the law enforcement officer would use on site to record information on each accident. Once this accident report form is completed with the accident information, it can be downloaded into a database for storage and analysis. With accident data in the database, the Indian officials then can determine specifically what to send to the state.

My graduate project was to create a prototype of the electronic accident report form and a program that will transfer the data from the report to fill a database. This report will describe the accident report form and the program that fills the database with the accident information. The creation of this electronic form and accompanying program that downloads information to a database was to show the separate Montana Indian tribes the feasibility of such a process.

This project was funded by Western Transportation Institute (WTI) located on the fourth floor of Cobleigh Hall on the Montana State University (MSU) campus in Bozeman, Montana. Professor Gary Harkin, a computer science professor at MSU, was my advisor for this project.

TABLE OF CONTENTS

1. LIST OF TABLES

2. LIST OF FIGURES

3. INTRODUCTION

3.1. Background

There are many vehicular accidents on Montana's Indian reservations. Statistics from the report Montana Indian Fatality Crash Information by the Traffic Safety Bureau of the Montana Department of Transportation, MDT show that traffic accidents among Native Americans are roughly three times that of the non-native Americans.

Along with the high accident rates on Montana Indian reservations, there is some reluctance for these individual sovereign nations to disclose accident information to the state. There could be many reasons for this. Some of the reasons may be that the Native Americans do not understand the accident reporting procedure; they could consider accident reporting time-consuming; and they find no benefit or incentive to record vehicular accidents. There also may be insufficient resources such as money, time, and manpower to fill out these forms. Native Americans may not desire to disclose personal information to the state like names, addresses, registration, licensing or insurance information which they consider private.

Regardless of the reasons, the Native Americans rarely record accidents, and thus, the Native Americans are not accessing safety funds from the state that are a byproduct of sharing with the state their accident numbers, locations and causes of the accidents. By knowing accident locations and causes, the state may be able to implement safety measures on the reservations such as road signs, educational programs and improving the roads themselves to eliminate some of the dangers from driving the roads. These and other safety measures will help reduce the number of reservation accidents.

The Native Americans will also be able to collect reports on the individual accidents on their reservations and to keep them private while at the same time supplying the state with information that benefits both the Native Americans and the state.

3.2. Proposal

The proposal contained two parts that would possibly help alleviate the high accident rates on Montana Indian reservations. The first part of the proposal was to contact the several Indian reservations in Montana and collect information about their present accident reporting and understand why there is a small number of accidents recorded and reported. Also, information would be gathered to see if there was any interest in decreasing the accidents on the reservations and if there was any interest if those involved were shown a means of easily collecting accident information and submitting this or some of the information to the state to access state highway safety funds.

The second part of the proposal was to implement a prototype of an automated accident reporting system that would demonstrate the available technology to record accidents and to be able to keep this information private and held by them. This prototype would also show the different Indian tribes the ease of recording accidents and saving this information.

The automated accident reporting system prototype contains a computer program that resides in a computer device like a PDA, laptop or tablet to record the accident information at the site of the accident. This program will use a graphical user interface for easy data input to an electronic accident report form, ARF, which will be similar to the paper accident report form that is used

presently. Having this ARF program, the user's entry errors can be reduced by catching bad data on data input. This ARF will also make it easier for law enforcement officers to record the details of the accident.

The second part of the automated accident reporting prototype is a program that will take this accident information from the computer device and download it into a database that each individual Indian tribe owns, administers, and manages.

The third part of the automated accident reporting prototype is to develop a means of taking the information from the database and sending it to the state either in a paper format or more preferably in an electronic format that is compatible with the state's database.

This is only the prototype but eventually these programs, the data collection, the transfer of data from the computer device to the database and the means of transferring data to the state will be configurable by the user so that the Native Americans can customize them so that only the data they want to collect and share with the state are acquired and sent.

My graduate project was to develop the ARF for the computer device and to develop the program to take the accident report data and populate a database.

4. IMPLEMENTATION

4.1. Accident Report Form

A Compaq H3850 PocketPC was selected for the computer device to be used in the field. The specifications for the PDA are found in table 1.

