VACCINATOR

1. INTRODUCTION

1.1 ABOUT THE PROJECT

Vaccines reduce risks of getting a disease by working with your body's natural defenses to build protection. When you get a vaccine, your immune system responds. We now have vaccines to prevent more than 20 life-threatening diseases, helping people of all ages live longer, healthier lives.

Our project “**Vaccinator”** is a Vaccination Management System for Kids, which is an advanced software solution designed to simplify and automate the management of vaccination processes for children. The system consists of four primary modules: Admin, Vaccination Provider, Vaccination Receiver, and Health Inspector. The Admin module acts as the central control hub, responsible for user role management and access privileges. The Vaccination Provider, who is a nurse or a designated healthcare professional, interacts with the system to input user details and provide registration numbers along with the Date of Birth (DOB) of the recipients.

The main aim of this initiative is to streamline the management vaccination processes for children and improve the immunity for children. We also aim to encourage other to get vaccinated regularly to keep our nation disease free.

In summary, the Vaccinator offers a comprehensive and efficient solution for managing the vaccination process for children. By incorporating the Admin, Vaccination Provider, Vaccination Receiver, and Health Inspector modules, the system facilitates seamless coordination among various stakeholders, enhances data accuracy, and expedites the immunization process for children.

2. LITERATURE SURVEY

INTRODUCTION

System analysis work with users to identify goals and build systems to achieve them. System analysis is the important phase of any system development process. The system is studied to the minutest details and analyzed. The system analyst plays the role of an interrogator and dwells deeps into the working of the present system. In analysis, a detailed study of these operations performed by a system and their relationship within and outside of the system is done. A key question considered here is what must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not the candidate system should be considered. The system is viewed as a whole and the input to the system are identified. The output form the system are traced through the various data are collected on available files, decision point, and transaction handled by present system. Based on analysis, a cost/benefit analysis is considered. The cost for the system is calculated on the basis of analysis, a cost/benefit solution are given as a proposal. The proposal is presented to the user request and suitable change is made. This is a loop that ends as soon as the user is satisfied with the proposal. After all, this proposal become a formal agreement that paves way for the design implementation.

.

2.1 Purpose of the Project

It is an open website used for vaccination for children. Any general public can use this system to get a vaccination appointment for their children. The Medical Professionals authorized by the Admin conduct the vaccination under strict supervision.

2.2 Existing System

The existing system has many drawbacks that are rectified when the system is computerized. In the existing system data are collected, stored, processed, tabulated and entered in the respective tables manually. Since the data are received from different registers and after each transfusion the records re to be updated, it demands much of hard work. Complexity in managing the data related to the blood donation

Limitations of Existing System

- Lack of integrity: - donation records not integrated.
- Poor reliability: - since the data is handled manually, it is prone to data entry error and manipulation errors making the data less reliable.
- Lack of security: - the data stored in registers can be corrupted by unauthorized users or authorized users.
- Time consuming: - since the system is maintained manually, it is a very tedious job to generate reports by checking all the needed data.
- Corruption of data: -The organ data and results are stored as paper files, so there is a chance for corruption of data.

2.3 Proposed System

The proposed system "Vaccinator", all the activities are done through automation. The system being an online system, it will be very much helpful for the peoples as well as the administrators to manage the donation system.

Through this site, it incorporating modules for Admin, parent, heaalth inspector, and Medical Professional, the system facilitates seamless coordination among stakeholders , enhances immunization process, and improves child care outcomes..

- Reduce complexity in managing the data related to the vaccination.
- Current system provides different access levels for security.
- Rich user interface is provided in order to interact with application.
- Reports are generated dynamically on a periodic basis.
- Efficiency in querying details.

User Queries and Answers are maintained

Advantage Proposed System

- Simple and easy to use without training,
- Interactive and user friendly,
- Save Time,
- Know which vaccines are available without any trouble
- All information's recorded are safe and provides quick and easy access to details
- Low maintenance cost
- Faster and better management of the system.
- Integrated system.
- Accurate, reliable

2.4 FEASIBILITY STUDY

The main objective of feasibility study is to test the technical, social and economic feasibility of developing a system. This is done before developing a system. This is done by investigating the existing system in the area under investigation and generating ideas about the new system.

