**A PROJECT REPORT ON**

**GPS Tracking System**

**For**

**Neni Infotech**

**SUBMITTED BY**

**Mr. Amol Gadhave**

**Miss. Deepali Gavhale**

**Miss. Shital Kadam**

**Department of Master of Computer Applications**

**UNDER THE GUIDANCE OF**

**Miss Shweta Patil**

**Mr. BibinBabu**(company’s guide)

**SUBMITTED TO**

**YADAVRAO TASGAONKAR INSTITUTE OF ENGINNERING & TECHNOLOGY**

**CHANDHAI, KARJAT -410 201**

**UNIVERSITY OF MUMBAI**

**2012-2013**

**Internship Completion Certificate**

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Gadhave Amol S, Miss. Gavhale Deepali R, and Miss. Kadam Shital S Seat No. 9202,9059, 9071 students** of MCA Course from **Yadavrao Tasgaonkar Institute of Engineering and Technology** has successfully completed his sixth semester Project training at **Neni Infotech.**

**Name of the Project : GPS Tracking System**

**Duration : 1st January 2013 to 30thJune 2013**

**Front End : ASP. NET 2012**

**Back End : SQL Server 2012**

Their project is found to be useful in the relevant business and they have submitted a copy of the project report to us. During their project period we find them sincere, hard working and having good behavior and moral character. We wish them all success in their future endeavor.

**C E R T I F I C A T E**

This is to certify that **Mr.Gadhave Amol S, Miss.Gavhale Deepali R, Miss.Kadam Shital S**Seat No. **9202, 9059, 9071**has completed Project on“**GPS Tracking System”** for the requirement of 3rd year (VI Sem) of Master of Computer Application of University of Mumbai during the year 2012-13.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Internal Guide Coordinator MCA

College Seal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner Principal

Date: \_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgement**

Any accomplishment requires the endeavor of many people, and we are indebted to many individuals who contributed to the development of this project work. It is with a great pleasure that we present our first venture in real life application of computing in the form of project work.

We are indebted & thankful to our Project guide and to whom we owe this piece of knowledge his valuable guidance, cooperation and encouragement and time spent for this project work.First of all we extend our sincere thanks to the faculty members of M.C.A department who tuned our knowledge in fieldof I.T. We sincerely thank them for their unconditional support and also for their co-operation and guidance throughout the course of this project with respect to every aspect of the project.

We extend our sincere thanks to H.O.D ofM.C.A (DEPT) Prof. Rupesh Bhoir.We are also thankful to Mr. Bibin Babu and Prof. Miss. Shweta Patil who showed faith in me and guided me throughout the course of the project.

We are also grateful to my family members for constantly supporting us and we would like to thank all those who helped me directly or indirectly during my project encouraging us to put our best efforts for continued improvements of our project.

Gadhave Amol S.

Gavhale Deepali R.

Kadam Shital S.

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr no** | **Contents** | **Page No** |
| **1** | **Introduction**   * 1. **Abstract**   1.1.1 About the Organization  1.1.2 Services Provided  1.1.3 About the Project   * 1. **Objective & Scope Of the Project**   2. **Theoretical Background**   1.3.1 Overview of Front End  1.3.2 Technology Used  1.3.3 Overview of Back End | 8  8  8  9  10  14  13  17  27 |
| **2** | **System Analysis**  **2.1 Feasibility Study**  2.1.1 Technical  2.1.2 Operational  2.1.3 Resource  (S/W development Model depends on Project Type)  **2.2 System Planning and Schedule**  2.2.1 GANTT Chart | `  35  35  36  36  40  40 |
| **3** | **System Design**  **3.1 Software Requirement Specification**  3.1.1 Introduction  3.1.2 Selection of Technology/Specific Requirements  3.1.2.1 Hardware to be used  3.1.2.2 Software to be used  3.1.2.3 Tools to be used  **3.2 Detailed Life Cycle of the Project**  3.2.1 Modules  3.2.2 Object Oriented Analysis & Design Diagrams  3.2.2.1 UML Diagrams  3.2.2.2 DataFlow Diagrams  3.2.3 Database  3.2.3.1 Database table | 42  42  45  45  46  46  48  51  51  52  52  56  67 |
| **4** | **Testing (V Model)**  **4.1 Methodologies used for testing**  **4.2 Types of testing**  4.2.1 White Box Testing  4.2.2 Black Box Testing  **4.3 Test Cases**  4.3.1 Functionality Testing  4.3.2 Usability Testing  4.3.3 Performance Testing  **4.4 Test Reports** | 74  76  76  78  80  80  80  80  81 |
| **5** | **System Implementation**  **5.1Hardware Required at client**  **5.2 Software Required at Client**  **5.3 Testing done at End User** | 85  85  85 |
| **6** | **System Maintenance & Evaluation**  **6.1 Maintenance**  **6.2 Evaluation** | 88  90 |
| **7** | **User/Operational Manual**  **7.1 Security Aspects, Access Rights**  **7.2 Back Up’s**  **7.3 Screen Shots** | 93  94  96 |
| **8** | **Future Enhancements** | 110 |
| **9** | **Limitations** | 119 |
| **10** | **Conclusion** | 121 |
| **11** | **Appendices** | 123 |
| **12** | **Bibliography** | 126 |

**1.INTRODUCTION**

**1.1 Abstract**

The main aim of our project is to create a system for a company so that it can keep track of its vehicle and even give the alert if the vehicle goes outside of geofence area on their own maps. The main objective of our project will be to get the things on hand whenever required through Database and can check the entries about each and every user.

The system can be used by Admin and even user to check there system. Admin can create its account required and even assign each account to the user. Only Admin has the right to assign the account. Admin can even track the vehicle and those values will be stored in the database and again retrieved from the database and plotted on the map. Admin can even see the directions followed by the driver. Admin can update any information about his system as per his requirements and can even change the pickup points of the user.

* + 1. **About the Organization**

NeniInfotech is software development and service Provider Company mainly working for clients in USA.

The company’s expertise is in development of unique software which helps automating day to day task in the back office operations of Investment companies.

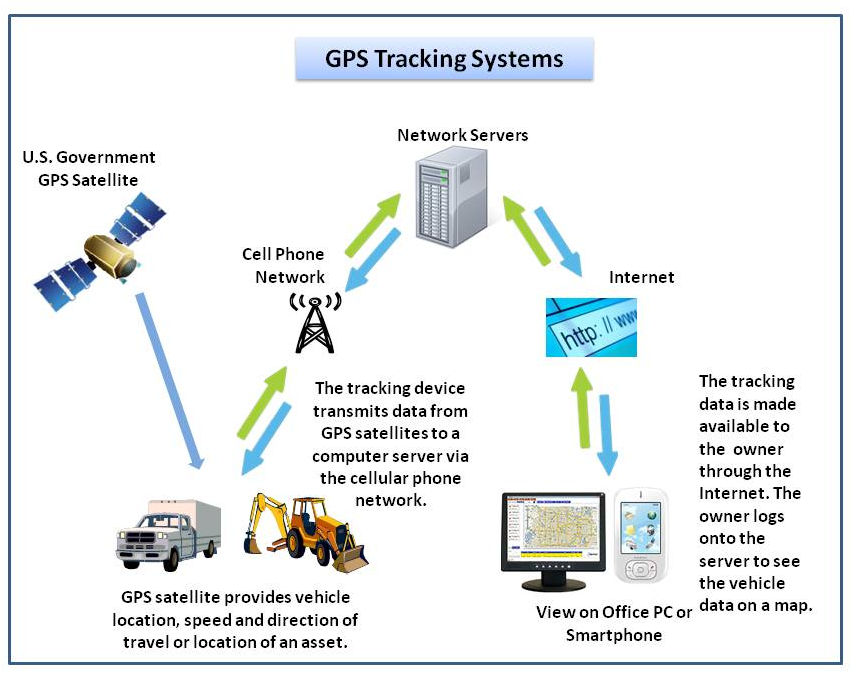
They target complex environments where other companies fail or do not deliver properly. They like to solve problems, accept new challenges and provide a satisfactory solution to end users.

They interact with end users continuously in all the phases of design, development, testing and production. They also maintain, release system in client environment.

**1.1.2 Services Provided**

NeniInfotech has in-depth experience in developing business applications and strives to provide the custom software solutions to the clients that will help him to remain in step with the competition by continuously improving their information technology based business solutions. For this, the company use modern software development platforms, software development tools as well as employ latest project management techniques and software. The company has gained vital process knowledge and has distinctive library of software codes. Company also provides other services such as, Website Designing & Development.

* + 1. **About the Project**

****

GPS Tracking System is a system where your vehicle can go anywhere within your sight monitoring. GTS provides live GPS tracking that enables up to the minute information regarding your fleet vehicles.

The easy-to-use web interface allows the most convenient way to continuously track, get stop times, set geo-fences, set speed limit alerts, and always instantly know where your trucks and drivers are located. Vehicle locations are stored in a database and archived for future replay and review.

A GPS Tracking System is an electronic device installed in your vehicle to enable the user to monitor its current location at live time.

Many systems also combine a communications component such as cellular or satellite transmitters to communicate the vehicle’s location to a remote user. Vehicle information can be viewed on electronic maps via the Internet or specialized software.

* 1. **Objective & Scope of the Project**

**Objective of the Project**

The most basic function in all tracking systems is the vehicle and person tracking component. This component is usually GPS-based. Once vehicle location, direction and speed are determined from the GPS components, additional tracking capabilities transmit this information to a tracking software application.

Methods for data transmission include both terrestrial and satellite. Users can see actual, real- time locations of their vehicle on a map. This is often used to quickly respond on events in the field.

The main objective of the project is to design and develop a user friendly, easy to use and an efficient computerized system.

To help the management to keep a strong-hold on the all the process involved specially in process activities when vehicle is in running.

Computerization can be helpful in reducing the manual workload.

Computerization can be viewed as a means of saving time and cost.

To provide better Graphical User Interface (GUI).

Perform necessary validations & minimizes errors.

It can be suitably modified & expanded in future for further needs.

* **Prevention of Car Theft**: By installing a tracking device on a vehicle, the location of it can always be known.
* **Taking Care of the Child &Elderly**: The child and elderly can benefit from tracking services that include the ability to place emergency calls.
* **Police & Private Detectives**: Police can benefit by attaching tracking devices to vehicles during investigations, allowing them to easily track movement and come up with solid evidence.
* **Company Vehicle Fleet Tracking**: Using the devices for fleet tracking lets companies manage their routes of their fleets.
* **Pets Tracking**: A tracking device can be easily embedded under a pet’s skin.
* **Child Tracking:** A tracking device can be easily embedded into toys.

**Feasibility Report**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility
* Economical Feasibility

**Technical Feasibility:**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

**Operational Feasibility:**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

**Economic Feasibility:**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

**Scope:**

These type of system can be used in various industries ambulance services; fleet management; public transport service management systems; computers in the cab; vehicle location and position monitoring using satellite and cellular telephone; vehicle tracking system; mapping techniques; and fleet management systems integration.

* User can create the many devices for his one account and maintains the records of devices for each user.
* Also maintains the records of users along with its account and devices.
* Maintains the records of location of the device (or person who carry that device).
* Maintains the records ofall geofence areas.
* Gives alert to the User if device goes out of predefined geofence area.
* Maintains the records of the live points (i.e. latitude & longitude) along with creation date time.
  1. **Theoretical Background**
     1. **Overview of Front End**

**C# .net**

C# is a new computer-programming language developed by Microsoft Corporation, USA. C# is a fully object-oriented language like Java and is the first Component-orientedlanguage. It has been designed to support the key features of .NET Framework, the newdevelopment platform of Microsoft for building component-based software solutions. It is asimple, efficient, productive and type-safe language derived from the popular C and C++, it isa purely object-oriented, modern language suitable for developing Web-based applications.C# is designed for building robust, reliable and durable components to handle real-world applications.

Major highlights of C# are:

It is a brand new language derived from the C / C++ family

It simplifies and modernizes C+

It is the only component-oriented language available today

It is the only language designed for .Net Framework

It is a concise, lean and modern language

Microsoft wanted an environment that is completely in tune with current and emergingWeb-programming practices and one that easily integrates with existing systems. Microsofttherefore decided to design a new language starting with a clean slate. The result is C#, asimple and modern language that directly addresses the needs of component-based software development. Theplatform should be a collection of readily available Web service that can be distributed andaccessed via standard Internet protocols. He wanted to make the Web both programmable andintelligent. The outcome is a new generation platform called .Net.

C# fulfills the need for a language that is easy to write, read and maintain and alsoprovides the power and flexibility of C++. The language that is designed for both computingand communication is characterized by several key features. It is

* Simple and Consistent
* Modern
* Object-Oriented
* Type-Safe
* Version able
* Compatible
* Flexible
* Inter-operability

**1.3.2 Technology Used**

**MVC**

MVC is one of three ASP.NET programming models.

MVC is a framework for building web applications using a MVC (Model View Controller) design:

* The Model represents the application core (for instance a list of database records).
* The View displays the data (the database records).
* The Controller handles the input (to the database records).

The MVC model also provides full control over HTML, CSS, and JavaScript.

|  |  |
| --- | --- |
| MVC | The MVC model defines web  applications with 3 logic layers:  The business layer (Model logic)  The display layer (View logic)  The input control (Controller logic) |

**The Model** is the part of the application that handles the logic for the application data.  
Often model objects retrieve data (and store data) from a database.

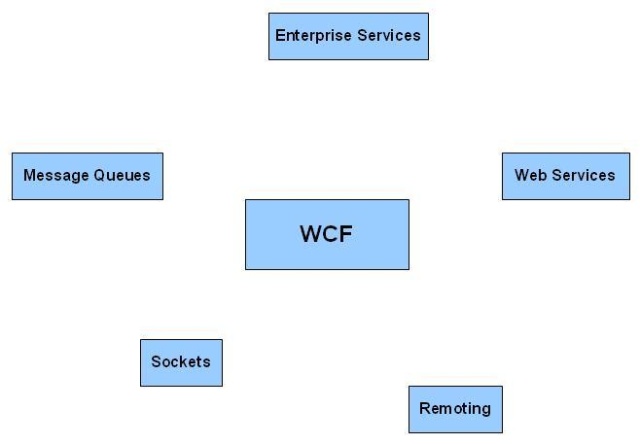
**The View** is the parts of the application that handles the display of the data.  
Most often the views are created from the model data.

**The Controller** is the part of the application that handles user interaction.  
Typically controllers read data from a view, control user input, and send input data to the model.

**WCF**

Windows Communication Foundation (**WCF**) comes to the rescue when we find ourselves not able to achieve what we want to achieve using web services, i.e., other protocols support and even duplex communication. With **WCF**, we can define our service once and then configure it in such a way that it can be used via HTTP, TCP, IPC, and even Message Queues. We can consume Web Services using server side scripts (ASP.NET), JavaScript Object Notations (JSON), and even REST (Representational State Transfer).

