HAND FOR THE WORLD

A PROJECT REPORT

Submitted by

SENTHIL S

in partial fulfilment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

HAJI C.H.M.M. COLLEGE FOR ADVANCED STUDIES
CHAVARCODE, PALAYAMKUNNU P O – 695146
THIRUVANANTHAPURAM DIST KERALA

UNIVERSITY OF KERALA, THIRUVANANTHAPURAM

AUGUST 2023

HAJI C.H.M.M. COLLEGE FOR ADVANCED STUDIES
CHAVARCODE, PALAYAMKUNNU P O – 695146
THIRUVANANTHAPURAM DIST KERALA

MASTER OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

Certified that this project report **"HAND FOR THE WORLD** "is the bona fide work of **SENTHIL S** who carried out the project work under my supervision.

Reg.No :95521801033

Mr.HARIKRISHNAN.S R	Mrs. SARANYA R B
Associate Professor	Assistant Professor
HEAD OF THE DEPARTMENT	**INTERNAL GUIDE**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I would like to express my gratitude to God for giving me good health and better courage to accomplish this project successfully.

I express my sincere gratitude to the Director of **MCA Prof. (Dr). SIRAJUDEEN. M,** for providing me an opportunity for doing this project work.

Special thanks to **Mr. HARIKRISHNAN S R ,**Associate Professor, Head of Department for his expert and valuable advice, inspiration and facilities rendered throughout for successful completion of the project.

I take this opportunity to express my sincere gratitude and indebtedness to my Internal guide **Mrs.SARANYA R B,**Assistant Professor, Department of MCA for providing all possible fruitful discussions to make this project be a success.

With great pleasure I may record my deep gratitude to my parents, friends and to all staff members of **MCA Department** for the immeasurable help rendered to me during the course of the project.

SENTHIL S

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

Tables

LIST OF FIGURES

Figure

ABSTRACT

our primary purpose is to coordinate response to natural or man man made disastersand for capacity-building in disaster resiliency and crisis respons. agency is responsible for framing policies, laying down guidelines and best-practices and coordinating with the State Disaster Management Authorities (SDMAs) to ensure a holistic and distributed approach to disaster management

We introduce our site to help the all the disaster team to find out the persons and overcome all the problems much more efficient way here all the site will help to give Guidelines on Management of School Safety,Guidelines on Management of Hospital Safety,Guidelines on Minimum Standards for Shelter, Food, Water, Sanitation, Medical Cover in Relief Camps,Guidelines on Management of Earthquakes,Guidelines on Management of Tsunamis,Guidelines on Management of Cyclones,Guidelines on Management of Flood,Guidelines on Management of Urban Flooding,Guidelines On Drought Management

If anything is happened all the data will be lost so with the help of our site we can simply overcome it

CHAPTER 1

INTRODUCTION

1.1 COMPANY PROFILE

Softzane Solution, Pvt Ltd is an IT solution provider specialized in the development of software and web applications, aiming to deliver quality services to customers.

We design and develop serious software for ambitious clients. Digital Marketing, Software Development, Designing… over the past years we've been on the leading edge of these and many other trends, all while staying true to our enduring principles.

While we would be building software even if it weren't our job, it is our job, and we take it very seriously. We have a duty to build the best software we can for our clients and their users. At the same time, professional integrity requires that we acknowledge that the software we write can always be better.

We're glad that software engineering and design isn't merely about pretty pictures and algorithms on a whiteboard. The products we build run real businesses and serve real users. We like it that way, because, while we can be pretty geeky, we actually like people, so we want to build useful software that makes them happy.

They providing the service are;

- Software Development

  Industry specific custom Software development solution for business enhancement and good will maintenance for customer satisfaction.

- Web Development
- Advanced Tech

  Our advanced technology features Artificial Intelligence and other main technology solutions for industry 4.0.

- Advisory

  Having right advices from experts and services makes your firm topper; we provide many marketing services which needed to your business growth.

1.2 STATEMENT OF THE PROBLEM

1.2 Statement of the Problem:

In the face of increasing occurrences of natural and man-made disasters, the need for efficient disaster response and crisis management has become more pressing than ever before. The existing framework for disaster management lacks a comprehensive and centralized system that can effectively coordinate response efforts, provide timely information, and offer guidelines for disaster resiliency. This deficiency leads to challenges in terms of resource allocation, timely communication, and standardized crisis management practices.

Furthermore, the absence of a unified platform for disaster response and coordination impedes the ability of National Disaster to facilitate a holistic and distributed approach to disaster management. Critical data related to disaster response strategies, guidelines for various scenarios, and contact information for disaster response teams are scattered across various sources and may be susceptible to loss during emergencies.

While National Disaster plays a pivotal role in policy formulation and coordination, there is a noticeable gap in providing practical and accessible tools for disaster management teams, State Disaster Management Authorities (SDMAs), and other stakeholders. The lack of a centralized platform hinders the efficient utilization of resources, impedes knowledge sharing, and potentially prolongs the response and recovery phases during crises.

In light of these challenges, the introduction of a dedicated website serves as a potential solution to bridge the gaps in disaster management. This platform aims to bring together guidelines, best practices, and critical information necessary for efficient disaster response, while also acting as a repository for resources that could be otherwise lost in the event of a disaster. By offering a comprehensive and accessible online resource, National Disaster seeks to enhance disaster resiliency, streamline coordination, and empower disaster management teams to respond effectively to emergencies.

CHAPTER 2

SYSTEM ANALYSIS

System Analysis is a detailed study of various operation performed by a system and their relationship within and outside of the system. Here the key question is: What must be done to solve the problem? One aspect of analysis is defining the boundaries of a system and determining whether or not a candidate system should consider other related system. Analysis begins when a user or manager begins a study of the program using an existing system.

During analysis, data is collected on the various files, decision points and transactions handled by the present system. The commonly used tools in system are dataflow diagrams, interviews, onsite observations etc. System analysis is application of the system approach to the problem-solving using computers. The ingredients are the system elements, process and technology. This means that to do system works, one is to understand the system concepts and how the organizations operate as a system and the design appropriate computerbased system that will meet the organizations requirements. It is actually customized approach to the use of computer problem solving.

Analysis can be defined as the separation of a substance into parts for study an interpretation, detailed examination. System development revolves around a lifecycle that being with the recognition of user needs. The critical phase of managing system project is planning. To launch a system investigation, we need a master plan detailing the steps taken, the people to be questioned and outcome expected.

System analysis can be categorized into four parts:

- System planning and initial investigation.
- Information gathering.
- Applying Analysing tools for structured analysis.
- Feasibility study.
- Cost/ Benefits analysis

System analysis begins when a user or manager request a studying of a program in either an existing system or a project one. It involves studying the base of the organizations currently operating, retrieving and processing data to produce information with goal of about determining how to make it work better. System analysis itself breaks down into stages preliminary and detailed. During preliminary analysis, the analyst and the user list the objectives of the system .If management approves preliminary report, the system analysis or study phases advantage to the second stage, the detailed analysis. If the management approves the result of a preliminary analysis, the analyst conducts a

detailed analysis, gathering facts about the old system, outline objectives for a new one, estimating costs, listing possible alternatives and making recommendation

Thus, the objective of analysis phase of the system analysis and design exercise is the establishment of the requirement for the system to be acquired, developed and installed. Fact finding or gathering is essential of requirements. In brief analysis of the system helps an analyst to make a clear view of an existing system and thereby can give suggestions for the improvement of the new system information about the organization's policies, goals, objectives and structure explains the kind of environment that promotes the introduction of computerbased system

2.1 PRESENT SYSTEM

. Present System:

The current state of disaster management operates within a framework that involves multiple entities, policies, and processes, but lacks a cohesive and centralized approach. The responsibilities are distributed among various stakeholders, including National Disaster, State Disaster Management Authorities (SDMAs), local government bodies, and other relevant organizations. While there are efforts in place to respond to disasters, the system faces several limitations:

1.Lack of Centralization: The current system lacks a unified platform that brings together critical information, guidelines, and resources for disaster management. Important data and resources are often scattered across various sources and might not be easily accessible during emergencies.

2. Limited Coordination: The decentralized nature of the present system can lead to challenges in terms of effective coordination and communication between different entities involved in disaster response. This may result in inefficiencies, overlapping efforts, or gaps in the response process.

3. Inconsistent Practices: Different regions and entities might adopt varying disaster management practices, leading to inconsistencies in preparedness and response strategies. This lack of standardization can impact the effectiveness of crisis management.

4. Resource Allocation Challenges: The absence of a centralized system for resource allocation can lead to delays in providing assistance to affected areas. Efficient distribution of resources such as relief materials, medical supplies, and personnel might be hindered.

5. Data Vulnerability: Since important data related to disaster response plans, guidelines, and contact information are often stored across multiple platforms, there's a risk of data loss during emergencies, making it difficult to access crucial information when needed the most.

6. Limited Accessibility: The current system might not always provide easily accessible information and guidelines to disaster management teams, first responders, and other stakeholders. This can lead to delays in decision-making and response efforts.

7. Lack of Real-time Information:The absence of a centralized system might result in delays in disseminating real-time information about disaster situations, hindering prompt decision-making and response.

Overall, the present system of disaster management, while having its strengths, is in need of a more streamlined, integrated, and accessible approach. The introduction of a dedicated website could potentially address these challenges by providing a centralized hub for information, guidelines, and resources, fostering better coordination, standardization, and improved crisis response capabilities.

2.2 LIMITATIONS OF PRESENT SYSTEM

The limitations of the present system in disaster management are crucial to recognize, as they highlight the areas that need improvement and serve as the basis for proposing enhancements. Here are some limitations of the current system:

1. Fragmented Information: Critical information, guidelines, and resources are scattered across various sources, making it difficult for disaster management teams to access comprehensive and up-to-date data during emergencies.

2. Inefficient Communication: Lack of a centralized communication channel can lead to delays and miscommunication between different stakeholders involved in disaster response, hampering coordinated efforts.

3. Resource Inefficiency: The absence of a standardized resource allocation system can result in inefficient distribution of resources like relief supplies, medical assistance, and personnel to affected areas.

4. Disparate Practices: Different regions and entities might follow varied disaster management practices, leading to inconsistent preparedness and response strategies, reducing the overall effectiveness of the system.

5. Data Vulnerability: Since data is stored in different locations, there's a risk of data loss during disasters, hindering the access to crucial information that is necessary for quick and informed decision-making.

6. Limited Accessibility: Information and guidelines might not always be easily accessible to disaster management teams and first responders, potentially leading to slower response times and inadequate decision-making.

7. Delayed Real-time Information: Without a centralized platform, disseminating real-time information about disaster situations might be delayed, affecting the ability to make prompt and informed decisions.

8. Reduced Standardization: The lack of a standardized approach across different regions can result in varied levels of preparedness, response, and recovery, impacting the overall effectiveness of disaster management efforts.

9. Inadequate Training: Without a centralized repository of training materials and best practices, the training of disaster management teams might lack standardization and uniformity.

10.Missed Opportunities for Improvement: A lack of centralized data and feedback mechanisms can hinder the collection of insights and lessons learned from past disasters, impeding the ability to continuously improve response strategies.

Recognizing these limitations presents an opportunity to address them and create a more streamlined, efficient, and effective disaster management system. The introduction of a dedicated website with centralized information, guidelines, and resources could serve as a solution to overcome these challenges.

2.3 PROPOSED SYSTEM

The proposed system aims to address the limitations of the current disaster management framework by introducing a centralized and comprehensive online platform. This platform, in the form of a dedicated website, will serve as a hub for disaster management teams, State Disaster Management Authorities (SDMAs), first responders, and other stakeholders. The proposed system will offer several key features and benefits:

1. Centralized Information Hub:The heart of the proposed system is a centralized repository of information, guidelines, and resources related to disaster management. This includes guidelines for various disaster scenarios, best practices, contact information for disaster response teams, and relevant policies.

2. Efficient Communication: The website will provide a streamlined communication channel, facilitating real-time information sharing among different stakeholders. This will enhance coordination and collaboration during disaster response and recovery phases.

3. Resource Allocation Optimization: The proposed system will include tools for efficient resource allocation and distribution. This will prevent duplication of efforts and resources, ensuring that aid reaches affected areas promptly.

4. Standardization of Practices: By offering standardized guidelines and best practices, the system will promote consistent disaster preparedness and response strategies across different regions, improving the overall effectiveness of the system.

5. Data Security and Accessibility: The centralized platform will ensure data security and accessibility. Information will be available to authorized users, reducing the risk of data loss during emergencies.

6. Real-time Information Dissemination:The system will enable the dissemination of real-time information about disaster situations, enhancing decision-making and response capabilities.

7. Training and Capacity Building: The proposed system will provide training materials, simulations, and resources for capacity building among disaster management teams, ensuring uniformity in training efforts.

8. Lessons Learned and Continuous Improvement: The platform will facilitate the collection and analysis of insights and lessons learned from past disasters. This will support the refinement of response strategies over time.

9. Holistic Approach: With the involvement of National Disaster, the proposed system will facilitate a holistic approach to disaster management, ensuring that policies, guidelines, and resources are aligned at both the national and local levels.

10. User-friendly Interface: The website will feature a user-friendly interface designed for easy navigation, information retrieval, and interaction. It will cater to users with varying levels of technical expertise.

In conclusion, the proposed system aims to revolutionize disaster management by providing a centralized, accessible, and efficient platform that empowers disaster management teams to respond effectively to crises. By addressing the limitations of the current system, this innovative approach promises to enhance coordination, standardization, and resiliency in disaster management efforts.

2.4 ADVANTAGES AND FEATURES OF PROPOSED SYSTEM

3. Advantages and Features of the Proposed System:

The proposed centralized platform for disaster management offers a range of advantages and features that significantly improve the efficiency, coordination, and effectiveness of disaster response and recovery efforts. Here are the key advantages and features:

Advantages:

1. Comprehensive Information Repository:The platform serves as a single source of truth for disaster management information, guidelines, and resources, ensuring that all stakeholders have access to accurate and up-to-date information.

2. Enhanced Coordination: The centralized system promotes seamless communication and coordination among National Disaster, State Disaster Management Authorities (SDMAs), local government bodies, and other stakeholders, minimizing confusion and inefficiencies.

3. Standardized Practices:By offering standardized guidelines and best practices, the system ensures that disaster response strategies are consistent and effective across different regions, leading to improved crisis management outcomes.

