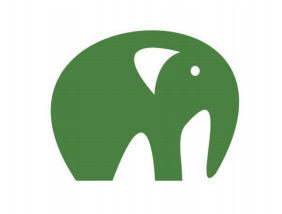
****

D.R.T.THILANKA 2493 ICT 09/10/051

M.R.R.Y.W.RANATHUNGA 2482 ICT 09/10/041

D.M.M.A.D.SUMANARATHNA 2491 ICT 09/10/049

M.G.L.L.JAYASEKARA 2461 ICT 09/10/020

D.M.A.KOTIKAWATTA 2468 ICT 09/10/058

M.G.S.S.GUNARATHNA 2454 ICT 09/10/011

Human Elephant Conflict Mitigation System and Elephant information sharing  
hub with behavior analyzer

**Bachelor of the Science in Information and Communication Technology**  
**Department of Physical Sciences**  
**Faculty of Applied Sciences**  
**Rajarata University of Sri Lanka**

EthPaura

# DecLaRation

Title Human Elephant Conflict Mitigation System and Elephant Information Sharing hub with behavior analyzer

Name EthPaura

Department Department of Physical sciences

Supervisor Mr. Amarakoon

Date 2/5/2015

Award Title Bachelor of Science in Information and Communication Technology

We hereby declare that the project work entitled “Human Elephant Conflict Mitigation System and Elephant Information Sharing hub with behavior analyzer” submitted to the Faculty of Applied Sciences of Rajarata University of Sri Lanka, is a record of an original work done by Group of “EthPaura” under the guidance of Dr. Shantha Fanando, Mr.K.G.T.Jayawardena and Mrs.Thilini Irugalbandara members of the Faculty of Applied Sciences of Rajarata University of Sri Lanka and this project work has not performed the basis for the award of any Degree or diploma/associate ship/fellowship and similar project if any.

D.R.T.THILANKA 2493 ICT 09/10/051

M.R.R.Y.W.RANATHUNGA 2482 ICT 09/10/041

D.M.M.A.D.SUMANARATHNA 2491 ICT 09/10/049

M.G.L.L.JAYASEKARA 2461 ICT 09/10/020

D.M.A.KOTIKAWATTA 2468 ICT 09/10/058

M.G.S.S.GUNARATHNA 2454 ICT 09/10/011

# Abstract

All most all the students among the Undergraduate society facing reasonable amount of challenge when they're about to leave the university for industrial training or jobs due to the fact that they have to find the right career path & right place to be. Even the academic staffs or career guidance units will not only have to process and sort huge amount of applicants, but also will need to be in communication with organizations who provide career opportunities to the students. To overcome this problem a system is been developed with ability of simplifying the process above mentioned. This system will use algorithms for Detecting elephant, predict Health stats and use web-service for sending data to central database and send SMS

# Dedication

We would like to dedicate the final report of Ethera project to Dr. Shantha Fernando and for all the ICT staff members in Rajarata University. There is no doubt that without their continued support and counsel we could not have completed this process with this much of success. A special feeling of gratitude to Dr. Sriyani Wickramasinghe for her encouragement and guidance.

# Table of Contents

Contents

[Acknowledgement 9](#_Toc411062278)

[Preamble 10](#_Toc411062279)

[1. EXcecutive summary 10](#_Toc411062280)

[2. Document conventions 10](#_Toc411062281)

[3. Intended audience and reading suggestions 10](#_Toc411062282)

[1. Introduction and overview 11](#_Toc411062283)

[1.1 Introduction 11](#_Toc411062284)

[1.2 Background and motivation 12](#_Toc411062285)

[1.2.1 Background 12](#_Toc411062286)

[1.2.2 Motivation 13](#_Toc411062287)

[1.3 Project scope 14](#_Toc411062288)

[1.4 Objectives 16](#_Toc411062289)

[1.5 Project nonfunctional requirements 17](#_Toc411062290)

[1.5.1 Product requirements 17](#_Toc411062291)

[Usability Requirements 17](#_Toc411062292)

[Efficiency Requirements 17](#_Toc411062293)

[Performance Requirements 17](#_Toc411062294)

[Space Requirements 17](#_Toc411062295)

[Reliability Requirements 17](#_Toc411062296)

[1.5.2 Organizational Requirements 18](#_Toc411062297)

[Delivery Requirement 18](#_Toc411062298)

[1.5.3 External requirements 18](#_Toc411062299)

[Interoperability requirement 18](#_Toc411062300)

[Ethical requirement 18](#_Toc411062301)

[1.5.4 Legislative requirements 18](#_Toc411062302)

[Privacy requirements 18](#_Toc411062303)

[Safety requirements 18](#_Toc411062304)

[2 Literature exploration and research 19](#_Toc411062305)

[2.1 Literature review 19](#_Toc411062306)

[2.2 Research and development 21](#_Toc411062307)

[2.2.1 Algorithm Development 21](#_Toc411062308)

[2.3 Technologies explored adapted 23](#_Toc411062309)

[2.3.1 Restful Web services 23](#_Toc411062310)

[2.3.2 Ardiuno 23](#_Toc411062311)

[2.3.3 Spring and Hibernate 23](#_Toc411062312)

[3 Project design 25](#_Toc411062313)

[3.1 Project concept 25](#_Toc411062314)

[3.2 Project development methodology and approach 26](#_Toc411062315)

[3.2.1 Agile Methodology 26](#_Toc411062316)

[3.2.2 Spring’s web MVC framework 26](#_Toc411062317)

[3.2.3 Object Oriented Approach 27](#_Toc411062318)

[3.3 Project design 28](#_Toc411062319)

[3.3.1 High level components and their interactions 28](#_Toc411062320)

[3.3.2 Components 28](#_Toc411062321)

[3.3.3 Interfaces 30](#_Toc411062322)

[3.4 Alternatives considerd 33](#_Toc411062323)

[3.4.1 Using Frameworks 33](#_Toc411062324)

[3.4.2 Using RESTful services for Dashboard and Android Application 33](#_Toc411062325)

[3.4.3 Using Java instead of other language 33](#_Toc411062326)

[3.4.4 Using Arduino instead of Raspberry pi 34](#_Toc411062327)

[3.4.5 Using AngularJS 34](#_Toc411062328)

[3.5 Detailed project design 34](#_Toc411062329)

[4 Project implementation 36](#_Toc411062330)

[4.1 Case tools used 36](#_Toc411062331)

[4.1.1 Visual Paradigm for UML – Sequence Diagrams 36](#_Toc411062332)

[4.2 Frameworks 36](#_Toc411062333)

[4.2.1 Spring 36](#_Toc411062334)

[4.2.2 Hibernate 37](#_Toc411062335)

[4.2.3 Bootstrappe 37](#_Toc411062336)

[4.2.4 Font Awesome 37](#_Toc411062337)

[4.2.5 AngularJS 37](#_Toc411062338)

[4.3 Business process flow 38](#_Toc411062339)

[4.4 Front end appearance for different system functionalities 39](#_Toc411062340)

[4.4.1 User Login view of ethPaura Information Hub Project 39](#_Toc411062341)

[4.4.2 User Registration form 40](#_Toc411062342)

[4.4.3 User Dashboard 43](#_Toc411062343)

[4.4.4 Elephant Detail 43](#_Toc411062344)

[4.4.5 Elephant Health data 44](#_Toc411062345)

[4.4.6 Elephant Health Report View 45](#_Toc411062346)

[4.4.7 Map of death places 45](#_Toc411062347)

[4.4.8 Map of Elephant Positions 46](#_Toc411062348)

[4.4.9 Recent Attacked places 47](#_Toc411062349)

[4.4.10 Admin Login 48](#_Toc411062350)

[4.4.11 Admin Dashboard 48](#_Toc411062351)

[4.4.12 Add Attack 49](#_Toc411062352)

[4.4.13 Add Attack zone 50](#_Toc411062353)

[4.4.14 Add Elephant view 51](#_Toc411062354)

[4.4.15 Add Record view 52](#_Toc411062355)

[4.5 device implementation 53](#_Toc411062356)

[4.5.1 Arduino device and WIFI shield 53](#_Toc411062357)

[4.5.2 Laser fence 56](#_Toc411062358)

[4.5.3 Village device 57](#_Toc411062359)

[4.6 Android application development 58](#_Toc411062360)

[4.6.1 Login view 58](#_Toc411062361)

[4.6.2 Admin Panel 59](#_Toc411062362)

[4.6.3 User Panel 61](#_Toc411062363)

[4.6.4 Elephant list 62](#_Toc411062364)

[4.6.5 Elephant detail 65](#_Toc411062365)

[5 Testing and result 66](#_Toc411062366)

[5.1 Test scenarios OF Information hub 66](#_Toc411062367)

[5.1.1 User Login 66](#_Toc411062368)

[5.1.2 Admin Login 66](#_Toc411062369)

[5.1.3 Add Elephant 67](#_Toc411062370)

[5.1.4 Register User 68](#_Toc411062371)

[5.1.5 Confirm User by Admin 69](#_Toc411062372)

[5.1.6 Add Record 70](#_Toc411062373)

[5.1.7 Add Attack zone 71](#_Toc411062374)

[5.2 test secenarios of device 72](#_Toc411062375)

[5.2.1 First laser cross 72](#_Toc411062376)

[5.2.2 Second laser cross 72](#_Toc411062377)

[5.2.3 Third laser cross 73](#_Toc411062378)

[5.2.4 First laser cross 73](#_Toc411062379)

[5.2.5 First laser and second cross 73](#_Toc411062380)

[5.2.6 Third and Fourth laser cross 74](#_Toc411062381)

[5.2.7 All the lasers cross at once 74](#_Toc411062382)

[5.2.8 First Laser and third Laser cross at once and then second and fourth 74](#_Toc411062383)

[5.2.9 First Laser and Fourth Laser cross at once and then second and third 74](#_Toc411062384)

[5.2.10 First Laser and second Laser cross at once and then third and fourth 75](#_Toc411062385)

[5.3 Test secenarios of android application 76](#_Toc411062386)

[5.3.1 User Login 76](#_Toc411062387)

[5.3.2 Admin Login 76](#_Toc411062388)

[6 Report summery 78](#_Toc411062389)

[6.1 Conclusion 78](#_Toc411062390)

[6.2 Recommendation 79](#_Toc411062391)

[6.3 Future work 80](#_Toc411062392)

[Referencess 81](#_Toc411062393)

# acknowledgement

We are highly indebted to Dr. Shantha Fernando and all ICT staff members of Rajarata University for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the EthPaura project with great success.

We would like to acknowledge Mr. Amarakoon who is the supervisor of the EthPaura project for his inspirational instruction, guidance, countless hours of reflecting, reading, encouraging, and most of all patience throughout the entire process. And also we would like to express my special gratitude and thanks to Dr. Sriyani Wickramasinghe for giving us such attention and time.

Finally it is appreciated and special thanks to all colleagues of group EthPaura in developing the project. Without the team spirit and dedication of each, we could not have completed this effort with this much of success.

# Preamble

## 1. EXcecutive summary

This document is intended to assist all interested parties by providing sufficient information about the project EthPaura, a multifaceted student industrial placement system. This project would be carried out to meet the software development requirements of the ICT-3411 module conducted by the Faculty of Applied Sciences Rajarata University of Sri Lanka. The document focuses on the overview and introduction, literature exploration and research, design, implementation and testing of the project EthPaura and will also serve as a document discussing the scope, technologies used and concepts explored, testing and results obtained and also the recommendations for better functionality and future improvements. This also discusses the novelty of the project and the uniqueness of the solution. This document is composed to provide a fully awareness on the project EthPaura.

## 2. Document conventions

|  |  |  |  |
| --- | --- | --- | --- |
| Headings Style | Font size(pt) | Font Type | Font Color |
| Master Heading | 14 | Times New Roman | White |
| Sub Heading | 13 | Times New Roman | White |
| Other Heading | 12 | Times New Roman | Black |
| Body | 12 | Times New Roman | Black |

## 3. intended audience and reading suggestions

This report on the project EthPaura will serve as the foremost reference material by the project team assigned to the project EthPaura. This document will be used as the reference material for all the academic staff members of ICT course in Faculty of Applied Sciences for the purpose of supervising and evaluating the project. It is also intended to provide as a legal document on EthPaura project with the Career Guidance Unit. Other than above mentioned categories, it will also serve as a great resource for anyone who is interested in this project.

**Chapter 01**

# 1. introduction and overview

## 1.1 introduction

The aim of the Faculty of Applied Sciences Rajarata University of Sri Lanka is building science graduates who have the ability to use their science knowledge and apply them precisely to solve challenging real world problems. So it makes the room to come into a clear thought that students coming out from the faculty as graduates should be skilled to apply their science knowledge in to various industrial needs. This output makes the difference between a typical science faculty and an applied sciences faculty.

Human-Elephant conflict (HEC) is not only a key conservation concern but also a major socio economic concern with no stand-alone universal solution. Influx of humans and conversion of natural habitat to human dominated land-use causes fragmentation and loss of elephant habitat. Because of this reason elephants raid crop fields and villages searching for food and water. Harmful methods employed by people in the process result in death and injury of both humans and elephants thereby escalating Human-Elephant conflict. Prevailing solutions include construction of elephant barriers such as rubble walls, ditches and canals, biological and electric fences, deployment of alarms, development of communication systems, capture, translocation and culling of problem animals etc. But none of them provide a successful control over the problem.

# 1.2 background and motivation

## 1.2.1 background

This proposal, prepared for the Rajarata University of Sri Lanka, will describe Human Elephant Conflict Mitigation System and Elephant Information Hub (ETH PAURA) which is targeted on mitigating elephant and death incidents attached. Recent literature, researches and interviews will be analyzed to determine the best possible solution. Final target deliverables will be a solution to mitigate elephant invasions and a web application with an analytical component to work as a hub for elephant information and researches.

## 1.2.2 motivation

* Identify their needs of motivation using Maslow’s hierarchy of needs.
* Appreciate their work at the group discussions and via the social media group.
* Give challenges to everyone in the group.
* Recognition and putting trust on them is important

# 1.3 project scope

Ethpaura is Human elephant conflict Mitigation system as well as it facilitates Information sharing among the elephant interested parties. The HEC Mitigation component send every warnings to subscribed villages and to the wild life officers if required.

The Early warning system is an efficient mechanism which facilitate tim information noceisd also the detection mechanism has a very high accuracy padsd.

Information sharing HUB is a place where Professors, Elephant Lovers, Researchers and all other interested parties can meet and share their documents, pictures, Questions and answers. Additionally,

* System takes raw collar data thrice a day and update elephant positions and their health data.
* Analyze those raw data to predict meaningful predictions and health states.
* Mapping elephant positions in a very that is easy to compare and analyze data.
* Detection mechanism was built around a virtual fence which defects elephant intrusions and send early warning SMS to the villages near that area.
* Also we update the database at the point of intrusion and allow administrators to update the information late with information on damages and casualties occurred.

**Project Capacity**

Ethpaura is mainly an elephant interaction system developed to overcome the social economic problem of Human-Elephant conflict using better and timely information sharing. This problem is not only apphicadsd to Sri Lanka, but also it common to many Asian and African countries. So when developing a common solution had many constraints, social, economic, technological etc.

Since soalability financial feasibility and the efficiency has to be highly vegarded while developing the system. We use simple, cheap but yet robust components to develop the system. Also to increase the efficiency and the accuracy we added simple, robust algorithms and eliminated elephant leaving detection mechanism from the device.

Since the additional needs of the system is basically a WiFi range and a central village computer with an internet connectivity this system can be easily applied to any village or a country.

Ethpaura is a Dynamic web application and it is update dynamically from data received from many parties.

