

# **ACCIDENT NOTIFICATION SYSTEM**

## **ACKNOWLEDGEMENT**

**Prepared by**

**AJAY BULANI 14030121006**

**BHAVIN PATHARE 14030121016**

**SYMBIOSIS INTERNATIONAL UNIVERSITY**

**YEAR OF SUBMISSION 2017**

**DEPARTMENT OF BACHELOR OF COMPUTER  
APPLICATION, SYMBIOSIS INSTITUTE OF  
COMPUTER STUDIES AND RESEARCH, PUNE,  
MAHARASHTRA.**

## **Abstract**

Accident Notification System is a notification system which helps, in case of car accidents. Whenever a car accident happens on highways there is a high possibility of accident gets ignored by people around the accident place, because of which medical help gets delayed to injured people. As a result of this, lots of deaths happens. This situation can be handled by using ANS. The purpose of this study is to provide emergency help to the injured people who are victims of the car accidents and reduce the response time taken currently. This help will be in the form of notification to nearest police station about the accident. The ANS will be installed already in the car so whenever accident takes place, it gets noticed by ANS and it will send the details about it to the web portal. So by monitoring the web portal, police can get information of the accident. Tracking every accident manually and providing necessary help to victims is not possible fully with the technology that Highway traffic police currently use. This problem can be solved by automating the task of getting police help at the accident place. After implementing this technology in the almost every car in India we can achieve a good success rate in providing emergency help to accident victims.

# Certificate

*This is to certify that*

Mr./Ms. Ajay Bulani, Bhavin Pathare

Has successfully completed the project entitled

**Accident Notification System**

For

**Dissertation 2014-2017**

In the partial fulfillment of the

Bachelor of Computer Application

From

Symbiosis Institute of Computer Studies & Research (Constituent of Symbiosis  
International (Deemed University), Pune)

Ms. Priti Kulkarni  
In Charge, BCA

Mr. Lalit Kathpalia  
Director

Name of Project Guide:  
Sign :

Name of Examiner :  
Sign :

# Study of Existing System

## Project Profile

### Literature Review

We did research about our topic and we found that similar work has been done already, but we found some gaps in ongoing and implemented projects, here is the summary of the followings :-

In 2011, the research is carried out by Olugbemiga and Emmanuel. They have developed the accident alert system and in which they said that their system will send the accident alert to 5 pre-configured numbers.

Another research states that their system will send the accident notification to the nearest police station and hospitals with the health condition of the victim using video camera which will be installed inside the car.

Similar type of research was done but their system can't be implemented in old cars and it's not affordable.

Another research said that their system will be based on in-vehicle GPS and GSM which is provided by car manufacturer. In their system they had used vibration sensor to sense the accident that accident is occurred or not and their system requires 3G technology for internet connectivity. Their project is totally based on vibration sensor and no other advance sensor.

Also some researchers are working for the same system using Geographical Information System (GIS) but they are still in development phase and they need to improve their network connectivity with their service provider.

**Drawbacks in the existing work** is that they are dependent on pre-configured numbers, video cameras, high speed Internet, and some parts from car manufacturer and also car models.

**Over the existing system ANS (Accident Notification System)** has advantages like, it displays the nearby hospitals name and their address to the police station with the accident mark, there is no pre-configured number for sending notification, user (Police Station) can login into the system and can change account settings like change password etc.

## Proposed Objective

**Accident Notification System** is a notification system which helps, in case of car accidents. Whenever a car accident happens on highways there is a high possibility of accident gets ignored by people around the accident place, because of which medical help gets delayed to injured people. As a result of this, lots of deaths happens. This situation can be handled by using ANS.

The **purpose** of this study is to provide medical help to the injured people who are victims of the car accidents and reduce the response time taken currently. This help will be in the form of notification to nearest police station about the accident. The ANS will be installed already in the car so whenever accident takes place, it gets noticed by ANS and it will send the details about it to the web portal. So by monitoring the web portal, police can get information of the accident.

Tracking every accident manually and providing necessary help to victims is not possible fully with the technology that Highway traffic police currently use. This problem can be solved by automating the task of getting police help at the accident place. After implementing this technology in almost every car in India we can achieve a good success rate in providing medical help to accident victims in case the car meet with the accident.

## Project Methodology

**Triple Axis Accelerometer ADXL345**, GPS Module and SIM900A GPRS MODULE will be connected with the Arduino Microcontroller Board Robodo Mega 2560. The Triple Axis Accelerometer will continuously monitor the x, y, and z coordinates to determine car's current position and speed. GPS Module will continuously monitoring the car's exact location. In case of accident the car's body will get damaged and this change will detect by Triple Axis Accelerometer ADXL345, if the car is in upside down position then the system will detect that accident is happened.

After the accident Arduino Mega 2560 will send the details/data such as car's location, licence no, car name to Apache web server. Then the web server will add this information/data into MySQL database, by taking location coordinates, Google Maps API will detect the exact location of the car and nearest police station around the accident place, then web server will send the accident notification or accident details in the form of alert message to the account of nearest police station. Notification will be in the form of popup on the webpage. Police will be able to see the accident location on their account then police will take further actions.

## Conclusion

This dissertation has described a new method for accident notification to police stations. Various existing systems were analyzed to get the in depth knowledge about the accident notifications. Existing systems are lacking some important features those features are covered in this dissertation. Also new features are described in the dissertation which increases the feasibility of this dissertation. Findings from study of existing systems helped to identify the flaws in existing system which were corrected in the proposed accident notification system. ANS is the accident notification system ,which is useful when a car accident happens. Vehicle's location is continuously track by GPS module. If accident happens then it transfers the vehicle's location to the Web Server using GPRS module. Then ,On server side, Google Maps API analyzes the location and send the notification of accident to the account of nearest police station.

## Problems Faced By The User (Pertaining to System and Software Requirement)

Any end user must have :-

- **Laptop/PC :-** In order to leverage all available features Accident Notification System.
- **ANS system on the Car :-** As a solution to human safety, in order to receive accident notifications with the location of the accident victim.