Feasibility study is a test of System proposed 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.

2.4.1 Economic Feasibility

Economical analysis is most frequently used method for evaluating the effectiveness of a candidate system. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise further justifications or alternatives in the proposed system will have to be made if it is to have a chance of being approved. Economic analysis is the most frequently used method for evaluating the effectiveness of the proposed system. Considering the benefits and savings that are expected from the proposed system, the administrators decided to design and implement a new system. By using the latest HTML technology the system can be developed at low cost and the cost of hardware is also less.

2.4.2 Technical feasibility

In Technical feasibility, the management determines whether the current level of technology can support the proposed system. Considering all the advantages of the proposed system time consumption for accessing a web site and retrieval of information has been reduced, we recommended to use the proposed system. Technologies used in the system are PHP and My SQL.

2.4.3 Behavioral feasibility

In Behavioral feasibility, the management considers that the proposed system will fulfill the requirements of the users, ie whether the proposed system covers all the jobs that were expected by the end user and whether it has considerable improvements. Understanding the advantages and efficiency of the proposed system, we decided to develop a new system.

Since the new proposed system is nothing to do with the current users, worker resistance to the system is very much less. The proposed system helps to avoid the delay in processing the data thereby reducing the amount of time consumed.

2.4.4 Operational Feasibility

Proposed project would be beneficial only if they can be turned into information system that will meet the organization operating requirements. One of the main problems faced during the development of a new system is getting acceptance from user. Proposed project is beneficial because they can be turned in to information systems that will meet the user requirements. The test of operational feasibility asks if the system will work when developed and installed.

3. SYSTEM ANALYSIS

3.1 HARDWARE SPECIFICATION

Development environment need Computer system with

Processor	:	Dual-core or Higher
Memory	:	4 GB RAM
Hard Disk	:	80 GB or higher
Keyboard	:	Standard 102/105 keys
Mouse	:	Optical Mouse

3.2 SOFTWARE SPECIFICATION

OS	:	WINDOWS 10 & ABOVE
Web server	:	APACHE
Database	:	MY SQL
Front end	:	PHP
Browser	:	Google Chrome, Mozilla Firefox or any others

3.3 REQUIREMENT SPECIFICATION

This is the requirement document for the “E-Blood Donor” website to be developed based on the various input and output requirement of the system

Purpose

It is an open website used for vaccination for kids. Any general public can use this system for getting a vaccination for their children.

Scope

- Providing accessibility to all users who have a valid user ID and password.
- Generating monthly and daily updates up to date.
- User can post query. They can see answers only after they have logged in.
- Communication is provided for the user through chat.

3.3.1 Functional Requirement

Module Description

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of a system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity, that is, software is divided into separately named and addressable components called modules, which are integrated to satisfy problem requirements. Modularity is the single attribute of software that allows a program to be intellectually manageable. Monolithic software (i.e., a large program comprised of a single module) cannot be easily grasped by reader.

Module Design

1. ADMIN

In the Admin module, Admin can allow authorized personnel to oversee user accounts, ensuring proper access controls and data confidentiality. It includes the following:

- Add Nurse
- Add Health Inspector
- Nurse management
- User management

2. NURSE

A Nurse utilizes the system to record relevant user information during the vaccination process. It includes the following:

- Give Vaccine
- Notifications
- Document Submission
- Chat

3. USER

The Vaccination Receiver, typically the parent or guardian, can access the system to verify registration and appointment details for their child. It includes the following:

- Vaccination
- Registration
- Notifications
- Chat

4. HEALTH INSPECTOR

Health Inspectors, authorized by the Admin, have access to the system to oversee and ensure compliance with health and safety standards. It includes the following:

- Vaccination List
- Contact Nurse
- Contact User

3.3.2 Non-Functional Requirements

- Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore it is required to take the database backup.

- Security Requirements

Security is an important consideration. The first step in securing the application is deciding where the security is needed and what it needs to protect. Authentication is the process of determining a user`s identity and forcing the users to prove they are who they claim to be usually this involves entering the credentials (user name and password) in some sort of login page or windows. Once the user is authenticated, authorization is the process of determine where the user has sufficient permission to perform a given action, such as viewing a page or retrieving information from the database.

