Safety and Operations of Large-Area Rural/Urban Intermodal Systems

SOLARIS

A Tier 1 university transportation center led by the University of Nevada, Reno.

Final Report

12/31/2016

In-depth investigation of the system currently used by the Las Vegas Metropolitan Police Department to store and process crash data and all other interconnected systems

SOLARIS Consortium, Tier 1 University Transportation Center

Center for Advanced Transportation Education and Research

Department of Civil and Environmental Engineering

University of Nevada, Reno

Reno, NV 89557

UNLV | Howard R. Hughes College of Engineering

Alexander Paz, Ph.D., P.E.

University of Nevada Las Vegas

DISCLAIMER:

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

UNLV | Howard R. Hughes College of ENGINEERING

Page intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Some of the existing software and hardware used by law enforcement agencies to collect crash data are obsolete for several reasons, ranging from budget constraints to lack of coordination across various groups. The most significant consequence of using obsolete tools are location errors, which preclude the correct use and reliability of the data. In addition, the time required for law enforcement agents to be at the scene could be lengthy, especially to collect data adequately.

Accurately locating crashes is key to geographic analyses of crash statistics and patterns as well as for the development of safety recommendations for crash 'hotspots'. Generally, a complex and involved process is required to locate crashes and collect relevant data on public roads, using text formats and hand drawings. Many crashes are unlocated or misallocated, and the data is hard to register. The main impediments to locate crashes accurately and collect crash data are well known, and include errors in data entry, street name errors by the recording officer, the existence of alias names, and county coding errors as well as many other factors.

Crash data can be analyzed to study the incidence of the various factors in crashes; for example, information of events involving drivers under the influence of alcohol (DUI) can be used to support decision making. The methodologies for traffic safety management recommended by the *Highway Safety Manual* (AASHTO, 2010) require accurate crash and location data (Paz et al., 2015), that currently is collected by law enforcement agencies. This data is needed for performance-based traffic safety programs as well, and must be prepared by state agencies to address requirements from the legislators (NCHRP, 2010; FHWA, 2013).

To address data-collection issues and provide better technology for law enforcement agents, the Transportation Research Center (TRC) at the University of Nevada, Las Vegas (UNLV) has developed and implemented a system for the accurate and efficient collection of crash data, including location. The proposed solution uses a Geographic Positioning System (GPS) and a Geographic Information System (GIS) to geolocate crashes and provide a map-based data-collection environment. Compared to existing processes and technology in use, this system greatly reduces the time and resources involved in consistency checking and error correcting during data collection.

The proposed system was developed with help from various law enforcement agencies in Nevada. Considering the challenges associated with collecting location information as well as the data needs of various stakeholders, in addition to the geospatial coordinates of the crash, the proposed system includes a scene diagram that captures screenshots of the crash location by using a GIS map as well as screenshots of a map where the crash has been located by the police officer.

The development, implementation, and testing of the proposed system included continuous interaction between users and developers in order to take full advantage of field experience and associated needs (Racheva and Daneva, 2010). This ensured that the expectations and needs from law enforcement agencies and data users were fully addressed.

2. PROPOSED SYSTEM

The system involves a server that hosts a geospatial database, a mobile application, and a web portal, as shown in Figure 1. Law enforcement agents collect crash and citations data by using the mobile application. The data is sent in real time to the geospatial database hosted by the server application, which makes the data available by means of the web portal. In addition to a website, the web portal offers a REST-API (Fielding, 2000) web-service endpoint, which allows external systems to extract raw or aggregated information. This web service endpoint was built using Open Data Protocol (OData, 2010).

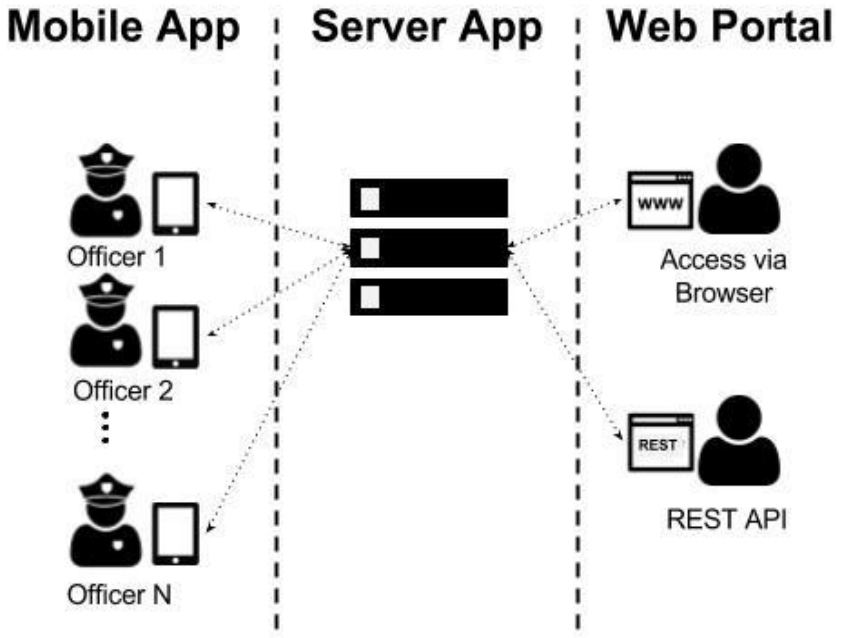

Figure 1: Conceptual illustration of the proposed system.