Table 1: PDA Specifications

RAM	64 MB
ROM	32 MB
PROCESSOR	206 MHz Intel StrongArm
OP SYSTEM	Microsoft PocketPC 2002
DISPLAY VIEWABLE IMAGE SIZE	2.26 x 3.02 inches

A Teletype GPS unit was selected and used with the PDA to make the recording of the accident location easier and more accurate. The ARF was developed so that it would be faithful in recreating the large paper accident form that was 8 ½ inches by 14 inches large.

Several different implementations were explored to find the most feasible way to create this ARF for the PDA. The first idea was to program the ARF using Microsoft's CE language. The problem with CE was that there would have been a learning curve to understand and use the language and if the program were written in CE it would not be portable to other PDAs that use a different operating system.

Another idea was to use PHP. For PHP to work with the PDA, the PDA would have to be continuously connected to a server for the gathering of data. So PHP was not used. J2ME Java micro edition, which in the mineralized java library used for programs on cellular phones and PDAs was too limiting and is still maturing. The last implementation to run the ARF on the PDA was to load Jeode's java runtime environment on the PDA and develop the java code on a desktop computer and download this program into the PDA. A problem with this was that Java 1.1 was required to write the program and due to its age and capabilities, was inadequate in providing the necessary user interface elements for writing the program

The biggest disadvantage of using a PDA was the size of its screen. If at most three questions were displayed on the PDA screen at a time, there would be at least 40 to 50 different screen images, each with a different set of questions, needed to fill out one accident report. This was inadequate for the amount of information that needed to be gathered.

This prototype had to be built quickly which led to a different approach to implement the ARF. It was decided that the ARF would be developed on a desktop computer and then downloaded to a tablet computer with java runtime on it.

Java version 1.4 was used for the ARF and was developed on Bluej version 1.2.0. Microsoft Access 2000 was used as the database and the JDBC API was used to connect and transfer data from a java program to Access.

Different related sections of the paper accident report form were grouped together and represented as a page in the program. Each page was a Jframe with a scroll pane associated with it. The scroll pane carried the user interface objects such as text fields, choice objects and buttons. Choice objects are pull down lists of elements for the user to choose from. The scroll pane was used so that all the questions for that page could be accessed when the page was longer than the length of the screen. The Grid layout was used to position the user interface objects on the scroll pane.

Action, text and item listeners were incorporated into the program to respond to buttons clicked or choice elements selected.

Images that were created on the form such as the accident, vehicle damage diagrams and some required signatures were saved as JPEG files with the appropriate file names to associate them to this particular accident.

The important ARF pages and their descriptions are listed in table 2 and all the classes used for this program are shown in figure 1.

Table 2: Page Description

Initial	Sets up appropriate tables and number of tables
DriverInfo	Contains Driver Information
VehicleInfo	Contains Vehicle Information
Pedestrian	Contains Pedestrian Information
Cyclist	Contains Cyclist Information
AllInvolved	Information on all people involved in accident
AccidentDescription	Information Area for relevant info not in other pages
AccidentLocation	Contains location or accident in detail
AccidentIdInformation	Contains city, county codes
Help	Data entry instructions and description of calling page
Error	Indicates input errors on page