A **WCF** service can be visualized as:



When we say that a **WCF** service can be used to communicate using different protocols and from different kinds of applications, we will need to understand how we can achieve this. If we want to use a **WCF** service from an application, then we have three major questions:

1. Where is the **WCF** service located from a client's perspective?
2. How can a client access the service, i.e., protocols and message formats?
3. What is the functionality that a service is providing to the clients?

Once we have the answer to these three questions, then creating and consuming the **WCF** service will be a lot easier for us. The **WCF** service has the concept of endpoints. A **WCF** service provides endpoints which client applications can use to communicate with the **WCF** service. The answer to these above questions is what is known as the ABC of **WCF**services and in fact are the main components ofa**WCF**service.

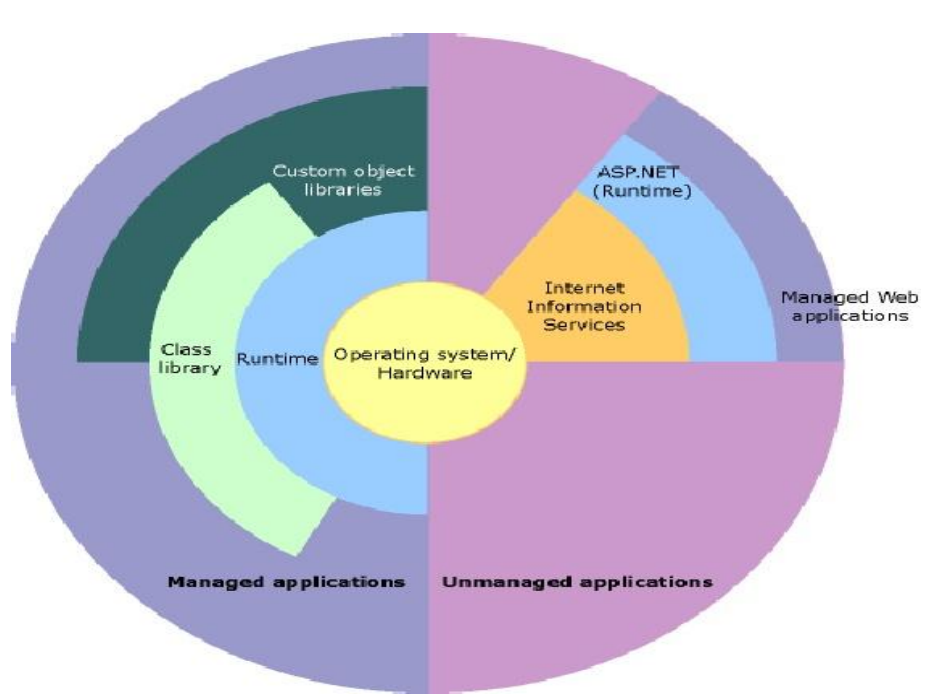
**Address**: Like a web service, a **WCF** service also provides a URI which can be used by clients to get to the **WCF** service. This URI is called as the Address ofthe WCFservice. This will solve the first problem of "where to locate the **WCF**service?" for us.

**Binding**: Once we are able to locate the **WCF** service, we should think about how to communicate with the service (protocol wise). The binding is what defines howthe

**WCF** service handles the communication. It could also define other communication parameters like message encoding, etc. This will solve the second problem of "how to communicate with the **WCF** service?" for us.

**Contract**: Now the only question we are left up with is about the functionalities that a **WCF** service provides. Contract is what defines the public data and interfaces that **WCF** service provides to the clients.

# Introduction To .Net Framework



The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* .NET Framework can integrate with any other code.

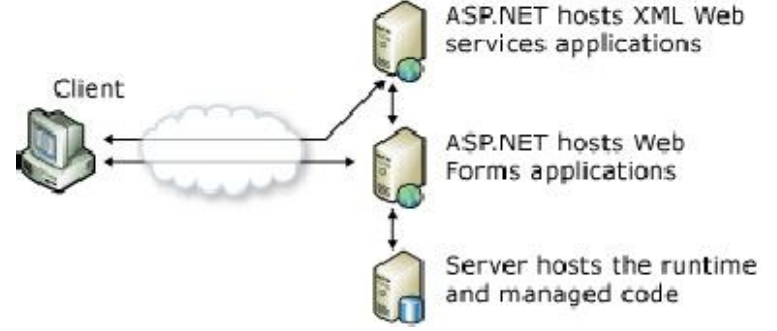
The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension).

**Server Application Deployment:**

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the Performance and scalability of the host server. The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code. Server-side managed code.



## Features of Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

## .NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

* Console applications.
* Scripted or hosted applications.
* Windows GUI applications (Windows Forms).
* ASP.NET applications.
* XML Web services.
* Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

**1.3.3 Overview of Back End**

**Microsoft SQL Server**

Microsoft SQL Server is a relational database server, developed by Microsoft: it is as software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time).

About the time Windows NT was released, Sybase and Microsoft parted ways and each pursued its own design and marketing schemes. Microsoft negotiated exclusive rights to all versions of SQL Server written for Microsoft operating systems

**Services**

SQL Server also includes an assortment of add-on services. While these are not essential for the operation of the database system, they provide value added services on top of the core database management system. These services either run as a part of some SQL Server component or out-of-process as Windows Service and presents their own API to control and interact with them.

**Service Broker**

Used inside an instance, it is used to provide an asynchronous programming environment. For cross instance applications, Service Broker communicates over TCP/IP and allows the different components to be synchronized together, via exchange of messages. The Service Broker, which runs as a part of the database engine, provides a reliable messaging and message queuing platform for SQL Server applications.

**Replication Services**

SQL Server Replication Services are used by SQL Server to replicate and synchronize database objects, either in entirety or a subset of the objects present, across replication agents, which might be other database servers across the network, or database caches on the client side. Replication follows a publisher/subscriber model, i.e., the changes are sent out by one database server ("publisher") and are received by others ("subscribers"). SQL Server supports three different types of replication:

* + Transaction replications
  + Merge replication
  + Snapshot replication

**Analysis Services**

SQL Server Analysis Services adds OLAP and data mining capabilities for SQL Server databases. The OLAP engine supports MOLAP, ROLAP and HOLAP storage modes for data. Analysis Services supports the XML for Analysis standard as the underlying communication protocol. The cube data can be accessed using MDX and LINQ queries. Data mining specific functionality is exposed via the DMX query language.

**Reporting Services**

SQL Server Reporting Services is a report generation environment for data gathered from SQL Server databases. It is administered via a web interface. Reporting services features a web services interface to support the development of custom reporting applications. Reports are created as RDL files. Reports can be designed using recent versions of Microsoft Visual Studio (Visual Studio.NET 2003, 2005, and 2008) with Business Intelligence Development Studio, installed or with the included Report Builder. Once created, RDL files can be rendered in a variety of formats including Excel, PDF, CSV, XML, TIFF (and other image formats), and HTML Web Archive.

**Notification Service**

Originally introduced as a post-release add-on for SQL Server 2000, Notification Services was bundled as part of the Microsoft SQL Server platform for the first and only time with SQL Server 2005. SQL Server Notification Services is a mechanism for generating data driven notifications, which are sent to Notification Services subscribers.

**Integration Services**

SQL Server Integration Services is used to integrate data from different data sources. It is used for the ETL capabilities for SQL Server for data warehousing needs. Integration Services includes GUI tools to build data extraction workflows integration various functionality

**Full Text Search Service**

SQL Server Full Text Search service is a specialized indexing and querying service for unstructured text stored in SQL Server databases. The full text search index can be created on any column with character based text data. It allows for words to be searched for in the text columns. While it can be performed with the SQL LIKE operator, using SQL Server Full Text Search service can be more efficient.

**SQL CMD**

SQLCMD is a command line application that comes with Microsoft SQL Server, and exposes the management features of SQL Server. It allows SQL queries to be written and executed from the command prompt. It can also act as a scripting languageto create and run a set of SQL statements as a script. Such scripts are stored as a .sql file, and are used either for management of databases or to create the database schema during the deployment of a database.SQLCMD was introduced with SQL Server 2005 and this continues with SQL Server 2008.

**Visual Studio**

**Microsoft Visual Studio** includes native support for data programming with Microsoft SQL Server. It can be used to write and debug code to be executed by SQL CLR. It alsoincludes a *data designer* that can be used to graphically create, view or edit databaseschemas. Queries can be created either visually or using code. SSMS 2008 onwards,provides intelligence for SQL queries as well.

**SQL Server Management Studio**

**SQL Server Management Studio** is a GUI tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. SQL Server Management Studio replaces Enterprise Manager as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as *SQL Server Management Studio Express* (SSMSE). SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

**2. SYSTEM ANALYSIS**

* 1. **Feasibility Study**

**Feasibility studies** aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements and tax obligations.

**2.1.1 Technical Feasibility**

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project. When writing a feasibility report the following should be taken to consideration:

* A brief description of the business to assess more possible factor/s which could affect the study
* The part of the business being examined
* The human and economic factor
* The possible solutions to the problems

At this level, the concern is whether the proposal is both *technically* and *legally* feasible (assuming moderate cost).

**2.1.2 Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

* + 1. **Resource Feasibility**

Resources for Change have a long standing involvement and core interest in the development of projects linked to the countryside and to locally-based eco-tourism. Our philosophy is that successful and sustainable projects can be developed around the particular characteristics and resources of our rural communities, when the local community is fully involved and on board, and where there has been a realistic assessment of the opportunities and economics of the project. In carrying out feasibility studies Resources for Change works hard to consult and involve (in the short or longer term) all stakeholders and to provide a realistic assessment of the options available.

**Software Development Modules**

**Iterative model of SDLC**

An iterative lifecycle model does not attempt to start with a full specification of requirements.Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model. Consider an iterative lifecycle model which consists of repeating the following four phases in sequence:

******

A Requirements phase, in which the requirements for the software are gathered and analyzed.Iteration should eventually result in a requirements phase that produces a complete and final specification of requirements.

A Design phase, in which a software solution to meet the requirements is designed. This may be a new design, or an extension of an earlier design. An Implementation & Test Phase, when the software is coded, integrated and tested.

**Advantages of the Iterative Model**

* + **Flexibility:**

A comparison of the traditional lifecycle to the iterative one reveals that the iterative model is more flexible, as it presents more opportunities to introduce change. In the iterative model, change is an acknowledged, integral component, change. In the iterative model, change is an acknowledged, integral component

* + **Quality Improvement:**

The embrace of source code revision has a profound and positive impact on software quality. When errors are found, they can be corrected at best real-time, at worst in the next iteration. Contrast this to the waterfall model, where software is often released with major defects--because it is too late in the lifecycle to rewrite, or redesign components.

* + **Low Complexity:**

Advantage of the iterative model is that the complexity of implementing the system is never overwhelming. Because elements are designed, developed and integrated in iterations, the "analysis paralysis" that is common on enterprise scope projects is alleviated. In addition, the developers get a chance to grow with the project. Each iteration can leverage the business knowledge gained on the previous and the team gets used to delivering finished software.

* + **The Requirements Phase:**

As with any lifecycle model, during the requirements analysis phase the quality team is engaged in test planning and design. This is nothing new. What would be different from the waterfall model is the vision of how the test plan design will be applied later in the project. For example, the waterfall team would be planning an integration and system test as the last steps to complete before software release.

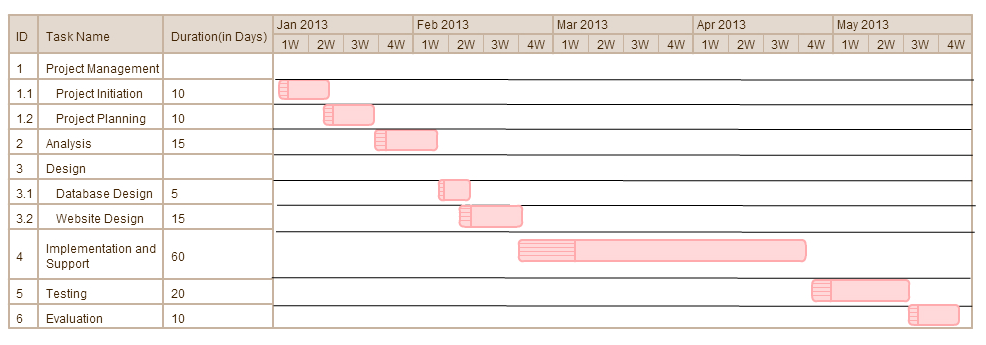
* + **The Design Phase:**

In the design phase, software architecture and components to meet the requirements are designed. Testing activity here begins to focus on the most atomic element i.e.the unit. In the first iteration of a project, there may not be much to do here. In subsequent iterations, the focus will be on how new or changed components will affect the system i.e. regression testing.

* + **The Implementation Phase**

In all iterations, the implementation phase will be saturated with testing. Additional time will be required in early cycles, as the bulk of the test creation and coding will be perform then. A well-designed automated test will have a lifecycle parallel to, and should be considered a sibling of, the project source code.

**2.2 System Planning and Schedule**

**2.2.1GANTT chart**

**3. SYSTEM DESIGN**

**3.1 Software Requirement Specification**

**3.1.1 Introduction**

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. It is an early stage in the more general activity of requirements engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing software or system requirements. Conceptually, requirements analysis includes three types of activities:

**Eliciting requirements:**

The task of identifying the various types of requirements from various sources including project documentation, (e.g. the project charter or definition), business process documentation, and stakeholder interviews. This is sometimes also called requirements gathering.

**Analyzing requirements:**

Determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.

**Recording requirements:**

Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases, user stories, or process specifications. Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. These may include the development of scenarios (represented as user stories in agile methods), the identification of use cases, the use of workplace observation or ethnography, holding interviews, or focus groups (more aptly named in this context as requirements workshops, or requirements review sessions) and creating requirements lists. Prototyping may be used to develop an example system that can be demonstrated to stakeholders.

**Requirements analysis topics**

**Stakeholder Identification**

Stakeholders (SH) are people or organizations (legal entities such as companies, standards bodies) that have a valid interest in the system. They may be affected by it either directly or indirectly. A major new emphasis in the 1990’s was a focus on the identification of stakeholders. It is increasingly recognized that stakeholders are not limited to the organization employing the analyst.

Other Stakeholders will include:

* anyone who operates the system (normal and maintenance operators)
* anyone who benefits from the system (functional, political, financial and social beneficiaries)
* organizations which regulate aspects of the system (financial, safety, and other regulators)
* organizations responsible for systems which interface with the system under design
  + Stakeholder Interview.

Stakeholder interviews are a common technique used in requirement analysis. Though they are generally idiosyncratic in nature and focused upon the perspectives and perceived needs of the stakeholder, often this perspective deficiency has the general advantage of obtaining a much richer understanding of the stakeholder's unique business processes, decision-relevant business rules, and perceived needs.

**Joint Requirements Development Sessions**

Requirements often have cross-functional implications that are unknown to individual stakeholders and often missed or incompletely defined during stakeholder interviews. These cross-functional implications can be elicited by conducting JRD sessions in a controlled environment, facilitated by a trained facilitator, wherein stakeholders participate in discussions to elicit requirements, analyze their details and uncover cross-functional implications. A dedicated scribe and Business Analyst should be present to document the discussion. Utilizing the skills of a trained facilitator to guide the discussion frees the Business Analyst to focus on the requirements definition process.