The primary requirements of the proposed solution include:

1. Accuracy of the location information;
2. Efficiency to minimize the time required by the agent to be in the field;
3. Flexibility to navigate through menus;
4. Synchronization across crash and citation data, when required;
5. Capabilities to create a scene diagram powered either by a map or using a freehand sketch view;
6. Capabilities to attach all types of files to reports, including pictures, and a screenshot of the crash location;
7. Capabilities to read information from driver licenses and vehicle registrations by using a barcode reader, and
8. Querying capabilities by using a web portal to generate key graphs, charts, and reports.

In addition, it is desirable that a data collection system considers real-time statistics, ease of access to the data, data completeness, and safety of the data collector, among other primary

issues. A description of how the proposed solution addresses these issues is provided in this report.

The proposed system is being tested by law enforcement agencies in Nevada. Therefore, the system was implemented in compliance with the standard data dictionary for crash and citations information for the State of Nevada, Nevada Citations, and Accident Tracking System (NCATS) (NHTSA, 2010).

2 DATA ACCURACY

Although all collected data is important, currently, the most challenging issues focus on information about the location and the scene. The proposed system allows on-site location capture. Figure 2 illustrates the user interface that allows location capture. The coordinates and corresponding address for a crash or citation, if existing, is provided by GPS and displayed on the screen. If the accuracy of the GPS is sufficient or the data collection device is located away from the crash location, the agent can use the touch-screen map to set the correct location.

Figure 3 illustrates the user interface that enables the construction of a crash scene diagram. This interface shows, by default, the GPS location of the mobile device hosting the mobile app. In order create a realistic and detailed representation of a crash, the interface allows the agent to drag, with his or her finger, all the elements involved in the crash scene. In addition, the agent can create a freehand sketch of the scene, and take an unlimited number of photos of the crash scene.

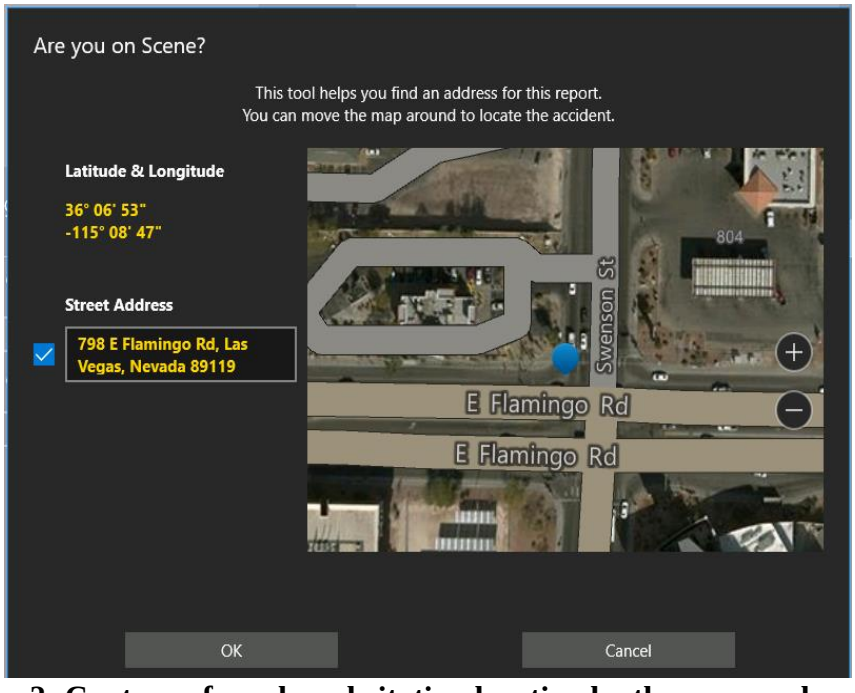

Figure 2: Capture of crash and citation location by the proposed system.

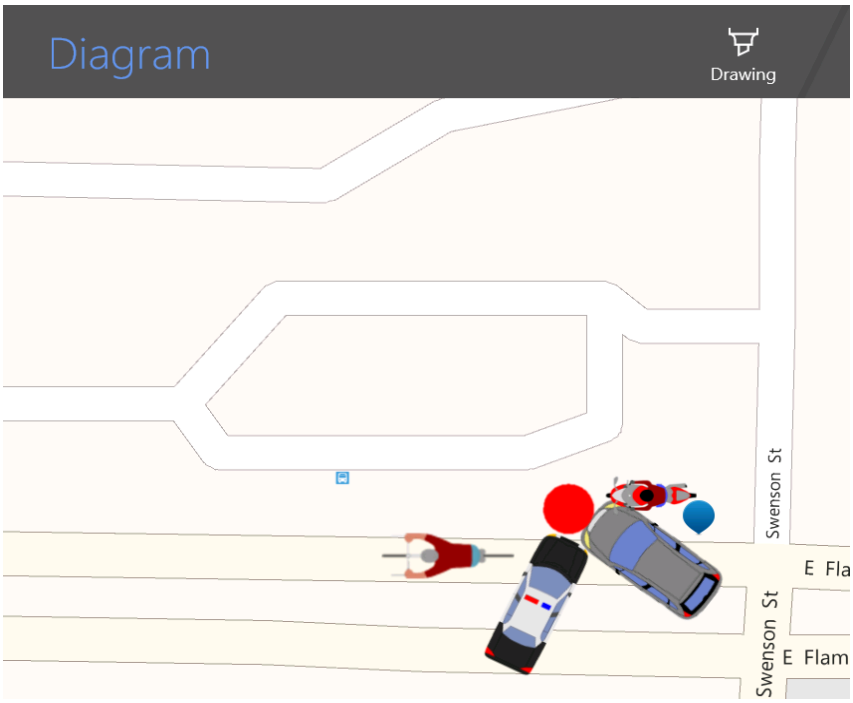

Figure 3: Crash scene diagram.