## Analysis Phase

### Objective of Proposed System

**Accident Notification System** is a notification system which helps, in case of car accidents. Whenever a car accident happens on highways there is a high possibility of accident gets ignored by people around the accident place, because of which medical help gets delayed to injured people. As a result of this, lots of deaths happens. This situation can be handled by using ANS.

The **purpose** of this study is to provide medical help to the injured people who are victims of the car accidents and reduce the response time taken currently. This help will be in the form of notification to nearest police station about the accident. The ANS will be installed already in the car so whenever

accident takes place, it gets noticed by ANS and it will send the details about it to the web portal. So by monitoring the web portal, police can get information of the accident.

Tracking every accident manually and providing necessary help to victims is not possible fully with the technology that Highway traffic police currently use. This problem can be solved by automating the task of getting police help at the accident place. After implementing this technology in almost every car in India we can achieve a good success rate in providing medical help to accident victims in case the car meet with the accident.

## Scope of the Proposed System

The scope of the system is limited in Maharashtra State only for the temporary period. Later on if it gets success in the market then it will be implement across all the states of the India.

## Feasibility Study

Some feasibility aspects are the followings:-

### Technical Feasibility

#### A. Hardware :-

1. Microcontroller
2. GPS Module
3. Network Connectivity Module
4. Speed Monitoring Module
5. Connection Wires
6. Power Supply
7. Bread Board

#### B. Software :-

1. Arduino IDE
2. Google Map
3. Browser
4. Database Server(AWS RDS)
5. Scripting Languages
6. MDL Front-End Framework
7. Apache Server on Public Cloud(AWS EC2)

## Economic Feasibility

Components	Quantity	Cost (in Rs)
Microcontroller	1	744 /-
GPS Module	1	1090 /-
Network Connectivity Module	1	1068 /-
Speed Monitoring Module	1	270 /-
Connection Wires	60	150 /-
Power Supply	1	
Bread Board	1	117 /-

## Operational Feasibility

This system will automate the detection of vehicle accidents and provide medical help to the victims as soon as possible to save their life. This system will display all the information on the website which will help to police station to get the accident location and nearby hospitals details such as name of the hospital and address of the hospital.

It will require the active database which will be manage by the admin and all other members who are managing the database. As the system will be fully automate. Although it is not compulsory but database manager can keep the track of all the accidents. As it is mentioned early that whenever police station wants to see the accident is happened or not, first they need to login into the system, if they are registered user, if not then they need to register first from the system itself.

## Schedule Feasibility

It would take over all estimate time will be 4 months to make the system , operational product for use. It will take around 15 days of gathering the requirements and analyzing it , and get approved. System demo will be ready within 1 month for Beta-testing. Beta-testing will add the add-on features and all the bug which are found at the time of beta-testing in 20-25 days.

## Legal Feasibility

The identity of the user (police station) will not be theft and data should not be used for any illegal purpose like any advertisement purpose. Patent and copyright of the logo and website.



# **Hardware and Software Requirements**

## **Hardware Requirements**

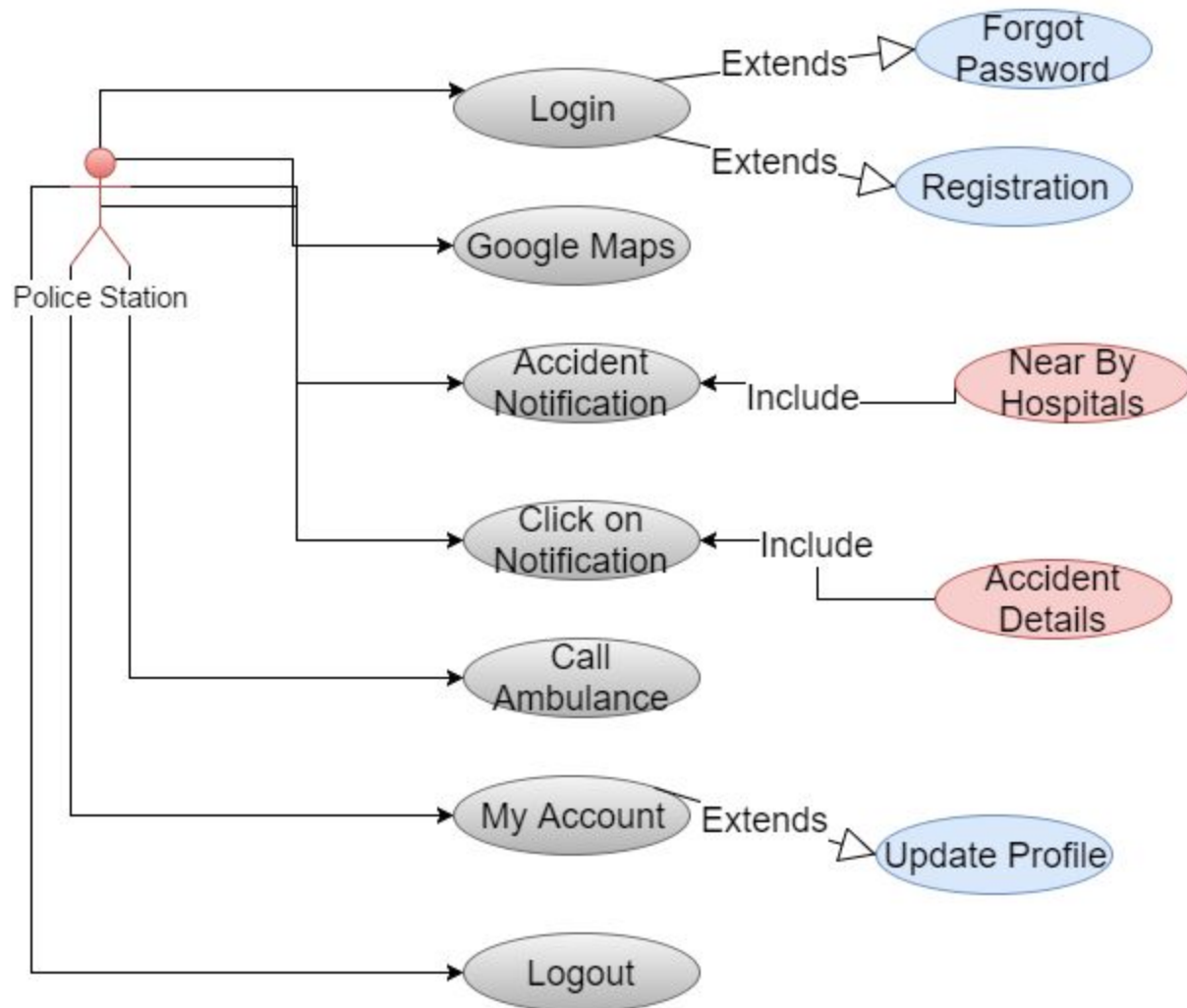
1. Arduino Microcontroller Board Robodo Mega 2560
2. SIM900A GSM/GPRS Serial & TTL Modem
3. Triple Axis Accelerometer- ADXL345
4. GPS Module
5. Jumper Wire
6. Bread Board
7. External Power Supply