4. Efficient Resource Allocation: The platform's resource allocation tools prevent duplication of efforts, optimize resource distribution, and ensure timely provision of aid to affected areas, enhancing the overall response efficiency.

5. Real-time Information Dissemination: The system enables rapid dissemination of real-time information about disaster situations, facilitating prompt decision-making and enabling quick response efforts.

6. Access to Training Materials: The platform provides training materials, simulations, and resources to disaster management teams, enhancing their readiness and capacity to handle emergencies.

7. Lessons Learned and Continuous Improvement: By capturing insights and lessons learned from past disasters, the system supports continuous improvement of response strategies, minimizing errors and enhancing effectiveness over time.

8. Holistic Approach:With National Disaster's involvement, the platform fosters a holistic approach to disaster management, ensuring that policies, guidelines, and resources are aligned at both national and local levels.

9. Data Security and Accessibility:The platform offers secure data storage and authorized access, reducing the risk of data loss during emergencies and enabling stakeholders to retrieve critical information.

Features:

1.Centralized Repository: A centralized hub for disaster management information, guidelines, policies, and contact details for relevant stakeholders.

2.Communication Tools:Real-time communication channels for efficient collaboration and information sharing among different disaster management entities.

3. Resource Allocation Tools: Tools to optimize resource allocation, ensuring that aid is distributed effectively and without duplication.

4. Standardized Guidelines: Standardized guidelines and best practices for various disaster scenarios, enabling consistent and effective response strategies.

5. Real-time Information Updates: Real-time updates on disaster situations, weather conditions, and other critical information for informed decision-making.

6. Training and Capacity Building:Access to training materials, simulations, and resources for capacity building among disaster management teams.

7. Lessons Learned Repository:A repository to capture and share insights from past disasters, facilitating continuous improvement.

8. User-friendly Interface:An intuitive and user-friendly interface designed for easy navigation and interaction.

9. Multi-device Accessibility: Accessible from various devices, ensuring stakeholders can retrieve information from anywhere.

10. Data Security: Robust data security measures to safeguard critical information during emergencies.

Incorporating these advantages and features, the proposed system transforms disaster management into a more efficient, coordinated, and resilient process, contributing to better preparedness and response capabilities.

FUNCTIONS OF PROPOSED SYSTEM

Functions of the Proposed System

The proposed centralized platform for disaster management is designed to provide a range of functions that streamline disaster response, coordination, and preparedness efforts. These functions aim to empower National Disaster, State Disaster Management Authorities (SDMAs), first responders, and other stakeholders with the tools they need to effectively manage and mitigate the impact of disasters. Here are the key functions of the proposed system:

1. Information Repository:

- Centralized storage of disaster management information, including guidelines, best practices, policies, and contact details.
- Categorization of information for easy retrieval based on disaster types, response phases, and roles.

2. Communication and Collaboration:

- Real-time communication channels for stakeholders to share updates, coordinate efforts, and exchange information.
- Collaboration features such as discussion forums, group chats, and announcement boards.

3. Resource Allocation Optimization:

- Resource allocation tools that analyze disaster impact, resource availability, and needs to optimize distribution.
- Monitoring and tracking of resource deployment to prevent duplication and ensure efficient utilization.

4. Guidelines and Best Practices:

- Comprehensive guidelines and best practices for different disaster scenarios, aiding decision-making and response planning.
- User-friendly access to step-by-step instructions for effective crisis management.

5. Real-time Information Updates:

- Real-time updates on disaster situations, weather forecasts, evacuation notices, and other critical information.
- Notifications and alerts sent to authorized users based on their roles and responsibilities.

6. Training and Capacity Building:

- Access to training materials, simulations, and interactive modules for disaster management teams and first responders.

- Progress tracking and assessments to evaluate training effectiveness.

7. Lessons Learned Repository:

- Repository for capturing insights, lessons learned, and post-disaster evaluations.

- Sharing of experiences to foster continuous improvement and enhance response strategies.

8. Standardization and Policy Implementation:

- Implementation of standardized practices and policies across different regions and entities.

- Monitoring and enforcement of compliance with disaster management regulations.

9. Secure Data Storage:

- Robust data security measures to ensure the confidentiality, integrity, and availability of critical information.

- Secure user authentication and role-based access controls.

10. Accessibility Across Devices:

- Multi-device accessibility, allowing stakeholders to access information and updates from smartphones, tablets, and desktops.

- Mobile app support for on-the-go access.

11. Holistic Approach:

- Integration of policies, guidelines, and resources from National Disaster and SDMAs to ensure a cohesive approach to disaster management.

12. Disaster Analytics and Reporting:

- Data analytics capabilities to analyze disaster trends, response effectiveness, and resource allocation outcomes.

- Generation of reports and visualizations for informed decision-making and policy formulation.

By providing these functions, the proposed system equips disaster management stakeholders with the tools they need to respond efficiently, coordinate effectively, and improve their overall preparedness and response capabilities.

2.5 FEASIBILITY STUDY

The main objective of the feasibility study is to test the economical, technical and operational feasibility while developing the system. This analysis is done by investigating the existing system in the

area under generating an idea about the new system. Feasibility study is a test of proposed system regarding its workability, impact on the organization, ability to meet the needs and effective use of resources. Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development. The study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that is spend on it.

The development of a computer-based system or product is more likely plagued by resources and difficult delivery data. It is both necessary and prudent to evaluate the feasibility of a project at a time months and years of effort, thousands or millions and untold professional embarrassment can be averted if ill-convinced system is recognized early in the definition phase.

All the projects are feasible given unlimited resources and infinite time. Unfortunately, the development of the computer-based system is more likely to be played by a security of resources and difficulty delivery dates. Feasibility and risk analysis are related in many ways.

Three are three phases:

1. Operational feasibility.
2. Technical feasibility
3. Economic feasibility

2.5.1 Operational feasibility

The operational feasibility depends up on whether system performed in the expected way or not. The application developed is so simple and user friendly there is no special user training is required. So, this application can be said to be operationally feasible. The proposed system is very much user friendly and operations on it can be done very easily. The language used in is English. This system has the capability to solve complex manual problems and can achieve its operational objective It is also user friendly and don't require technical knowledge to use. Moreover, it will save time with its effectiveness and efficiency.

The new system is very much easier and user friendly than the existing system. There is no barrier for implementing the system. For the system to work network is mandatory. The system also helps to access the information immediately as need arises. Thus, the system is found to be operational feasible.

2.5.2 Technical feasibility

The site must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the site requirement in the terms of input, output, programs and procedures. Technical feasibility centres around computer system and to what extend it can support the proposed addition. For example, if the current computer system is operating at 80% capacity, then running another application could overload the system or requires additional hardware. This involves the financial considerations to accommodate the additional technical enhancements.

If budget is not a serious constraint, then the project is judged technically feasible. Sinceno further addition of hardware or software is needed. Technical feasibility centers on the required existing computer system (Hardware/Software) and to what extent it can support the proposed application. It concerned with specifying equipment's and software that will successfully support the required task.

The main points that are considered to prove that the project is technically feasible are:

1. Currently following method is sufficiently to develop project
   .
2. The proposed system provides less time-consuming way to retrieve data.

3. The system can be expanded and develop.

4. A user friendly interface, which can be operated by anyone with minimum computer knowledge.

5. The project outputs given are reliable and it is easy to access.

2.5.3 Economic feasibility

The developing system must be justified cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. This deals with whether expected cost saving, increase the profits and reductions in required investment, and other benefits exceed the cost of developing and operating the proposed system. Its preliminary investigation is

concentrated on costs of hardware and software. This system is developed using Python and MYSQL. Here an evaluation of development cost weight against the ultimate income or benefit derived from the developing system. One of the factors which affect the development system is the cost it would require. Since the system is developed as part of the project work there is no manual cost to spend for the proposed system. The proposed system satisfies the economic feasibility

CHAPTER 3

SYSTEM SPECIFICATIONS

3.1 HARDWARE REQUIREMENTS

Processor	:	Intel i5 7 th Gen
RAM	:	8 GB DDR4
Hard Disk	:	256 GB SSD
Display Size	:	Compatible Size (Recommend 15'inch)
Screen Resolution	:	1920 * 1080
Pixels Keyboard	:	Wireless Enabled Keyboard (Recommend: Logitech)
Mouse	:	Wireless Enabled Mouse (Recommend: Logitech)
Monitor	:	Touch Capacity LED Monitor Dedicated
Graphics Card	:	Nvidia GeForce 920m 2GB DDR4

3.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows 11
Platform	:	Visual Studio Code
Language	:	Python
Framework	:	Django
Frontend	:	HTML, CSS, JS
Database	:	SQLite3
Backend	:	Python
Web Browser	:	Google Chrome

CHAPTER 4

SYSTEM DESIGN

Design is the second phase in the system development life cycle. Software design is the first of the three technical activities in the software development process such as design, code writing and testing.

During this phase, the analyst schedules design activities, works with the user to determine the various data inputs to the system, plans how data will flow through the system, designs required outputs and writes program specifications. Again the analyst's activities focus on solving a user's problem in logical terms. During this second step, analysts employ a variety of tools such as data flow diagrams, entity-relationship diagrams, data dictionaries and Gantt chart. The system's design converts the theoretical solution introduced by the feasibility study into a logical reality.

During design the analyst

1. Draws a model of the new system, using data flow and entity-relationship diagrams

. 2. Develop methods for collecting and inputting data.

3. Defines the detailed data requirements with a data dictionary.

4. Writes program specifications.

5. Specifies control techniques for the system's outputs, databases and inputs.

6. Identifies and orders any hardware or software that the system will need.

In the physical design phase, necessary software is developed to accept input from the user, to perform necessary calculations through the manipulation of data stored in the databases to produce the appropriate result

There are two levels of system design:

- Logical design.
- Physical design.

In the logical design, the designer produces a specification of the major features of the system which meets the objectives.

The delivered product of logical design includes current requirements of the following system components:

• Input design.

• Output design

• Database design

Physical design takes this logical design blue print and produces the program software, files and a working system. Design specifications instruct programmers about what the system should do. The programmers in turn write the programs that accept input from users, process data, produce reports, and store data in files. Structured design is a data flow-based methodology that partitions a program into a hierarchy of modules organized top-down manner with details at the bottom. Data flow diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes may be described logically and independently of the physical components.

4.1 Module Description

Modules

1) Person Finding Module
   This module will be help the persons to find any missing persons in the disaster time .
   They can post their basic details of a person to this site.
   The system will check the details is present in the disaster camp database ,if it present the system will give an alert to the corresponding persons
   Else the post will be transfer to the police stations and the rescue teams for the recovery operations.
   This module will avoid the disadvantage of social media [automatically post deletion when the team find a persons]
2) Fund allocation module
   The fund allocation process can be done in different ways , the present time the fund transferring is the most time consuming procedure from the government side . this problems will be easily avoided with the help of our system
   Here we categorize the people in two ways
   The first team, the persons in the rescue camp they can apply their basic recovery fund applications through our site, the fund will be transfer to their accounts after they are leaving form their camp. The amount always treated as primary relief
   The Second team; the persons they are not in the camp: here we give them better help for them to get the government benefits, they can send an application though online here the amount will not pass directly to their accounts , in this case the application will get a special government employee and after the report submission of the employee the applicant get the corresponding amount
   The secondary fund transferring for house rebuilding, material purchasing or the new shelter arrangements this amount only transferring after the detailed case of the government employee

Vehicle Module

Here we mainly focus on vehicle insurance procedure . after the disaster if any type of the damage is happened to the vehicle the insurance company should accept the application , the application should contain the pictures of the damaged vehicle ,after the direct enquiry from the insurance company the owner will get the insurance amount.

Their will be second chance for vehicle losing

In this time an alert or notification will pass to nearest police station and the same message will get to insurance company. If the police is find out the vehicle itt will be pass to the licensed owner

Login Module

Here we introduce a special login program, and this will helps to get all the details about a persons and with the help of this we can avoid normal login procedure and also restrict the unauthorized login procedure

Team Allocation Module

- Food collection team
- Dress collection team
- Medicine collection team
- Cleaning team

Flood alerting & notification Passing Module

Here the flooded areas will be pass to the rescue teams and the public with easy way and also here we support alert and notification to government to public

Modules

1. Admin Module
2. Station Module
3. Camp Module
4. Volunteer Module
5. Public Module

Module Description

Admin Module

- ❖ Camp Handling
- ❖ Station Handling
- ❖ Volunteer Handling
- ❖ Fund Allocation
- ❖ Details passing And verification

Camp Module

- ❖ Registration
- ❖ Needs passing
- ❖ Person details adding

- Emergency request and alert passing
- Communication
- Duty scheduling

Public module

- Registration
- Person finding
- Fund requesting
- Alert and Notification checking
- Confirmation adding
- Vehicle finding request passing
- Insurance requesting
- Complaint posting
- Photos and documents uploading

Volunteer Module

- Registration
- Communication
- Request checking
- Confirmation adding
- Duty list checking

Station module

- Registration
- Case checking
- Vehicle finding
- Person finding
- Alert passing
- Communication

