UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

POLICE ELECTRONIC CITATION MOBILE SYSTEM FOR STATEWIDE DEPLOYMENT IN OKLAHOMA

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

HESHAM MAKHLOUF
Norman, Oklahoma
2018

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/299792

POLICE ELECTRONIC CITATION MOBILE SYSTEM FOR STATEWIDE DEPLOYMENT IN OKLAHOMA

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

_____
Dr. Joseph P. Havlicek, Chair

_____
Dr. Mohammed Atiquzzaman

_____
Dr. Ronald D. Barnes

© Copyright by HESHAM MAKHLOUF 2018
All Rights Reserved.

To my mom and dad, who in my mind, are always standing tall.

To my wife, who in my heart, is always sitting comfortably.

Acknowledgements

I would like to express my gratitude to my advisor Professor Joseph Havlicek, a.k.a. Joebob, for his continual guidance and support throughout this effort. I feel that large portion of my education at OU is a direct result of working with Professor Joebob. I also like to thank Professor Mohammed Atiquzzaman and Professor Ronald Barnes for serving as my committee members.

I am grateful for the assistance that Rodrigo, Captain Eric Cannaday, Dr. Patrick Campbell, and other CITS members have given me during the time I worked toward my degree in Norman. I also greatly appreciate the support I got from my supervisor at work Jay Kangsudarmanto throughout the years we worked together.

Lastly my deep gratitude goes to my wife and my parents for their consistent support, encouragement, and love.

Table of Contents

Acknowledgements ..... **iv**
List of Tables ..... **vii**
List of Figures ..... **viii**
Abstract ..... **x**

Chapter 1 Introduction and Thesis Organization **1**
1.1 Thesis Organization 3

Chapter 2 Background and Problem Formulation **5**
2.1 Police and DDACTS 6
2.2 DDACTS Guidelines 8
2.3 Traffic Crashes and Fatalities Statistics 10
2.4 Police and Technology 12
2.5 Traffic Citations: Then and Now 14
2.6 Handwritten Citation Process 15
2.7 Electronic Citation Process 17
2.8 Case Studies 18
2.9 Electronic Citation System Cost Analysis 22

Chapter 3 Smartphone Technology for Electronic Citations in Oklahoma **26**
3.1 Smartphone Peripheral Devices for Electronic Citation System 29
3.2 PARIS+ app Use Case 32
3.3 PARIS+ app Design Considerations 35
3.4 PARIS+ app Software Architecture 40

Chapter 4 PARIS+ app Implementation **47**
4.1 Using Xamarin Forms to Build the PARIS+ app 48
4.2 PARIS+ app User Interface Flow 52
4.3 Contact Page Holder Class 55

4.4 Environment Page 58
4.5 Location Page 59
4.6 Person Page 59
4.7 Vehicle Page 61
4.8 Violation Pages 63
4.9 Print Page 67
4.10 Location Lookup Class 69
4.11 The Great Circle Theorem 72
4.12 Driver License Parser Class 76
4.13 VIN Decoder Class 79

Chapter 5 Testing the PARIS+ app 83
5.1 Testing the Driver License Parser Class 86
5.2 Testing the VIN Decoder Class 86
5.3 Testing the Location Lookup Class 87
5.4 Installing the app 88

Chapter 6 Conclusion and Future Work 89

References 93

List of Tables

Table Page

Table 2.1 National motor vehicle fatality causes distribution and percentage of change between 2015-2016 [61]. 11

Table 2.2 The amount of savings and additional revenue using digiTICKET from FY 2009 to FY 2014 in Sand Springs Police Department [24]. 22

Table 2.3 Market prices for hardware needed to deploy electronic citation system in one Oklahoma Highway Patrol vehicle. 24

Table 3.1 Summary of technical specifications of Samsung J7 smartphone [77]. 39

Table 3.2 Technical specification of QS5801/8001 printer [72]. 39

Table 4.1 Element identifiers in the Driver License barcode [8]. 78

Table 5.1 Android OS versions tested with PARIS+ app. 85

Table 6.1 Comparison between different types of citations methods for law enforcement 91

List of Figures

Figure Page

Figure 2.1 Example of a hand-written ticket [13]. 15

Figure 3.1 The Components of a typical Electronic Citation System [44]. 28

Figure 3.2 PARIS+ app Use Case Diagram. 36

Figure 3.3 PARIS+ app Layered Software Architecture. 43

Figure 4.1 PARIS+ UI Flow 53

Figure 4.2 MVVM Architecture. 54

Figure 4.3 Main Page while Create Citation button is loading. 56

Figure 4.4 Map Page with closest highway and closest intersection locations populated. 56

Figure 4.5 Get Location Command binding between Contact Page Holder View and ViewModel. 57

Figure 4.6 Environment Page with extra fields displayed depending on Bond Type field selection 58

Figure 4.7 Location Page with populated fields 58

Figure 4.8 Scanning driver license barcode. 60

Figure 4.9 Person Page with populated fields 60

Figure 4.10 Using MVVM in Vehicle Page. 62

Figure 4.11 Violations Search Page with search text “sb3”. 64

Figure 4.12 Violations List Page with added violations including a speeding violation. 64

Figure 4.13 Violation Page containing details about selected violation. 64

Figure 4.14 Searching, adding, and removing violations. 65

Figure 4.15 Print and Print Preview Flow. 68

Figure 4.16 LocationLookup Class. 70

Figure 4.17 Great circle distance 72

Figure 4.18 Angles and distances between 2 points X and Y on a sphere surface 73

Figure 4.19 getCounty Method. 75

Figure 4.21 VIN number 1-D barcode. 80

Figure 4.22 VIN number pattern 80

Figure 4.23 VINDecoder Class. 81

Figure 5.1 Android OS usage distribution [31] and the PARIS+ app testing 85

Abstract

POLICE ELECTRONIC CITATION MOBILE SYSTEM FOR STATEWIDE DEPLOYMENT IN OKLAHOMA

Hesham Makhlouf
The University of Oklahoma, 2018

Supervisor: Dr. Joseph P. Havlicek

Collecting timely and accurate data for traffic citations and accident reports are important goals for law enforcement. Manually filled paper forms have historically contained content errors, took longer time to complete, and involved longer delay points to reflect in the records management system. Electronic citation systems were first introduced in the year 2000 where police officers used computers to fill digital citation forms, print out traffic tickets on-site, and send digital copies of the records to courts via wireless networks. To date, electronic citations systems have seen limited deployments in the State of Oklahoma. Law enforcement officials have estimated that approximately 80% of the 481 police agencies in the state are still using the paper and pen method to issue traffic citations due to financial reasons. In this thesis, I collaborated with another developer at the University of Oklahoma Center for Intelligent Transportation Systems (CITS) lab to develop a smartphone-based electronic citation system to help Oklahoma police agencies that still use the paper and pen method to issue electronic traffic citations. The proposed system is significantly cheaper than other computer-based electronic

citation systems that are currently present. The developed software can be installed on most of the commercially available Android smartphones to collect accurate and reliable traffic citation data, print legible tickets on-site, and send reports to headquarters. Ideally, it can take less than a minute to complete a traffic citation using the proposed system.

Chapter 1

Introduction and Thesis Organization

The use of technology in law enforcement operations is a benchmark in the 21st century towards effective and efficient law enforcement [63]. Using computers and modern software to digitally collect and process data allows government agencies to generate proper temporal and spatial statistics, identify inefficiencies, put action plans, and assess the results [63].

Law enforcement officials prioritize the available resource and provide myriad services on daily basis [63]. As the scope of those services expand every day, the quality of the collected data becomes more significant. At the same time, the available resources are not enough which increase the demand to operate with higher degree of efficiency [48,52,63]. U.S. government agencies provide models that help the police agencies understand the importance of data collection as a cornerstone for efficient policing operations. The collected data represents the official understanding of all the circumstances behind any incident and it's quality significantly affects all the processes that comes after such as data analysis, causation factors for future planning and strategic solutions to deter crime and enforce law [42].

Over the past decade, the University of Oklahoma Center for Intelligent Transportation Systems (CITS) lab collaborated with law enforcement agencies in Oklahoma to deploy electronic police forms to streamline and automate incident data transfer in the field. This collaboration has led to significant enhancement in

the data collection techniques which improved the quality, timeliness, and correctness of the police records in the state [44].

The available technology utilizes laptop computers that run advanced software to provide electronic traffic forms and communicate with external peripheral devices such as digital scanner, barcode readers, magnetic stripe readers and GPS, etc. [85]. These peripheral devices make data collection more accurate, easier, and faster. On the other hand, acquiring these systems require extra resources that are sometimes not available [48,52].

With the continuous increase of the scope of the policing services and the limited resources of the smaller agencies in the State of Oklahoma, the available programs are monitored, evaluated, and adjusted on an ongoing basis to make sure that law enforcement agencies can get the most benefit of the available technology. Unfortunately, around 80% of the police agencies in Oklahoma are still using old-fashioned paper and pencil method to issue traffic citations [25].

Nowadays, technology is pushing limits on handheld mobile devices [39]. The increasing use of mobile devices in our lives resulted in a critical need to have well-functioning smartphone [74]. Technology is not just making this available, it also enabled brilliant innovations to improve services, user experience, and eliminate inefficiencies [74]. As technology raises the bar for users, it can also help control complexity and can easily bring more personalized services as needed.

In 2013, the International Telecommunications Union (ITU) expected that there will be more mobile phones than people on earth [80]. As global mobile-cellular penetration approaches 100 percent, the use of mobile phones has

become more friendly and easier than using computers specially with versatile smartphones' apps [75].

For the police, Laptops and external peripheral devices attached to them cost thousands of dollars per unit [24]. In addition, maintaining a running fleet needs extra time away from the focus of the core duty of law enforcement [85,24]. Smartphones running suitable software apps can be used instead of computers and peripheral devices attached to them to achieve the same result at a lower cost and easily maintainable system. Nevertheless, using smartphones add portability and provide extra features like built-in cameras and artificial intelligence capabilities. These features pave the road for enhanced data collection techniques and improved user experience.

In the presented work, I collaborated with another CITS lab member to develop the PARIS+ mobile app that can record electronic traffic citations on Android smartphones and print tickets. The estimated cost of PARIS+ app is below $300 as per the market prices on January 26th, 2018. This price is significantly lower than the prices of the typical electronic citation systems that are currently available for the police agencies in Oklahoma. This price includes buying both the smartphone and the printer i.e. the whole system. For the agencies that already issued smartphones to their officers, the cost of the system will be only the price of the Bluetooth printer which brings the price of the whole system to below $100.

1.1 Thesis Organization

In the next Chapter, I will explain the importance of accurate and fast data collection methods for law enforcement agencies and how it serves established

government models to increase public safety. Then I will discuss the benefits of using the electronic citation systems over the paper and pen method to issue traffic citations and the reasons preventing the law enforcement agencies from acquiring such systems.

In Chapter 3, I will discuss how the proposed system can be used to collect citation data in the field. I will then present the steps I went through to build the software solution including the benefits and limitations of using a smartphone. I will also present how these limitations were handled while developing the software architecture and implementation.

In Chapter 4, I will discuss the contribution that I did towards developing PARIS+ app and any optimization that was done on the road to develop a working solution.

In Chapter 5, I will discuss the testing that was done for the application.

In Chapter 6, I will present the conclusions and future work.

Chapter 2

Background and Problem Formulation

It wouldn't seem too strange to think that the police officer was a figure that existed since the begging of civilization. In fact, the police force in the United States is not a very old invention, sparked by changing notions of public order, driven sometimes by politics and the economy state [67].

The first modern metropolitan police agency was established in London in 1829, five hundred and forty years after the statute of Winchester in 1285 [22] when King Edward I of England codified how the policing system should work. The early law enforcement history in the United States of America was modeled after the first police agency in London, where the first American police agency was established in Boston and New York in the 1840's [49].

Sir Robert Peel was the English prime minster when he created the London police in 1829. The Peelian principles are referred to as the rules by which the police will act. These principles used preventive policing model in its earliest form [5]. The Peelian principles suggested that the lower the crime rates, the more effective are the police, information about the criminal activity should be shared with the public, police will be adequately spread geographically and temporally, and that good record keeping is important for deploying the police and resources where they are most needed [5].

Before the 1920's the American police were nationally viewed as professional crime fighters. After that, due to reformers such as August Vollmer, Orlando W. Wilson and William H. Parker the police professionalism model started to

appear [45]. These early reformers realized that the average citizen didn't know a lot about the police, but did know about crime, especially when they were the victims [4]. These reformers decided that the police image needed to be regenerated as a professional police model. This could be achieved with the aid of college educated officers and vehicles capable of chasing criminals using any resources that are needed to achieve this target [49].

The evolution of the police models continued and the problem-oriented policing model became more popular [21]. Problem oriented policing identifies police problems and develops intelligent solutions to solve and evaluate any problems through data analysis and crime mapping to see where those crimes occur and to identify trends and patterns [73]. It sometimes goes deeper to examine the underlying social conditions causing the crimes [41].

2.1 Police and DDACTS

Crime and traffic violation incidents represent a threat to public safety [76]. Based on the statistics obtained from the National Highway Traffic Safety Administration (NHTSA) and the Federal Bureau of Investigation (FBI), there were approximately 33,561 traffic crash fatalities, 2.36 million crash-related injuries [65], and 1.21 million violent crimes reported in the United States in 2012 [26]. Motor-vehicle accidents and crime have serious impact on human social life.

Law enforcement agencies work day and night to monitor and improve traffic safety and reduce crime rates [63]. Law enforcement officers try to keep DUI (Driving Under Influence) drivers off roads by using effective approaches and devices to identify such drivers. Government provide law enforcement agencies with traffic

safety programs to support effective and comprehensive enforcement operation [38].

The Data-Driven Approaches to Crime and Traffic Safety (DDACTS) paradigm was introduced in July 2008 [42]. DDACTS was a result of cooperation between NHTSA, the Bureau of Justice Assistance (BJA), the National Institute of Justice (NIJ) and many local enforcement leaders in the United States. DDACTS uses the ideas of Problem Oriented Policing to help law enforcement agencies operate efficiently. It places renewed focus on traffic law enforcement as an important factor to reduce crime, crashes, and traffic violations in the communities across the country [42].

Law enforcement officials prioritize the available resources and provide myriad services on daily basis [63]. As the scope of those services expand every day, the quality of the collected data becomes more significant. At the same time, the available resources always lag the need which increases the demand to operate with higher degree of efficiency [63]. DDACTS demonstrated that the use of timely and accurate localized data to direct law enforcement operations is a benchmark in the 21st century towards effective and efficient law enforcement [42].

DDACTS provides active means for minimizing crime and crashes in the community. It utilizes a problem-based policing model as opposed to the traditional person-based policing model. It also addresses how to better plan allocation of police personnel and resources as a means of achieving “high visibility policing” to target both crime and traffic violations [42].

The high visibility policing model demands that the police are highly visible around “hot spots” i.e. locations that have higher probability of crime occurrence and

traffic violations. This is an effective and proven way to fight higher crime rates and traffic violations in the community [63]. The use of location mapping tools provides a great value towards achieving the DDACTS targets which demonstrates the importance of the role of the information technology in the modern policing models [42, 63].

2.2 DDACTS Guidelines

According to the NHTSA DDACTS Operational Guidelines [63], the main target of DDACTS is to reduce crime and crash rates through directed patrol. DDACTS relies on seven main guidelines to enhance the law enforcement services in the community which are: Community Partnership, Data Collection, Data Analysis, Strategic Operations, Information Sharing and Outreach, Program monitoring and measuring outcomes [63]. The following section discuss these guidelines.

1. *Participation of Partners and Stakeholders*

Collaboration with other law enforcement agencies is crucial as it provides support and opportunities to decrease crime and improve the quality of life [63].

2. *Data Collection*

The effective and timely acquisition of accurate crash, crime, and enforcement related data including incident details such as type, time and location of the incident are cornerstones for DDACTS [63]. Other data such as arrests, citations, warnings, motor vehicle stops, citizen complaints, and field interviews are very important pieces of information. All this information needs to be collected to be used for police analysis and planning [63].

3. *Strategic Operations*

The collected data is used for performing analysis, from which the police

agencies should be able to identify the hot spots that are likely to have the higher rates of incidents, crashes, and crime [63]. The hot spots should be targeted with strategic and highly visible enforcement activity at the needed area and at the most needed time to help reduce the overall social harm [63].

4. Information Sharing and Outreach

Opportunities should be given to share the results and actionable information inside and outside the individual police agency which helps to promote community partnership and document the accomplishments [63]. Periodic progress reports are critical to keep the leadership aware of the progress and to keep officers and community members aware of the advancement. Progress reports also inform the government administrators and elected officials [63].

5. Monitoring, Evaluation, and Adjustments

Supervisors should review and analyze the strategic operations and enforcement activities for enhancement [63]. The collected data and the analysis should allow for dynamic adjustment strategies in order to continuously reduce crash and crime rates, assess cost saving strategies, and any other criteria that define success [63].

6. Outcomes

The planned target for every hot spot and its related strategic plans are transformed into outcome measures [63]. These measures are used to assess the outcome of the applied plans in reducing the crime and traffic violation rates and the effectiveness of the cost saving methods. The DDACTS model supports more long-term outcomes over immediate results in determining the effectiveness of law enforcement operations [63].

This concludes the principle guidelines for the DDACTS model.

2.3 Traffic Crashes and Fatalities Statistics

The latest available data from the Centers for Disease Control and Prevention (CDC) in 2014 shows that unintentional motor vehicle traffic-related injuries accounted for the second cause of injury deaths in the United States [46]. Collected data shows that Seat belts are the most effective method to reduce or prevent injury in the event of a vehicle crash [16].

Wearing a seat belt is a primary enforcement law in Oklahoma [16] (primary enforcement law means that you can get a police stop only for not wearing a seat belt). In 2008, a sample of 7,743 drivers in the state showed that only 82.3% of the drivers wore a seat belt which leaves around 17.7% of the drivers without seat belts on the roads of Oklahoma [16].

The CDC used data from the National Electronic Injury Surveillance System–All Injury Program (NEISS-AIP) for the calendar year of 2009 for the crash injuries in the United States and estimated that 2.3 million adults had non-fatal motor vehicle injuries that was treated in the emergency section [16]. The nonfatal motor vehicle–occupant injury rate was 1,007.5 per 100,000 population in the same year [16].

As per the NHTSA 2016 Fatal Crash Data report, which was collected from all 50 states and the District of Columbia, 37,461 lives were lost on U.S. roads in 2016 [62]. This was a 5.6% increase from the previous year numbers. NHTSA concluded that deaths related to bad driving habits including DUI, speeding, and not wearing seat belts continued to increase [61]. Table 2.1 shows the national fatality distribution and percentage increase/decrease between 2015-2016 [61].

Table 2.1 National motor vehicle fatality causes distribution and percentage of change between 2015-2016 [61].

Category	Number of fatalities	Percentage of increase/decrease compared to calendar year 2015
Drunk-driving deaths	10,497	+1.7%
Unrestrained deaths	10,428	+4.6%
Speeding-related deaths	10,111	+4.0%
Pedestrian deaths	5,987	+9.0%
Motorcyclist deaths	5,286	+5.1%
Distraction-related deaths	3,450	-2.2%
Bicyclist deaths	840	+1.3%
Drowsy-driving deaths	803	-3.5%

Also, NHTSA's 2016 Fatal Motor Vehicle crashes report shows that Oklahoma had 683 traffic related fatalities in 2016 due to alcohol impaired driving, which also accounted for the biggest cause of traffic related deaths on U.S. roads [61].

In the same year, the National Law Enforcement Officer Memorial Fund reported that 53 officers were killed in traffic related incidents in the United States. Out of these 53 traffic-related deaths, 15 officers were struck and killed while they were outside of their vehicles [81].

NHTSA suggests that a starting point for a long-term change is the technology that will continue to improve the practice and operation of law enforcement [63]. Current and future technologies, such as smaller and faster computers, improved scanners or cameras, and the further application of information technology will greatly enhance the effectiveness of law enforcement operation and data collection, as suggested by the DDACTS model [63].

The presented traffic fatalities' statistics depict the importance of the police role to protect the public and decrease transportation fatalities to a minimum. This target

requires aiding the policing agencies with the proper technology needed to help them performs their duties easily and collect field data efficiently and quickly.

Timely and accurate data collection helps law enforcement agencies to identify the hot spots, put the proper action plans in place and assess the results [63].

2.4 Police and Technology

Recent advancements in technology changed the way people live and opened new paths to serve human welfare by supporting services that were previously thought to be difficult to accomplish [74]. Government agencies also use technology to contribute to peoples' way of life and open more prospects [74]. This change not only opened more opportunities for policing organizations but also for crime [53]. Crime fighters try to use technology to serve the community, achieve efficiency, and limit the crime rate.

As both criminals and police benefit from technology, the technology creates new disadvantages for both sides [53]. The Use of technology by criminals represents more risks and challenges to law enforcement and vice versa. Proactive regulations do not always provide solutions to make sure that the crime fighters get an edge over criminals and the borderless nature of technology may not allow for firm regulations. It may instead contest the principle of criminal laws to add new definitions for crimes committed using new technologies [53].