3 REAL-TIME COLLECTION AND REPORTING

By means of the cellular network, the data collected is sent automatically to the server in real time on a regular basis. This was designed to minimize the risk of losing data because of a special event, such as the loss or damage of the mobile device. If network connection is not available, the data is stored in the mobile device until the connection is reestablished and the data is completely received by the server.

Once in the server, the data can be aggregated and filtered in real time to generate reports. Figure 4 shows how the system provides color-coded locations and the status of crashes, based on how old they are, and/or an applied filter. This type of information could be important for real-time operations and tactical decisions by managers and supervisors. Figure 5 illustrates how a heat map is used that allows the user to zoom into specific zones and apply filters to analyze crash patterns in detail.

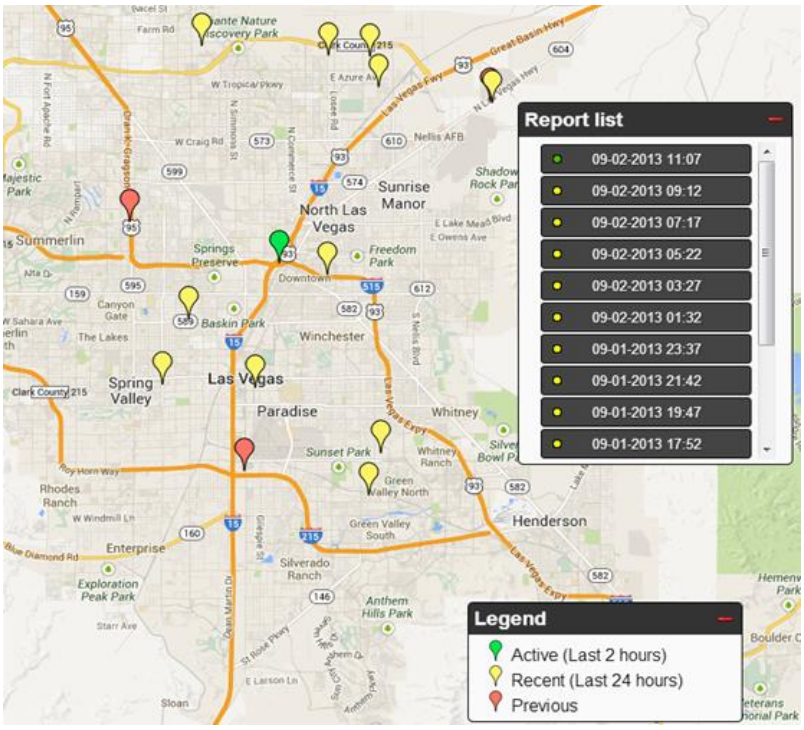

Figure 4: Crash location and status.

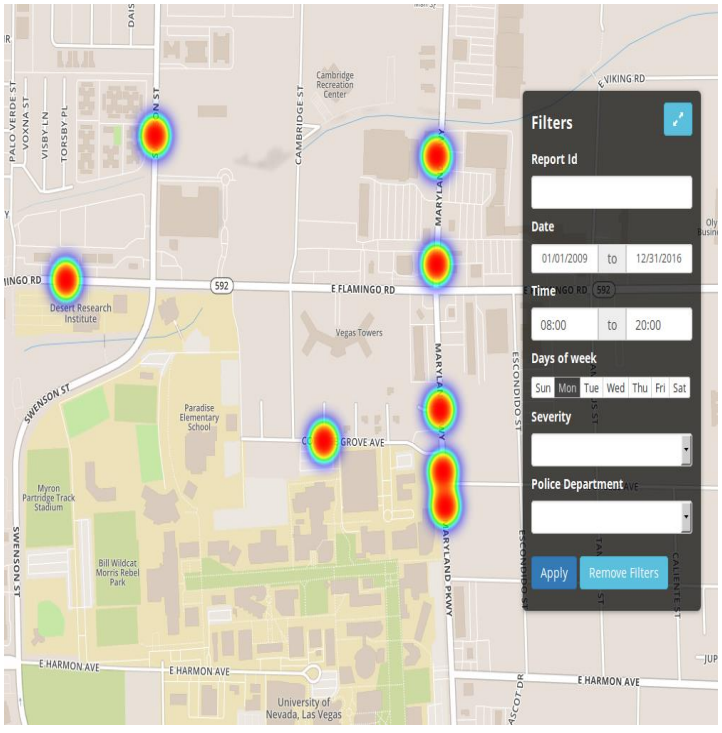

Figure 5: Heat map of crash densities.

Figure 6 illustrates the density of crashes by the day of the week and the time of day for a selected year. By default, this summary includes the entire number of crashes in the system; in addition, it can be filtered by the severity of the crash (i.e., injuries, fatalities, property damage). Other available filters and diagrams include severity of crashes, involvement of drugs, and gender or age.