## **Software Requirements**

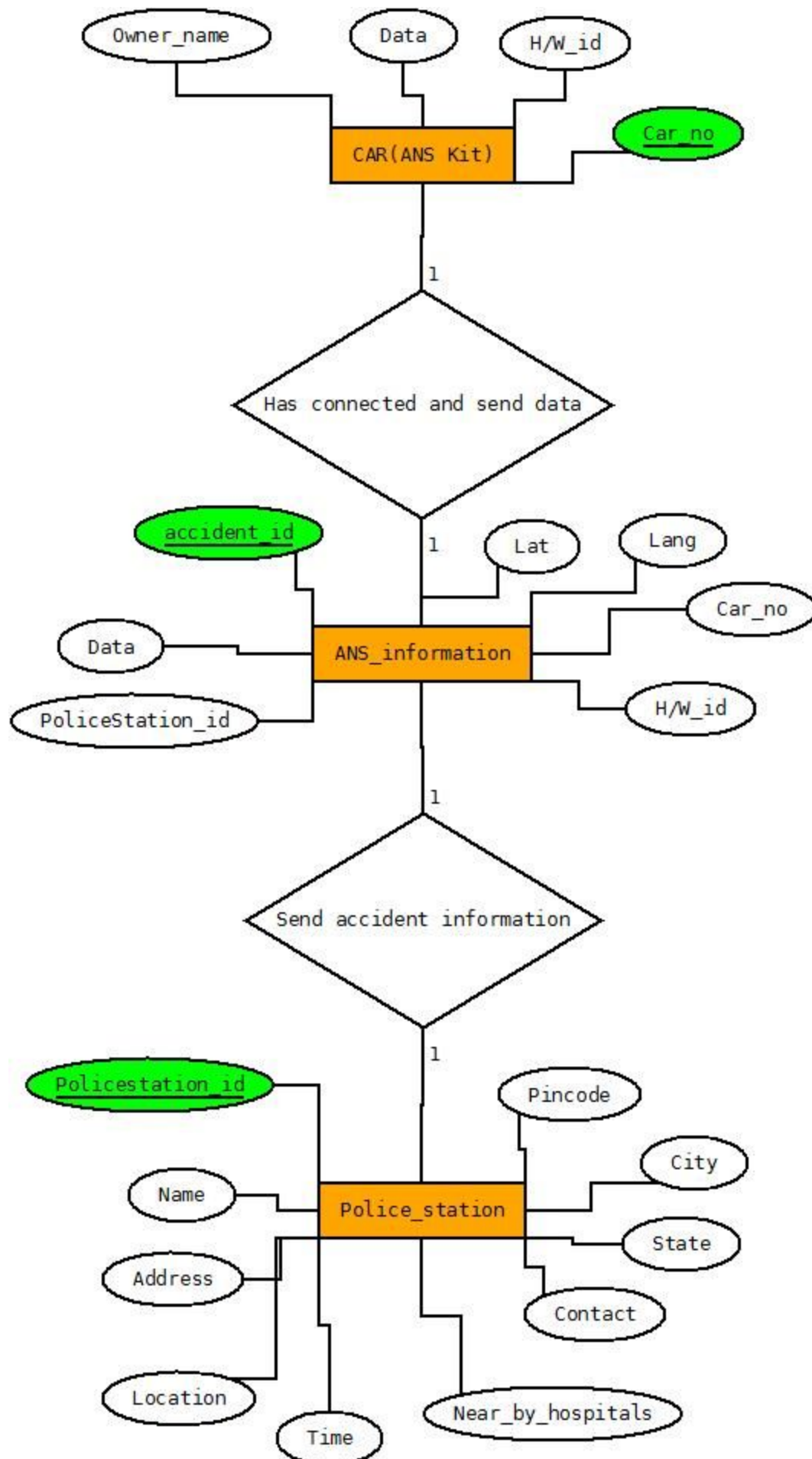
1. Web Server
2. Google Maps API
3. MySQL Database
4. Scripting Languages
5. MDL Framework
6. Arduino IDE
7. Public Cloud (AWS)

## **Use Case Diagram**

### ANS Use Case Diagram



### Entity Relationship Diagram (ERD)



# Design Documents

## 1. Software Architectural Design

The model used to elaborate this document is the **MVC model** which has become the most and frequently used model in the Web Application Programming in today's world.

**Model-View-Control**, it is a design pattern customise software; MVC is not an enterprise model, it is not used for product design but it's used for small organization. In other words MVC means

1. Model:- Structure
2. View:- User Interface
3. Control:- Controller

### Description

As with other software architectures, MVC expresses the “core of the solution” to a problem while allowing it to be adapted for each system. The central component of MVC, the model, captures the behavior of the application in terms of its problem domain, independent of the user interface.