**Contract-style requirement lists**

One traditional way of documenting requirements has been contract style requirement lists. In a complex system such requirements lists can run to hundreds of pages. An appropriate metaphor would be an extremely long shopping list. Such lists are very much out of favor in modern analysis; as they have proved spectacularly unsuccessful at achieving their aims; but they are still seen to this day.

**Strengths:**

* Provides a checklist of requirements.
* Provide a contract between the project sponsor(s) and developers.
* For a large system can provide a high level description.

**Weaknesses:**

* Such lists can run to hundreds of pages. They are not intended to serve as a reader friendly description of the desired application.
* Such requirements lists abstract all the requirements and so there is little context. The Business Analyst may include context for requirements in accompanying design.

**3.1.2 Selection of Technology / Specific Requirements**

**3.1.2.1 Hardware to be used**

Below is the Technical Specification for Asset Management System Application’s smooth functioning, which covers hardware and software details

* Processor:Pentium and above.
* RAM:256MB.
* Hard Disk:Free Space 2GB and above.
* Sim Card: Standard GSM sim
* Tk102 tracing device

**3.1.2.2 Software to be used**

* Software : Visual Studio .Net 2012
* Language : C# .Net
* Database : SQL Server-2012
* Operating System : Windows 7,XP
* Web Technologies : WCF, MVC4
* Web server : IIS
* GUI Tools : HTML, JavaScript, JQuery, CSS

**3.1.2.3 Tools to be used**

**jQuery**

jQuery is free, open source software, dual-licensed under the MIT License or the GNU General Public License, Version 2. jQuery’s syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

jQuery includes the following features:

* DOM element selections using the cross-browser open source selector engine Sizzle, a spin-off out of the jQuery project.
* DOM traversal and modification (including support for CSS 1-3
* DOM manipulation based on CSS selectors that uses node elements name and node elements attributes (id and class) as criteria to build selector
* Events
* Effects and animations
* Extensibility through plug-in
* Utilities - such as user agent information, feature detection
* Compatibility methods that are natively available in modern browsers but need fallbacks for older ones - For example the inArray() and each() functions.
* Cross-browser support

**JavaScript**

Imperative and structured

JavaScript supports much of the structured programming syntax from C (e.g., if statements, while loops, switch statements, etc.). One partial exception is scoping: C-style block-level scoping is not supported (instead, JavaScript has function-level scoping). JavaScript, however, supports block-level scoping with the let keyword. Like C, JavaScript makes a distinction between expressions and statements. One syntactic difference from C is automatic semicolon insertion, in which the semicolons that terminate statements can be omitted.

**Dynamic**

Dynamic typing as in most scripting languages, types are associated with values, not with variables.

**Run-time evaluation**

JavaScript includes an eval function that can execute statements provided as strings at run-time.

**Functional**

First-class functions are first-class; they are objects themselves. As such, they have properties and methods, such as length and call(); and they can be assigned to variables, passed as arguments, returned by other functions, and manipulated like any other object. Any reference to a function allows it to be invoked using the () operator.

**Nested functions and closures**

"Inner" or "nested" functions are functions defined within another function. They are created each time the outer function is invoked. In addition to that, each created function forms a lexical closure: the lexical scope of the outer function, including any constants, local variables and argument values, become part of the internal state of each inner function object, even after execution of the outer function concludes.

**Prototype-based**

JavaScript uses prototypes instead of classes for inheritance. It is possible to simulate many class-based features with prototypes in JavaScript. Functions as object constructors Functions double as object constructors along with their typical role. Prefixing a function call with new creates a new object and calls that function with its local this keyword bound to that object for that invocation.

**Functions as methods**

Unlike many object-oriented languages, there is no distinction between a function definition and a method definition. Rather, the distinction occurs during function calling; a function can be called as a method. When a function is called as a method of an object, the function's local this keyword is bound to that object for that invocation.

**CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.

This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed.

**HTML**

Unlike many object-oriented languages, there is no distinction between a function definition and a method definition. Rather, the distinction occurs during function calling; a function can be called as a method. When a function is called as a method of an object, the function's local this keyword is bound to that object for that invocation.HTML5 is a markup language for structuring and presenting content for the World Wide Web, and is a core technology of the Internet originally proposed by Opera Software. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML4 as of 1997) and, as of May 2012, is still under development. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.).

HTML5 is intended to subsume not only HTML 4, but XHTML 1 and DOM Level 2 HTML as well. Following its immediate predecessors HTML 4.01 and XHTML 1.1, HTML5 is a response to the observation that the HTML and XHTML in common use on the World Wide Web are a mixture of features introduced by various specifications, along with those introduced by software products such as web browsers, those established by common practice, and the many syntax errors in existing web documents. It is also an attempt to define a single markup language that can be written in either HTML or XHTML syntax.

**3.2 Detailed life Cycle of the Project**

**3.3.1Modules**

**1. Account Module**

**2. User Module**

**3. Device Module**

**4. History Map Module**

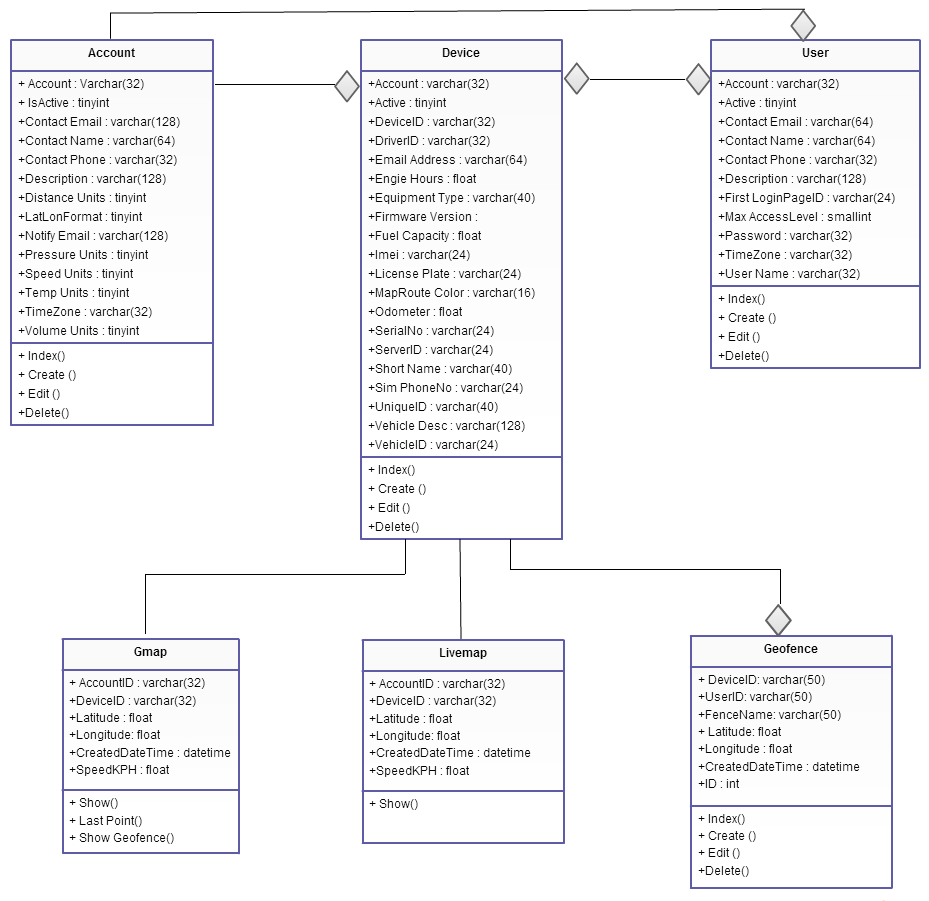
**5. Live Tracking Map Module**

**6. Geofence Module**

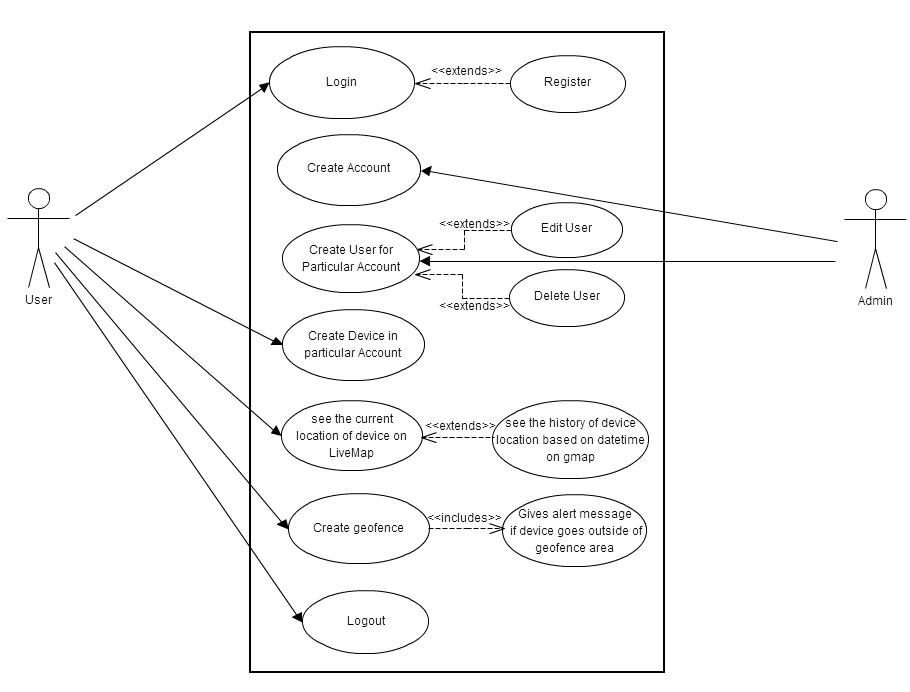
**3.3.2 Object Oriented Analysis & Design Diagrams**