The massive diffusion of data on the internet and new media technologies has led to more efficient police work [53]. The rapid data transfer and easy acquisition techniques are very important elements. Achieving more goals in less time for law enforcement leads to less crime and hence successful policing [63]. Data transfer and

sharing among law enforcement agencies and intelligence organizations facilitates efficient and proactive law enforcement [63].

Sharing offender databases between police agencies nationwide and sometimes even internationally saves time and resources, especially in the case of repeat offences or trans-border crimes [53]. For example, police can be only a few mouse clicks away from accessing DNA databases to analyze whether a suspect can be linked to a particular crime scene. Traffic violation records can be used as reliable methods to track and find suspects [25]. The internet contains a countless number of secure databases of information that cover a lot of aspects of life and can provide leading information about criminals and offenders [53]. Acquiring such information is not only inexpensive, but it also doesn't require unreasonably high-level or specialized skill because the information is in many cases provided by user friendly software [53].

In 1994, Compstat was first used by William Bratton, the police chief of the New York Police Department (NYPD) where New York City was experiencing high crime rate and disorder [28]. Also, the NYPD was suffering from lackadaisical attitudes among many officers and leaders and a lack of accuracy in data about public safety issues and crime [45].

Compstat is a software that was introduced to collect and harvest police data and resources available and track statistics to allocate resources in the most efficient way [45]. It was recognized as a major innovation in American policing [88]. Compstat grew over the following fifteen years and came to be handy with other electronic mapping software to show where crime occurred geographically [45]. Up until today it is still being used by some police agencies in the US in varying incarnations from the

first Compstat [20].

The use of Compstat was an early demonstration of how empowered can law enforcement perform with the aid of technology [53]. Compstat demonstrated how computer software can be used to analyze and run police departments. The evolution of using technology to aid police did not stop at this point, in fact it was just the start.

In the next section, I will be focusing on the use of the technology to aid the police in issuing traffic citations and the impact of streamlining this process for police agencies in their mission to protect the public.

2.5 Traffic Citations: Then and Now

The world's first automobile accident happened in 1891 in Ohio City, Ohio, when James Lambert hit a tree [68]. He lost control over the vehicle and crashed into a hitching post. Five years later, in 1896, the first speeding fine was issued to Walter Arnold for speeding 8 mph in a 2 mph zone [71]. In 1904, the first handwritten traffic citation was issued to Harry Myers in Dayton Ohio for racing down the street at 12 mph. He was fined for $10 [69]. A traffic citation (or ticket) is a notice issued by a law enforcement official to a driver or any road user to show that the user has violated the traffic law. The traditional form of issuing a traffic citation used a pen and paper to write a ticket. Recent advancements in technology allowed the use of computers to easily issue traffic citations on electronic forms rather than using a paper and a pen [83] which is referred to as electronic citation. The first electronic citation system was used in Cumberland county, North Carolina in 2000. Electronic citations were issued on a system that let officers print out a legible paper ticket on-site while simultaneously sending a copy to the court via wireless technology [83].



Figure 2.1 Example of a hand-written ticket [13].

2.6 Handwritten Citation Process

Hand written citations were the original form of citations issued to violators. They involved using a paper and a pen to issue traffic citations. An example of a handwritten ticket is shown in Figure 2.1 [13]. Nowadays, technologies enabled faster ways to issue citations that have overcome lots of inefficiencies in the handwritten citation process as follows:

1. *Officer Safety and time*

First comes the safety of the officer and public. Issuing a handwritten citation is not the fastest way to process a ticket. The officer needs to fill in information about the location and time of the violation, information about the person involved, the type of violation, the court date and time, and sometimes even more. Every minute a law enforcement officer spends to complete this process means longer time on the side of the road for both parties and hence increased chance of being hit by another motorist.

In addition, maximized contact with civilians make officers more vulnerable to verbal and physical exploitation by violators who are typically frustrated.

As per the preliminary 2017 law enforcement officer fatalities report [82], traffic-related fatalities were the leading cause of law enforcement deaths in the United States in 2017 when 47 officers were killed in traffic related incidents. Out of the 47 fatalities, 9 officers were struck and killed while outside of their vehicles [82]. Decreasing the time an officer has to spend on every traffic stop will directly impact the probability of having another crash and will better protect the public and law enforcement personnel while on duty.

2. Delays and Errors

Officers turn in copies of issued handwritten citations at the end of the shift. A supervisor must approve citations before they go to the records bureau where a clerk enters the data into the Records Management System (RMS) [3,51].

This process involves multiple delays and error-prone points before citations reach the court. It also adds more work for the clerk trying to decipher correct information from reports due to smudges or poor handwriting. For example, the State of Iowa reports that it used to take up to 18 months for a handwritten crash report to complete a full cycle, starting with the reporting officer and ending with U.S. Department of Transportation (DOT) analyzers [83].

3. Loss of Revenue

As more delays and errors are involved due to the problems and issues explained above, tickets might be dismissed due to trivial mistakes which means loss of revenue to municipalities and courts [24].

2.7 Electronic Citation Process

The electronic citation process means using a software on a computer to enter and/or import the information needed to issue a traffic citation [83]. Electronic citation systems utilize mobile computers installed in patrol vehicles that can be equipped with peripheral devices to help officers populate driver license information, pull driver history, scan vehicle information, search and populate violations, easily print citations to violators and finally transmit all information wirelessly to police headquarters or to the courts [83].

Using the electronic citation systems can reduce citation errors as the software can check for valid entries on the spot [44,80]. This helps to make sure that citations are properly filled and hence decreases or sometimes eliminates any dismissals from courts which means capturing revenue that was lost before.

Also, some software allows officers to check the driver's record history and hence make better decisions to identify reckless or DUI drivers and repeat offenders [17]. At the agency end, the clerk no longer needs to do manual entries after citations are approved as the system automatically transfers the data to the RMS and court computers which means alleviating some of the manual duties which saves clerk time and reduces errors [83]. The implementation of the electronic citation system shows an improved impact on officers and clerks productivity, allows for cost saving on paper, person hours and minimize errors on citations [83].

According to the Melissa Mann article "How eCitation systems improve officer safety, reduce costs" on www.policeone.com, it generally takes an officer 10-15 minutes to issue a moving violation, multi-part ticket. The average time it takes to issue

an eCitation is less, requiring only few minutes, which reduces the traffic stop time three to five times, and dramatically increase officer and public safety [51].

Using the electronic citation systems benefit police agencies that don't have enough resources to meet the targets by alleviating some of the manual duties. It can also benefit the municipal court divisions by receiving electronic forms rather than scanned reports [86].

2.8 Case Studies

As electronic citation systems have been widely used in many states since the first use in 2000, I dedicated this section to show a few real examples for some police agencies from different states that used the electronic citation systems to help law enforcement to improve police records timeliness and correctness.

1. *The State of Iowa*

In a Joint report, "The use of Electronic Citations: A Nationwide Assessment" by the Bureau of Justice Assistance and the U.S. DOT in 2003 [83], it was reported that the State of Iowa used to take up to 18 months for a crash report to complete a full cycle, starting with the reporting officer and ending with U.S. DOT analyzers. With the development and implementation of a system called Traffic and Criminal Software, or *TraCS*, which transmits data directly to all interested parties, this time has been reduced to 18 hours, improving the timeliness and correctness of the traffic records [83].

TraCS is a software that supports the deployment and management of the electronic police forms. TraCS began as a federally funded project in the state of Iowa and has evolved into a comprehensive software architecture used by 18 U.S. states and

2 Canadian provinces that were part of North American TraCS consortium [85].

2. The State of Maryland

Maryland State police used to take over 8 minutes to issue a handwritten citation till they developed the electronic traffic information exchange (E-TIX) software to issue traffic citations and warnings electronically [17]. E-TIX was certified by the district court in March 2008 as an electronic citation system in the state. With E-TIX, officers can now scan the bar code on the driver's license and vehicle registration to populate the information they need in a traffic citation [17]. The populated information is automatically checked against the E-TIX database for history of stops and is also automatically checked against the National Crime Information Center (NCIC) database [17]. Once ready, the ticket is sent to a thermal printer that is installed in the patrol vehicle.

Maryland State Police claim that using the electronic citation system reduced the average time issue a citation from 8 to 3 minutes which improves the officer safety significantly [17]. E-TIX is a software that was developed by the Maryland State Police from scratch to be used by different police agencies across the State of Maryland [47]. Although the Maryland State Police are collaborating with other police agencies in the state to provide the software free of charge, the cost of the hardware still sits at $1,000 per patrol unit, which is covered through grants from several agencies [47].

As per the latest data on the E-TIX portal, "Delta Plus E-TIX Training" website, out of the 142 police agencies in Maryland [12], 132 police agencies in the state are currently partnering with the Maryland State Police to train their officers to use E-TIX software. This means that approximately 93% of the police agencies in Maryland

have already deployed the electronic citations system, E-TIX.

3. The State of Oklahoma

Oklahoma DPS has collaborated with the Center for Intelligent Transportation Systems (CITS) lab at the University of Oklahoma to implement electronic police forms, mobile data collection and automated data validation to meet federal expectations through improving the timeliness and correctness of the traffic records. They adopted multiple pilot projects that started by implementing a TraCS pilot project in 2007 [85].

Traffic and criminal software or TraCS was designed by the State of Iowa to streamline and transfer incidents from the field before it was made available to other states [85]. Although TraCS is available at low cost to all the police agencies in the country, police agencies still need to spend money to buy computers, printers and other peripheral devices that make using the electronic forms come at the full value to the police agencies.

TraCS was developed for a consortium of government agencies led by the State of Iowa. It was nationally available to these states as a fixed software engine. Iowa DOT served as the main point of contact for any state or province technical issues related to TraCS [85]. Oklahoma agencies that used TraCS experienced some issues with the communication link to headquarters and the inability to perform agile modifications through the committee because it was costly and difficult to obtain [85]. Local modifications to TraCS code weren't possible as a private company designated by Iowa Department of Transportation owned the sole rights to implement any changes as per TraCS user agreement [85]. These difficulties didn't make TraCS a comprehensive

solution for Oklahoma police agencies.

Later, CITS lab led by Joseph Havlicek and Ronald Barnes developed the Police Automated Record Information System (PARIS) software in collaboration with Oklahoma DPS to help Oklahoma police agencies to electronically capture traffic records [44]. PARIS was used by multiple police agencies including the Oklahoma Highway Patrol, City of Tulsa Police Department, City of Oklahoma Police Department and Oklahoma County Sheriff's Office which led to big improvements in acquiring and maintaining traffic records [44]. Using PARIS, it takes less than 4 minutes to complete a traffic citation [25].

Some of the police departments in Oklahoma such as Sand Springs Police Department had the financial ability to acquire commercial electronic citation systems from private companies such as "digiTICKET®" [24].

digiTICKET is an automated citation delivery system that is owned by Satus Technologies™, LLC. digiTicket is a wireless mobile application that can run on laptop computers or windows handhelds/tablets to produce citations in an electronic format that is instantly transmitted to the court system [76].

Sand Springs Police Department acquired 23 digiTicket units and related software between March 2009 and January 2010 at a total cost of $109,600 with an additional annual fee of $4,815 for annual maintenance [24]. The average cost was $4,765 per unit plus maintenance fee. The Economic Impact Group, LLC reported that Sand Springs Police Department experienced a 63% increase in citation and court revenue, $8,500 annual reduction in administrative costs and a 67% reduction in traffic accidents, resulting in an estimated saving of $2 million in societal costs. Over the

Table 2.2 The amount of savings and additional revenue using digiTICKET from FY 2009 to FY 2014 in Sand Springs Police Department [24].

Year	Cost Saving from Reduction in Citation Entry	Net Court Revenue from Additional Citations
2009	$2,094.00	($463.48)
2010	$8,766.32	$47,262.11
2011	$9,174.83	$47,639.94
2012	$9,602.38	$48,020.79
2013	$10,049.85	$48,404.69
2014	$2,629.54	$48,791.65
Total	$42,316.93	$239,655.70

period of 5 years, the net value of benefits is estimated to be $470,00, excluding the societal savings [24]. Data from Sand Springs Police Department in Table 2.2 shows additional court revenue and savings in Sand Springs Police Department after deploying digiTICKET in 2009 [24].

2.9 Electronic Citation System Cost Analysis

An electronic citation system in a police vehicle typically consists of a computer to run the software and peripheral devices to facilitate data acquisition. Peripheral devices typically include a Global positioning system (GPS) to capture location coordinates, wireless or wired magnetic stripe reader or 2D bar code scanner to capture driver license information and vehicle identification numbers, communications hardware that provides internet to allow submitting traffic forms when completed and a printing machine to print legible paper tickets [85]. Although the idea of using electronic citation systems has been widely spread and accepted as a very efficient replacement for the handwritten citation system [17,24,83], it doesn't come at zero cost [24,85].

Even if government-offered software solutions were developed to fit state

requirements and were provided at little or no cost, police agencies would still need to spend money to buy computers and peripheral devices that are needed to allow using the software properly. Also, a team of professionals are needed to install, support, maintain, and help operate the system [85].

Table 2.3 list the hardware peripheral devices that were used in Oklahoma Highway Patrol units in TraCS Pilot Project and market prices as of January 26th 2018. A simple calculation for the numbers in the Table 2.3 shows that it can cost anywhere from $2,000 to $5,000 to equip one patrol vehicle with the needed hardware to run TraCS, or similar government owned software. Unfortunately, Law enforcement agencies across the united states are already suffering from budget cuts due to financial reasons and to be able to keep more officers employed rather than spending more money to equip police vehicles as needed [48,52].

With the great advantages that the electronic citation system brings to any police agency, the main factor that prevents using the electronic citations system as compared to the handwritten citation system is the cost to acquire, maintain, and operate [47]. Lieutenant Tony Perez of Miami-Dade County police reported in 2008 that the equipment and software costs about $4,500 per officer, while he hopes to install 3,500 devices in police cars. This requires a funding of $15.75 million that doesn't include the cost for training troops or maintaining software, technical support and help desk [47]. In June 2017, Mayor Tom Barrett of Milwaukee, Wisconsin reported that the city could be forced to cut 84 police officer positions in their 2018 budget. In an article in the Journal Sentinel, a part of USA Today network, he stated that [52]

"I do not want to have fewer police officers in this city."

Table 2.3 Market prices for hardware needed to deploy electronic citation system in one Oklahoma Highway Patrol vehicle.

Hardware	**Use of Hardware**	**Hardware Model***	**Market price**
Mobile Data Terminal	Runs Software and peripherals	Panasonic CF-Series**	Range: $1,289.49-$3,659.00
2-D Barcode Scanner	Scan ID to import driver information	Code Reader 2300	$375.10
Magnetic Stripe Reader	Scan ID to import driver information	MAGTEK Mini Magnetic Stipe Reader	$47.39
Electronic signature pad	For Signing tickets	Topaz Sigpad T-S460	$93.59
Communications Hardware	To transmit data to/from servers	Depends on Network	$50***
Printer	Print Tickets	Cannon iP90	$319.99
USB Hub	To connect peripheral devices to laptops	Depend on Hardware Req	$20***
GPS	Capture location coordinates	Maptech Chart Navigator Standard	$249.99

* The above hardware models were used in Oklahoma TraCS Pilot Project in 2008, prices current as of Jan 23rd, 2018 [85].

** OHP used Panasonic CF-29 which is obsolete now. Current Panasonic ruggedized laptops range from $1,289.49-$3659 based on specifications.

*** Estimated prices depending on chosen models and services.

In a May 2017 story on Oklahoma News 4, Oklahoma City Police Chief Bill Citty claimed that the Oklahoma City Police Department's budget was cut by $1.7 million, which will affect personnel. He mentioned that the police department will eliminate eight positions that are currently vacant to deal with the budget cut while operating with 34 vacancies in the department [48]. Citty Stated:

"We don't like to lose any officers."

There are 481 police agencies in Oklahoma [12]. It is estimated that around 80% of these police agencies still use handwritten paper forms to issue citations [25], where most of the agencies are smaller ones that employ fewer officers and can't afford buying and maintaining electronic citation systems.

For the reasons discussed above, the need to have an optimized low cost electronic citation system became very important to the State of Oklahoma in order to comply with the Federal requirement to improve traffic record timeliness and accuracy [43], which is currently hard to achieve with the high cost of the electronic citation system.

In the next Chapter, I will discuss a proposed solution that allows police agencies in Oklahoma to have lower cost electronic citation systems.

Chapter 3

Smartphone Technology for Electronic Citations in Oklahoma

For the reasons mentioned in Chapter 2, the benefit of having electronic citation system available for a state-wide deployment in Oklahoma became very clear. For mainly financial reasons, around 80% of the police agencies on Oklahoma still use paper and pencil to issue citations [25]. The rest of the agencies either use government supported software as PARIS or buy commercial software if they can financially afford it [24, 44]. The CITS lab at the University of Oklahoma along with the transportation and law enforcement leaders in the state were aware of that and are researching newer technologies that can be used to achieve the same results as the electronic citation system but for lower prices.

Nowadays, technology is pushing limits on handheld mobile devices [39]. The increasing use of mobile device in our lives resulted in a critical need to have well-functioning smart mobile phone [74]. Technology is not just making this available, it also enabled brilliant innovations to improve services, user experience and eliminate inefficiencies [74]. As technology raises the bar for users, it can also help control complexity and can easily bring more specific services as needed.

In 2013, the International Telecommunications Union (ITU) expected that there will be more mobile phones than people on earth [80]. As global mobile-cellular penetration approaches 100 percent using smartphones has become more friendly and easier than using computers specially with versatile mobile applications and the easy interaction using touch screens [75].

A smartphone is perhaps the best new tool an officer can have on his patrol shift as per [50]. It's a vital asset where it can be used as direct communication link when the police radio signal is week or unavailable, especially in rural areas. Cell Phone pictures and videos are of great benefit on any incident scene [50].

Typical electronic citation systems consist of laptops and external peripheral devices attached to them that cost police agencies thousands of dollars per unit [24]. In addition, maintaining a running fleet needs extra resources and money that takes away the focus from the core duty of the law enforcement [24,85]. Though, with smartphone technology and proper software applications, we suggest that computers and peripheral devices functions can be integrated to achieve the same results at a lower cost and easily maintainable system. Nevertheless, using smartphones adds portability and even extra features like the built-in cameras and artificial intelligence capabilities, which paves the road for future enhanced data collection techniques and improved user experience.

Big part of the cost of the citation system lies in buying hardware, for this reason, using smartphones to issue citations can be a reasonable solution for agencies that can't afford the expensive electronic citation system. Add to that, some police agencies already assigned mobile phones to their police officers [50].

Beside the ease of use and popularity among people, most of the smartphones nowadays have integrated peripheral devices that can be employed to do most of the functions of the external peripheral devices used in the computer-based electronic citation systems. Figure 3.1 shows the components of electronic citation system used with Oklahoma PARIS software [44].

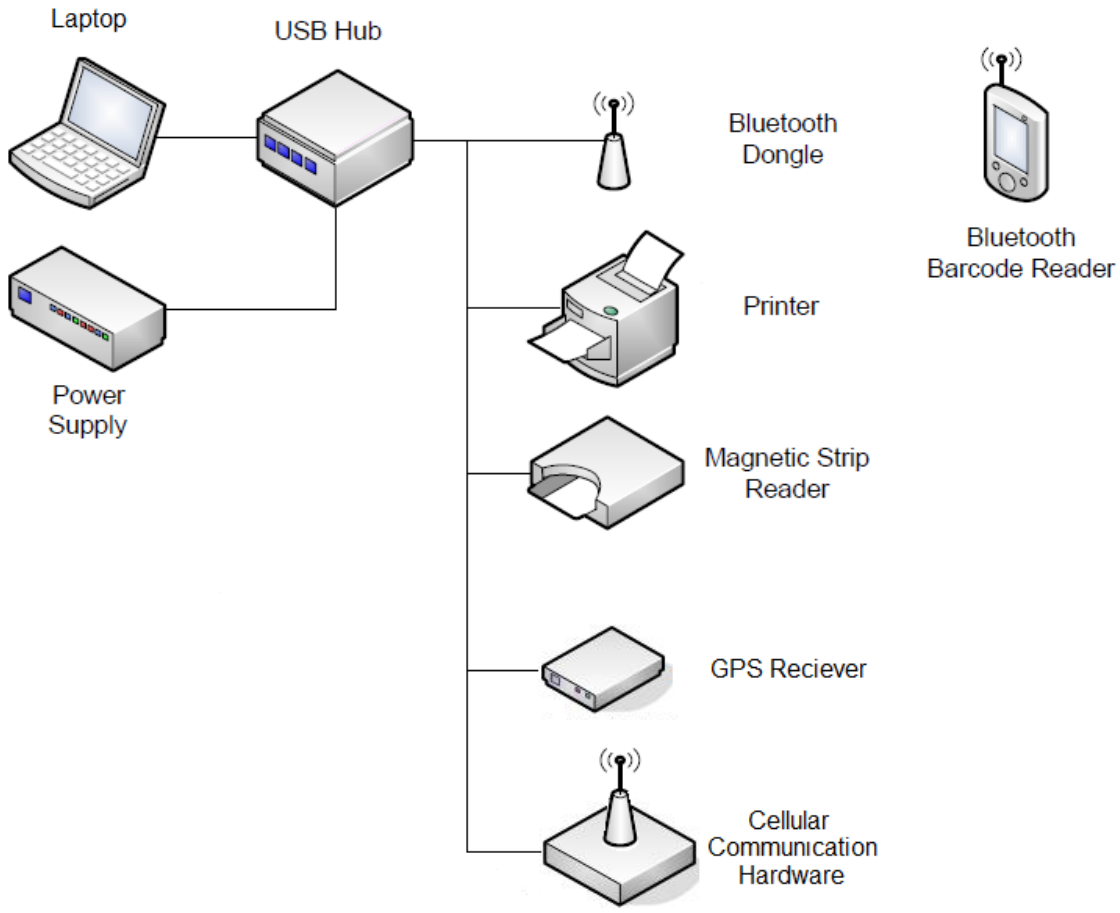

Figure 3.1 The Components of a typical Electronic Citation System [44].