- Hardware Constraints

The system requires a database in order to store persistent data. The database should have backup capabilities.

3.3.3 Software Technology Review

HYPER TEXT TRANSFER PROTOCOL (HTTP)

HTTP is the protocol “spoken” by Web servers. Client programs that can speak HTTP, known as browsers, are used by the people on the Internet to connect to HTTP servers. These servers provide access to distributed hyper linked documents, applications and databases. HTTP is a stateless, object oriented application level protocol that has been in the existence since the early days of the WWW . NSCA HTTP is a HTTP/1.0 compliant Web Server and is credited with being one of the first HTTP servers available. It supports multiple schemes of authentication.

PHP

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor. PHP code may be embedded into HTML or HTML5 markup, orit can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHPcode, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implementstandalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web serverson almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

ABOUT MySQL

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is officially pronounced /maɪˌɛskjuːˈɛl/ ("My S-Q-L"),[2] but is often also pronounced /maɪˈsiːkwəl/ ("My Sequel"). It is named for original developer Michael Widenius's daughter My.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Members of the MySQL community have created several forks such as Drizzle, Our Delta, Percona Server, and MariaDB. All of these forks were in progress before the Oracle acquisition (Drizzle was announced 8 months before the Sun acquisition).

MySQL is primarily an RDBMS and therefore ships with no GUI tools to administer MySQL databases or manage data contained within. Users may use the included command line tools,[16] or download MySQL frontends from various parties that have developed desktop software and web applications to manage MySQL databases, build database structure, and work with data records. One of the popular front end is PhpMyAdmin

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations arerequired. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well.

XAMPP Server

XAMPP is a free and open source cross-platform web server package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

XAMPP is an easy to install Apache distribution containing MySQL, .net and Perl. XAMPP is really very easy to install and to use - just download, extract and start.

It is used as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet.

Many important security features are disabled by default. XAMPP is sometimes used to actually serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQL among others.

Once XAMPP is installed you can treat your localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a contain management system (CMS) like Joomla. You can also connect to localhost via FTP with your HTML editor.

4. SYSTEM DESIGN

Introduction

The system design translates the system requirements into ways of operationalizing them. The design is a solution, a “how to” approach, compared to analysis, a “what is” orientation. The emphasis is on translating performance specifications into design specifications. The design phase is a translation from a user – oriented document (system proposal) to a document oriented to the programmers or database personnel. The system design covers the following.

- Reviews the current physical system
- Prepares output specifications
- Prepares input specifications
- Prepares edit, security and control specifications
- Specifies the implementation plan
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.

Characteristics of Design

- A design should exhibit a Hierarchical organization that makes intelligent use of control among components of the software.
- A design should be modular this is , the software should be logical.
- A design should contain distinct and separable representation of the data and procedure.
- A design should lead to module.

 A design should lead to interface that reduce the complexity of the connections between modules and with the external environment.

Design Objective

The design of the system is correct if the system built precisely according to the design, satisfies the requirements of that system. Clearly, the design phase is to produce correct designs. Some properties for the software system design are:

- Completeness
- Consistency
- Efficiency
- Traceability

4.1 INPUT DESIGN

It is the process of converting user organization inputs to a computer based format since inaccurate input data, cause errors in data processing. It is controlled by input design. That is, the input design must be made to make data entry easy, logical and error free as possible.

The major approaches for entering data into computer are menus, formatted forms and prompts. A menu is a selection list: a form is brief printed form or template that user to enter data in appropriate locations where as in prompt. The system displays an enquiry at a time asking the user for response.

Input design is a basic theory to be considered during system design. In input design the screens are designed according to the user needs. In our system the inputs are developed according to the user. The main input of the system will be the administrator authentication, details of the application program to be executed at the client side etc.

4.1.1 Input Objective

Controlling Amount of Input: Whenever user input is required, giving possible input values as default in the area reduces the amount of user keystrokes. Thus the user can pass on the next data without much typing. This makes the data entry much fast and error free. When the user to give the input in the same format.

- Avoiding Delay:

A processing delay resulting from data operations or data entry operations is called a bottleneck. Such bottlenecks are made obsolete in this project by breaking up the amount of data to be entered in each form into different smaller and simpler forms.

