



Child Safety System Based Mobile App

KONATHALA HANUMAN KARTHEEK KUMAR

Under the Esteemed Guidance of
K. RATNAREDDY

ASSISTANT PROFESSOR
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
SRINIVASA INSTITUTE OF MANAGEMENT STUDIES
(Affiliated to ANDHRA UNIVERSITY)
VISAKAPATNAM
2021-2023

ABSTRACT

These Days parents are worried about their children's so they want a complete track of them and monitor them all the time, This is physically not possible so we introduce **Safety Monitoring system** which is helpful for monitoring or last geo location of the child and their activities from anywhere in the world. The major issue of child missing can be solved with the help of child tracking system as well as parents who need to keep a track of their every steps, this system plays a vital role. The android application uses google geo and sms, email services to locate their child's location. This application help to send sos messeage for the parents the major advantage of this feature.

This application sends all the data from the child's phone to the server and from the server to the parent's phone when sos button trigger manually. Parent where they can get geo location when sos trigger from children and other is the Child Part.

CHAPTER 1 –INTRODUCTION

Now a day 80% of people in the world having smart phones the smart phones uses the people by different purposes. The major issue of child missing can be solved with the help of child tracking system android application. The android application uses SMS services and google geo help in locating the missing child's location by the survey of missing children in 2004. There are of total 5996 Childs are missing. Out of these only 4092 children return or found by police. However 1904 children are missed.

And the children ages 14 years and 17 years are missed or ran away from home. The parents are worried about their children. By missing the children the parents are scared to go to the family trip. These are lots of chances to miss the child in trip. The project is developed for those parents that they have worried to miss their child. In Today's world lots of Childs have smart phones. With the help of smart phones geo and SMS based tracking application parents can watch on their child. geo is combined to GSM based SIM card into mobile to watch on child's location.

The google geo uses longitude and latitude to track the location the SMS(Short Message Service) is used to communicate child side and parent side application. SMS service used when smart phones does not support internet connectivity. System able to send the child's smart phones exact location in the parents smart phone when parent demand to check the child's location.

1.1 Objectives

The Childs are missing have ages between 14 to 17. Parents are worried about Childs. The paper explains about an android application which is used to track the missing children. The android application based on google geo and SMS services in Android mobile. The geo service is used for tracking exact location of Child.

1.2 System Specifications

Software Requirements: -

Front End: HTML5, CSS3, Bootstrap
Back End: PHP7.4, MYSQL
Control End: Angular Java Script

Tools:

xampp-win64-8.1

Hardware Requirements:

Processor : Intel 3

Installed memory (RAM) : 4 GB

Hard Disk : 500 GB

Operating System : Windows 7,8,10 - 64 bit

CHAPTER 2 – LITERATURE REVIEW

Locating Friends and Family Using Mobile Phones with google geo. Client server based approach used in the architecture. The registration of client phone done by server and after that login saved in database of server. Then client sends location coordinate updates to server the updates saved in database of server. Then with the help of Location Updates the location is tracked. This application was developed for helping to locate the family members and the friends. In 2011 the Chandra et al. used an approach with the help of SMS services. Application was implemented for JAVA mobile devices which supports google geo. The client shares his location through SMS to the web server. The Client views his location on the map.. This system architecture is Client-server based application and mobile application. In server side it uses longitude and latitude and SMS for storing user details. At Client side have a box which contains google geo tracking device and GSM modem. When user is registered and logged in web server then user details are saved to server. This application developed for monitoring driving behavior of their employees.

APPLICATION DEVELOPMENT**Requirements**

This paper design work for parents and children. The parents and children both have geo Based smart phones. The application is used to track the Childs location for implementation of application, Android SDK tools and Eclipse which support android is used. Reason for choosing android OS is that to target more users. Lots of people using android mobile phones.

Application Architecture

Solution for missing Childs with the help of google geo and GSM technologies. The application uses two main services that is geo location and SMS. For location services is longitude and latitude and telephony services is SMS. Generally selected operating system is android to over all the features. SMS is used for communicating between child

Side and parent side. The System can be designed in a simple way. The application developed to make user-friendly approach on both sides. The architecture consists of two sides. First is parent side and another is child side. Parent side acts as server and child side acts as a client. Basically

there are two android phones one is at child side and another is at parent side. Parent's side used SMS service for communicating to child's mobile and with the help of map parent used to view child's location on map.

That's why it uses internet and telephony services enable at parent side to track child's location. At child's side another android phone supports google geo and SMS facilities. Child side uses telephony services to communicate with parent side. In child side location services that is enabled and running in child side .And parent side uses internet connectivity to view child's location on the map. At parent side requirements are mapped for tracking and service(listener)for listening messages coming from child's side. On the child's side listener service always runs in background, at parent side used to send SMS for location of child. Parent side listener used to listen Childs reply for location request.

There are two main functions at child side. To listen and gets location from satellites or network provider. Listener is a very important service at Childs side to listen all SMS incoming and reply SMS which only related to location requests. When parent sends location requests to the Childs side, at child side it listens message and code for this message to reply the location requests on map at parent side. The application is programmed to listen the predefined strings. At the listener the instructions or commands are saved such as "\$getUpdate\$" will be used by sending parent side location requests. Whereas "\$update\$Coord" will be used at child side for location update.

Detection of Safety Distance

When parent starts the system, she/he should select where they stay; indoor or outdoor; in order to calculate the distance between them and each child. Indoor option limit child range to 3m by reducing child transmitter source so signal attenuated will not be detected. But if parent selects outdoor option, distance will be calculated by taking child and parent locations from google geo [5]. Child module will send to parent module which calculates distance and compares it with distance threshold; this threshold value should be entered by parent after selecting outdoor option.

Indoor Circuit Calculation In order to determine indoor distance between each child and their parents, three methods were tested: Time of Flight, Using logic gates with RF and change amplitude of RF signal.

Time of Flight In this method, the distance will be determined by calculating the time that RF signal will take in flight. The hardware implementation of this method is shown in figure 4 with the following steps that must be applied:

- Reset the counter.
- Send request from parent for a predefined character.
- Start the counter.
- Wait until receive the character from child.

- Stop the counter and take its reading.

2.1 Existing Solution:

In the existing system, we use a voice recognition module in which the alert commands from the child are stored and kept for further reference. If the same child delivers the same command, it will compare with the alert command which was previously stored and sets an emergency level according to the alert command. The GSM has a SIM which is used to send an alert message or an alert call to the trusted peoples.

The disadvantage of this project are,

- i. The child could not produce the exact alert command during a panic condition.
- ii. The command produced may not match with the previously stored command.
- iii. This project requires manual intervention

2.2 Proposed Solution:

This system is designed for parents and children. It can also be used in offices by the employer for the employees. The application is mostly to be used by parents to track their child's location. In a later phase for implementation purposes, the system will be eclipse supporting android. The main reason why the Android Operating System was chosen for the implementation of this work is to target more users. Statistics shows that the market share for the Android OS 48.8. This makes it the highest market shares over other smartphones operating systems currently in the market

CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

3.1 Module Description

The basic operation of the proposed child tracking system is that when a violation of child safe is detected, a specific sensor in child module will produce a signal. The parent module will take the decision and start the violation handling procedure. The operation of the child tracking system requires certain hardware between child model and parent one.

This includes a certain driving circuit that activates the sensors. The main hardware parts of child tracking system

3.2 System Features

In the life of the software development, problem analysis provides a base for design and development phase. The problem is analyzed so that sufficient matter is provided to design a new system. Large problems are sub-divided into smaller once to make them understandable and easy for finding solutions. Same in this project all the task are sub-divided and categorized.

System Modules:

PARENT

- Registration and login
- Create site details
- View Google geo location
- Manage sites
- Block Site
- Receive sms

ADMIN LOGIN (Act like police control room)

- Login
- Mointor Emerencies SOS
- Receive only email

CHILD

- Registration and Login
- Search site
- Update Geo location
- SOS Button - For emerency
- SMS Alert to parent
- Add Guardin - Email & Mobile No
- Manage Guardian

3.3 MODULES:

Parent:

Create site details:

The main activities in the application are the user create site for user. The other modules are followed by this create site. This module records only user and password of the user.

View Google geo location:

The android application uses google geo location map services to locate their child's location

Manage sites:

Parent are manage the sites details.

Block Site:

The parent complaint details to the admin directly the admin receive the sms and email and then block to the wrong person id.

Receive sms & email:

Receive sms and email from child in emergency time.

Admin Login:**Mointor Emerencies SOS:**

we introduce Safety Monitoring system which is helpful for monitoring or tracking the child and their activities from anywhere in the world.

Receive only email

User views the status about their complaint for receive only email

Child:**Login**

The main activities in the application are the user login page for user. The other modules are followed by this login page. This module records only user and password of the user.

Search site

The search to website in the child safety application and then login page for user. To create the email id and user password.

Update Geo location

The user can request for the location of a family member at any time based on longitude and latitude location.

SOS Button - For emergency

Its primary role is to periodically receive messages and in response send email to the parent module and alert them if the child is in danger.

SMS Alert to parent

This application help to send sos messeage for the parents the major advantage of this feature.

Add Guardian - Email & Mobile No

Feature in the application add guardian details for calling secondary emergency purpose.

Manage Guardian

Guardian app which is installed in the guardian mobile and initiated by giving the mobile number.