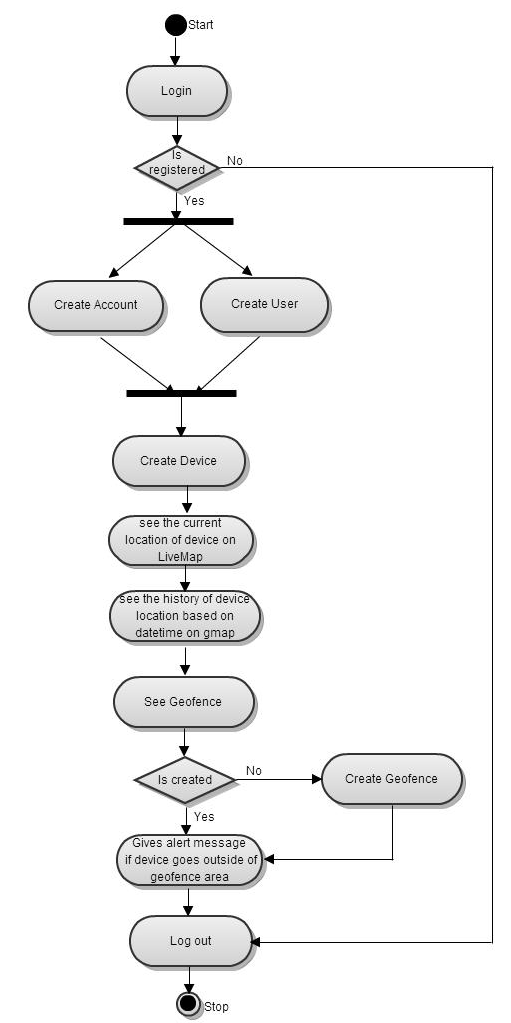
**3.3.2.1 “*UML Diagrams*”**

**Class Diagram**

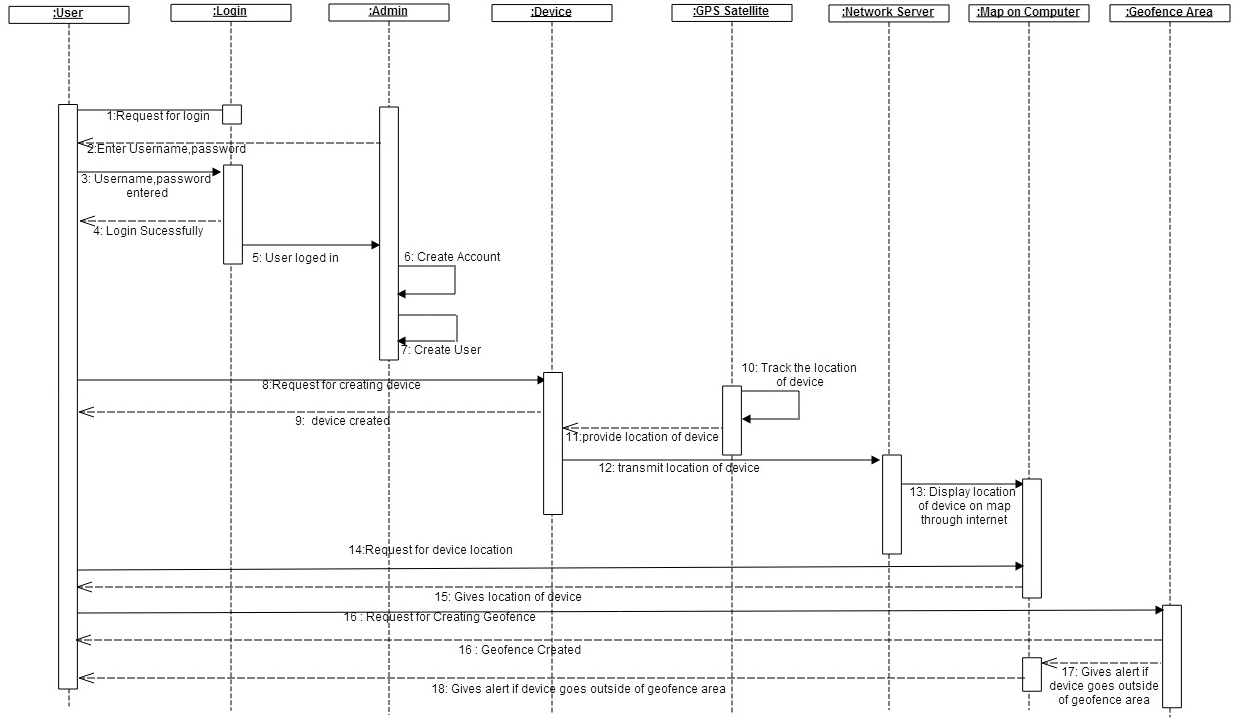
****

**Use Case Diagram**

****

**Activity Diagram**

**Sequence Diagram**

****

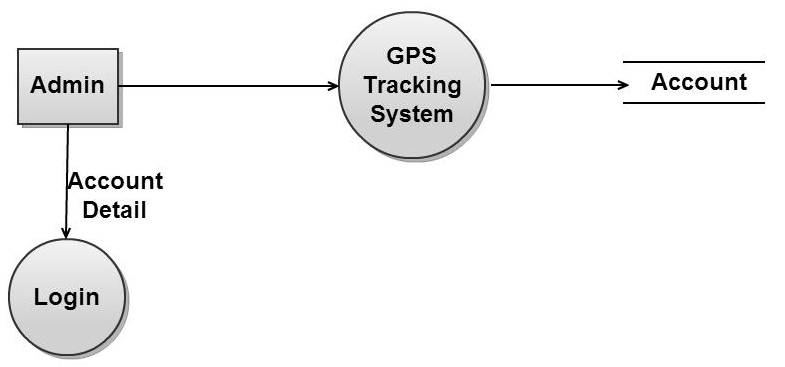
**DataFlow Diagrams:**

**Context Diagram**

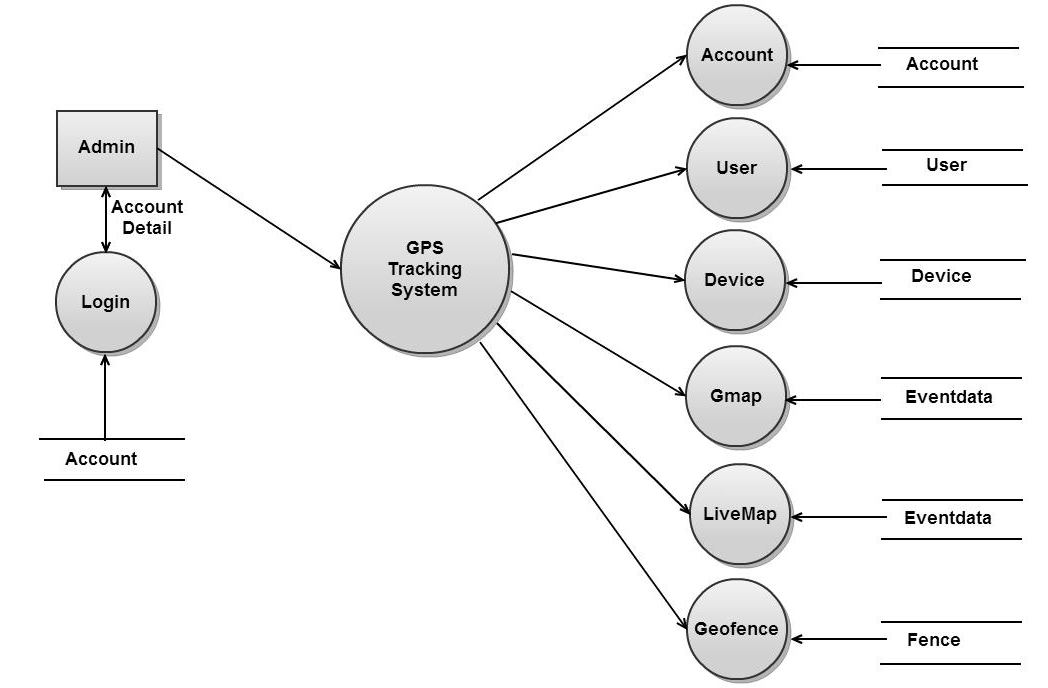
****

**For Account**

**DFD Level 0**

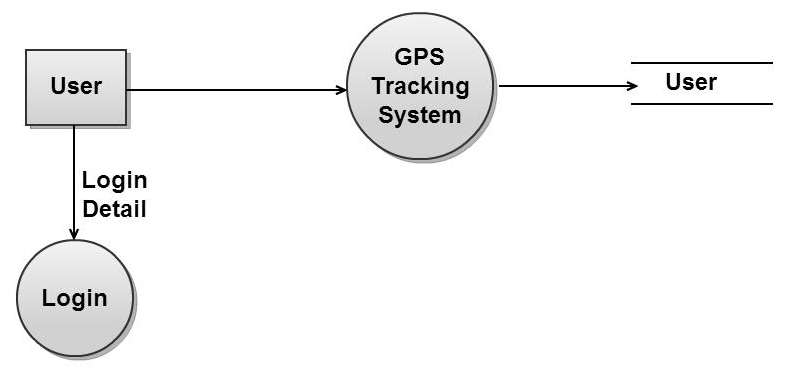
****

**DFD Level 1**

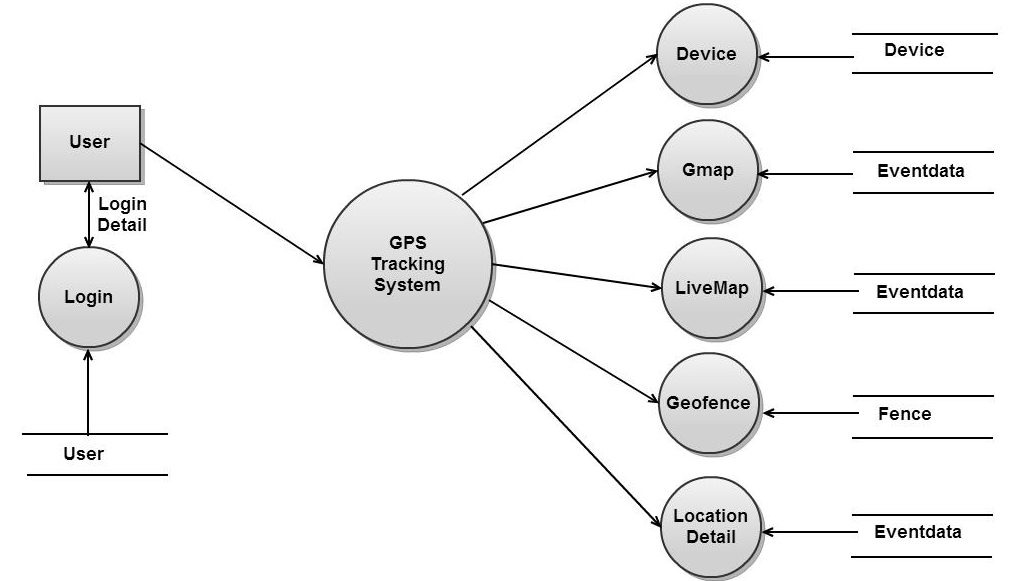
****

**For User**

**DFD Level 0**

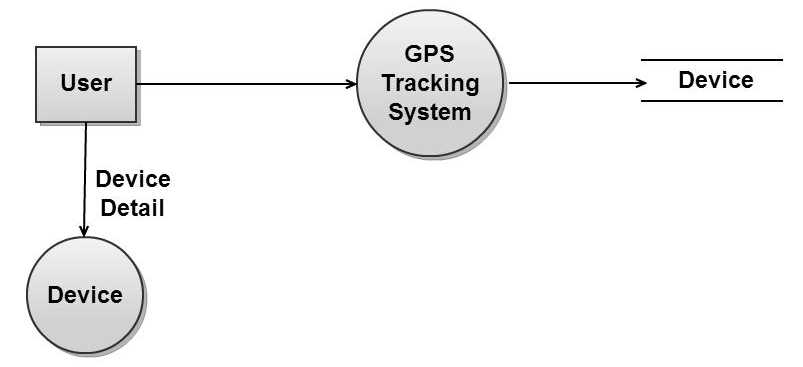
****

**DFD Level 1**

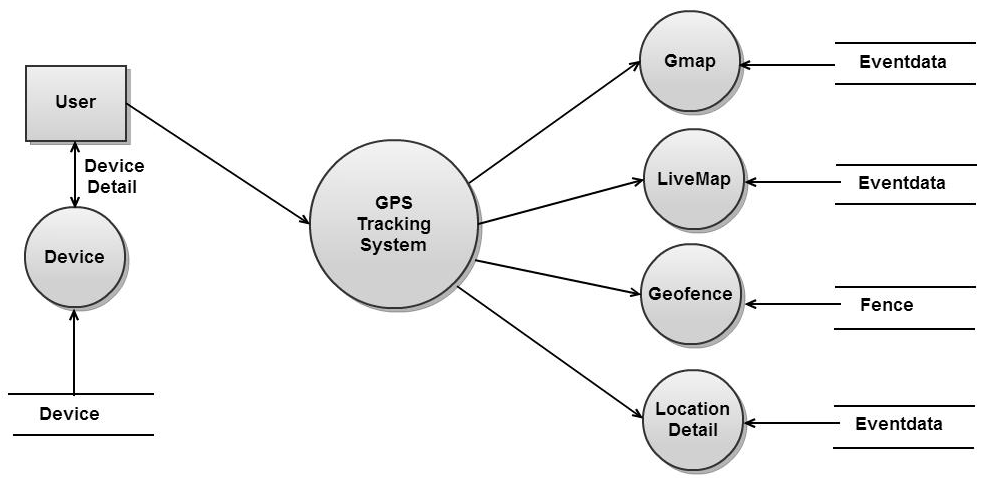
****

**For Device**

**DFD Level 0**

****

**DFD Level 1**

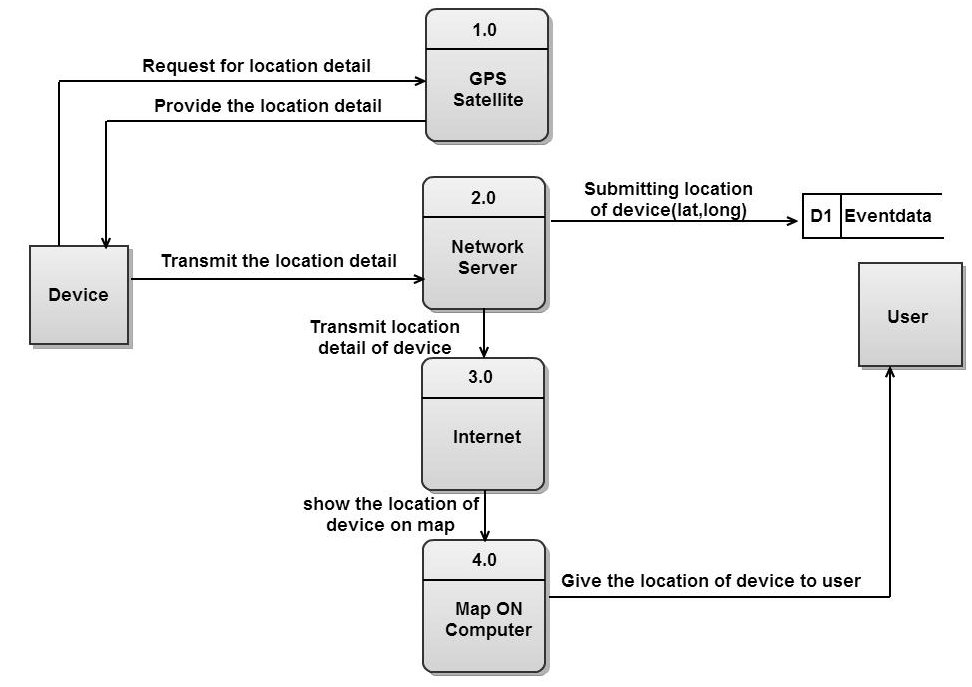
****

**For tracking the device**

**DFD Level 0**

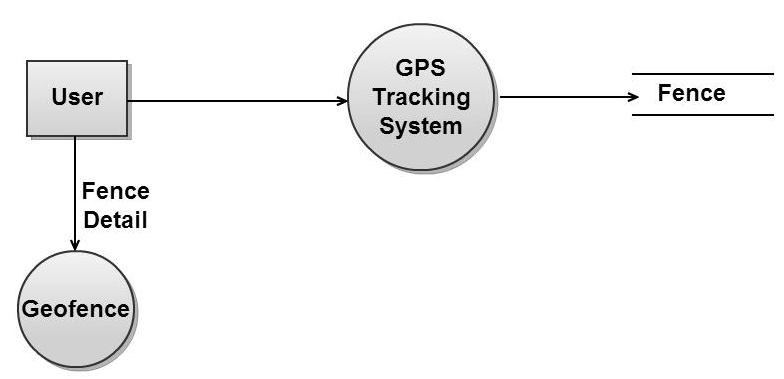
****

**DFD Level 1**

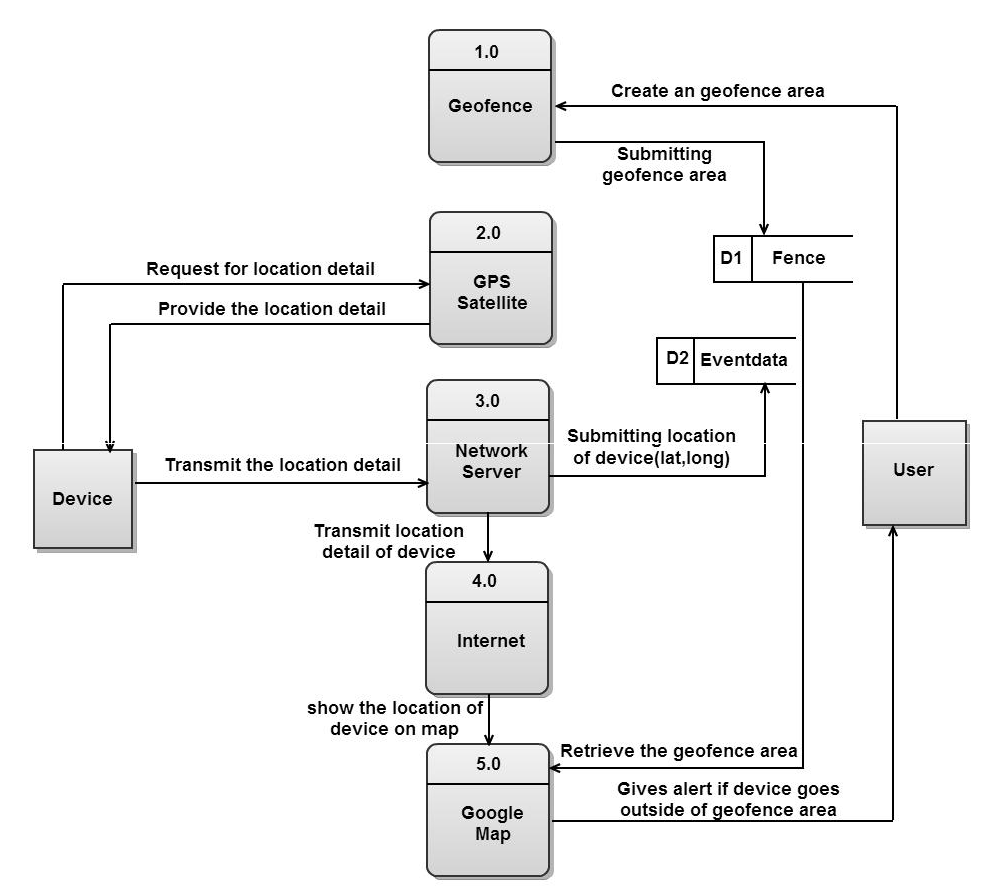
****

**For giving alert if user goes outside of geofence area**

**DFD Level 0**

****

**DFD Level 1**

****

**3.3.3 Database**

**3.3.3.1 Database Table**

After carefully understanding the requirements of the client the entire data storage requirements are divided into tables. The below tables are normalized to avoid any anomalies during the course of data entry.