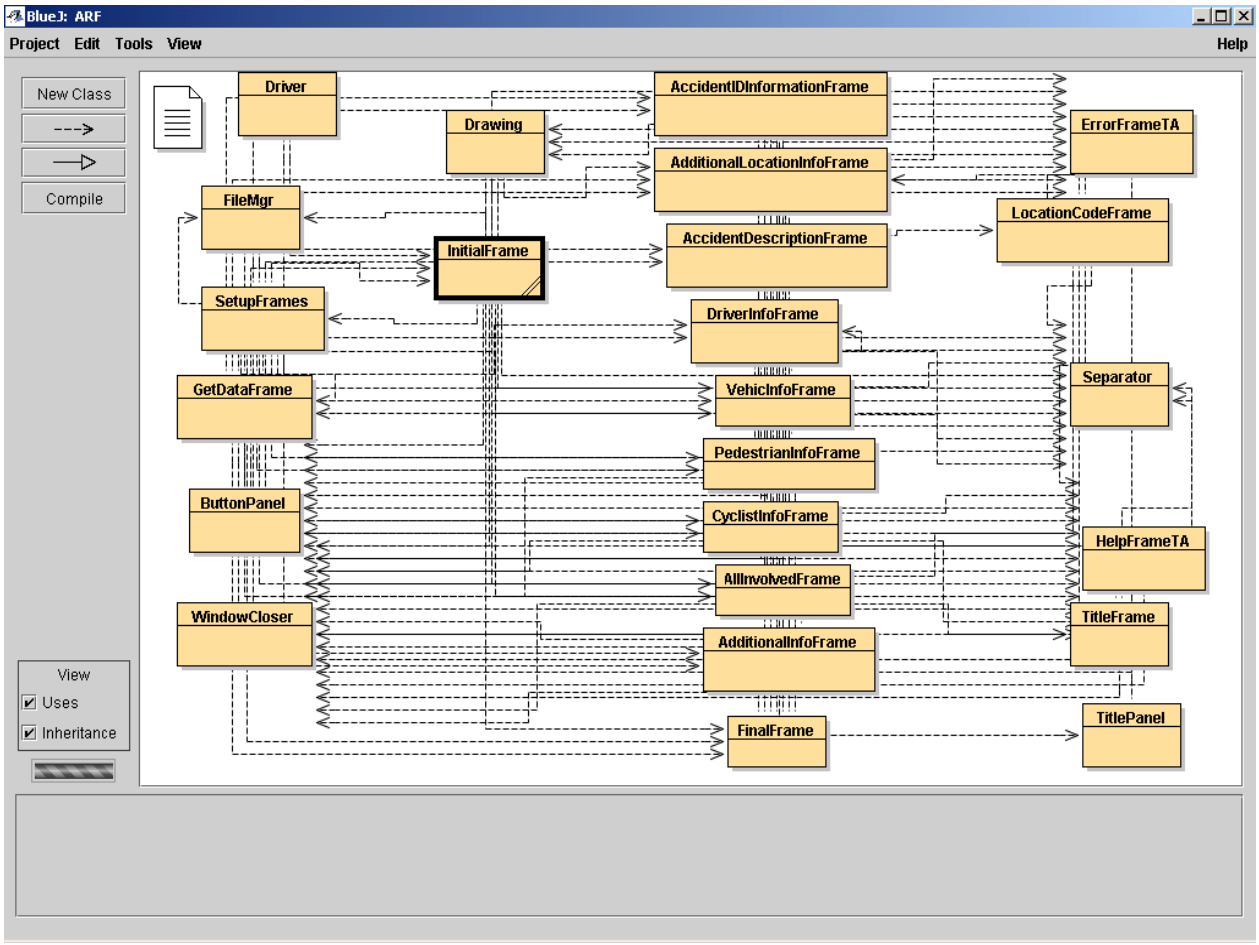

Figure 1: ARF Pages and Associations

The initial page contains questions such as report number for this month, number of vehicles involved, total number of people involved with the accident, number of cyclists and pedestrians. When the next button at the bottom of this initial page is clicked, the program in the background creates the correct number and types of pages that need to be built from the data inputted on this page, and then all the pages to the report are linked sequentially and in the proper order For example, if the user indicated that there were three vehicles involved in the accident then three separate driver information pages would be built and labeled vehicle one, vehicle two and vehicle three.

Other than the initial page, all the other pages in the report have a next, previous and help button at the bottom of each page. The previous page button returns you to the previous page that was filled out in the report. The next button if clicked will take the user to the next page in the report. If there were errors in the current page and the next button is clicked then an error page appears and tells the user the errors that need to be corrected before the user can continue to the next page in the report. Once these errors are fixed, then the user can go to the next page in the report. An error on the page for now is if the user leaves a question on the page unanswered.

The help button when clicked will take the user to a page which describes the entries for that page and how to properly fill those entries. Hitting the return button on the help frame will take the user back to the page that originally called that help page. Each of the report pages has a

help page that is unique to that report page. The help information on the help pages comes from the instruction manual that accompanies the paper ARF.

Some pages of the ARF have buttons that will take the user to another page that requires a signature or a diagram to be drawn. For example, on the vehicle information page, there is a button that takes the user to the vehicle damage page. On the vehicle damage page a vehicle must be marked by an 'X' to indicate where the damage to that vehicle was. Once the diagram is finished, the user clicks the save button. Clicking on the save button will save the image just marked on and give it a unique file name which includes the accident number for that month and the vehicle number for that accident report. After the program creates this new file and saves it, the program returns the user to the page being worked on before the vehicle damage page was called.

