**CHAPTER I**

**INTRODUCTION**

Any system that can provide intelligent vehicle location and navigation information will let us avoid congested freeways and find more efficient routes to our destinations, saving millions of euros in gasoline and tons of air pollution. Travel aboard ships and aircrafts will be safer in all weather conditions. Businesses with large numbers of outside plants (e.g., railroads and utilities) will be able to manage their resources more efficiently, reducing consumer costs. A Global Positioning System (GPS) provides the answer to facilitate all these issues.

* 1. **Project Overview**

Vehicle Navigation system with Police Base Station is Client Server Application Designed and Programmed to track police vehicles and managing event alerts. IOs Application is client application that is directly connected to server of base station to send location data and live Video Streaming based on server notification. Base Station Server is responsible to manage History of Vehicle Locations and Events Data and Video along with Location inside Video. IOS Client Application is operated by server and officer inside vehicle has no rights to use or operate the application for security purpose.

* 1. **Security Needs of Project**

During the development of this system all security and others points have been kept in mind. Only the valid person i.e. the admin of police base station who is registered as an authenticated users can see the info about the Vehicles. The common users have no access to the system.

* 1. **Objective**

Though I have developed this system during a learning process but I have done my best to make this system a professional one. The objectives of this project are to Capture vehicle location and sync location with live stream from client to server.

Following are the important objectives of Vehicle Navigation System.

* It has been tried to make it user friendly.
* It is capable of quick response to the server.
* Obtain video streaming from client to server side.
* For lives video Synchronizing video with location (lat, long)
* Using goggle map for tracing location on both sides.
* Sending latitudes &longitudes of location of Aircraft from client to server, by using steganography.
* Allowing multiple clients to connect the server.
* Notification Alert about crime scenes from server to client.

**CHAPTER II**

**LITERATURE REVIEW**

**2.1** [**C# Language**](javascript:void(0))

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly typed query a first-class language construct.

As an Object Oriented Programming, C# supports the concepts of encapsulation, inheritance and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

* Encapsulated method signatures called delegates, which enable type-safe event notifications.
* Properties, which serve as assessors for private member variables.
* Attributes, which provide declarative metadata about types at run time.
* Inline XML documentation comments.
* Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop." Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical.

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events.

**2.2** [**DOT NET Framework Platform Architecture**](javascript:void(0))

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.

Language interoperability is a key feature of the .NET Framework. Because the IL code produced by the C# compiler conforms to the Common Type Specification (CTS), IL code generated from C# can interact with code that was generated from the .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single assembly may contain multiple modules written in different .NET languages, and the types can reference each other just as if they were written in the same language.

In addition to the run time services, the .NET Framework also includes an extensive library of over 4000 classes organized into namespaces that provide a wide variety of useful functionality for everything from file input and output to string manipulation to XML parsing, to Windows Forms controls. The typical C# application uses the .NET Framework class library extensively to handle common "plumbing" chores.



**2.3 Mobile Technology**

It is the technology used for cellular communication. Mobile code division multiple access (CDMA) technology has evolved rapidly over the past few years. Since the start of this millennium, a standard mobile device has gone from being no more than a simple two-way pager to being a  mobile phone, GPS navigation device, an embedded web browser and instant messaging client, and a handheld game console. Many experts argue that the future of computer technology rests in mobile computing with wireless networking. Mobile computing by way of tablet computers are becoming more popular. Tablets are available on the 3G and 4G networks.

**2.4 IOS**

IOS is the operating system that runs on iPad, iPhone, and iPod touch devices. The operating system manages the device hardware and provides the technologies required to implement native apps. The operating system also ships with various system apps, such as Phone, Mail, and Safari that provide standard system services to the user. The *iOS Software Development Kit (SDK)* contains the tools and interfaces needed to develop, install, run, and test native apps that appear on an iOS device’s Home screen. Native apps are built using the iOS system frameworks and Objective-C language and run directly on iOS. Unlike web apps, native apps are installed physically on a device and are therefore always available to the user, even when the device is in Airplane mode. They reside next to other system apps, and both the app and any user data is synced to the user’s computer through iTunes.

**2.5 RTMP Protocol**

**[Real Time Messaging Protocol](http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol" \o "Real Time Messaging Protocol" \t "_blank)** or RTMP is mainly serving for high speed transmission of audio, video and data between flash player and a server. Initially developed by [Macromedia](http://en.wikipedia.org/wiki/Macromedia" \o "Macromedia" \t "_blank), the protocol is now owned by Adobe, and specifications about it have only partially been released for public use.