**3.3.3.1.1Account**

|  |  |  |
| --- | --- | --- |
| **Fields** | **Datatype** | **Constraints** |
| accountID | varchar(32) | Not Null |
| accountType | Smallint | Null |
| notifyEmail | varchar(128) | Null |
| speedUnits | varchar(128) | Null |
| distanceUnits | Tinyint | Null |
| volumeUnits | Tinyint | Null |
| pressureUnits | Tinyint | Null |
| economyUnits | Tinyint | Null |
| temperatureUnits | Tinyint | Null |
| accountID | varchar(32) | Null |
| accountType | Smallint | Null |
| notifyEmail | varchar(128) | Null |
| speedUnits | Tinyint | Null |
| distanceUnits | tinyint | Null |
| volumeUnits | tinyint | Null |
| pressureUnits | tinyint | Null |
| economyUnits | tinyint | Null |
| temperatureUnits | tinyint | Null |
| latLonFormat | tinyint | Null |
| geocoderMode | tinyint | Null |
| privateLabelName | varchar(32) | Null |
| isBorderCrossing | tinyint | Null |
| retainedEventAge | bigint | Null |
| maximumDevices | bigint | Null |
| totalPingCount | smallint | Null |
| maxPingCount | smallint | Null |
| autoAddDevices | tinyint | Null |
| dcsPropertiesID | varchar(32) | Null |
| smsEnabled | tinyint | Null |
| smsProperties | varchar(200) | Null |
| emailProperties | varchar(250) | Null |
| expirationTime | bigint | Null |
| defaultUser | varchar(32) | Null |
| password | varchar(32) | Null |
| contactName | varchar(64) | Null |
| contactPhone | varchar(32) | Null |
| contactEmail | varchar(128) | Null |
| timeZone | varchar(32) | Null |
| passwdChangeTime | bigint | Null |
| passwdQueryTime | bigint | Null |
| lastLoginTime | bigint | Null |
| isActive | tinyint | Null |
| displayName | varchar(40) | Null |
| description | varchar(128) | Null |
| notes | Text | Null |
| lastUpdateTime | bigint | Null |
| creationTime | bigint | Null |

**3.3.3.1.2 User**

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Constraints** |
| accountID | varchar(32) | Not Null |
| userID | varchar(32) | Not Null |
| userType | Smallint | Null |
| roleID | varchar(32) | Null |
| Password | varchar(32) | Null |
| Gender | Tinyint | Null |
| notifyEmail | varchar(128) | Null |
| contactName | varchar(64) | Null |
| contactPhone | varchar(32) | Null |
| contactEmail | varchar(64) | Null |
| timeZone | varchar(32) | Null |
| firstLoginPageID | varchar(24) | Null |
| preferredDeviceID | varchar(32) | Null |
| maxAccessLevel | Smallint | Null |
| passwdChangeTime | Bigint | Null |
| passwdQueryTime | Bigint | Null |
| lastLoginTime | Bigint | Null |
| isActive | Tinyint | Null |
| displayName | varchar(40) | Null |
| Description | varchar(128) | Null |
| Notes | Text | Null |
| lastUpdateTime | Bigint | Null |
| creationTime | Bigint | Null |

**3.3.3.1.3 Device**

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Constraints** |
| accountID | varchar(32) | Not Null |
| deviceID | varchar(32) | Not Null |
| groupID | varchar(32) | Null |
| equipmentType | varchar(40) | Null |
| vehicleID | varchar(24) | Null |
| licensePlate | varchar(24) | Null |
| driverID | varchar(32) | Null |
| fuelCapacity | Float | Null |
| fuelEconomy | Float | Null |
| speedLimitKPH | Float | Null |
| planDistanceKM | Float | Null |
| expirationTime | Bigint | Null |
| uniqueID | varchar(40) | Null |
| deviceCode | varchar(24) | Null |
| deviceType | varchar(24) | Null |
| pushpinID | varchar(32) | Null |
| displayColor | varchar(16) | Null |
| serialNumber | varchar(24) | Null |
| simPhoneNumber | varchar(24) | Null |
| simID | varchar(24) | Null |
| smsEmail | varchar(64) | Null |
| imeiNumber | varchar(24) | Null |
| dataKey | Text | Null |
| ignitionIndex | Smallint | Null |
| codeVersion | varchar(32) | Null |
| featureSet | varchar(64) | Null |
| ipAddressValid | varchar(128) | Null |
| lastTotalConnectTime | Bigint | Null |
| lastDuplexConnectTime | Bigint | Null |
| pendingPingCommand | Text | Null |
| lastPingTime | Bigint | Null |
| totalPingCount | Smallint | Null |
| maxPingCount | Smallint | Null |
| expectAck | Tinyint | Null |
| lastAckCommand | Text | Null |
| lastAckTime | Bigint | Null |
| dcsPropertiesID | varchar(32) | Null |
| dcsConfigMask | Bigint | Null |
| supportsDMTP | Tinyint | Null |
| supportedEncodings | Tinyint | Null |
| unitLimitInterval | Smallint | Null |
| maxAllowedEvents | Smallint | Null |
| totalProfileMask | Image | Null |
| totalMaxConn | Smallint | Null |
| totalMaxConnPerMin | Smallint | Null |
| duplexProfileMask | Image | Null |
| duplexMaxConn | Smallint | Null |
| duplexMaxConnPerMin | Smallint | Null |
| lastTcpSessionID | varchar(32) | Null |
| ipAddressCurrent | varchar(32) | Null |
| remotePortCurrent | Int | Null |
| listenPortCurrent | Smallint | Null |
| lastInputState | Bigint | Null |
| lastBatteryLevel | Float | Null |
| lastFuelLevel | Float | Null |
| lastFuelTotal | Float | Null |
| lastOilLevel | Float | Null |
| odometerOffsetKM | Float | Null |
| lastEngineHours | Float | Null |
| engineHoursOffset | Float | Null |
| lastIgnitionOnTime | Bigint | Null |
| lastIgnitionOffTime | Bigint | Null |
| lastIgnitionHours | Float | Null |
| lastStopTime | Bigint | Null |
| lastStartTime | Bigint | Null |
| lastMalfunctionLamp | Tinyint | Null |
| lastFaultCode | varchar(96) | Null |
| isActive | Tinyint | Null |
| displayName | varchar(40) | Null |
| Description | varchar(128) | Null |
| Notes | Text | Null |
| lastUpdateTime | Bigint | Null |
| creationTime | Bigint | Null |

**3.3.3.1.4 Eventdata**

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Constraints** |
| accounted | varchar(32) | Not Null |
| deviceID | varchar(32) | Not Null |
| Timestamp | Bigint | Not Null |
| statusCode | Bigint | Not Null |
| Latitude | Float | Null |
| Longitude | Float | Null |
| gpsAge | Bigint | Null |
| speedKPH | float | Null |
| Address | varchar(90) | Null |
| dataSource | varchar(32) | Null |
| rawData | text | Null |
| distanceKM | float | Null |
| odometerKM | float | Null |
| odometerOffsetKM | float | Null |
| geozoneIndex | bigint | Null |
| geozoneID | varchar(32) | Null |
| creationTime | bigint | Null |
| CreatedDateTime | datetime | Null |

**3.3.3.1.5 Fence**

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Constraints** |
| DeviceID | varchar(50) | Null |
| UserId | varchar(50) | Null |
| FenceName | varchar(50) | Null |
| Latitude | float | Null |
| Longitude | float | Null |
| CreatedDateTime | datetime | Null |
| ID | int | Not Null |

**4. Testing Report**

**4.1Methodologies used for Testing**

Software testing is an integral part of the software development life cycle (SDLC).

Testing a piece of code effectively and efficiently is equally important, if not more, to writing it. So what is software testing? Well, for those who are new to software testing and quality assurance, here are few useful facts. Software testing is nothing but subjecting a piece of code to both, controlled and uncontrolled operating conditions, in an attempt to observe the output and examining whether it is in accordance with certain pre-specified conditions.

Different sets of test cases and testing strategies are prepared, all of which are aimed at achieving one common goal i.e. removing bugs and errors from the code and making the software error-free and capable of providing accurate and optimum outputs. There are different types of software testing techniques and methodologies. A software testing methodology is different from a software testing technique. We'll take a look at few of the software testing methodologies in the latter part of this article.

**Software Testing Methodology**

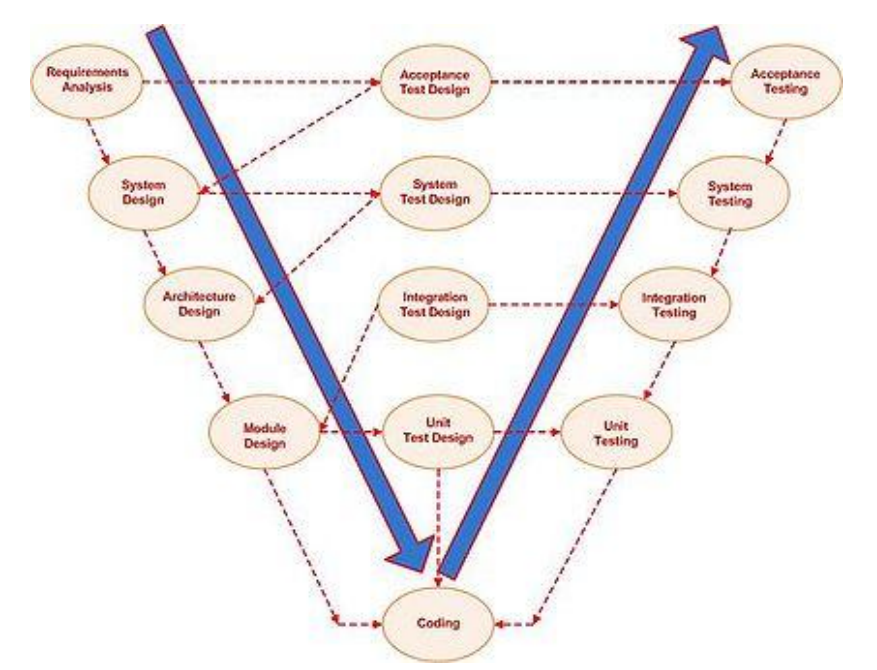
**VModel**

The issues seen in the traditional waterfall model gave birth to the V-Model;

It wasdeveloped with an intention to address some of the problems found in waterfall model. As you can see that in waterfall model defects were found very late in the development life cycle because testing was not involved until the end of the project.

In V-Model testing begins as early as possible in the project life cycle, it is always a good practice to involve testers at earlier phases of product life cycle. There are varieties of test activities that need to be carried out before end of the coding phase. These activities should be carried out in parallel to the development activities so that testers can produce a set of test deliverables.

The V-Model illustrates that testing activities (Verification and Validation) can be integrated into each phase of the product life cycle. Validation part of testing is integrated in the earlier phases of the life cycle which includes reviewing end user requirements, design documents etc. There are variants of V-Model however we will take a common type of V-model example.

****

The V-model generally has four test levels.

* Component Testing
* Integration Testing
* System Testing
* Acceptance Testing

**Advantages**

* Each phase has specific deliverables.
* Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.
* Time concern in comparison with the waterfall model is low or even we can say 50% less.
* Works well for small projects where requirements are easily understood.
* Utility of the resources is high.

**Disadvantages**

* Very rigid, like the waterfall model.
* Little flexibility and adjusting scope is difficult and expensive.
* Software is developed during the implementation phase, so no early prototypes of the software are produced.
* Model doesn’t provide a clear path for problems found during testing phases

**4.2 Types of Testing**

**4.2.1 White Box Testing**

White box testing strategy deals with the internal logic and structure of the code. It is also called glass, structural, open or clear box testing. The tests that are written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code, etc. In order to implement white box testing, the tester has to deal with the code, and hence is required to possess knowledge of coding and logic i.e., internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning. In other words, it is imperative that the tester has 'structural' knowledge about how the system has been implemented. Not only the code, but even the data flow and control flow have to be assessed. The areas of the code that are tested using white box testing are:

1. Code Coverage

2. Segment Coverage

3. Branch Coverage

4. Condition Coverage

5. Loop Coverage

6. Path Testing

7. Data Flow Coverage

There are three aspects of the code, which are validated in white box testing, namely

1. If the software has been designed as per the original design of the software.

2. If security measures have been implemented into the software and it is robust.

3. Find out vulnerabilities in the said software.

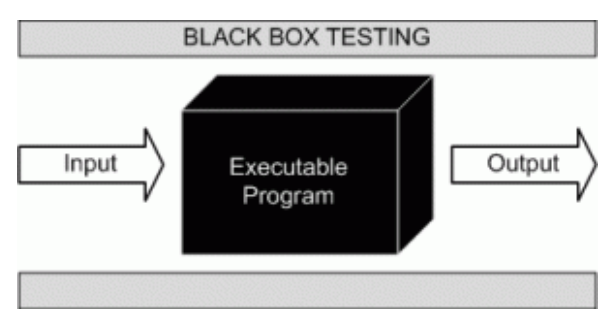
**Advantages ofWhite Box Testing**

* + As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
  + Yet another advantage of white box testing is that it helps in optimizing the code.
  + It helps in removing the extra lines of code, which can introduce defects in the code.

**Disadvantages of White Box Testing**

* + As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, and this, in turn, increases the cost of the software.
  + It is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.

**4.2.2 Black Box Testing**

****

Black box testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code, etc. As the name "black box" suggests, not knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. Black box testing is sometimes also called "Opaque Testing",

"Functional/Behavioral Testing" and "Closed Box Testing".

**These testing types are again divided in two groups:**

1. Testing in which the user plays a role of tester

2. User is not required

**4.3 Test Cases**

While testing a Website development application we need to consider following cases:

**4.3.1 Functionality Testing**

In functionality testing we have conducted following tests

* + **Web Forms**

i. Field validation

ii. Error message for wrong input

iii. Optional and Mandatory fields

iv. Live Tracking Map and History Map

v. Geofence alert

**4.3.2 Usability Testing**

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

* + - Ease of learning
    - Navigation
    - Subjective user satisfaction
    - General appearance

**Database**

Testing will be done on the database integrity

**4.3.3 Performance Testing**

Performance testing can be done to check the performance of the system in otheroperating environment. Here we have check how it is operated in other versions ofthe Internet Explorer, Google Chrome and Firefox and other configurations of the system

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case No** | **Purpose** | **Step to be Executed** | **Expected Result** | **Actual Result** | **Pass/ Fail** |
| 1. | Validation for | 1) Enter the valid data in email id Textbox.  2) Enter the valid data in Password textbox.  3)Click on Login Button | User Should be entering in Application | User is entering in Application | Pass |
| 2. | Validation for ‘Password’ field | 1) Enter the Invalid data in email id textbox.  2) Enter the valid data in Password textbox.  3)Click on Login Button | User Should be entering in Application | User is not entering  Application | Pass |
| **3.** | Validation for ‘Invalid Email Id’ field | 1)Enter the Invalid data in email id text box  2) Enter the valid data in Password textbox.  3) Click on Login Button | User Should be entering in Application | User is entering in Application | Pass |
| **4.** | Validation for ‘Invalid Password’ field | 1) Enter the valid data in email id text box.  2) Enter the Invalid data in Password textbox.  3)Click on Login Button | User Should be entering in Application | User is not entering in Application | Pass |
| **5.** | Creating new user | 1)Enter valid information in all textboxes  2)Click on Save button | User should be created | User should not be created | Fail |
| **6.** | Showing last point of device | 1)Click on Last Point checkbox | It should display last point of device on map | Last point should be displayed | Pass |
| **7.** | Geofence alert | 1)Click on Show Geofence checkbox | It should give alert if device goes outside of geofence area | Gives alert if device goes outside of area | Pass |
| **8.** | Show infowindow for each marker | 1)mousehover on each marker | It should display infowindow on marker | Infowindow should be displayed | Pass |

**4.4 Test Reports**

**Test Report 1**

**Bug Name:** Application crash on clicking the Save button while creating new User

**Area Path: Master 🡪 User🡪Create🡪User (View) 🡪Create.cshtml**

**Priority:** High

**Assigned By:** Deepali Gavhale

**Reported By:** Shital Kadam

**Reported On:** 3th feb, 2013

**Reason:** Defect

**Description:** Application crash on Clicking the save button while creating new user, hence unable to create a new user in the application.