When the user gets to the final page and answers all the questions on the ARF, the user will hit the submit button and leave the program. When this submit button is clicked, the AFR program will create a file with a unique name for that accident report and save all the data entered by the user into this file sequentially. The vehicle page in figure 2 illustrates a view of a typical page from the ARF.

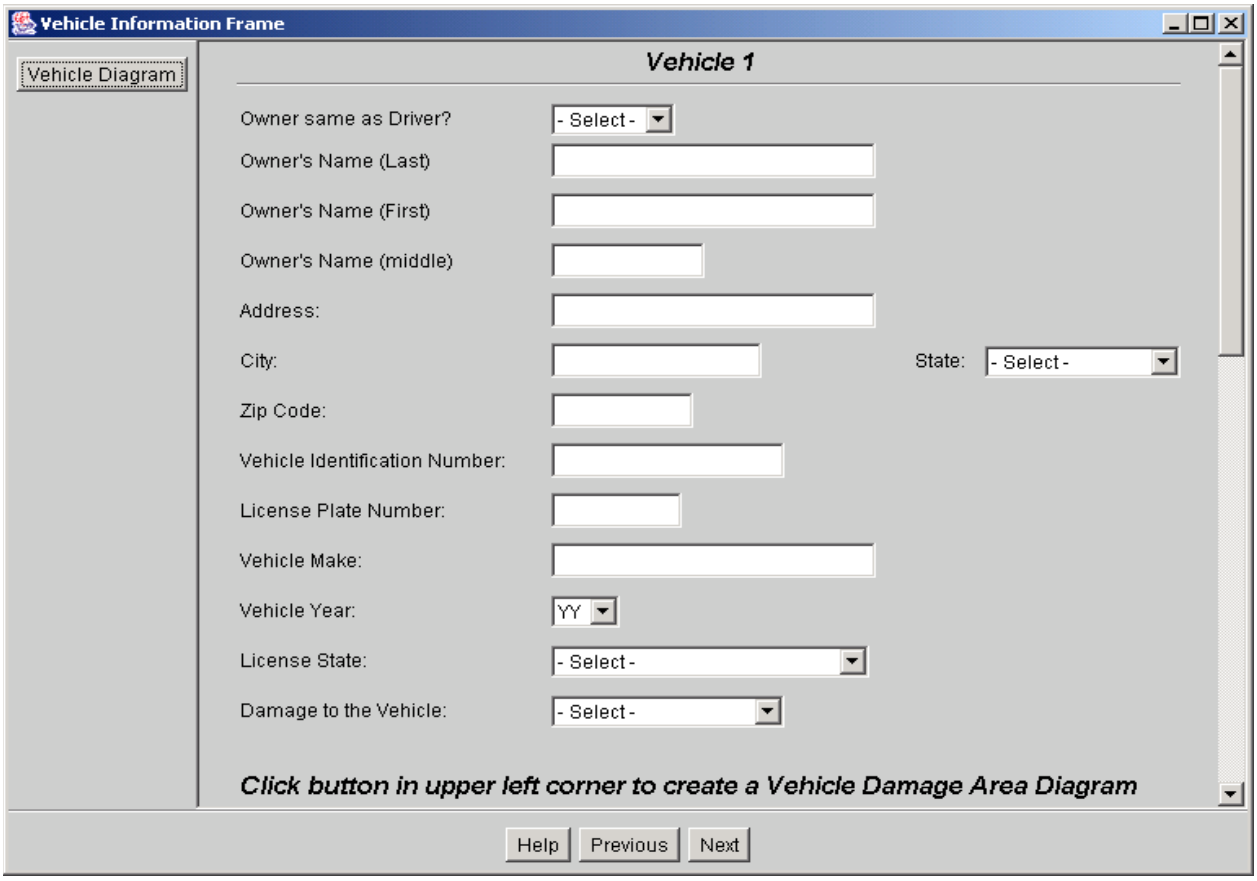

Figure 2: Vehicle Page

4.2. Data to Database Program

The next program that was created was called dataToDB. When the law enforcement officer returns to the office or to where the accident database is maintained, the officer will connect the electronic device that contains the files that were created from the ARF to the database where the accident information is stored. The user will then start this program that prompts the user for the accident number that is selected in order to download to the database. When the user enters the accident number and hits return, the program will open the file for that accident number and sequentially read data from that file and load the database. The classes for this dataToDB program were also developed in Bluej and are shown in figure 3.