CHAPTER 4 – DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.1 UML Diagrams:

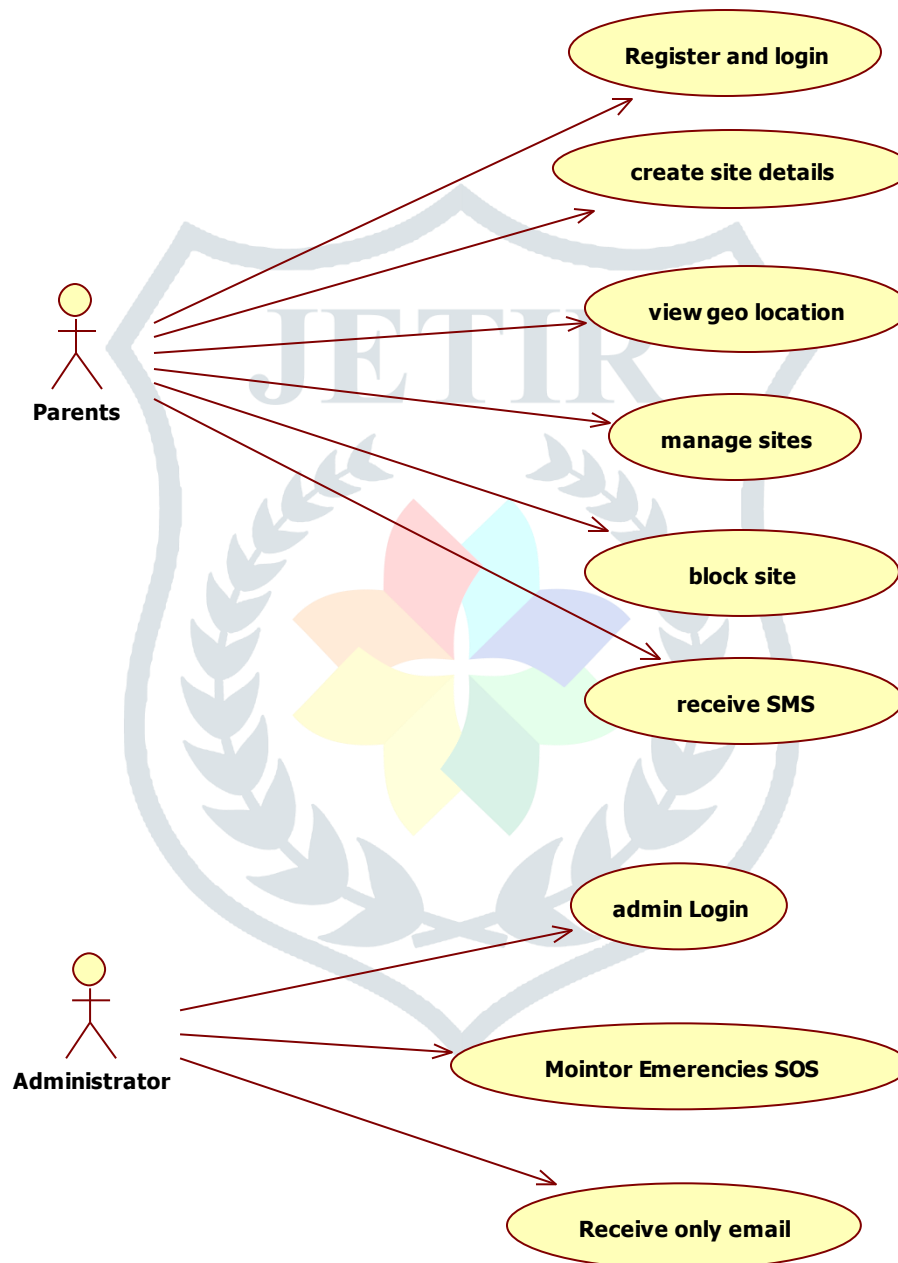
UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

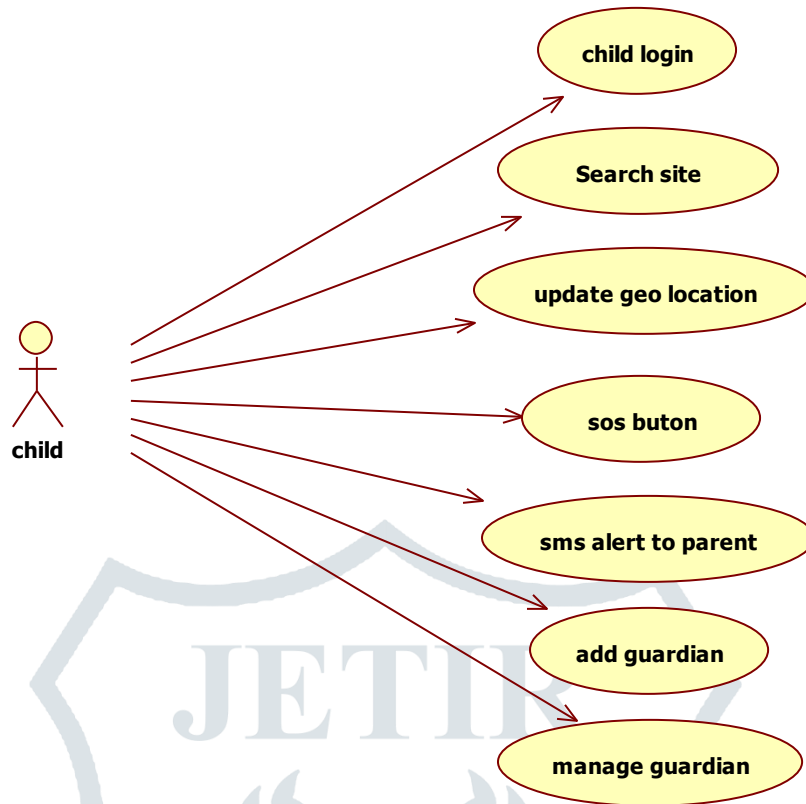
There are various kinds of methods in software design:

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- Class Diagram
- Dataflow Diagram

4.1.1 Usecase Diagrams:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do.





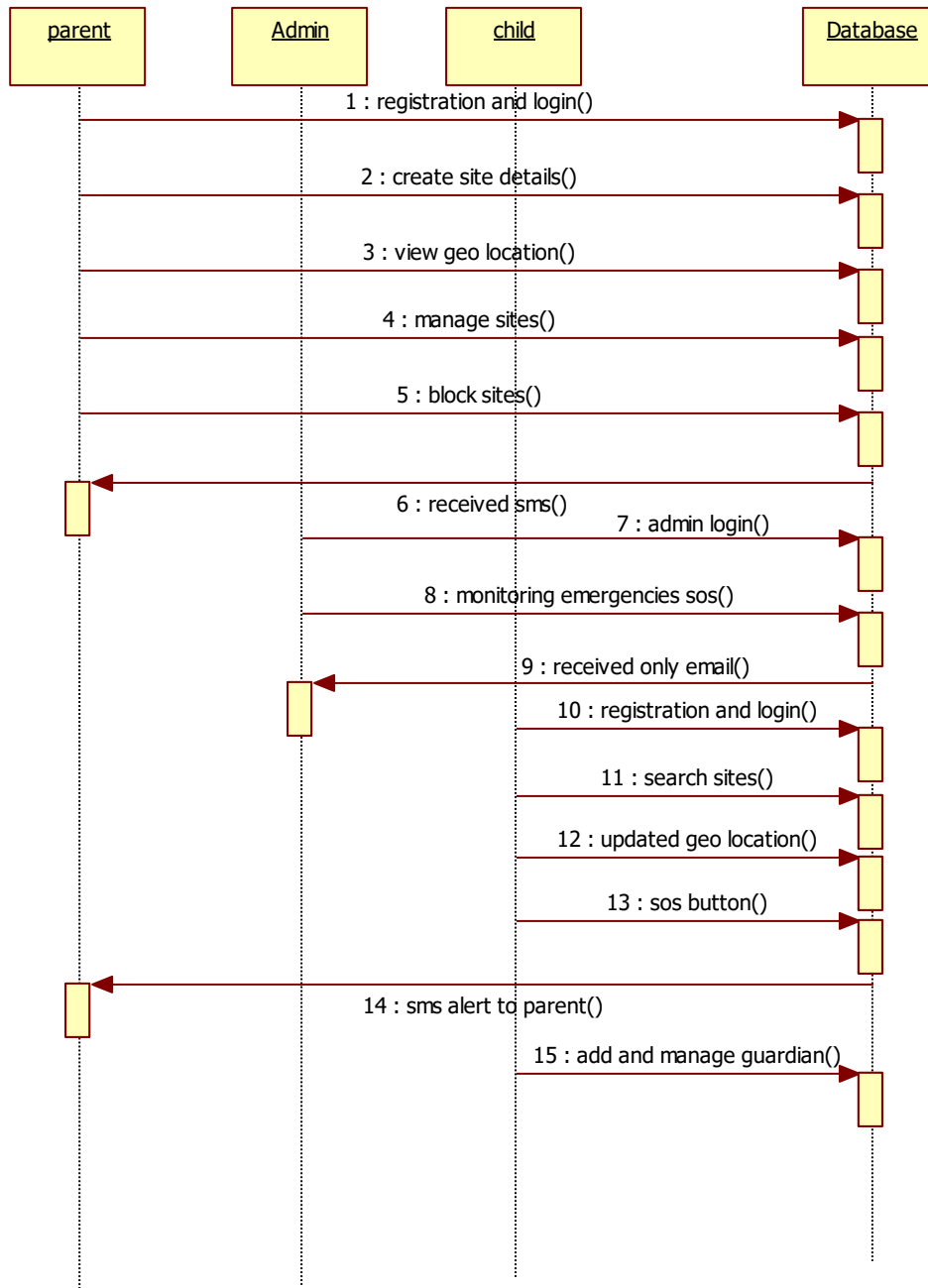
Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

4.1.3 Sequence Diagram:

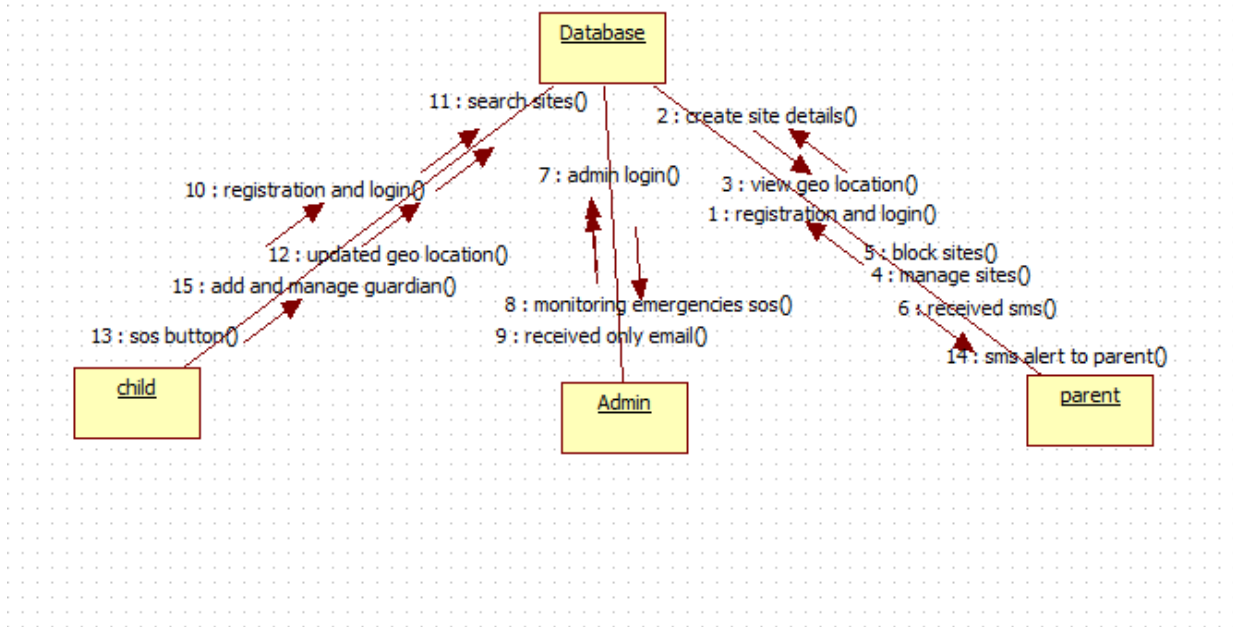
Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.