4.3 DATA FLOW DIAGRAM

4.2 CONTEXT LEVEL DIAGRAM

Level 1.0

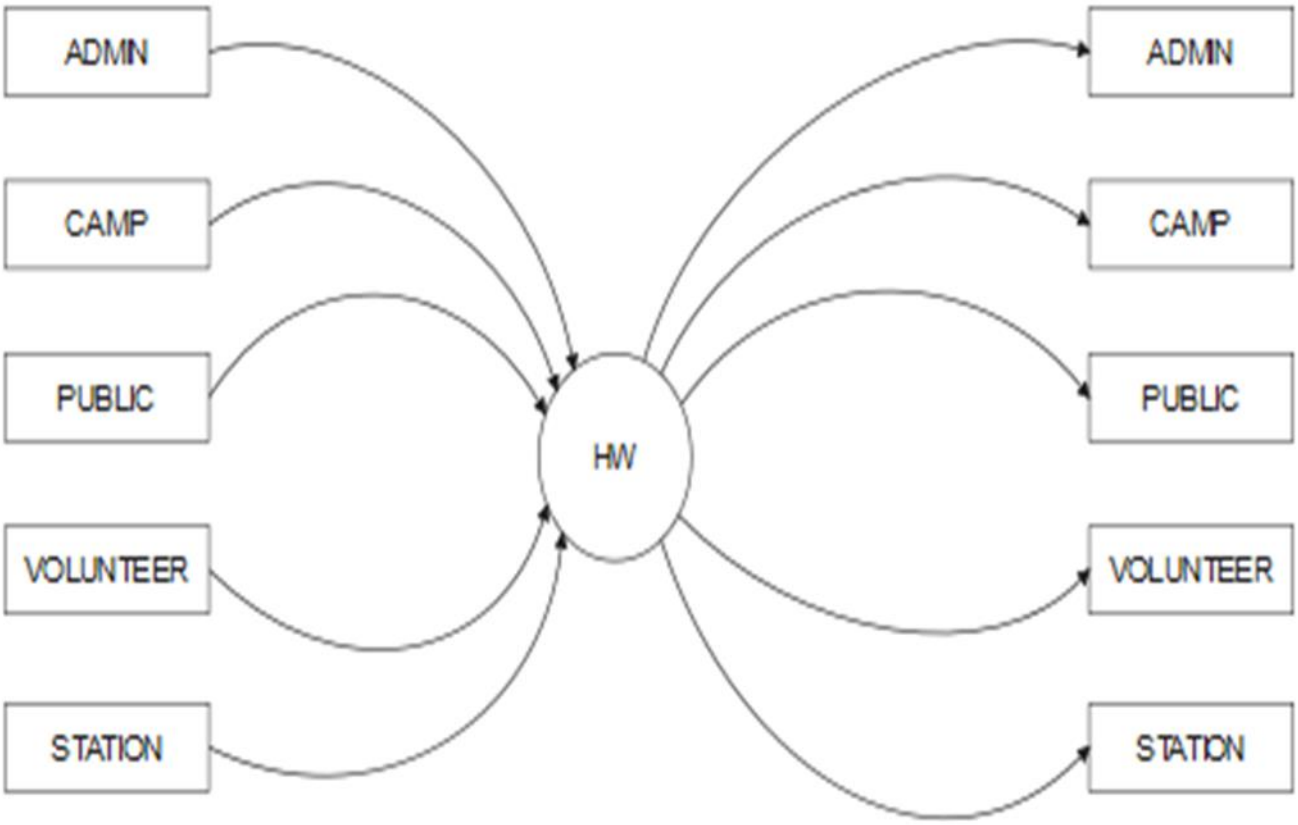


Fig 4.1 Context Level Diagram

Level 1.0

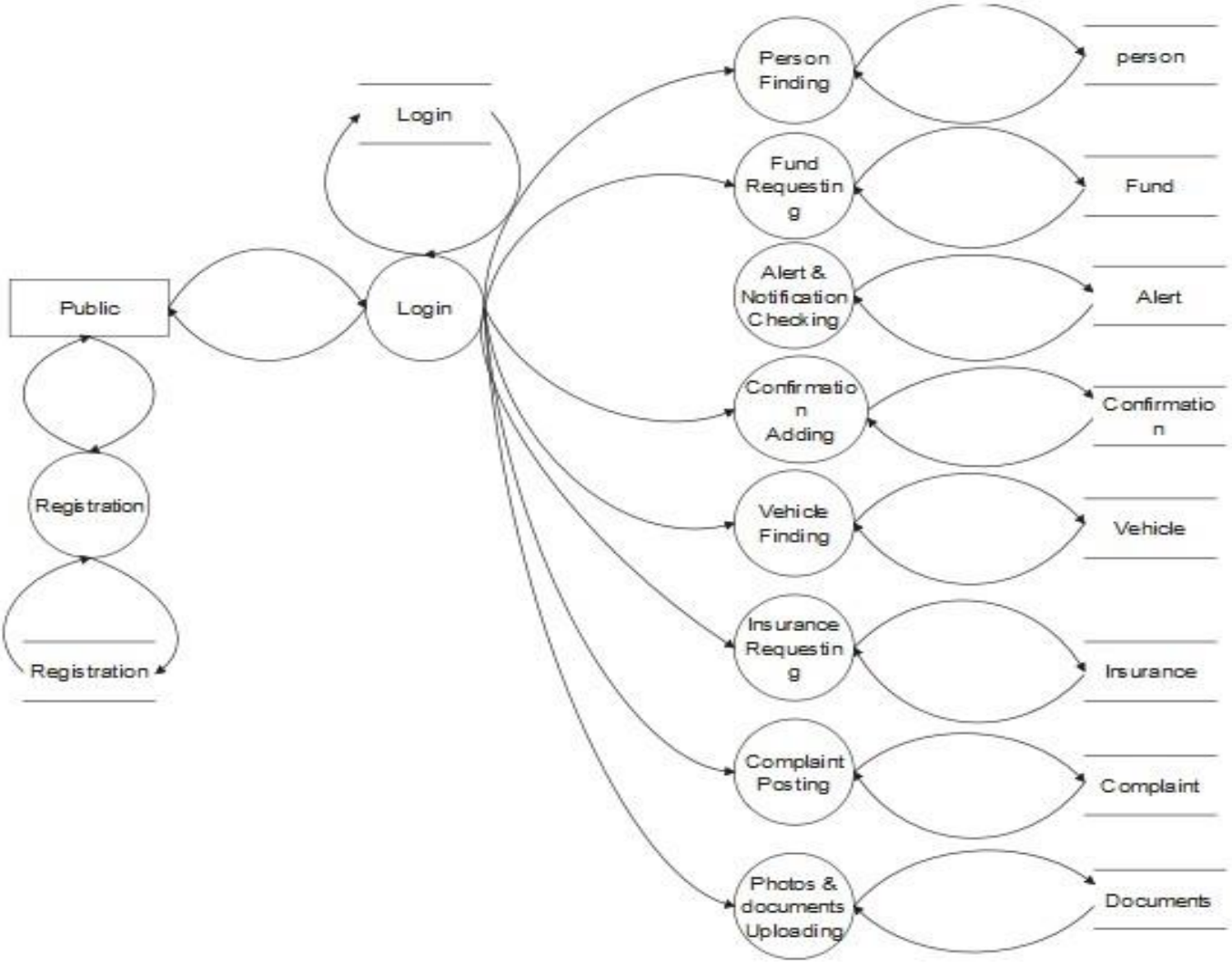


Fig 4.3.1 Level: 1

Level 1.1

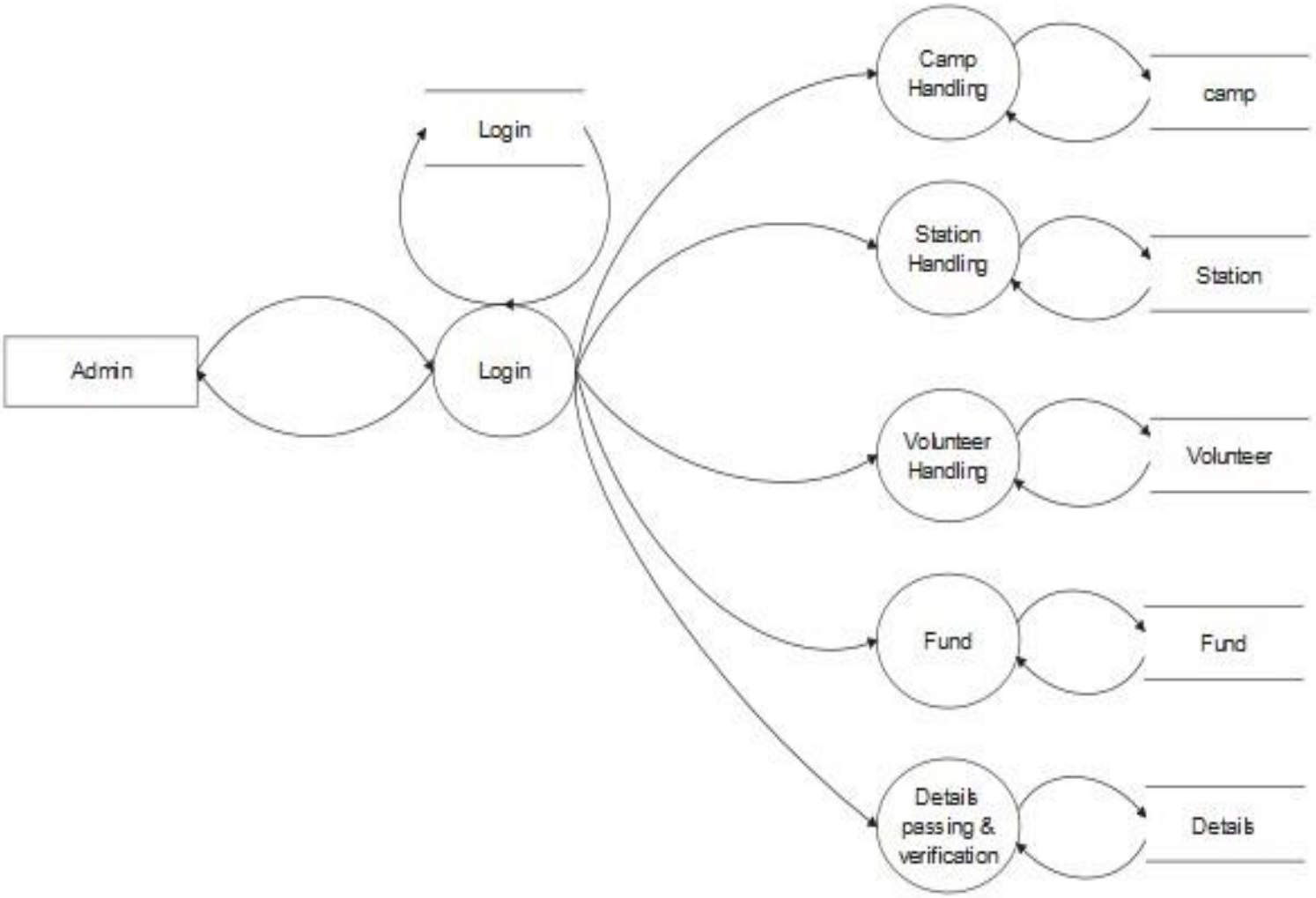
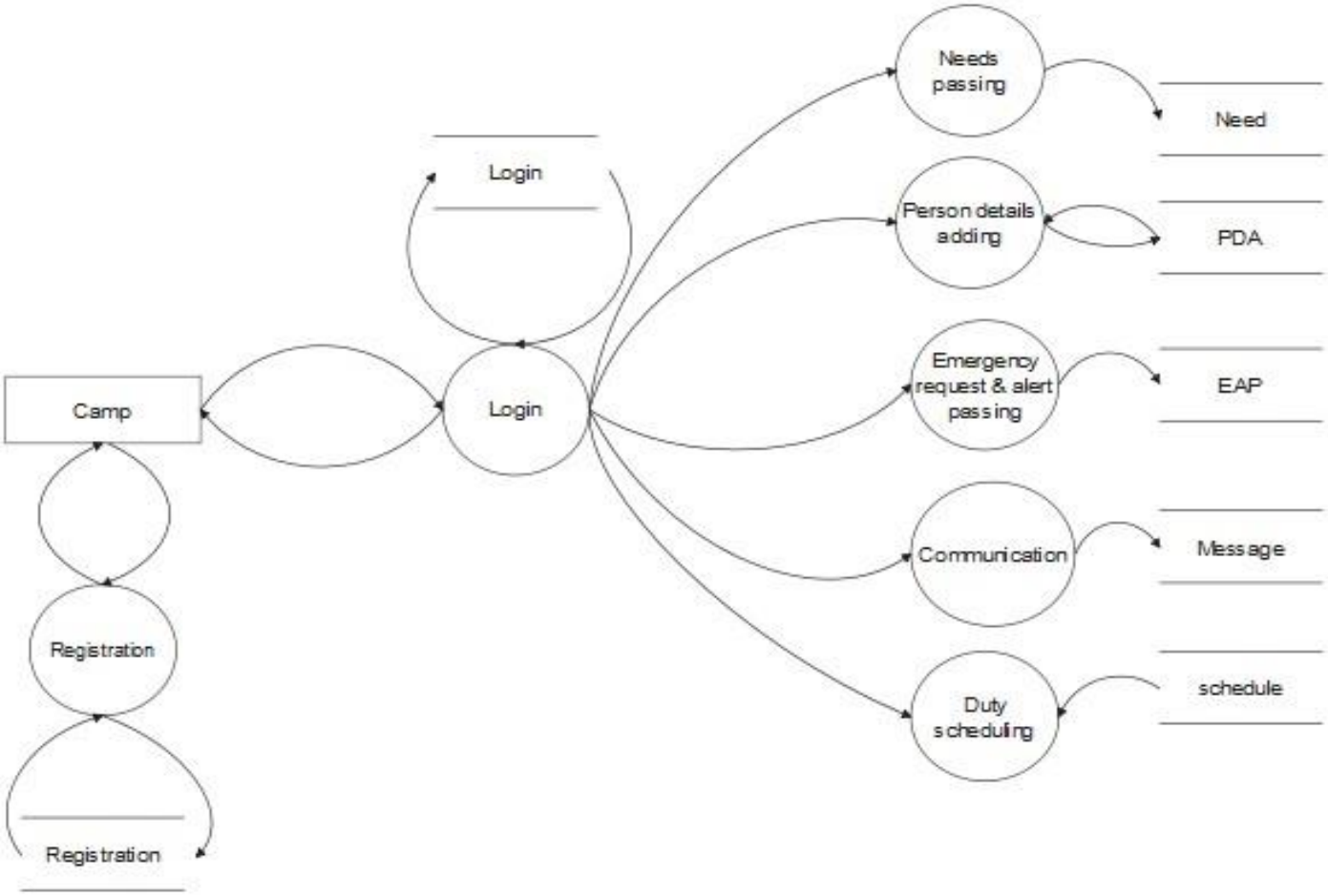
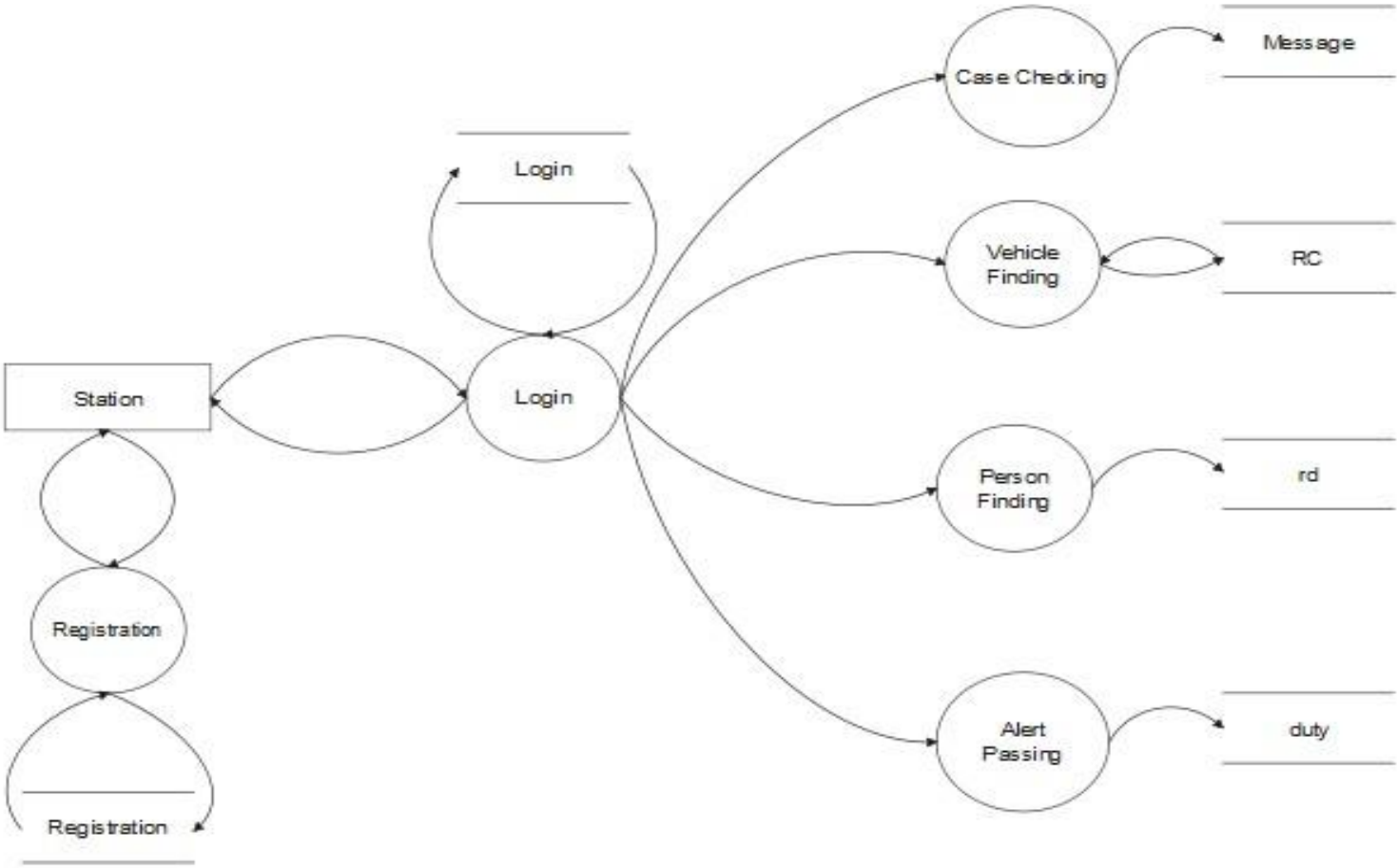