- Avoiding errors in data:

The rate at which errors occur depends on the quantity of data. As told in the above objective these errors are reduced by making the number of data to be entered in each form is reduced.

- Avoid extra steps:

To fulfill any operation the user has no need to do complex steps, instead operation can be done simple easy to do steps.

- Keeping the process simple:

This implies that that system has all measure to keep the errors out even if the user is giving wrong data. It handles the situation with grace and does not create much type about the situation to the user.

4.2 OUTPUT DESIGN

Output design is the most important to the user. Efficient, intelligible output design improves the system relationship with the user and helps the decision making. Computer outputs are the most important and direct source to the user. An efficient output system improves the interaction of the system with the user and it provides his/her required information. The output can be displayed on the screen or copied. In our system, hard copies are preferred because a document for further references. Careful considerations have been given while developing the output reports as if helps in decision making. Other than the remote system desktops visuals, received messages from other systems etc. are also outputted.

4.2.1 Output Objectives

The output form information system should accomplish the following objectives:

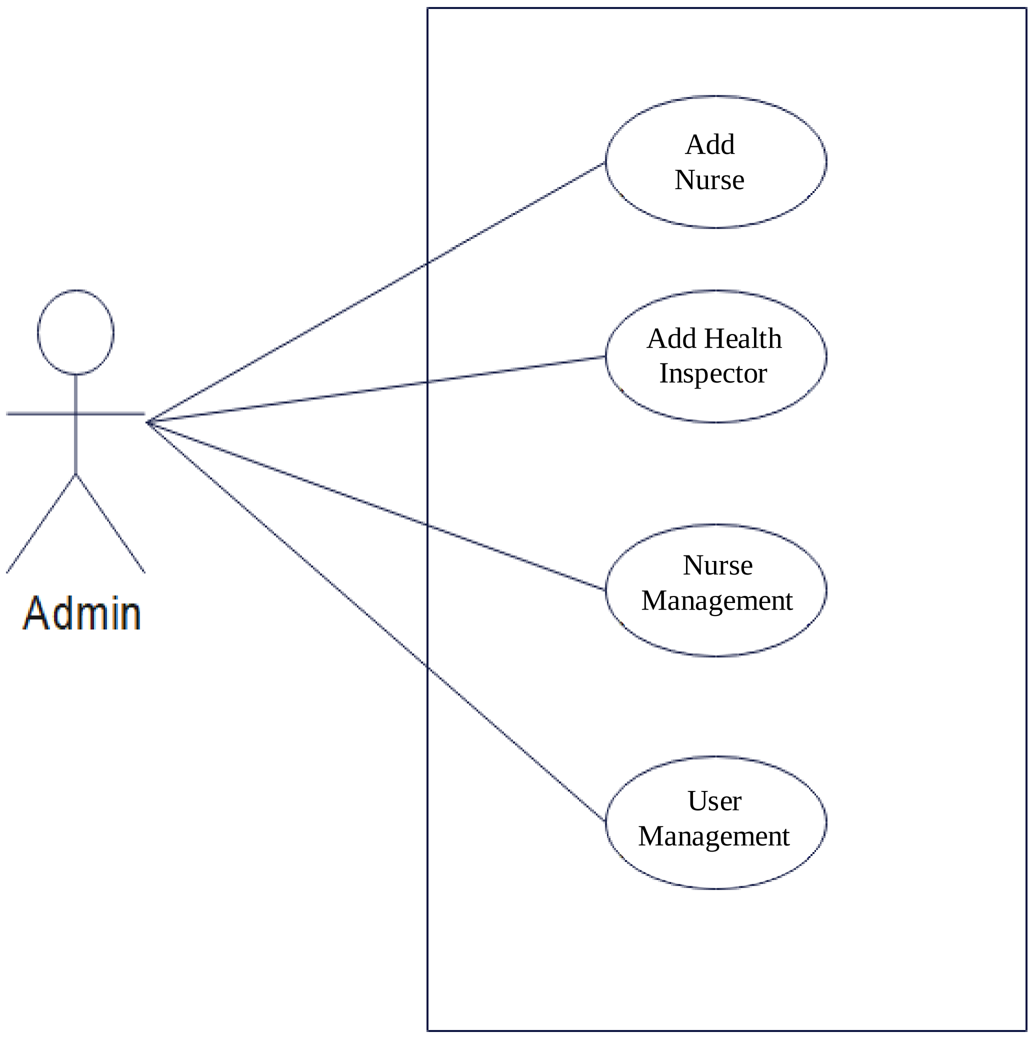
- Convey information about past activities, current status or projection of the future
- Signal important events, opportunities, problems or warning
- Trigger an action
- Confirm an action
- Output design of the project is made with objectives in mind
- External outputs, whose destination is within the organizations and is the main image of the organization.
- Internal outputs, whose destination is within the organization and which require careful design because it is user's main interface with the computer.
- Operational outs, whose use is purely within the computer departments.
- Interactive outputs, which the user in communicating directly with the computer.