A smartphone with the proper software and a printer can help police officers to issue and print citations similar to using a computer-based citation system. Most of the commercially available average price smartphones in the market now do have Global Positing System (GPS) hardware, high-resolution camera, Bluetooth hardware, touch screen, and high-speed data rate communication system [77]. Due to the recent advanced technologies that made the smartphone market very competitive, these features can be employed to harvest data in different ways to achieve similar results as the typical electronic citation system that uses laptop computer and peripheral devices.

3.1 Smartphone Peripheral Devices for Electronic Citation System

In this section I will discuss how the peripheral devices that are available in most of the commercial smartphones can be used to build a smartphone-based electronic citation system.

1. GPS

Electronic citation system computers are connected typically via USB to external GPS device to capture the location longitude and latitude coordinates while issuing a citation [85].

Most smartphones come with a built-in GPS [77] that is typically used to run location related applications to help users to find location, plan routes, etc. With the proper software on a smartphone, the location coordinates from built-in GPS module can be imported to mobile applications when needed, which can be recorded and processed for citation system.

2. Bar Code Reader

A wireless barcode reader is a very useful peripheral device in an electronic citation system [85]. Typical barcode readers connect to computers and use laser beams to scan driver licenses and vehicles barcodes. Wireless barcode readers are also available and can be connected to computers via USB dongles. The dongle connects to the barcode reader hardware via Bluetooth which allows the user to scan barcodes that are away from the computer [85]. 1D or 2D barcodes on driver licenses and vehicles hold a lot of the information that is required to be filled in to issue citations [8].

With the aid of a smartphone camera and the proper software, 1D or 2D barcodes can be decoded to retrieve the information they hold by analyzing their images

without the need for external barcode readers. Technology advancements in smartphones processors and software applications allowed smartphones to perform such operations easily.

3. Magnetic Stripe Readers

Older driver licenses used magnetic stripes technology to encode driver information which can be decoded with the aid of a magnetic stripe reader connected to a computer [7]. A Typical smartphone doesn't have capability to read magnetic stripe-encoded data without connecting external devices.

In 2016, the American Association of Motor Vehicle Administrators (AAMVA) Driver License/Driver ID 2016 standards specified that the PDF417 two-dimensional bar code symbology is the minimum mandatory machine-readable technology that must be present on driver license cards that are issued in the United States in order to be compliant with AAMVA standards [8]. This should reduce the need of having magnetic stripe readers in the electronic citation systems in the future. Currently all U.S. states are already issuing licenses with 2D barcodes symbology [84] which should reduce the need to have magnetic stripe reader in the electronic citation system.

4. Printers

A Printer in the electronic citation system is used to print paper tickets after a police officer fills in all needed information [85]. Printers are connected to computers via USB [85]. Most Smartphones allow connecting to printers using Bluetooth wireless technology. A feature that requires special printers and drivers that can connect wirelessly to smartphones.

5. Communication Hardware

Wireless communication hardware is needed in citation systems to transmit captured data to headquarters once completed in preparation to be sent to courts after approval [85]. Technology allowed newer smartphones to have internet via high data rate communication systems [77]. Internet connection on smartphones can allow transmitting packets wirelessly to and from headquarters.

Other features that are available in smartphones like voice recognition and Bluetooth makes the citation system less bulky and portable. Reliable voice recognition services allow for quicker text entry without having to type text, which adds a new degree of flexibility to the user. Bluetooth makes connecting a smartphone to any accessory that might be needed less bulky inside the patrol. Also, some smartphones allow for an external mouse and keyboard that makes data entry, if needed, very functional. To enable using these features that are available in smartphones, a mobile application is required to enable recording citation information and importing any data from the internal peripheral devices of the smartphone as needed. Available computer-based software can't do this function on smartphones as they run on different operating systems, communicate differently to peripheral devices, and aren't designed for smaller screen sizes.

In this study, I collaborated with Rodrigo Collao under the supervision of Dr. Joseph Havlicek at University of Oklahoma CITS lab to design a smartphone-based electronic citation system that can be used on a wide range of smartphones that run the Android Operating System (OS). It can also can be extended to run on Apple iOS devices using most of the code that was already developed. The proposed design does

not require any external devices except the printer to issue electronic citations.

The software application that was developed was named PARIS+ *app* after CITS lab PARIS software. It was designed to be a cheaper and simpler version of the PARIS software to provide low-cost electronic citation systems for smaller police agencies in Oklahoma that can't afford deploying and maintaining a full computer-based electronic citation system.

The PARIS+ app was designed to take a lot of the cost associated with the electronic citations process away by using the available smartphone peripheral devices and to make the best use of the current technology to achieve better results for government agencies. This is expected to help improve the quality, correctness, and timeliness of traffic records in the State of Oklahoma when deployed.

PARIS+ app system design included three main steps which are the software architecture design, implementation, and testing. The architecture design includes identifying what exactly the citation system should do, and then building a software architecture. The implementation part is the actual development of the PARIS+ app software. This part will be discussed in Chapter 4. The software testing results will be discussed in Chapter 5, which includes analyzing the technical specifications and business metrics related to PARIS+ app.

3.2 PARIS+ app Use Case

Modern software platforms and tools help in performing the task of building a software application, but they don't replace the need to design and structure them [55]. Like any other complex system, software applications must be structured on a solid basis. Failing to consider common scenarios can make the whole solution unusable in

deployment environments [55]. Software architecture is the model that defines how the structure of the software solution will meet all the technical and functional requirements [55]. Proper architecture identifies application requirements, eliminates the risks that can cause the software to be unstable, and helps to provide support for future needs [55]. The software architecture model specifies the major components of the application and how they communicate with each other to achieve the goal of the system [55].

The application architecture connects the business needs with the technical requirements [55]. A very important tool that help achieving this goal is the use case diagram [55]. A use case diagram uses the Unified Modeling Language (UML) to describe and capture the functional requirements of the software application. The UML is a standard language to write software blueprints. It is widely used to capture, visualize, and specify the artifacts of a software system [27].

The software architecture utilizes the use cases to design the structure needed to implement the software solution. A good architecture helps decrease the risk associated with using the technical solution to solve business problems and flexibly handles the evolution that will occur over time in software and hardware technology [60].

In the next section, I will discuss how we implemented PARIS+ use case to make sure it's properly designed to meet the expectations of the users. We collaborated with Captain Eric Cannaday at the Oklahoma Highway Patrol to build a proper use case model for PARIS+ app.

A Use Case diagram should describe the scenarios by which the application (PARIS+) interacts with actors (typically the police officer) and any external systems

(Printer and external database) to identify the software internal functionality [60]. It also helps identify the targets that the application helps the actors to achieve and the use of the system [60]. The PARIS+ use case diagram discusses only the functional requirements of the application. Other rules that are business or implementation related are not part of the use case diagram and will be discussed separately.

In PARIS+, the primary actor is the police officer who is using the software to achieve two main end goals. The first goal is to issue a citation to the violator. The second goal is to submit the citation wirelessly to the police agency or the court database which is represented in the use case diagram as the headquarter database (HQ database). The secondary actors are the printer and the HQ database which are external systems that the actor interact with to achieve his goals by using the application. The "Include" and "Extend" links defines specific relations from one use case to another.

The Include link from use case A to use case B means that use case B will always happen if use case A happened [60]. The Extend link from use case A to use case B shows that use case A might add functionality to use case B under certain circumstances [60]. Figure 3.2 shows the use case diagram for the PARIS+ app which includes the following functionalities:

a. Secure Login

The application should allow for secure login. The software should verify that the password is correct to allow users to login and may display an error message if user login credentials were incorrect.

b. Create a citation or warning

The police officer issues a citation or warning by filling data pertaining to

the traffic incident into electronic forms. To make filling these forms easier and faster, the data input fields should be organized inside these forms.

Some forms should allow the user to import the required information from the violator's U.S. driver license, vehicle identification barcode and location GPS coordinates. To achieve better record accuracy, form entries verification is needed to validate if all the required entries are complete. The software should notify the user in case of missing or incorrect entries. The software should validate if entries are correct and shouldn't allow submitting incorrect or incomplete records.

c. Print Citation

The police officer should be able to connect to a printer wirelessly through the application and print the citation or warning ticket that will be handed to the violator

d. *Submit to HQ database*

The police officer should be able to submit the electronic record of the citation to the headquarter database after all entries are validated.

a. Review Citations History

The police officer should be able to review the history of citations that he filled

The above use cases highlight the five main functionalities that PARIS+ app should provide.

3.3 PARIS+ app Design Considerations

Building an application to meet the technical and operational requirements and to be functional for the users requires a good software architecture that is designed with consideration for the user, the system, and the business goals [55].

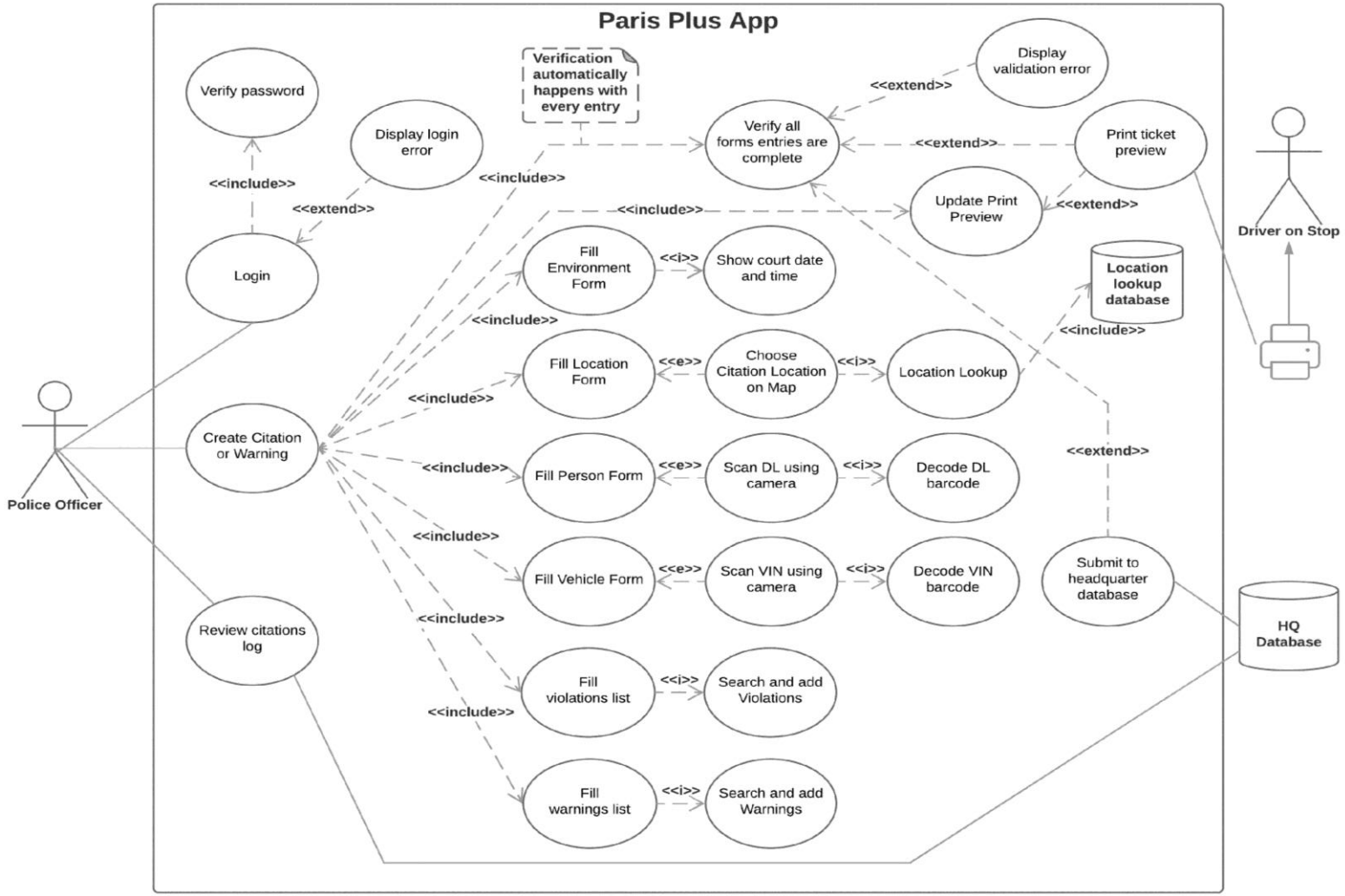

Figure 3.2 PARIS+ app Use Case Diagram.

Balancing between these three main parts is very important as they often overlap [55]. For each part, I will outline important quality attributes and areas of satisfaction and dissatisfaction that needs to be considered for the PARIS+ app architecture design.

a. User

As using a mobile device adds portability due to the small size and light weight, it also adds some limitations as small keyboard area which makes data entry more difficult than using a standard keyboard on a laptop. Such limitations should be accounted for in the design by utilizing the integrated capabilities of the smartphone to reduce data entry. Also, allowing the use of an external keyboard and mouse for data entry can be helpful in certain scenarios if extensive data entry can't be avoided.

As smartphones have considerably smaller screen display area than computers, the User Interface (UI) should be efficiently organized in this little screen so that it's not over populated with objects that make it complicated to use, nor too focused. Details such as tabs, font sizes and form views require proper selection. Feedback from end users helps make sure the design fits the needs.

b. System

Smartphones have limited storage and performance when compared to computers, which requires a lot of optimization and engineering when it comes to memory consuming operations, application responsiveness while loading forms, connecting to databases, printing, and data-entry. All these details need to be considered during implementation.

When choosing a smartphone device type, important considerations include

the screen size and resolution, processor unit characteristics, memory and storage. All these factors will directly affect the cost of the devices and an optimization needs to be done between the low-cost business objective and the performance requirements. A Samsung J7 Android smartphone and a Series Mobile Printer 5801/8001 were used for developing and testing the PARIS+ app. Designing a cheap and reliable electronic citation system was the main reason for choosing these models of hardware. Table 3.1 and 3.2 show the technical specification for both devices.

c. Business

The PARIS+ app was designed to provide a solution that is considerably affordable to police agencies. For this reason, cheaper smartphones and low-price Bluetooth printer needs to be selected which requires extra engineering in order to make it conveniently usable. Using the PARIS+ app with the minimum recommended hardware should cost less than $300 as per U.S. market prices on Jan 27th 2018. As the nature of police operations require operating in areas where there is no guarantee that internet will always be available, the design should account for a suitable application type that can perform local processing and operate with limited internet connectivity. Microsoft recommends that a rich client application is the best option for such requirements [56]. A rich client application is designed to provide an interactive, high-performance, and rich user experience as it needs to operate in stand-alone scenarios if there is no internet connection [56].

The PARIS+ app was developed as a mobile app to be used on Android smartphones after a recommendation that came from police agencies in Oklahoma [43] Some agencies have already issued Android devices to their officers. Nevertheless, the

Table 3.1 Summary of technical specifications of Samsung J7 smartphone [77].

Smartphone Feature	**Specification**
Model Number	SM-J700H
Camera Description	13 Mega Pixel
Screen Size	5.5 inches
Memory	1.5 Gigabyte
Internal Storage	16 Gigabyte
SIM Card	Dual
Bluetooth	Version 4.1
Dimensions	152.4 x 78.6 x 7.5mm
Battery	3,000mAh

Table 3.2 Technical specification of QS5801/8001 printer [72].

Printer Feature	**Specification**
Model Number	QS5801
Print Method	Thermal
Paper Width	58mm
Communication Interface	Bluetooth2.0/4.0 & USB
Printing Density	203dpi
Battery	1400mAH/7.4V chargeable Lithium battery

solution can be extended to run on Apple devices, which gives the police agencies the option in the future to run the PARIS+ app on different phones. For that reason, PARIS+ code was developed on Xamarin Forms, which should allow for easy extension in the future.

3.4 PARIS+ app Software Architecture

Irrespective of the type of the software application that is being designed, any application can be decomposed into logical groupings of software elements [57]. Each one of these groupings is called a layer. Layers help the designer to identify the different types of tasks performed by the software components, making it more feasible to design an application that supports reusability of the different software elements [57]. Inside every layer there can be a group of sublayers that perform specific tasks as needed, this type of architecture is called the "Layered Structure" [57].

Microsoft recommends using layered architecture for rich client mobile applications as it maximize the separation of concerns and improve the reuse and maintainability for the software application [56]. The layered structure categorizes the generic types of components that exist in the application where we can create a meaningful map for the architecture then use this map as a blueprint of the software application [57]. Dividing the application into different layers that have distinct functions helps the designer to optimize the mechanism of the application and maximize the maintainability of the software code [57].

The logical architecture of the layered system in its simplest form consists of a Presentation Layer, Business Logic Layer and Data Access Layer [56]. Each one of

these layers is responsible for performing a specific set of tasks and can communicate with the other layers to provide the functionality of the application.

The Presentation Layer provides the UI functionality by managing the user interaction with the software. Also, it provides a communication bridge into the Business Logic Layer logic as per the design [56]. The Business Logic Layer includes the components that implement the core functionality of the system and the business logic by encapsulating the business-related rules [56].

The Data Access Layer provides data access within the boundaries of the software and data exposed from other networked systems, which can be accessed through web services consumed by the software [56]. Figure 3.3 shows the PARIS+ app layered software architecture, which consists of a Presentation Layer, a Business Logic Layer and a Data Access Layer. The use of each one of these layers is discussed in detail.

a. *Presentation Layer*

The views in the Presentation Layer are responsible for the 3 main UI groups in the app which are the login page, the main page and the contact views page. The contact views contain all forms that are required to be filled to issue a citation. These forms are the environment form, the location form, the person form, the vehicle form, the violations list form, and the warnings list form. In addition to these forms, a print form is added to allow previewing and printing tickets.

The Model-View-ViewModel (MVVM) is an architectural pattern that supports the separation of layers [14]. The View refers to the UI, the Model refers to the underlying data, and the ViewModel refers to an intermediary layer between the View

and the Model [14]. The View and the ViewModel are often connected using data binding, where the binding context of the view is usually an instance of the ViewModel. MVVM helps produce a clean representation of UI [14].

The MVVM architectural pattern is heavily used in the PARIS+ app to support the UI in the Presentation Layer and the communication between the Presentation Layer and the Business Logic Layer. The User Control library comprises multiple elements such as the interfaces that are used to display popups and notifications for the view and the validators that help verifying all entries that are required for a complete citation are filled.

The validators communicate with the Business Logic Layer to provide the complete functionality of the validators and notify the UI to avoid submitting incomplete or incorrect citations. The CustomWebView employs Razor View engine to enable simple preview and printing of the actual ticket. The Razor View is a Microsoft package that enables fast and clean lightweight runtime assembly to combine server code with HTML to create dynamic web content. It allows for fast transition inside code between HTML format and C# code [11].