According to those specifications, the RTMP protocol has multiple variations, i.e. the “plain” RTMP protocol, RTMPS which is RTMP over an TLS/SSL connection, RTMPE which is RTMP encrypted using [Adobe’s](http://www.adobe.com/" \o "Adobe Systems" \t "_blank) own security mechanism, and RTMPT which is encapsulated within [HTTP requests](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol" \o "Hypertext Transfer Protocol" \t "_blank) to traverse firewalls. The use of RTMP is to avoid latency in communication, mainly, deliver audio or video streams smoothly, and by splitting them in fragments, make them interleaved and multiplexed over a single connection. Also, you save bandwidth. Interleaving and multiplexing is done at the packet level, with RTMP packets across several different active channels being interleaved in such a way as to ensure that each channel meets its bandwidth, latency, and other quality-of-service requirements. RTMP defines several virtual channels on which packets may be sent and received, and which operate independently of each other. During a regular RTMP session, several channels may be active simultaneously at any given time.In result, RTMP encapsulates MP3 or [AAC](http://en.wikipedia.org/wiki/Advanced_Audio_Coding" \o "Advanced Audio Coding" \t "_blank) audio and [FLV1](http://en.wikipedia.org/wiki/Flash_Video" \o "Flash Video" \t "_blank) video multimedia streams, and can make remote procedure calls, or RPCs

**CHAPTER III**

**PROPOSED SYSTEM**

**3.1 Scope of the Proposed System**

**Area of Utilization:**

The software can be used Police Satiations for vehicle tracking and live streaming of crime scene.

**Limitations:**

* Internet is required to connect to server

**3.2 Advantages of the Proposed System:**

Following are the advantages of the proposed system, which will be available to the users (officers) and Sever administration.

* **Easy to maintain**

Proposed system will be very easy to maintain. The vehicle will send the video along with NEMA string to the server and the server will get information about all vehicles.

* **Efficiency**

The proposed system will be more efficient & accurate. It will save the time of the server who wants to know the current position of the vehicle.

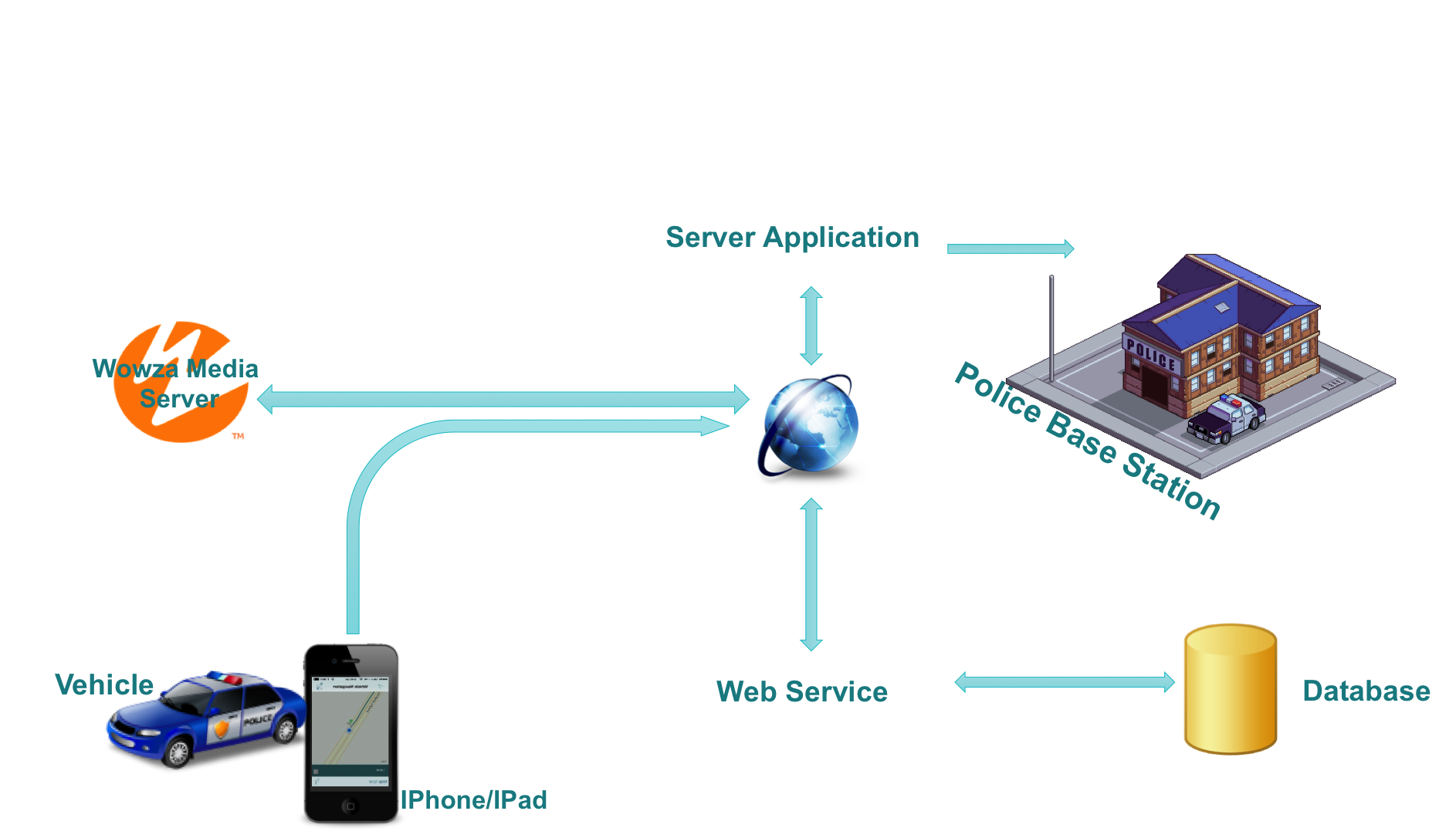
* **User Friendly**

The proposed system will be user friendly and efficient. Server& IOS application is designed according to police station’s needs & requirements.