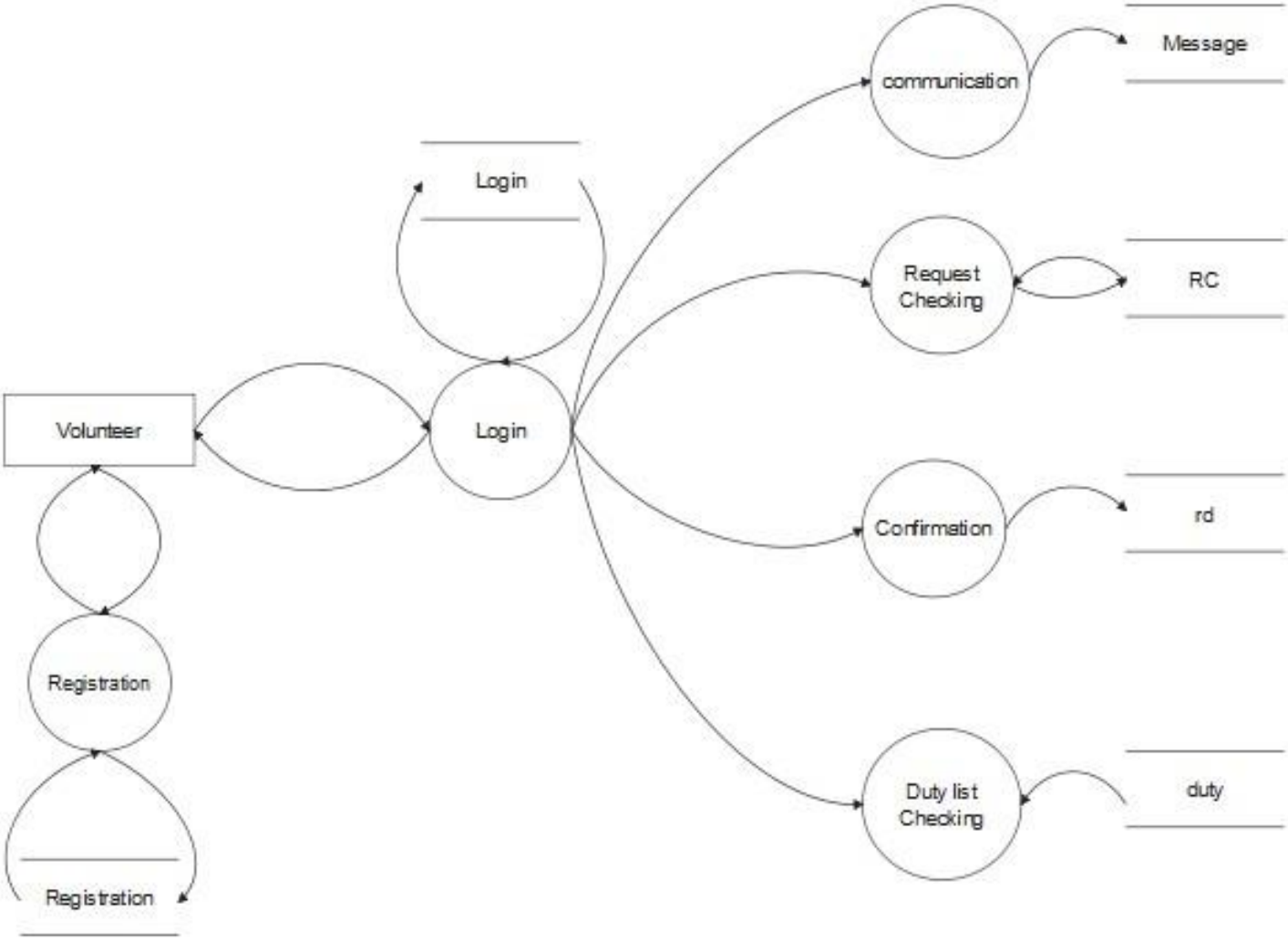
Fig 4.3.3 Level 1.1

Level 1.1.1

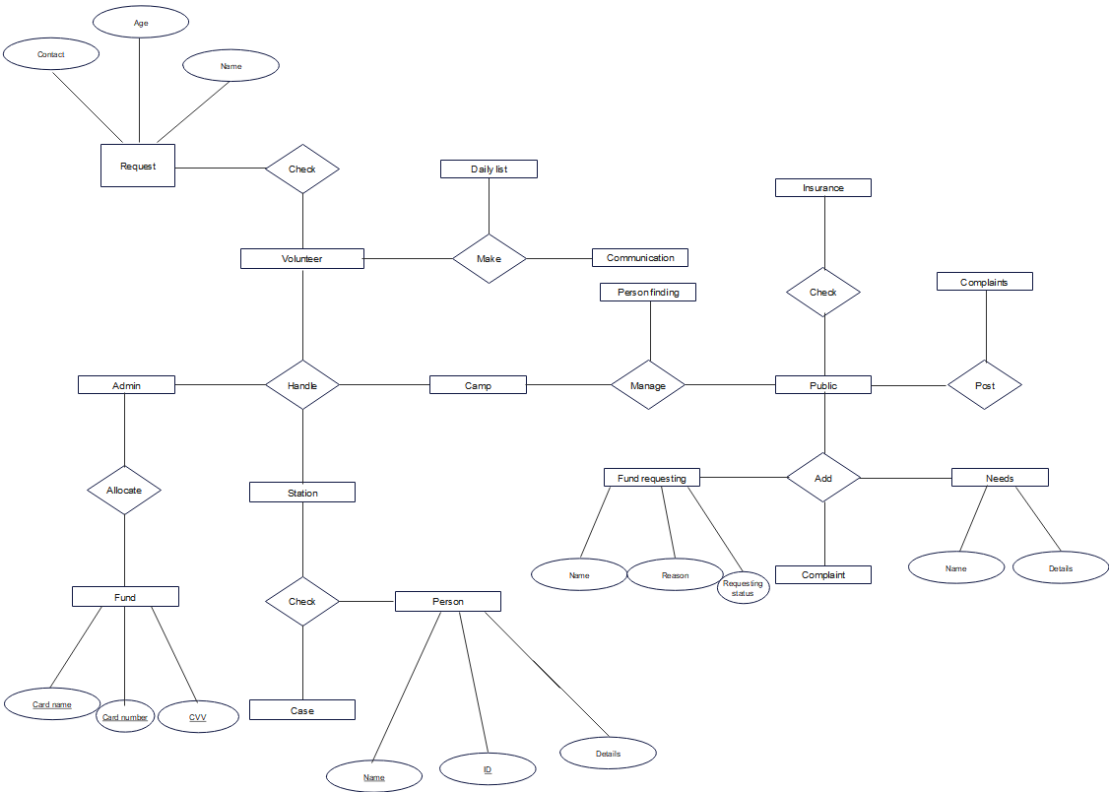

Needs passing
Need
Login
Person details adding
PDA
Emergency request & alert passing
EAP
Camp
Login
Communication
Message
Registration
schedule
Duty scheduling
Registration


Message
Case Checking
Login
Vehicle Finding
RC
Station
Login
Person Finding
rd
Registration
Alert Passing
duty
Registration

Level 1.2.


Message
communication
Login
Request Checking
RC
Volunteer
Login
Confirmation
rd
Registration
Duty list Checking
duty
Registration

4.4 ER-DIAGRAM



4.5 DATABASE DESIGN

A database is a collection of inter-related data stored with minimum redundancy, provides better data integrity and security and also to use many users quickly and efficiently. The general objective of database design is to make the data access easy, index endive and flexible to the user. Database design is recognized as a standard of management information system which is virtually available for every computer system.

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system.

The general theme behind a database is to integrate all the information. A database is the integrated collection of data and provides centralized access to data. Usually, the centralized data managing software use relational database concepts and hence called RDBMS

(i) Design Considerations

The system is analysed to the requirements and possible tables and fields are determined.

(ii) Identifying Keys

Once we have drawn upon the list of possible tables and fields, the next step in the logic database is to identify primary key and foreign key of the table.

(iii) Primary Key

The primary key (PK) of a relational table uniquely identifies each record in the table, it can either normal attribute that is guaranteed to be unique or it can be generated by the DBMS. Primary keys may consist of a single attribute or multiple attributes in combined ion.

- Product (product_id) is related to Inventory (product_id) in a one-to-many relationship.
- Vehicle (Vehicle_id) is related to Employee (employee_id) in a one-to-many relationship.

(iv) Foreign Keys

A foreign key (FK) is a key comprised of a field or multiple fields that to the primary key of another table. The concept of maintaining foreign keys is known as “referential integrity”.

- WareHouse in the "Supervisor" table, you can have "employee_id" as a foreign key referencing the "Employee" table's primary key.

(v) Defining Relationship

A relationship is the term used to describe a connection between related tables. Stated another way, it means having shared fields in different tables that allow records to reference records in other tables. There are three possible types of relationships.

(vi) One-to-One Relationships

A one-to-one relationship indicates that each record in the table may relate to only one in another table.

(i) One-to-Many Relationships

In a one-to-many relationship, any record in a table can relate to multiple records in a second table.

(ii) Many-To-Many Relationships

With many-to-many relationships many records in one table can link too many records in the second table.

(iii) Normalization

Normalization is a process of simplifying the database design to achieve the optimum structure. The steps in this process are known as normal form. These normal forms are a sequence of rules that are applied to progressively a database design. The higher the normal form of a database, the more efficient its underlying design. This is because, for a database to be simplified into third normal form, it must meet the criteria of first and second normal forms.

(iv) First Normal Form

To achieve first normal form, we must eliminate any repeating group. In the first normal form, we simplify our database structure to any repeating group. In other words first normal form include concept that field must be atomic or field represent one type of value for all the records.

- split "rack_bins" into its own table.

(v) Second Normal Form

To achieve second normal form, we must make sure that the non key fields depend on all the field in primary key every field in the table should be depend upon the entire primary key so that when new records are added, same value will not be repeated from records to records unnecessarily. Full functional dependency indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A, second normal form (2NF) is a relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key. The normalization of 1NF relations to 2NF involves the removal of partial

Dependencies. If a partial dependency exists, we remove the function dependent attributes from the relation by placing them in a new relation along with a copy of their determinant.