The Razor View also allows for Data Binding from the code behind which makes displaying dynamic content much lighter in terms of processing which have great value for mobile applications [2]. Using CustomWebView in the PARIS+ app allowed saving memory and CPU usage on the mobile phone that are typically very high as computer-based solutions involve using the common PDF formats to display dynamic content [2]. Map UI is added in the contact views to allow the user to precisely choose the location of the incident on a map view using the touch screen. The PARIS+ app

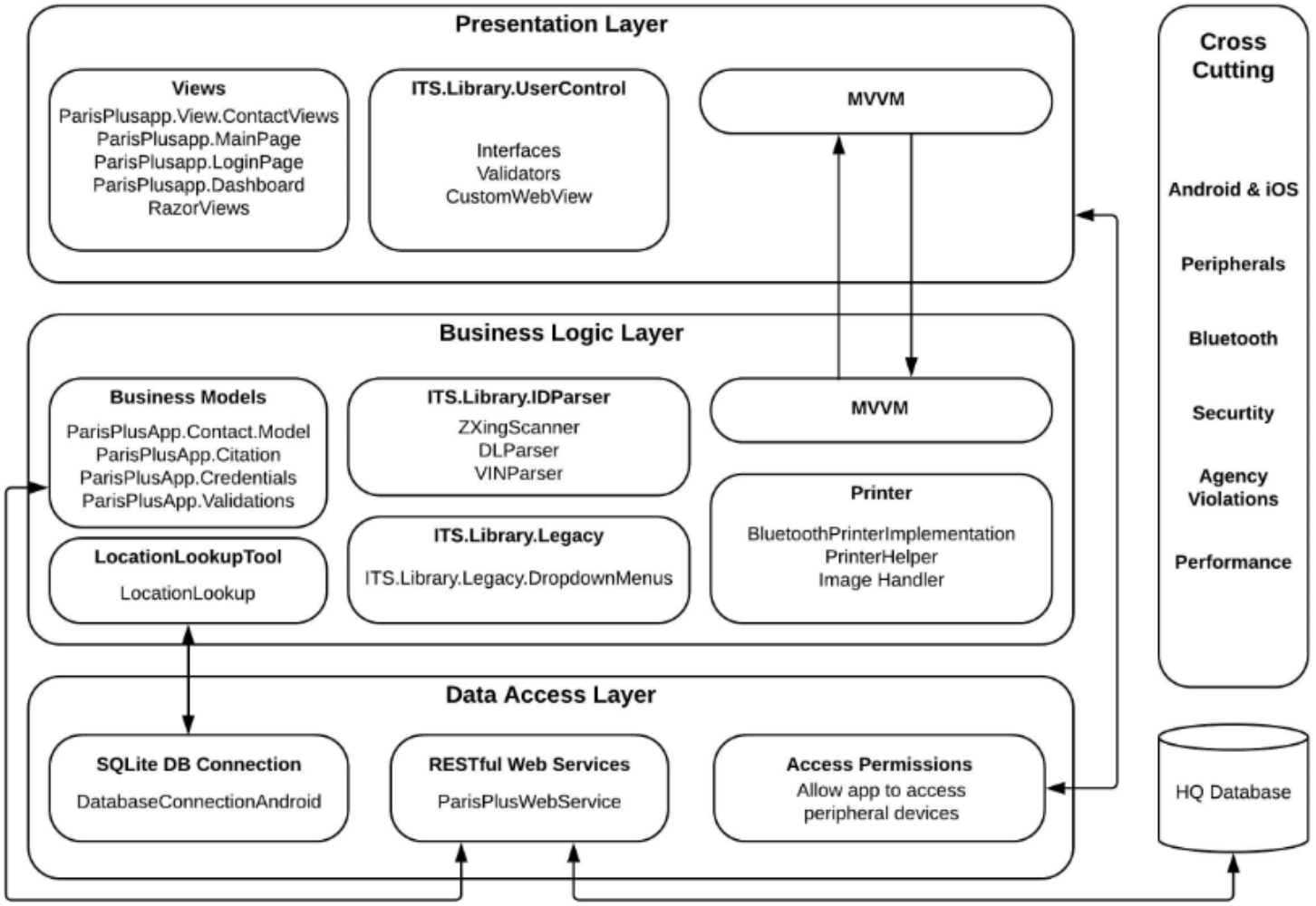

Figure 3.3 PARIS+ app Layered Software Architecture.

Location Lookup Tool uses the coordinates of the accident location to automatically fill the location form on the contact view.

b. *Business Logic Layer*

The Business Logic Layer in PARIS+ app encapsulates multiple components that implement the core functionality of the software. The Business Models consists of the forms models that will be filled with the data entered by the user. It also contains other models that implements the validation functionality and the user login credentials models.

The Location Lookup Tool helps the user to fill all the required entries on the location form automatically after the incident location is selected on a map. To achieve this functionality the Location Lookup Tool connects to an internal SQLite database that encloses several tables containing geographical information about Oklahoma cities, counties, streets, intersections, etc.

The DLParser module supports importing the driver license and vehicle information into forms in the UI. It opens the smartphone camera and scans the barcodes from the driver license and the vehicle, which are then decoded to populate the data to the respective form fields automatically. This feature significantly reduces data entry that is required to fill the person and vehicle forms.

The Legacy module supports standard entries using the dropdown menus in the forms. Dropdown menus have great value for the PARIS+ app as they allow for standard data inputs. This feature helps in reducing data entry errors at the user end and facilitate generating accurate reports and statistics at the agency side. The Legacy

module imports standard XML business data as defined by every police agency to identify specific business rules that are generic or agency dependent. This allows for reusability of the application between different agencies. The Printer module supports the printing functionality by allowing the application to communicate with the printer via a Bluetooth connection to allow printing tickets.

c. *Data Access Layer*

The Data Access Layer provide access to the data hosted within the boundaries of the system and to data exposed by networked systems. SQLiteDBConnection is responsible to querying the location tables to provide the Location Lookup class with the needed geographical data. Oklahoma Reference tables contains hundreds of thousands of data entries that are queried to populate the location entries.

SQLite is a free SQL database management system that implement a serverless, zero configuration SQL database to read and write directly to ordinary disk files. This makes it a popular choice for embedded systems [78]. The RESTful web services provide the communication between the application and the HQ database to submit citations to court or agency database as soon as they are issued.

RESTful stands for Representational State Transfer which is architectural web service that uses four interfaces, namely, create, read, update, delete or CRUD to allow Uniforms Resource Identifiers (URIs) to expose a set of resources that identify the targets of the interaction with other clients [23]. The Access Permissions module allows the application to use the smartphone resources as camera, internet, Bluetooth, GPS, flash light, read external memory and others as needed.

In the next Chapter, I will discuss PARIS+ app implementation.

Chapter 4

PARIS+ app Implementation

The second part of building the PARIS+ app is the software development of the application. The PARIS+ app was designed to target mainly Android OS smartphones that have the required peripheral devices.

Android OS is an open source software stack [34]. The original android source code was made freely available to organizations with the purpose that it can be a platform for carriers, Original Equipment Manufacturers (OEMs), and developers to make creative ideas a reality and to present a real-world product that enhance the mobile experience [34].

Android was introduced by a group known as the Open Handset Alliance (OHA). The OHA is a group of companies that was led by Google to maintain and further develop the Android system [34]. The OHA expanded after more companies joined to support introducing improved Android devices to the market [34].

Android features are built inside the Android OS. The Android OS is available through the framework Application Program Interface (API) [36]. Generally, as the API level increases, functionality adds up. In few cases, some functionality gets deprecated [36]. The Android apps can be written in Java, Kotlin and C++ languages [29]. The Android Software Development Kit (SDK) tools compile the written code along with other app resources to produce the Android Package (APK) file which is an executable file format for Android OS use [29]. Google developed Android Studio as an Integrated Development Environment (IDE) to help developers develop apps for

Android devices [33].

Since the first Android operating system 1.0 (API Level 1) was formally unveiled, Google and the OHA have developed a series of releases. Every new API level release represents an upgraded version of the Android OS. Along with the API levels sequence, the Android OS uses code names for every family of APIs that are named alphabetically after what have been described as “tasty treats” [30].

Similarly, Xcode is the major IDE for Apple devices like iPhone, iPad, iMac, etc. [10]. Apps can be developed using Swift or Objective-C programming languages [9]. Android Studio and Xcode provide developers with everything they need to build the best smartphone app experience by providing application models that allow deploying apps broadly to millions of users across a wide range of devices [90].

As the Android and iOS platforms are different from each other, a developed app for one platform using the standard IDEs can’t operate on the other because they are designed for different platforms and use different SDKs that produce different app file types.

4.1 Using Xamarin Forms to Build the PARIS+ app

Over the past few years, the available platforms for building mobile applications have expanded a lot where an entire ecosystem of platforms has emerged [1]. Xamarin is a unique platform in this space as it offers class libraries and runtime environment that is capable of developing apps that works for Android, iOS and Windows Phones using the same code [1]. It compiles native applications that are fast enough for demanding apps. All this is offered on a single language which is C# [1].

The PARIS+ app was developed to be used for Android phones. We elected

to use Xamarin Forms to develop the app. Xamarin Forms was first introduced by Xamarin in 2014 as a cross-platform UI toolkit that allows developers to easily create native user interface layouts that can be shared across the two most popular mobile platforms available in the market, namely Android and iOS [93].

Xamarin combines all the functionality of the native platforms and adds extra features that makes the platform widely preferred in the sense of efficiency and extendibility [1]. The following points summarize some of the features that Xamarin offers.

1. *Native languages Interoperability*

Xamarin allows for directly invoking the native language on each specific platform. Such as Java for Android or Objective-C for iOS. This gives developers the power to use a wide array of third party code that has already been created in the native languages if needed [1].

2. *Complete Binding for the underlying SDKs*

Xamarin contain bindings for almost all the underlying platform SDKs for both Android and iOS which decrease the runtime errors and allow for easy use and navigation. It also provides robust type checking during development [1].

3. *Modern Language Constructs*

Xamarin applications are written in C# code. This adds significant improvement over Java and Objective-C programming languages such as functional constructs, Lambda, LINQ and others [1].

4. *Modern Integrated Development Environment*

Xamarin is available on Visual Studio IDE on Windows OS and Visual

Studio for Mac on Mac OS. Visual studio is a modern IDE that allow for modern features such as code auto completion, project and solution management system, integrated source control and others [1].

5. *Mobile Cross Platform support*

Perhaps the most attractive feature of Xamarin is that it offers cross platform support for the two major mobile platforms, namely Android and iOS [1]. Applications for these platforms can share up to 90% of the C# code. Xamarin.Mobile library offers access to native resources across all three platforms. This feature adds efficiency, extendibility, and reduce both development cost and time [1].

6. *Xamarin Application output*

When Xamarin applications are compiled, the result is an Application Package which matches the type of operating system it's going to run, i.e. apk file for Android and app file for iOS. These files match exactly the application packages built with the platform's default environments and hence they are deployable in the same way [1].

By using Xamarin Forms to develop the PARIS+ app, extra flexibility is given to the police agencies in the future to deploy PARIS+ to different operating systems. Developers can use most of the code that was already developed rather than having to spend a lot of time to deploy the app on Apple iOS, which means more cost to develop another app.

Another benefit of using Xamarin forms to write the PARIS+ app in C# code is to be able to easily port some of the libraries and parts of the code that were already written for CITS PARIS software in C# code. While this code was already available, it

was also tested in the currently available PARIS desktop software for a long period of time.

PARIS+ app is a shared work between me and Rodrigo Collao. This work has led to police agencies in Oklahoma having a significantly cheaper electronic citation software. My own technical contribution towards building PARIS+ app includes:

- Presentation Layer Classes: Violation Pages, Warning Pages, Print Preview Page.
- Business Logic Layer Classes: Violation Page View Models, Warning Page View Models, Printer Preview View Model (Form View Model), Custom Web View, Driver License Parser, VIN Scanner and Decoder, Location Lookup class.
- Data Access Layer Classes: Local Database Connection.

I will also present the work that I contributed in collaboration with Rodrigo which includes:

- Presentation Layer Classes: Contact Page Holder, Environment Page, Person Page, Vehicle Page.
- Business Logic Layer Classes: Contact Page Holder View Model, Environment Page View Model, Person Page View Model, Vehicle Page View Model,
- Data Access Layer: Access Permissions.

PARIS+ app refers to the mobile app that was developed. PARIS software refers to the CITS PARIS desktop software.

4.2 PARIS+ app User Interface Flow

The Presentation Layer in the PARIS+ app consists of multiple pages to help the user to interact with the software as shown in Figure 4.1. When the app starts, it navigates to the Login Page where the user can enter his credentials to login or create an account. If the user credentials were incorrect, the software will display "Wrong Credentials" alert and navigate back to Login Page after the user acknowledges the message. After the user login is successful, the software navigates to the Main Page where the user can start using the app functionality. The "Dashboard" button opens another page that contains a list of the citations that were successfully submitted by the user.

The "Create Citation" button opens a new Xamarin Forms Tabbed Page. The Xamarin Tabbed Page consists of an array of tabs and a large detail area below where the user can select between tabs to load different content on the screen [18]. The information required to be collected to issue a citation is captured inside different input fields. Every related group of fields are located together inside the same tab. Tab headers are identified using meaningful icons with no text. This gives nice UI look and saves display space. When the user taps "Create Citation" button, a new instance of the PARIS+ app ContactPageHolder class is initialized to load a tabbed page with 8 tabs. Each tab is linked to a Content Page object. These Content Pages are separate view objects and appear whenever the respective tab is selected. The UI of all the pages is written using XAML markup language. XAML refers to the eXtensible Application Markup Language and is particularly suited for use with the MVVM application architecture.

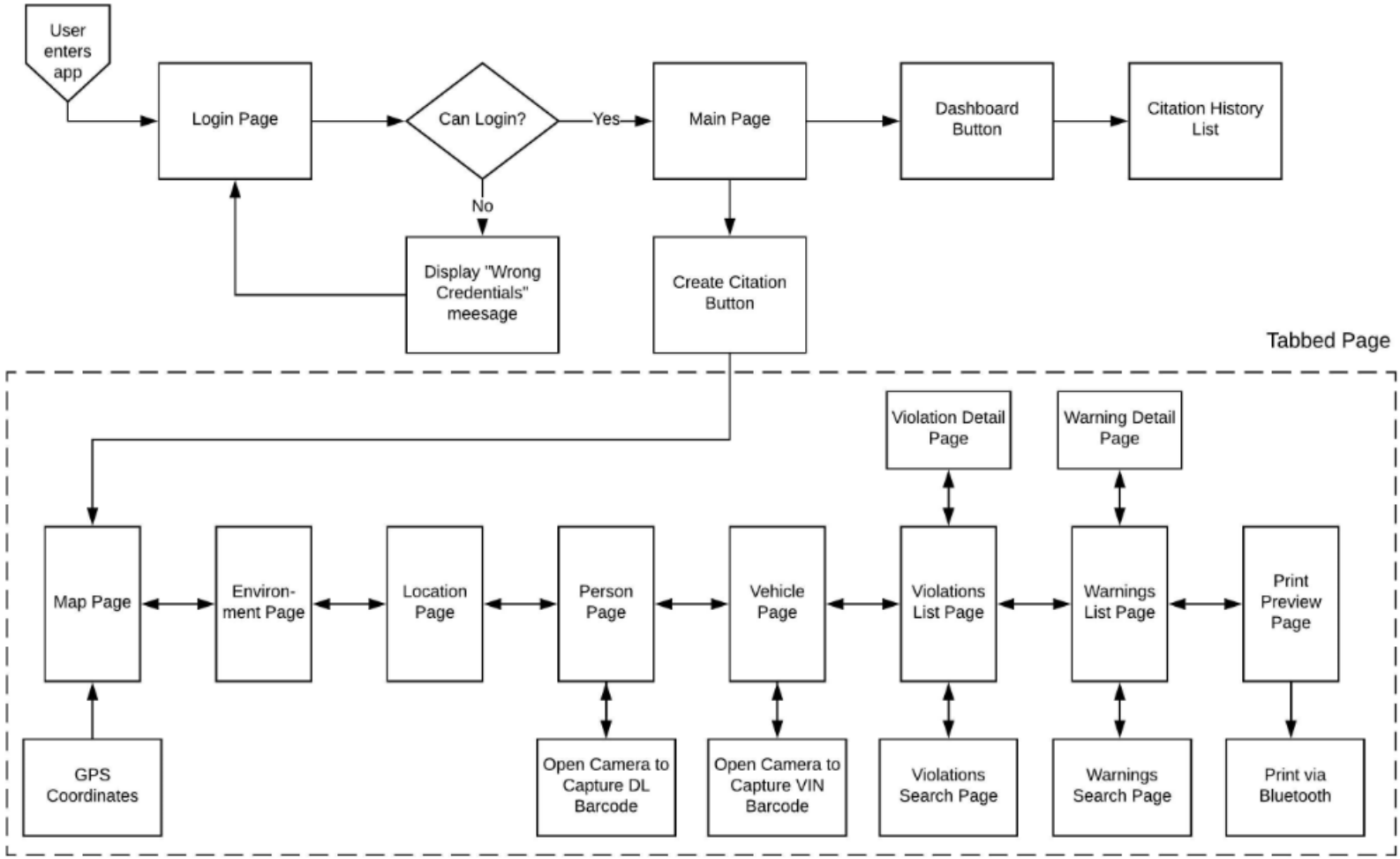

Figure 4.1 PARIS+ UI Flow.

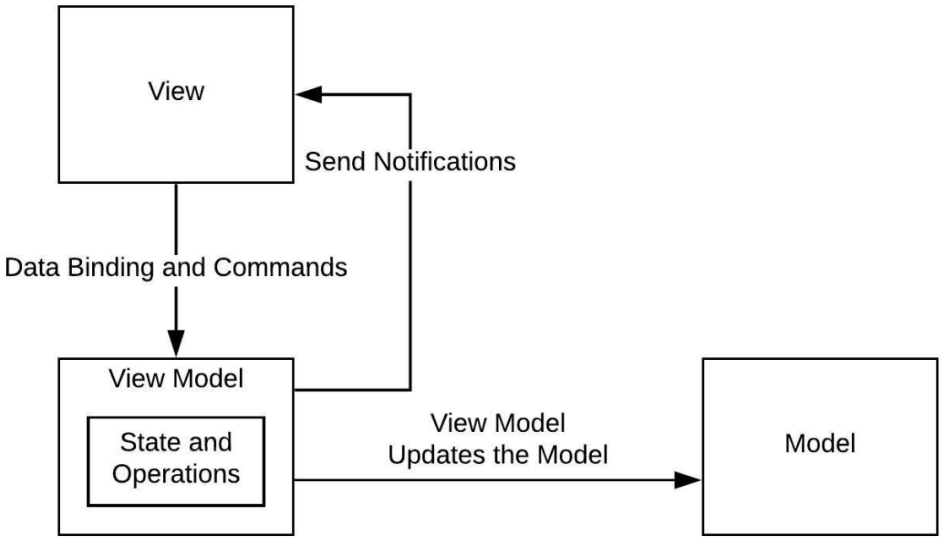

Figure 4.2 MVVM Architecture.

XAML defines the View. The view objects inside the View are linked to the ViewModel using data bindings [15]. Each Content Page has its own View, ViewModel and Model to support the overall functionality as depicted in Figure 4.2. The ViewModel is an intermediary state between the View and the Model. It handles the view logic by sending notifications and providing data from the Model to the View and vice versa [54]. The Model includes the application data model [54]. The "Commands" allow the View to make method calls directly to the ViewModel.

The AsyncTask Class in Xamarin Forms provides proper usage of the UI thread [89]. It allows performing short background operations asynchronously from the main thread and publish the results after the execution is complete. This prevents blocking UI responsiveness while performing internal operations [89].

The UI utilizes multiple view objects that allow the user to interact with the application such as buttons, entries, labels, pickers, switches, and listviews. A "button" object is used to execute an operation. An "Entry" object is used to allow entering a

single line of text. A “Label” object is used to display text. A “Picker” object is used to allow the user to select an item from a list. A “Switch” object is used to allow the user to choose a toggled value. A “ListView” object is used to display a vertically scrolling list. Alerts are smalls windows that prompt the user to make or acknowledge a decision before proceeding [32]. A Toast is a dialog that provide a simple feedback about an operation in a small popup [35]. Activity indicators are used to indicate to the user that an operation is loading [91].

In order to make clear explanation of the presented work, starting from the next section till the end of the thesis, I will style the class names in `Courier New` font. I will also style the UI button names in **Calibri** bold font.

4.3 Contact Page Holder Class

When the user taps the **Create Citation** button in the `Main Page`, the `Contact Page Holder` class in the PARIS+ app is executed inside `AsyncTask` code block. This avoids blocking the `Main Page` UI responsiveness. Before this task starts executing, the **Create Citation** button is disabled, and a running activity indicator icon is displayed to notify the user of an ongoing operation in the background as shown in Figure 4.3. This sequence was developed to prevent loading multiple instances of the `ContactPageHolder` class unnecessarily. The first tab on the `ContactPageHolder` contains the `Map Page`. The `Map Page` contains a cross hair on a google map view to allow the user to choose incident location easily and precisely. The **Get Location** button calls the `LocationLookup` class to populate location-related fields by running inside another `AsyncTask` code block as shown in Figure 4.4.



Figure 4.3 Main Page while Create Citation button is loading.

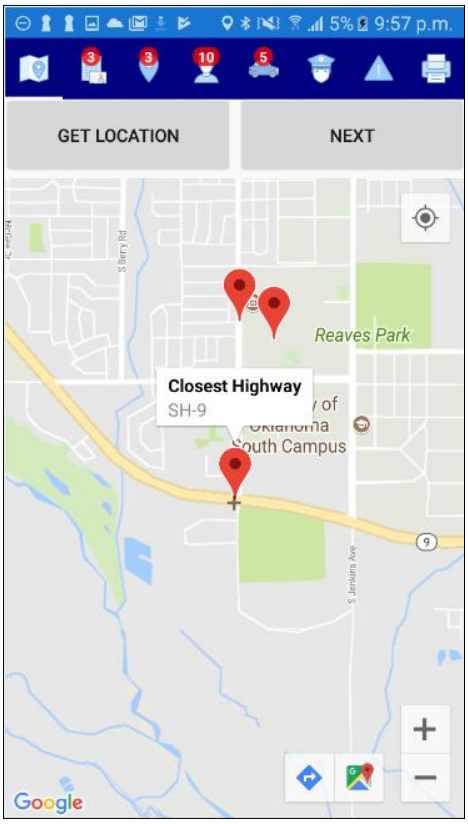

Figure 4.4 Map Page with closest highway and closest intersection locations populated.