**Test Report 2**

**Bug Name:**Not shows Last Point on Google Map

**Area Path:Google Map🡪Last Point**🡪**DeviceEvent (View) 🡪gmap.cshtml**

**Priority:**High

**Assigned:**Amol Gadhave

**Reported By:**Deepali Gavhale

**Reported on:**9th mar, 2013

**Reason:** Defect

**Description:**while click on last point checkbox map doesn’t show any point on it.

**TEST REPORT: 3**

**Bug Name:**Does not give alert when user goes outside of geofence area

**Area Path: Google Map 🡪Show Geofence**🡪 **DeviceEvent(View) 🡪gmap.cshtml**

**Priority:** High

**Assigned to:** Amol Gadhave

**Reported By:**Shital Kadam

**Reported On:**  15th Apr, 2013

**Reason:** Defect

**Description:** While clicking on outside of geofence area it does not give alert message when user goes outside of that area.

**TEST REPORT: 4**

**Bug Name:**Does not shows the infowindow for the marker on Live Tracking Map

**Area Path: LiveMap 🡪Show** 🡪 **DeviceEvent(View) 🡪Livemap.cshtml**

**Priority:** High

**Assigned to:** Shital Kadam

**Reported By:**Deepali Gavhale

**Reported On:** 5thmay, 2013

**Reason:** Defect

**Description:** While moving cursor over marker it does not shows the infowindow for that marker but that infowindows always displayed on first marker (i.e. first point) of the map.

**TEST REPORT: 5**

**Bug Name:**Does not show the error message on creating Device while error occurred.

**Area Path: Master 🡪 Device🡪Create🡪 Device (View) 🡪Create.cshtml**

**Priority:** High

**Assigned to:** Shital Kadam

**Reported By:** Deepali Gavhale

**Reported On:** 15th may, 2013

**Reason:** Defect

**Description:** While creating new device with same deviceID it does not allow to create device and also doesn’t give the error message.

**5. System Implementation**

**5.1 Hardware requirement at Client Side**

* + Hard Disk :Free Space 1GB and more
  + Processor speed 250 MHz and more

The processor does all the processing for the contents of the program and when the clock speed increases so does the processing speed.

* + RAM 256 and more.

The memory selection is done and preferred for higher memory because the program before running is flushed into memory buffer of computer, then it is executed. More the memory, more will be the speed and hence less time for execution.

* + Sim Card: Standard Gsm sim
  + Tk102 tracing device

**5.2 Software requirement at Client Side**

Software plays an important role in any project development. One should understand that which software he/she should use to develop the project. Window XP was used as the operating system.

The application has been developed using:

* Internet Connection on any Computer.

**5.3 Testing done as End User**

An alpha test is a preliminary software field test carried out by a team of users in order to find bugs that were not found previously through other tests. The main purpose of alpha testing is to refine the software product by finding (and fixing) the bugs that were not discovered through previous tests.

The team that conducts the alpha test is often an independent test team, perhaps made up of potential users/customers. Alpha testing involves simulating a real user environment by carrying out tasks and operations that the actual users might perform. Once software passes the alpha test, it is considered for the next phase of testing called beta testing. The meaning of alpha also can differ based on whether the project is custom software one for a client. In this case, alpha testing implies an initial meeting between software vendor and client to ensure that the client's requirements are properly met by the developer in terms of the performance, functionality and durability of the software program. Compare this versus the context of a web application, where alpha testing can be interpreted as an online application that isn't completely ready for usage, but that has been opened up to get some initial feedback. Commonly an alpha might allow power users to get their first look at the system via a private invitation. The meaning of alpha also can differ based on whether the project is custom software one for a client. In this case, alpha testing implies an initial meeting between software vendor and client to ensure that the client's requirements are properly met by the developer in terms of the performance, functionality and durability of the software program

**6. System Maintenance and Evaluation**

**6.1 Maintenance**

The term maintenance, when accompanied to software, assumes a meaning profoundly different from the meaning it assumes in any other engineeringdiscipline. In fact, many engineering disciplines intend maintenance as the process of keeping something in working order, in repair. The key concept is the deterioration of an engineering artifact due to the use and the passing of time; the aim of maintenance is therefore to keep the artifact’s functionality in line with that defined and registered at the time of release. Of course, this view of maintenance does not apply to software, as software does not deteriorate with the use and the passing of time.

A predominant proportion of changes are to meet ever changing user needs. This is captured by the first law of Lehman: “A program that is used in a real world environment necessarily must change or become progressively less useful in that environment”. Significant changes also derive from the need to adapt software to interact with external entities, including people, organizations, and artificial systems.

Software maintenance is a very broad activity often defined as including all work made on a software system after it becomes operational. This covers the correction of errors, the enhancement, deletion and addition of capabilities, the adaptation to changes in data requirements and operation environments, the improvement ofperformance, usability, or any other quality attributes. The IEEE definition is as follows:

“Software maintenance is the process of modifying a software system orComponent after delivery to correct faults, improves performances or other attributes, or adapt to a changed environment.”This definition reflects the common view that software maintenance is a post-deliveryactivity: it starts when a system is released to the customer or user and encompasses all activities that keep the system operational and meet the user’s needs Software maintenance activities are classified into following four classes:

1. **Adaptive -** modifying the system to cope with changes in the software environment
2. **Perfective** - implementing new or changed user requirements which concernfunctional enhancements to the software

**3. Corrective -** diagnosing and fixing errors, possibly ones found by users

**4. Preventative -** increasing software maintainability or reliability to prevent problems in the future

Application performance has top most priority these days, as organization face tremendous customer demands and competitions in the entire business sector. Application maintenance is a challenging job for any software solution. Over a period of usage, the application may run out of performance due to unexpected problems .Based on user experience there could be suggestions for changes on the functionalities or mode of operation. Also for enterprise applications there may require emergency updates due to unexpected business requirements. All these come under software application maintenance namely preventive maintenance and scheduled maintenance. Preventive maintenance is performed based on the analysis of modules or product before it fails for effective operations.

There are many underlying factors that need to be taken care of while providing an application. The constant support and maintenance services, of course, are top priority.

Defect free system:

We provide 100% error free application to our clients all over the world.

We understand the requirements at all user levels of our clients.

Technical and functional expertise:

We have the rich technology blended technical and functional expertise to maintain the scalability, stability and strategy of business process to keep them up to date.

**Cost effective:**

The software models that we have delivered ensures, cost effectiveness in all ways.

**Flexibility to change:**

The application maintenance services, sometimes requires the system to undergo sufficient changes.

**6.2 Evaluation**

In the present day market situation, there are several alternatives available when a customer wants to purchase a product or adopt a software system that meets the customer’s requirements. Software technologies have been evolving rapidly and for a given set of functional and non-functionalrequirements there usually exist several competing software products. The present day users are faced with a challenging situation that requires evaluation and selection of a suitable software product that satisfies the users’ operational and business needs. Unfortunately this evaluation is usually carried on in anad-hoc and informal way and with various degree of success.

Software evaluation methodologies can be divided into two categories.

The First category is used to evaluate software development methodology or processes such as those used to evaluate variousagent-based development methodologies.

The Second category is used to evaluate software products such as Cost evaluation and selection methodologies.

System Evaluation recognizes and integrates the important features of other frameworks, overcomes any obvious deficiencies, and adopts new features that generalize and extend its usability.

The evaluation stage of the software development process requires the client and developer to review the software.

Broadly speaking, they evaluate against the following questions:

1. Does this software meet the user requirements?

2. Is it fit for purpose?

**Evaluation - Criteria:**

To answer these questions, the original aims of the software must be evaluated against the

Following criteria.

1. Robustness

2. Reliability

3. Portability

4. Efficiency

5. Maintainability

**Evaluation - When:**

The software should be evaluated by client and developer at all stages in the process, not just when the software is complete.

**7. User / Operational Manual**

1) This is a guide for the user of this system.

2) This contains all the relevant screen display that will help the user to understand the system.

3) There are proper uses of comments that will help the user to understand in case of error.

4) Proper validation is done and message box are prompted to the user before entering data into the database.

5) The comments describe what exactly the code block does.

**7.1 Security Aspects, Access Rights**

Does the software protect itself and its data against unauthorized access and use? Does it allow its operator to enforce security policies? Are appropriate security mechanisms in place? Are those security mechanisms implemented correctly? Can the software withstand attacks that must be expected in its intended environment? Is the software free of errors that would make it possible to circumvent its security mechanisms? Does the architecture limit the impact of yet unknown errors? Security testing is any develop system is about finding loops and weaknesses of the system

There are two types of the user who can access the GPS Tracking System, they are:

* Administrator
* Normal User

Administrator has a right to access all the web forms of the project.

Normal user also has the permission to access the web forms of the Details Menu and the maps which shows the history n current location of device (i.e. History Map and Live Tracking Map)& also has permission to draw the geofence area which gives alert if device goes outside of that area.

**User's perspective:**

In addition to the technical qualities of software, the end user's experience also determines the quality of software. This aspect of software quality is called usability. It is hard to quantify the usability of a given software product. Some important questions to be asked are:

* Is the user interface intuitive?
* Is it easy to perform easy operations?
* Is it feasible to perform difficult operations?
* Does the software give sensible error messages?
* Do widgets behave as expected?
* Is the software well documented?
* Is the user interface self-explanatory/ self-documenting?
* Is the user interface responsive or too slow?

Also, the availability of (free or paid) support may determine the usability of the software.

**7.2 Back Up’s**

In information technology, backup refers to making copies of data so thatthese additional copies may be used to restore the original after a data loss event. These additional copies are typically called "backups." Backups are useful primarily for two purposes. The first is to restore a state following a disaster (called disaster recovery). The second is to restore small numbers of files after they have been accidentally deleted or corrupted. Backups aretypically that last line of defense against data loss, and consequently the least granular and the least convenient to use.

Since a backup system contains at least one copy of all data worth saving, the data storage requirements are considerable. Organizing this storage space and managing the backup process is a complicated undertaking. There are also many different ways in which these devices can be arranged to provide geographic redundancy, data security, and portability. Before data are sent to their storage location, they are selected, extracted, and manipulated.

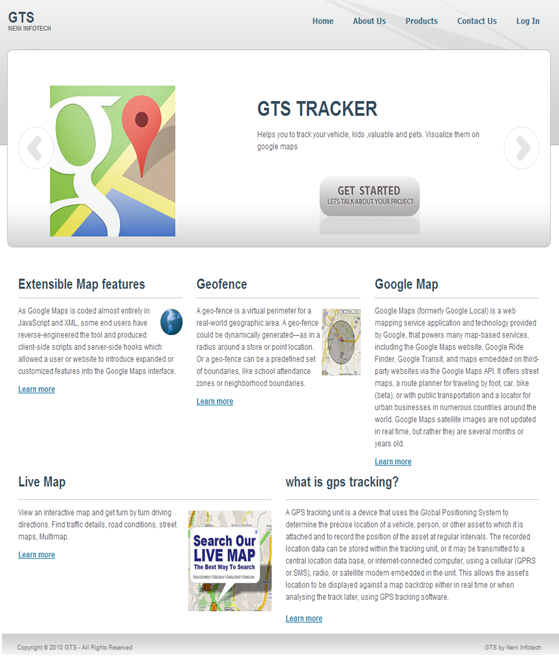
Many organizations and individuals try to have confidence that the process is working as expected and work to define measurements and validation techniques. It is also important to recognize the limitations and human factors involved in any backup scheme.

Due to a considerable overlap in technology, backups and backup systems are frequently confused with archives and fault-tolerant systems. Backups differ from archives in the sense that archives are the primary copy of data and backups are a secondary copy of data. Backup systems differ from fault-tolerant systems in the sense that backup systems assume that a fault will cause a data loss event and fault-tolerant systems assume a fault will not.

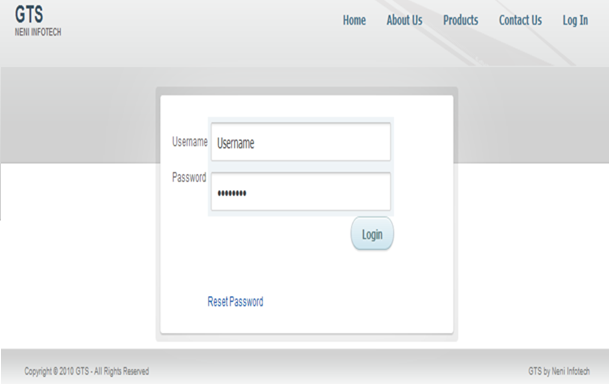
* + Database Backups is taken by Admin on weekly basis.
  + Backups help in Data Integrity.