4.3 USECASE DIAGRAM

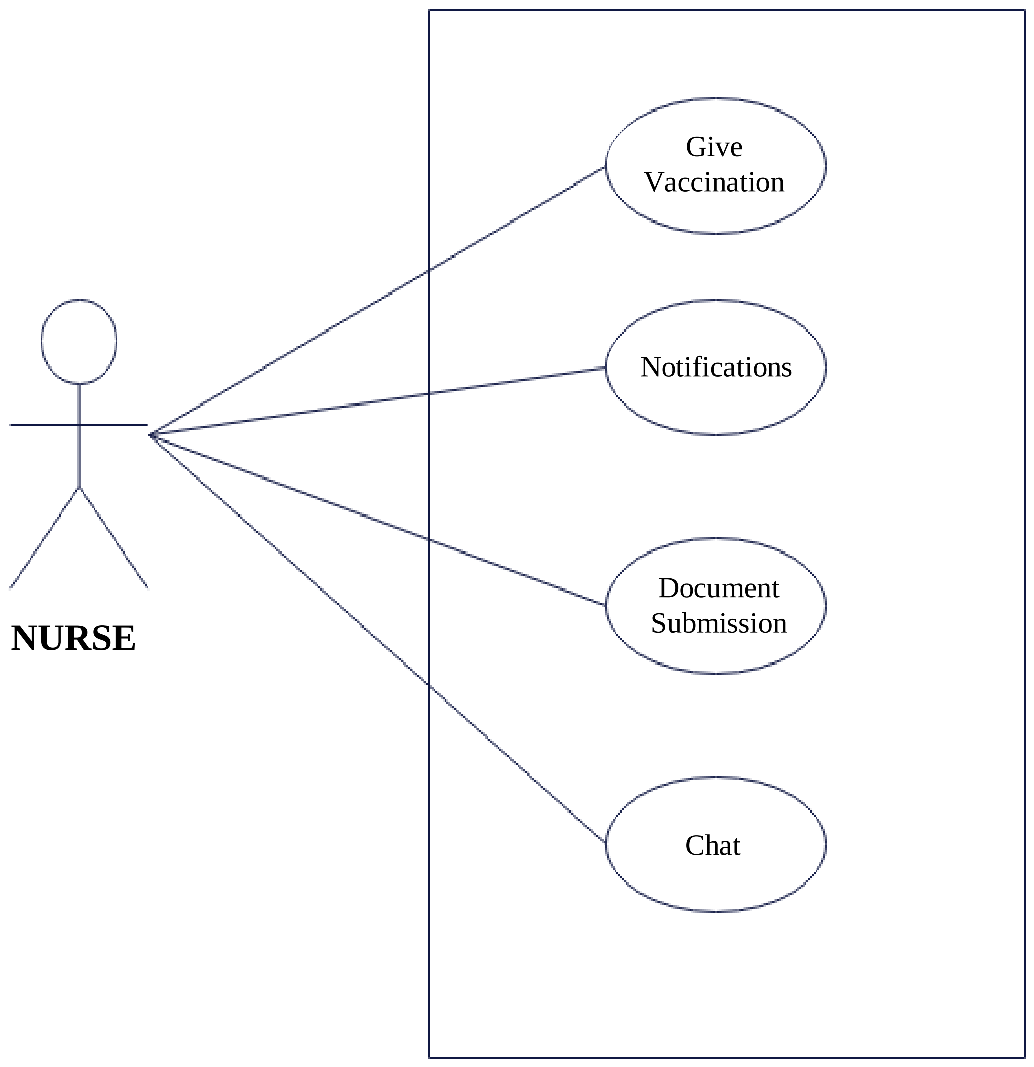
To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating. Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