**Beneficial Parties.**

As we started earthier in this document Ethpaura was built in the motivation of witiqating Human-Elephant conflicts which is a common problem to African and Asian. So the rural villagers who lives in the forest boarder areas in those countries will be benefitted with the intrustion detection mechanism. Apart from that the wildlife officers will get information about serous health alerts of collared elephant. When necessary. With the web application, professors, students, Researchers and any interested party will get a chance to share information rich this Hub.

From societies perspective this is a protective virtual fence which they can replay on most important part is this system does not contain any mechanism which will damage or harm elephant or any other animal. In other way elephant are also benefitted.

**Various aspects addressed by the project.**

Ethpaura is a Hub which connects elephants, villagers and wildlife Authorities by reducing communication gap using SMS and web applications to provide timely information.

System has the potential to integrate all the necessary parties to provide a better service supported by this system.

# 1.4 objectives

As mentioned above, it is convinced that the need of a cost effective, accurate elephant invasion detection system. Therefore apart from other objectives our main objective is to develop a solution to mitigate Human-Elephant-Conflict by using SMS, Wi-Fi technologies, laser modules etc.

**Specific objectives:-**

* To identify the best non-invasive elephant detection mechanism while studying different techniques available such as light and camera, ultra-sound, seismic waves, wireless sensors etc.
* To develop an effective solution that has the capability of deploying at village boarders considering the infrastructure barriers.
* To develop algorithms to maximize the accuracy rate of detection and to eliminate any false-triggers.
* To analyze elephant behavior patterns and their vital patterns to identify any critical situations.
* To improve awareness and information distribution via the information hub.

**General objectives:-**

* Reduce time wastage
* Preparation of final documentation
* Reliability & availability
* Reducing the cost of total solution

# 1.5 project nonfunctional requirements

## 1.5.1 product requirements

### Usability Requirements

* All the interfaces will be very user friendly where only basic knowledge of computers will be required.
* For the analyzer part the user should have expertise knowledge to deal with the system.

### Efficiency Requirements

* Web component provide an unlimited bandwidth where concurrent access is achieved.
* The HEC mitigation system will respond real time to protect the villagers from the attacks.

### Performance Requirements

* Need high performance WIFI Adapter with high range.
* More than 500m detected Laser.

### Space Requirements

* In space context the mobile apps will need fewer than 2MB space
* Analyzer will require lesser than 20MB space capacity
* Web component will need much space to store pictures hence taking up to 100MB.

### Reliability Requirements

* Since the HEC mitigation system is a life guarding system it will have 24x7 availability. And it will have more than 90% reliability.
* Analyzer will not active for 24hours. It will store data to the server when offline. The reliability of the information provided is a very challenging task. We try to achieve more than 75% reliability with the information accuracy.
* Web component will be available for 24x7. Also will have High bandwidth.

## 1.5.2 Organizational Requirements

### Delivery Requirement

* Goal is to deliver the system by June.
* We try to install it initially with a targeted village and then to expand the usage.

## 1.5.3 External requirements

### Interoperability requirement

* The whole system will work with each component to deliver a quality service.
* Communicate with servers and will use SMS gateways to communicate with the  
  users.

### Ethical requirement

* The main ethical problem we faced is the problem of animal right. We have to minimize the  
  harm and any inconvenience caused to the elephants.
* Also the concept of trapping them should be eliminated from the design.
* The design of the device of tracking them will be designed in a way to minimize the inconvenience.

## 1.5.4 Legislative requirements

### Privacy requirements

* The system shall ask for user-name and password at the registration interface of relevant components. Passwords will be encrypted before saving.
* Different access levels will be implemented so that some actions are prohibited to some users.
* Also the System will look into protecting necessary details and will not disclose to certain parties.
* Especially the locations of elephants will not be disclosed to citizens for the protective needs.

### Safety requirements

* There are no perceived risks for any external party or to the property. But there may be some issues in elephant hunting. So we have to organize disclosure levels in order to that.

**Chapter 02**

# 2 Literature exploration and research

## 2.1 literature review

The association between man and elephant in Sri Lanka is ancient. Elephants being the largest terrestrial herbivores require relatively large areas and diversity of environments to forage. With the increase in human population density and changes in the land-use patterns,  
elephant habitat is being continuously reduced. As a result, much of the present day elephant  
range extends into and overlaps with agricultural lands resulting in conflict with man. (Santiapillai et al., 2010) The surveillance and tracking of these herds are difficult due to their size and nature of movement. Therefore, there is a need for intelligent elephant surveillance and tracking system. The magnitude of impact of human– elephant conflict can be viewed from the fact that globally around 5 lakh families are affected by human– elephant conflict per year. (Sugumar and Jayaparvathy, 2014)  
**Related Projects / Literature** 1. An Improved Real Time Image Detection System for Elephant Intrusion along the Forest Border Areas, S. J. Sugumar and R. Jayaparvathy, January 2014 2. Wi-alert : a wireless sensor network based intrusion alert prototype for HEC, Ruwini Edirisinghe, Dileeka Dias, Rakhitha Chandrasekara, Lanka Wijesinghe, Prasanga Siriwardena and Prasad Kumara Sampath July,2013 3. An early warning system for elephant intrusion along the forest border areas by S. J. Sugumar and R. Jayaparvathy June,2013 Human–elephant conflict is a rapidly expanding area of research, with conservationists working hard to under- stand the circumstances under which tensions are highest between humans and elephants. A number of factors con- tribute to such conflicts, including population density of humans, elephant habitat structure, weather, time of year and animal life.(“2011 R.Sukumar J.Lenin Action Plan for the mitigation of Elephant - Human Conflict in India .pdf,” n.d.) Along the forest boarders, for many centuries, the villagers used many traditional approaches such as lighting fire crackers, making loud sounds and digging trenches to scare the elephants. A reasonably technical approach commonly introduced to control the movement of elephants worldwide is electric fences. Over 1000km has been erected with electric fences in Sri Lanka with over 300km being erected in 2009.(Wijesinghe et al., 2011). In electric fences elephants are deterred from forcing through the fence by an electric shock. The elephants tend to respect the fence and it acts as a psychological barrier. (“Wi-alert : a wireless sensor network based intrusion alert prototype for HEC, Ruwini Edirisinghe, Dileeka Dias, Rakhitha Chandrasekara, Lanka Wijesinghe, Prasanga Siriwardena and Prasad Kumara Sampath,” n.d.). Electric fence yet not regarded as the best possible solution since it produce many counter problems such as the high cost. It is approximately 0.5Million per Km. Apart from that the difficulty in maintenance, and the reaction from the elephants (They short circuit it with tree branches and logs) has made the electric fence redundant. Also 90% of the fence is not functioning now. (Sugumar and Jayaparvathy, 2014) used an automatic unsupervised elephant image detection system called EIDS. The EIDS was implemented for the well-known elephant corridors in Tamil Nadu, and the wireless cameras powered by a 12v batteries were mounted on the top of the trees and mountains. The pictures were taken every 5seconds and sent to a central facility to analyze using ‘Haar Wavelet’ algorithms and the pictures were cross-checked with 114 pictures in the Database. If more than 5 matches occurred an SMS will be sent to notify forest officials. The biggest problem we identified was the image retrieval time and the maintenance problem. (Sugumar and Jayaparvathy, 2014) mentions that they tried to overcome the retrieval time problem using a large distance. And the other problem which is the difficulty in maintaining (the tree branches blocking the cameras etc.) still prevails. Also the light conditions in nights will require switching into night vision modes. And we understand that the system costs high. As in the article, (“An early warning system for elephant intrusion along the forest border areas by S. J. Sugumar and R. Jayaparvathy,” n.d.) the intrusion detection system using geophones makes an event of interrupt when an elephant enters its radius of 24 meters. The footfall of the elephant produces a vibration in the ground which is sensed by the geophones(Sugumar and Jayaparvathy, 2013). But this system is considered not effective due to two major problems. In a situation where the soil moisture is very high due to rain fall it may not produce sufficient vibrations to be sensed. This problem is discussed in their next research article.(Sugumar and Jayaparvathy, 2014). Apart from that the system requires a high initial cost, approximately 75,000 Indian rupees. As we have studied, there prevails few different mechanisms as alternatives we can go along. Like ( Edirisinghe et al., 2013) says the requirements of a such system are,

* Should be non-invasive
* Tagging elephants with any kind of a device is not an option.
* Minimal installation complexity
* Energy efficiency
* Easy maintenance
* Scalability
* Minimal or no false alarms

Usage of light and camera as a detection mechanism will create problems of reliability and complexity.  
Given that the light conditions are varying over time during the day the reliability will be questioned.  
Ultra sonic waves for detection may be asked the question of distance limitation and the need of a complex rotating mechanisms.

Infra-sound and seismic waves may have the added advantage of chasing elephants back, but creating problems in complexity, high power requirement problem and need of high domain expertise.  
From the discussion we had on existing mechanisms raise the need of a robust, reliable and a simple  
cost-effective detection system. What will be addressed in our research will be the practicality,  
scalability and the cost effectiveness which had not addressed in the previous studies. The current and future studies are more towards the usage of wireless sensors and our research will cope with those technologies.

## 2.2 Research and development

Key issue addressed by EthPaura is identities elephant intrusions efficiently and Effectively Eliminating false alarms and developing the most efficient algorithm was the main concern we faced. Lots of alternatives solutions were considered and research was done to find out the most appropriate solution to detect system by considering various economic, social and infrastructure feasibility.

### 2.2.1 Algorithm Development

**Filter Algorithm for map component**

1. Load Filter View
2. Select Elephant name
3. Select map type
4. Select ‘Show map’

**Pseudo code**

Map\_period array = calculate date range from map type

Do while map\_period = NULL

If map period matches elephant

Return true

Else

Return false

End Do

**HealthData Filter Algorithm**

1. Load Filter View
2. Select elephant name
3. Select data type
4. select freaquency

**Pseudo code**

data array = calculate from data type and frequency

Do while data= NULL

If data matches elephant

Return true

Else

Return false

End Do

**Elephant Detection**

1. Both 1st and 2nd laser(placed in Forest side) crossed
2. Waiting for Other two crossed
3. Cross all the Lasers
4. Elephant is entering to the Village

**Pseudo code**

While always true

IF laser one and laser two both cross

IF laser three and four both cross

Send alert to village

End IF

End IF

End While

## 2.3 Technologies explored adapted

### 2.3.1 Restful Web services

Since this is a web based project, there should be a way to consume services from other parties and offer services to the relevant parties. Beside the ideal browser interaction, web services are the best way to interact with system in a hidden but robust way. In that fact, this system consist with several web service consuming end-points and & web-service offering end-points. Using JSON, as a way of constructing and retrieving data, send Data to mobile application, send data to scripting graph. REST, Representative State Transfer is a way of interacting with web services and enable an optimized way of communication. The design rationale behind the Web architecture can be described by an architectural style consisting of the set of constraints applied to elements within the architecture. By examining the impact of each constraint as it is added to the evolving style, people can identify the properties induced by the Web's constraints. Additional constraints can then be applied to form a new architectural style that better reflects the desired properties of a modern Web architecture.

### 2.3.2 Ardiuno

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package

Since we use device level programming we have to use Arduino board as a device. For the Ardiuno device we programming by C Language. And we use following libraries for ardiuno

**#include** <ccspi.h>

**#include** <SPI.h>

**#include** <string.h>

**#include** "utility/debug.h"

**#include**<stdlib.h>

Also we use following library for CC3000 WIFI shield.

**#include** <Adafruit\_CC3000.h>

### 2.3.3 Spring and Hibernate

"Spring Integrates very well with Hibernate". So what? One might think. Spring integrates very well with lots of technologies, what benefits we get out of Hibernate integration. I'm trying to list out important benefits that applications get from Spring and hibernate integration.

1. Services layer

The most important benefit is the Spring framework itself. Because of the Spring integration, applications can leverage all the features of spring framework. For example POJO style serve interfaces, IoC, Aspects, Remoting etc...

Spring framework provides all these features out-of-box. And a good thing about Spring framework is that it's really easy to plug any custom component into the framework. With spring POJOs, IoC and Remoting it's very easy to develop services layer. Otherwise, we need to implement EJB Session Bean (stateful or stateless) or depend on the frameworks like JSF, Struts or Servlets to implement the business logic which is not ideal. All of these frameworks business logic from Presentation layer or tedious to develop and maintain (Struts, for example).

1. HibernateTemplate

Spring provides template for managing sessions, transaction across the application. Without the hibernate template, applications need to manage these on their own. Note: With the Hibernate 3.0.1, Spring can manage the transactions and session management across the application without the template. Only thing you don't get without Hibernate template is Exception translation. But there are other options.

1. HibernateDaoSupport

Spring provides a class called HibernateDaoSupport. DAO Implementation class implements this class to get all the convenience methods that HibernateTemplate provides. If applications are not using HibernateTemplate, then it's good to take advantage of HibernateDaoSupport.

1. DAOException Translation

Hibernate 3 throws Runtime exceptions unlike checked exceptions thrown in the previous releases. Spring can translate these exceptions into SpringDAO exceptions and map them into Spring DAO exception hierarchy. But you have to use one of the following to get this

1. MVC Integration

If you are using Hibernate in you web application, which is using Spring MVC, then you can use the OpenSessionInViewFilter or OpenSessionInViewInterceptor and not create sessions per thread, instead create a session per Request.

**Chapter 03**

# 3 project design

## 3.1 project concept

As a Sri Lankan we face Human-Elephant conflict situations throughout the year. The government and other University & individual researchers are continuously doing researchers to find a robust and suitable solutions to this problem. As we noticed the solutions should be a system that should not address human problems, but it should protect elephant also (Elephant friendly).

ICT undergraduate always expect to go beyond developing technical systems. So we decided to experiment on this national problem to find a solution by timely information sharing.

Thousands of Human as well as Elephant lives get damage because of the Human-Elephant conflicts every year. Not only that but also millions of property damage happen every year. Village famers and their families lose their bread and breakfast due to elephant infractions to crop fields and barn yards. Not only that, but on the other side the elephants face problems while searching food and water. Common problem is in forest boarder areas Elephants & villagers share same resources like water tanks etc. So a better understanding between those parties is needed.

By studying the problem domain we came across many existing solutions and other applicable technologies. What was the biggest requirements was the system should be non inrasre, robust and should respond very casily. Also the scalability requirement was regarded highly.

Since the problem was worth enough to address we thought of applying our knowledge we gained from our degree program to come up with a new and better solution.

The idea of the concept of this project came up with the existing national & Socio-Economical problem of Human – Elephant conflict. And since we understood the absence of an information sharing mechanism which people can share their knowledge and how they can increase the awareness about elephants, we thought of developing a web application which will work as a Hub. Linking elephants and all elephant interesting parties.

## 3.2 project development methodology and approach

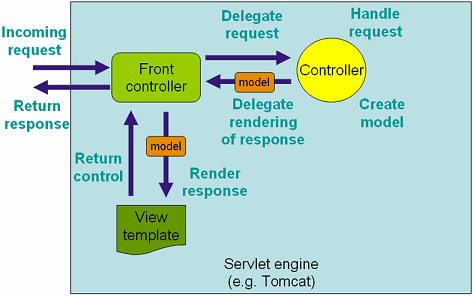
### 3.2.1 Agile Methodology

Agile development, in its simplest form, offers a lightweight framework for helping teams, given a constantly evolving functional and technical landscape, maintain a focus on the rapid delivery of business value. As a result of this focus and its associated benefits, organizations are capable of significantly reducing the overall risk associated with software development.

In particular, agile development accelerates the delivery of initial business value, and through a process of continuous planning and feedback, is able to ensure that value is continuing to be maximized throughout the development process. As a result of this iterative planning and feedback loop, teams are able to continuously align the delivered software with desired business needs, easily adapting to changing requirements throughout the process. By measuring and evaluating status based on the undeniable truth of working, testing software, much more accurate visibility into the actual progress of projects is available. Finally, as a result of following an agile process, at the conclusion of a project is a software system that much better addresses the business and customer needs.