**7.3 Screen Shots**

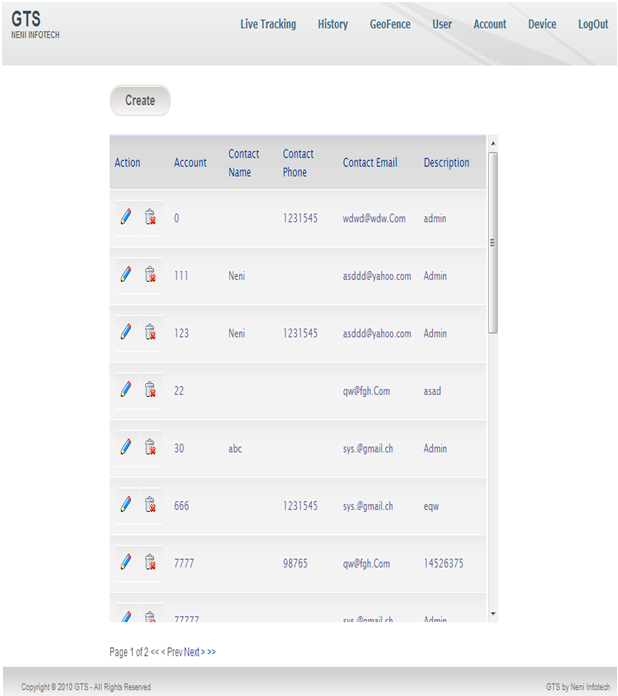
**7.3.1 Home Page**

****

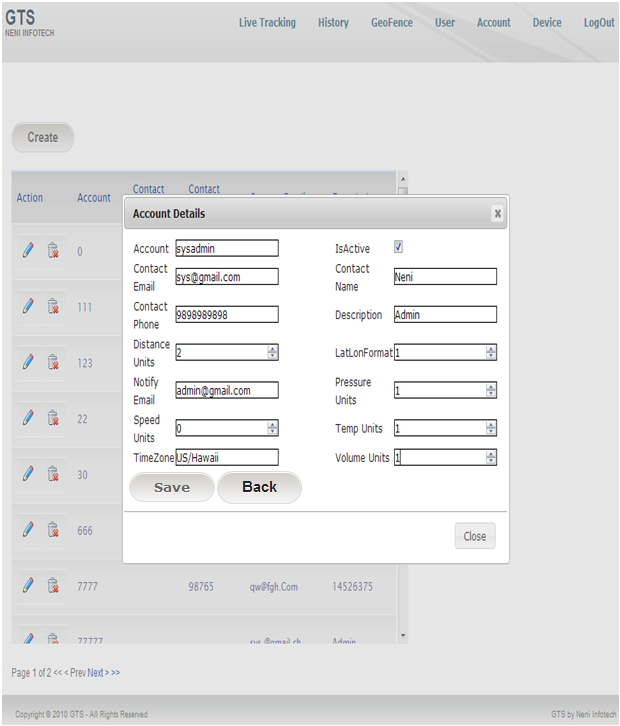
**7.3.2 Login Screen**

****

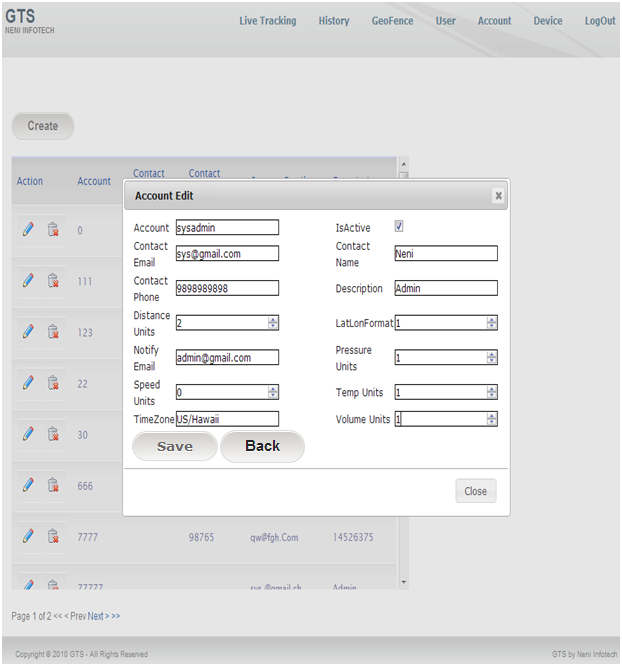
**7.3.3 Account-Index**

****

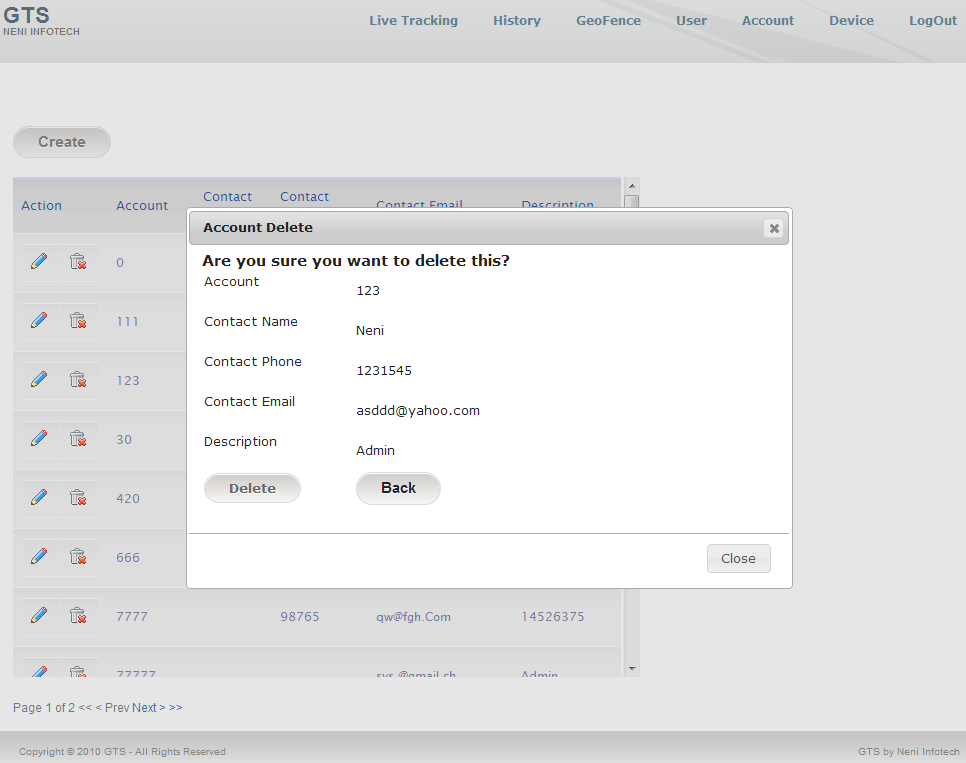
**7.3.4 Account-Create**

****

**7.3.5 Account-Edit**

****

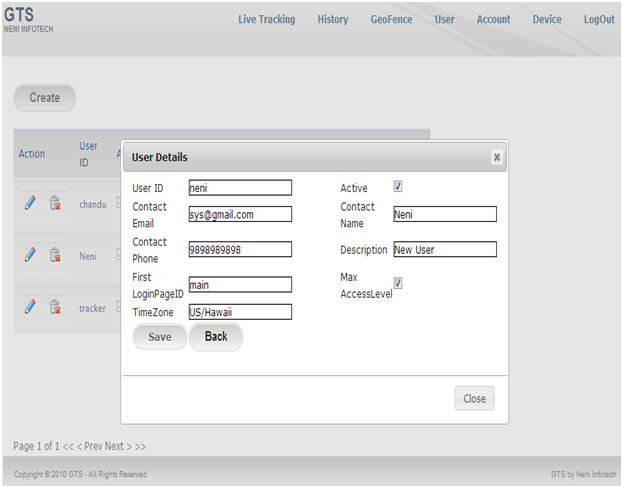
**7.3.6 Account-Delete**

****

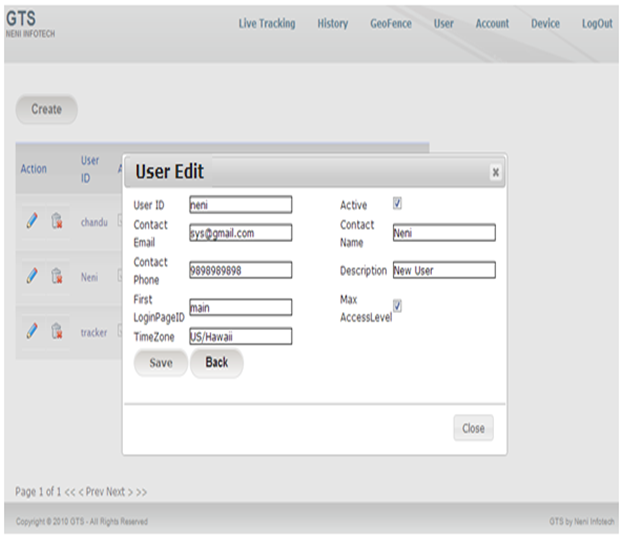
**7.3.7 User-Index**

****

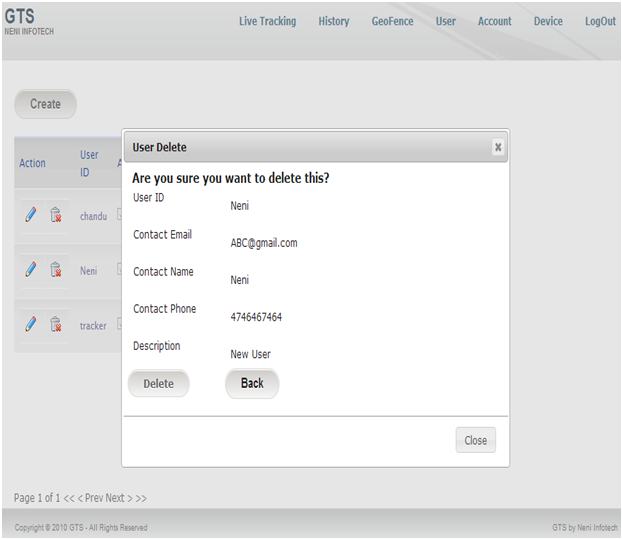
**7.3.8 User-Create**

****

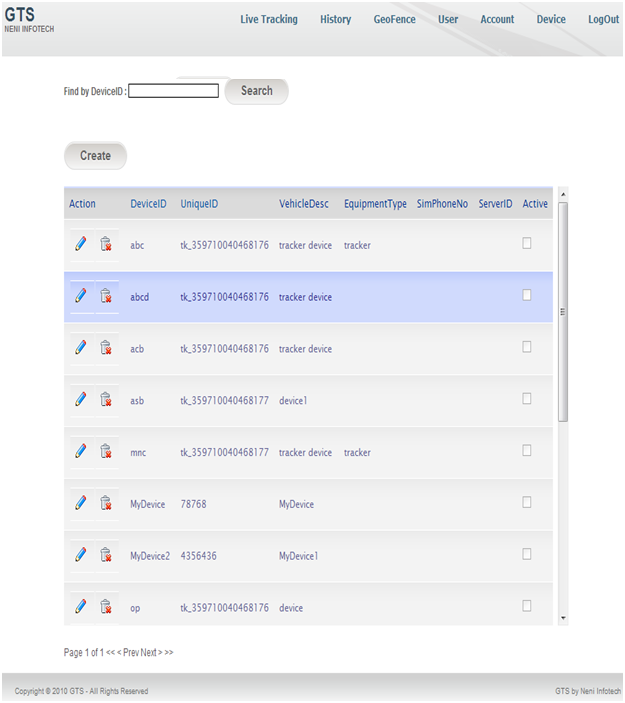
**7.3.9 User-Edit**

****

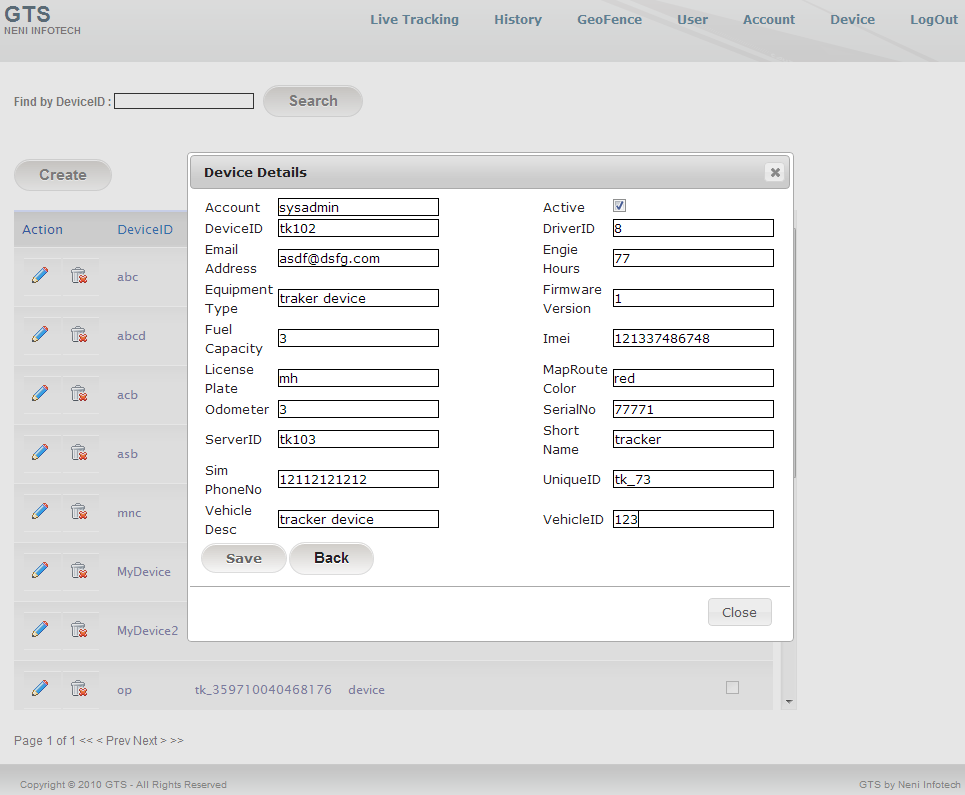
**7.3.9 User-Delete**

****

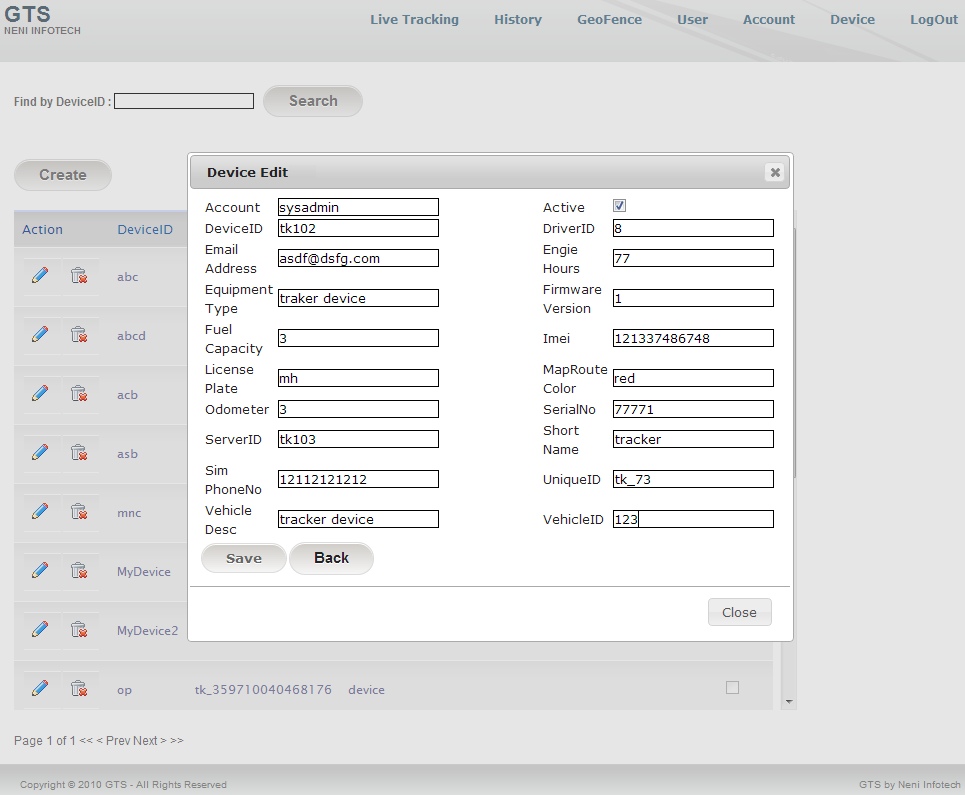
**7.3.10 Device-Index**

****

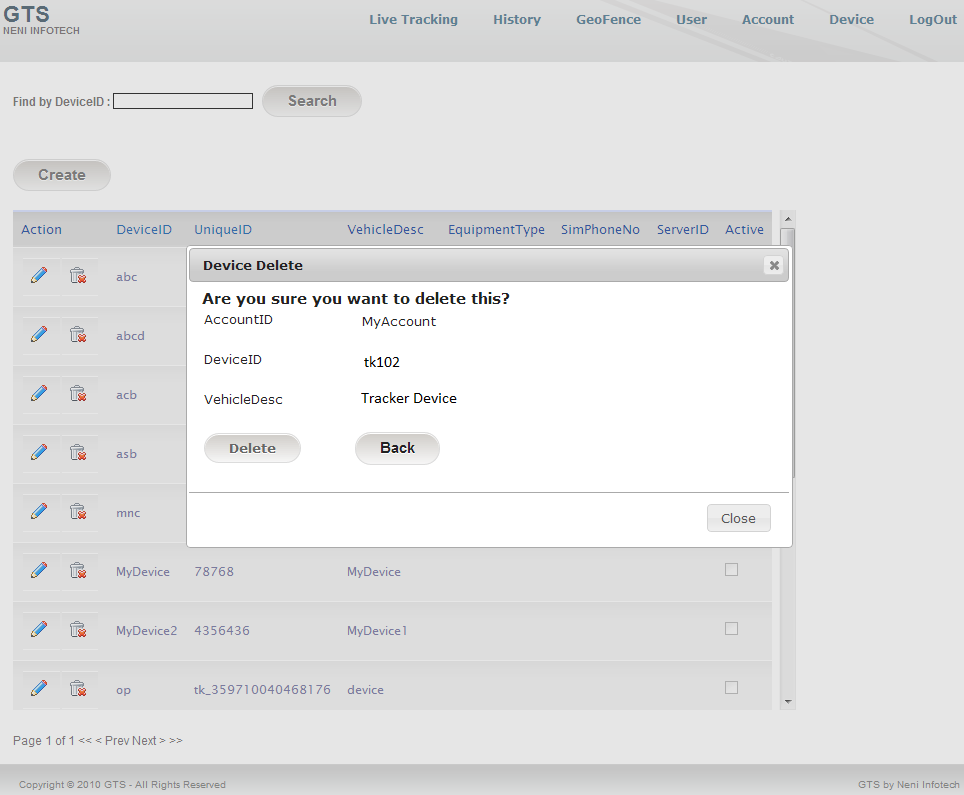
**7.3.11 Device-Create**

****

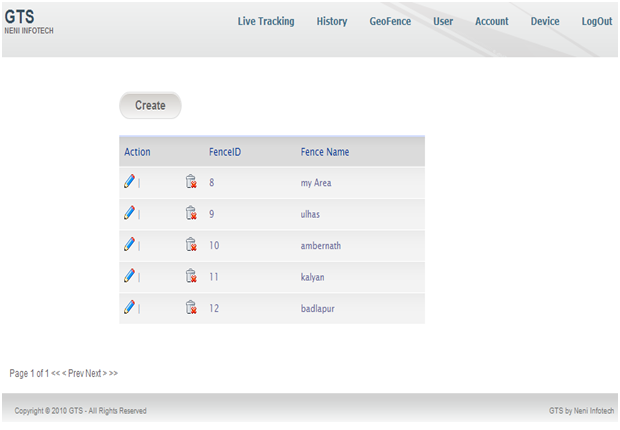
**7.3.12 Device-Edit**

****

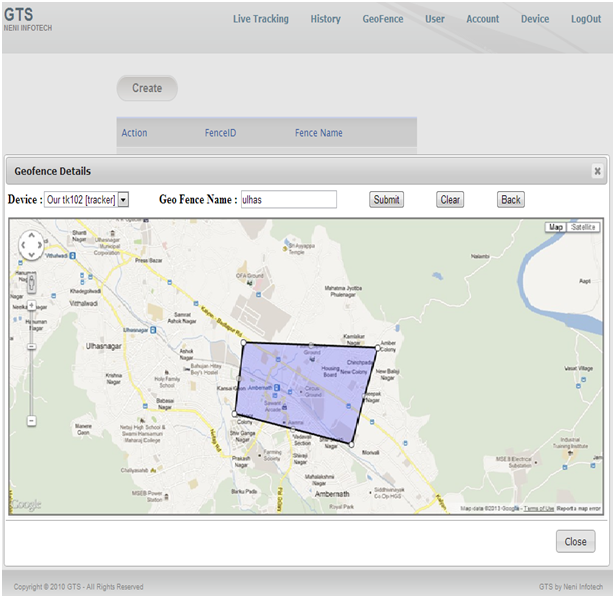
**7.3.13 Device-Delete**

****

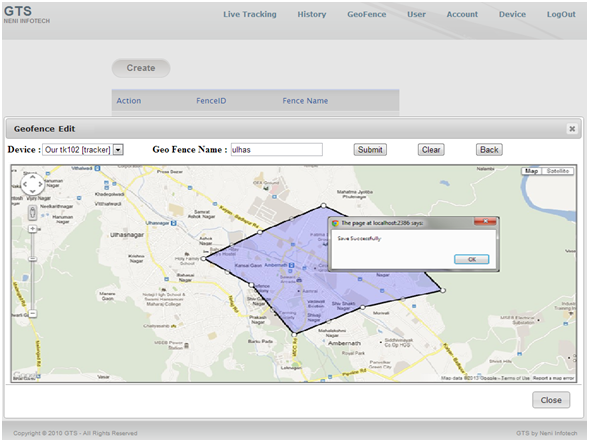
**7.3.14 GeoFence-Index**

****

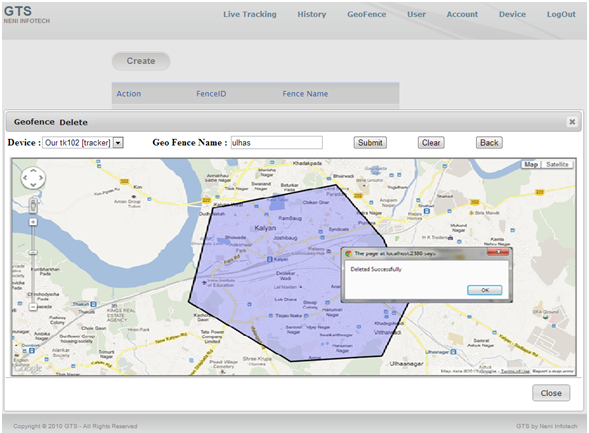
**7.3.15 GeoFence-Create**

****

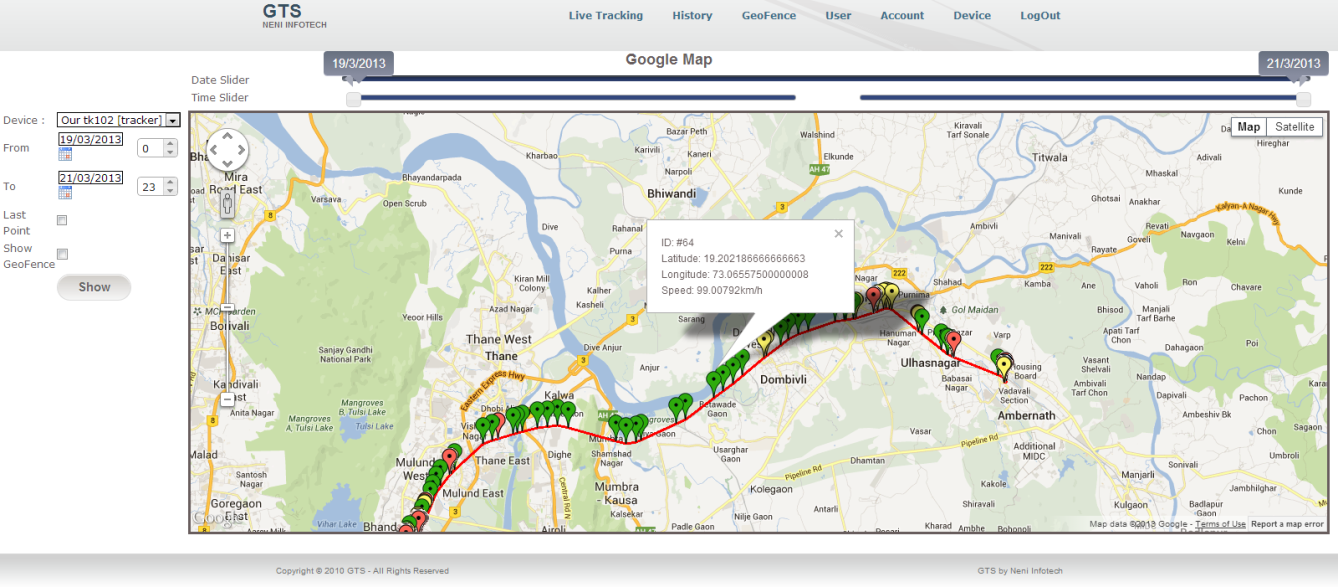
**7.3.16 GeoFence-Edit**

****

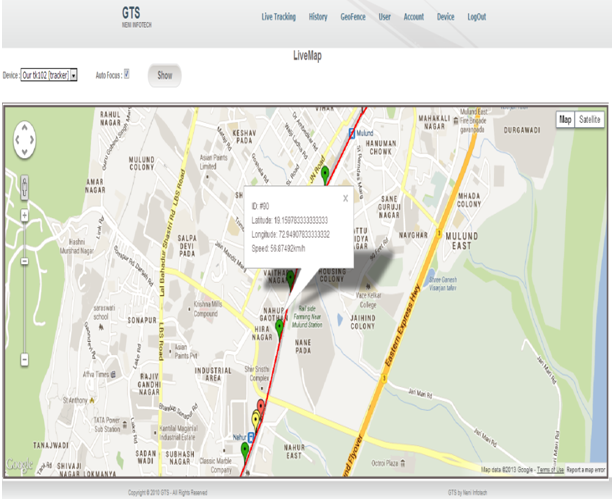
**7.3.17 GeoFence-Delete**



**7.3.18 Google Map**

****

**7.3.19 Live Map**

****

**8. Future Enhancement**

This project has dealt with the problem of integrating all the modules to create a system for the company. Results obtained from the project indicate that the proposed algorithm gives us the alert when device goes outside of geofence area & it is being tracked and stored in the database.

Further work on our project is of relevance learning to create an

* Operator interface:-

This Interface is designed to permit browsing and editing the system's databases in an intuitive and friendly way.

* Need to add facility of road assistance.
* Monthly automatic backup of complete database.
* Develop the website for android cell phones.

**9. Limitations**

The limitations of the website are as follows:

* The user has to be computer literate.
* Website will not work in case of internet unavailability
* The website will not work in case of power failure of tracking device
* Since satellite transmission power is low, certain geographical conditions may cause problems with a GPS receiver’s ability to record location data.
* Maintenance costs are high and expensive to fabricate
* Speed can be only monitored in levels
* Accuracy of up to 3 meters
* **Terrain**– Signals can become degraded and the receiver system may not provide location information if the view of the sky is severely limited. This situation can occur in deep canyons, or under dense vegetation.
* **Urban Canyons**– Large or tall buildings grouped closely together can cause large multi-path and fading errors that may affect the ability to track offenders.
* **Weather**– Signal strength can become degraded by moisture such as rainfall, fog, or snowfall.

**10.CONCLUSION**

In the development of the system we interacted with each and every modules and we came to known things get displayed in the History Map. Learning and to knowing things about History Map thought us many things.

To display things on the History Map through database interaction was shown. We tried to create a system for the company so that things can be integrated and even the records can be maintained and can be fetched from the data base.