- The *model* directly manages the data, logic, and rules of the application.
- A *view* can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- The *controller*, accepts the inputs and convert it to commands for the model or view.

### Interactions

In addition to dividing the application into three kinds of components, the model-view-controller design defines the interactions between them.

- A **model** stores data that is retrieved according to commands from the controller and displayed in the view.
- A **view** generates new output to the user based on changes in the model.

A **controller** can send commands to the model to update the model's state e.g. editing a document. It can also send commands to its associated view to change the view's presentation of the model e.g. scrolling through a document.

**Regarding our project, our software is being divided into these 3 respective parts.**

1. **View** as Accident Notification System UI (User Interface).
2. **Controller** as our PHP, JavaScript and MySQL.
3. **Model** as our Accident Notification System Database (RDS).

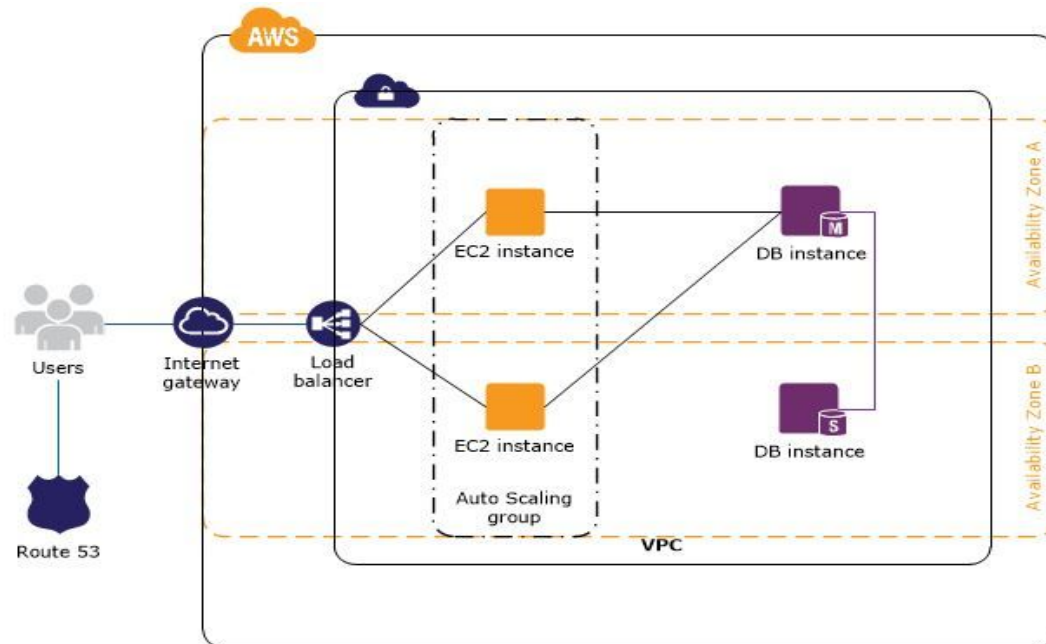
From the **view part**, multiple web pages are going to show up to the end users.

The **controller part** is going to accept the data inputs from the hardware, converts it to commands for the Model or View. In case of the model part, this should be able to reply to the commands given to it.

The **model part** is where the inputs collected from the database or hardware, treated or validated by controller are going to be stored and later on data going to be retrieved to produce outputs as reply to several queries or requests.

## Architectural Diagram

- Project is hosted on Amazon Web Services Cloud (AWS). Since it provides fault tolerance, high availability, security and scalability.
- Based on our project requirements we require AWS Relational Database Service (RDS) for satisfying the database related needs also Elastic Compute Cloud (EC2) as a Web server.
- Below diagram shows that we have setup two EC2 instances in different availability zones with RDS instances connected with them. The web traffic is going to evenly distribute across the two availability zones to provide fault tolerance, which will eventually reduce the load on web servers.
- All the resources are inside a Virtual Private Cloud (VPC) which provides the security to them. Security is added by using security groups which defines the inbound and outbound traffic for particular instance.
- *For example: web server can get http requests only and DB server is accessible through SSH from admin ip only.*
- Internet gateway is attached to VPC to provide access to instance from outside VPC. Only EC2 instance will have public ip address and database will have private ip address. Domain name is added in Route 53 which is DNS service. Users can access website using domain name of website.



## Why Amazon Web Services (Cloud) ?

### EC2 Features

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

- Virtual computing environments, known as instances.
- Preconfigured templates for your instances, known as Amazon Machine Images (AMIs) , that package the bits you need for your server (including the operating system and additional software).
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types.
- Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place).
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as instance store volumes.
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes.

- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as regions and Availability Zones.
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security.
- Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses.
- Metadata, known as tags, that you can create and assign to your Amazon EC2 resources.
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as virtual private clouds (VPCs).

## **RDS Features**

Amazon RDS is a managed relational database service that provides six familiar database engines to choose from, including Amazon Aurora, MySQL, MariaDB, Oracle, Microsoft SQL Server, and PostgreSQL. Which is :-

- Easy to Use
- Provides Automated Backups
- Multi-AZ Deployments
- Encryption at Rest and in Transit
- Database Event Notifications

At the end of mechanism, the end user will receive a processed data in the form of table, web page having all detailed information about accident.

## **Elastic Load Balancer (ELB)**

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to route application traffic.

## **Virtual Private Cloud (VPC)**

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

- Connect directly to the Internet (public subnets)– You can launch instances into a publicly accessible subnet where they can send and receive traffic from the Internet.
- Connect to the Internet using Network Address Translation (private subnets)– Private subnets can be used for instances that you do not want to be directly addressable from the Internet.

Instances in a private subnet can access the Internet without exposing their private IP address by routing their traffic through a Network Address Translation (NAT) gateway in a public subnet.

- Connect securely to your corporate datacenter– All traffic to and from instances in your VPC can be routed to your corporate datacenter over an industry standard, encrypted IPsec hardware VPN connection.
- Connect privately to other VPCs- Peered VPCs together to share resources across multiple virtual networks owned by your or other AWS accounts.