### 3.2.2 Spring’s web MVC framework

In Spring Web MVC you can use any object as a command or form-backing object; you do not need to implement a framework-specific interface or base class. Spring’s data binding is highly flexible: for example, it treats type mismatches as validation errors that can be evaluated by the application, not as system errors. Thus you need not duplicate your business objects’ properties as simple, untyped strings in your form objects simply to handle invalid submissions, or to convert the Strings properly. Instead, it is often preferable to bind directly to your business objects.



Spring’s web MVC framework is, like many other web MVC frameworks, request-driven, designed around a central servlet that dispatches requests to controllers and offers other functionality that facilitates the development of web applications. Spring’s DispatcherServlet is completely integrated with Spring IoC container and allows us to use every other feature of **Spring**.

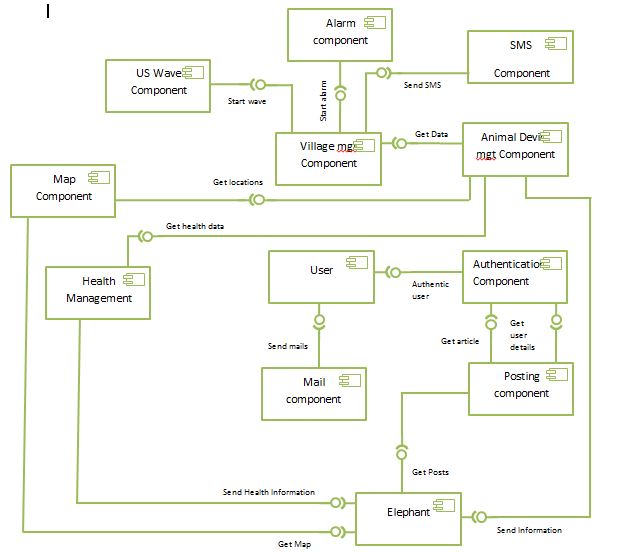
### 3.2.3 Object Oriented Approach

Object-oriented architecture is a design paradigm based on the division of responsibilities for an application or system into individual reusable and self-sufficient objects, each containing the data and the behavior relevant to the object. An object-oriented design views a system as a series of cooperating objects, instead of a set of routines or procedural instructions. Objects are discrete, and controllers. As shown in figure a request from the client (a browser in most cases) will come in to a Controller via Spring’s DispatcherServlet. And then through a method controller will instruct the models to fetch relevant data from the data sources. After manipulating data, the controller again instruct to the view to render the necessary data to client. In this simple scenario the MVC will keep the interconnection through Object-Oriented style and will pass messages to communicate between them.

## 3.3 project design

In design, this large and complex system it has divided into manageable small components. Partitioning large systems into subsystems and components, each does a separate task, make it easy to design and develop. Therefore entire system has divided into components. Each component operates separately and use interfaces to communicate with each other.

### 3.3.1 High level components and their interactions



### 3.3.2 Components

**Map Component**

This component has internal functionality of handling maps. Here we use algorithms to simplify and optimize the use of map component. This component in that sense is responsible for handling all map related functions.

**SMS Component**

SMS component will be called when there is a need for sending bulk or single SMS s. We hope to use two hosted SMS API from a service provider to implement this feature.

**Health Component**

Basically this component is responsible for keeping and handling health related data. Will get the service from Animal Device Manager Component and will mainly provide service to elephant component.

**User Authentication Component**

This component can be introduced as one of the key component in the system which bears the burden of security. And group based action restrictions. In that sense this will work as a gateway and will route users along their authorized parts of the system.

**Email Component**

Emailing component is responsible for single or bulk emails when needed. Other components can request the service from this component when needed. All the web registered users are supposed to have an email entered.

**Village Component**

This can be identified as a critical component. In the whole system responsible for handling data related to the villages, including whom to notify, when a threat what to notify, when to start other components as necessary. Will serve mainly to elephant device.

**User Component**

The main subsystem that has the internal functionality of handling and managing student related data. This component will maintain necessary categorization regarding users. Also this component is responsible for giving services to many other components as well.

**Posting Component**

This component is responsible for publishing posts. Also this is capable of handling multiuser posts. Some of the posts are automatically published by other components such as elephant component. And some of the posts published by users.

**Elephant Component**

Main component responsible for all the underline processes in web part. This component is responsible for handling and managing all the tasks related to elephants. Take the use of other components such as mail component etc. and give its service to other components such as post component.

**Elephant Device Component**

Main task of this component is to interpret the data between outside and the system. Will  
control the data formats and data set sending to the inner system. This is the main controlling body of  
the system.

### 3.3.3 Interfaces

Since this system implemented as tight MVC style there will be well defined interfaces between models, controllers and views. But for the sake of modularity, we’ll define some interfaces to breakdown the system structure. So below defined interfaces will use all the controllers within particular component as listed under each component.

**Map Component**

**getMap**

Will serve elephant maps particular to the selected elephant

Controllers involved

* buildmap Controller

**SMS Component**

**SendSMS**

Will act as a messenger component andwill send sms through an hostedAPI

Controllers involved

* smsController

**Health Component**

**SendHealthData**

Hold and send raw health data to the elephant component as received

**SendProcessInformation**

After analyzing received raw data send it to the elephant component

Controllers involved

* SendHealthController
* SendPredictionController

**UserAuthentication Component**

**authenticateUser**

Do a gatekeeper’s duty by authenticating the users and routing them with necessary control list privileges.

Controllers involved

* SystemUserController

**Mail Component**

**SendMail**

Serve as a postman. Another component can initiate the service asking to do the job.

Controllers involved

* Emails Controller

**Village Component**

**ThreadDetectorInterpreter**

Analyze the village ID and threat level by interpreting the message sent by the village device.

**getVillageContacts(ID, ……)**

Get necessary villagers’ contact details and make necessary duties to make the environment to send alerts.

**sendMassage**

Act as an intermediary who sends details necessary to send alert messages to the villagers and if necessary to wildlife officers.

Controllers involved

* villagerController
* threadDitectorController
* messageProcessController

**User Components**

**getNormalUser**

All the services related to non-logged, normal user can be obtained through this interface.

**getRegisterdUser**

Services related to logged Registered user can be obtained through this interface.

**getAdmin**

Services related to admin can be obtained through this interface.

Controllers involved

* AdminController
* RegisterdUserController
* UserController

**Posting Components**

**Getpost**

This interface is basically for the creation and notification of posts.

Controllers involved

* postController
* feedbackController

**Elephant Component**

**getElephant**

All the services related to the elephant is provided through this interface. This is one of the busiest interfaces in the system.

Controllers involved

* Elephant Controller
* Attacks Controller
* Prediction Controller
* Positions Controller

**ElephantDeviceManagement Component**

**InterpretMessage**

The communication among Elephant device and other inner components in the system by doing interpretations.

Controllers

* DataManagementController

## 3.4 Alternatives considerd

### 3.4.1 Using Frameworks

EthPaura project has used frameworks such as Spring, hibernate for development and Bootstrap for interface design. Foremost reason for this decision is easiness. Frameworks are adapted to well-known Software Engineering principles that make easy for the system to adapt to the SE principles. Also they are really comprehensive and complex therefore allows complex processing without any hassle. Code reusability is highly admired in complex projects, ease of code reusability is provided in frameworks. Enormous online help and support is available for frameworks and make it easy for system development.  
In contrast from scratch development take time and make it difficult to develop. Complex systems become more complex when develop from scratch and once any error encountered most of the time no help is available. Considering above facts ETHPAURA project has used frameworks instead from scratch development

### 3.4.2 Using RESTful services for Dashboard and Android Application

RESTful web services are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

Using RESTful we get data to Mobile application from the server. Then the android application can read that data and display. And also for the Dashboard widgets those are wrote by Java Scripts. For that we use RESTful service as a middle layer, which can transmit data from backend to frontend which is wrote by html and java script.

### 3.4.3 Using Java instead of other language

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependencies as possible. And one of the most popular programming languages in use, particularly for client-server web applications. Java is fast, secure, and reliable. Also we use framework Spring and hibernate which is based on java.

### 3.4.4 Using Arduino instead of Raspberry pi

Arduino is perfect for electronics projects. It contains a set of input and output that can often be connected directly to components and sensors, and is incredibly easy to just jump straight into making something. This makes it ideal for prototyping things.

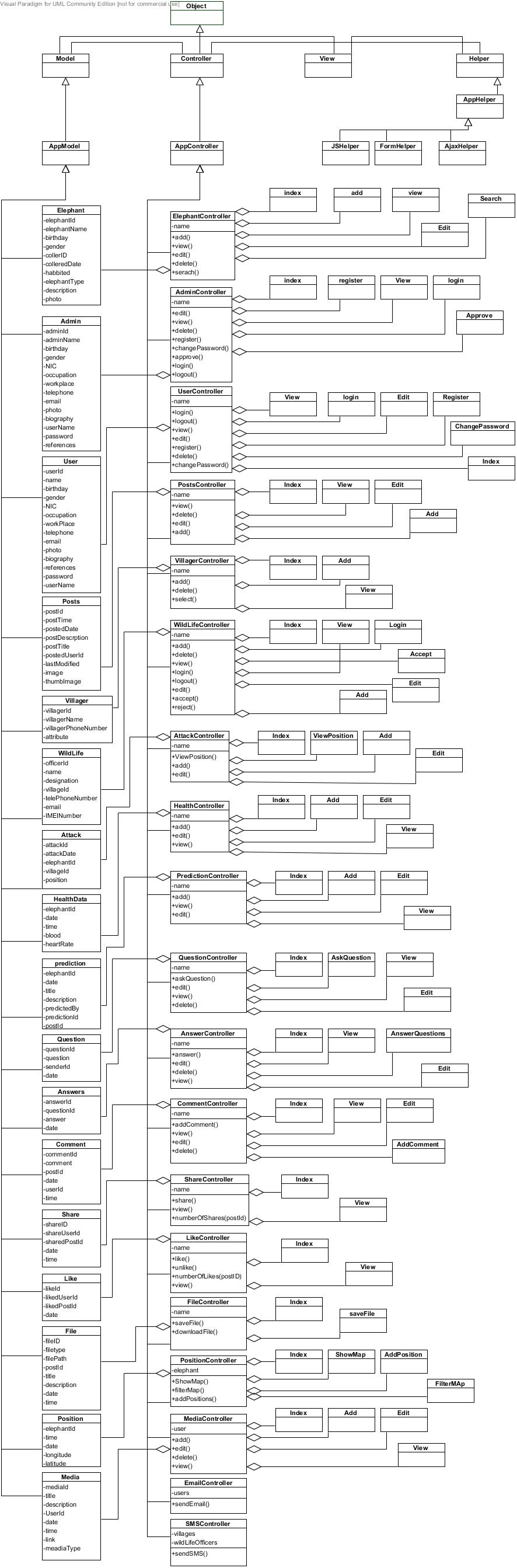
But the using Raspberry Pi we can't connect sensors directly to the device, so there will be a delay. For our project accuracy and efficiency is most important, so we use Ardiuno instead of Raspberry Pi

### 3.4.5 Using AngularJS

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop. So we use Angular for form validation.

## 3.5 detailed project design

In the detailed design, the project was brought in to a state of planning the whole design in different perspectives. In significant focus, there were “High-level component view”, “ER diagrams and data dictionary”, “Detailed Class diagram”. Thus above design diagrams had the firm structure to start the coding of the system with proper understanding for the developers. Attached appendices in numbers, are the detailed design structures formulated at that stage.



**CHAPTER 04**

# 4 project implementation

## 4.1 case tools used

Computer-aided software engineering (CASE) is the application of a set of tools and methods to a software system with the desired end result of high-quality, defect-free, and maintainable software products. It also refers to methods for the development of information systems together with automated tools that can be used in the software development process.

In EthPaura development process many CASE tools were used in drawing various diagrams such as class diagrams, ER diagram, activity diagrams, and sequence diagrams and in designing interfaces. The CASE tools we used are described below one by one

### 4.1.1 Visual Paradigm for UML – Sequence Diagrams

Visual Paradigm for UML (VP-UML) is a UML design tool and UML CASE tool designed to aid software development. VP-UML supports key modeling standards such as Unified Modeling Language (UML).

**Reason for selecting:** Visual Paradigm's cutting edge GUI allow software analysts to capturing requirements faster, better and easier. If needed detailing use case with flow of events in use case description and subsequently sequence diagrams can be generated automatically based on the descriptions. Understanding complex business workflow and discover use cases in it also provided. Publish work in popular format such as Word and PDF make easy to get drawn diagrams

## 4.2 Frameworks

### 4.2.1 Spring

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

**Reason for selecting:** We use Spring because of it has very high security architecture and we can use it useful and easily.

### 4.2.2 Hibernate

Hibernate ORM (Hibernate in short) is an object-relational mapping library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

**Reason for selecting:** Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from manual result set handling and object conversion. Applications using Hibernate are portable to supported SQL databases with little performance overhead.

### 4.2.3 Bootstrappe

**Reason for selecting:** Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. And Bootstrap is completely free to download.

### 4.2.4 Font Awesome

**Reason for selecting:** Font Awesome icons have used considering its Neat Clean Modern Styling, no white edge, so work well on a variety of backgrounds. And it is very fast and efficient. Also easy to use.

### 4.2.5 AngularJS

AngularJS is a relatively new JavaScript framework by Google, designed to make your front-end development as easy as possible. There are plenty of frameworks and plugins available. As such, it can sometimes prove difficult to sift through all of the noise to find useful tools.

**Reason for selecting:** AngularJS is very light and using it we can get real time form validations (double data-binding).

## 4.3 business process flow

**Elephant**

Detected

Update DB

Notify Village

Send SMS

Access Information

Maintaining

Problem

Posting

User

Register

Confirm

Admin

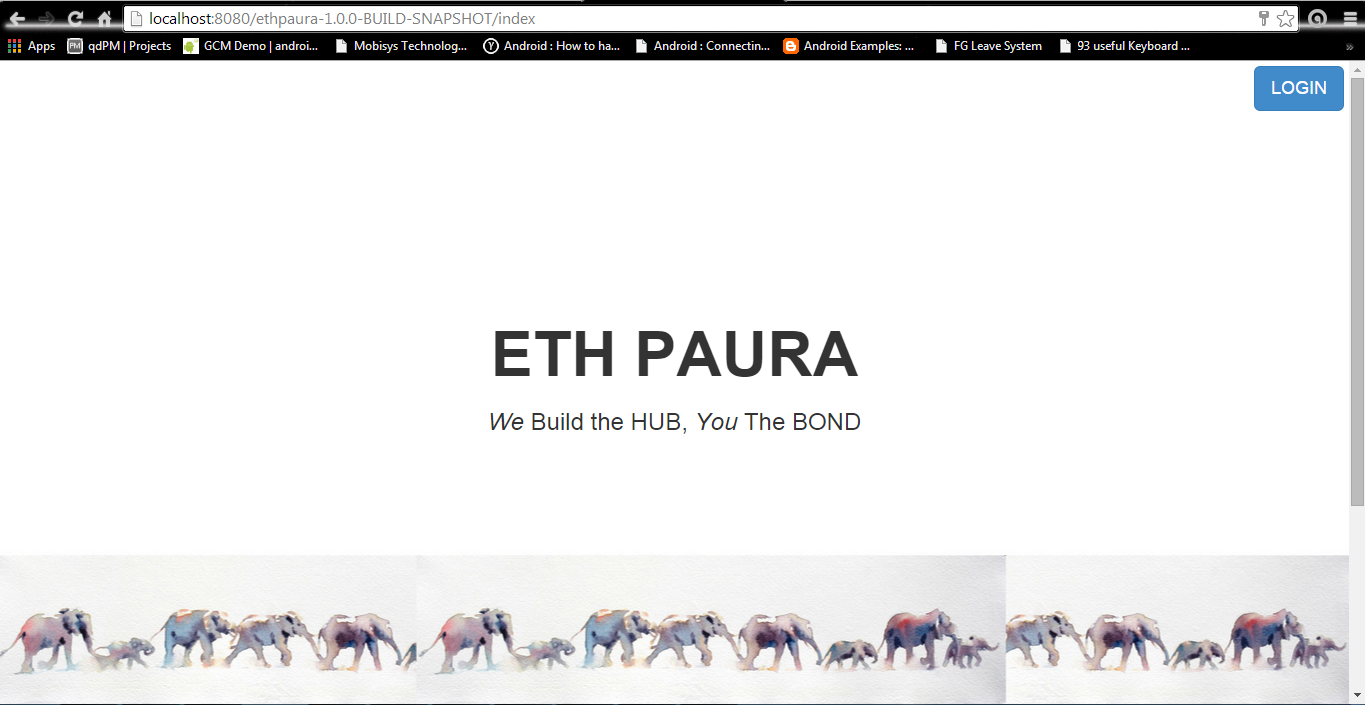
Login

Show in System

Villager

## 4.4 Front end appearance for different system functionalities

### 4.4.1 User Login view of ethPaura Information Hub Project



For the ethPaura Information HUB we use more Security methods. Bcrypt for password encryption and Spring Security for login authentication.