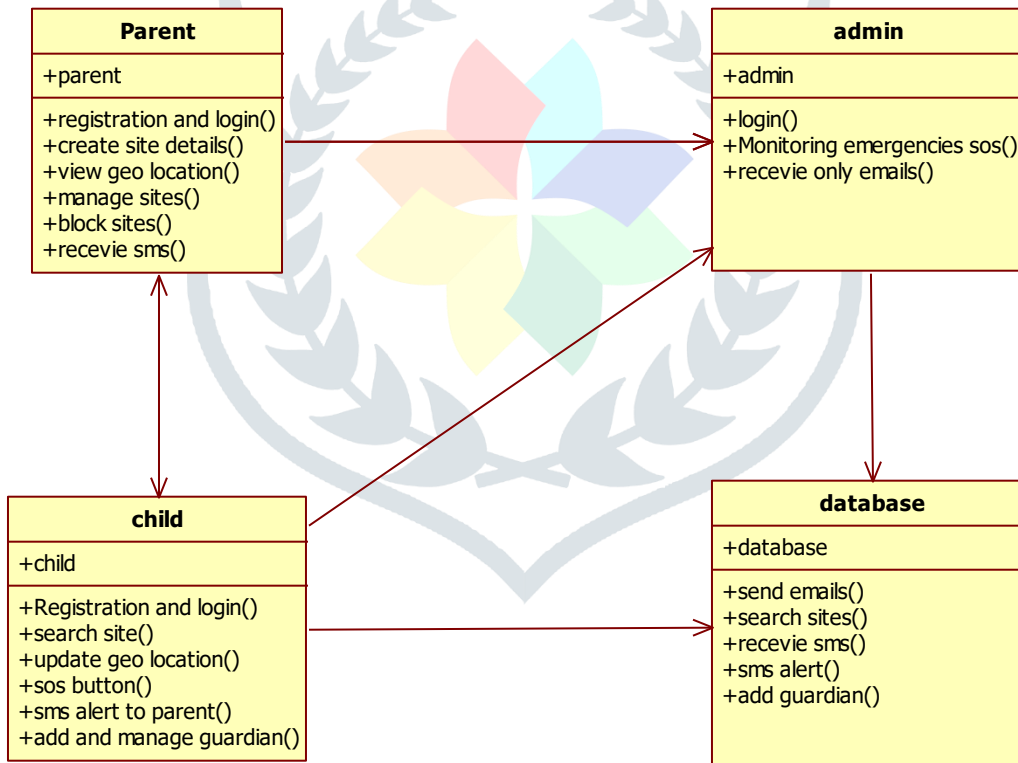


4.1.3 Collaboration Diagram:

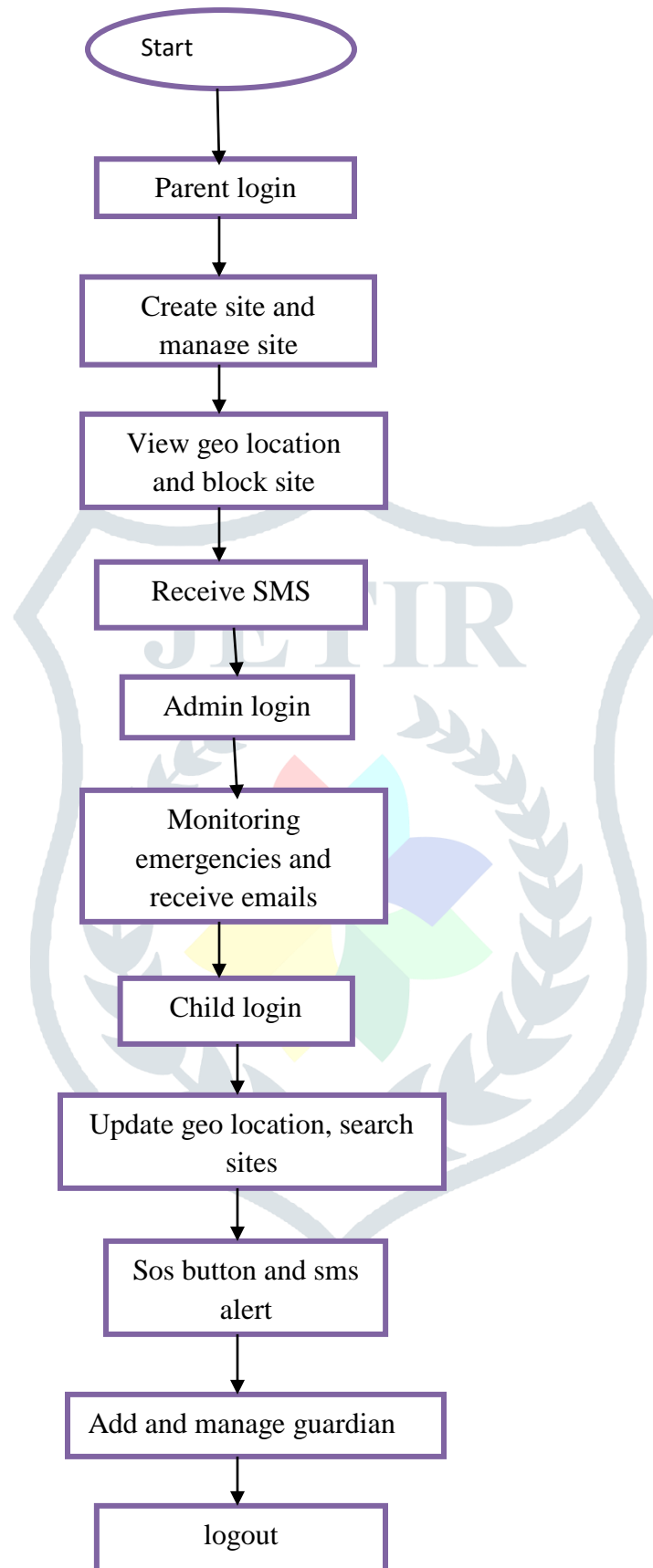
A **collaboration diagram** is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this **collaboration**. A **collaboration diagram** often comes in the form of a visual chart that resembles a flow chart.



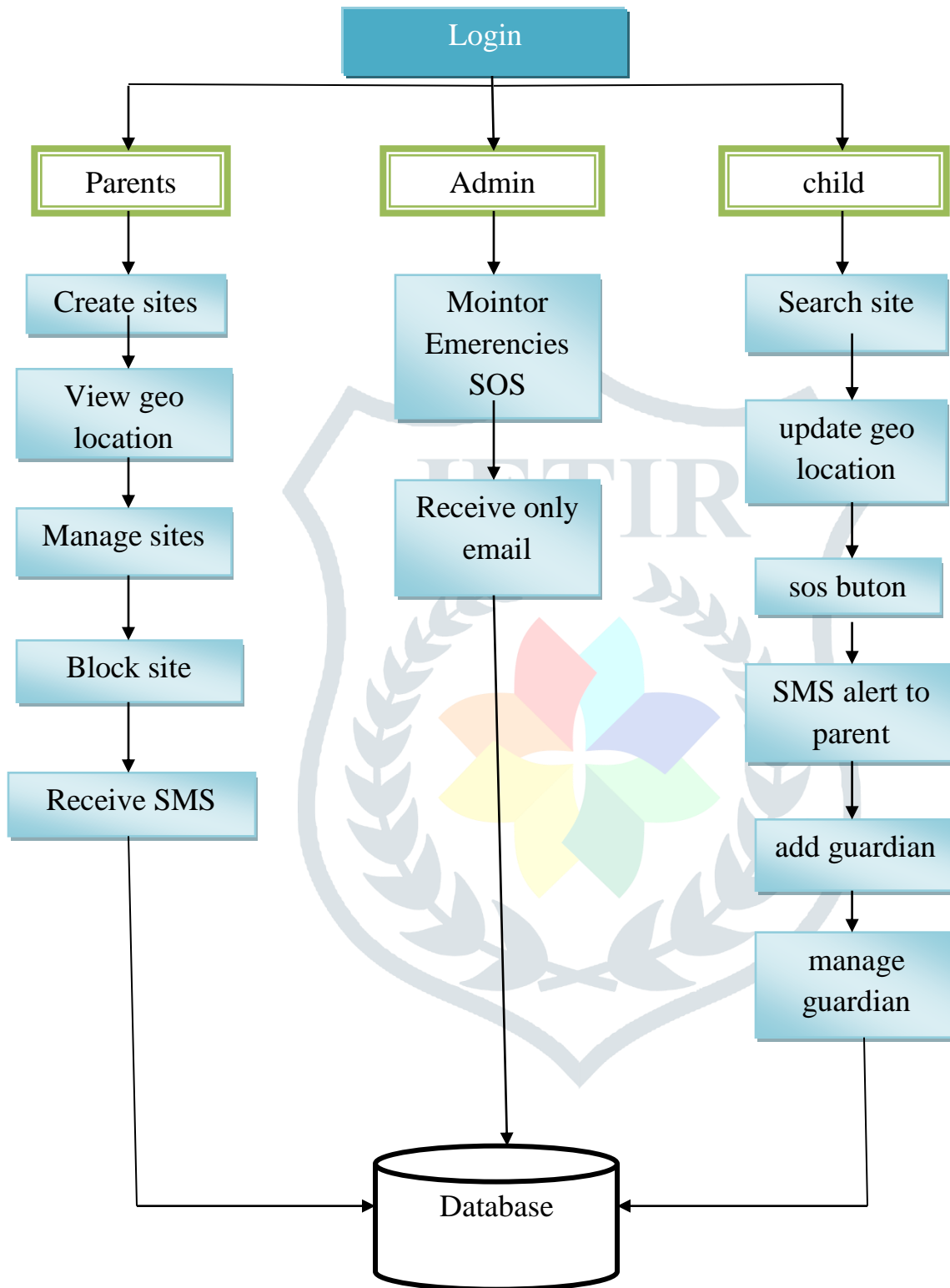
4.1.4 Class Diagram:



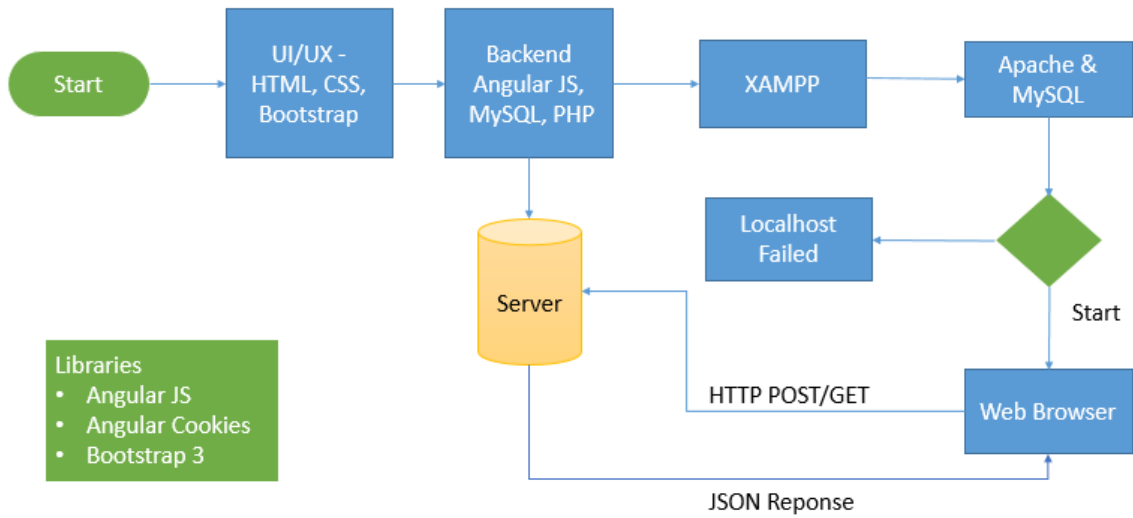
4.1.5. Data Flow Diagram:



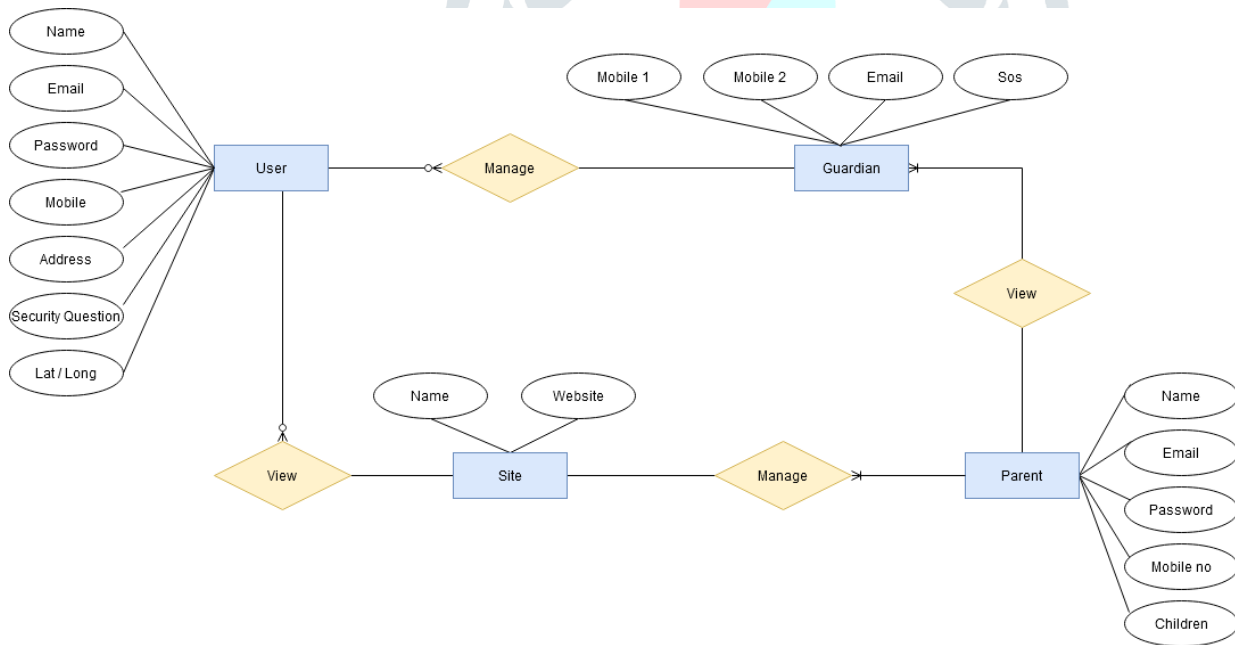
System Design:



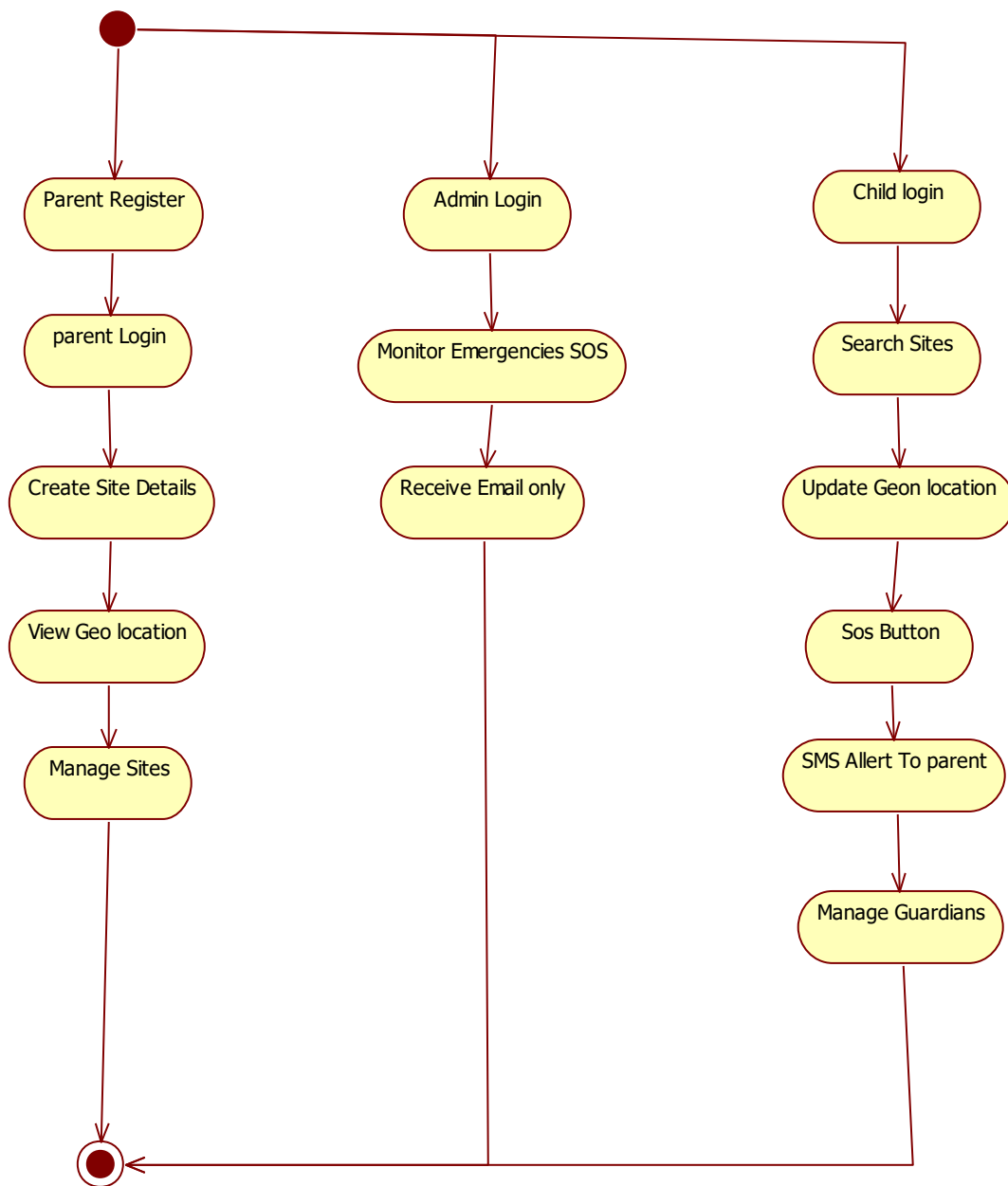
4.1.6. WORKFlow Diagram



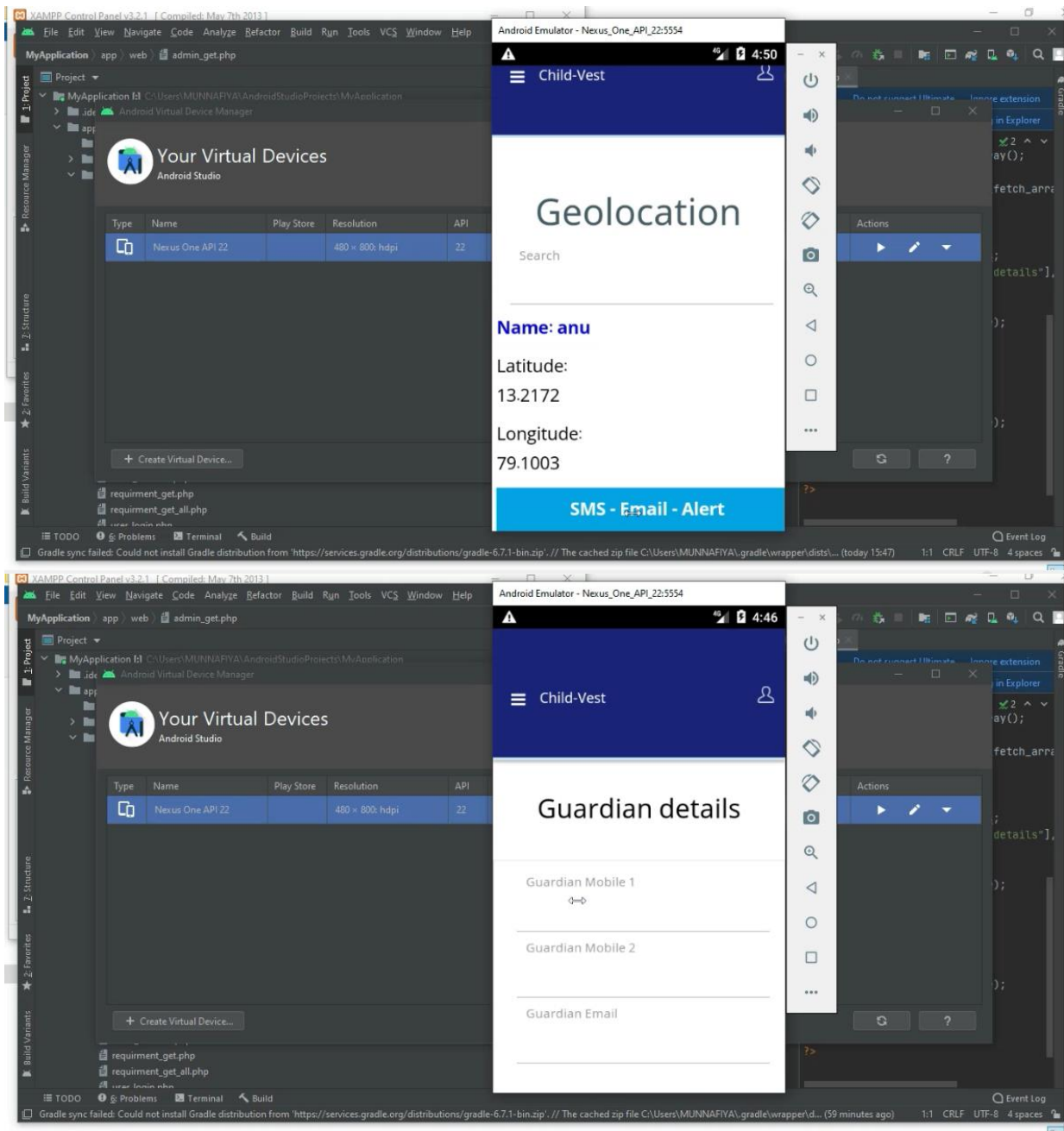
4.1.7. ER Diagram

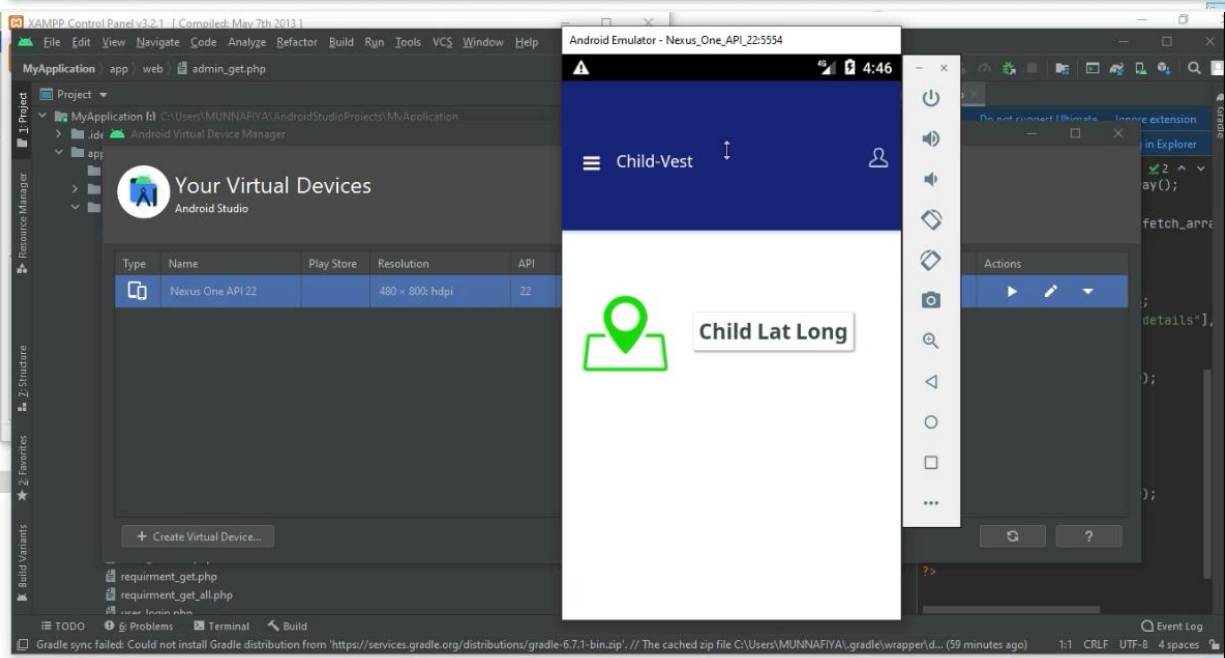
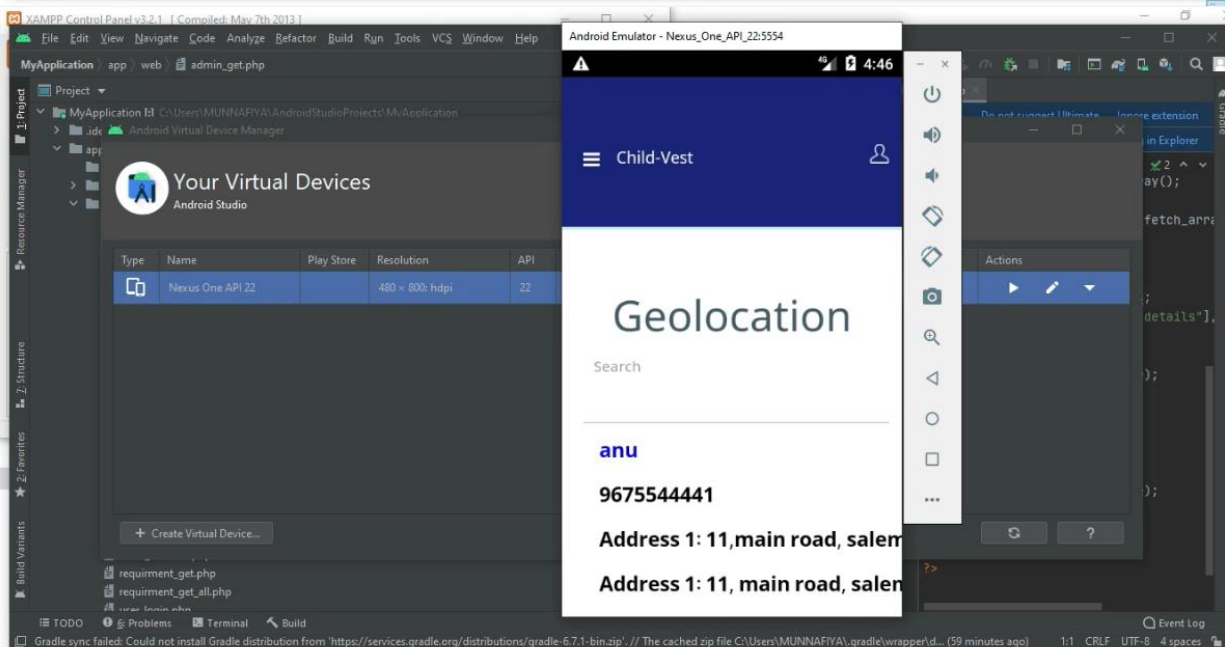
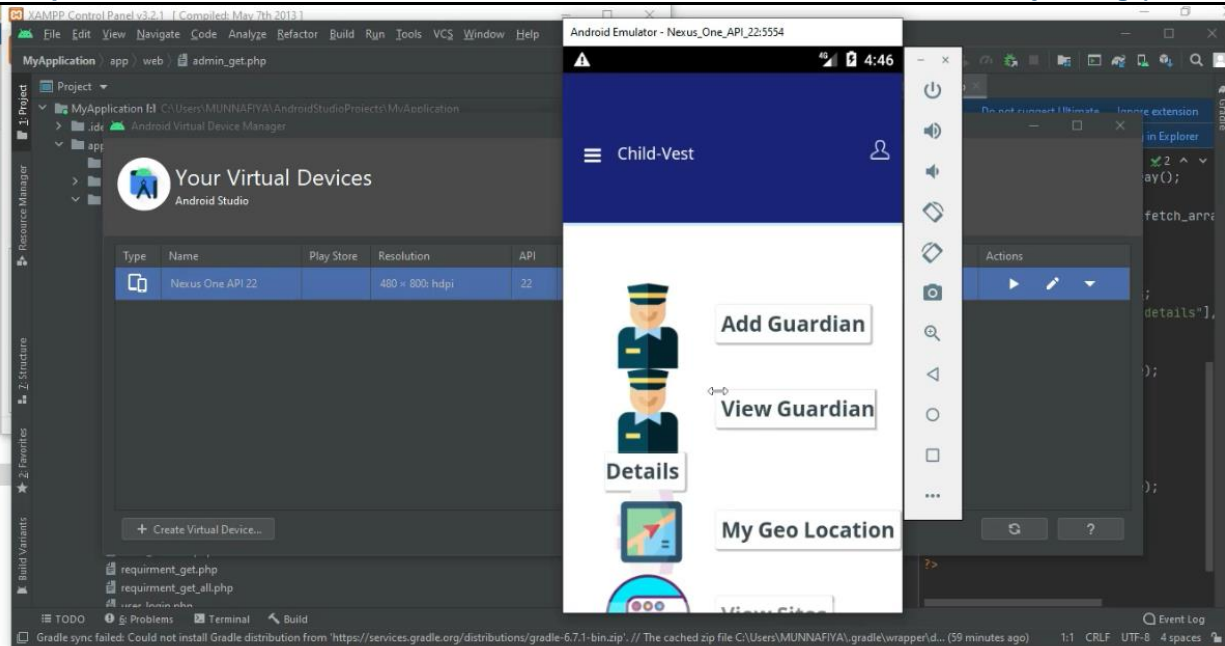