## **Route 53**

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like `www.example.com` into the numeric IP addresses like `192.0.2.1` that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.

Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS.

You can use Amazon Route 53 to configure DNS health checks to route traffic to healthy endpoints or to independently monitor the health of your application and its endpoints.

Amazon Route 53 Traffic Flow makes it easy for you to manage traffic globally through a variety of routing types, including Latency Based Routing, Geo DNS, and Weighted Round Robin—all of which can be combined with DNS Failover in order to enable a variety of low-latency, fault-tolerant architectures.

Using Amazon Route 53 Traffic Flow's simple visual editor, you can easily manage how your end-users are routed to your application's endpoints—whether in a single AWS region or distributed around the globe. Amazon Route 53 also offers Domain Name Registration – you can purchase and manage domain names such as `example.com` and Amazon Route 53 will automatically configure DNS settings for your domains.



## 2. UI Design

The UI pattern expected from this software application is the group of multiple web pages performing each one a more specific task with respect to the software goal. Here are some important screen from actual UI :-

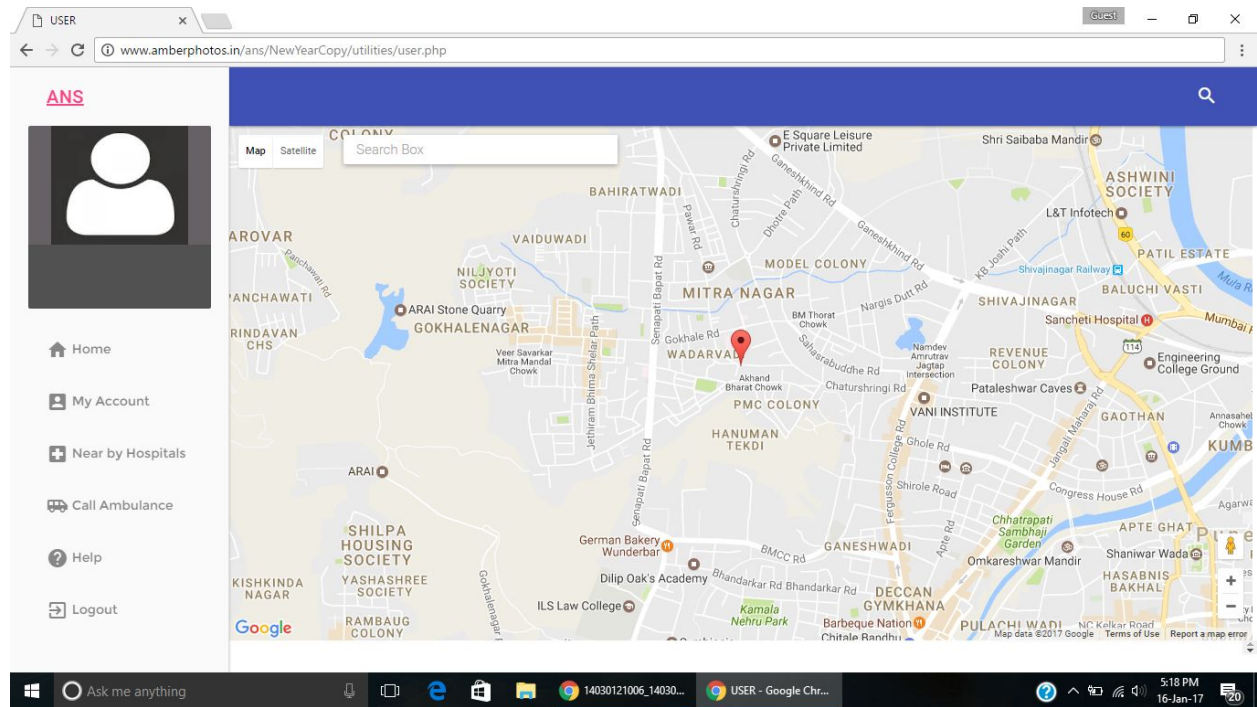
### Login Screen

The screenshot shows a web browser window with the URL [www.amberphotos.in/ans/NewYearCopy/](http://www.amberphotos.in/ans/NewYearCopy/). The page title is "Accident Notification System". The navigation bar includes links for Home, Help, Contact Us, and About Us. The main content area is titled "Login Here!" and features two input fields: "Enter ID" and "Enter Password". Below these fields are two buttons: "LOGIN" and "REGISTER". A link for "Forgot Password" is also present. At the bottom of the form, there is a paragraph of placeholder text: "Commodo nisi non consectetur voluptate incididunt mollit dui dolore amet amet tempor exercitation. Qui amet aute ea aute id ad aliquip proident. Irure dui qui labore deserunt enim in quis nisi sint consequat aliqua. Ex proident labore et laborum tempor fugiat sint magna veniam minim. Nulla dolor labore adipisicing in enim mollit laboris fugiat eu. Aliquip minim cillum ullamco voluptate non dolore non ex dui fugiat dui ad. Deserunt cillum ad et nisi amet non voluptate culpa qui do. Labore ullamco et minim proident est laborum mollit ad labore deserunt ut irure dolore. Reprehenderit ad ad irure ut irure qui est eu velit eu excepteur adipisicing culpa. Laborum cupidatat ullamco eu dui anim reprehenderit proident aute ad consectetur eiusmod."