**Authentication user for login**

<authentication-manager erase-credentials=*"false"*>

<authentication-provider>

<password-encoder ref=*"encoder"* />

<jdbc-user-service data-source-ref=*"dataSource"*

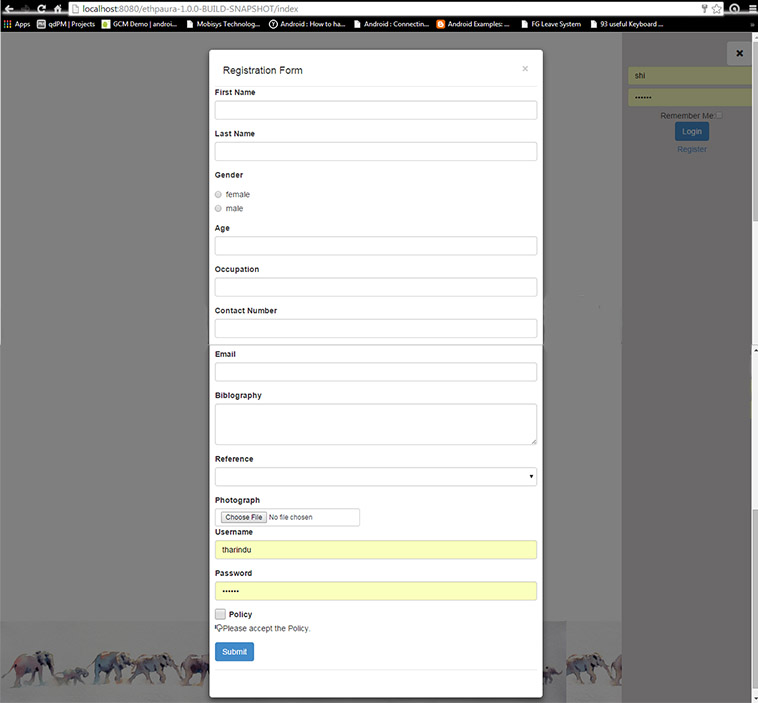
users-by-username-query=*"select username,password,enabled from user where username=?"*

authorities-by-username-query=*"select username, role from user where username =? "* />

</authentication-provider>

</authentication-manager>

### 4.4.2 User Registration form



**Controller for the user register**

@RequestMapping(value = "/addDetails", method = RequestMethod.*POST*)

**public** String addUser(

@ModelAttribute(value = "user") @Valid UserModel user,

BindingResult result) **throws** IOException {

UserEntity userEntity = **new** UserEntity();

userEntity.setAge(user.getAge());

userEntity.setBiblography(user.getBiblography());

userEntity.setEmail(user.getEmail());

userEntity.setFname(user.getFname());

userEntity.setGender(user.getGender());

userEntity.setLname(user.getLname());

userEntity.setOccupation(user.getOccupation());

userEntity.setPassword(PasswordEncoderGenerator.*getHash*(user

.getPassword()));

userEntity.setPhoneNo(user.getPhoneNo());

userEntity.setPolicy(user.getPolicy());

userEntity.setReference(user.getReference());

userEntity.setUsername(user.getUsername());

**if** ((user.getPhotofile().getOriginalFilename() != **null**)

&& (user.getPhotofile().getOriginalFilename() != "")) {

MultipartFile multipartFile = user.getPhotofile();

userEntity.setPhoto(user.getPhotofile().getOriginalFilename());

File outputfile = **null**;

String fileName = multipartFile.getOriginalFilename();

String extensionOfFileName = fileName.substring(

fileName.indexOf(".") + 1, fileName.length());

outputfile = **new** File("C:\\Image\\" + fileName);

File thoutputfile = **null**;

thoutputfile = **new** File("C:\\Image\\thumb\\" + fileName);

BufferedImage thumb = ImageIO.*read*(multipartFile.getInputStream());

**double** percent = 0.8;

**double** per = 0.5;

**int** scaledWidth = (**int**) (thumb.getWidth() \* percent);

**int** scaledHeight = (**int**) (thumb.getWidth() \* per);

// creates output image

BufferedImage outputImage = **new** BufferedImage(scaledWidth,

scaledHeight, thumb.getType());

// scales the input image to the output image

Graphics2D g2d = outputImage.createGraphics();

g2d.drawImage(thumb, 0, 0, scaledWidth, scaledHeight, **null**);

g2d.dispose();

// extracts extension of output file

String thextensionOfFileName = fileName.substring(

fileName.indexOf(".") + 1, fileName.length());

// writes to output file

ImageIO.*write*(outputImage, thextensionOfFileName, thoutputfile);

**if** (**null** != extensionOfFileName

&& extensionOfFileName.equalsIgnoreCase("png")) {

BufferedImage imBuff = ImageIO.*read*(multipartFile

.getInputStream());

ImageIO.*write*(imBuff, extensionOfFileName, outputfile);

} **else** **if** (**null** != extensionOfFileName

&& extensionOfFileName.equalsIgnoreCase("jpg")) {

BufferedImage imBuff = ImageIO.*read*(multipartFile

.getInputStream());

ImageIO.*write*(imBuff, extensionOfFileName, outputfile);

} **else** {

System.*out*.println("Unknown file extension"

+ extensionOfFileName);

}

} **else** {

userEntity.setPhoto("prof.jpg");

}

**if** (result.hasErrors()) {

**return** "redirect:/index";

} **else** {

user.setPassword(PasswordEncoderGenerator.*getHash*(user

.getPassword()));

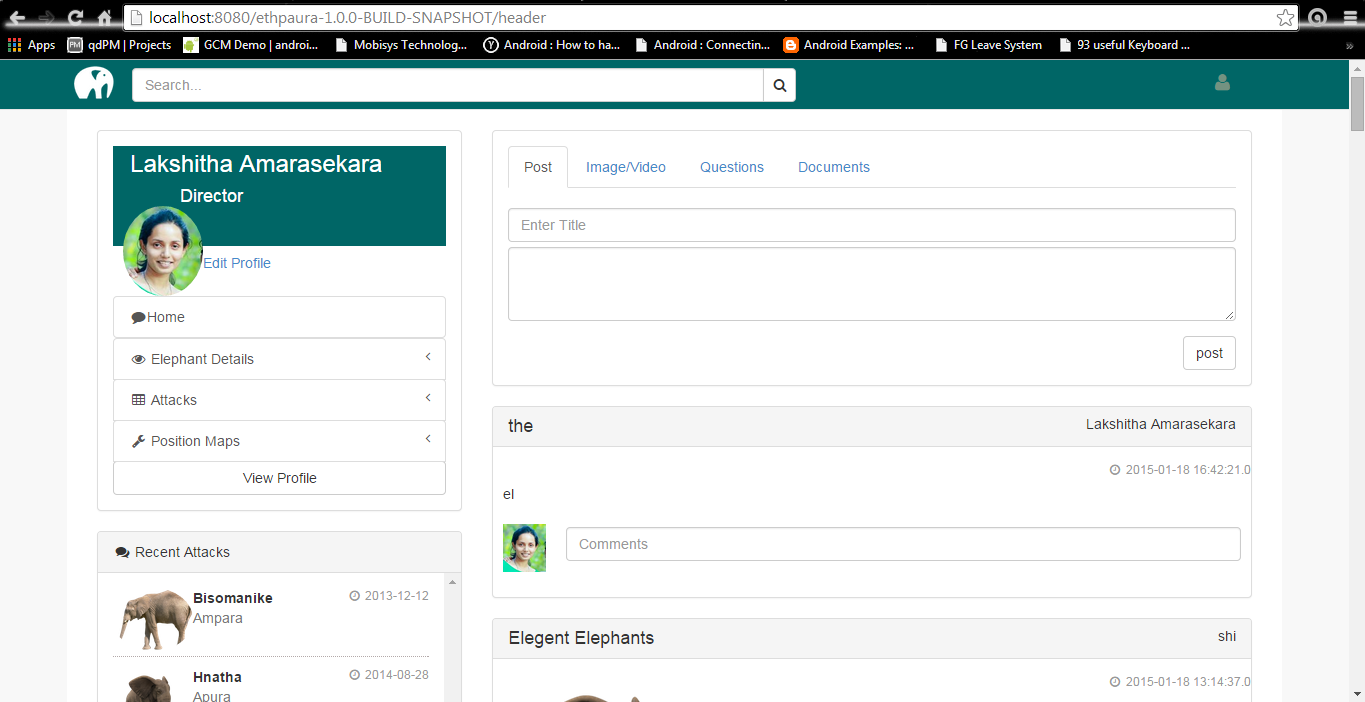
userManager.addUser(userEntity);

**return** "redirect:/index";

}

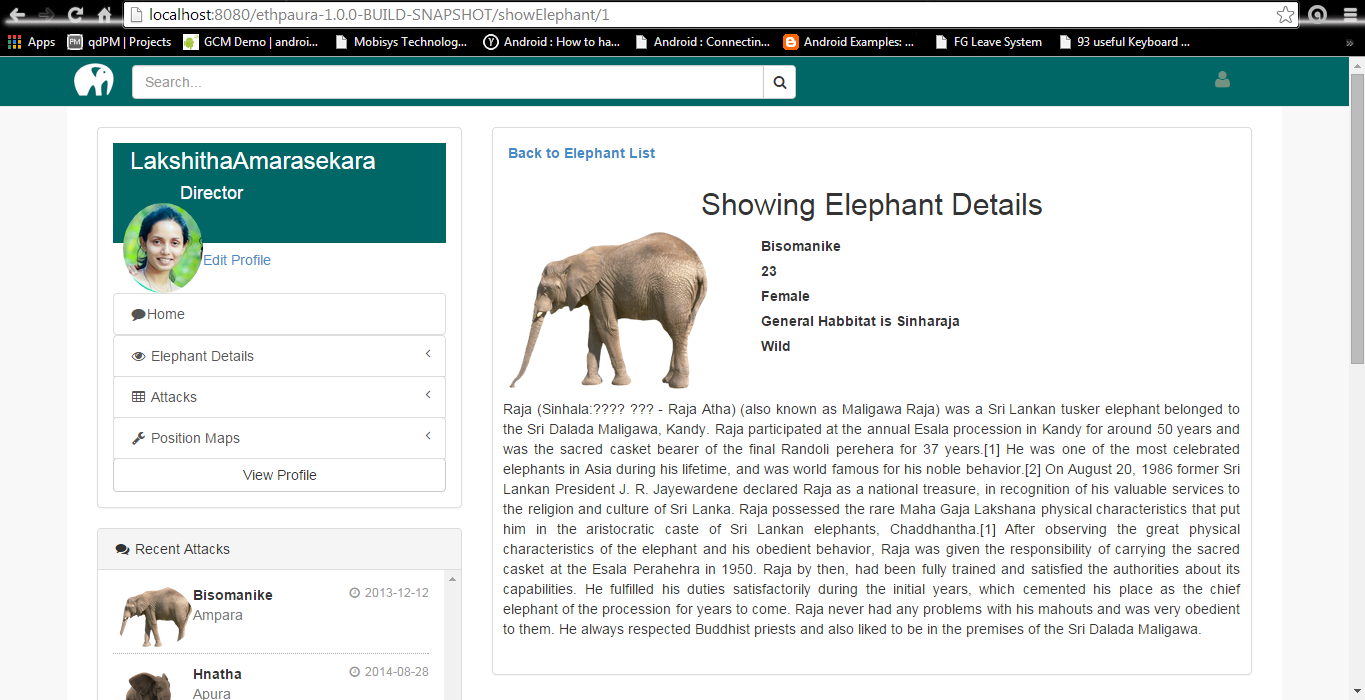
}

### 4.4.3 User Dashboard

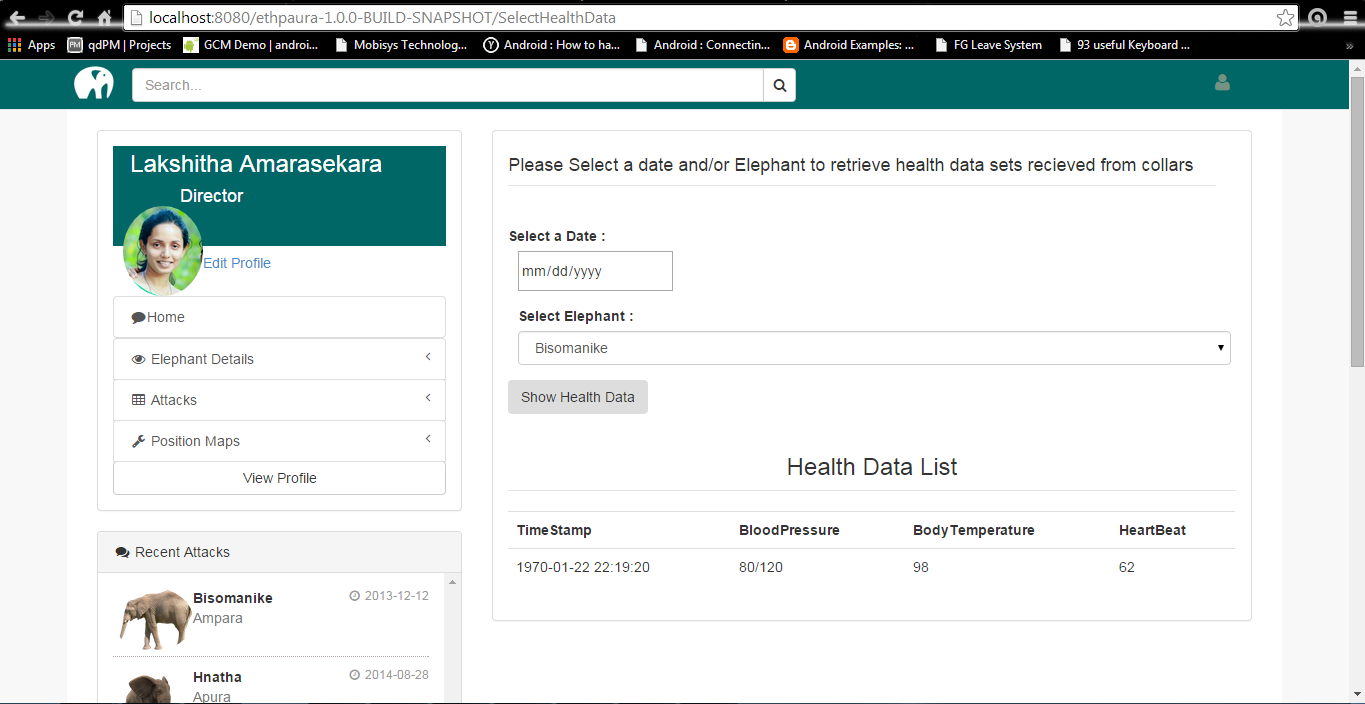


In the dashboard, it has Add Post, Add Image, Add Question and Add Document Functions. For the Images, Video and Documents, We upload it in to the server and save file name in database.