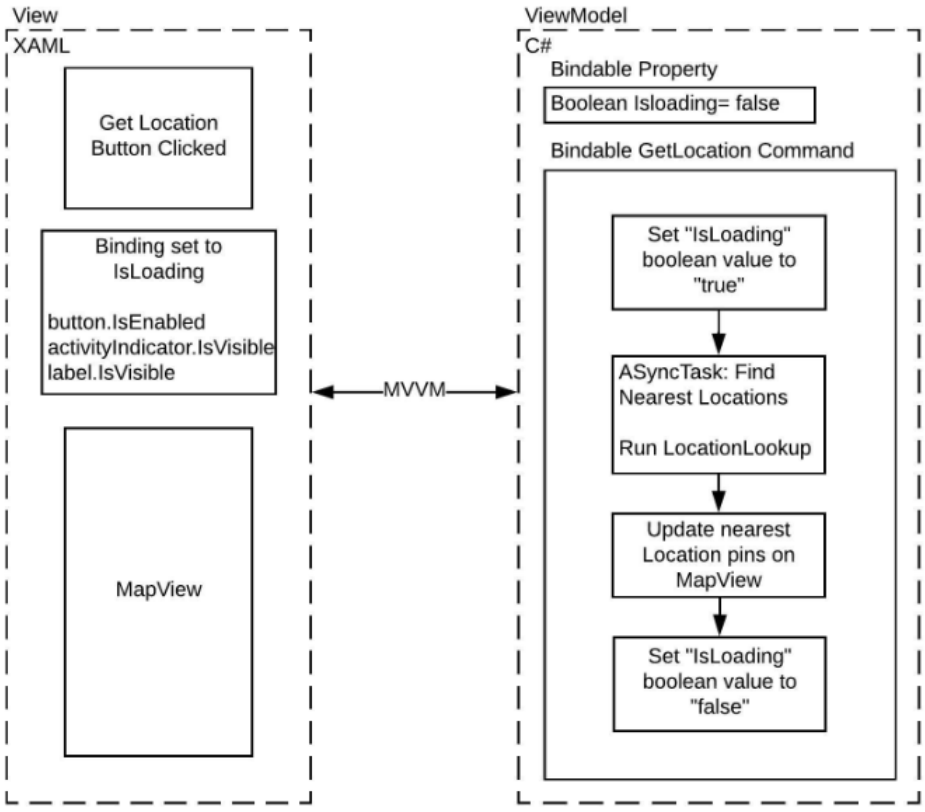

Figure 4.5 Get Location Command binding between Contact Page Holder View and ViewModel.

The following commands are executed in sequence when the user clicks **Get Location** button as depicted on Figure 4.5:

- Notify the view to display an activity indicator and disable the button till the task is completed.
- Call `LocationLookup` class to calculate the location-related fields.
- After this process finishes execution, the `Map Page` is updated with name, latitude and longitude coordinates of the closest Highway and closest intersection that were calculated in the `LocationLookup` class.
- Notify the view to remove the activity indicator and enable the button.

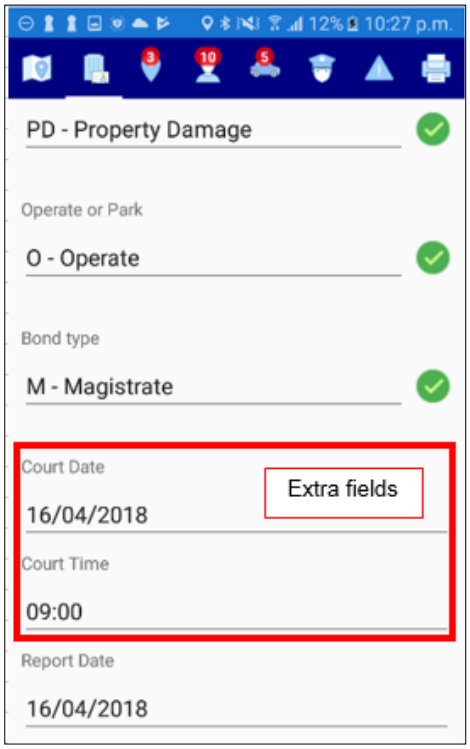

Figure 4.6 Environment Page with extra fields displayed depending on Bond Type field selection.

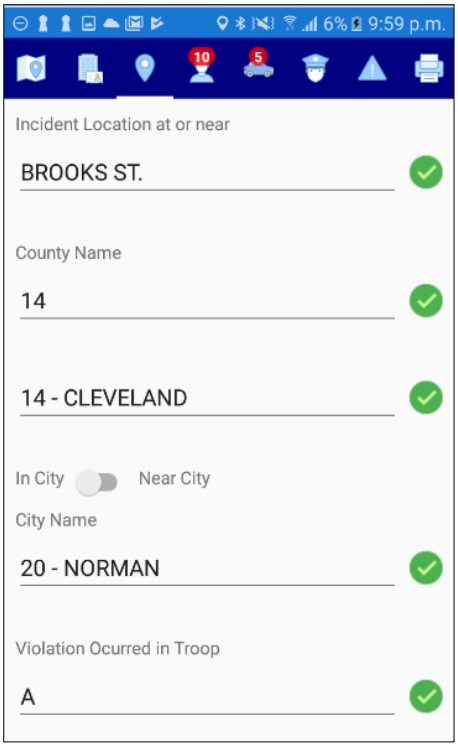

Figure 4.7 Location Page with populated fields.

4.4 Environment Page

The `Environment Page` contains fields that are related to the citation environment such as the date and time of the incident, collision type, operate or park, bond type, and the report date. The default values for the date and time fields are filled with the current date and time.

The Bond Type field is chosen via a picker menu object. The available options are displayed when the picker is tapped. Based on the selection made, extra fields might appear in the `Environment Page` as shown in Figure 4.6. The `EnvironmentPageViewModel` has a property that is bind to the selected Bond Type value to notify the view. This is achieved by changing the “IsVisible” property of

the extra fields' layout.

4.5 Location Page

The `Location Page` contains location-related fields such as closest highway, closest street, county, city, near or in city switch, troop, east and north grid that pertain to the incident. All the fields in the `Location Page` are imported from variables calculated by the `LocationLookup` class. These fields are updated every time the user selects a location on the `Map Page` and taps the **Get Location** button.

When the user navigates to the `Location Page`, the calculated values are imported to update all the fields in the page as shown in Figure 4.7. If the user decided to do changes for the fields that were populated, manual entry will still be available. The county and the city are both selected via picker menus. These two menus contain a list of all counties and cities that are in Oklahoma. When the user selects the county, the `LocationPageViewModel` class is notified to run a method that filters the contents of the city menu list. The filtered list contains only the cities that are inside the selected county. Choosing a city from a shorter list save user's time.

4.6 Person Page

The `Person Page` contains multiple fields to capture information related to the violating person involved such as name, date of birth, driving license number, Employer, etc. The `Person Page` also contains **Import Person** button which opens the camera to allow the user to scan a driver license barcode as shown in Figure 4.8. Upon successful scanning, the barcode information is decoded by the `DLParser` class to populate the fields in the `Person Page` with the decoded information as shown in Figure 4.9.



Figure 4.8 Scanning driver license barcode.

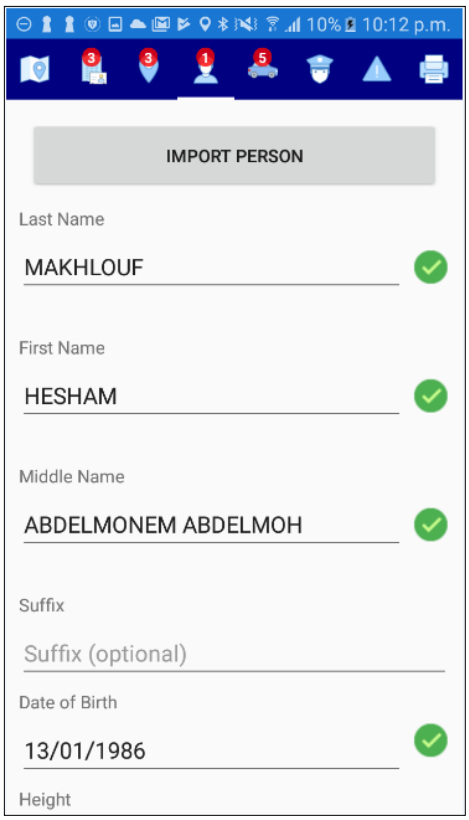

Figure 4.9 Person Page with populated fields.

4.7 Vehicle Page

The `Vehicle Page` contains fields that are related to the incident vehicle as license plate number and state, Vehicle Identification Number (VIN), vehicle information and Hazmat information. Similar to the `Person Page`, the `Vehicle Page` contains a **Scan VIN Number** button which opens the camera to allow the user to scan the VIN barcode. Upon successful scanning, the `VINDecoder` class decodes the barcode information to populate the VIN number, the vehicle model year, and the vehicle make fields. Figure 4.10 depicts the sequence of scanning the VIN number between the `VehiclePageView` and `VehiclePageViewModel` as follows:

- The **Scan VIN Number** button in the `VehicleView` triggers the Scan VIN Number command in the `VehiclePageViewModel`.
- The triggered command checks if the scanner is not already scanning and push a new page to open the camera. The camera remains open and adjust focus automatically till the barcode is detected. This function is implemented using a library called ZXing.Net.Mobile.Forms ported to the PARIS+ app [92].
- The `VINDecoder` class starts by checking for invalid VIN code. If the VIN number is invalid, the user is notified with a toast message that scanning the VIN number was unsuccessful.
- Upon successful scanning, the result of barcode is passed to the `VIN decoder` class to decode the VIN number

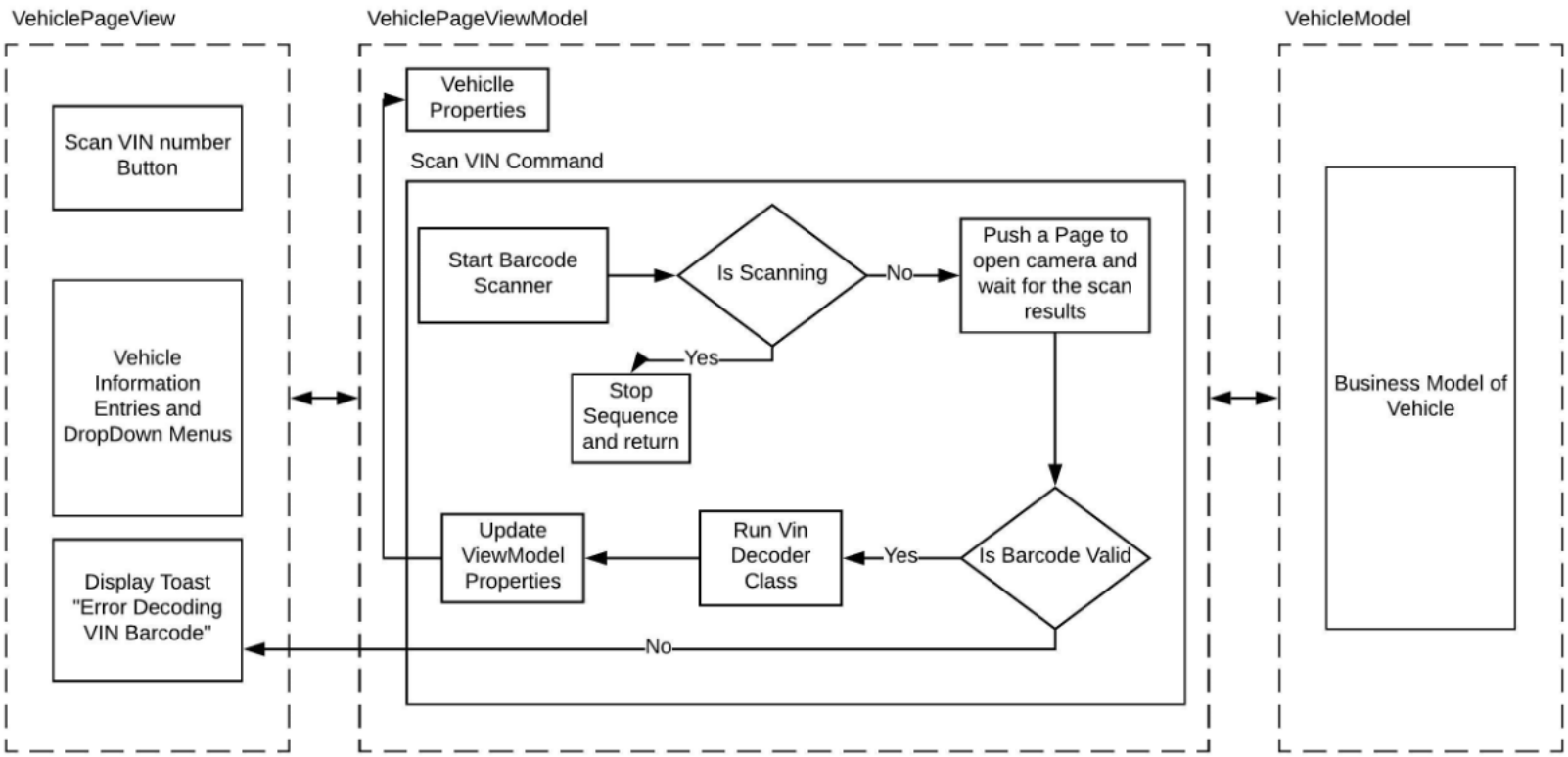

Figure 4.10 Using MVVM in Vehicle Page.

, vehicle make, and model year.

- The fields in the Vehicle Page are then populated with the decoded information.

4.8 Violation Pages

The Violation tab contain three pages that allows the user to search, add, delete, and review violations. The user can also add remarks for every added violation on the `ViolationPage`.

The `ViolationsListPage` contains the **Add Violation** button and a `ListView` object. The `ListView` displays a list of the violations added to the ticket. When the user clicks **Add Violation** button, the software navigates to the `ViolationsSearchPage` to allow the user to search and add violations as shown in Figure 4.11. The **Add Violation** button is bound to a command in the `ViolationsListPageViewModel` class to open the `ViolationsSearchPage`.

The `ViolationsSearchPage` contains a `ListView` object that displays a list of all the available violations as given by municipalities. I ported the `Dropdown` class from PARIS software and added some methods to make handling the violation `Dropdown` objects easier to filter and search. When the user enters text in the Searchbar field in the `ViolationsSearchPage`, the `ViolationsSearchPageViewModel` class is notified to run a method that search the agency violations list. This method uses the search text to search the available violation wording and display a filtered violations list.



Figure 4.11 Violations Search Page with search text “sb3”.

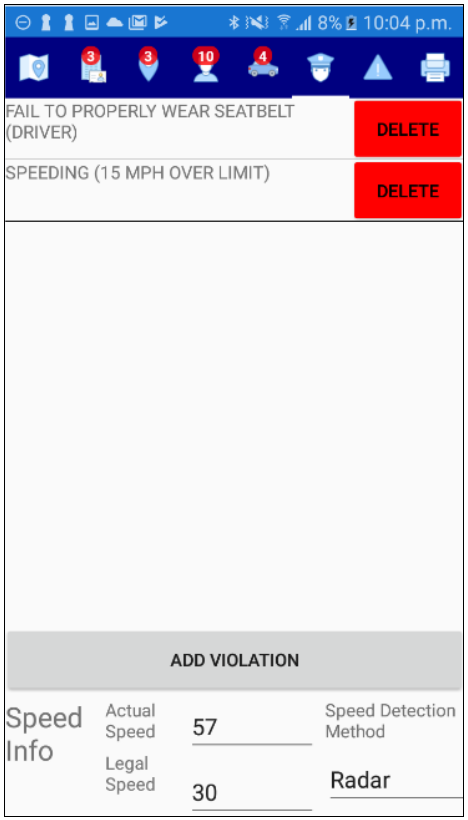

Figure 4.12 Violations List Page with added violations including a speeding violation.

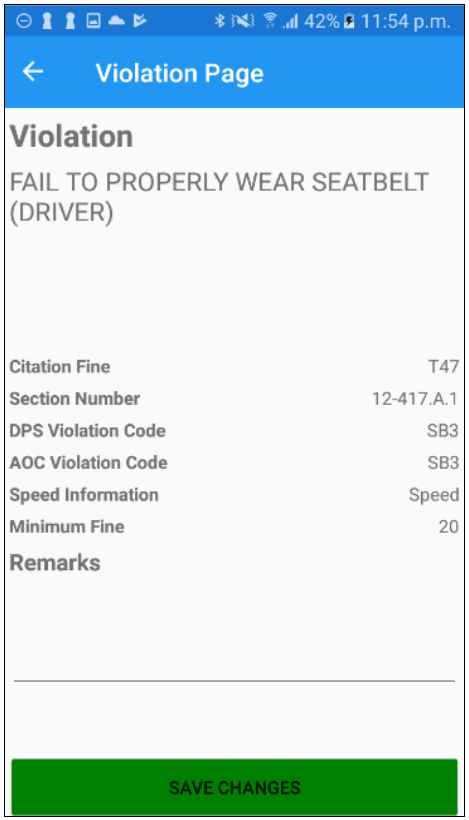

Figure 4.13 Violation Page containing details about selected violation.

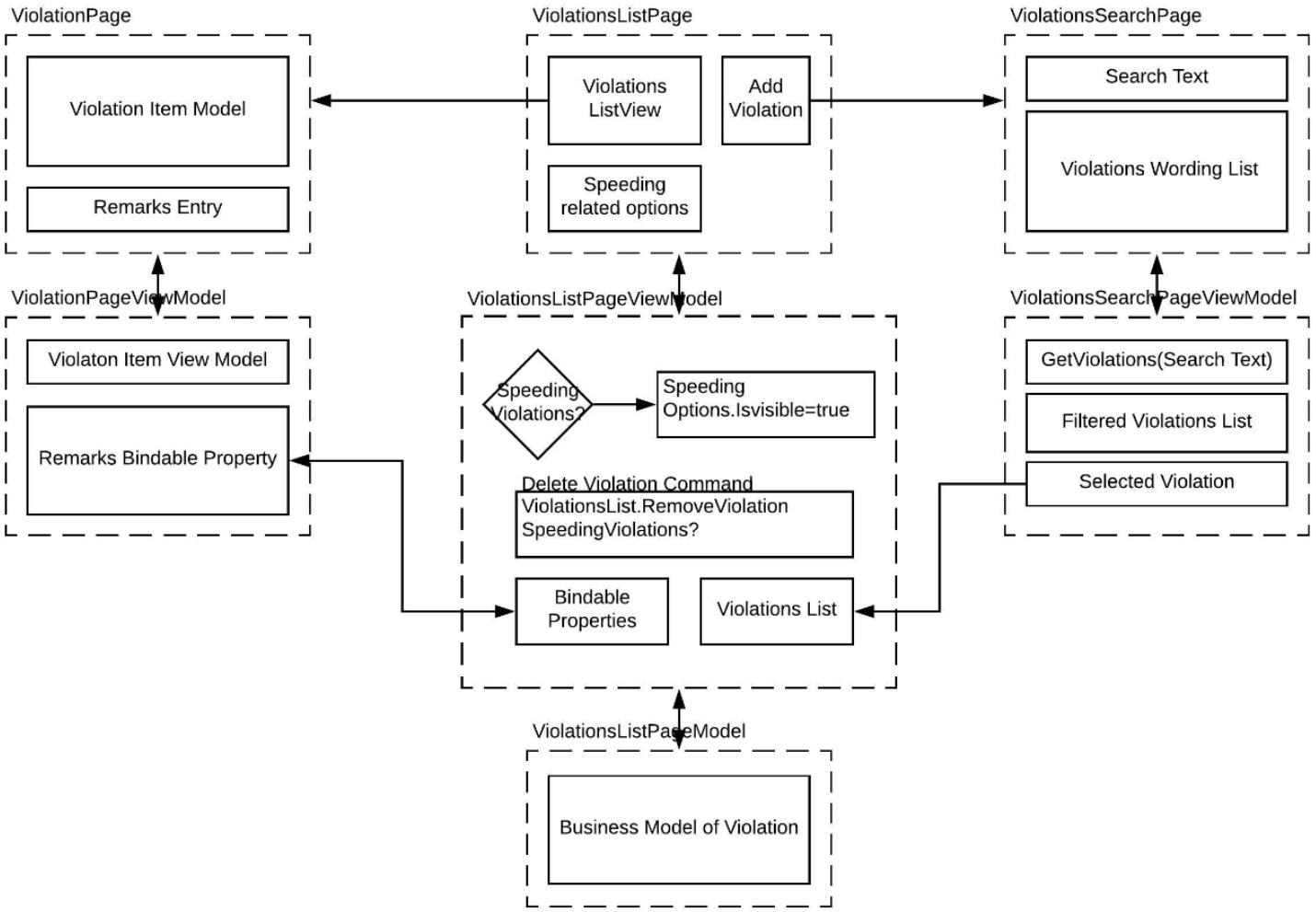

Figure 4.14 Searching, adding, and removing violations.