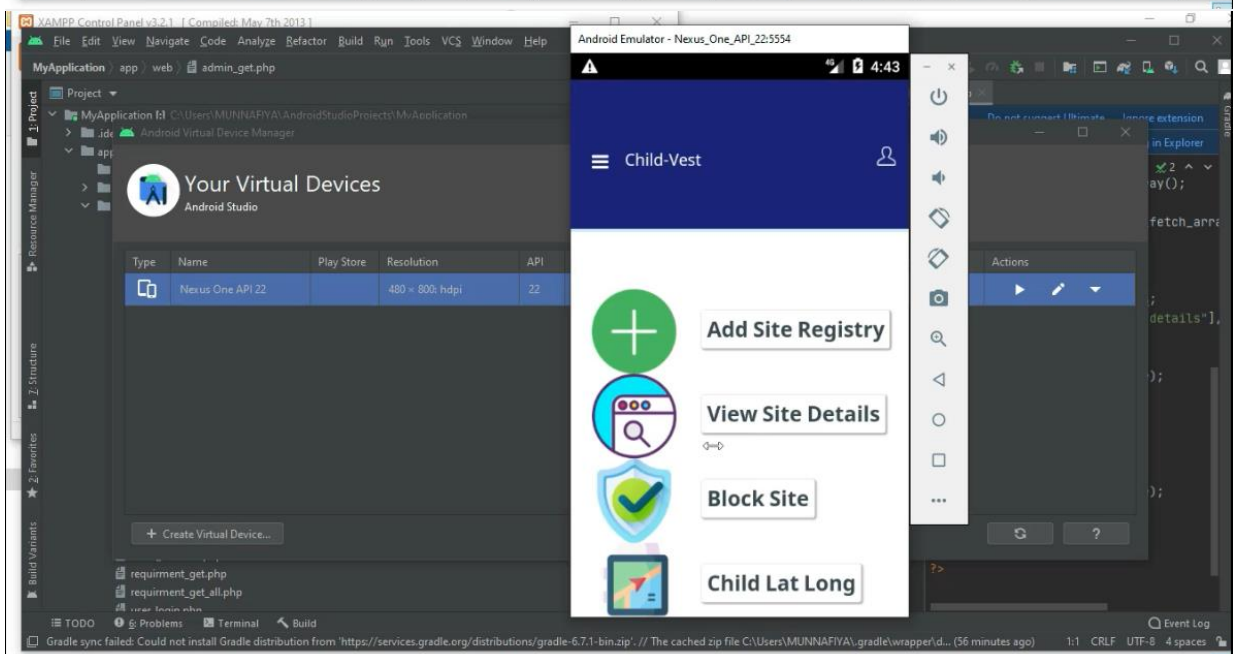
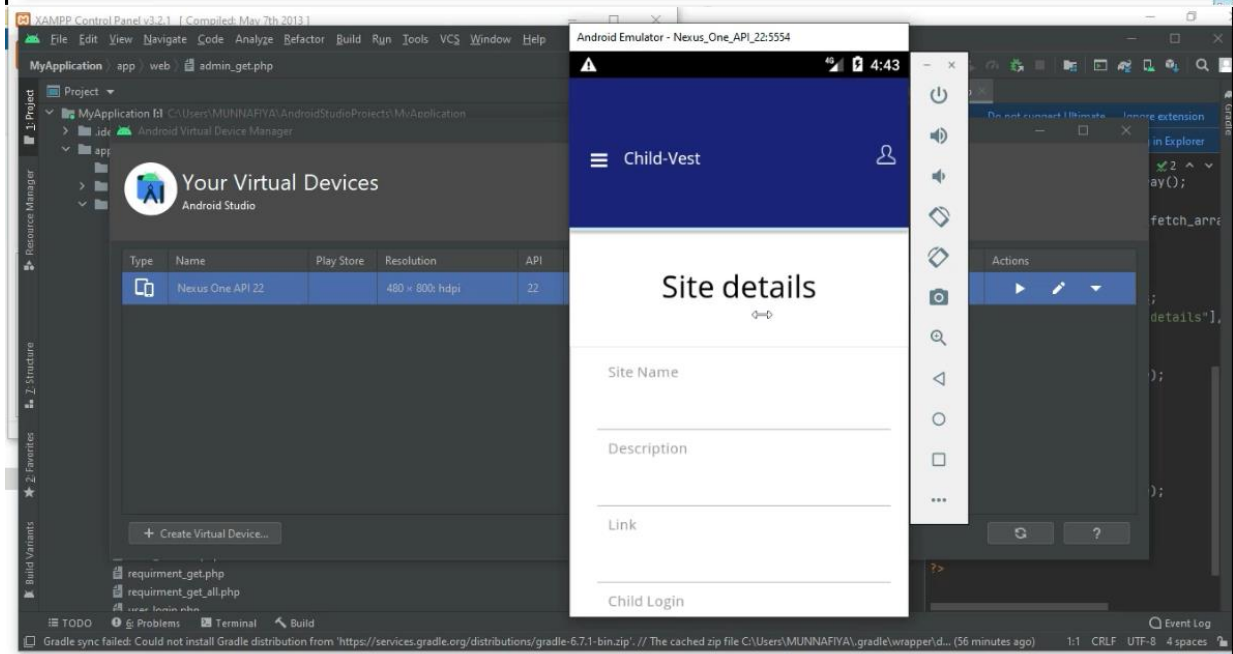
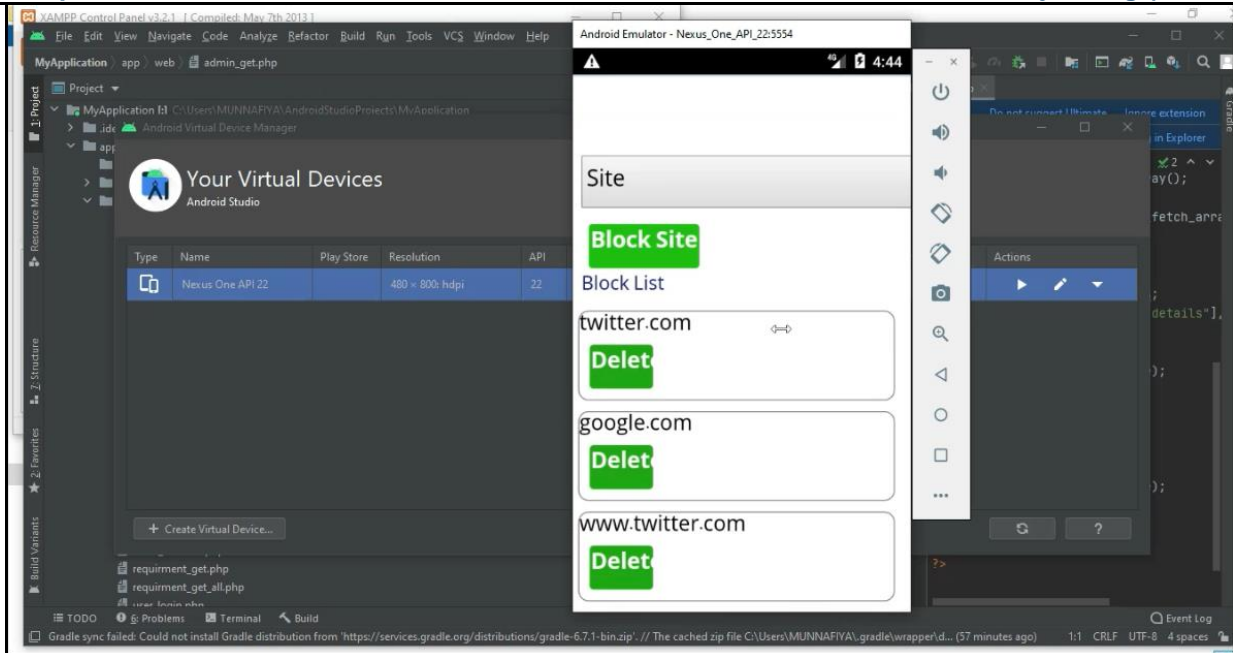
4.1.8 . Activity Diagram

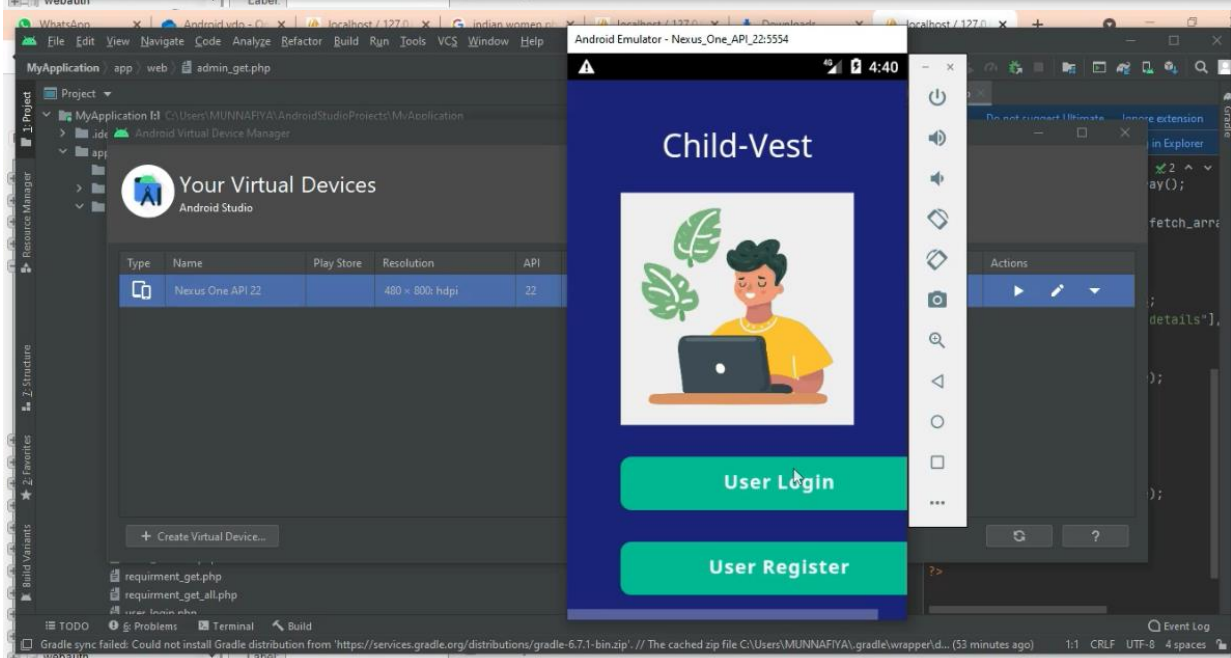
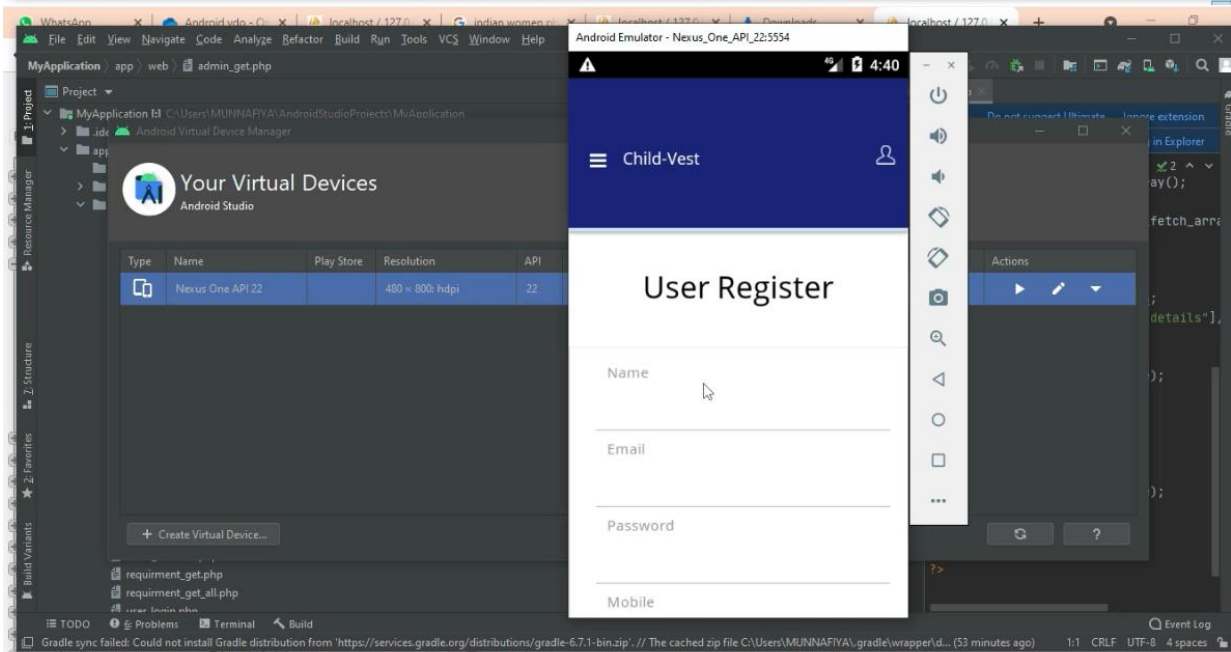
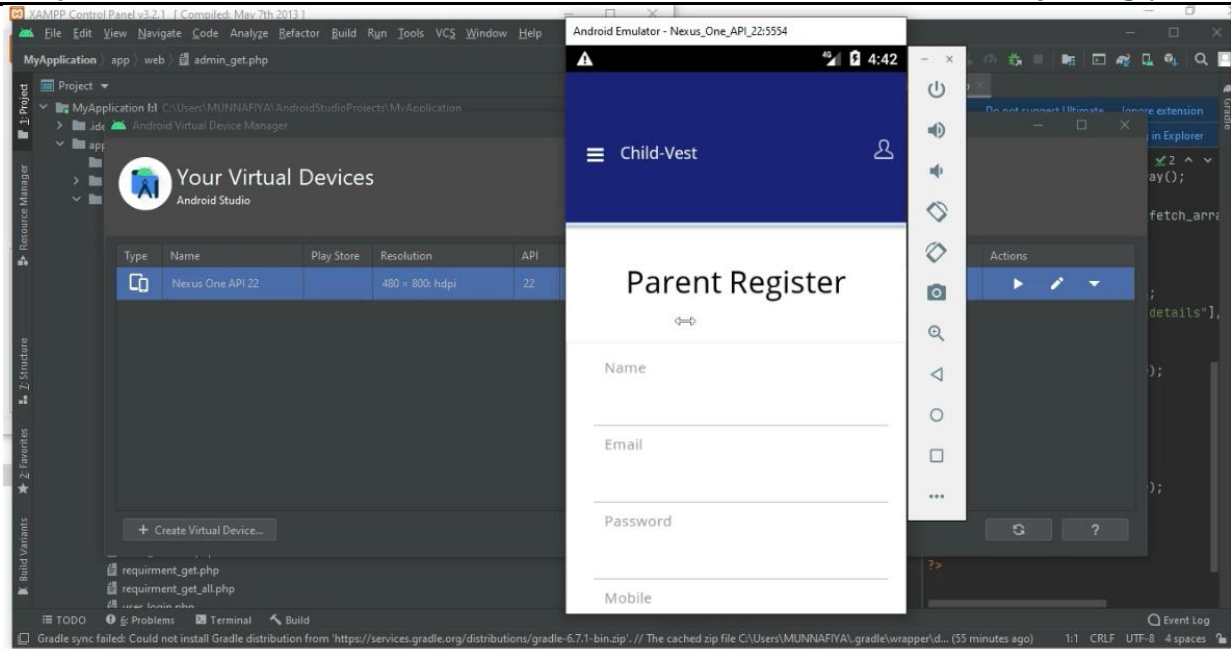