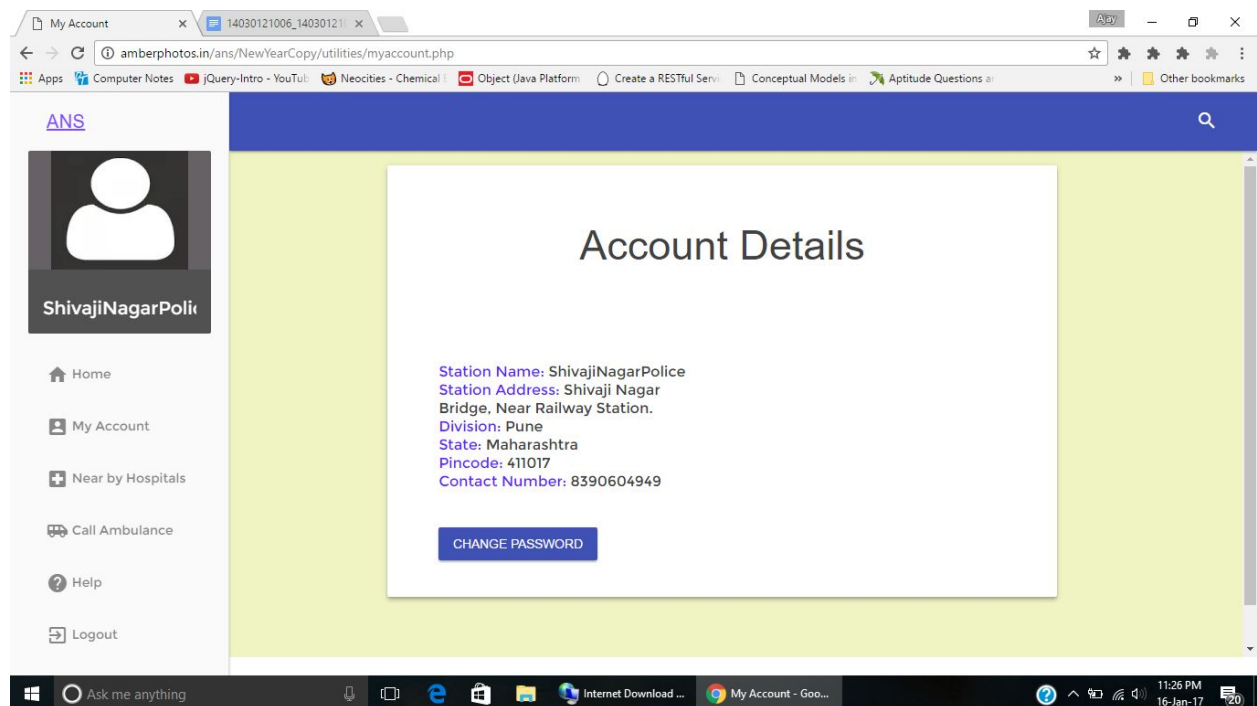
### Registration Screen

The screenshot shows a web browser window with the URL [www.amberphotos.in/ans/NewYearCopy/utilities/register.php](http://www.amberphotos.in/ans/NewYearCopy/utilities/register.php). The page title is "Accident Notification System". The navigation bar includes links for Home, Help, Contact Us, and About Us. The main content area is titled "Registration form". A note states: "Note: Email id will be your username". There are four input fields: "Enter Police Station Name" (with a message "Input is not a username"), "Enter Password" (with a message "Input is not a password"), "Confirm Password" (with a message "password not match"), and "Enter Email ID" (with a message "input is not email"). At the bottom of the form, there is a label for "Phone Number" which is partially obscured.

## User Screen



## User's account screen (My account)



The 4 screens shown on precedent and rest all screens are interdependent each other as after login the user screen can be shown. The same as only from user screen then my account screen could be shown.

Finally, you can log out if you want.

**Google's MDL design pattern** is used. Main reason of using MDL framework for front-end is to keep the ANS portal very easy to use and simple because our target audience will be police mens who are not techno savvy and they will be looking for very simple UI. Material Design Lite lets you add a Material Design look and feel to websites. It doesn't rely on any JavaScript frameworks and aims to optimize for cross-device use, gracefully degrade in older browsers, and offer an experience that is immediately accessible. The MDL component library includes new versions of common user interface controls such as buttons, checkboxes, and text fields, adapted to follow Material Design concepts. The MDL library also includes enhanced and specialized features like cards, column layouts, sliders, spinners, tabs, typography, and many more.

### 3. Program Design

Programs are designed using various programming and scripting languages. Codes are very optimized and we've made it so reusable to ensure consistency. Code is divided into functions, into separate files and wherever they will need they can be called by calling external files instead of writing the same code again and again.

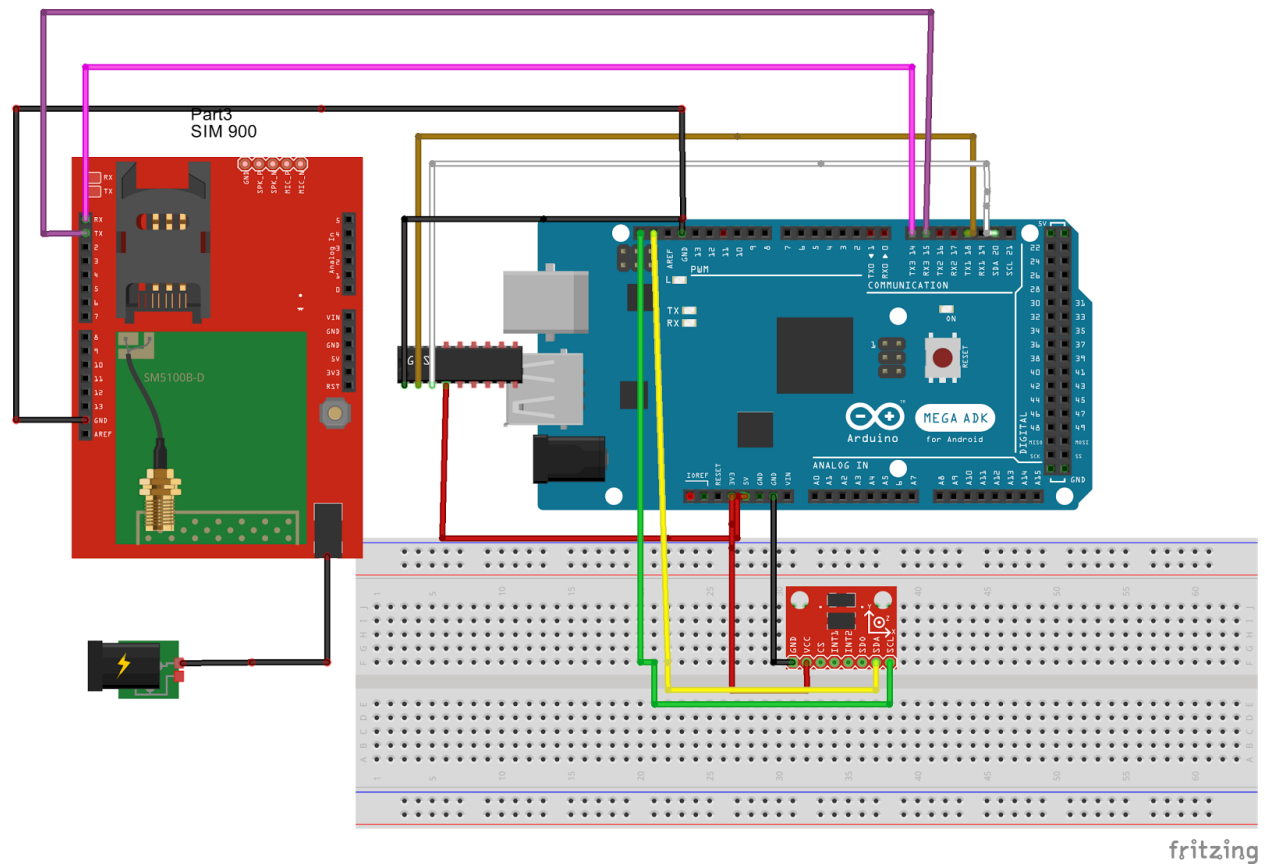
*The software has been built using the web development technology with the following programming and scripting languages:-*