### 4.4.4 Elephant Detail



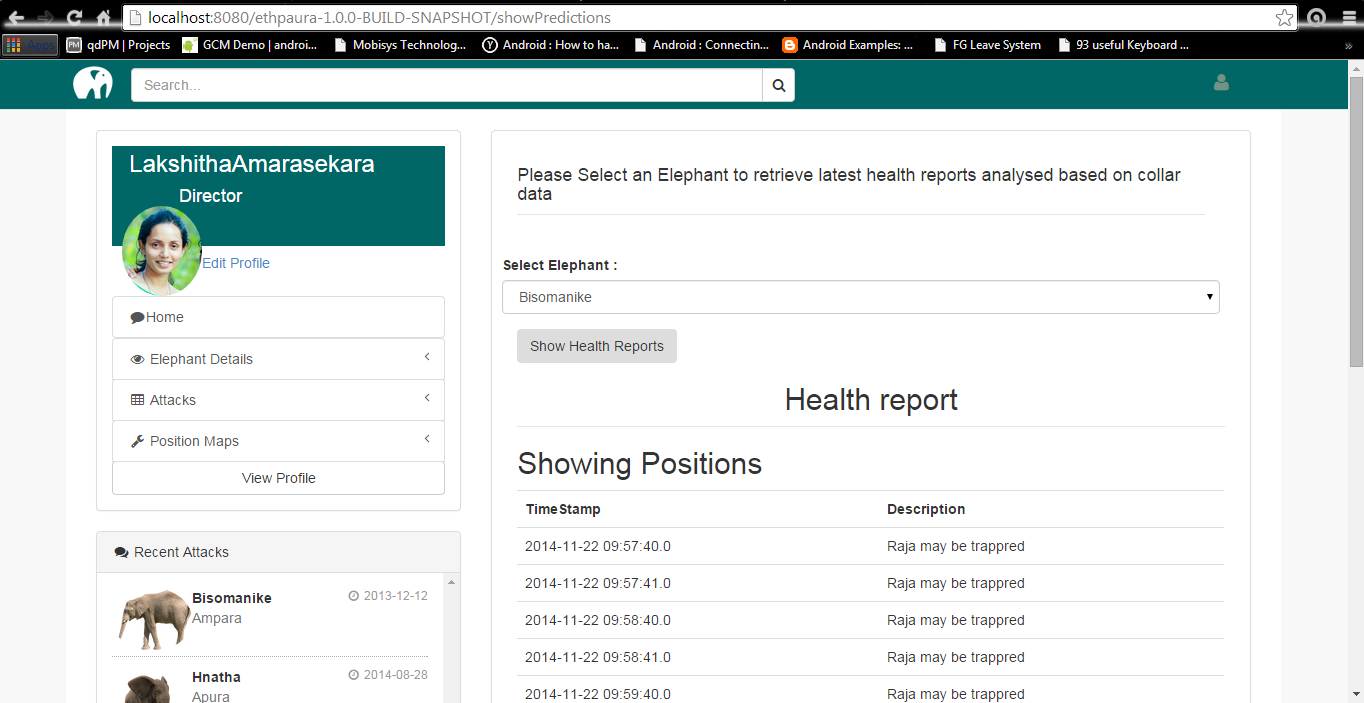
### 4.4.5 Elephant Health data



This view shows the Elephant health data without prediction. This data will send to server directly from the Elephant collar which is setup on few of elephant for the research.

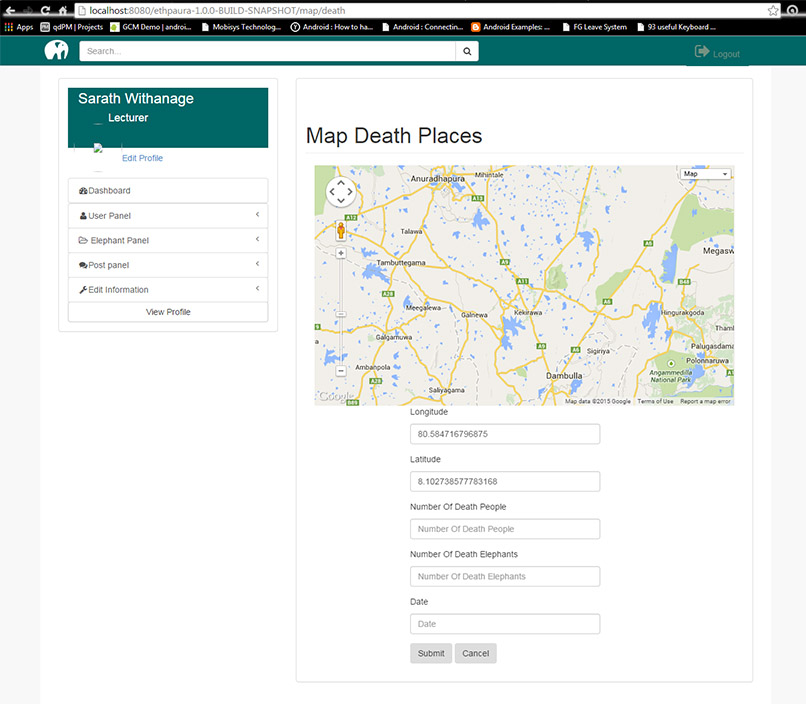
For the development we developed a simulator of collar by using socket programming. Server is open the port to client which is collar. Then send data in frequent time.

### 4.4.6 Elephant Health Report View

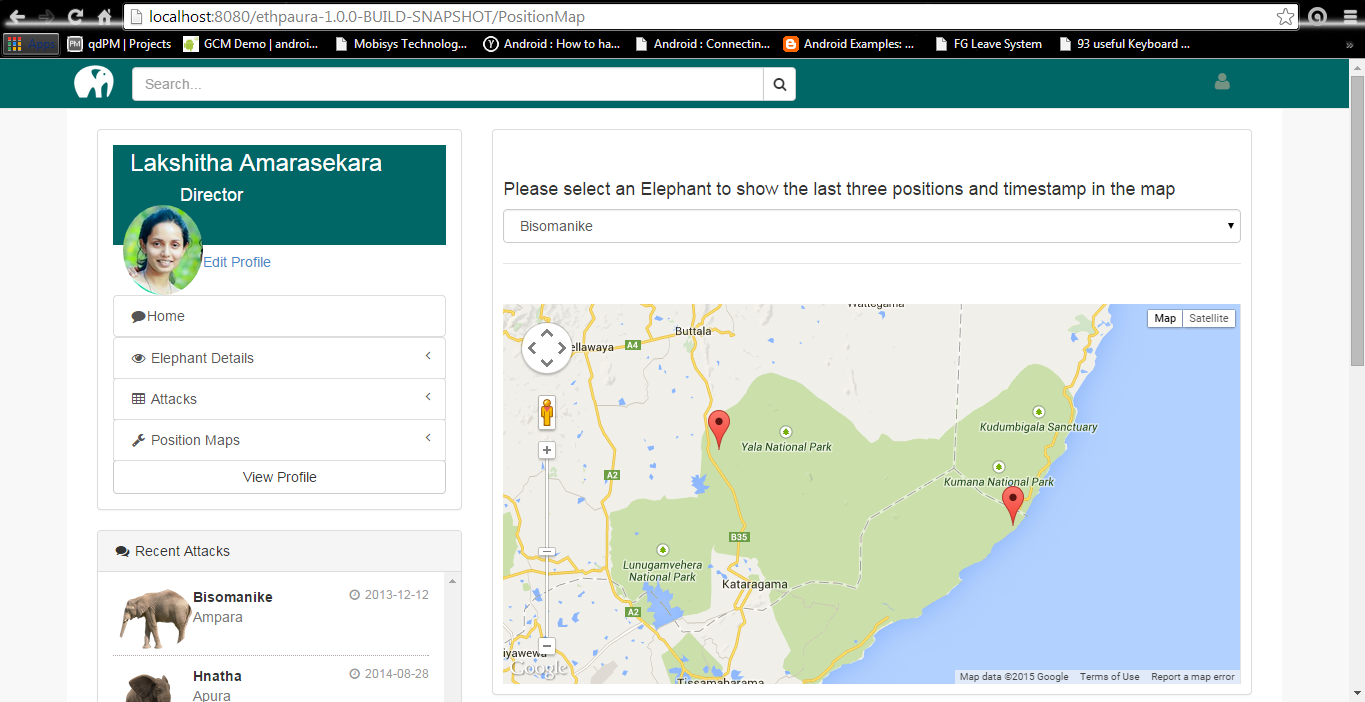


Using algorithms we do some prediction and save that predicted data in our server. Using this, we send alert to wild life officers and doctors, if there are some critical thing.