CHAPTER 5 - OUTPUT SCREENSHOTS









CHAPTER 6 – IMPLEMENTATION DETAILS

6.1 Introduction to Html Framework

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of HTML pages. HTML allows images and other objects to be embedded and it can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. HTML markup can also refer the browser to Cascading Style Sheets (CSS) to define the look and layout of text and other material

6.2 Cascading Style Sheets (CSS)

CSS is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve

content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold", leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a <bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

6.3 MYSQL Server

MySQL is an open-source relational database management system (RDBMS); [6] in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client-server model RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as

under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

6.4 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

6.5 ANGULAR JAVA SCRIPT

AngularJS (commonly referred to as "Angular" or "Angular.js") is an open-source web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for

client-side model–view–controller (MVC) and model–view–viewmodel(MVVM) architectures, along with components commonly used in rich Internet applications.

The AngularJS framework works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

According to JavaScript analytics service Libscore, AngularJS is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and approximately 8,400 other sites out of 1 million tested in July 2015.

AngularJS is the frontend part of the MEAN stack, consisting of MongoDB database, Express.js web application server framework, Angular.js itself, and Node.js runtime environment

CHAPTER 7- SYSTEM STUDY

7.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

CHAPTER 8-TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

8.1Non Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well software does what it has to do. These are Quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

8.2 Accuracy:

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy then the system will provide alternative ways to solve the problem.

8.3 Usability:

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

8.4 Accessibility:

The system will be accessible through internet and there should be no any known problem.

8.5 Performance:

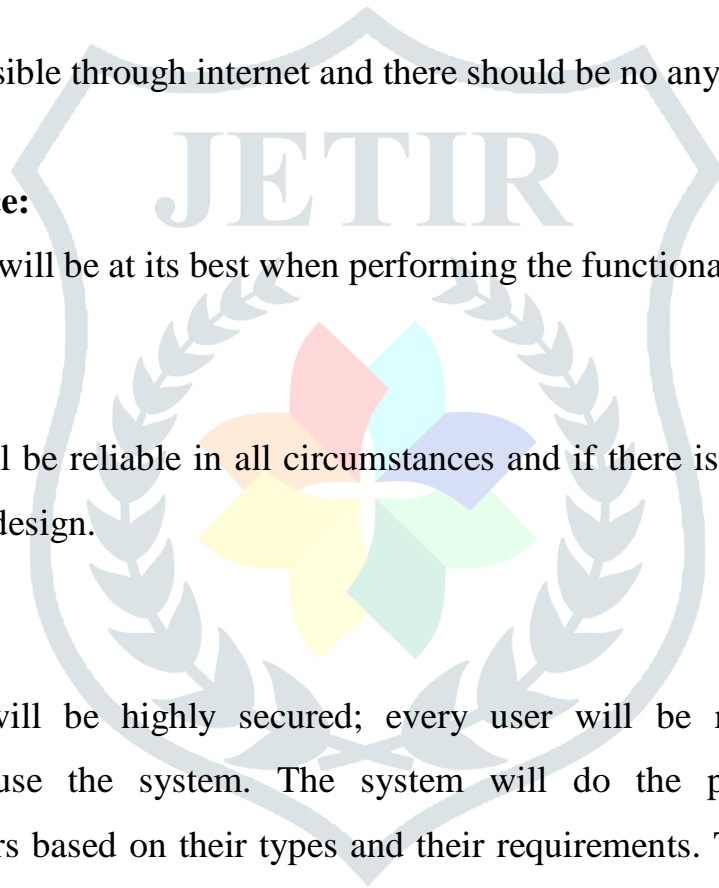
The system performance will be at its best when performing the functionality of the system.

8.6 Reliability:

The proposed system will be reliable in all circumstances and if there is any problem that will be affectively handle in the design.

8.7 Security:

The proposed system will be highly secured; every user will be required registration and username/password to use the system. The system will do the proper authorization and authentication of the users based on their types and their requirements. The proposed system will be designed persistently to avoid any misuse of the application.



CHAPTER 9-SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Testcase

S.NO	SCENARIO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Parent login Details	Parent enter login details	Login successfully or if incorrect login details “Login successfully”	Login successfully or Login unsuccessfully
2	Create Site details	Site created by all details like user name ,password	All the details Create successfully	Created successfully or created unsuccessfully
3	View Google GEO location	Parent check and view all GEO location	User view GEO location	View Google GEO location
4	Manage Site	Parent add the Site details	all the Site details “updated successfully”	updated successfully or unsuccessfully
5	Block Site	Parent check and view all Block Site	Parent view Site details	View division Site details
6	Receive sms and email	Parent add the sms and email	all the receive email and sms “updated successfully”	updated successfully or unsuccessfully

7	Admin login Details	Admin will enter email and password	Login successfully or if incorrect login details “Login unsuccessfully”	Login successfully or Login unsuccessfully
8	View Monitor Emerencies SOS	Admin will check the Monitor Emerencies SOS	Admin will update Monitor Emerencies SOS	Updated successfully or unsuccessfully
9	Receive only email	Admin add the email	all the receive email “updated successfully”	updated successfully or unsuccessfully
10	Child Login	Email and Password	If correct directed to home page otherwise show “Login Successfully”	Login successfully or Login unsuccessfully
11	Search site	Child can check and view all mechanic details	view site details	View all details
12	Update Geo Location	Mechanic update location of shop in google map	Drap and drop the locaiton	Updated successfully or unsuccessfully
13	SOS Button	Click SoS Button	Guardian will recieve sms and email	SMS Alert Sent successfully or unsuccessfully
14	View SMS Alert to parent	Child check and view all SMS alert to parent	Child view SMS Alert to parent	View division SMS alert to parent
15	Add Guardian Details	Child add the Guardian details like email and phone no	all the Guardian details “created successfully”	created successfully or unsuccessfully
16	Manage Guardian	Child add the Guardian	all the guardian details “updated successfully”	updated successfully or unsuccessfully

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

9.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

9.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 10– CONCLUSIONS

The word Future resembles the word Children. The future pillars of one's nation", today's children are tomorrow's youngsters, preserving their dreams and life for a better future is necessary. Therefore, each and every parent should take care of their own children, without letting them to fall into the dark world of abuse, which entirely ruin them physically, mentally and emotionally destroying our future. Hence, considering the importance of our future, our project makes it easy for parents to track their children and to visually monitor them on regular basis, which makes them ensure the safety of their children and reduces the rate of incidents of child abuse.

CHAPTER 11- REFERENCES

[1] Akash Moodbidri, Hamid Shahnasser, "Child Safety Wearable Device", Department of Electrical and Computer Engineering San Francisco State University.

[2] Anand Jatti, Madhvi Kannan, Alisha RM, Vijayalakshmi P, Shrestha Sinha, "Design and Development of an IOT based wearable device for the Safety and Security of women and girl children", IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India

[3] "RFID-based System for School Children Transportation Safety Enhancement", Proceedings of the 8th IEEE GCC Conference and Exhibition, Muscat, Oman, 1-4 February 2015.

[4] Dr. R. Kamalraj, "A Hybrid Model on Child Security and Activities Monitoring System using IoT", IEEE Xplore Compliant Part Number: CFP18N67-ART; ISBN: 978-1-5386-2456-2.

[5] Pooja. K. Biradar¹, Prof S. B. Jamge², "An Innovative Monitoring Application for Child Safety", DOI: 10.15680/IJIRSET.2015.0409093.