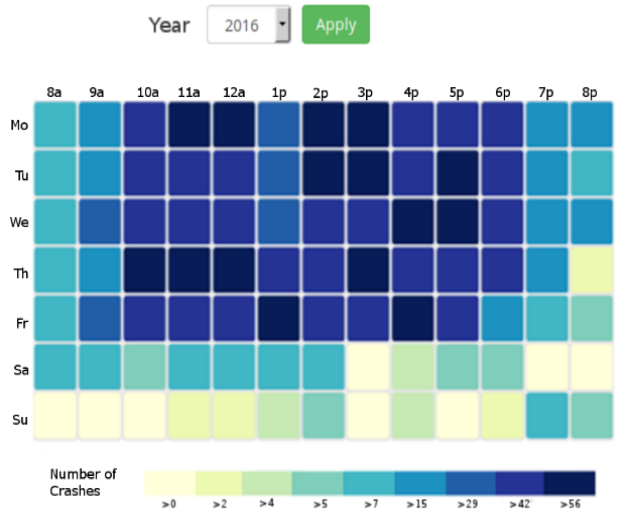

Figure 6: Crash density by the day of the week and the time of the day.

4 EASE ACCESS TO DATA

The data collected is available online, and can be accessed easily through the web portal or the REST-API. For the web portal, an account is required to register, modify, or aggregate the data. A reporting feature has been included in the proposed system to allow exporting aggregated or disaggregated reports in 1) common formats for data interchange, such as XML, CSV, and JSON; and 2) common document formats, such as PDF, DOC, and XLS. If desired, some of this information can be made available for public access, including special statistics.

If required by government agencies or interconnected systems, this proposed system offers a REST-API with OData protocol implementation. The use of an OData implementation makes information retrieval of linked entities easier (Carey, 2012). The access to the REST-API is available only for authenticated users.

5 DATA COMPLETENESS

Law enforcement agencies have requirements concerning the collected information. Data dictionaries are designed to standardize the structure and codification of the information. For illustration purposes, the proposed solution for this system implements all the required fields that are specified in the NCATS data dictionary, version 2010.

Implementations using a different data dictionary requires changes to the system, which could be minor or large, depending on the differences with NCATS. However, all important and

significant capabilities, such as the collection of location information and scene diagram, require no changes to the system unless special needs or upgrades are demanded.

In order to minimize the risk of mistyped information, and reduce the effort required to manually type in the data, such features as barcode readers for driver license and vehicle registrations were implemented in this proposed system.

6 SAFETY OF THE AGENT

It is very important to minimize the time that the agents are exposed to traffic and danger. The proposed system was designed to minimize the time required by an agent to collect field data. In critical or urgent situations, an agent can collect on-site critical information rapidly by using a minimum number of touches to the screen on the GPS device, the camera, and the barcode reader. Later, when in a less risky environment, the agent can complete the rest of the data collection by using the mobile application or the web portal.

Table 1 shows estimated time to collect critical information for a typical crash scenario involving two cars and two occupants. This estimation assumes that the agent has access to the drivers' licenses and vehicle registrations.

Table 1 Estimate time to collect critical crash data

Information Item	Time (seconds)
Location	5
Vehicles	12
Drivers	12
Pictures	10
Total	**39**

2. RESULTS

Multiple meetings with law enforcement agencies in the State of Nevada have been conducted to review the design and implementation of the proposed system. Law enforcement agents have acknowledged that the mobile application provides enough accuracy to capture location information. They agreed that the crash-scene diagram tool provides all the necessary elements to create a reliable representation of the scene. In addition, they have suggested usability improvements, such as pre-filled values and 'favorites' lists, for commonly used fields.

Administrative staff has validated compliance of the collected information with the NCATS data dictionary. It has been confirmed that the reports and statistics generated in the web portal contain the required information. Additional statistics and reports have been suggested by administrative staff to increase the benefits of the reporting tool.

Feedback has been collected from multiple agencies, and new features have been added to the proposed solution based on their suggestions. A field test – to be conducted by law enforcement

agents from the Nevada Highway Patrol – is scheduled within the next few weeks. Results from the field test will be used to improve the system, and make decisions to upgrade the existing tools currently in use in Nevada.

Appendix A provides important technical information about the implementation of the proposed system. Appendix B provides information about various meetings with law enforcement groups in Nevada. The information focuses on comments provided by enforcement personnel who have experience in crash data collection and the corresponding actions taken during implementation of the proposed software system.

3. CONCLUSIONS

The developed system is able to collect crash data *in situ* and store it in a geodatabase. Data that is collected is characterized and processed in real time to generate reports, maps, charts, and statistics. The proposed data collection system facilitates the data collection while saving time, reducing errors, and enabling the collection of the more valuable information from crashes, such as the scene diagram. The proposed system is the result of a combined effort involving law enforcement agencies, the Nevada Department of Transportation, and the University of Nevada, Las Vegas.

Future work includes the development of additional performance measures. A key capability required by law enforcement is the ability of multiple officers to be able to work on the same report at the same time. In addition, certain capabilities are desirable when generating collision diagrams and visuals for such statistics as expected crash frequencies and the rankings of sites based on them.

ACKNOWLEDGMENTS

This study was sponsored by the Nevada Department of Transportation and the Federal Highway Administration. Special thanks to Major Thom Jackson and Trooper Nicholas O'Conner from the Nevada Highway Patrol for their guidance and recommendations. Similarly, many thanks to Lieutenant Brandon Brooks from the Henderson Police Department and Lieutenant Leonard Marshall from the Las Vegas Metropolitan Police Department for their support and provided information. Many thanks to the Technical Writer of UNLV's Howard R. Hughes College of Engineering, Julie Longo, for editing this manuscript.