* **Data Security & Integrity**

The system will be more secure and integrate. They will have specific user name & password to open the admin sections.

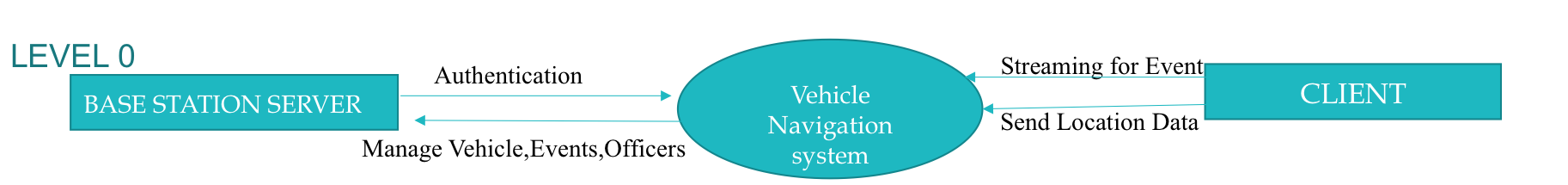
**3.3 Conceptual Diagram:**

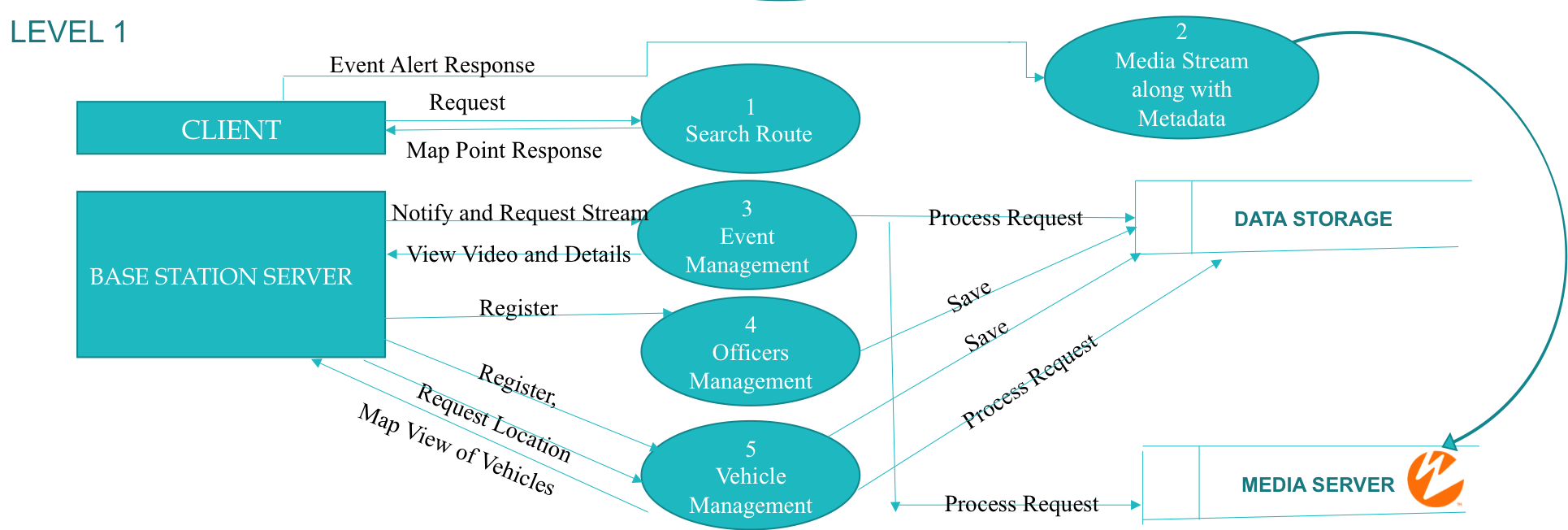
****

**Figure 3.3 : Conceptual Diagram**

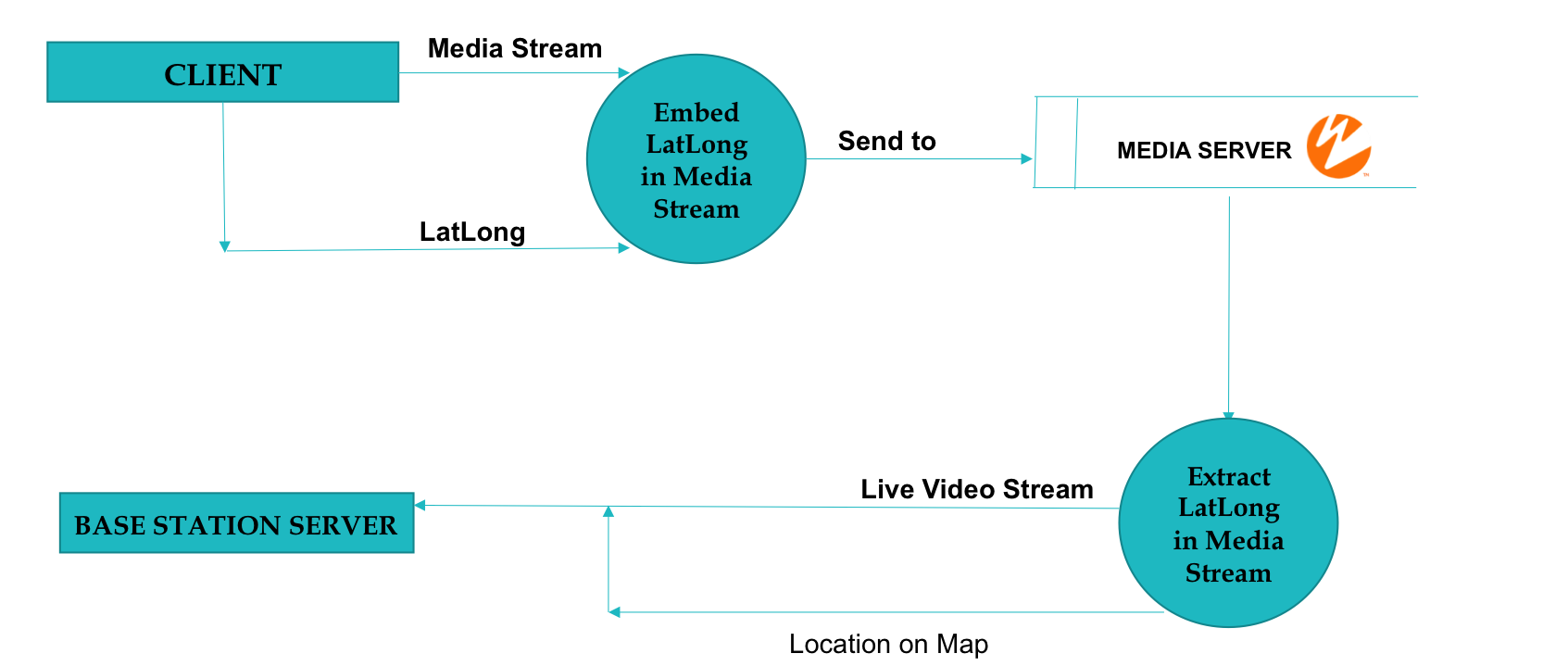
**3.4 Data Flow Diagram:**

**3.4.1 Level 0 DFD:**

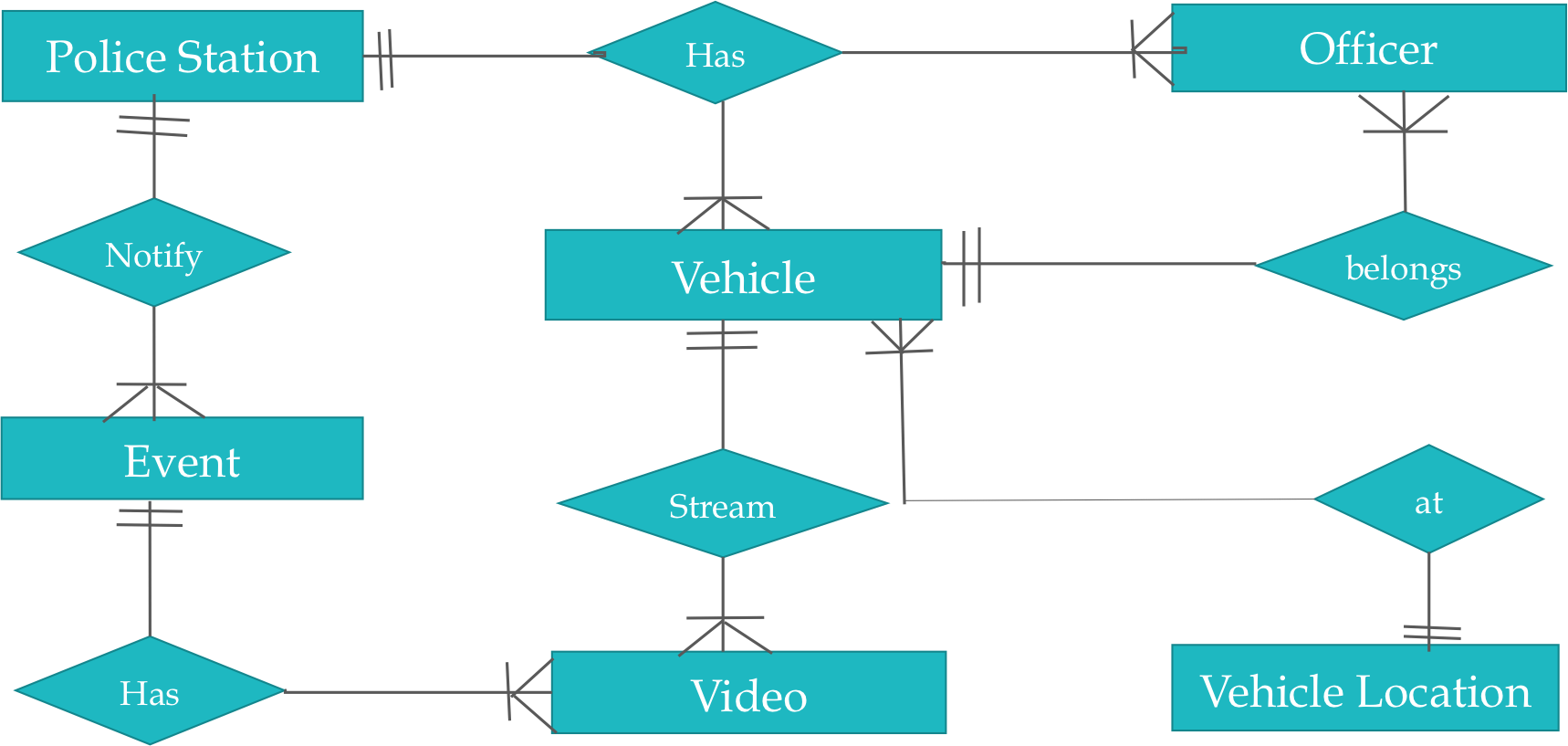
****

** 3.4.2 Level 1 DFD:**

**3.4.3Process 1:**

****

**3.5 Entity Relationship Diagram:**

****

**CHAPTER V**

**FUTURE DIRECTIONS**

The concluding remarks and future directions for the development and up gradation of the existing projects are given below. Some of the limitations of the project are also given below.

**5.1 Concluding Remarks**

Vehicle Navigation System with an IOS Client mode is an excellent project to implement. It covers both aspects server as well as IOS Client Application, which is a challenging task to implement.