CHAPTER 12- CODING

```
<!DOCTYPE html>
```

```
<html ng-app="myapp">
```

```
  <head>
```

```
    <!-- Basic -->
```



```
<meta charset="utf-8">

<meta name="keywords" content="HTML5 Template" />

<meta name="description" >

<meta name="author" content="pixelgeeklab.com">

<!-- Mobile Metas -->

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Web Fonts -->

<link href='css/css.css' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="fonts/font.css">

<!-- Vendor CSS -->

<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.css">

<link rel="stylesheet" href="vendor/owlcarousel/owl.carousel.css" media="screen">

<link rel="stylesheet" href="vendor/owlcarousel/owl.theme.css" media="screen">

<link href="vendor/owl-carousel/owl.transitions.html" rel="stylesheet" media="screen">

<link rel="stylesheet" href="vendor/flexslider/flexslider.css" media="screen">

<link rel="stylesheet" href="vendor/chosen/chosen.css" media="screen">

<link rel="stylesheet" href="vendor/magnific-popup/magnific-popup.css" media="screen">

<!-- Theme CSS -->

<link rel="stylesheet" href="css/theme.css">

<link rel="stylesheet" href="css/theme-animate.css">

<!-- Style Switcher-->

<!-- Head libs -->

<script src="vendor/modernizr/modernizr.js"></script>
```



```
<!--[if IE]>
```

```
<link rel="stylesheet" href="css/ie.css">
```

```
<![endif]-->
```

```
<!--[if lte IE 8]>
```

```
<script src="vendor/respond/respond.js"></script>
```

```
<script src="vendor/excanvas/excanvas.js"></script>
```

```
<![endif]-->
```

```
<style>
```

```
.container {
```

```
background-color: #0b2239;
```

```
position: relative;
```

```
margin-top: 100px;
```

```
}
```

```
.row {
```

```
margin-left: 50px;
```

```
margin-right: 50px;
```

```
text-align: center;
```

```
}
```

```
.login p {
```

```
position: relative;
```

```
text-align: center;
```

```
font-size: 30px;
```

```
color: #ffffff;
```

```
}
```

```
p {
```

```
position: relative;
```



```
text-align:center;
```

```
font-family:'Quicksand';
```

```
font-size:35px;
```

```
color:#ffffff;
```

```
font-weight:200;
```

```
}
```

```
h2{
```

```
position:relative;
```

```
text-align:center;
```

```
font-size:30px;
```

```
color:#ffffff;
```

```
font-weight:200;
```

```
}
```

```
h3{
```

```
text-align:center;
```

```
font-size:80px;
```

```
color:#ffffff;
```

```
font-weight:200;
```

```
}
```

```
img{
```

```
align:middle;
```

```
}
```

```
</style>
```

```
<!-- script back button -->
```

```
<script src="cordova.js"></script>
```

```
<script>
```

```
function onLoad()
```

```
{
```

```
document.addEventListener("deviceready", deviceReady, false);

}

function deviceReady()

{

    document.addEventListener("backbutton", backButtonCallback, false);

}

function backButtonCallback()

{

    navigator.app.exitApp();

}

</script>

<!-- angular js -->
<script src="js/angular-1.3.js"></script>
<script src="js/angular_cookies.js"></script>
</head>

<body >

<div class="container">

<h2>Criminal Identification App</h2>

<div class="row">

<div class="col-md-4" >

</div>

<div class="col-md-4" >


```



```
<a href="user_login.html"> <button style="width:50%;background-color:#4fd8c6;font-weight:bold;color:#000;letter-spacing:1px">User Login</button> </a> <br><br>
```

```
<a href="user_register.html"><button href="user_register.html" style="width:50%;background-color:#4fd8c6;font-weight:bold;color:#000;letter-spacing:1px">User Register</button> </a><br><br>
```

```
<a href="admin_login.html"><button href="admin_login.html" style="width:50%;background-color:#4fd8c6;font-weight:bold;color:#000;letter-spacing:1px">Admin</button> </a><br>
```

```
</div>
```

```
<div class="col-md-4" >
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script src="vendor/jquery/jquery.js"></script>
```

```
<script src="vendor/bootstrap/bootstrap.js"></script>
```

```
<script src="vendor/jquery.validation/jquery.validation.js"></script>
```

```
<script src="vendor/owlcarousel/owl.carousel.js"></script>
```

```
<script src="vendor/flexslider/jquery.flexslider-min.js"></script>
```

```
<script src="vendor/countdown/countdown.min.js"></script>
```

```
<script src="vendor/chosen/chosen.jquery.min.js"></script>
```

```
<script src="vendor/pricefilter/jquery.pricefilter.js"></script>
```

```
<script src="vendor/masonry/imagesloaded.pkgd.min.js"></script>
```

```
<script src="vendor/masonry/masonry.pkgd.min.js"></script>
```

```
<script src="vendor/uikit/uikit.js"></script>
```

```
<script src="vendor/magnific-popup/jquery.magnific-popup.js"></script>
```

```
<!-- Theme Base, Components and Settings -->
```

```
<script src="js/theme.js"></script>
```

```
<!-- Style Switcher -->
```

```
<script type="text/javascript" src="style-switcher/js/switcher.js"></script>
```

```
<!-- angular js -->
```

```
<script src="js/angular_product.js"></script>
```

```
<body onload="onLoad()">
```

```
</body>
```

```
</html>
```

```
Angular_user.js
```

```
var app = angular.module("myapp", ['ngCookies']);
```

```
app.controller("myappCtrl", function($scope, $cookieStore, $cookies, $http)
```

```
{
```

```
/***/
```

```
/***/ User Login *****/
```

```
/***/
```

```
// sign in button
```

```
$scope.user_login = function()
```

```
{
```

```
$http.post('user_login.php',
```

```
    {'email': $scope.email, 'password': $scope.password})
```

```
.success(function(data, status, headers, config)
```

```
{
```

```
    if(data.success == 1)
```

```
    {
```

```
        alert("Login Successful");
```

```
        $cookieStore.put("cook_user_email", data.email);
```

```
window.location = "home.html"; // Home Page
```

```
return;
```

```
}
```

```
else if(data.success == 2)
```

```
{
```

```
    alert("Please Fill All Fields");
```

```
}
```

```
else
```

```
{
```

```
    alert("Login Unsuccessful");
```

```
}
```

```
});
```

```
}
```

```
/****** Cookies *****/
```

```
    $scope.cook_user_email = $cookieStore.get("cook_user_email");
```

```
/****** User Logout *****/
```

```
/****** User Logout *****/
```

```
/****** User Logout *****/
```

```
    $scope.user_logout = function()
```

```
{
```

```
    if(confirm("Are You Sure?"))
```

```
{
```

```
        $cookies.cook_user_email = "";
```

```
        $cookies.cook_admin_email = "";
```

```
        window.location = "index.html";
```

```
return;
```



```
    }  
    else  
    {  
        return false;  
    }  
}
```

```
//***** admin_register *****//
```

```
$scope.user_register = function()  
{  
    $http.post('user_register.php',{  
        'name': $scope.name, 'email': $scope.email, 'password': $scope.password,  
        'mobile': $scope.mobile, 'field_1': $scope.field_1, 'field_2': $scope.field_2,  
        'field_3': $scope.field_3, 'field_4': $scope.field_4 })  
    .success(function(data, status, headers, config)  
    {  
        if(data.success == 1)  
        {  
            alert("Registered successfully");  
            window.location = "user_login.html";  
            return;  
        }  
        else if(data.success == 2)  
        {  
            alert("Please Fill All Fields");  
        }  
        else if(data.success == 0)  
        {  
            alert("Error");  
        }  
    }  
}
```

```
    }
    else
    {
        alert(" Un Successfull");
    }
});
}

/*****
/***** Admin Login *****/
/*****/

// sign in button
$scope.admin_login = function()
{
$http.post('admin_login.php',
    {email: $scope.email, 'password':$scope.password})
.success(function(data, status, headers, config)
{
    if(data.success == 1)
    {
        alert("Login Successful");
        $cookieStore.put("cook_admin_email",data.email);
        window.location = "admin_home.html"; // Home Page
        return;
    }
    else if(data.success == 2)
    {
        alert("Please Fill All Fields");
    }
    else
    {
```

```
alert("Login Unsuccessful");
```

```
}
```

```
});
```

```
}
```

```
/****** Cookies Login *****/
```

```
$scope.cook_admin_email = $cookieStore.get("cook_admin_email");
```

```
/****** admin_register *****/
```

```
$scope.admin_register = function()
```

```
{
```

```
$http.post('admin_register.php',{
```

```
'name':$scope.name,'email':$scope.email,
```

```
'password':$scope.password,'mobile': $scope.mobile))
```

```
.success(function(data, status, headers, config)
```

```
{
```

```
if(data.success == 1)
```

```
{
```

```
alert("Registered successfully");
```

```
window.location = "admin_login.html";
```

```
return;
```

```
}
```

```
else
```

```
{
```

```
alert("Invalid Inputs");
```

```
}
```

```
});
```

```
}
```

```
/****** Update Admin Info *****/
```

```
$http.post('get_admin_info.php')  
.success(function(data, status, headers, config)  
{  
    if(data.success == 1)  
    {  
        $scope.details = data.details;  
    }  
});
```

```
/****** Update User Info *****/
```

```
$http.post('get_user_info.php',  
{  
    'email': $scope.cook_user_email  
})  
.success(function(data, status, headers, config)  
{  
    if(data.success == 1)  
    {  
        $scope.userdetails = data.details;  
    }  
});
```

```
/******
```

```
/****** User Login *****/
```

```
/******
```

```
$scope.myinfovar = true;
```

```
/***/
```

```
/***/ Admin Update Login *****/
```

```
/***/
```

```
$scope.update_info = function(email,password,name,mobile)
```

```
{
```

```
    $scope.myinfovar = false;
```

```
    $scope.email = email;
```

```
    $scope.password = password;
```

```
    $scope.name = name;
```

```
    $scope.mobile = mobile;
```

```
    //window.location = "home.html";
```

```
}
```

```
$scope.save_info = function()
```

```
{
```

```
    $http.post('admin_update.php',{
```

```
        'name':$scope.name,'email':$scope.email,
```

```
        'password':$scope.password,'mobile': $scope.mobile})
```

```
    .success(function(data, status, headers, config)
```

```
    {
```

```
        if(data.success == 1)
```

```
        {
```

```
            alert("Submitted successfully");
```

```
            window.location = "admin_post_info.html";
```

```
            return;
```

```
        }
```

```
    } else
```

```
{
    alert("Invalid Inputs");
}

});
}

/*****
/***** User Login *****/
/*****

// sign in button

$scope.newpassword = function()
{
$http.post('newpassword.php',
    {
        'email': $scope.email, 'password': $scope.password,
        'field_3': $scope.field_3, 'field_4': $scope.field_4
    })
.success(function(data, status, headers, config)
{
    if(data.success == 1)
    {
        alert("Password Reset Successful");
        window.location = "index.html"; // Home Page
        return;
    }
    else if(data.success == 2)
    {
        alert("Please Fill All Fields");
    }
    else
    {
```



```
alert("Login Unsuccessful");
```

```
}
```

```
});
```

```
}
```

```
});
```