4. REFERENCES

Carey, M.J., Onose, N. and Petropoulos, M., 2012. Data services. *Communications of the ACM*, *55*(6), pp. 86-97.

Fielding, R.T., 2000. *Architectural styles and the design of network-based software architectures* (Doctoral dissertation, University of California, Irvine).

Manual, H.S., 2010. American association of state highway and transportation officials (AASHTO). *Washington, DC*, *10*.

National Cooperative Highway Research Program, Cambridge Systematics, American Association of State Highway and Transportation Officials, 2010. *Target-setting Methods and Data Management to Support Performance-based Resource Allocation by Transportation Agencies* (Vol. 666). Transportation Research Board.

Nevada Highway Traffic Safety Administration. (2010). *Nevada NCATS Data Dictionary*. [online] Available at: http://www.nhtsa.gov/nhtsa/stateCatalog/states/nv/nevada.html [Accessed 23 Nov. 2016].

National Highway Traffic Safety Administration. (2010). *Nevada NCATS Data Dictionary*. [online] Available at: http://www.nhtsa.gov/nhtsa/stateCatalog/states/nv/nevada.html [Accessed 23 Nov. 2016].

OData. (2015). *OData - the Best Way to REST*. [online] Available at: http://www.odata.org [Accessed 23 Nov. 2016].

Paz, A., Veeramisti, N., Khanal, I., Baker, J. and de la Fuente-Mella, H., 2015. Development of a comprehensive database system for Safety Analyst. *The Scientific World Journal*, *2015*.

Racheva, Z. and Daneva, M., 2010. Clients' participation in software projects: comparative case study between an agile and a 'traditional' software company. *first Workshop on Leveraging Empirical Research Results for Software Business Success*.

APPENDIX A:

Microsoft Visual Studio Community Edition, Version 2015, was used for the development of the entire solution. The GIT versioning system was used to manage collaborative work.

Figure 7 provides a high-level overview of the proposed architecture. The gray tags below each component represent the corresponding technology used for implementation. A more detailed explanation of each components is provided below.

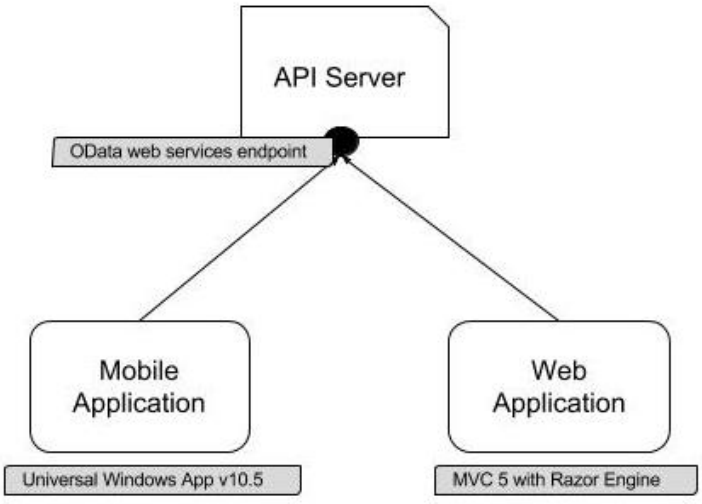

Figure 7: Conceptual illustration of the proposed system.

7 API SERVER

The API server implements all the business logic of the system. It is implemented using C# programming language with .Net Framework version 4.5.2. The interaction with the client is performed by a web services endpoint (OData) using a standard HTTP protocol. The proposed system uses Microsoft Entity Framework 6 for the Object Relational Mapping (ORM) between the software and the database. The connection with the database is performed by means of a standard TCP/IP.

Figure 8 illustrates the components of the API server and their interactions. An explanation of each of the components is provided below.

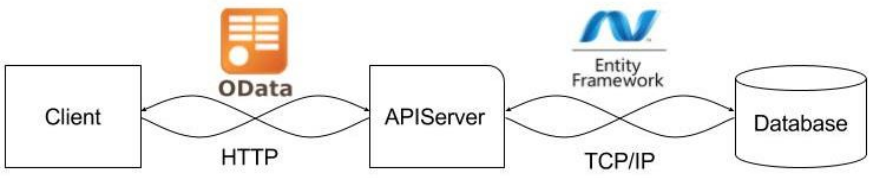

Figure 8: Conceptual illustration of the API server.

8 ODATA (OPEN DATA PROTOCOL)

OData is a standard created by the company OASIS® that defines a set of best practices for building and consuming RESTful APIs. OData focus mainly on business logic while building RESTful APIs without issues regarding the technology. It abstracts and makes transparent to the developer the definition of request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc.

The most recent and stable version, OData v4, was used in this project. Microsoft possesses a set of libraries tailored to work with OData. These libraries integrate OData with standard LINQ (Language Integrated Query), which is a Microsoft .NET Framework component that adds native data-querying capabilities to .NET languages.

Each entity in the database contains an OData controller with the basic Create, Update, Delete (CRUD) and business logic operations.

The standard URL is illustrated in the following snippet of code:

```
http://server:port/api/{controller}({id})
```

For example, if a citation with Id=28 is required, the URL should be written as:

```
http://server:port/api/Citations(18)
```

9 MICROSOFT ENTITY FRAMEWORK

The Microsoft ADO.NET Entity Framework is an ORM framework that enables developers to work with relational data as domain-specific objects, eliminating the need for most of the data-access plumbing code that developers usually need to write. Using the Entity Framework, developers issue queries using LINQ, and then retrieve and manipulate the data as strongly typed objects. The Entity Framework's ORM implementation provides such services as change tracking, identity resolution, lazy loading, and query translation. In this way, developers can focus on their application-specific business logic rather than the fundamentals of data access.