Database name: Hand for the World

Table Number: 4.5 .1

Field	Data type	Size	Constraint	Description
Id	Integer	20	Primary Key	id
Champ_Name	VARCHAR	20	NOT NULL	Name of Champ
Address	String	10	NOT NULL	Champ address
Pincode	VARCHAR	20	NOT NULL	Champ pincode
District	DATE	10	NOT NULL	District of Champ
City	VARCHAR	20	NOT NULL	City of Champ
Village	VARCHAR	25	NOT NULL	Village of Champ

Thaluk_Municipality_ Panchayathu	VARCHAR	25	NOT NULL	Thaluk/Municipality/Panchaya thu of Champ
Contact_Number	Integer	25	NOT NULL	Contact_Number of Champ
Email_ID	VARCHAR	25	NOT NULL	Email_ID of Champ
Password	VARCHAR	25	NOT NULL	Password of Champ

4.5.1Table 1:Champ

Field	Data type	Size	Constraint	Description
Station_ID	Integer	20	Primary Key	Supervisor id
Address_Line_1	VARCHAR	20	NOT NULL	Address_Line_1 of station
Address_Line_2	VARCHAR	25	NOT NULL	Address_Line_2 of station
Pincode	Integer	20	NOT NULL	Pincode of station
District	VARCHAR	25	NOT NULL	District of Station
City	VARCHAR	25	NOT NULL	City of Station
Panchayathu	VARCHAR	25	NOT NULL	Panchayathu of Station
Contact_Number	Integer	25	NOT NULL	Contact_Number of Station
Email_ID	VARCHAR	20	NOT NULL	Email_ID of Station
Password	VARCHAR	25	NOT NULL	Password of Station

4.5.2Table 2:Station

Field	Data type	Size	Constraint	Description
Public_id	Integer	20	Primary Key	Employee id
Name	VARCHAR	20	NOT NULL	Name of Employee

Cntoact_Number	Integer	20	NOT NULL	
Email_ID	VARCHAR	20	NOT NULL	Employee user
Password	VARCHAR	25	NOT NULL	Employee Password

4.5.3Table 3:Public

Field	Data type	Size	Constraint	Description
Volunteer_id	Integer	20	Primary Key	Default id
Name	String	20	NOT NULL	Name of Volunteer
Age	Integer	25	NOT NULL	Age of Volunteer
Gender	Integer	20	NOT NULL	Gender of Volunteer
Contact Number	Integer	25	NOT NULL	Contact of Volunteer
User_Name	String	25	NOT NULL	User_Name
Password	String	25	NOT NULL	Password of Volunteer

4.5.4Table 4:Volunteer

Field	Data type	Size	Constraint	Description
id	Integer	20	Primary Key	Default id
District	String	20	NOT NULL	Which District get alert
Alert	String	25	NOT NULL	About the alert
Currentdate	Integer	20	NOT NULL	Get from system

4.5.5Table 5:Alert

Field	Data type	Size	Constraint	Description
id	Integer	20	Primary Key	Default id
Name	String	20	NOT NULL	Name of Person
Address	String	25	NOT NULL	Address of Person
Pincode	Integer	20	NOT NULL	Pincode of Person
District	String	25	NOT NULL	District of Person
Village	String	25	NOT NULL	Village of Person
Panchayathu	String	25	NOT NULL	Panchayathu of Person
Contact_Number	Integer	25	NOT NULL	Contact_Number of Person
Aadhaar_Number	Integer	25	NOT NULL	Aadhaar_Number of Person

4.5.6 Table 6:Champ person

Field	Datatype	Size	Constraint	Description
Number_of_volunteers	Integer	20	Primary Key	Number_of_volunteers
Other_Details	String	20	NOT NULL	Other_Details
Currentdate	Integer	20	NOT NULL	Currentdate

4.5.7Table 7:Volunteer Request

.

Field	Data type	Size	Constraint	Description
Name	String	20	Primary Key	Name
Address	String	20	Foreign Key	Address
Pincode	Varchar	20	NOT NULL	Pincode
District	String	20	NOT NULL	District
City	String	20	NOT NULL	City
Panchayathu	String	20	NOT NULL	Panchayathu
Ration_Card_No	Integer	50	NOT NULL	Ration_Card_No
Aadhaar_Number	Integer	10	NOT NULL	Aadhaar_Number
Reason	String	20	NOT NULL	Reason
Image	file	30	NOT NULL	Image
Currentdate	date		NOT NULL	Currentdate
RequestStatus	String	20	NOT NULL	RequestStatus of request

4.5.8Table 8: Fund request

Field	Datatype	Size	Constraint	Description
Card_Name	Integer	20	Primary Key	Card_Name of card
Card_Number	Varchar	20	NOT NULL	Card_Number of card
CVV	Varchar	20	NOT NULL	CVV of card
Expire_Date	Varchar	20	NOT NULL	Expire_Date of card
Amount	Integer	20	NOT NULL	Amount
User_ID	Vaerchar	25	NOT NULL	User_ID
Current_date	Date	25	NOT NULL	Current_date

4.5.9Table 9**:** Payment

Input Design

Input designing is the basic theory to be considered during system study. The input media used in the system is the keyboard. Details are entered in the system through different data entry screens. The system is designed in a user-friendly manner. Appropriate error messages are displayed when a false data is entered. Design of the system is web-oriented and is highly interactive to the users. The user interface design is very important for any application. The interface design defines how the software communicates within itself, to system that interpreted with it and with human who use it. In this project all the fields are validated. If any field then error message will be displayed, so as to help the user while giving inputs. The drop-down lists are used to reduce the user inputs and to select a preferred item from the list easily. Check boxes are used for user's category selection.

In **'Warehouse Management System'** the input design is done in such a way that the users of the system will never get confused or enter wrong data. The simplicity and ease of use lies in the act that the desired objectives can be accomplished with a few mouse clicks.

Output Design

Computer output is the most important one to the user. A major form of the output is the display of the information gathered by the system and the servicing the user requests to the system. Output generally refers to the results or information that is generated by the system. It can be in the form of operational documents and reports. Since some of the users of the system may not operate the system, but merely use the output from the system to aid them in decision-making, much importance is given to the output design. Output generation hence serves two main purposes, providing proper communication of information to the users and providing data in a form suited for permanent storage to be used later on. The output design phase consists of two stages, output definition and output specification. Output definition takes into account the type of outputs, its contents, formats, its frequency and its volume. The output specification describes each type of output in detail. Efficient and well-defined output design improves the relationship of the system and the user, thus, facilitating design making.

4.5 UML DIAGRAMS

4.6.1 Use Case Diagram

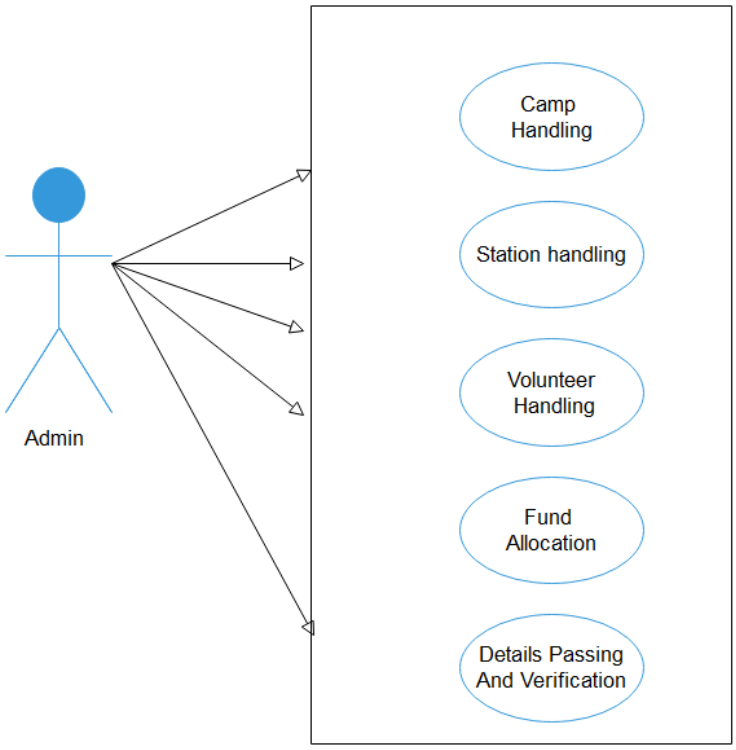


fig 4.6.1

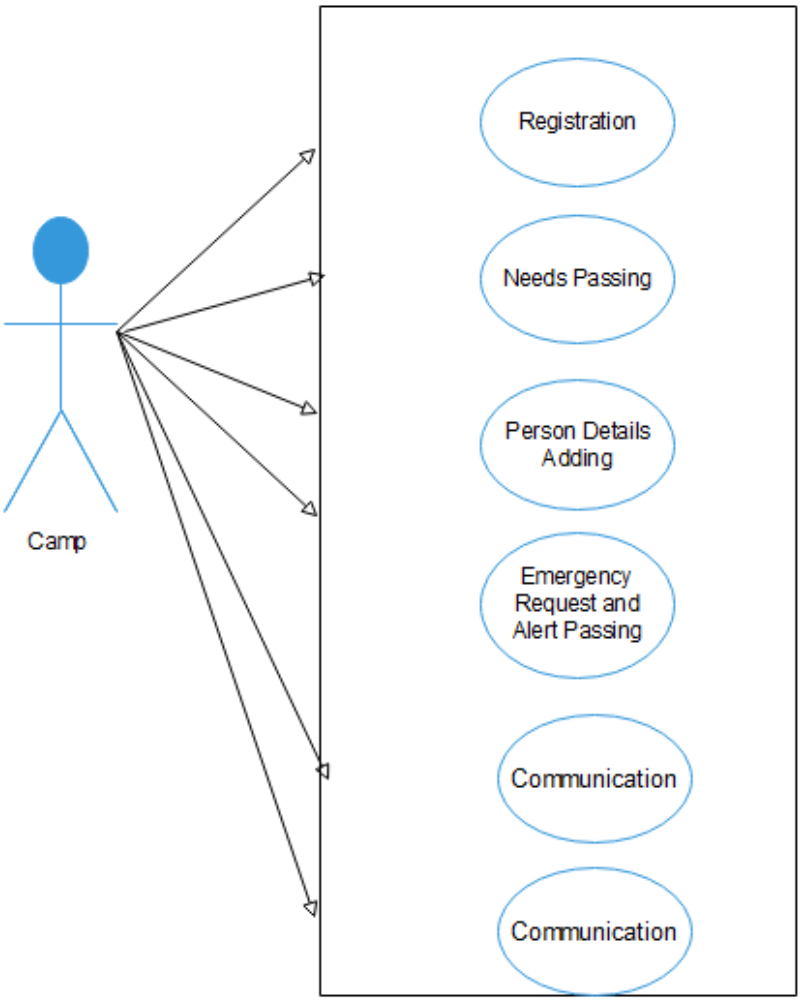
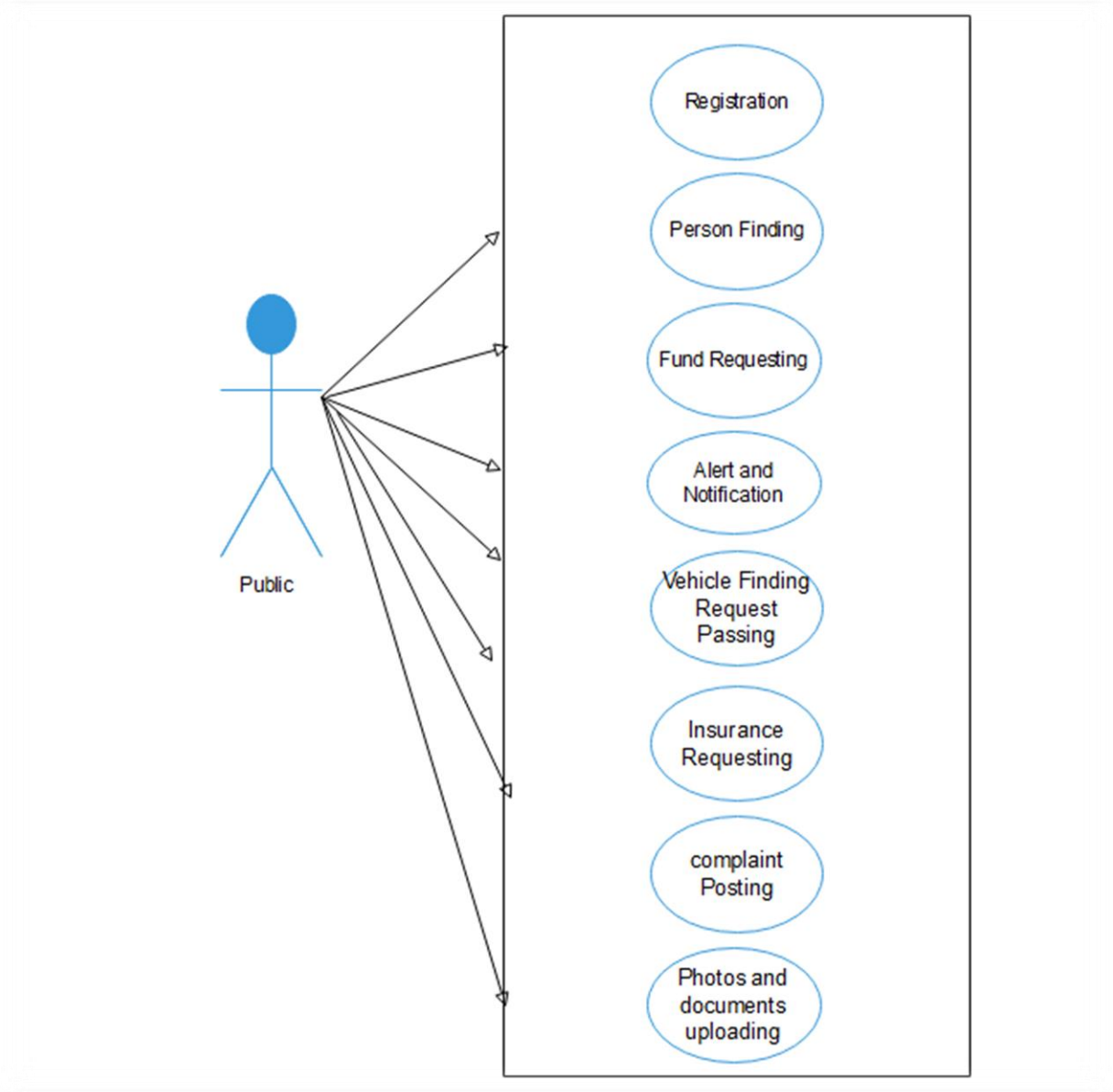
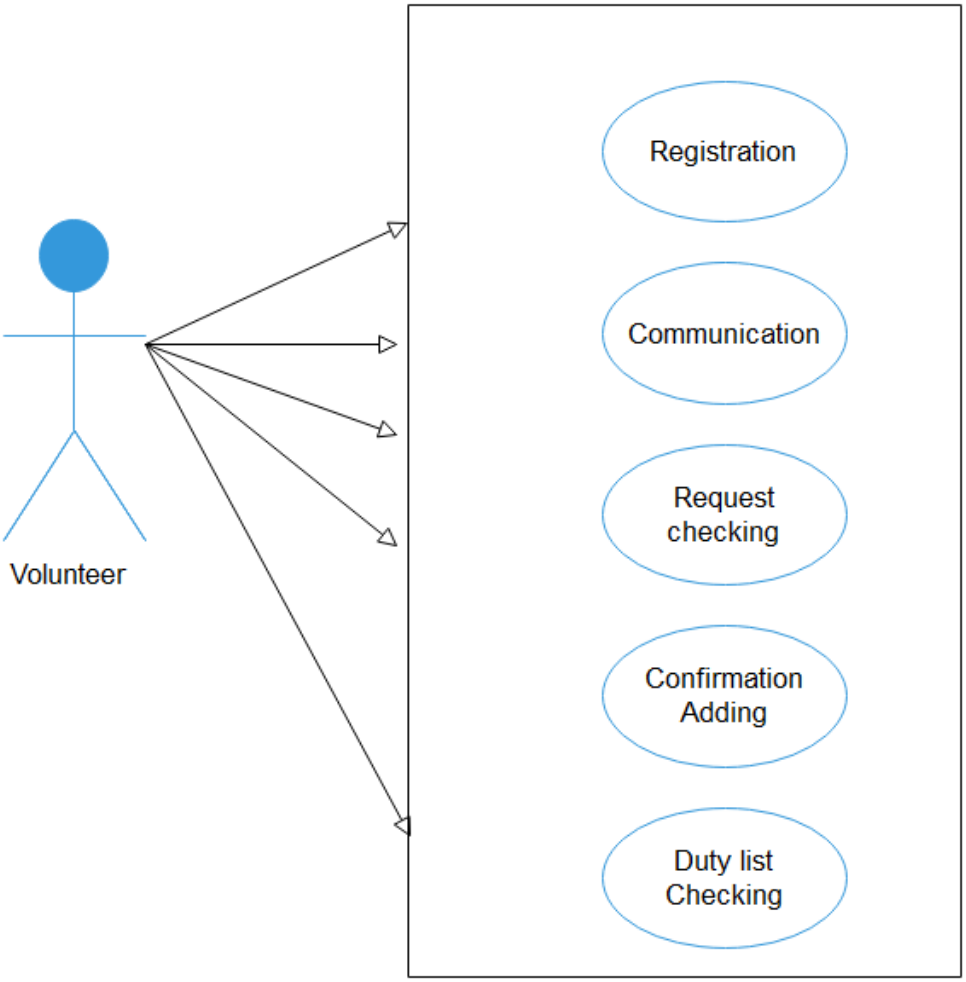
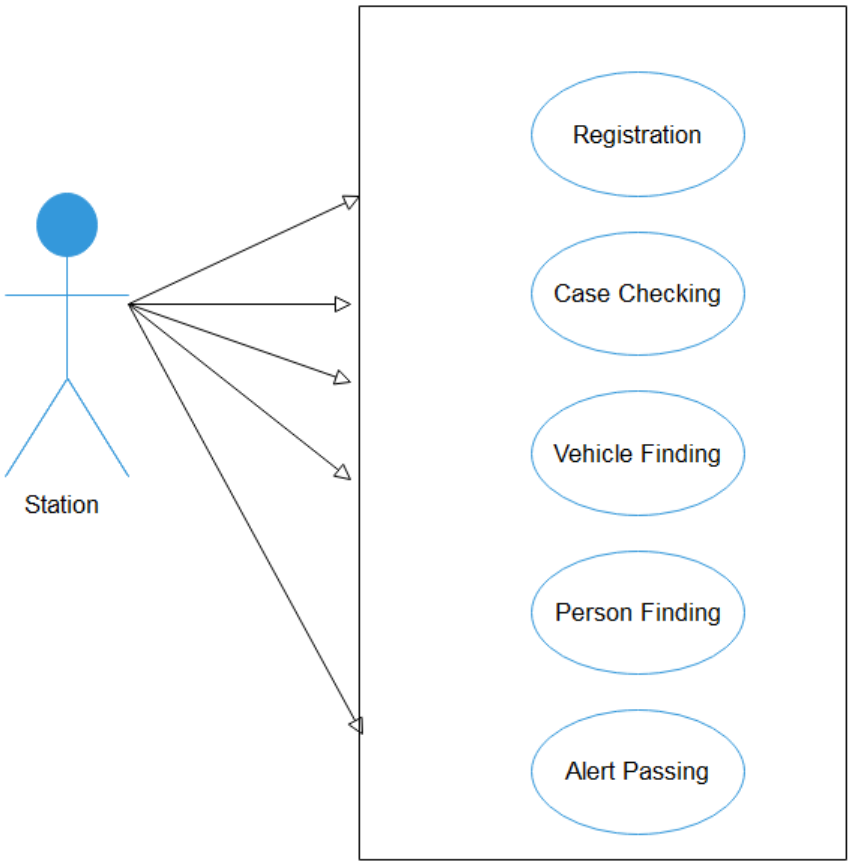