1. HTML
2. CSS
3. JavaScript
4. PHP
5. MySQL

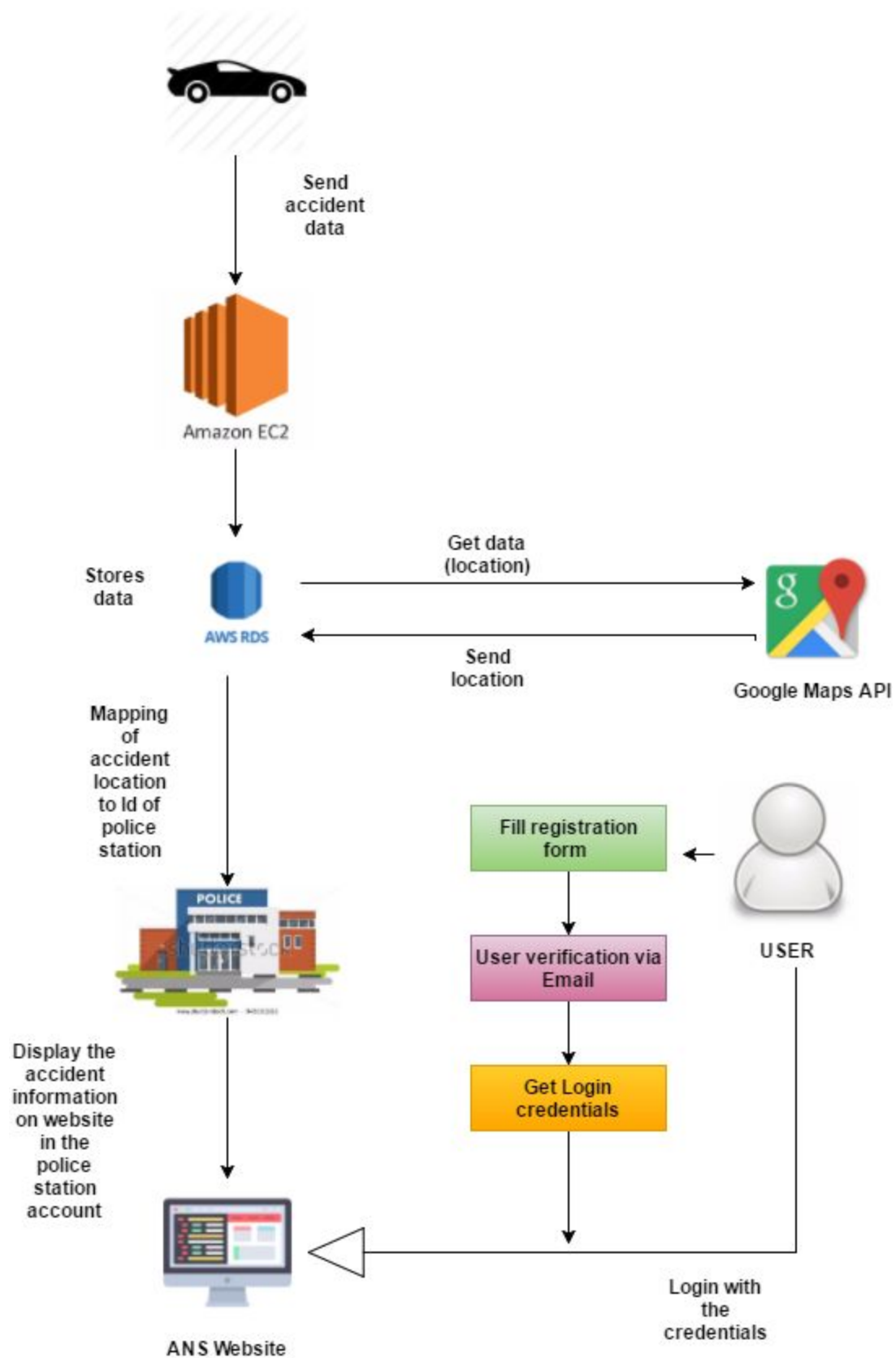
For the hardware ( i.e. microcontroller and rest of the sensor's) coding is done using C programming language with the help of **Arduino IDE**.