In the current project, the creation and edition of the models is performed through C# code, and is applied to the database with Entity Framework migrations. No direct manipulation of SQL code has been performed; instead, everything is achieved by means of entity framework commands.

10 DATABASE

The database consists of 11 tables. These tables and their relationships to one another are depicted in Figure 9. The white diamond with a dashed line leading to a black circle represents a many-to-one relationship. The white diamond leading to a filled circle represents a one-to-many relationship. For example, a single vehicle may have many occupants, but an occupant can only belong to one vehicle.

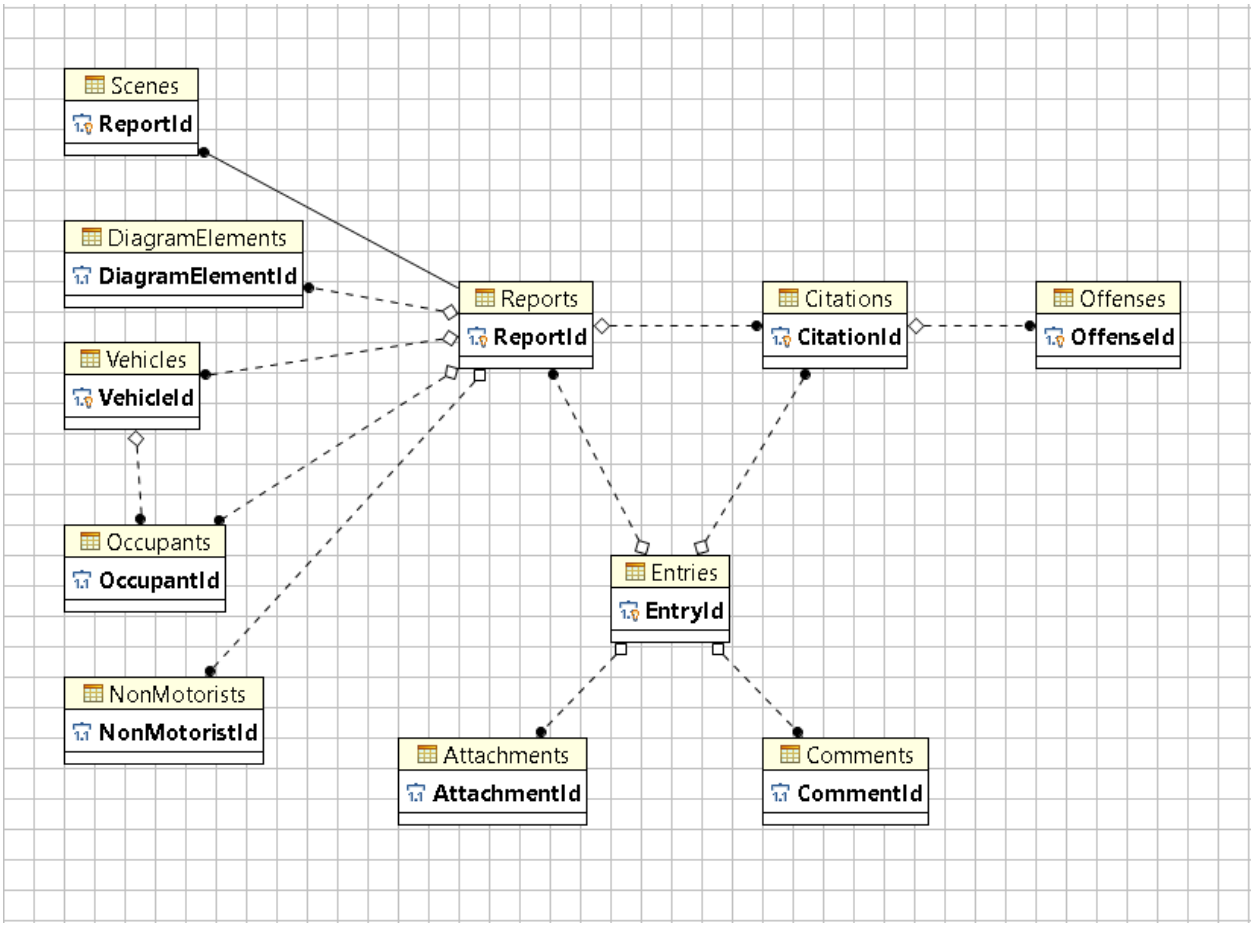

Figure 9: Database table relationships.

11 MOBILE APPLICATION

The mobile application keeps a local state, and synchronizes the required information with the API Server.

12 LOCAL DATABASE

The local database is implemented using SQLite, a relational database management system, and object-relational mapping (ORM) was performed with Entity Framework 5. Contrary to many other database management systems, SQLite is not a client-server database engine; instead, it is embedded into the end program. This means that once the mobile application is uninstalled from the device, the local database also is removed.

The entities that exist in the local database were not created from scratch. Instead, they were inherited from entities that are in the server. Figure 10 illustrates this concept.

```
[Table("Scenes")]
public class SceneModel : LVMPD.Server.Models.Scene
```

Figure 10: Local entity inherits from the sever.