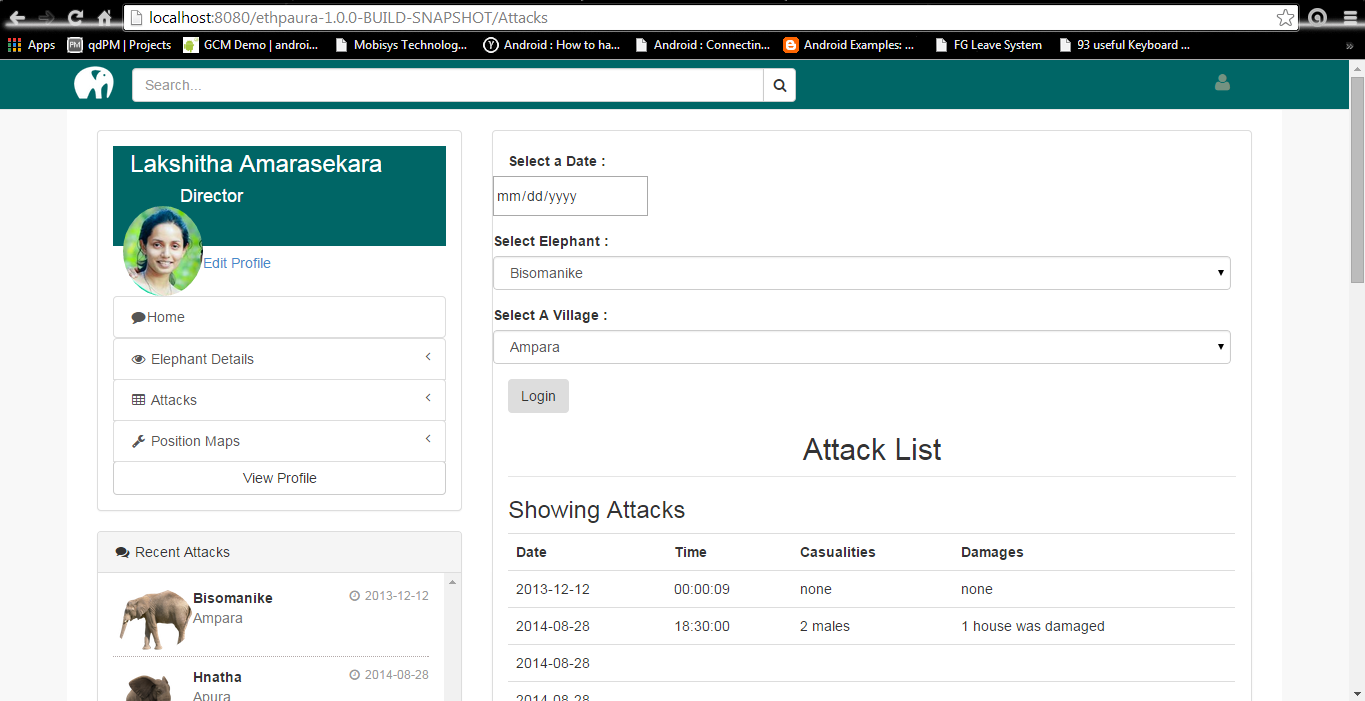
### 4.4.7 Map of death places



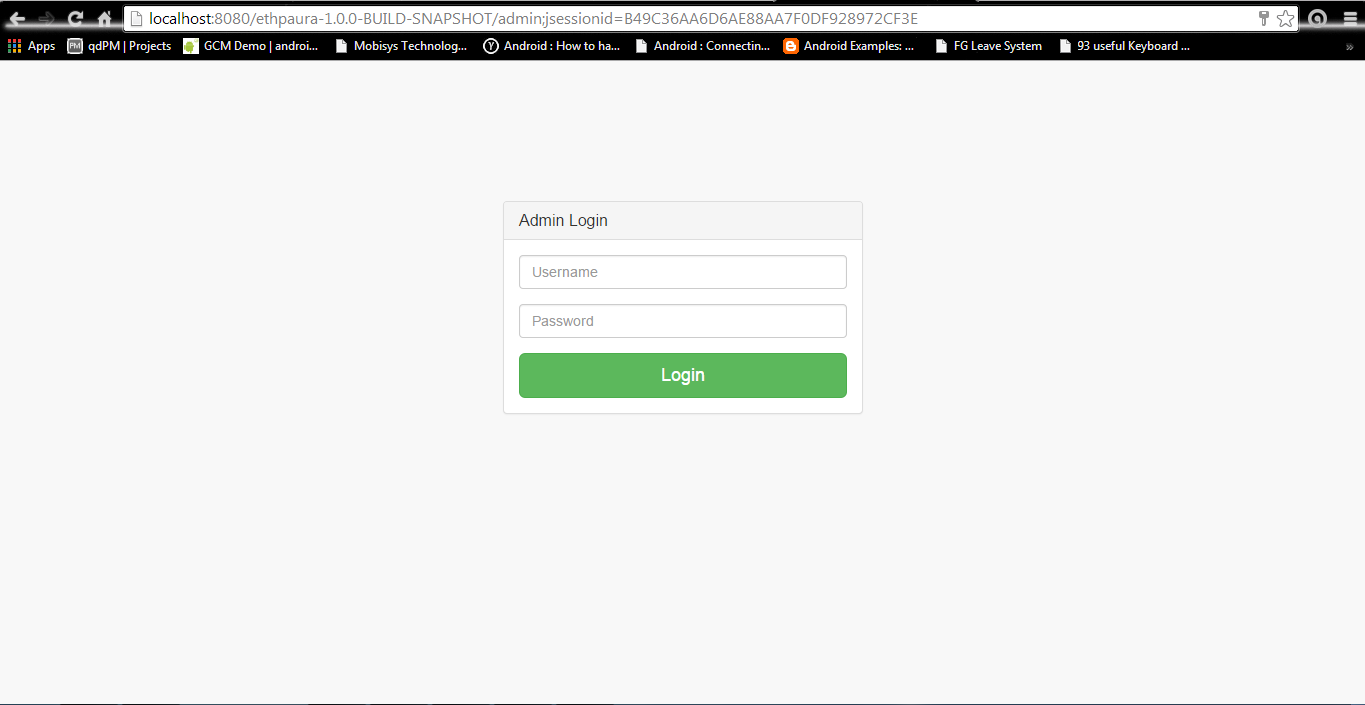
### 4.4.8 Map of Elephant Positions



### 4.4.9 Recent Attacked places

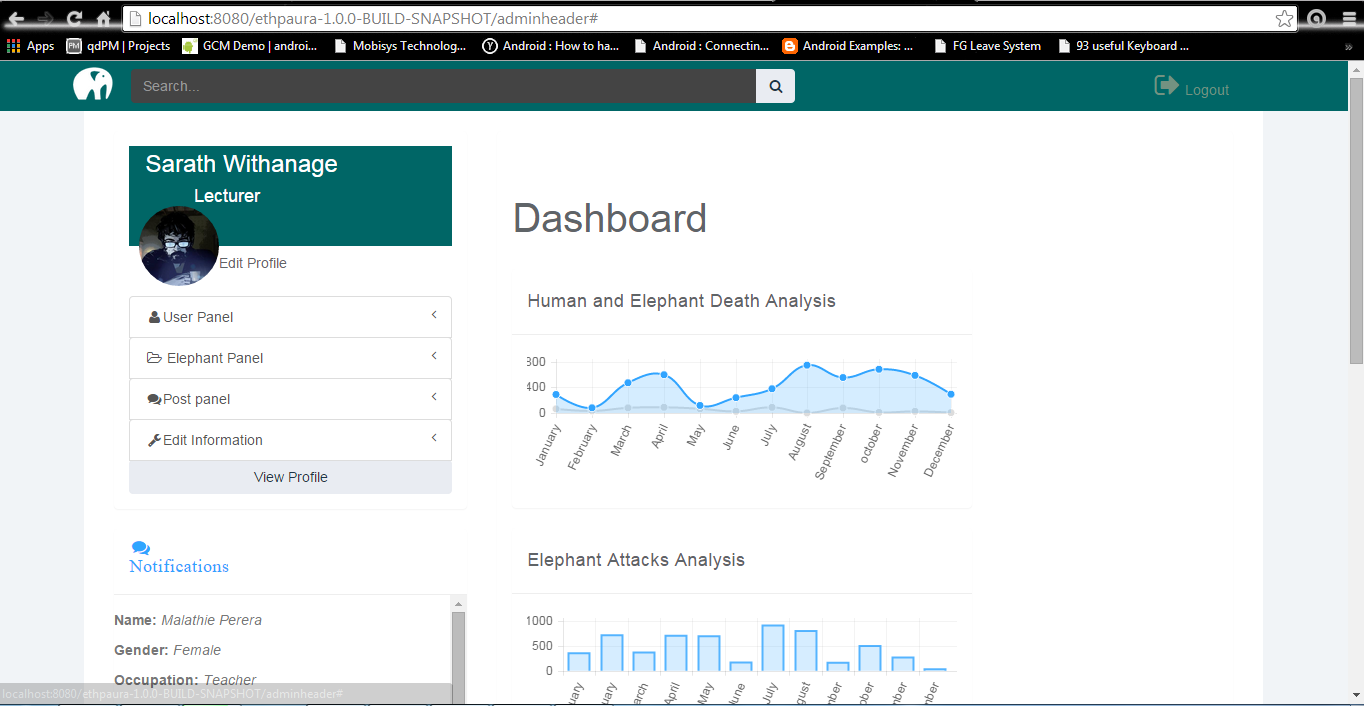


### 4.4.10 Admin Login



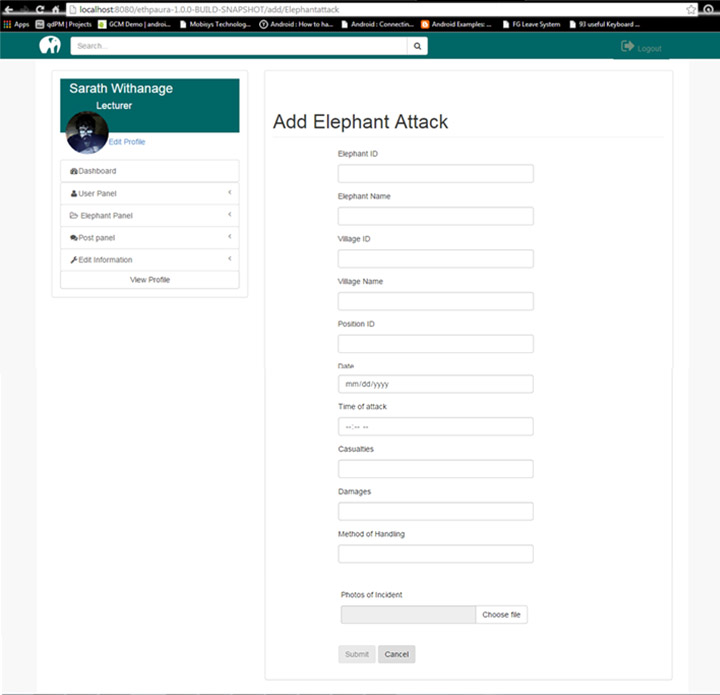
Same as User login, we use spring security and Bcrypt for login and password encryption. And build another war deploy it in server. For the admin login we provide different url and use same database.

### 4.4.11 Admin Dashboard



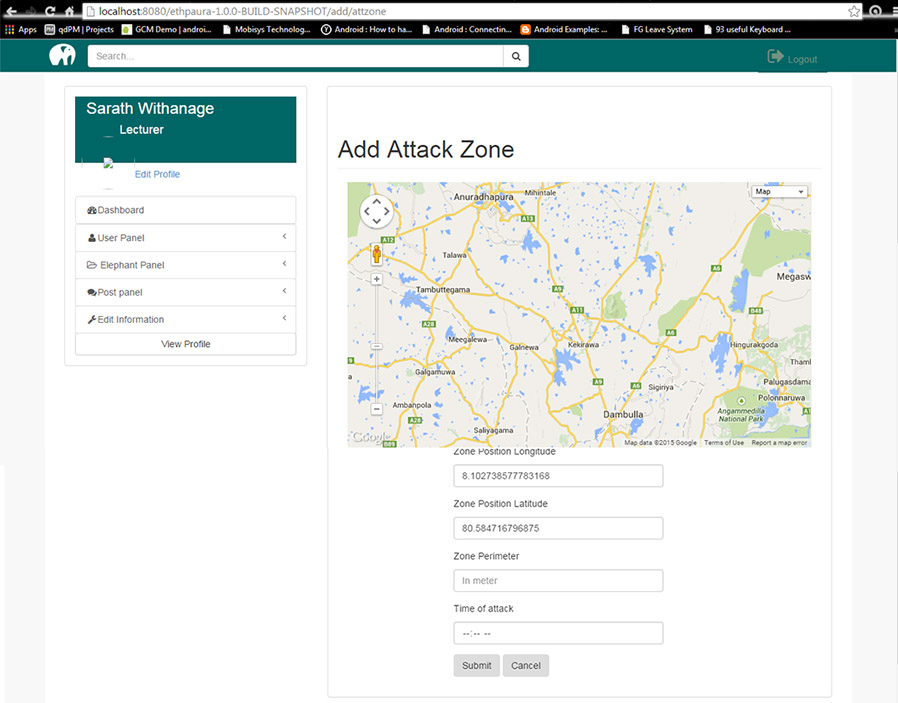
In this dashboard we use some of charts, which is running by java script. To pass data to script, we use JSON services.

### 4.4.12 Add Attack



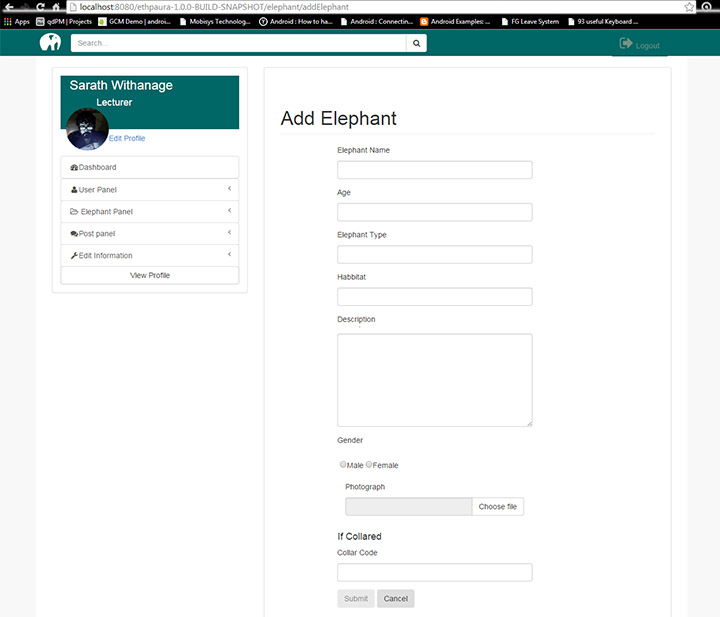
In every form in this application we use POST method to send data from form to controller.

### 4.4.13 Add Attack zone

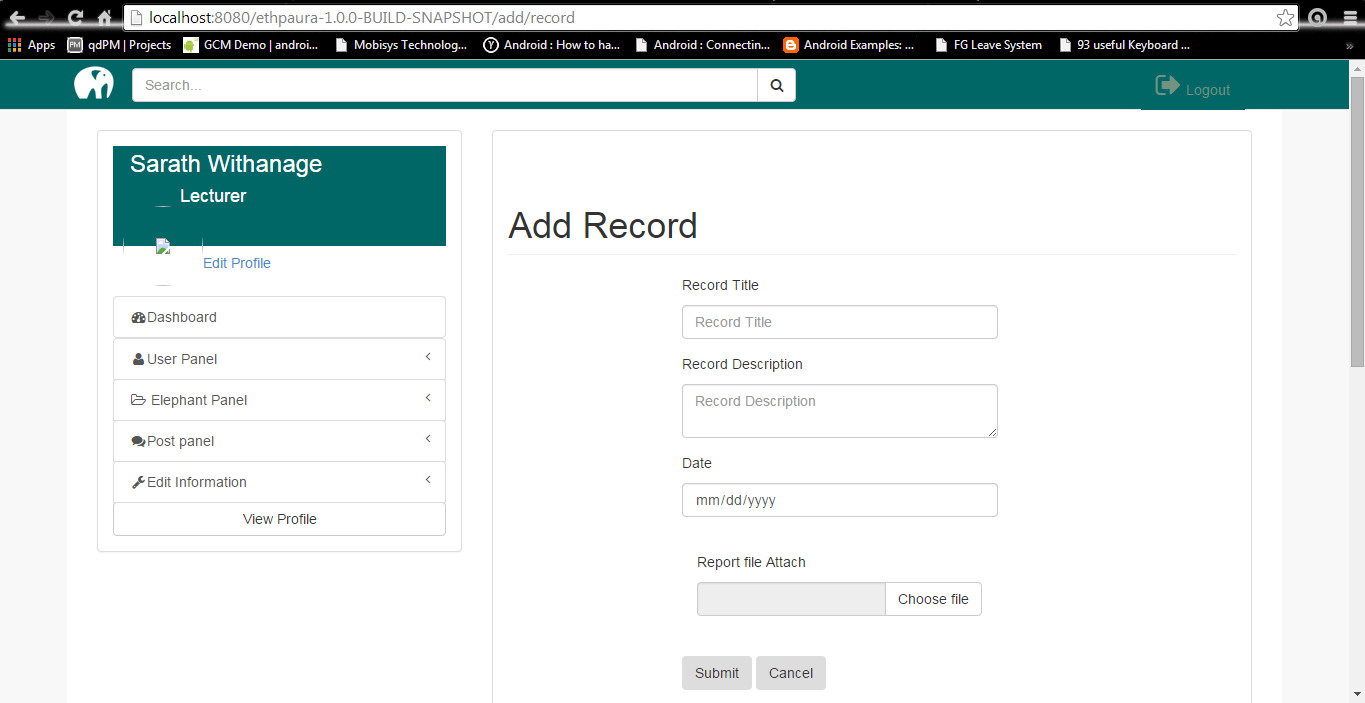


Use Google map version 3 and java script to when click on map to get longitude and latitude.

### 4.4.14 Add Elephant view

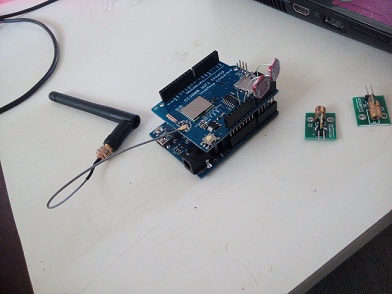


### 4.4.15 Add Record view



## 4.5 device implementation

### 4.5.1 Arduino device and WIFI shield



For the Detection System we used Arduino and the four laser beams. As an arduino we get LAKDUINO board and it is a Sri Lankan Product. Arduino is used for detection mechanism of laser beams interrupted. We placed four LDR for the detect laser beams. Four Laser beams need 5v for an each one.

After detect the elephant is entering to the Village then we need to send Alert to village. For that we used WIFI shield. For the development we used CC3000 WIFI Shield.

// Include required libraries

**#include** <Adafruit\_CC3000.h>

**#include** <ccspi.h>

**#include** <SPI.h>

**#include** <string.h>

**#include** "utility/debug.h"

**#include**<stdlib.h>

// Define CC3000 chip pins

**#define** ADAFRUIT\_CC3000\_IRQ 3

**#define** ADAFRUIT\_CC3000\_VBAT 5

**#define** ADAFRUIT\_CC3000\_CS 10

**char** buffer[10];

// WiFi network

**#define** WLAN\_SSID "local"

**#define** WLAN\_PASS "rilekez5"

**#define** WLAN\_SECURITY WLAN\_SEC\_WPA2

**const** **unsigned** **long** dhcpTimeout = 60L \* 1000L, // Max time to wait for address from DHCP

connectTimeout = 15L \* 1000L, // Max time to wait for server connection

responseTimeout = 15L \* 1000L; // Max time to wait for data from server

uint32\_t t;

Adafruit\_CC3000 cc3000 = Adafruit\_CC3000(ADAFRUIT\_CC3000\_CS,

ADAFRUIT\_CC3000\_IRQ, ADAFRUIT\_CC3000\_VBAT, SPI\_CLOCK\_DIV2);

// PHP's server IP, port, and repository

uint32\_t ip = cc3000.IP2U32(192, 168, 211, 1);

**int** port = 8888;

String repository = "/wifi/";

Adafruit\_CC3000\_Client client;

**void** **setup**(**void**) {

Serial.begin(56700);

//digitalWrite(13, HIGH);

Serial.print(F("Initializing..."));

**if** (!cc3000.begin()) {

Serial.println(F("failed. Check your wiring?"));

**return**;

}

Serial.print(F("OK.\r\nConnecting to network..."));

cc3000.connectToAP(WLAN\_SSID, WLAN\_PASS, WLAN\_SECURITY);

Serial.println(F("connected!"));

Serial.print(F("Requesting address from DHCP server..."));

**for** (t = millis(); !cc3000.checkDHCP() && ((millis() - t) < dhcpTimeout);

delay(500)) {

Serial.println("....waiting");

}

**if** (cc3000.checkDHCP()) {

Serial.println(F("OK"));

} **else** {

Serial.println(F("failed"));

**return**;

}

randomSeed(analogRead(0));

}

**void** **loop**(**void**) {

**if** ((analogRead(0) < 500) && (analogRead(1) < 500) && (analogRead(2) > 500)

&& (analogRead(3) > 500)) {

Serial.println("entering....");

delay(500);

**if** ((analogRead(3) < 500) && (analogRead(0) < 500)

&& (analogRead(1) < 500)) {

Serial.println("Elephant Entered....");

pinMode(13, OUTPUT);

digitalWrite(13, LOW);

pinMode(12, OUTPUT);

digitalWrite(12, LOW);

delay(3000);

//Open Socket

Serial.println("...Connecting to server");

t = millis();

**do** {

client = cc3000.connectTCP(ip, port);

} **while** ((!client.connected()) && ((millis() - t) < connectTimeout));

// Send request

**if** (client.connected()) {

Serial.println("Connected");

String request = "GET " + repository

+ "test.php?dir=west&village=mihintale";

Serial.print("...Sending request:");

Serial.println(request);

send\_request(request);

} **else** {

Serial.println(F("Connection failed"));

**return**;

}

Serial.println("...Reading response");

show\_response();

Serial.println(F("Cleaning up..."));

Serial.println(F("...closing socket"));

client.close();

//wait 10000 before sending to the PHP service.

delay(10000);

}

}

**else** {

Serial.println(F("...no elephant...."));

pinMode(13, OUTPUT);

digitalWrite(13, HIGH);

pinMode(12, OUTPUT);

digitalWrite(12, HIGH);

}

}

bool **send\_request**(String request) {

// Transform to char

**char** requestBuf[request.length() + 1];

request.toCharArray(requestBuf, request.length());

// Send request

**if** (client.connected()) {

client.fastrprintln(requestBuf);

} **else** {

Serial.println(F("Connection failed"));

**return** false;

}

**return** true;

free(requestBuf);

}

**void** **show\_response**() {

**while** (client.available()) {

// Read answer and print to serial debug

**char** c = client.read();

Serial.print(c);

}

}

// Float to String conversion

String **floatToString**(**float** number) {

dtostrf(number, 5, 2, buffer);

**return** String(buffer);

}

**char** **timedRead**(**void**) {

**unsigned** **long** start = millis();

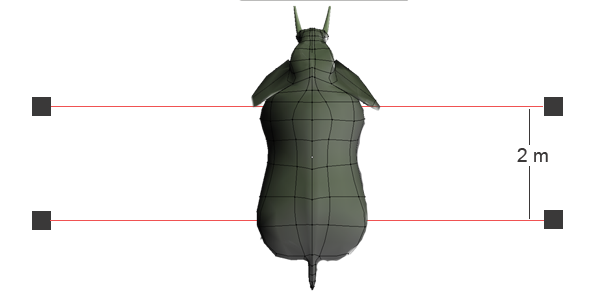
**while** ((!client.available()) && ((millis() - start) < responseTimeout))

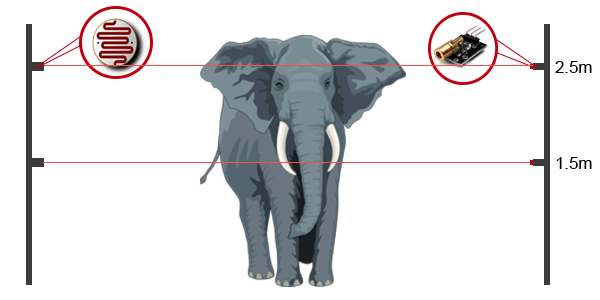
;

**return** client.read(); // -1 on timeout

}

### 4.5.2 Laser fence





Above Pictures will show how the laser beams will placed.