- For the front end part we've used **HTML and CSS**.
- To control web pages on the client side **Javascript** is used.
- For server side programming we've used **PHP** because of it has many libraries and object oriented functionalities. Also it works well with the **MYSQL** database.
- We've used **AWS RDS mysql server** instance as database server which is highly scalable and fault tolerant.
- **MySQL** is used for database operations.

### 4. Fritzing Diagram



## 5. Conceptual Diagram



# Construction of Code

## Code Organization Structure

Since the Accident Detection System is based on IOT technology and it follows the IOT code organizational structure.

IOT code is structured in such a way that all the folders have a special significance to our system. Code organizational structure can be follows :-



### Description of each and every folder of the project

- **ANS Website :-** This folder contains several subfolders which are followings
  - **db :-** This folder contains php files which have configuration related to the database connection and functions which are going to communicate with the database and as well as system also.
    - **db\_config :-** This contains the code related to database connection.
    - **db\_func :-** This contains the code which is used to send the email after fill the registration, verification of the user and all other functions related to the database.

- **images :-** This folder contains one subfolder and one image file which is related to the system.
  - **icons :-** It is the subfolder which contains the icon of the hospital.
- **js :-** This folder contains all the javascript files which are used in the system for duplication of the records which will be enter by the end user.
- **style :-** This folder contains all the CSS (cascading style sheets)files which are used in the system.
- **utilities :-** It contains all the core files of the system which are used in the system. It contains all the important files such as login, registration, google maps api, nearby hospitals and and many more which are very important for the system.
- **Hardware coding :-** This folder contains several subfolders which are followings
  - **ANS\_Final :-** Contains microcontroller programming which fetch the user's location, check the gravity using accelerometer and upload the accident data using gprs and gsm module.
  - **Modules :-** Contains subfolder which are following :-
    - **ADXL345 :-** Contains the code to monitor the gravity of the car using accelerometer and also monitor the coordinates of the car x,y,z.
    - **GPS :-** Contains the code to fetch the location of the user.
    - **SIM900 :-** Contains the code to provide the Internet to the arduino to upload the accident data to the server.

## Version Control

Version control is a concept whereby developers keep track of changes to a file or set of files over time so that it becomes easy to recall specific versions later. In this project three major tools Git, Repo, and Gerrit is used for version control. Brief description of these three tools is given below :-

**Git** is an open source version-control system designed to handle very large projects that are distributed over multiple repositories. In the context of IOT, we use Git for local operations such as local branching, commits, diffs, and edits. One of the challenges in setting up the IOT project was figuring out how to best support the outside community--from the hobbyist community to large OEMs building mass-market consumer devices. We wanted components to be replaceable, and we wanted interesting components to be able to grow a life of their own outside of IOT. We first chose a distributed revision control system, then further narrowed it down to Git.

**Repo** is a repository management tool that we built on top of Git. Repo unifies the many Git repositories when necessary, does the uploads to our revision control system, and automates parts of the IOT

development workflow. Repo is not meant to replace Git, only to make it easier to work with Git in the context of IOT. The repo command is an executable Python script that you can put anywhere in your path. In working with the IOT source files, you will use Repo for across-network operations. For example, with a single Repo command you can download files from multiple repositories into your local working directory.

**Gerrit** is a web-based code review system for projects that use git. Gerrit encourages more centralized use of Git by allowing all authorized users to submit changes, which are automatically merged if they pass code review. In addition, Gerrit makes reviewing easier by displaying changes side by side in-browser and enabling inline comments.

## Verification And Validation

- In the following test cases priority is displayed as 0,1,2 representing low, medium, and high respectively.

Test Case ID	Test Cases	Precondition	Input Test Condition	Steps To Be Executed	Expected Results	Actual Results
1	Test if user is able to login successfully		Correct email-id and correct password	1. Enter correct email-id and password on the respective field.  2. Click on login button.	User must be successfully login and to be redirected to desired page	User is logged in and redirected to the Map page
2	Test if incorrect, user is not able to login to the website		Incorrect email-id and incorrect password	1. Enter incorrect email-id and password on the respective fields.  2. Click on login button.	Proper error message must be displayed	Error message "Enter valid credentials"
3	Test with valid email-id and empty password such that login must get failed	User must be registered already	Valid email-id and empty password	1. Enter the valid email-id in the email-id field and enter no password in the password field	Proper error message must be displayed	Error message "Enter valid credentials"



4	Test with valid email-id but user's account is not activated	User must be registered already and he/she must be activate his/her account	Valid email-id and password	1. Enter the valid email-id and password  2. Click on login button	Proper error message must be displayed	Error message "Please activate your account, the link is sent to your email-id"
5	Test with, user forgot the password	User must be registered user	Registered email-id	1. Click on the forgot password link.  2. Enter registered email-id 3. Click on "Generate Link button"	Link to reset the password to be send to the user's email-id	Message "Link sent"
6	Test with registration of the user		Correct details of the user for e.g. name, email, etc.	1. Click on the "Register button"  2. Fill the registration form  3. Click on "Register" button	Link must sent to the user's email-id if all the details are valid in the form	Message "Thanks for registration,  To verify your account, Please check your email email, email can be in spam folder"

# **Drawbacks And Future Enhancements**

## **Future Enhancements**

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed system of Accident Notification System (ANS) which is ready and fully functional, the police station is now able to see the accident of himself near by the police station location and hence can take the action in a much better, accuracy and error free manner. **The followings are the future scope for the project :-**

- ❖ Call to the police station's location nearby ambulance services.
- ❖ Android application of the system.
- ❖ Graph generation of the accident data with respect to each and every police station.

## **Drawbacks**

No one expects science to be perfect at the first time while our peers can be highly critical, no one's work is beyond limitations. **The following are the limitations or drawbacks of the project :-**

- ❖ GPS sensor needs an open space to fetch the location.
- ❖ End user has to monitor the screen continuously to check the accident, whether it occurs or not.
- ❖ To implement the hardware kit inside the car it needs to be connected to the power which will be provided by the car's battery.
- ❖ For Internet connectivity car's owner has to insert the SIM card inside the SIM900 GPRS module to get the Internet.