13 VIEWS

All views in the application were created using Microsoft Extensible Application Markup Language (XAML). The views all followed theme-based styles and, where appropriate, used adaptive triggers to handle different screen sizes.

14 BARCODE READER AND GPS

Both the barcode reader and the GPS were implemented using the Universal Windows Platform (UWP) standard libraries. Upon the first use of the mobile application, the system will request permission to access the user's GPS, barcode reader, and camera.

15 MAPS

The mobile application uses Microsoft's Windows.UI.Xaml.Controls.Maps namespace to implement its maps. Offline maps are provided by the map application of Windows 10.

16 WEB APPLICATION

The web application was implemented using Microsoft MVC5, with a razor view engine. The web application does not read directly from the database. Instead, it extracts all the data from the API server through the OData web services endpoint.

17 MODEL VIEW CONTROLLER (MVC)

A design pattern can be separated into three parts, the Model, the View, and the Controller. The model represents only the data, and nothing else. It is not dependent on either the controller or the view. The view is how the user sees the data contained within the model, and sends the user's actions to the controller by means of buttons and other actions. The controller interprets the user's actions from the view, and changes the model's data accordingly. The controller is dependent on both the view and the model. Figure 11 illustrates these relationships.

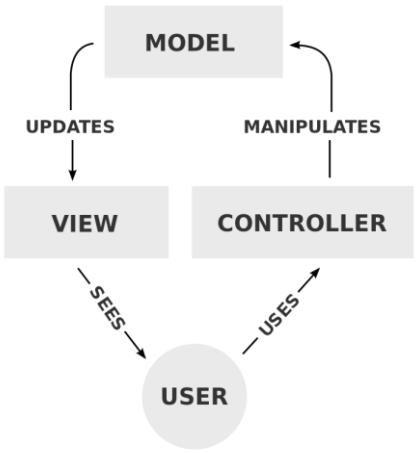

Figure 11: Diagram of the Model View Controller.

The purpose behind this design is to reduce software complexity and make the existing code more maintainable. The MVC design pattern also allows for reusable parts without modification. For example, the web application uses a partial view – a view that is rendered within another view – to implement attachments for both reports and citations. Considering that the implementation of attachments is identical for reports and citations, the corresponding code also is identical. Instead of having the same code in two different places within the application, it can be extracted into a standalone partial view and referenced in the parent views for reports and citations. This provides the advantage of being able to update a single partial view for attachments and have those changes reflected in two locations on the website. This approach was used for comments as well.

18 ASYNCHRONOUS JAVASCRIPT AND XML (AJAX)

AJAX enables several useful features for a web page, including, among others:

- Update of a web page without the need to refresh,
- Request and receive data from a server after the page has been loaded, and
- Send data to a server in the background.

AJAX must be used to be update comments and attachments dynamically when a user makes changes to a view in real time, such as uploading an attachment. Given that records in a table are updated only once when a page is loaded, an AJAX request is made to append the newest attachment or comment to the appropriate table.

In addition, AJAX is used to apply filters on all of the maps of the application dynamically. This was done by editing and placing map layers on top of the base map in order to add or remove plot points on the map surface. Based on the user specified filters, AJAX sends a query to the server to get the data that will be used in the map layer to be added onto the base map.

19 LEAFLET

Leaflet is an open-source Javascript library used for mobile-friendly interactive maps. It was designed to provide simplicity, performance, and usability. In this project, Leaflet was used to create all the maps on the web portal. In addition, it was used for all map icons as well as the heatmap.

20 BOOTSTRAP

Bootsrap is an open-source front-end web framework for designing web applications and websites. It has HTML- and CSS-based templates for buttons, forms, navigation, and other interface components. The web application in this project uses many of Bootstrap's styles, buttons, and themes for its appearance.

21 CHART.JS JAVASCRIPT LIBRARY

Chart.JS was used to create HTML5 based Javascript charts. On the statistics page of the web application, Chart.js was used for the pie charts.

22 D3.JS JAVASCRIPT LIBRARY

Using HTML, SVG, and CSS, D3.js is a Javascript library for manipulating documents based on data. On the statistics page of the web application, D3.js was used to create the Calendar View and Days-Hours Heatmap.

APPENDIX B: FEEDBACK FROM LAW ENFORCEMENT AGENCIES IN NEVADA

The following, comments were provided by the Nevada Highway Patrol (NHP), the Las Vegas Metropolitan Police Department (Metro), and the Henderson Police Department. Included are responses for each of the comments. Most of the comments have been addressed or are about to be addressed.

2.1 COMMENTS FROM METRO - FEBRUARY 16, 2016

Comments	Developer Observations	Status
It would be great to have the ability to draw the scene. Leonard Marshall from Metro explained how he prefers to use E/O measurements from a reference point, Area of Initial Contact and skid marks.	This capability is now implemented. Multiple colors, pen sizes, and pen types are available. Precise measurements are not possible on a drawing.	Implemented
The current scene diagram which uses pre-coded icons could be useful for PDO or minor crashes.	Both scene diagrams are available. Both can be completed for the same report.	Implemented
The screen of the Getac tablet is too big. The application is supposed also to be used by bikes who require smaller devices. The Panasonic Toughpad FZ-M1 tablet has the right size, anything smaller would be unusable.	We already have a Panasonic Toughpad FZ-M1 tablet. The application is compatible with any screen size.	Implemented
Data access and retention by Brazos is an issue. It is difficult to get access to statistics.	Our application gives access to all the raw data, along with auto-generated statistics. Additional statistics and charts are being implemented.	Pie charts are implemented
The interface is too small for big fingers.	The overall size of the application has been increased.	Implemented
Have the capability to file citations directly from the drivers listed in the crash, with pre-filled values from what was already collected.	Implemented	Implemented