When the user selects an item from the filtered violations list, the selected item raises an event and sends a `DropdownItem` object to the `ViolationsListViewModel` class. This object is sent to the added violations list. On every violation item added or removed from the added violations list, the `ViolationsListViewModel` class checks if the received violation wording contains the word "Speeding" and notifies the view to display extra options that are related to speeding violations as shown in Figure 4.12.

Every item on the `ViolationsListPage` has a **Delete** button attached to it. Every **Delete** button allows the user to delete one violation from the list. The **Delete** button is bound to a command in the `ViolationsListViewModel` that passes a parameter defining which item will be deleted. If the user clicked an item on the `ViolationsListPage`, a method in the `ViolationsListViewModel` class is executed which opens the `Violation Page` shown in Figure 4.13. The `ViolationPage` contains more details about the selected violation as violation wording, violation code, minimum fine, etc. Also, the user can add his remarks on every violation that will be later printed on the ticket. When the user finishes editing the remarks in the `Violation Page`, he can click save changes or navigate back to the `ViolationsListPage`.

The `ViolationPage` and the `ViolationsSearchPage` utilize MVVM in is simplest form [54] by having only a View and a ViewModel without a Model. Figure 4.14 depicts how the MVVM model is used in the different violation pages to support the needed functionality.

The `WarningsListPage` contains a list of the warnings that will be charged to the violator. The functionality is very similar to the `ViolationsListPage` except for using warnings instead of violations as a source for the List and ListView objects.

4.9 Print Page

The `Print Page` allows the user to print and preview the ticket as explained in Figure 4.15. The ticket view is displayed as a cshtml web page.

A `CustomWebView` custom renderer was developed to display the web page and to allow:

1. Dynamic display of the contents of the ticket page. This was achieved through binding different objects inside the `FormsViewModel` class. The `FormsViewModel` class imports the values from every content page ViewModel. These values are then bound to a cshtml web page that contain the ticket template with the aid of the Razor Syntax. Razor Syntax allows combining html markup with C# code to display dynamic web pages.
2. Construct an image from the filled ticket view for printing. An instance of the Android `WebView` is used to draw a canvas object of the displayed content.
3. When the **Print** button is clicked, the canvas object is converted to a bitmap image. The bitmap image is then used for printing.

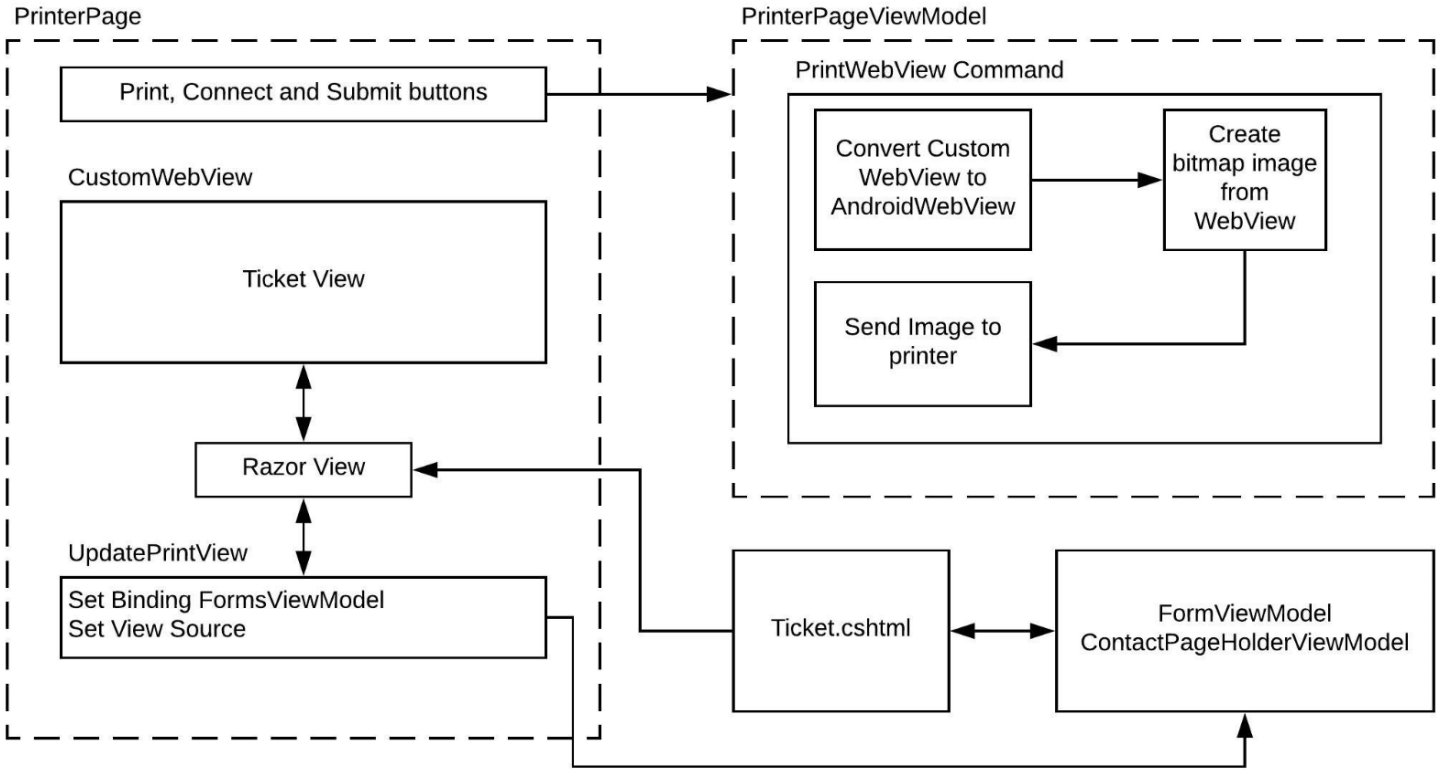

Figure 4.15 Print and Print Preview Flow.

4.10 Location Lookup Class

Location data is very critical on any citation form. The quality of the collected location data is very important for the reasons mentioned earlier in Chapter 2. PARIS desktop software contains a location lookup tool to increase the accuracy of the recorded location data and reduce officer error in field entries. In addition, auto populating all the needed location information from a point selected on a map using its latitude and longitude coordinates makes location selection easy, precise, and save officer's time. The PARIS+ app provides a similar function to the PARIS location tool. The user can choose the incident location on the `Map Page` then click the **Get Location** button to fill location-related fields automatically.

I ported parts of PARIS location tool code to develop the `LocationLookup` class on the PARIS+ app. The `LocationLookup` class uses the location longitude and latitude coordinates to calculate the county code and name, city code and name (if inside city boundaries), east and north grid, closest highway, closest city street, and troop. The `LocationLookup` class depicted in Figure 4.16 use the latitude and longitude coordinates from the `Map page` to run multiple methods that query a database and perform mathematical operations to estimate location-related data. These results are then used to fill the `Location Page` fields and update pins on the `Map Page` tab.

PARIS Location Tool connects to a "Reference" database that contains thousands of entries that link different Oklahoma counties, cities, street names, highways, grid corners, troops, etc. to their latitude and longitude coordinates. This data

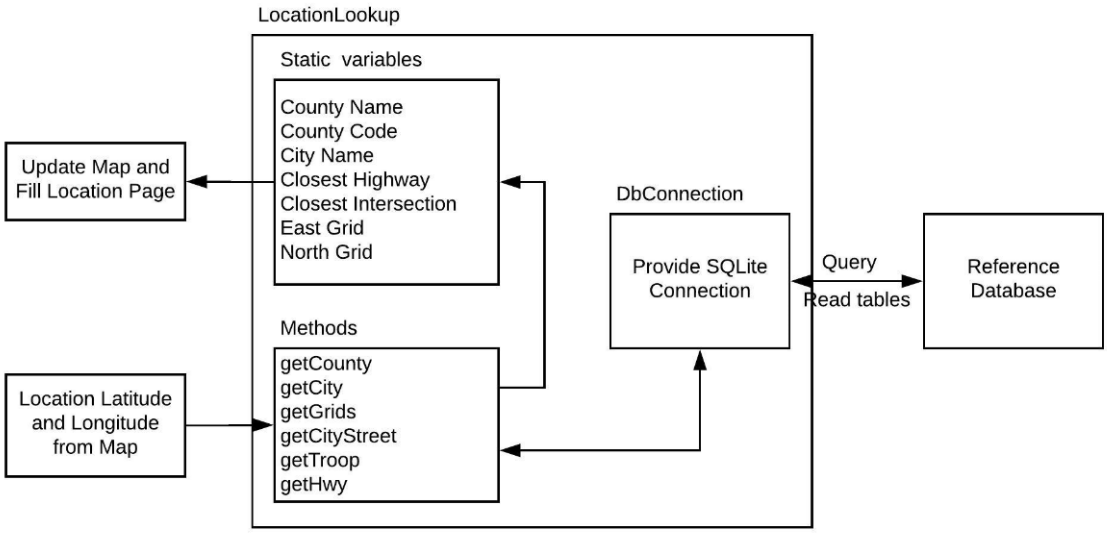

Figure 4.16 LocationLookup Class.

is used to calculate location-related information.

The database connection is done in PARIS software uses Microsoft's `System.Data.SqlServerCe` namespace which is a is a collection of classes that provide access to SQL Server Compact 4.0 databases [59]. The Reference data is given by the Oklahoma DOT Planning and Research Division in SQL Server Compact Edition Database File (SDF) format.

Xamarin does not support using Microsoft `SqlServer` on Xamarin Forms for working with databases. Though, Xamarin supports using SQLite to load and save objects to local databases in code that can be shared between different platforms. SQLite is pre-installed on Android and iOS which makes it good option for cross-platform apps that require local database connections [6].

SQLite is a free SQL database management engine that implement serverless,

zero configuration SQL database to read and write directly to ordinary disk files, which makes it a popular choice for embedded systems [78].

Also, the size of the SQLite library is very small and can run on minimal disk space which makes it a very good candidate for memory constrained devices [6,78]. SQLite can be much faster than direct system I/O and can perform faster when more memory is available [78]. Using SQLite to access Reference database in PARIS+ app required changes to the database base file format and added limitation on database queries that included trigonometric functions as they are not supported in SQLite queries.

Also, SQLite reads db3 database files which is different from the Reference data SDF file. To solve this, the Reference data was converted to the db3 data by exporting all the tables in CSV format. The exported CSV files was then imported to a db3 file. The `DbConnection` class was developed to provide `SQLiteConnection` object to the `LocationLookup` class. The `LocationLookup` class methods uses the `SQLiteConnection` object to pass queries the database. The `SQLiteConnection` uses the database path and can copy the query results into C# tables. These tables stored the data that is later used by `LocationLookup` methods to calculate location information.

In the next section I will discuss the functionality of the `LocationLookup` getCounty method to calculate the incident location county which involved calculating the distance between two points using their latitude and longitude coordinates. The rest of the methods on the `LocationLookup` class used similar code

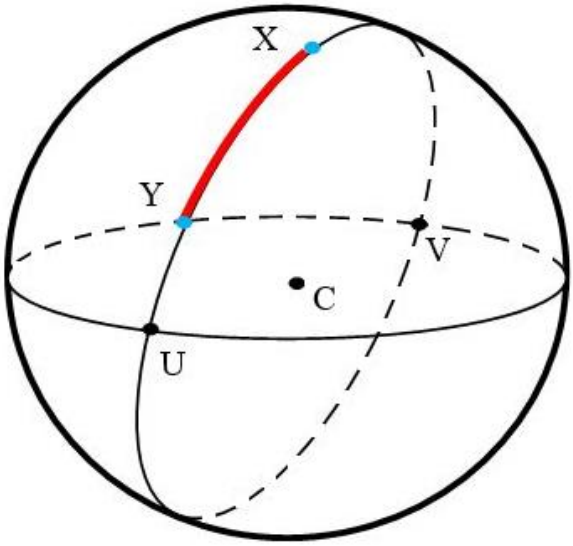

Figure 4.17 Great circle distance.

to calculate the distances between different points. The `LocationLookup` class calculate distances between different locations in the C# code while PARIS location tool use SQL queries to calculate and compare distance calculations. I performed the calculations and comparisons inside the C# code as Microsoft SQL support queries that include trigonometric functions while SQLite doesn't.

4.11 The Great Circle Theorem

The Great Circle theorem defines how to calculate the distance between two points on the surface of a sphere. The great circle distance is given as the shortest measured distance between two points on a spherical surface [37].

In the Euclidean space, the distance between two points is a straight line, while in shapes with curvatures, the distance between two points is given by the curve length [37]. A circle on the surface of the sphere whose center coincide with the center of the sphere "C" is called the "Great Circle" as shown in Figure 4.17 [37].

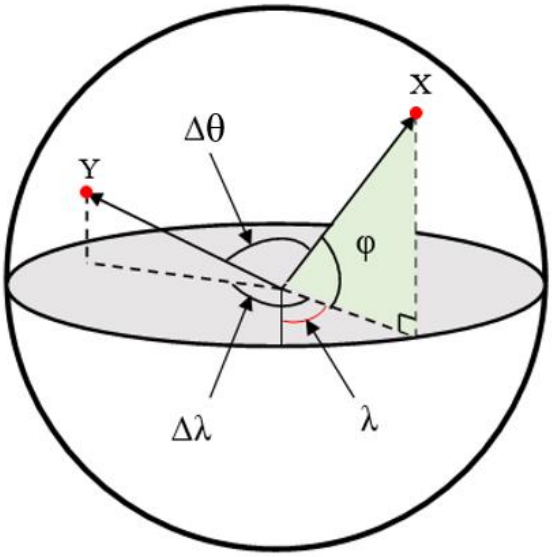

Figure 4.18 Angles and distances between 2 points X and Y on a sphere surface.

A great circle of a sphere is formed from the intersection between a sphere and a plane that passes through the center of the sphere and hence it forms the greatest circle inside this sphere [37].Similarly, any two points on the surface of a sphere that are not directly opposite to each other, as given by points (X and Y) on the sphere on Figure 4.17 forms a unique great circle. These two points will separate the circle into 2 arcs. The shortest distance between the 2 points on the sphere is the same as the length of the shorter arc [37].

The shortest distance d between any two points lying on a great circle of a sphere can be calculated knowing the radius of the circle r and the angle between the two points Δθ by [37]:

$$d = r\triangle\theta \tag{4.1}$$

The angle Δθ between the points is given by [87]:

$$\triangle\theta = \arccos(\sin(\phi_1)\sin(\phi_2) + \cos(\phi_1)\cos(\phi_2)\cos(\triangle\lambda) \tag{4.2}$$

Where ϕ_1 , λ_1 and ϕ_2 , λ_2 are the geographical latitude and longitude of point X and Y as shown in Figure 4.18. The formulas above are accurate for a sphere. As the earth is not exactly a sphere, the calculation remains accurate to a high degree [37].

Errors introduced for assuming a spherical earth is not more than 0.5% for latitude calculations and 0.2% for longitude calculations [37]. To find the county of the incident location the approach that was used is shown in Figure 4.19 and involved the following two steps:

1. Find closest county points:
    - Find a list of county locations from county definitions table within one latitude unit and one longitude unit that have the shortest distance from incident location using formula (4.2).
    - Arrange counties in an ascending order according to the distances calculated between the incident location and any points in the table that are within +/- 1 latitude and longitude unit.
2. Check which county contain the closest point:
    - Iterate over the counties list from step 1 and find the first county where the location coordinates lie inside its polygon. Consider this location county.

The location latitude and longitude degrees are given as two decimal numbers that define the location a point on earth.

The decimal places on these two numbers will define the accuracy of the location on the map. Latitude degrees are approximately 69 miles apart [37]. This means that 0.1 latitude degree is equal to 6.9 miles on the map. This highlights the importance of accurate mathematical calculations.

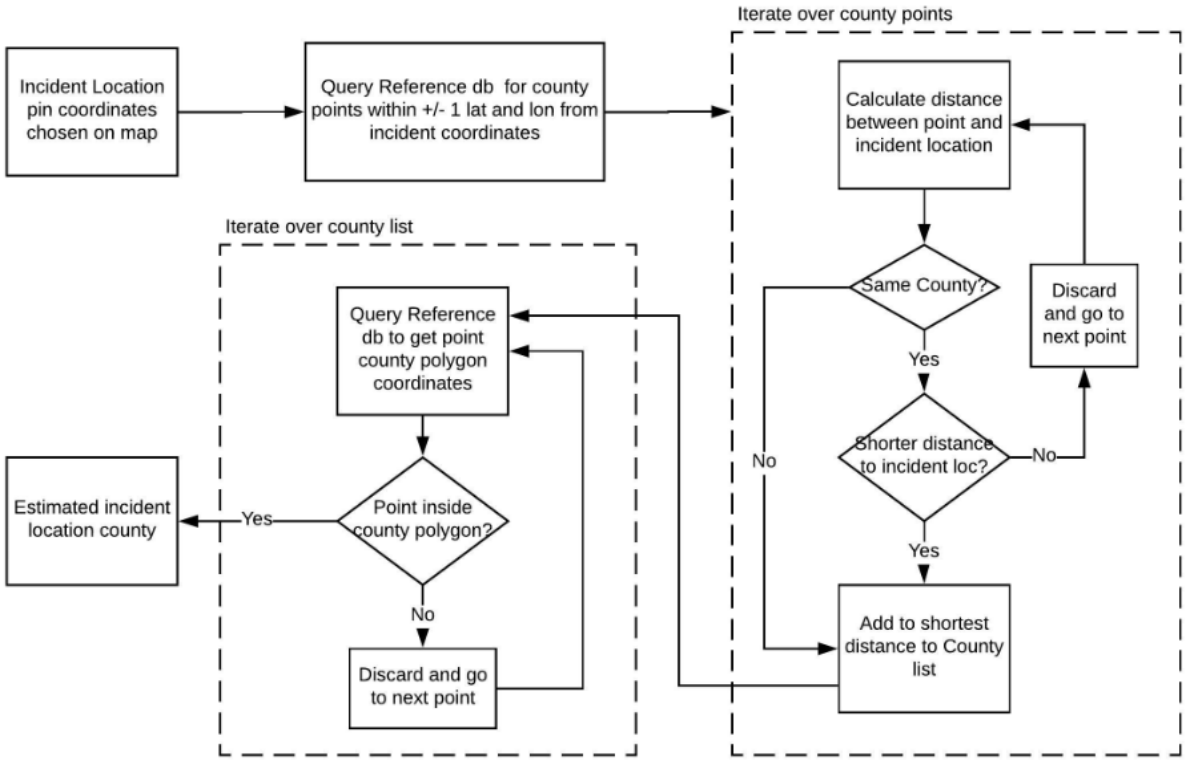

Figure 4.19 getCounty Method.

Although SQLite supports most of the standard SQL language, it doesn't support trigonometric functions in the SQL query. I imported raw latitude and longitude values from tables to the C# code and performed the calculations using the System.Math class.

4.12 Driver License Parser Class

The `DLParser` class parse the driver license information that is read from the barcode on the driver license. The barcode contains information about the card holder including name, date of birth, sex, eye color, height and other information that is required to be filled for citations issued in Oklahoma [70].

The PDF417 (Portable Data File) symbology is 2D stacked bar code. It was invented in 1991 by Symbol Company [94]. Each symbol character on the barcode consist of 4 bars and 4 blanks and the total width is 17 [94]. PDF417 is the machine-readable technology that is used on all the new driver license cards issued in the United States [8]. The user scans the barcode with the phone camera and the result of the scan is decoded by the `DLParser` class. Figure 4.20 shows the sequence used to obtain information from the driver license to fill the Person Page fields.

The `DLParser` class creates a new instance of the `Decoder` class. The scanned barcode consists is a string that should contain a header and a sub file. The header is designed to allow interpreting the encoded data [8]. The sub file carries the encoded information [8]. The `Decoder` class reads the header code to identify sub file type, length, offset and other header information. The sub file contains information about the type of data encoded in the sub file. For Driver licenses, the sub file type is always set to "DL".