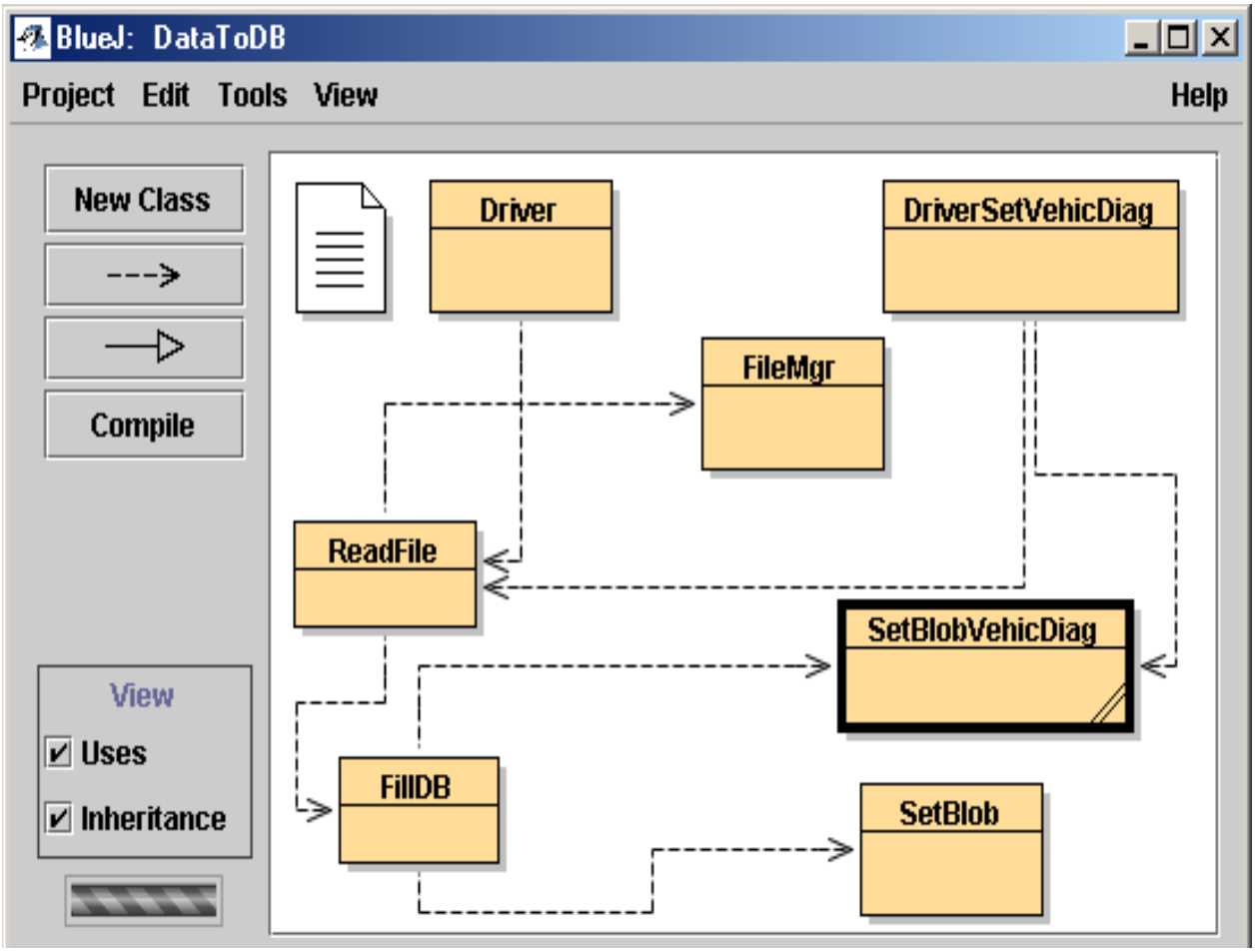

Figure 3: Data To DB Program

The driver object starts this program and prompts the user for an accident number. When the user responds with an accident number, the program calls the ReadFile class that opens the proper file, creates string arrays to hold the data that will be retrieved from the file and then fill the arrays with the data from that accident. String arrays were created that corresponded to the pages of the ARF. For example, when the ReadFile object is called it creates a vehicle array to temporarily store the vehicle data collected in the ARF. Likewise a driver array is created to

temporarily store the vehicle driver data that was collected from the ARF. These tables also corresponded to a table in the database. Having this correlation between the data arrays and tables in the database made the program simpler to design and execute.