Registration
Needs Passing
Person Details Adding
Emergency Request and Alert Passing
Communication
Communication
Camp

fig 4.6.1.1


Registration
Person Finding
Fund Requesting
Alert and Notification
Vehicle Finding Request Passing
Insurance Requesting
complaint Posting
Photos and documents uploading
Public


Registration
Communication
Request checking
Confirmation Adding
Duty list Checking
Volunteer


Registration
Case Checking
Vehicle Finding
Person Finding
Alert Passing
Station

.

CHAPTER-5

CODING

5.1- LanguageStudy

HTML: stands for Hyper Text Mark-up Language. It is used to design web pages using a mark-up language. HTML is a combination of Hypertext and Mark-up language. Hypertext defines the link between web pages. A mark-up language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most mark-up languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text. HTML is a mark-up language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1995.

Elements and Tags: HTML uses predefined tags and elements which tell the browser how to properly display the content. Remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

HTML page structure: The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

<! DOCTYPE html>: This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

<Html>: This is called the HTML root element. All other elements are contained within it

<Head>: The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage. HTML elements used inside the<html> element include:

- <Style>-This html tag allows us to insert styling into our webpages and make them appealing to look at with the help of CSS.
- <Title>The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.
- <Base>-It specifies the base URL for all relative URL‟s in a document.
- <no script>– Defines a section of HTML that is inserted when the scripting has been turned off in the users browser
- <Script>-This tag is used to add functionality in the website with the help of JavaScript.
- <Meta>-This tag encloses the Meta data of the website that must be loaded every time the website is visited. For e.g.:- the metadata charset allows you to use the standard UTF-8 encoding in your website. T
- His in turn allows the users to view your webpage in the language of their choice. It is a self-closing tag.
- <Link>– The „link‟ tag is used to tie together HTML, CSS, and JavaScript. It is self-closing.

<Body>: The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end. An HTML document can be created using any text editor. Save the text file using .html or .html. Once saved as an HTML document, the file can be opened as a webpage in the browser.

5.2 Cascading Style Sheets:

Fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independently of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colours, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser. While HTML uses tags, CSS uses rule sets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

- CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.
- Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
- Search Engines: CSS is considered a clean coding technique, which means search engineswon‟t have to struggle to “read” its content.
- Superior styles to HTML: CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.CSS Syntax: CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule set consists of a selector and declaration block
- The selector points to the HTML element you want to style
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.

5.3 Bootstrap:

Bootstrap is freely available for every. The main features of bootstrap is, it is very simple and easy to use, hug JavaScript plugins are available, easily design mobile friendly website.

- Easy to Use
- Mobile-Friendly
- Customizable Bootstrap
- Simple Integration
- Pre-styled Components
- Responsive Features
- Browser Compatibility
- Great Grid System

- Extensive list of Components
- Bundled JavaScript plugins
- Good Documentation

Features of bootstrap

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-Friendly: Mobile-first approach: In Bootstrap 3, mobile-first styles are part of the core framework

Simple Integration: Bootstrap can be simply integrated along with distinct other platforms and frameworks, on existing sites and new ones too and one more things you can also utilize particular elements of Bootstrap along with your current CSS.

Pre-styled Components: Bootstrap approaches with pre-styled components for alerts, dropdowns, nav bars, Customizable Bootstrap: The Bootstrap can be customized as per the designs of your project.

Browser compatibility: Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera) Great grid system: Bootstrap is built on responsive 12-column grids, layouts and components. Whether you need a fixed grid or a responsive, it's only a matter of a few changes.

Bundled JavaScript plugins: The components such as drop down menu are made interactive with the numerous JavaScript plugins bundled in the bootstrap package.

Extensive list of components: Whether you need drop down menus, pagination or alert boxes, Bootstrap has got your covered. Some of the components pre styled are; Dropdowns, Button Groups, Navigation Bar, Breadcrumbs, Labels & Badges, Alerts, Progress Bar, And many others.

Base styling for most HTML elements: A website has many different elements such as headings, lists, tables, buttons, forms, etc. The HTML elements for which styles are provided are; Typography Code, Tables, Forms, Buttons, Images, Icons

5.4 JavaScript:

JavaScript is a popular programming language. JavaScript features are flexible. Many open- source libraries are available. GitHub contains a large volume of JavaScript code by developers across the world. JavaScript works well in the front end and back end. JavaScript has a simple syntax. Without any settings, anyone can execute JavaScript programs and make them user-friendly. One individual having basic knowledge of HTML, CSS and coding can work with JavaScript.

Features of JavaScript

Scripting: JavaScript executes the client-side script in the browser. Interpreter The browser interprets JavaScript code. Event Handling Events are actions. JavaScript provides event-handling options.

Case Sensitive: In JavaScript, names, variables, keywords, and functions are case-sensitive.

Control Statements: JavaScript has control statements like if-else-if, switch case, and loop. Users can write complex code using these control statements.

Objects as first-class Citizens:

JavaScript arrays, functions, and symbols are objects which can inherit the Object prototype properties. Objects being first-class citizens mean Objects can do all tasks.

Supports Functional Programming:

JavaScript functions can be an argument to another function, can call by reference, and can assign to a variable.

Dynamic Typing: JavaScript variables can have any value type. The same variable can have a string value, an integer value, or any other.

Client-side Validations: JavaScript client-side validations allow users to submit valid data to the server during a form submission.

Platform Independent: JavaScript will run in the same way in all systems with any operating system.

Async Processing: JavaScript async-await and promise features provide asynchronous nature. As the processes run in parallel, it improves processing time and responsiveness.

Prototype-based: JavaScript follows 'Object. Prototype' functions instead of class inheritance.

5.5 Python:

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation.

We have listed below a few essential features.

Easy to Learn and Use: Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

Expressive Language: Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print ("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

Interpreted Language: Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

Cross-platform Language: Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

Free and Open Source: Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

Object-Oriented Language: Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

Extensible: It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

Large Standard Library: It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such

as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

GUI Programming Support: Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

Integrated: It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C, C++ Java. It makes easy to debug the code.

Embeddable: The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

5.6 Django framework

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement. By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only. History :Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains

its development and release cycle. Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

Features of Django

Rapid Development: Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

Secure: Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable: Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Fully loaded: Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

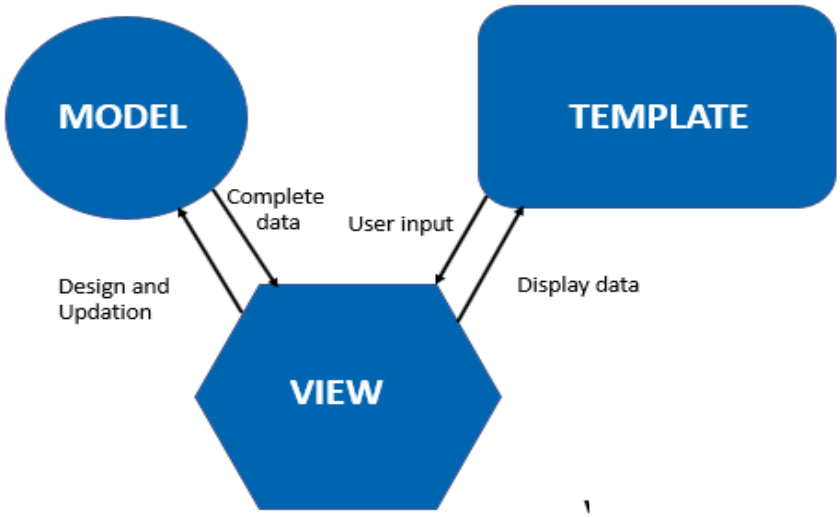
Versatile: Django is versatile in nature which allows it to build applications for differentdifferent domains. Now a day, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

Open Source: Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

Vast and Supported Community: Django is a one of the most popular web framework. It has widely supportive community and channels to share and connect.

Django MVT

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data. The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template. Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the framework itself. There is no separate controller and complete application is based on Model View and Template. That's why it is called MVT application. Here, a user requests for a resource to the Django, Django works as a controller and check to the available resource in URL. If URL maps, a view is called that interact with model and template, it renders a template. Django responds back to the user and sends a template as a response.



Django Model

In Django, a model is a class which is used to contain essential fields and methods. Each model class maps to a single table in the database. Django Model is a subclass of django.db.models.Model and each field of the model class represents a database field (column). Django provides us a database-abstraction API which allows us to create, retrieve, update and delete a record from the mapped table. Model is defined in Models.py file. This file can contain multiple models.

Django Views

A view is a place where we put our business logic of the application. The view is a python function which is used to perform some business logic and return a response to the user. This response can be the HTML contents of a Web page, or a redirect, or a 404 error. All the view function are created inside the views.py file of the Django app

Django Templates

Django provides a convenient way to generate dynamic HTML pages by using its template system. A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. In HTML file, we can't write python code because the code is only interpreted by python interpreter not the browser. We know that HTML is a static mark-up language, while Python is a dynamic programming language. Django template engine is used to separate the design from the python code and allows us to build dynamic web pages.

5.7. SQLite3 DATABASE

Being a very lightweight database management system, SQLite is very popular. No administration was required for operating a program in SQLite. However, it can only handle low to medium traffic HTTP requests. Also, the size of the database is usually restricted to 2GB. Even with these limitations, the SQLite advantages have gained more attention from the users. Some of the SQLite advantages are listed below:

Lightweight database management system: Easy to use as embedded software with various electronic devices. Better Performance:

- Very flexible.
- Fast reading and writing operations.
- Loads only the required data and not the entire file.
- Only overwrite the edited parts of a file and not the entire file
- Facilitates an efficient way for data storage.
- Variable column length of columns thus allows allocating only the spaces that a field needs.

Installation not needed:

- Easy to learn.
- No need to install.
- No Configuration Required.

Reliable:

- Contents are continuously updated.
- Less bug-prone than custom-written I/O code files
- Smaller queries than equivalent procedural codes.

Portable:

- Portable across all 32-bit and 64-bit operating systems and big- and little-endian architectures.
- Facilitates work on multiple databases on the same session at the same time. • Cross-platform DBMS.