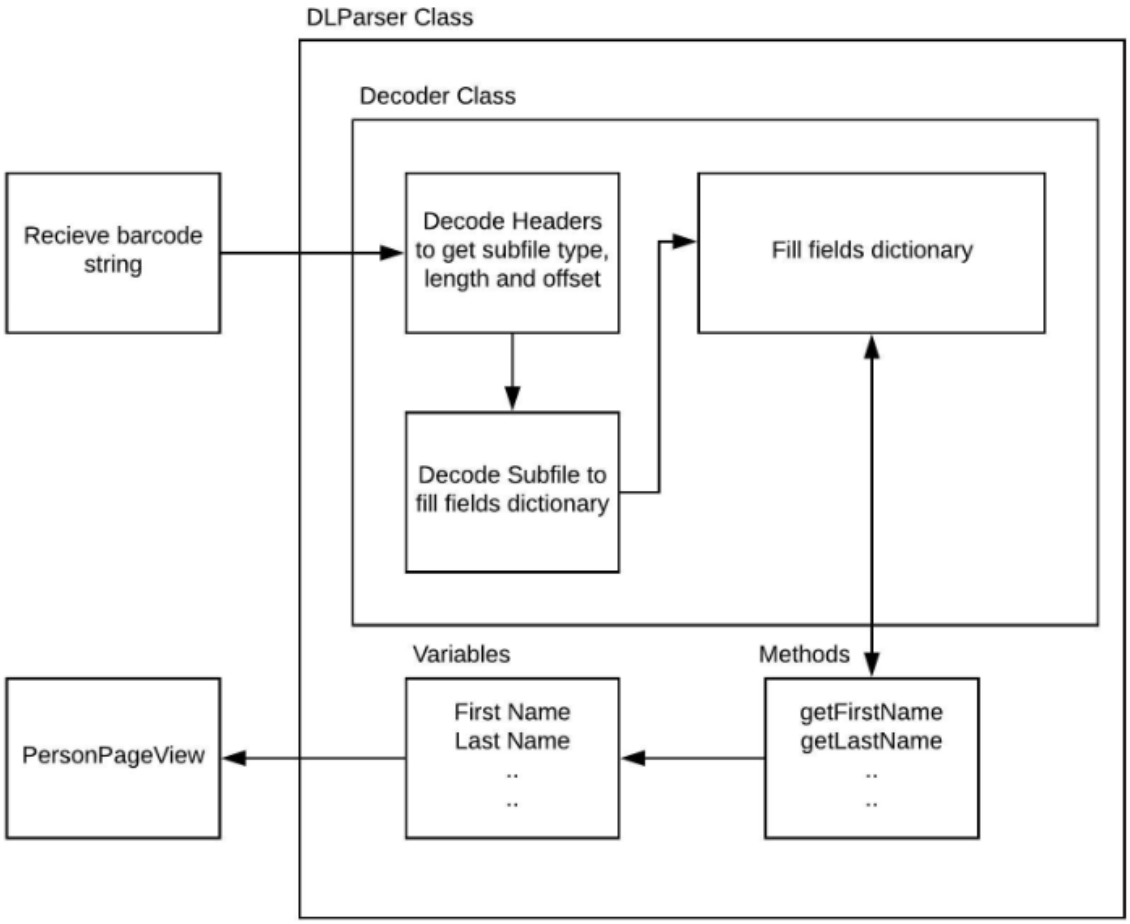

Figure 4.20 DriverLicenseParser Class.

The Offset is a 4-digit numeric value that specifies the number of bytes from the beginning of the file to the start of the sub file [8]. This allows the decoder to identify the header from the sub file. The sub file contains the actual information in separate lines [8].

The AAMVA standard specifies that a 3-letter barcode element identifier must precede the encoded information [8]. A list of the mandatory element identifiers and what they correspond to are given in table 4.1. Every line in the sub file contains one piece of information preceded by the proper element identifier.

Table 4.1 Element identifiers in the Driver License barcode [8].

Element	Data Encoded
DCA	Type of vehicle cardholder has privilege to drive
DCB	Restrictions, if any
DCD	Endorsements, if any
DBA	Driver license expiration date
DCS	Family name of the cardholder
DAC	First name of the cardholder
DAD	Middle name of the cardholder
DBD	Driver license issue date
DBB	Date of birth of the cardholder
DBC	Gender of the cardholder. 1=male, 2=female
DAY	Eye color of the cardholder
DAU	Height of the cardholder
DAG	Street address of the cardholder
DAI	City of the cardholder
DAJ	State of cardholder
DAK	Zip code of the cardholder
DCG	Country of issue

All the information extracted is added to a dictionary object. The dictionary object maps the field names to their values. Other `DLParser` class methods then read the value from the dictionary and update the variables. After the fields are decoded, the `PersonPageViewModel` class properties are updated with the decoded information which notifies the view to update the fields in the `Person Page`.

4.13 VIN Decoder Class

The Vehicle Identification Number (VIN) is used to identify any vehicle operated on the roads throughout the world. The VIN is typically coded in 17 characters to give information about vehicle type, manufacturer, model year, manufacturing plant location, and vehicle serial number [64].

In the United States, NHTSA manages how the VIN numbers are applied according to [64]. The VIN number is coded in a 1-D barcode on different places on each vehicle according to the manufacturer stanadards. Scanning the barcode makes reading the 17-character code much easier. The decoded information can be used to fill some of the fields in the Vehicle Page tab automatically. An example VIN barcode is shown in Figure 4.21.

The `VINDecoder` class decode the 17-character string to identify the VIN number, vehicle manufacturer, and model year. The VIN code is divided into 3 main sections that contain information about the vehicle as shown in Figure 4.22. The first 3 digits define the country and manufacturer of the Vehicle [64]. The next 5 characters contain details about the vehicle like vehicle model type, engine type, and gross vehicle weight rating (GVWR). The encoding for these 5 characters is not standard and is determined by the manufacturer [64]. The 9th character is a check digit and is used to

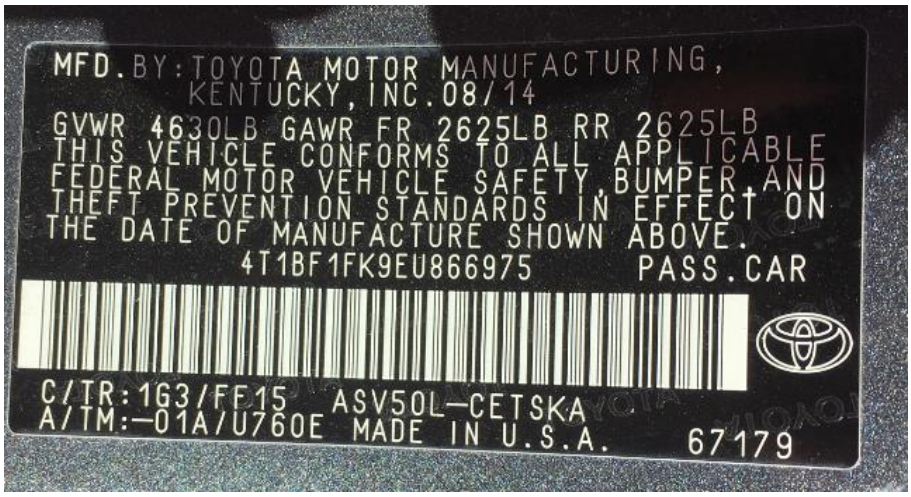

Figure 4.21 VIN number 1-D barcode.

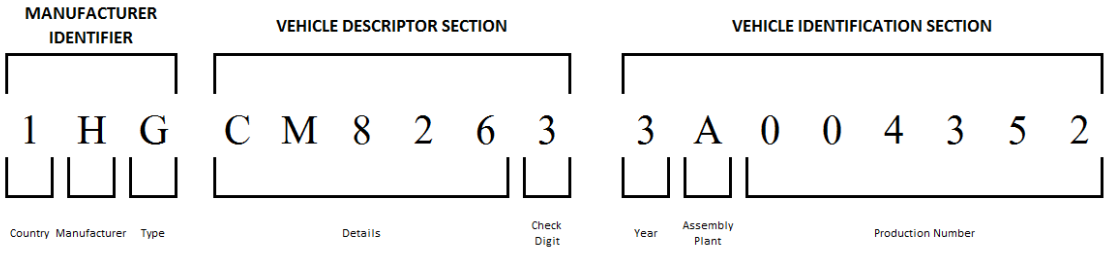

Figure 4.22 VIN number pattern.

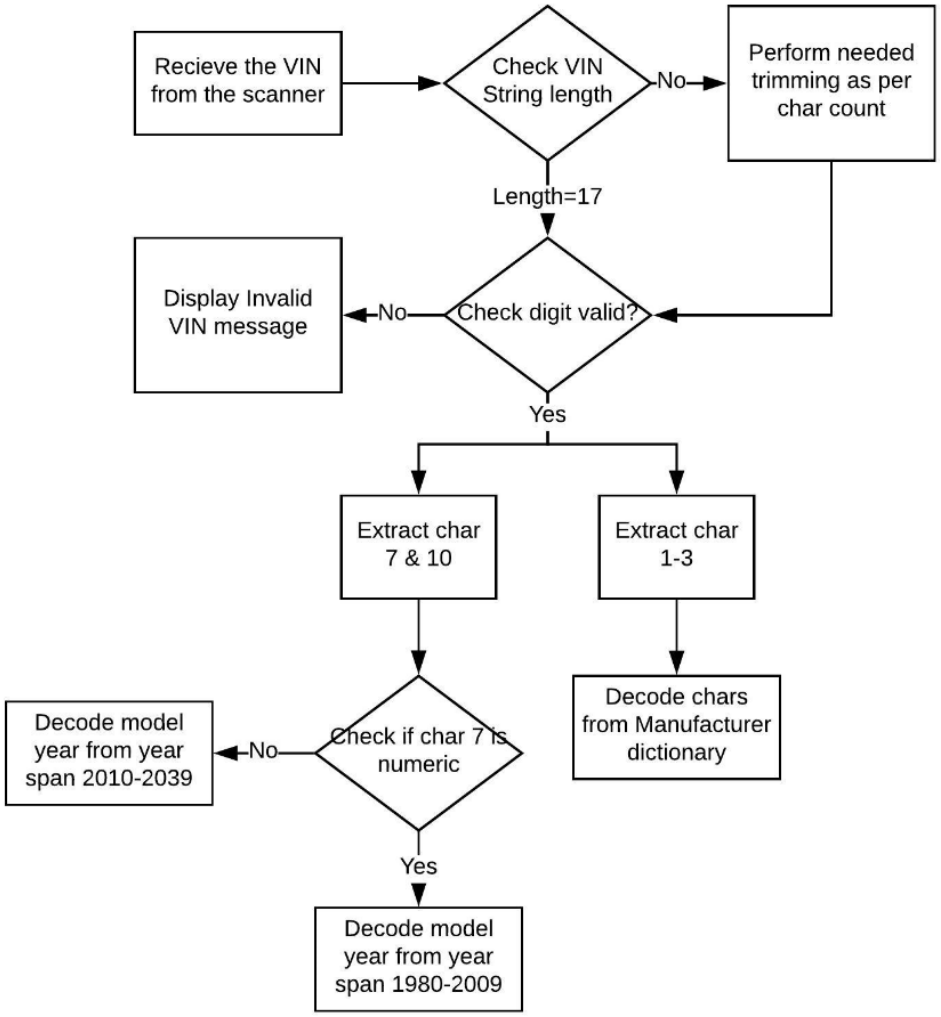

Figure 4.23 VINDecoder Class.

validate that the VIN number is correct. The rest of the characters in the VIN is called the vehicle identification section. This part of the VIN contains information about the model year, the manufacturing plant, and the production number of the Vehicle [64].

The `VINDecoder` sequence is shown in Figure 4.23. The `VINDecoder` class receives the VIN number as a string object from the scanner. The received string should be 17 characters long. Though, it was noticed during debugging the code that imported vehicles might have an extra "i" letter at the beginning of the VIN number.

The `VINDecoder` class checks the length of the VIN and remove any extra characters if needed.

The VIN check digit is then checked to confirm that the VIN number is correct. If the check digit is incorrect the software will notify the view to display a toast message to notify the user of “Incorrect VIN” and to abort the sequence. If the check digit is correct, the first three characters are used to lookup the manufacturer in a dictionary object containing a list of all the manufacturer in the world.

The model year is encoded by alphanumeric characters at VIN position 10 to define a span of 30 years starting from 1980 to 2009. The same alphanumeric characters are used to define another span of model years from 2010 to 2039 [66]. Position 7 will define which span of year should be used to decode the model year. If position 7 is numeric, the model year in position 10 refers to year range 1980-2009 [66]. If position 7 is alphabetic, position 10 refers to a year range 2010-2039 [66]. After the Manufacturer and the model year are decoded, the `VehiclePageViewModel` is updated. The `VehiclePageViewModel` notifies the View to update the fields in the `Vehicle Page` tab.

Chapter 5

Testing the PARIS+ app

The last part of building PARIS+ app is testing the code functionality, setting the app limitations and user expectations. Testing and debugging was done throughout the development progress to test the implemented features. It was also done at the very end with Captain Eric Cannaday with the Oklahoma Highway Patrol. In this Chapter, I will discuss details about the testing that was done, how to install the app and the app specifications.

Software testing is the process of examining if the software solution satisfies the requirements that it was designed for [79]. Developers should always consider that the application does have defects and bugs and not all these bugs are already discovered [79]. Software testing is the process that allow developers to find and fix those problems [79]. A software bug is an error that causes a program to malfunction or to generate incorrect results due to incorrect logic or code [79].

An important aspect of mobile app testing is the type of device used for testing [19]. Apps can be tested on emulators, simulators, or real devices. Mobile device emulators are desktop applications that can provide a similar response as a real device by translating instructions of the compiled app source code. Emulators allow developers and testers to debug apps without having a real device on hand [19]. Since the simulated app runs on a desktop computer, not all hardware elements as touch gestures or internal sensors will be available. Though, testing using an emulator still provide fast feedback specially in the early stages of development [19].

Emulators don’t help measuring the performance characteristics of the code, though it saves money on buying a bunch of devices to test things like different screen resolutions of different devices. Simulators are less complex applications that can simulate a small subset of the real device behavior [19]. Similar to emulators, simulator don’t test the internal hardware elements, but they are faster. They also provide fast feedback in the early stages of the development [19].

To test the implemented functionalities in PARIS+ app, I alternated between an Android Emulator on Visual Studio 2015 and Visual Studio 2017 and a real device. I used Samsung Galaxy J7 smartphone running Android OS version 6.0.1 (API level 23) to test the app during the development process. I also performed full functional test on a Samsung Galaxy Note 4 smartphone. Table 5.1 shows the results of testing the PARIS+ app along with the APIs that the app was tested on.

I tested the app on the most common android API levels as of February 2018. Figure 5.1 shows the most common API levels that Android OS devices run as per [31] and depicts which APIs where tested with the app. While testing the app on Android OS version 7.0, it was noticed that the searchbar view was not showing up in the display.

The searchbar view is used in `ViolationsSearchPage` and `WarningsSearchPage` for the user to enter violation and warning text to be searched. This finding also conforms with [40]. A workaround to fix this issue suggested by [40] is by setting the HeightRequest property of the searchbar view to a constant value. Setting the HeightRequest property to a constant value was found fix the issue on API 7.0 without compromising the operation of the app on the other APIs that were tested.

Table 5.1 Android OS versions tested with PARIS+ app.

Android OS Version	API Level	Device Model	Device Type	Comments
5.1	22	Samsung Note 4	Real Device	Functional
6.0.1	23	Samsung Galaxy J7	Real Device	Functional
7.0	24	Nexus 7	Android Emulator	Functional*
7.1.1	25	Nexus 5	Android Emulator	Functional

*Need to set search bar height request to a constant value. This is a bug in API 24 as per [40].

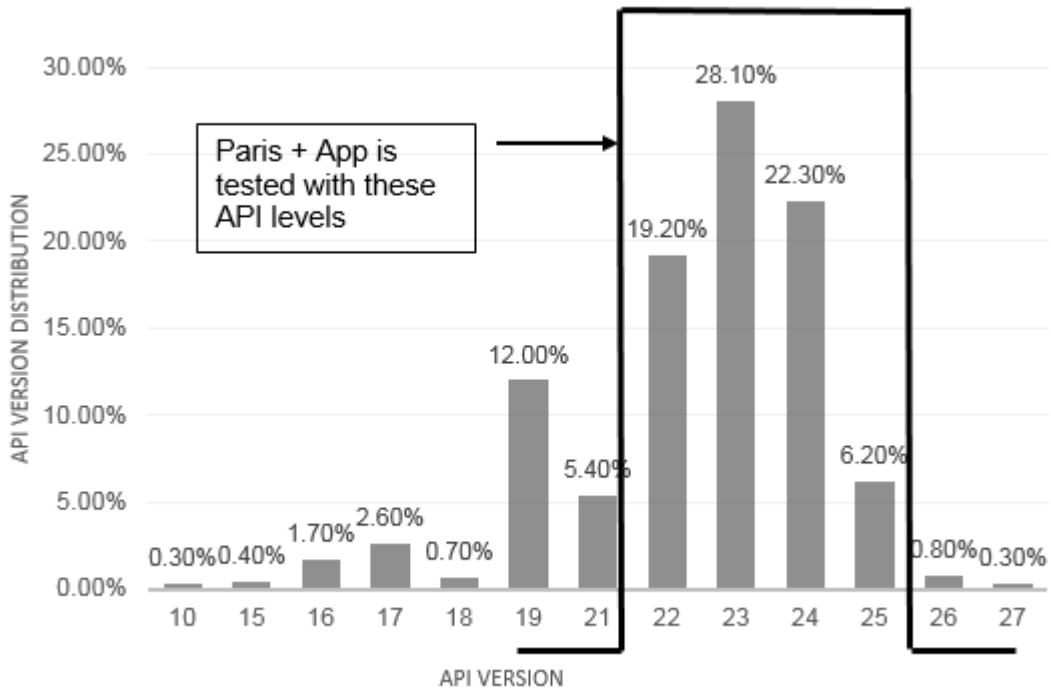

Figure 5.1 Android OS usage distribution [31] and the PARIS+ app testing.

*The given data in Figure 5.1 for API usage distribution was collected during a 7-day period ending on Feb 5th, 2018 [31].

**Any API level with less than 0.1% distribution is not shown in Figure 5.1 [31].

The tests that were done on the PARIS+ app were all manual tests and were limited to checking the functionality of all the buttons and that the logic behind them is functioning as expected. The PARIS+ app was also manually tested in the wild. Testing in the wild means to test the app under common real-world scenarios that the app will be used in [31]. This was done outdoors on a real network connection on the side of a highway to confirm that the app will continue to function as expected. Also, the parts of the app that require using the camera to scan a driver license or VIN barcode were tested under different lighting conditions to verify that the barcode scanner can swiftly detect 1D and 2D barcodes in a real deployment environment.

Testing results shows that in ideal conditions, it can take the user less than a minute to use the PARIS+ app to issue and print a citation to a violator.

5.1 Testing the Driver License Parser Class

The driver license parser class is used every time the user attempts to scan the barcode. The `DLParser` class successfully decoded more than 15 different driver licenses from the following states:

- Oklahoma
- Texas

More testing needs to be done to ensure that the software can decode the driving license from other states.

5.2 Testing the VIN Decoder Class

The `VINdecoder` is used when the user attempts to scan a vehicle VIN code. It was tested on multiple vehicles models and returned correct result for the VIN number, vehicle manufacturer, and vehicle model year.

The `VINDecoder` was tested on more than 10 vehicles of the following vehicle manufacturers:

- Toyota
- Honda
- Nissan
- Chevrolet
- Ford

The VIN number contains more information that can be used towards filling citation information than what's implemented in the PARIS+ app. The Vehicle description section (position 4 to position 8 in VIN code) can be decoded to retrieve the model and body type [64]. This part of the VIN is not decoded in a standard way where every manufacturer decides his own letter assignment [64].

5.3 Testing the Location Lookup Class

I tested the `LocationLookup` class with more than 50 random points on the map and compared the results to PARIS Location Tool. Compared values for county, city (if inside city), troop, closest highway and closest city street were matching what PARIS location tool returned.

Although location lookup class can give correct result, results accuracy will depend on the accuracy of the tables given and the `Location Lookup` class might give inaccurate results. The user should always confirm all the calculated values to make sure that they conform with location chosen on `Map Page`.

The `Location Lookup` class doesn't require online access to execute. Though, the map view requires online connection. It wouldn't be possible for the user to

properly select the location on a map without being able to load the map view. This part makes the location data provided by PARIS+ app an online tool that requires internet connection to be usable.

5.4 Installing the app

The PARIS+ app version 1.0 is only available through CITS lab as .apk file. It requires at least 150MB of free memory to be installed. Its recommend running the app on the API levels that were tested.

Chapter 6

Conclusion and Future Work

Collecting timely and accurate data for traffic citations and accident reports are important goals for law enforcement. Manually collected data was proven to have content errors, took longer time to complete, involved longer delays to reflect in the records management system, and needed multiple layers of data entry from different parties within the police agency [83].

The electronic citation system was first used in 2000 where officers were able to fill citation data on computers, print legible paper tickets, and send the records to courts via wireless network [83]. Since then, the electronic citation system has gained interest among law enforcement agencies throughout the nation as it was proven to eliminate a lot of the inefficiencies of the paper and pen method [51,83,86].

Laptops and external peripheral devices attached to them cost thousands of dollars per unit [24]. In addition, maintaining a running fleet needs extra time away from the focus of the core duty of law enforcement [24,85]. In Oklahoma, some police agencies have deployed the electronic citation systems. Though, an estimate of 80% of the agencies are still using the paper and pen method due to financial reasons.