The FillDB object when called uses the Java Database Connectivity (JDBC) API to connect to the database and send the data to the database. The FillDB object would then call the SetBlob object to insert the ARF's signature, accident and XY images and call the SetBlobVehicDiag object to insert the vehicle diagrams of that accident. A separate table was created to store all the ARF images called VehicleImages. The VehicleImages table uses the OLE Object datatype for all image record attributes. An OLE object for Access can hold long binary data. The Jpeg format of the images had to change to a ByteArrayInputSteam data type for the image to be inserted into the database. To do this, the data from the jpeg file was read into a byte array. Then this byte array was used as a parameter to a ByteArrayInputSteam object that the database would accept with the executeUpdate command. Another program called GetImage was created that would verify by retrieval that the ARF images were being properly loaded into the database.

The Access 2000 database was designed with tables that corresponded to the pages in the ARF with the accident number being the primary key to most of the tables. An exception to this was on tables that had multiple entries for that accident such as the vehicle and driver tables. For tables with multiple entries for an accident such as the driver table, the primary key would be the accident number and the number of the driver. The vehicle table has the accident number and the vehicle number for its primary key.

5. CONCLUSIONS

This automated accident report system prototype was developed and run on a desktop computer that contained both the ARF and the Access database. This made the connection from the data files produced by the ARF and the database simple. In actuality, some other means of linking the remote recording device to the database should be implemented. Also since the programs for this automated ARF system were developed on a desktop computer no GPS was purchased. Because there wasn't a GPS unit purchased for the desktop, there was no need to incorporate code into the programs that would take the location data from the GPS and fill the location entries in the ARF. Also, due to the lack of funds left for this project, no tablet computer was purchased to complete this phase of development.

For rapid development, MS Access 2000 was used as the database for this project because of its familiarity to the developer. Later in development, it was found that Access did not meet SQL standards and made inserting Jpeg images into this database difficult to code and was not dependable for extracting the image.

Once this prototype was developed and running, it was presented to the Blackfeet Indians and to a BIA representative for the Western Region based in Portland, Oregon. There was some interest from the BIA to continue funding this project and build it to their specifications. But since demonstrating this project to the BIA representative, the agency's interest has waned.

6. RECOMMENDATIONS

The project goal was to build a prototype to demonstrate to the Montana Native Americans an accident reporting system and give them an idea of how this automated accident reporting system would work. Until there is some feedback on the usability of this automated accident reporting system from the Native Americans, there is no need to develop some of the features that were excluded. These excluded features can be integrated into the project at a later date if there is any interest to continue this endeavor.

Some suggestions for further development of this project are to use a different database other than MS Access 2000 that does not conform to SQL standards. MySql is a free relational database that complies with SQL standards better than MS Access does.

Another recommendation is the development of a program that would create a user defined ARF that collects the accident data that the user defines from a list of questions from the standard ARF. Likewise it is suggested to develop a program that would generate a file of accident data that could be sent to the state in a mutually agreed upon format. This program would allow the user to select data from a list of accident data items that the user is comfortable giving to the state.

Another suggestion is to develop a program that will allow the user to download the information collected from the accident through the Internet from a remote site. This would alleviate the restriction that the user needs to download the accident information at the physical location of the database.

7. REFERENCES

http://java.sun.com/docs/books/tutorial/jdbc/basics/index.http

http://java.sun.com/docs/books/tutorial/#guis

Horstman, C. Core Java, Volume 1 – Fundamentals, Sun Microsystems Press, 1999.

Reese, G. JDBC and Java, O'Reily, 2000.

Ringel, C, Traffic Accident Reporting on Indian Reservations in Montana, Power Point Presentation, 2002.

Traffic Safety Bureau of Montana Department of Transportation, Montana Indian Fatality Crash Information, 2000.