### 4.5.3 Village device

In village device we used the Computer which have WIFI Adapter and Internet Connection. Using WIFI Adapter it can connect to the Arduino device and show the defense is active now. And the Internet connection is used to Send SMS to villages and update database.

Also village device need to run web server on background. We use Apache server.

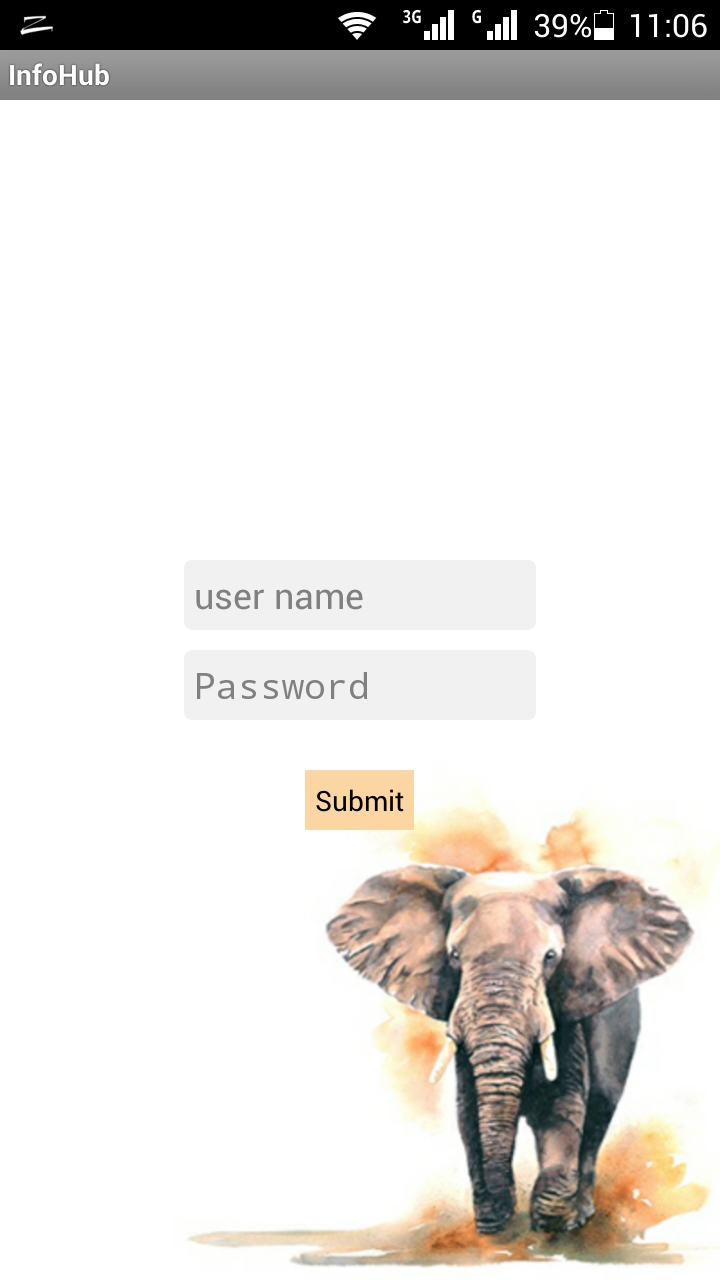
## 4.6 Android application development

In the android application, we have to use two extra libraries.

* Googleplay services library
* ABS library

Gooleplay is used to view maps in android application and ABS library is used to slide menu of application, which is very easy to navigate application

### 4.6.1 Login view



User and admin can login using same application. If login person is a user app will navigate to user panel. If admin, it will navigate to admin panel.

To get data from server we use JSON. We create JSON services using spring and called that data to get data.

**public** **static** String readJsonFromUrl(String url) {

String htmlResponce = **null**;

HttpGet httpGet = **new** HttpGet(url);

HttpClient httpClient = **new** DefaultHttpClient();

**try** {

HttpResponse httpResponse = httpClient.execute(httpGet);

StatusLine statusLine = httpResponse.getStatusLine();

**if** (statusLine.getStatusCode() == 200) {

HttpEntity httpEntity = httpResponse.getEntity();

htmlResponce = EntityUtils.*toString*(httpEntity);

}

} **catch** (ClientProtocolException e) {

Log.*e*(CommonService.**class**.getName(), e.getMessage());

**return** **null**;

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

Log.*e*(CommonService.**class**.getName(), "" + e.getMessage());

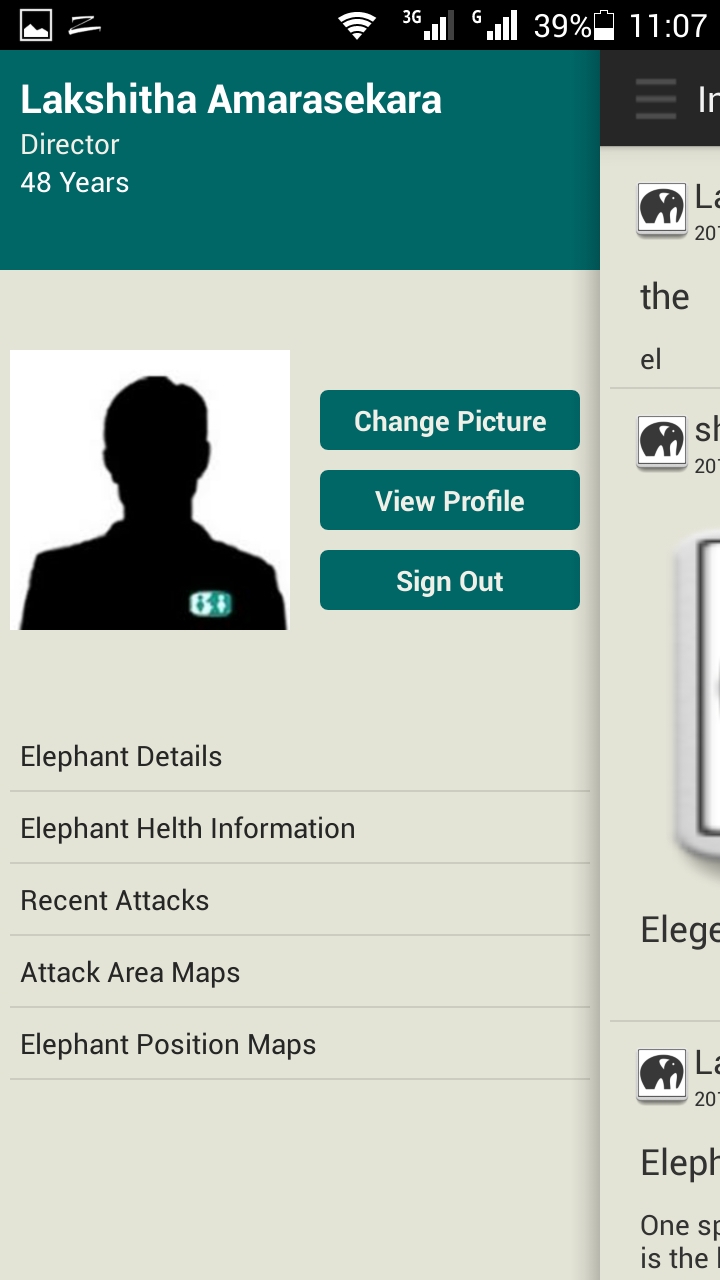
**return** **null**;

}

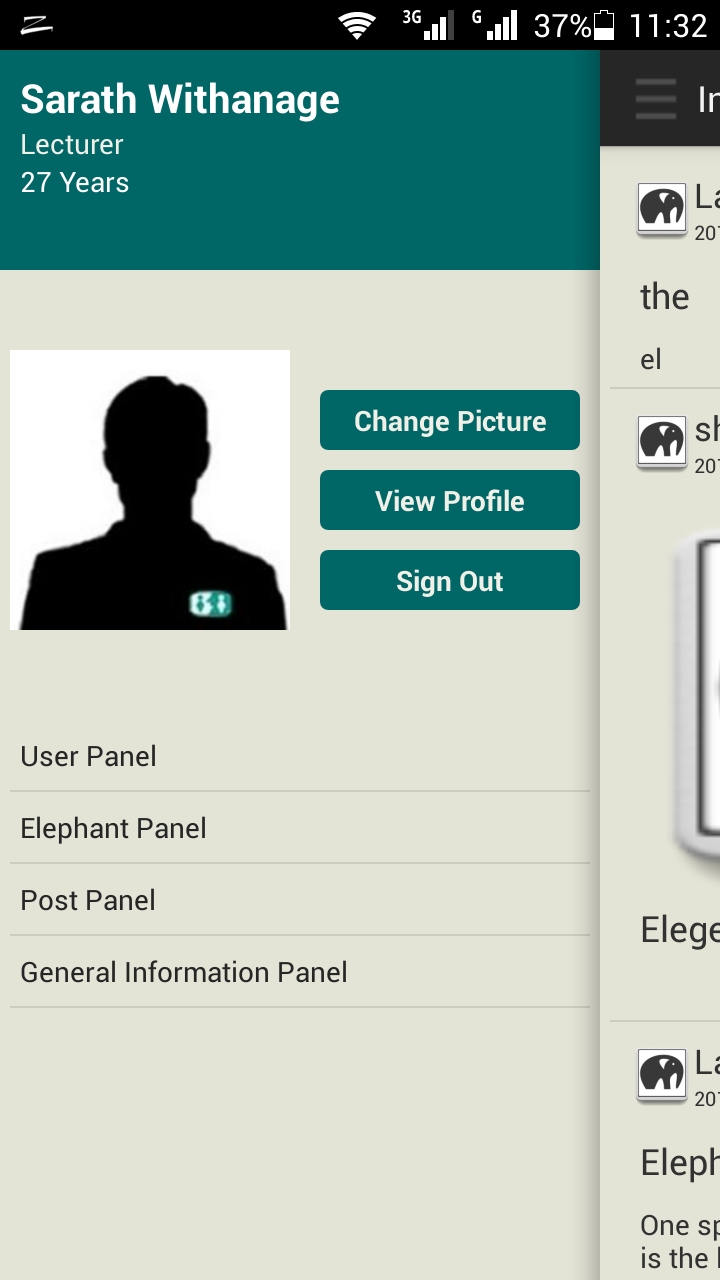
**return** htmlResponce;

}

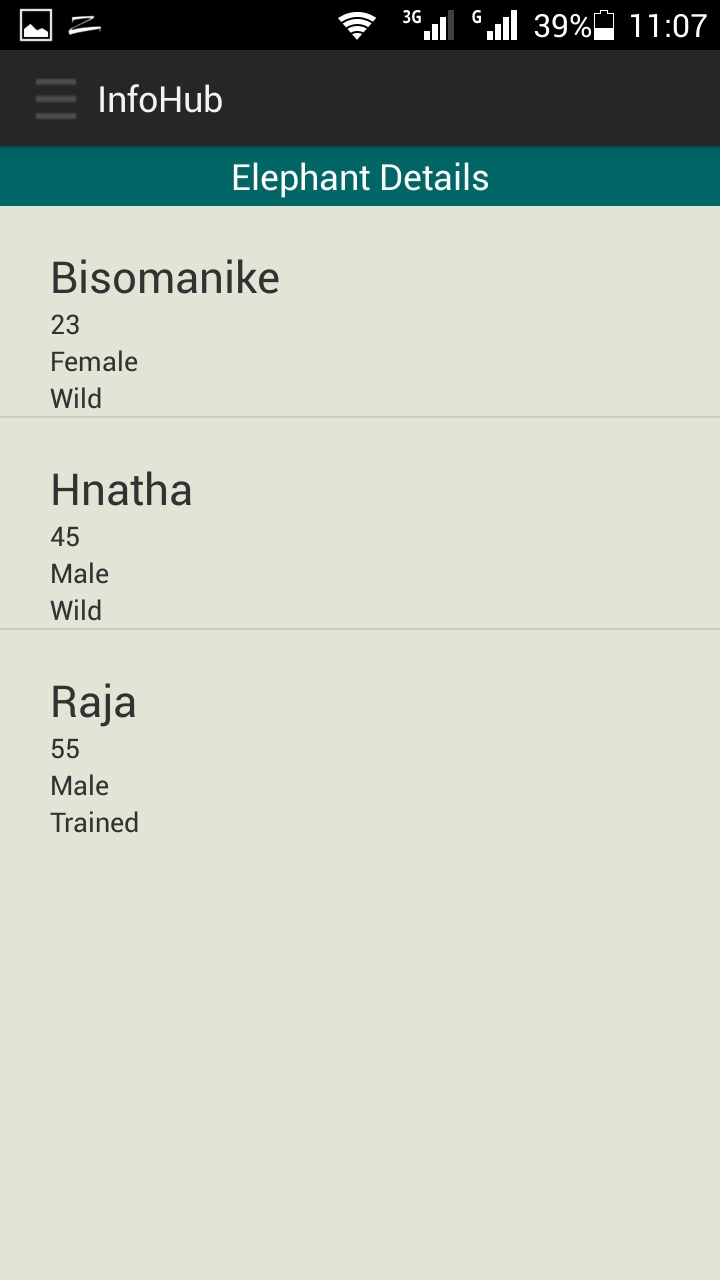
### 4.6.2 Admin Panel



### 4.6.3 User Panel



### 4.6.4 Elephant list



We get elephant list in to a ListView. Using BaseAdaper we create class called “AdapterList.java”. Using this class we set data to ListView.