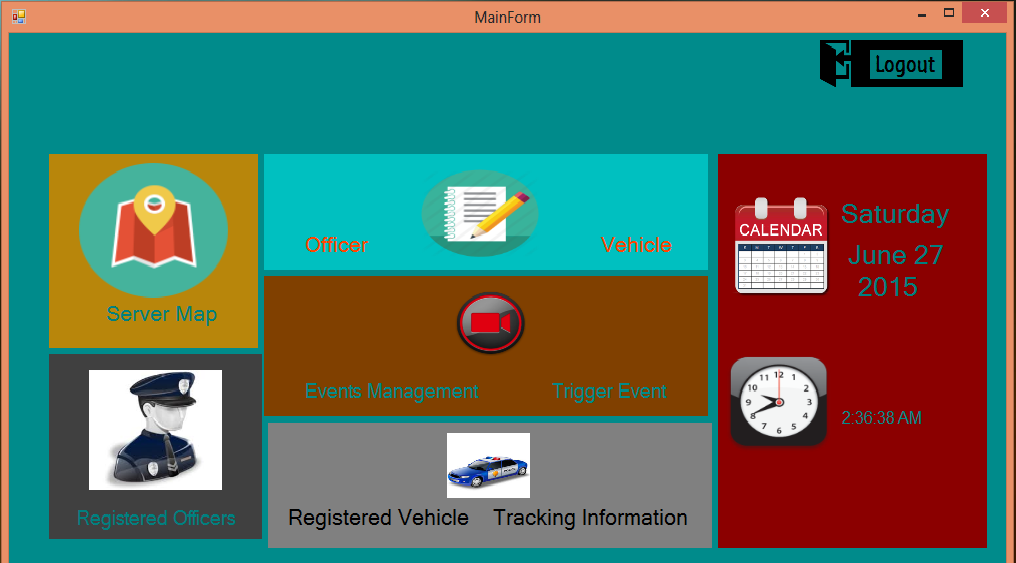
**5.2 Future Directions**

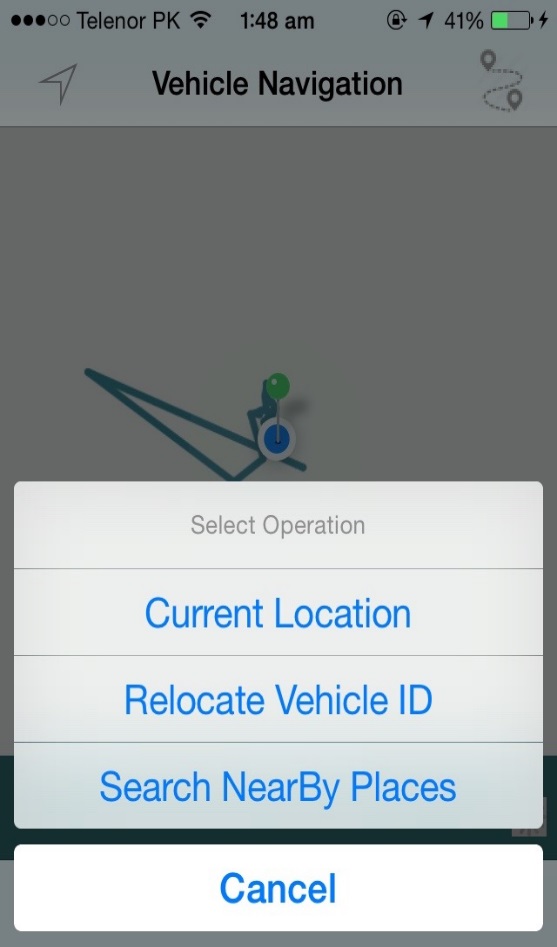
In future I would recommend that Vehicle information should be enhanced to handle speed and direction of vehicle by server as well as add calling feature in IOs client app so that