- Available on both UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).
- No compatibility issue with any programming languages.
- Facilitates API for a large range of programming languages.
- Facilitates simple and easy-to-use API

Accessible:

- Accessible through a wide variety of third-party tools.
- More likely to be recoverable if data has been lost.
- Data in SQLite lives longer than co

Reduce Cost and Complexity:

- Free to use.
- Open-source.
- No license required to work with SQLite
- Doesn’t require a different server process or system to operate and is thus Server less.
- No need for lengthy and error-prone procedural queries
- Content can be accessed and updated using concise SQL queries.

Open-Source.

- No license required to work with SQLite.
- Doesn’t require a different server process or system to operate and is thus Server less.
- No need for lengthy and error-prone procedural queries.
- Content can be accessed and updated using concise SQL queries

5.8 FUNCTIONAL DESCRIPTION

A functional description outlines the specific functionalities and features that a proposed system or application will offer. In the case of your proposed centralized disaster management platform, a functional description would detail the various capabilities and interactions that the platform will provide to its users. Here's an example of a functional description for your system:

Functional Description of the Centralized Disaster Management Platform:

1. User Registration and Authentication:

- Users can register accounts with roles such as disaster management teams, first responders, administrators, etc.
- Secure user authentication and password management.

2. Dashboard and Overview:

- A user-friendly dashboard providing an overview of ongoing disasters, alerts, and critical information.
- Real-time data visualization for disaster trends and impact.

3. Communication and Collaboration:

- Real-time messaging and chat features for effective communication among stakeholders.
- Discussion forums for sharing insights, experiences, and updates.

4. Guidelines and Resources:

- Comprehensive repository of disaster management guidelines, best practices, and policies.
- Access to step-by-step instructions for different disaster scenarios.

5. Resource Allocation Tools:

- Resource allocation algorithms that analyze disaster impact and optimize resource distribution.
- Tracking and monitoring of resource deployment to prevent duplication.

6 .Real-time Information Updates:

- Notifications and alerts about disaster situations, weather forecasts, and evacuation notices.
- Instant updates based on user preferences and roles.

7. Training and Capacity Building:

- Access to training materials, simulations, and interactive modules for capacity building.
- Progress tracking and assessments to evaluate training effectiveness.

8. Lessons Learned Repository:

- Repository for capturing insights, lessons learned, and post-disaster evaluations.
- Sharing of experiences to support continuous improvement.

9. Standardization and Policy Implementation:
- Implementation of standardized disaster response practices and policies.
- Monitoring compliance with disaster management regulations.

10. Secure Data Storage and Access:
- Data encryption and security measures to safeguard critical information.
- Role-based access controls to ensure authorized user access.

This functional description provides a comprehensive overview of the capabilities that the proposed centralized disaster management platform will offer to its users. Each functionality listed contributes to the system's ability to enhance coordination, communication, and preparedness for disaster response and recovery efforts.

SOURCE CODE

```
from django.shortcuts import render
from django.shortcuts import redirect
from django.http import HttpResponse
from django.template import loader
from . forms import Reg
```

```
from . models import Regera
from . models1 import Station
from . forms1 import Statio
from . models2 import Public
from . forms2 import Pub
from . forms3 import Vol
from . models3 import Volunteer
from . models4 import Campperson
from . forms4 import Campp
from . models5 import Alert
from . forms5 import Alt
from datetime import date
from . models6 import VOlrequest
from . forms6 import Volre
from . models7 import Fund
from . forms7 import  Fundform
from . forms8 import Logim


def Ind(request):
   template=loader.get_template('new.html')
   return HttpResponse(template.render())


def cam(request):
   if request.method=='POST':
      f=Reg(request.POST)
      if f.is_valid():
```

```
            f.save()
    else:
        f=Reg()
    return render(request,'camp.html',{'form':f})

def camtab(request):
    t=Regera.objects.all().values()
    return render(request,'camptable.html',{'k':t})



def login(requet):
    template=loader.get_template('login.html')
    return HttpResponse(template.render())

def admin(request):
    template=loader.get_template('admin.html')
    return HttpResponse(template.render())

def camphome(request):
    template=loader.get_template('camphome.html')
    return HttpResponse(template.render())

def stationhome(request):
    tem=loader.get_template('stationhome.html')
    return HttpResponse(tem.render())
```

```
def publichome(request):
    tm=loader.get_template('publichome.html')
    return HttpResponse(tm.render())


def Volunt(request):
    if request.method=='POST':
        x=Vol(request.POST)
        if x.is_valid():
            x.save()
    else:
        x=Vol()
    return render(request,'volunteer.html',{'v':x})


def voltab(request):
    tm=Volunteer.objects.all().values()
    return render(request,'volunteertable.html',{'t':tm})


def campersondetails(request):
    if request.method=='POST':
        x=Campp(request.POST)
        if x.is_valid():
            x.save()
    else:
        x=Campp()
    return render(request,'camperson.html',{'p':x})
```

```
def camppersondetailstable(request):
    q=Campperson.objects.all().values()
    return render(request,'camppersontable.html',{'cpd':q})

def deletepersondetails(request,id):
    jd=Campperson.objects.get(id=id)
    jd.delete()
    return redirect('campresontab')

def editpersondetails(request,id):
    if request.method=='POST':
        mydata=Campperson.objects.get(id=id)
        w=Campp(request.POST,instance=mydata)
        if w.is_valid():
            w.save()
            return redirect('campresontab')
    else:

def alerttableindex(request):
    u=Alert.objects.all().values()
    return render(request,'indexalerttable.html',{'cpd':u})

def alerttablecamp(request):
    u=Alert.objects.all().values()
    return render(request,'campalerttable.html',{'cpd':u})
```

```
def alerttablestation(request):
    u=Alert.objects.all().values()
    return render(request,'stationalerttable.html',{'cpd':u})


def volunteerrequest(request):
    if request.method=='POST':
        q=Volre(request.POST)
        if q.is_valid():
            fq=q.save(commit=False)
            fq.Currentdate=date.today()
            fq.save()
            return redirect('cm')
    else:
        q=Volre()
    return render(request,'volunteerrequest.html',{'st':q})


def deletevolunteerrequest(request,id):
    jd=VOlrequest.objects.get(id=id)
    jd.delete()
    return redirect('volunteerrequesttable')


def editvolunteerrequest(request,id):
    if request.method=='POST':
        mydata=VOlrequest.objects.get(id=id)
        w=Volre(request.POST,instance=mydata)
        if w.is_valid():
```

```
            w.save()
            return redirect('volunteerrequesttable')
        else:
            mydata=VOlrequest.objects.get(id=id)
            w=Volre(instance=mydata)
        return render(request,'editvolrequest.html',{'edr':w})


    def volunteerrequesttbaleadmin(request):
        t=VOlrequest.objects.all().values()
        return render(request,'adminvolreq.html',{'q':t})     if not user_id:
            return redirect('login')
        try:
            user = Supervise.objects.get(pk=user_id)  # Use pk instead of id
        except Supervise.DoesNotExist:
            return redirect('login')
```

HTML Code

```
    <!DOCTYPE html>
    <html lang="en">
        <head>

            <link                                                rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.
    min.css"                                                integrity="sha384-
    ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT
    2MZw1T" crossorigin="anonymous">
```

```
</head>
<body>
  <div class="container mt-5">
    <center><h2 class="text-center alert alert-danger">VOLUNTEER</h2></center>

  {% block content %}
  <center>
  <form action="" method="POST">
    {% csrf_token %}
    {{v.as_p}}
    <input type="submit" class="btn btn-success" value="SUBMIT">
  </form> <br>
  </center>

  {% endblock content %}
```

CHAPTER – 6

TESTING

System Testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It certifies that the whole set of program hang together. System testing requires a test plan that consists of several keys, activities and steps to run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system.

TESTING OBJECTIVES

- Testing is the process of correcting a program with intend of finding an error. \
- A good test is one that has a high probability of finding a yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

6.1Levels of Testing

6.1.1 Unit Testing

In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as „module" testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In this testing step each module is found to work satisfactory as regard to the expected output from the module. There are some validation checks for verifying the data input given by the user. It is very easy to find error and debug the system.

6.1.2 Integration Testing

Data can be lost across an interface; one module can have an adverse effect on the other sub functions when combined by May not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performance.

6.1.3 Black Box Testing

This testing attempts to find errors in the following areas or categories: Incorrect or missing functions, interface errors, errors in data structures, external database access, performance errors and initialization and termination errors.

6.1.4 Validation Testing

At the culmination of Black Box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and final series of software tests, validation tests begins. Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.

After validation test has been conducted one of the two possible conditions exists.

- The function or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created

6.1.5 Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it doesn't produce the required data in the specific format. The output displayed or generated by the system under consideration is tested by, asking the user about the format displayed. The output format on the screen is

6.1.6 White Box Testing

White box testing is a testing case design method that uses the control structure of the procedural design to derive the test cases. The entire independent path in a module is exercised at least once. All the logical decisions are exercised at least once. Executing all the loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. In our project testing was conducted at every step. Initially each module was tested separately to check whether they gave the desired output for the given input. The forms used to enter data by user were validated and appropriate error messages were displayed if incorrect data was entered. Once the data was entered correctly, the processing was done and testing was done to check whether the correct output was obtained. Once the test cases were conducted successfully for each module, the modules were integrated together as a single system. After integration, the test cases were again applied to check whether the entire system as a whole produced the desired output. At times, the test cases failed and the shortcomings were noted down and appropriate corrections were done. Once the integration

testing was performed correctly, output testing was done and it did not result in any change or correction in the system. Black box testing and white box testing was also conducted successfully. All the loops, decisions, relations were executed at least once before giving it to the users for testing. In black box testing, it was checked whether the data in the proper format was stored in the database or not. Also, it was checked whether the interfaces were working properly or not. On successful completion of these tests, the system was then given to undergo user acceptance testing where the users entered test data to check whether the correct output was obtained. The users were satisfied with the output and thus the testing phase was completed successfully.

6.1.7 Test Data and Results

The primary goal of software implementation is the production of source code that is easy to read and understand. Clarification of source code helps in easier debugging, testing and modification. Source code clarification is enhanced by structural coding techniques, by good coding style, by appropriate supporting documents, by good internal comments and by the features provided in the modern programming language. In our implementation phase, source code contains both global and formal variables. It contains predefined functions as well as the user defined functions. The result of the new system is compared with old system and supposes if the result is wrong the error must be debugged.

After the acceptance of the system by the user, the existing system should be replaced by this system. Any user handles this package very easily. It does not require any intensive training for the user. Procedures and functions involved in this system are very simple that anyone can understand and correspondingly act to the system with no difficulty

CHAPTER – 7

IMPLEMENTATION

7.1 Implementation of Proposed System

The implementation phase involves turning the proposed centralized disaster management platform into a fully functional system. This phase includes designing, developing, testing, and deploying the system. Here's a high-level overview of the implementation process:

1. System Design:
   - Detailed design of the user interface (UI) and user experience (UX).
   - Architectural design of the platform, including database structure and system components.
   - Defining the technical stack and frameworks to be used.

2. Development:
   - Front-end Development: Creating the user interfaces, navigation, and interactive elements using HTML, CSS, and JavaScript.
   - Back-end Development: Developing the server-side logic, business rules, and data processing using appropriate programming languages (e.g., Python, Java, PHP).
   - Database Development: Setting up the database schema and designing data storage and retrieval mechanisms.

3. Communication and Collaboration Features:
   - Implementing real-time messaging and chat features using WebSocket or similar technologies.
   - Setting up discussion forums for user interaction and knowledge sharing.

4. Resource Allocation Tools:
   - Developing resource allocation algorithms based on disaster impact assessment and optimization techniques.
   - Integrating tracking and monitoring functionalities for resource deployment.

5. Real-time Information Updates:
   - Implementing notification and alert systems for sending real-time updates to users based on their preferences.

- Integrating APIs for weather forecasts and disaster alerts.

6. Training and Capacity Building:

- Creating interactive training modules, simulations, and quizzes for capacity building.
- Developing progress tracking mechanisms to monitor user training progress.

7. Security Implementation:

- Implementing user authentication and authorization mechanisms for secure access.
- Setting up data encryption and securing communication channels.

8. Disaster Analytics and Reporting:

- Developing data analytics modules for analyzing disaster trends and response effectiveness.
- Implementing report generation tools for creating visualizations and insights.

9. Testing:

- Conducting unit testing to ensure individual components work as intended.
- Performing integration testing to check the interaction between different system modules.
- Carrying out user acceptance testing (UAT) to ensure the system meets user requirements.

10. Deployment:

- Deploying the platform on appropriate hosting environments (e.g., cloud servers).
- Configuring the server and database for production use.
- Ensuring scalability and performance optimization.

11. Training and User Onboarding:

- Providing training sessions for users to familiarize them with the platform's features.
- Assisting users in setting up profiles and preferences.

12. Launch and Monitoring:

- Launching the platform for users to access.
- Monitoring the system for any issues, performance bottlenecks, or security vulnerabilities.

13. Maintenance and Updates:

- Regularly updating the platform with bug fixes, improvements, and new features.
- Conducting periodic security audits and implementing patches when needed.

Throughout the implementation phase, close collaboration between developers, project managers, disaster management experts, and users is crucial. Feedback loops and iterative development can help ensure that the final system aligns with user needs and expectations.

CHAPTER – 8

SECURITY BACKUP AND RECOVERY MECHANISMS

Online Help

Simply speaking, a backup is a copy of data. This copy includes important parts of database such as the control file and data files. A backup is a safeguard against unexpected data loss and application errors, should we lose the original data, can use the backup to make it available again. After developing our portal, a proper backup copy is stored in a separate system or medium, like CD and drives, from the primary data to protect against the possibility of data loss due to primary hardware or software failure. Recovery from a backup typically involves restoring the data to the original location, or to an alternate location where it can be used in place of the lost or damaged data. A database is a very huge system with lose of data and transaction. The transaction in the database is executed at each seconds of time and is very critical to the database. If there is any failure or crash while executing the transaction, then it expected that no data is necessary to revert the changes of transaction to previously committed point. That means, any transaction in the system cannot be left at the stage of its failure. It should either be completed fully or rolled back to the previous consistent state.

User Manuals

A user manual is a technical document with a quite specific purpose to help non-technicalpeople pinpoint and solve problems without assistance. Since user manual translate what's not comprehend to a language for everyone to understand, they are essential in technical sectors and most commonly associated with software and hardware, IT systems.