In the presented work, I contributed to developing a low-cost smartphone-based electronic citation system to be used by the police agencies in Oklahoma that can’t afford buying the computer-based electronic citation system. PARIS+ app is a

smartphone-based electronic citation software that can be used on some Android phones that are commercially available in the market. The app utilizes the internal peripheral devices that are already built inside most of the nowadays smartphones like the camera, the GPS, and the Bluetooth. The PARIS+ app helps users to issue citations accurately and efficiently by importing data from driver licenses and VIN barcodes, populating location information on a button tap, and printing legible paper tickets wirelessly. It can also send the records to headquarters via wireless networks.

The average cost on a computer-based electronic citation system is around $5000 dollars. The proposed system is estimated to have a total cost per unit less than $300 as per U.S. market prices on Jan 27th, 2018. This cost is significantly lower than other computer-based electronic citation systems that are available for law enforcement agencies. For agencies that already issued smartphones to their officers, the cost of the system will be only the cost of the Bluetooth printer, which is $59 as per the market prices on January 26th, 2018 [72].

Nevertheless, operating and maintaining smartphones is easier than maintaining computers and external peripheral devices that are used in the typical electronic citation system. Table 6.1 lists a comparison between the advantages and disadvantages of using the paper and pen method, the computer-based electronic citation system (specifically digiTicket), and the PARIS+ app electronic citation system. This comparison shows how the PARIS+ app can perform most of the computer-based electronic citation system functions at a considerably low cost. Adding extra features like querying the license plate and the license plate history can add great benefit to PARIS+ app also.

Table 6.1 Comparison between different types of citations methods for law enforcement

Criteria	Paper and Pen method	digiTicket electronic citation system	PARIS+ app electronic citation system
Average price of the original setup	Zero	$4,765[24]	Less than $300
Average annual maintenance fees	Zero	$209.34[24]	Estimated to be less than $100
Data reporting	Manual data entry for crash and citation reporting	Crash and Citation reporting	Citation Reporting
Time taken to complete a traffic citation	10-15 minutes	Estimated to be few minutes	Can be less than 1 minute
Needs manual processing at the agency side	Needs	Doesn't need	Doesn't need
Record errors	35-40% of citations are lost due to errors [51]	Almost 0% chance of errors due to software validation	Almost 0% chance of errors due to software validation
Accurate and timely records	Inaccurate	Accurate and timely	Accurate and timely
Fulfill NHTSA requirements	Doesn't fulfill	Fulfill	Fulfill
Officer Safety	Longer dangerous roadside time	Less roadside time	Less roadside time
Extensive data entry	Very inconvenient	Very Convenient	Considerably convenient
License plate and driver license query	No available	Available	Currently not available, but can be added

In an ideal environment, it can take the user less than a minute to issue a citation using the PARIS+ app. Using the app in police work is expected to save officer time and improve timeliness and correctness of the citation records. The PARIS+ app requires internet to load maps and send data to headquarters. Future work includes making the PARIS+ app completely available offline so that users can issue citations and print legible tickets while the app schedule sending offline-generated records to

courts when internet connection becomes available. Also implementing electronic crash forms in the PARIS+ app is part of the future work to make PARIS+ a comprehensive solution for police agencies.

References

[1] A. Burns, C. Dunn, B. Umbaugh, Introduction to mobile development, Microsoft, March 2017, Retrieved from https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/introduction-to-mobile-development#Introduction_to_Xamarin, Access date 4-2-2018.

[2] A. Burns and C. Dunn, Building html views using razor templates, Microsoft, Feb 2018, Retrieved from https://developer.xamarin.com/guides/cross-platform/advanced/razor_html_templates/, Access date 3-28-2018.

[3] A. Ortlieb and R. Kanable, That's the ticket!, March 2011, Retrieved from https://www.officer.com/command-hq/technology/traffic/article/10171458/thats-the-ticket, Access date: 3-16-2018.

[4] A. Vollmer, Pioneer in Police Professionalism; Oral History Transcript, Forgotten Books, London, United Kingdom, June 2012.

[5] A.C. Germann, F.D. Day, R.J. Gallati, Introduction to Law Enforcement and Criminal Justice, Springfield, IL, Charles C Thomas, 1988.

[6] A.D. Sole, Xamarin – working with local databases in xamarin.forms using sqlite, Microsoft, July 2016, Retrieved from https://msdn.microsoft.com/en-us/magazine/mt736454.aspx, Access date 4-2-2018.

[7] AAMVA IT Committee, Bar Code Data Encoding Requirements – AAMVA International Specification – Motor Vehicle Documents, AAMVA, Arlington, Virginia, May 2011.

[8] American Association of Motor Vehicle Administrators, 2016 AAMVA DL/ID Card Design Standard, AAMVA, Arlington, Virginia, Sep 2016.

[9] Apple Inc., Swift 4, Retrieved from https://developer.apple.com/swift/, Access date 4-3-2018.

[10] Apple Inc., Tools you'll love to use, Retrieved from https://developer.apple.com/xcode/ide/, Access date 4-3-2018.

[11] Aspnet, AspNet.Razor, Feb 2018, Microsoft, Retrieved from https://www.nuget.org/packages/Microsoft.AspNet.Razor/, Access date 3-23-2018.

[12] B.A. Reaves, Census of state and local law enforcement agencies, 2008, Report NCJ 233982, Bureau of Justice Statistics, Washington, DC, July 2011.

[13] C. Giancursio, The history of traffic citations, L-Tron Corporation, Victor, NY, Retrieved from https://www.l-tron.com/resource-page/traffic-citations-then-and-now/, Access date: 2-4-2018.

[14] C. Petzold, C. Dunn, B. Umbaugh, Part 5. From data bindings to mvvm, Microsoft, Oct 2017, Retrieved from https://docs.microsoft.com/en-us/xamarin/xamarin-forms/xaml/xaml-basics/data-bindings-to-mvvm, Access date 3-1-2018.

[15] C. Petzold, D. Britch, C. Dunn, Xamarin.forms xaml basics, Microsoft, Oct 2017, Retrieved from https://docs.microsoft.com/en-us/xamarin/xamarin-forms/xaml/xaml-basics, Access date 4-2-2018.

[16] Center of Disease Control and Prevention, Vital signs: Nonfatal, motor vehicle–occupant injuries (2009) and seat belt use (2008) among adults -- united states, Morbidity and Mortality Weekly Report, Vol. 59, pp. 1681-1686, U.S. Department of Health & Human Services, Atlanta, GA, Jan 2011.

[17] Criminal Justice Information Advisory Board, 2009 annual report of interoperability, State of Maryland CJIAB, Pikesville, Maryland, Nov 2010.

[18] D. Britch, C. Dunn, C. Petzold, Tabbed Page, Microsoft, July 2017, Retrieved from https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/navigation/tabbed-page, Access date 4-2-2018.

[19] D. Knott, Hands-On Mobile App Testing: A Guide for Mobile Testers And Anyone Involved In The Mobile App Business, Pearson Education, Crawfordsville, Indiana, May 2015.

[20] D. Weisburd, S.D. Mastrofski, A.M. Mcnally, R. Greenspan, J.J. Willis, Reforming to preserve: Compstat and strategic problem solving in american policing, Wiley Online Library, pp. 421-456, Indianapolis, IN, July 2003.

[21] D. Weisburd, C.W. Telep, J.C. Hinkle, J.E. Eck, The effects of problem-oriented policing on crime and disorder, Document 224990, National Crime Justice Reference Service, Rockville, MD, Nov 2008.

[22] D.W. Allen and Y. Barzel, "The Evolution of Criminal Law and Police during the Pre-modern Era," *The Journal of Law, Economics, and Organization*, Volume 27, Issue 3, pp. 540–567, 2011.

[23] E. Jendrock et al., The java ee 6 tutorial: Web services, Oracle and/or its affiliates, Jan 2013, Retrieved from https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html, Access date: 3-29-2018.

[24] Economic Impact Group, Case Study: A cost-benefit analysis of the digiticket implementation for the sand springs police department, EIG, Edmond, OK, May 2011, Retrieved from http://www.saltustechnologies.com/wp content/uploads/docs/digiTICKET%20SSPD%20Case%20Study.pdf. Access date: 2-15-2018.

[25] E. Cannaday, Oklahoma Highway Patrol, Personal Communication, Dec 2017.

[26] Federal Bureau of Investigation, Uniform crime report: crime in the united states, 2012, U.S. Department of Justice, fall 2013, Retrieved from https://ucr.fbi.gov/crime-in-the-u.s/2012/crime-in-the-u.s.-2012/violent-crime/violentcrimemain.pdf, Access date 2-25-2018.

[27] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Second Edition, Addison Wesley, Boston, Massachusetts, 2005.

[28] G.L. Kelling and W. Bratton, "Declining Crime Rates: Insiders' Views of the New York City Story," *Journal of Criminal Law and Criminology*, Volume 88, Article 3, 1998.

[29] Google LLC and Open Handset Alliance, Application fundamentals, Retrieved from https://developer.android.com/guide/components/fundamentals.html, Access date 2-22-2018.

[30] Google LLC and Open Handset Alliance, Codenames, Tags, and build numbers, Retrieved from https://source.android.com/setup/build-numbers, Access date 2-22-2018.

[31] Google LLC and Open Handset Alliance, Dashboards, Retrieved from https://developer.android.com/about/dashboards/index.html#Screens, Access date 4-7-2018.

[32] Google LLC and Open Handset Alliance, Dialogs, Retrieved from https://developer.android.com/guide/topics/ui/dialogs.html, Access date 4-2-2018.

[33] Google LLC and Open Handset Alliance, Everything you need to build android, Retrieved from https://developer.android.com/studio/features.html, Access date 2-22-2018.

[34] Google LLC and Open Handset Alliance, The android source code, Nov 2017, Retrieved from https://source.android.com/setup/, Access date 3-15-2018.

[35] Google LLC and Open Handset Alliance, Toasts, Retrieved from https://developer.android.com/guide/topics/ui/notifiers/toasts.html, Access date 4-2-2018.

[36] Google LLC and Open Handset Alliance, What is api level?, Retrieved from https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels Access date 2-22-2-18.

[37] Great Britain Ministry of Defense, Admiralty Manual of Navigation, Volume 1, HMSO, London, UK, 1964.

[38] International Association of Chiefs of Police Highway Safety Committee, Traffic safety Strategies for Law Enforcement, IACP,Alexandria, VA, Aug 2003.

[39] J. Gold, 11 tech breakthroughs that led to today's smartphones, Sep 2017, Retrieved from https://www.computerworld.com/article/3222855/mobile-wireless/11-tech-breakthroughs-that-led-to-todays-smartphones.html, C to-todays-smartphones.html, Access date 3-7-2018.

[40] J. Weber, Is there support for SearchBar on nougat 7.0? , Xamarin Forums, Retrieved from https://forums.xamarin.com/discussion/79446/is-there-support-for-searchbar-on-nougat-7-0/, Access date 4-8-2018.

[41] J.E. Eck, Assessing Responses to Problems: An Introductory Guide for Police Problem-Solvers, Center for Problem Oriented Policing, U.S. Department of Justice, Washington, DC, Aug 2011.

[42] J.H. Burch and M.N. Geraci, Data-driven approaches to crime and traffic safety (DDACTS), The Prosecutor Magazine, National District Attorneys Association, Arlington, VA, April/May/June 2009.

[43] J.P. Havlicek, CITS Lab, University of Oklahoma, Personal Communication, Aug 2016.

[44] J.P. Havlicek and R.D. Barnes, Gps location data enhancement in electronic traffic records (Report OTCREOS11.1-27-F), Oklahoma Transportation Center, Midwest City, Oklahoma, Jan 2013.

[45] J.W. Beck, The effects of directed patrols in unincorporated areas: A case study of the data-driven approach to crime and traffic safety, UMI number: 1476776, ProQuest LLC, Ann Arbor, MI, May 2010.

[46] K.D. Kochanek, S.L. Murphy, J. Xu, B. Tejada-Vera, Deaths: Final Data for 2014, National Vital Statistics Reports, Volume 65, Number 4, Center of Disease Control and Prevention, June 2016, Retrieved from https://www.cdc.gov/nchs/data/nvsr/nvsr65/nvsr65_04.pdf, Access date 1-22-2018.

[47] K. Lackey, Electronic citations speed up ticketing process for police, ABC News, June 2008, Retrieved from https://usatoday30.usatoday.com/tech/news/2008-06-25-Electronic-tickets_N.htm, Access date: 2-23-2018.

[48] KFOR-TV and K.QUERRY, Oklahoma city police submit $1.7 million budget cut as city funding falls, Oklahoma's News 4, 2017, Retrieved from http://kfor.com/2017/05/23/oklahoma-city-police-submit-1-7-million-budget-cut-as-city-funding-falls/, Access date 2-28-2018.

[49] L.A. Steverson, Policing in America, ABC-CLIO Inc, Santa Barbra, CA, 2008.

[50] M. Doyle, Cell phones on duty, March 2014, Retrieved from http://www.policemag.com/channel/technology/articles/2014/03/cell-phones-on-duty.aspx, Access date 3-23-2018.

[51] M. Mann, How ecitation systems improve officer safety, reduce costs, PoliceOne, April 2017, Retrieved from https://www.policeone.com/police-products/traffic-enforcement/Electronic-Citations/articles/327666006-How-eCitation-systems-improve-officer-safety-reduce-costs/, Access date 2-23-2018.

[52] M. Spicuzza, Mayor tom barrett: Milwaukee could lose 84 police officers due to budget shortfall, Journal Sentinel, June 2017, Retrieved from https://www.jsonline.com/story/news/local/milwaukee/2017/06/11/mayor-tom-barrett-milwaukee-could-lose-84-police-officers-due-budget-shortfall/385354001/. Access date 3-2-2018.

[53] M.S. Nuth, "Taking advantage of new technologies: For and against crime," *Elsevier Computer Law and Security Review*, Volume 24, Issue 5, Pages 437-446, 2008.

[54] Microsoft Developer Network, The mvvm pattern, Microsoft, Retrieved from https://msdn.microsoft.com/en-us/library/hh848246.aspx, Access date 4-2-2018.

[55] Microsoft, Chapter 1: What is software architecture?, Retrieved from https://msdn.microsoft.com/en-us/library/ee658098.aspx, Access date 3-24-2018.

[56] Microsoft, Chapter 24: Designing mobile applications, Retrieved from https://msdn.microsoft.com/en-us/library/ee658108.aspx, Access date 2-24-2018.

[57] Microsoft, Chapter 5: Layered Application Guidelines, Retrieved from https://msdn.microsoft.com/en-us/library/ee658109.aspx, Access date 2-26-2018.

[58] Microsoft, Introducing visual studio's emulator for android, Nov 2014, Retrieved from https://blogs.msdn.microsoft.com/devops/2014/11/12/introducing-visual-studios-emulator-for-android/, Access date 4-11-2018.

[59] Microsoft, System.data.sqlserverce namespace, Retrieved from https://msdn.microsoft.com/enus/library/system.data.sqlserverce(v=vs.100).aspx, Access date 4-1-2018.

[60] Microsoft, Uml use case diagrams: Guidelines, Retrieved from https://msdn.microsoft.com/en-us/library/dd409432.aspx, Access date 2-28-2018.

[61] National Center for Statistics and Analysis, 2016 fatal motor vehicle crashes: Overview, (Report DOT HS 812 456), NHTSA, Washington, DC, Oct 2017.

[62] National Highway Traffic Safety Administration, 2016 quick facts, NHTSA, Washington, DC, Oct 2017.

[63] National Highway Traffic Safety Administration, Data driven approaches to crime and traffic safety (ddacts): Operational guide, Report No. DOT HS 811 185, NHTSA, Washington, DC, March 2014.

[64] National Highway Traffic Safety Administration, Product information catalog and vehicle, NHTSA, Washington, DC, Retrieved from https://vpic.nhtsa.dot.gov/, Access date 2-21-2018.

[65] National Highway Traffic Safety Administration, Traffic safety facts 2012: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system, Report DOT HS 812 032, NHTSA, Washington, DC, 2014.

[66] National Highway Traffic Safety Administration, Vehicle Identification Number Requirements, 49 CFR Part 565, Docket no. NHTSA 2008-0022, NHTSA, Washington, DC, 2008.

[67] O.B. Waxman, How the U.S. got its police force, May 2017, Retrieved from http://time.com/4779112/police-history-origins/, Access date 2-12-2018.

[68] Ohio History Central, World's first automobile accident, Retrieved from http://www.ohiohistorycentral.org/w/World's_First_Automobile_Accident?rec=2596, Access date 2-21-2018.

[69] Ohio History Central, World's first speeding ticket, Ohio History Connection, Retrieved from http://www.ohiohistorycentral.org/w/World%27s_First_Speeding_Ticket, Access date 2-21-2018.

[70] Oklahoma State Courts Network, Record of Traffic Cases - Report of Convictions to Department, Title 47 O.S.§.18-101, OSCN, May 2012.

[71] P. Dinham, "Pictured: First car to ever get a speeding ticket in 1896 (even if it was doing just 8mph), Dailymail, Feb 2018, Retrieved from

http://www.dailymail.co.uk/news/article-4498456/Pictured-car-speeding-ticket.html, Access date 3-16-2018.

[72] Qunsuotech, QS5801 Specification, Retrieved from http://www.qsprinter.com/product_detail82.html#jquery, Access date 3-12-2018.

[73] R.V. Clarke and J.E. Eck, Crime analysis for problem solvers in 60 small steps, Center for Problem Oriented Policing, U.S. Department of Justice, Washington, DC, 2016.

[74] S. Ressler, How local governments are using technology to serve citizens better, Harvard Business Review, Jan 2016, Retrieved from https://hbr.org/2016/01/how-local-governments-are-using-technology-to-serve-citizens-better, Access date 3-12-2018.

[75] S. Sharwood, Developing world hits 98.7 per cent mobile phone adoption, Aug 2017, Retrieved from https://www.theregister.co.uk/2017/08/03/itu_facts_and_figures_2017, Access date 3-19-2018.

[76] Saltus Technologies, Digiticket benefits, Saltus Technologies, Tulsa, OK, Retrieved from http://www.saltustechnologies.com/digiticket., Access date 22-2-2018.

[77] Sammobile, Galaxy j7 sm-j700h specifications, Retrieved from https://www.sammobile.com/devices/galaxy-j7/specs/SM-J700H/, Access date 3-12-2018.

[78] SQLite Consortium, About sqlite, Retrieved from https://www.sqlite.org/about.html, Access date 2-25-2018.

[79] T. Dawkins and E. Downing, Software testing fundamentals, Microsoft, March 2014, Retrieved from https://mva.microsoft.com/en-us/training-courses/software-testing-fundamentals-8305?l=Phj8iiYy_8504984382/, Access date 4-8-2018.

[80] T.M. Vinod Kumar, E-Governance for Smart Cities, Springer, New York, New York, 2015.

[81] The National Law Enforcement Officer Memorial Fund, Preliminary 2016 law enforcement officer fatalities report, NLEOMF, Washington, DC, Dec 2016.

[82] The National Law Enforcement Officer Memorial Fund, Preliminary 2017 law enforcement officer fatalities report, NLEOMF, Washington, DC, Dec 2017.

[83] The Office of Justice Programs and Department of Transportation, The use of electronic citations: A nationwide assessment, Bureau of Justice Assistance, Washington, DC, June 2003.

[84] Tokenworks Inc, Id scanner coverage chart, March 2014, Retrieved from https://www.idscanner.com/knowledge-base/coverage-chart/, Access date 3-23-2018.

[85] University of Oklahoma ITS Lab, Final report: Oklahoma tracs pilot project, University of Oklahoma ITS Lab, Feb 2008.

[86] V.T. Montez, Electronic Citation System and the Advantages to Small Police Departments, LEMIT, Huntsville, Texas, Feb 2013.

[87] W. Gellert, S. Gottwald, M. Hellwich, H. Kästner, and H. Küstner, The VNR Concise Encyclopedia of Mathematics, Second Edition, Chapter 12, Springer, New York, New York, 1989.

[88] W.F. Walsh and G.F. Vito, "The meaning of Compstat," *Journal of Contemporary Criminal Justice*, Vol. 20 No. 1, Feb 2004.

[89] Xamarin Inc., Android.os.asynctask class, Retrieved from https://developer.xamarin.com/api/type/Android.OS.AsyncTask/, Access date 4-1-2018.

[90] Xamarin Inc., Platform, Retrieved from https://www.xamarin.com/platform, Access date 2-22-2018.

[91] Xamarin Inc., Xamarin.Forms.ActivityIndicator, Retrieved from https://developer.xamarin.com/api/type/Xamarin.Forms.ActivityIndicator/, Access date 4-2-2018.

[92] Xamarin Inc., Zxing.net.mobile for forms, July 2016, Retrieved from https://components.xamarin.com/gettingstarted/zxing.net.mobile.forms, Access date 3-30-2018.

[93] Xamarin Inc., Xamarin.Forms, Microsoft, Retrieved from https://developer.xamarin.com/guides/xamarin-forms/, Access date 4-2-2018.

[94] Z. Hou, Measuring Technology and Mechatronics Automation in Electrical Engineering, Springer, New York, New York, 2012.