Comments	Developer Observations	Status
The entire map of Nevada should be preloaded onto the tablet, so that a loss of connectivity will not affect the application.	The application now stores street data, but not satellite images.	Implemented
For the scene diagram, the map should zoom more.	The maximum possible zoom currently is being used, and cannot be increased.	Currently using the highest available resolution
AutoCAD is used for fatal crashes as high precision is required. AutoCAD cannot be integrated with the application.	A file can be attached to a report by means of the Web Portal.	Implemented
The application should be able to take as many photos as possible, in the best resolution available.	Partially implemented.	Implemented
Completion percentage is a tricky problem, as some fields may be required for injury/fatal but not for PDO, and for example the number of lanes is unrealistic if Off-Road is selected.	A capability to compute and report the percentage of fields that have been filled now is implemented. However, NHP has recommended to remove this capability in order to avoid confusion.	Implemented

2.2 COMMENTS FROM HENDERSON POLICE – APRIL 15, 2016

Comments	Developer Observations	Status
Have the two options for the scene diagram	Implemented	Implemented
The tablet is hard to read in direct sunlight. The UNLV team showed the feature to change from black/white background but that was not enough.	We cannot have more contrast than Black over White (or the opposite). The solution is to look at tablet specifications, and use the one that provides the best display.	Implemented two color backgrounds which can be switched back and forward with two clicks
Convert coordinates to address	This is possible (and has been implemented) for arterials when network communication is available. This is not possible for freeways and ramps. For a future phase of the project, we have a plan to provide a dynamic workaround of the limitations for freeways and ramps.	Implemented.
Reports should be editable directly via the Web Portal		Implemented

2.3 COMMENTS FROM NHP - JUNE 3, 2016

Comments	Developer Observations	Status
Very important: Location (capture the coordinates of the crash but also create a screen shot of the location of the crash; this screen shot is to be attached to the report)	The pdf report provides tree screenshots for tree zoom levels for the location of the crash.	Implemented
Move the scene diagram to the end of the process		Implemented
Have the capability to access a report by multiple officers at the same time while the report is being completed	One-way synchronization now is implemented. In a future phase, complete synchronization will be implemented.	Partially implemented
Capability to save data while the report is being completed (could be every minute)		Implemented
Have Drop Boxes for the number of lanes (check where else this applies)	Other fields may need to be modified as well.	Implemented
Pre-load insurance information from most common insurance companies (a menu to select the insurance company)		Implemented
Pre-load information for most common towing companies (a menu to select the company)		Implemented
Auto populate fields with the VIN number		Implemented
Investigate how to select multiple fields (even with the touch screen)	Controls of boxes were rewritten because Windows does not provide a way to do so without a keyboard.	Implemented
Popup only numerical keyboard when the entries are supposed to be numbers	Changing the keyboard type can be challenging if no Windows library exists for that task.	Implemented
Similarly, popup only alphabetical keyboard when the entries are supposed to be letters	See the above comment.	Implemented
Enter the codes for sequence of events (have the most common first – we can get the most common from NCATS)		Implemented

Comments	Developer Observations	Status
Have no limitation on the number of people or vehicles	Already implemented.	Already implemented.
Make the scrollbar more visible (larger or more contrast or both)	It is difficult to change the behavior and style of Windows elements.	It is not possible. The UWP framework does not allow it.
Use a very large field for the signature		Implemented
Add a capability to send a report by email from the website	Need to generate a pdf file.	Implemented
Make sure that the rotation of the table does not affect anything including the picture	Already addressed.	Implemented
Query tools by officer id, report, id, police department, etc.		Implemented

2.4 COMMENTS FROM NHP NOVEMBER 15, 2016

Comments	Developer Observations	Status
Include new statistics: - How many crashes involve injury - How many people got injured - How many with commercial vehicles - How many citations for cellphone		In Progress
Change everything to Military Time		Implemented
Remove confirmation for address		Implemented
Add feature that allows distance to reference point (feet)	Research is required to determine if the framework for UWP maps framework allows measuring distances.	Pending
Generate public pdf		Pending
In general, enable the textbox only when selected "Other" in the options		Implemented
Split first and last names		Implemented
Print capability for citations	Research is required to learn how to integrate with printers.	Pending
Resize vehicle to 50% less in scene diagram		Pending
Improve location in highways with GIS data	Research required to define and label the different GIS zones.	Pending
Improve behavior of tab button		Implemented

2.5 COMMENTS FROM NHP DECEMBER 6, 2016

Comments	Developer Observations	Status
Maximize size of map for scene location		Implemented
Include diagram drawing in the pdf report		Implemented
The accident number should not be auto generated		Implemented
Lock the small tablet to landscape		Implemented
Change default map to Satellite in Scene Diagram		Implemented
Update Form 5 to new format		In Progress
Change the fields Highway Environment Factors, Pavement Markings, Pavement Markings Type to multi selection		Implemented
Change Travel Lane and Turn Lane to dropdown number		Implemented
Add preloaded lists for endorsements codes, restriction codes and EMS companies.		Implemented
Research about information that Google maps provides for Highways	Google Map services provides only the name of the highway, which is useful but what is desired.	Done