The objective of this project was to build a program for tracking of child, adults and vehicles. The website developed is able to meet all the basic requirements of tracking device. The management of the records (both user and device) will be also benefited by the proposed website, as it will automate the whole procedure, which will reduce the workload. The security of the website is also one of the prime concerns.

There is always a room for improvement in any software, however efficient the system may be. The important thing is that the website should be flexible enough for future modifications. The website has been factored into different modules to make system adapt to the further changes. Every effort has been made to cover all user requirements and make it user friendly.

Goal achieved: The website is able provide the interface to the owner so that he can replicate his desired data.

User friendliness: Though the most part of the website is supposed to act in the background, efforts have been made to make the foreground interaction with user (owner) as smooth as possible. Also the integration of the existing system with the project has been kept in mind throughout the development phase.

**11. APPENDICES**

**Fundamental System Namespaces**

|  |  |
| --- | --- |
| System.Collection | Interfaces anad classes that define various collections of objects |
| System.ComponentModel | Classes that are used to implement the runtime and design-time behavior of components and controls |
| System.Configuration | Classes that are used to configure an assembly and allow custom installers to be created. |

**Data Management Namespaces**

|  |  |
| --- | --- |
| System.Data.Common | Classes that constitute the ADO.NET relational data access and management architecture for multiple data sources. |
| System.Data.SqlClient | Classes that support the Sqlserver .Net data provider |
| System.XML | Classes that provide standards-based support for processing xml |

**Windows Forms Application Namespaces**

|  |  |
| --- | --- |
| System.windows.Forms | Classes for creating windows-based executables applications to run under the .Net Framework |
| System. windows.Forms.design | Classes for extending design-time support for windows forms |

**Web Forms Application Namespaces**

|  |  |
| --- | --- |
| System.Web.UI | Classes and interfaces for creating user interface pages and controls in web applications. |
| System.Web.UI.Design | Classes for extending design-time support for webforms. |
| System.Web.UI. Design.webcontrols | Classes for extending design-time support for webcontrols. |
| System.Web.UI.htmlcontrols | Classes for creaing HTML servercontrols that map directly to standard Html elements. |
| System.Web.UI.Webcontrols | Classes for creaing ASP.NET webcontrols,which provides a consistent and abstracted interface. |

**12. Bibliography**

**Books**

* Kogent Learning Solutions Inc., “.NET 4.0 Programming Black Book”,dreamtech press, 2011
* B.Y.T. James, Fundamentals of global positioning system receivers a software approach, A John Wiley& sons, New York, 2004.
* Peter DeGette, Greg Low, Mark Whitethorn, “Introducing Microsoft SQLServer 2008”, Microsoft Press, 2008
* Ray Casting Algorithm for Vehicle Routing.

# Pro ASP.NET MVC Framework, Steven Sanderson

**Websites**

* + http://www.dotnetcurry.com
  + http://www.codeproject.com
  + http://www.java2s.com
  + http://www.quackit.com
  + http://www.w3schools.com