base station can make direct voice connection with officers.

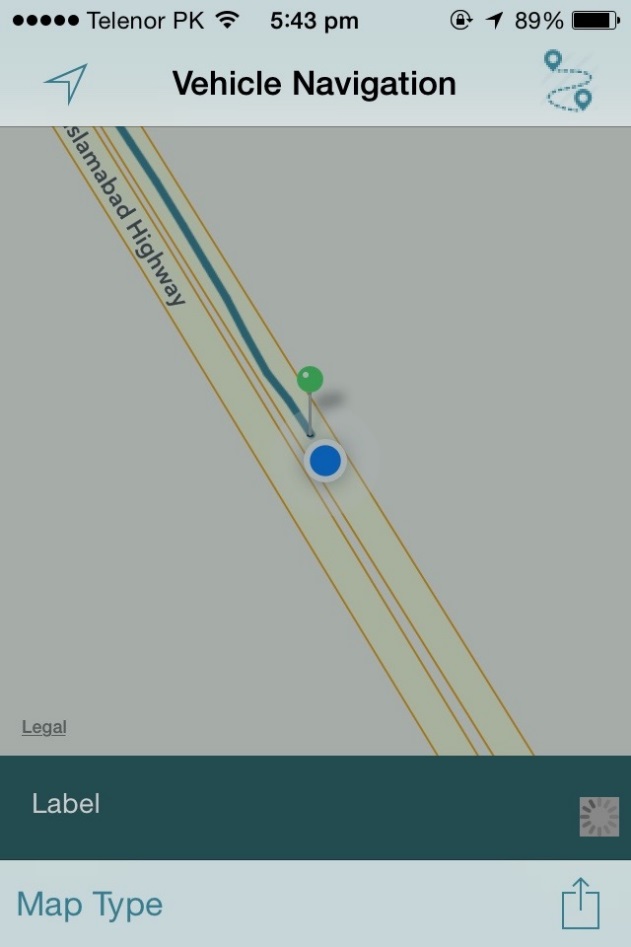
**5.3 Limitations**

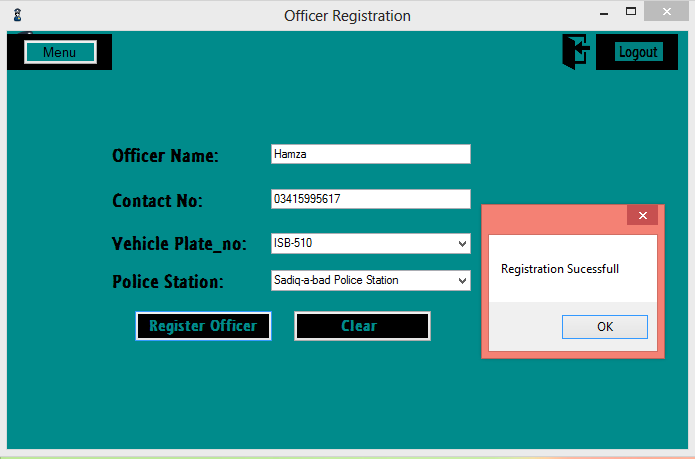
Vehicle Navigation System Client and Server App is limited for only Police Stations which is its one of the biggest limitation.