In our application user manual includes images and notes relating every functions inorder to help the user to understand features of our system. Admin can add complaints against hotel or their food which can help the hotel managements for trying their best solutions. The hotels can add their images about menus which can be viewed by the user and this also helps them immensely. It also includes screenshots. The main working ofthe system and its data flow is all elaborately described with the help of simplified diagrams and descriptions.

CHAPTER – 9

CONCLUSION

In conclusion, the proposed centralized disaster management platform represents a significant leap forward in enhancing the effectiveness, coordination, and preparedness of disaster response and recovery efforts. By addressing the limitations of the current system, this platform provides a comprehensive solution to streamline communication, optimize resource allocation, and facilitate standardized practices.

The feasibility study demonstrated the practicality of the platform by assessing its technical, economic, operational, and regulatory aspects. It highlighted the potential benefits of reduced response time, improved collaboration, and better resource utilization.

The functional description outlined a wide range of features and capabilities, from real-time communication tools and resource allocation algorithms to training modules and lessons learned repositories. Each of these functions is designed to empower disaster management teams, first responders, and stakeholders with the tools they need to manage disasters efficiently.

During the implementation phase, careful attention was given to system design, development, testing, and deployment. The end result is a user-friendly platform that offers secure access to critical information, ensures effective communication, and supports data-driven decision-making.

As the platform is launched and adopted by disaster management entities, it will pave the way for a more resilient and coordinated response to disasters. Lessons learned from past incidents will be shared, capacity building will be supported, and best practices will be standardized, ultimately contributing to safer and more resilient communities.

In this era of rapid technological advancements, the proposed centralized disaster management platform stands as a testament to the potential of technology to transform how we prepare for and respond to emergencies. With continuous updates, maintenance, and a commitment to improvement, this platform has the capacity to become an invaluable asset in safeguarding lives and minimizing the impact of disasters on communities.

CHAPTER 10

FUTURE ENHANCEMENT

The proposed centralized disaster management platform holds immense potential for future enhancements and continuous improvement. As technology evolves and user needs evolve, there are several avenues for further development and expansion of the platform. Here are some potential areas for future enhancement:

1. Predictive Analytics:
   - Integrate advanced data analytics and machine learning to predict disaster trends and impact.
   - Provide insights into potential disaster scenarios, allowing for proactive planning and resource allocation.

2. Artificial Intelligence (AI) Assistants:
   - Develop AI-powered virtual assistants to provide real-time responses to user queries.
   - Assist in decision-making by analyzing data and offering recommendations based on historical patterns.

3. Internet of Things (IoT) Integration:
   - Incorporate IoT devices for real-time data collection during disasters (e.g., environmental data, sensor readings).
   - Improve situational awareness and enable timely response.

4. Geographic Information Systems (GIS):

- Enhance the platform with GIS capabilities for mapping disaster impact, resource locations, and evacuation routes.
- Provide visual insights for better decision-making.

5. Mobile Application Enhancement:
- Continuously improve the mobile app's features, ensuring seamless access to critical information on the go.
- Implement offline capabilities for accessing essential data during connectivity disruptions.

6. Social Media Integration:
- Integrate social media monitoring and analysis tools to capture real-time updates from affected communities.
- Improve situational awareness and response coordination.

7. Wearable Technology Integration:
- Explore the integration of wearable devices for first responders to provide real-time health data during emergencies.
- Enhance the safety of response teams.

8. Remote Sensing and Satellite Imagery:**
- Utilize satellite imagery and remote sensing data to assess disaster impact from space.
- Provide accurate and up-to-date information to aid decision-making.

9. Virtual Reality (VR) Training Simulations:
- Develop VR training simulations for disaster management teams to simulate real-world scenarios.
- Enhance training effectiveness and preparedness.

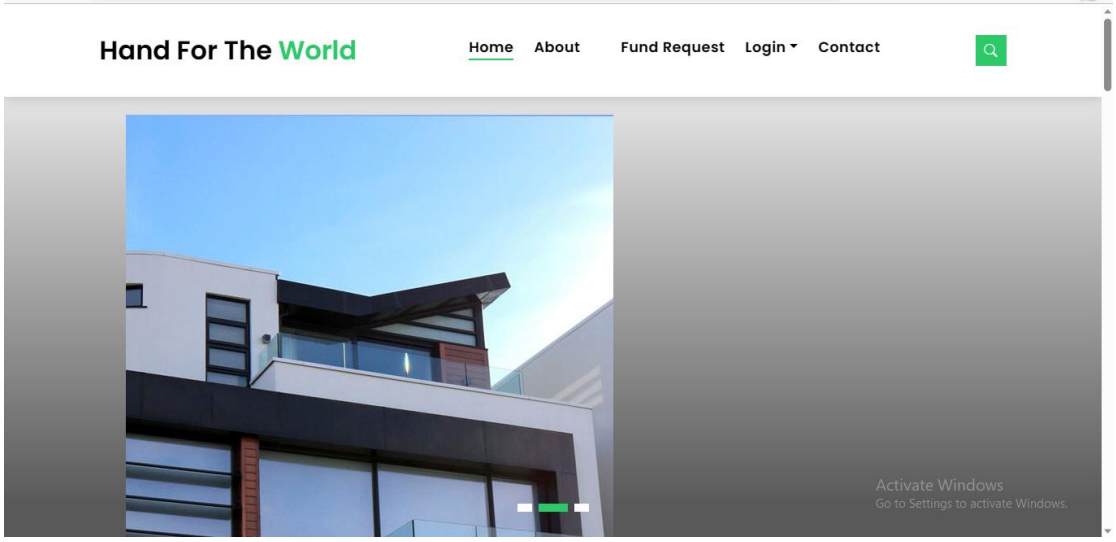
10. Cross-Platform Compatibility:
- Expand compatibility to a wider range of devices and operating systems, ensuring broader accessibility.

The future enhancements of the centralized disaster management platform will be driven by advancements in technology, user feedback, and emerging challenges. By staying agile and responsive to changing needs, the platform can evolve into an indispensable tool for disaster management entities, contributing to safer, more resilient communities worldwide.

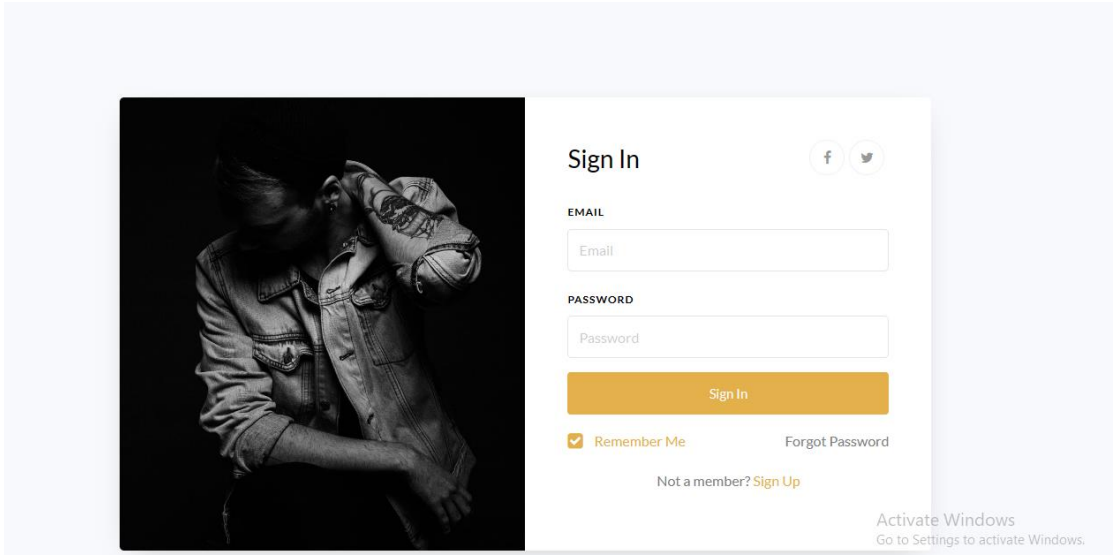
APPENDIX

INPUT AND OUTPUT FORMS

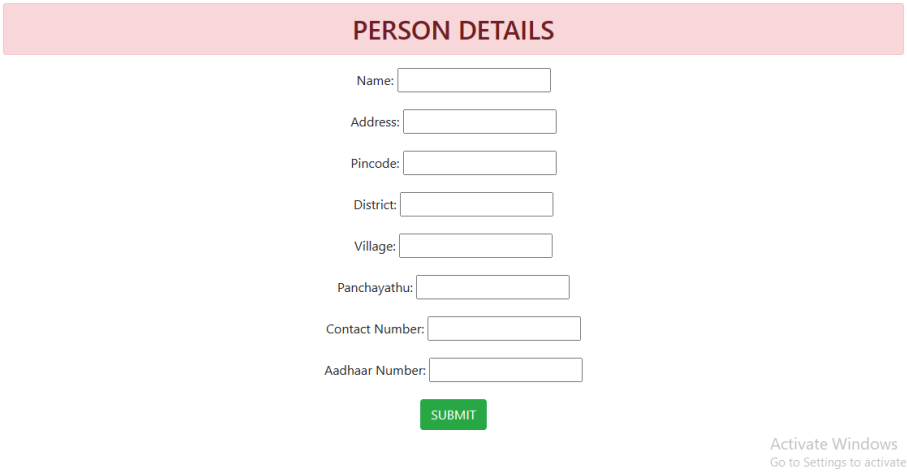
- INDEX PAGE



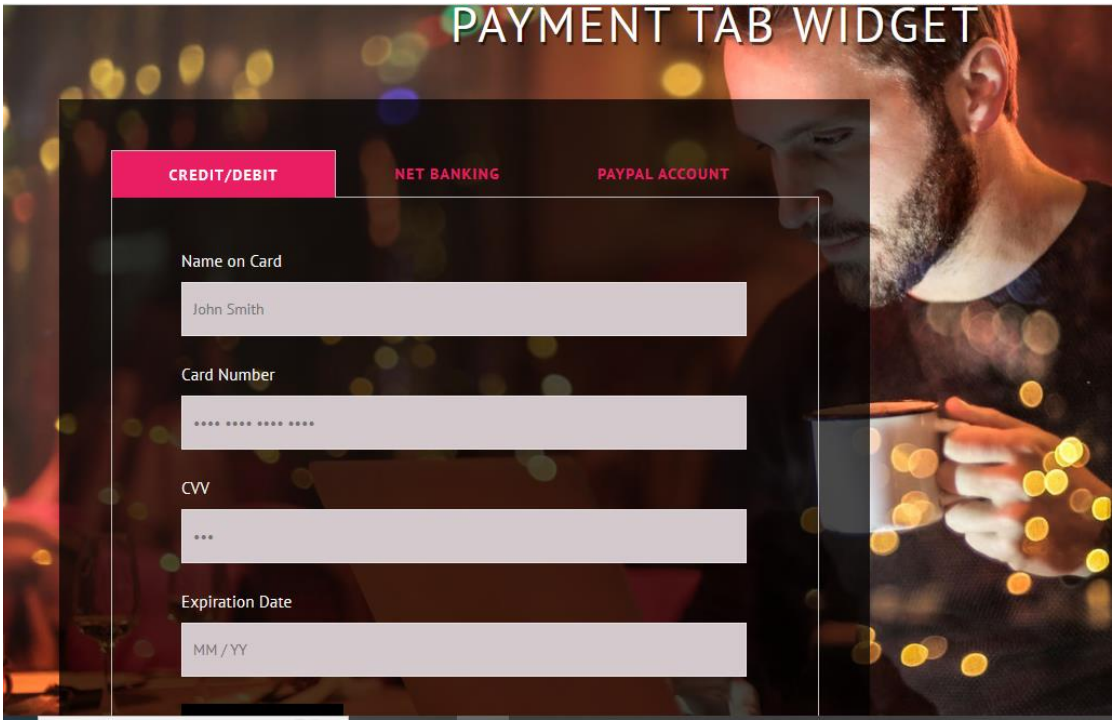
- LOGIN PAGE



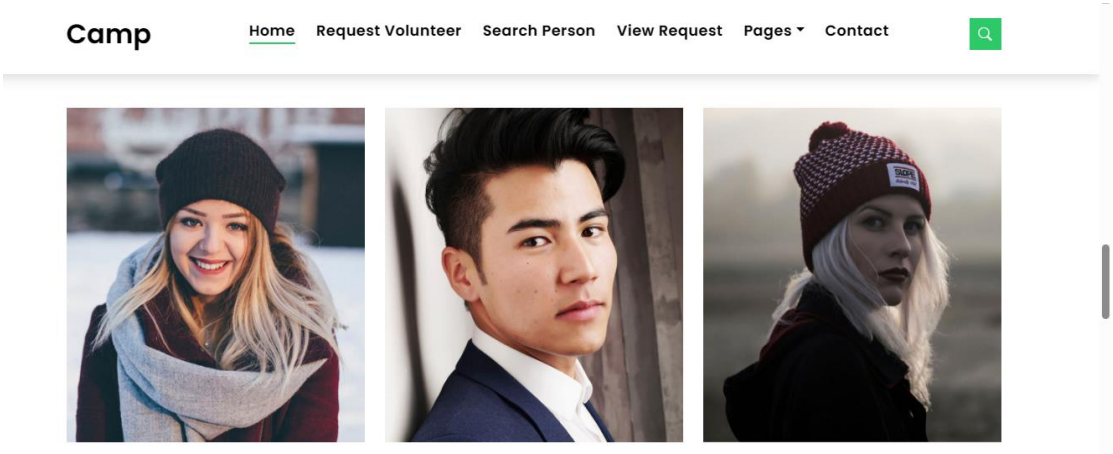
- REDISTRATION



- PAYMENT



- CAMP HOME PAGE



BIBLIOGRAPHY

1. BOOKS

- **Kenneth. S. Rubin**, Essential Scrum: A Practical Guide to the Most Popular Agile Process, First Edition.
- **Allen B. Downey**, Think Python: An Introduction to Software Design, Second Edition
- **OpenCV (2019)** OpenCV—Python Core— [online]

2. WEBSITES

- https://stackoverflow.com/
- https://www.javatpoint.com/python-tutorial
- https://www.tutorialspoint.com/python/index.html
- https://api.jquery.com
- https://www.djangoproject.com
- https://www.tutorialspoint.com/sqllite3/index.html
- https://searcherp.techtarget.com/definition/warehouse-management-system-WMS