**public** **class** AdapterList **extends** BaseAdapter {

**private** **int** type;

**private** ArrayList<BaseElement> items;

**private** Activity activity;

**private** Fragment fragment;

**private** LayoutInflater inflater;

**private** ImageLoader imageLoader;

/\*\*

\* **@param** type

\* **@param** items

\* **@param** activity

\* **@param** fragment

\*/

**public** AdapterList(**int** type, ArrayList<BaseElement> items,

Activity activity, Fragment fragment) {

**super**();

**this**.type = type;

**this**.items = items;

**this**.activity = activity;

**this**.fragment = fragment;

imageLoader = **new** ImageLoader(activity);

inflater = (LayoutInflater) **this**.activity

.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);

}

@Override

**public** **int** getCount() {

// **TODO** Auto-generated method stub

**return** items.size();

}

@Override

**public** Object getItem(**int** position) {

// **TODO** Auto-generated method stub

**return** position;

}

@Override

**public** **long** getItemId(**int** position) {

// **TODO** Auto-generated method stub

**return** position;

}

@Override

**public** View getView(**int** position, View convertView, ViewGroup parent) {

View view = **null**;

**if** (type == Elements.*ELEPHANT\_DETAIL*.getElement()) {

view = inflater.inflate(R.layout.*elephant\_detail\_layout*, **null**);

Elephant elephant = (Elephant) items.get(position);

TextView text = (TextView) view.findViewById(R.id.*name*);

TextView text2 = (TextView) view.findViewById(R.id.*elephantage*);

TextView text3 = (TextView) view.findViewById(R.id.*elephantgender*);

TextView text4 = (TextView) view.findViewById(R.id.*elephanttype*);

text.setText(elephant.getName());

text2.setText(elephant.getAge());

text3.setText(elephant.getGender());

text4.setText(elephant.getType());

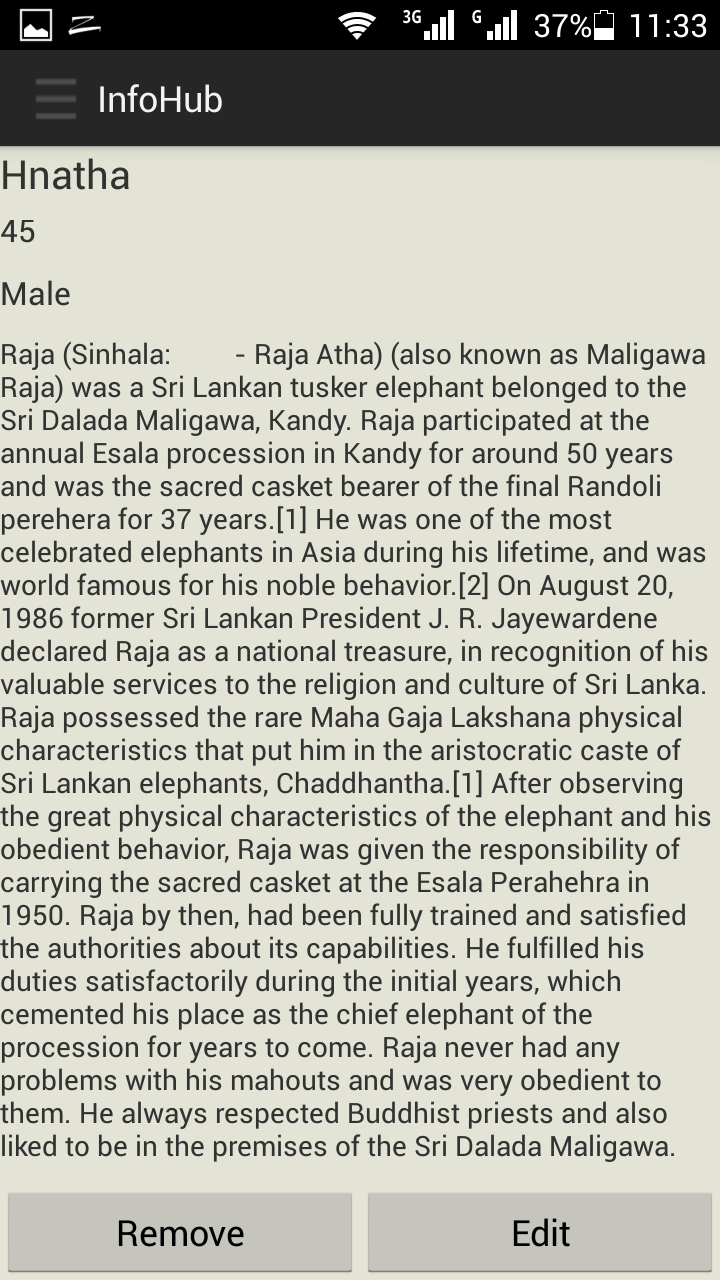
}

**return** view;

}

}

### 4.6.5 Elephant detail



**Chapter 5**

# 5 Testing and result

## 5.1 test scenarios OF Information hub

### 5.1.1 User Login

Title User Login

Description Login to information HUB as user

Input Username: shi

Password: 12456

Expected result after click login button user will redirect to Dashboard

Actual result User is been able to login system and redirect to Dashboard

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to http://ethpaura.lk | Open the login view |
| Step 3 | Enter the username and password |  |
| Step 4 | Enter login button | Redirect to Dashboard |

### 5.1.2 Admin Login

Title Admin Login

Description Login to “Information HUB” as admin

Input Username: Sarath

Password: 111111

Expected result after click login button admin will redirect to admin Dashboard

Actual result Admin is been able to login system and redirect to admin Dashboard

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to http://ethpaura.lk/admin | Open the login view |
| Step 3 | Enter the username and password |  |
| Step 4 | Enter login button | Redirect to Dashboard |

### 5.1.3 Add Elephant

Title Add Elephant

Description Add new Elephant to System without Collar

Input Elephant Name: Bisomanike

Age: 11

Elephant type: Wild

Habitat: \_

Description lives in Yala

Gender: male

Photograph: bis.png

Collar ID \_

Expected result after click Submit button admin will redirect to same page and Add elephant to database.

Actual result Elephant is added to table

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to http://ethpaura.lk/admin | Open login view |
| Step 3 | Login to the System | Redirect to the Dashboard |
| Step 4 | Click Elephant panel | Open Sub menu |
| Step 5 | Click Add Elephant | Open Add Elephant Form |
| Step 6 | Insert the data |  |
| Step 7 | Click Submit button | Redirect to same page |

### 5.1.4 Register User

Title Register User

Description Register a user with details.

Input First Name: Tharindu

Last Name: Thilanka

Gender: male

Age: 25

Occupation: Student

Contact Number: 0716163989

Email: tharindupro@gmail.com

Bolograph: \_

Reference: Sarath

Photograph: bis.png

Username: TharinduPro

Password: user@1234

Expected result after click Submit button admin will redirect to same page and Add elephant to database.

Actual result Elephant is added to table

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to <http://ethpaura.lk/admin> | Open login view |
| Step 3 | Login to the System | Redirect to the Dashboard |
| Step 4 | Click Elephant panel | Open Sub menu |
| Step 5 | Click Add Elephant | Open Add Elephant Form |
| Step 6 | Insert the data |  |
| Step 7 | Click Submit button | Redirect to same page |

### 5.1.5 Confirm User by Admin

Title Confirm User

Description Admin have to confirm the new user, who can’t login until admin confirm

Input \_

Expected result Change privilege level of user, and redirect to dashboard.

Actual result Redirect to dashboard and user can login.

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to <http://ethpaura.lk/admin> | Open login view |
| Step 3 | Login to the System | Redirect to the Dashboard |
| Step 4 | Click Confirm button, which is under the username on notification area. | Redirect to dashboard |

### 5.1.6 Add Record

Title Add Recode

Description Add recode by admin. It can upload image, document or video

Input Recode Title: Research Paper

Recode Description: About Asian Elephants.

Date: 2014.12.20

Attach File: paper.doc

Expected result upload paper.doc to server and add user dashboard.

Actual result Added to user dashboard and paper.doc was uploaded to server

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to <http://ethpaura.lk/admin> | Open login view |
| Step 3 | Login to the System | Redirect to the Dashboard |
| Step 4 | Click Edit Information | Open Sub menu |
| Step 5 | Click Add Recode | Open Add Recode Form |
| Step 6 | Insert the data |  |
| Step 7 | Click Submit button | Redirect to same page |

### 5.1.7 Add Attack zone

Title Add Attack zone

Description Add elephant attack zone to the system

Input Longitude: 8.012544

Latitude: 80.5422

Zone perimeter: 20

Attack time: 2014.12.20

Expected result Update database

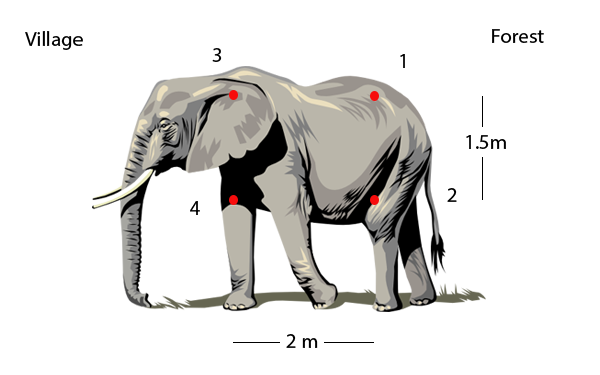
Actual result Database updated successfully

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Browser | The Browser opens |
| Step 2 | Go to <http://ethpaura.lk/admin> | Open login view |
| Step 3 | Login to the System | Redirect to the Dashboard |
| Step 4 | Click Edit Information | Open Sub menu |
| Step 5 | Click Add Attack zone | Open Add Attack zone Form |
| Step 6 | Click on the map to get position | Auto fill the position |
| Step 7 | Insert other field |  |
| Step 8 | Click submit button | Redirect to same page |

## 5.2 test secenarios of device

We use four laser beams to detect elephant breach. And we test all the possibilities. Only the all the four beams cross, then only it is an elephant breach.



Above Diagram will explain how the lasers will placed in fence.

### 5.2.1 First laser cross

Title First Laser cross

Description Somehow only the first leaser is blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.2 Second laser cross

Title Second Laser cross

Description Somehow only the Second leaser is blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.3 Third laser cross

Title Third Laser cross

Description Somehow only the third leaser is blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.4 First laser cross

Title Fourth Laser cross

Description Somehow only the fourth leaser is blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.5 First laser and second cross

Title First Laser and Second cross

Description Somehow both the first and second leaser is blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.6 Third and Fourth laser cross

Title Third and Fourth Laser cross

Description Somehow both the third and fourth leaser is blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.7 All the lasers cross at once

Title All the lasers cross at once.

Description Somehow all the leasers are blocked by something.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.8 First Laser and third Laser cross at once and then second and fourth

Title First and third laser cross at once then second and fourth

Description Somehow First and third leasers are blocked at once, then other two lasers are blocked at once.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.9 First Laser and Fourth Laser cross at once and then second and third

Title First and fourth laser cross at once then second and third

Description Somehow First and fourth leasers are blocked at once, then other two lasers are blocked at once.

Expected result No Elephant Detect.

Actual result No Elephant Detect.

Status Pass

### 5.2.10 First Laser and second Laser cross at once and then third and fourth

Title First and second laser cross at once then third and fourth

Description Somehow First and Second leasers are blocked at once, then other two lasers are blocked at once.

Expected result Elephant Detect and Entering to the village. Send SMS

Actual result Elephant Detect and Entering to the village. SMS received

Status Pass

## 5.3 test secenarios of android application

### 5.3.1 User Login

Title User Login

Description Login to Android Application as user

Input Username: shi

Password: 12456

Expected result after click login button user will redirect to Dashboard

Actual result User is been able to login system and redirect to Dashboard

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Android Application | Application Open |
| Step 3 | Enter the username and password |  |
| Step 4 | Enter login button | Change View to Dashboard |

### 5.3.2 Admin Login

Title Admin Login

Description Login to Android Application as admin

Input Username: Sarath

Password: 111111

Expected result after click login button Admin will go to Dashboard

Actual result User is been able to login system and go to Dashboard

Status Pass

|  |  |  |
| --- | --- | --- |
| Steps | Description | Result |
| Step 1 | Open Android Application | Application Open |
| Step 3 | Enter the username and password |  |
| Step 4 | Enter login button | Change View to Dashboard |

**Chapter 6**

# 6 report summery

## 6.1 conclusion

The Project EthPaura was design to mitigate the prevailing Human – Elephant conflict problem. The approach was to provide effective, efficient timely information to people and to make a understanding among the society, in order to establish good long term solution. The necessary information was gathered from the expert of the field such as Dr. Devaka and Dr. Pruthuwiraj in order to provide a win-win solution for people as well as Elephants. Also a background research was done to get a better understanding about the problem in scientific viewpoint.

The major requirement of the project was to providing a non-invasive, real time elephant detection system apart from that a web application to support elephant researchers and to log information and data which is currently logging manually was also concerned.

The biggest problem was the inefficiency of the existing solution and the systems. The electric fence was proved inefficient since the inability to provide a concrete solution and because of the high electrical consumption.

Also the concerns about elephant rights was raised. The solution for HEC, had the need of providing a non-invasive, scalable, energy efficient, effective and finally an economic features. The technology for detective was decided according to all these constrains. What was chose is a combination of best technologies when the accuracy and the efficiency can be addressed at the same time.

To achieve these qualities what we have used is a simple algorithm to make it less complicate to compute.

## 6.2 recommendation

1. EthPaura provide a novel approach to minimize Human-Elephant conflict.
2. EthPaura should deployed in forest boarders areas provide better information sharing and alert planning to mitigate elephant-human conflict situations.
3. Villagers and forest officers should be given with good understanding about the system and the system functionality.
4. Introduce the information HUB to the wild life ministry a better circulation of information can be achieved.
5. People and elephant interested parties will get benefits.

## 6.3 future work

1. **Improved SMS service.**

Optimizing SMS service to get filter subscribers from central database according to the village the attack accrued.

1. **Official document and booklets for training.**

A user manual like a thing to troubleshoot and configure the system can be given to necessary parties.

1. **Adding ultrasound noise to the system**

Adding ultrasound noise to the prevent elephants from entering the village also being considering at the moment. But because of some social and animal concerns. We need to do a separate experiment research before adding the feature.

1. **Enhancing the web application**

Some enhancement to make the application more user friendly is being considered. Some cosmetic additions are proposed.

# referencess

Wi-alert : a wireless sensor network based intrusion alert prototype for HEC, Ruwini Edirisinghe, Dileeka Dias, Rakhitha Chandrasekara, Lanka Wijesinghe, Prasanga Siriwardena and Prasad Kumara Sampath, n.d.

2011 R.Sukumar J.Lenin Action Plan for the mitigation of Elephant - Human Conflict in India .pdf, n.d.

An early warning system for elephant intrusion along the forest border areas by S. J. Sugumar and R. Jayaparvathy, n.d. Santiapillai, C., Wijeyamohan, S., Bandara, G., Athurupana, R., Dissanayake, N., Read, B., 2010. An assessment of the human-elephant conflict in Sri Lanka. Ceylon J. Sci. Biol. Sci. 39. doi:10.4038/cjsbs.v39i1.2350

Sugumar, S.J., Jayaparvathy, R., 2014. An Improved Real Time Image Detection System for Elephant Intrusion along the Forest Border Areas. Sci. World J. 2014, e393958. doi:10.1155/2014/393958

Wijesinghe, L., Siriwardena, P., Dahanayake, S., Kasthuriratne, D., Corea, R., Dias, D., 2011. Electric Fence Intrusion Alert System (eleAlert), in: 2011 IEEE Global Humanitarian Technology Conference (GHTC). Presented at the 2011 IEEE Global Humanitarian Technology Conference (GHTC), pp. 46–50. doi:10.1109/GHTC.2011.16