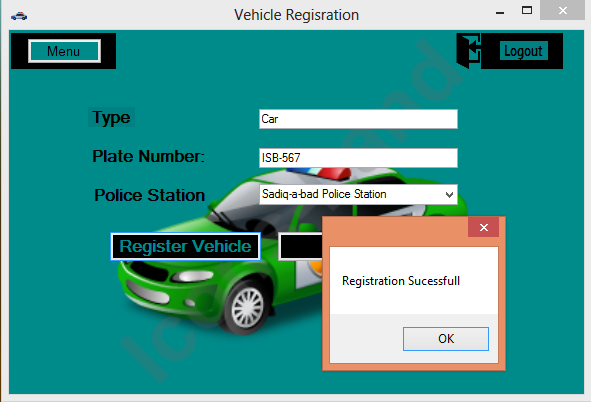
**Screen Shots of Project**

1. **Server Main View**
2. **Client Main View**

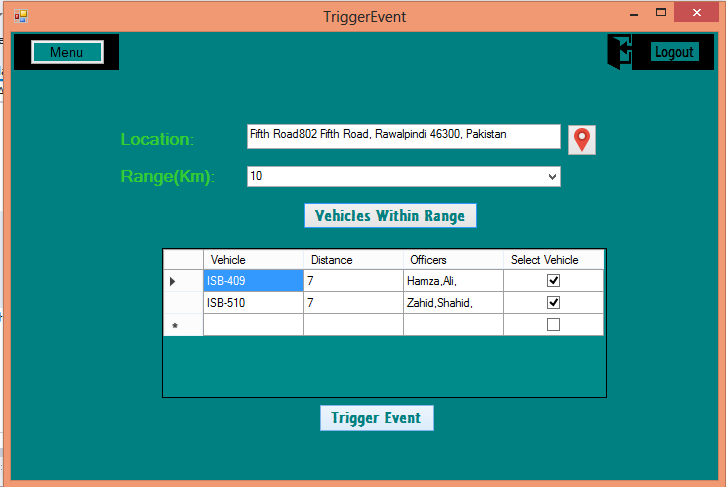


****

1. **Registration**
2. **Officer**
3. **Vehicle**

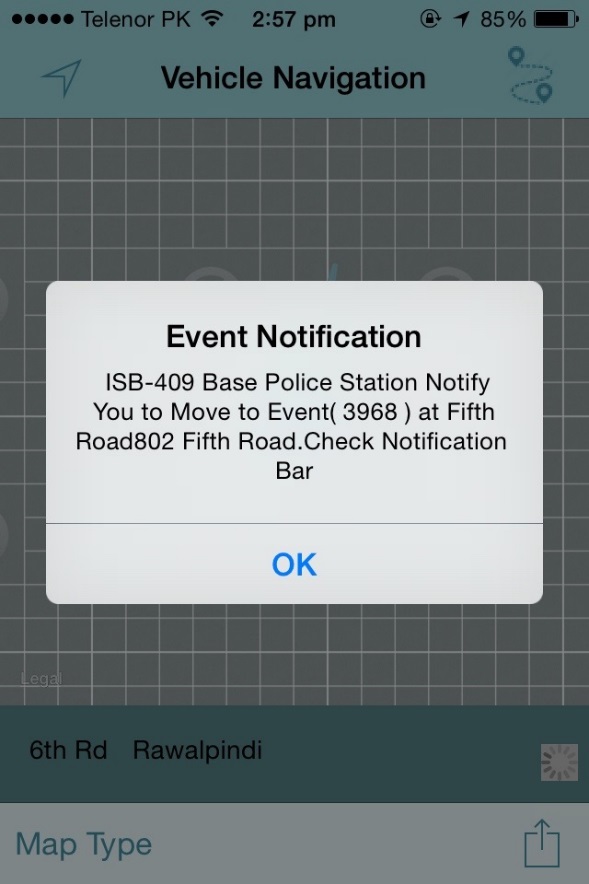
****

1. **Streaming Activity**
2. **Server Trigger Event**

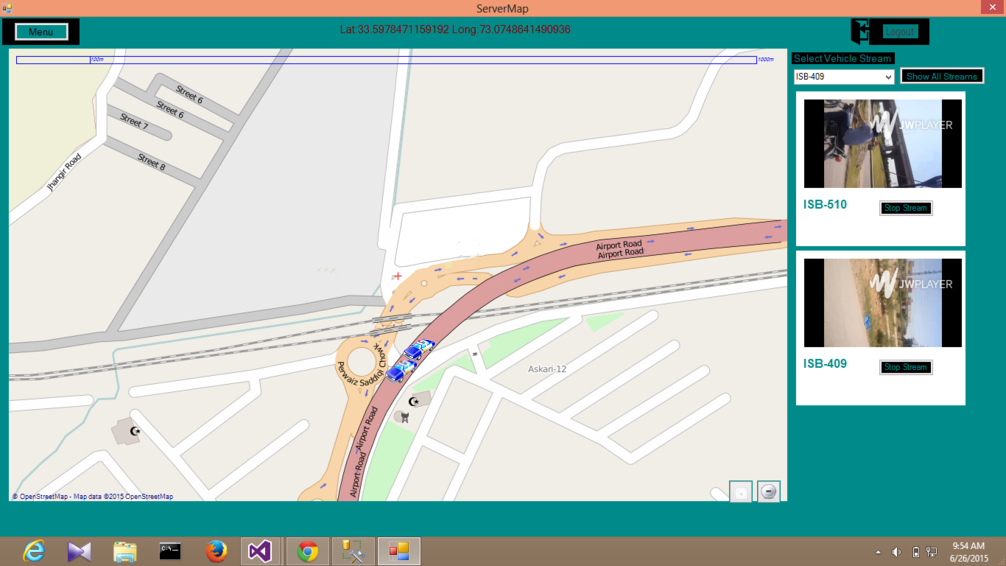


1. **Client Receive Event**

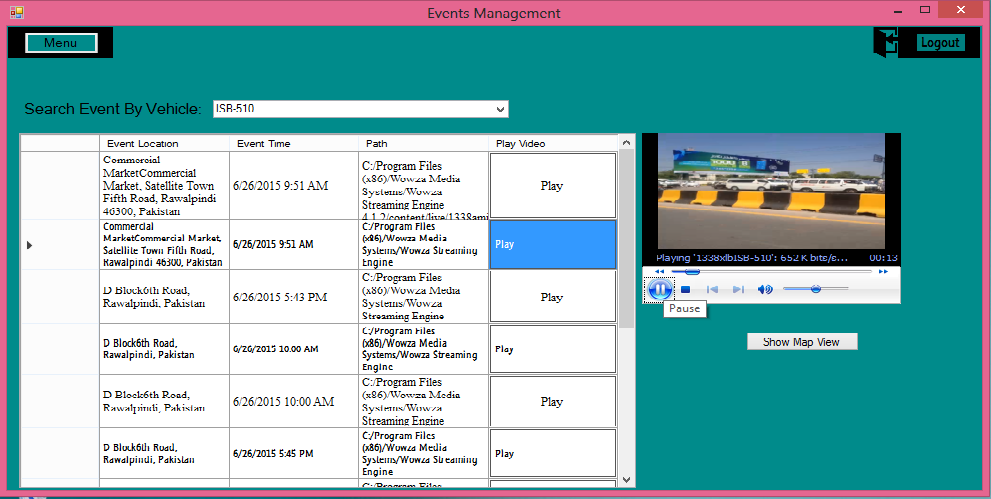


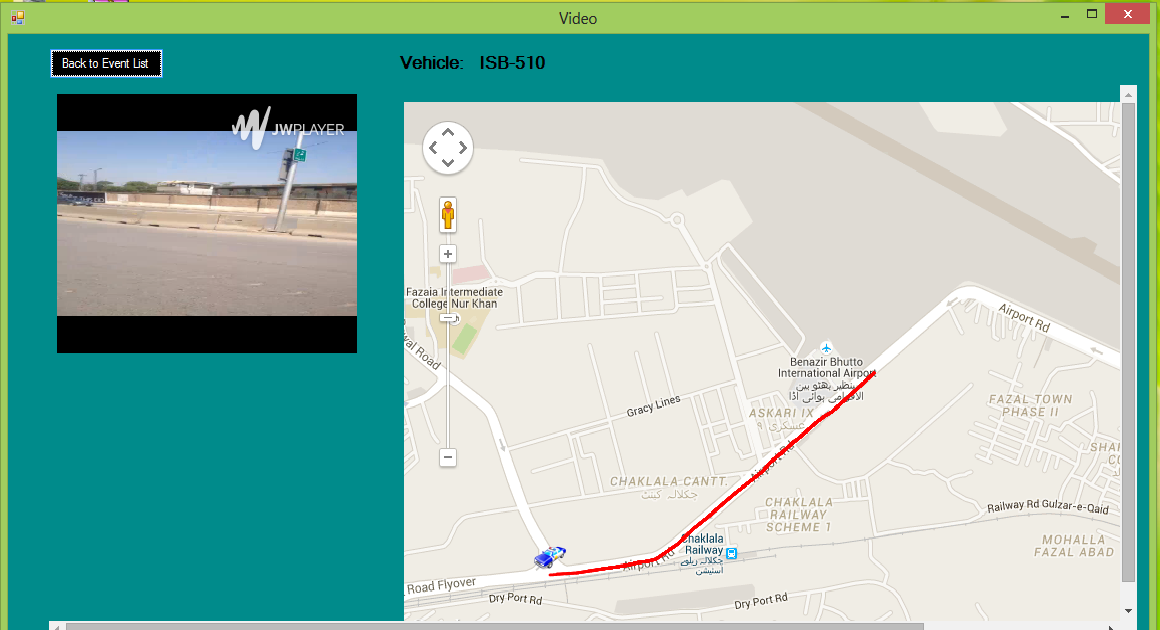
****

1. **Vehicle Location and Live Video Streaming on Server**



1. **Event Video On Demand**

****



**REFRENCES**

Following are the references related to project.

Www.google.com.pk (1998) IOS Development (Accessed 2014,October 19)

En.wikipedia.org/wiki/Main Page (2002) GPS (Accessed 2014, September 24)

Msdn.microsoft.com/en-US/(2000) Web based (ADO.NET) (Accessed 2014 May 25)

[www.codeproject.com](http://www.codeproject.com) C# development (Accessed August 2014- May 2015)

[www.appcoda.com](http://www.appcoda.com) IOS Concepts (Accessed August 2014- May 2015)

[www.**raywenderlich**.com/](http://www.raywenderlich.com/) IPhone Tutorials (Accessed August 2014- May 2015)