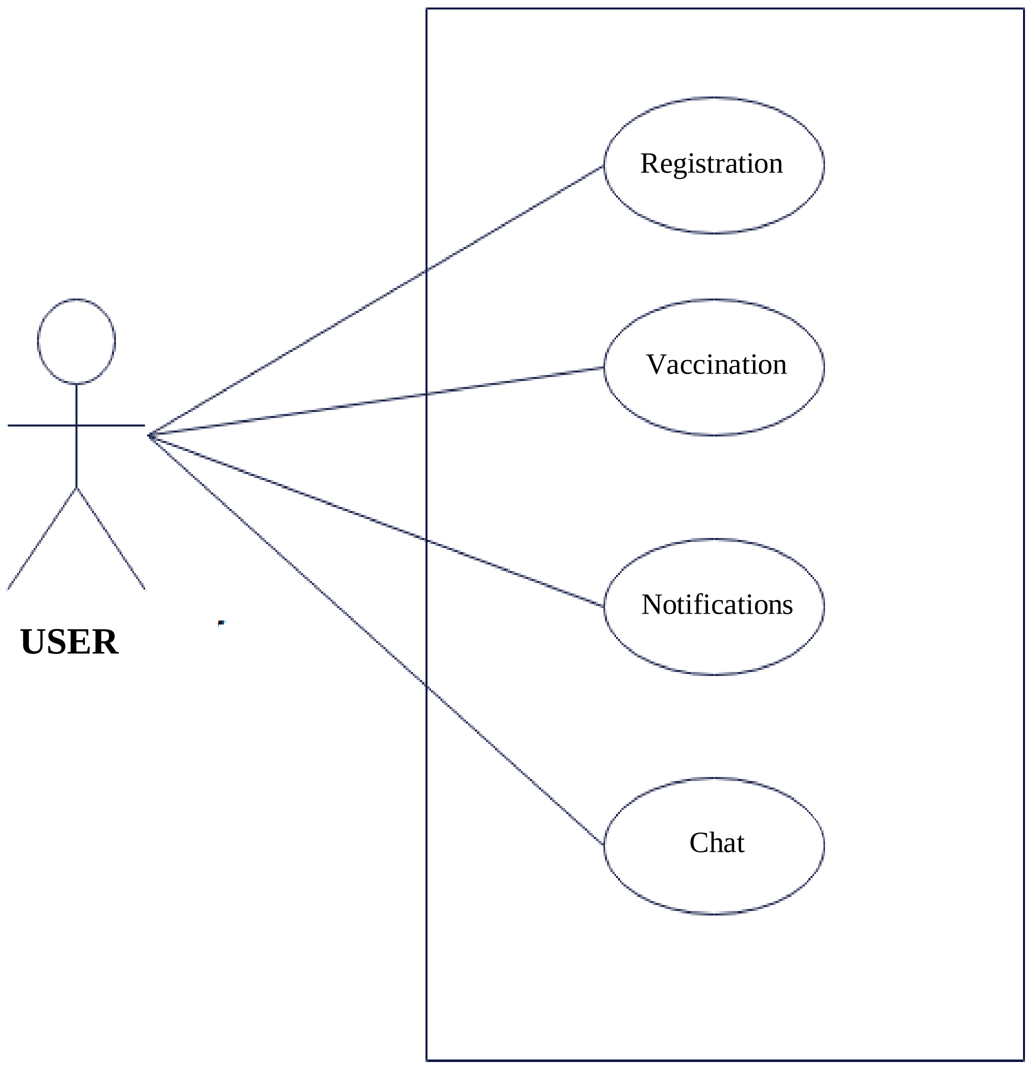
ADMIN


Admin
Add Nurse
Add Health Inspector
Nurse Management
User Management

NURSE

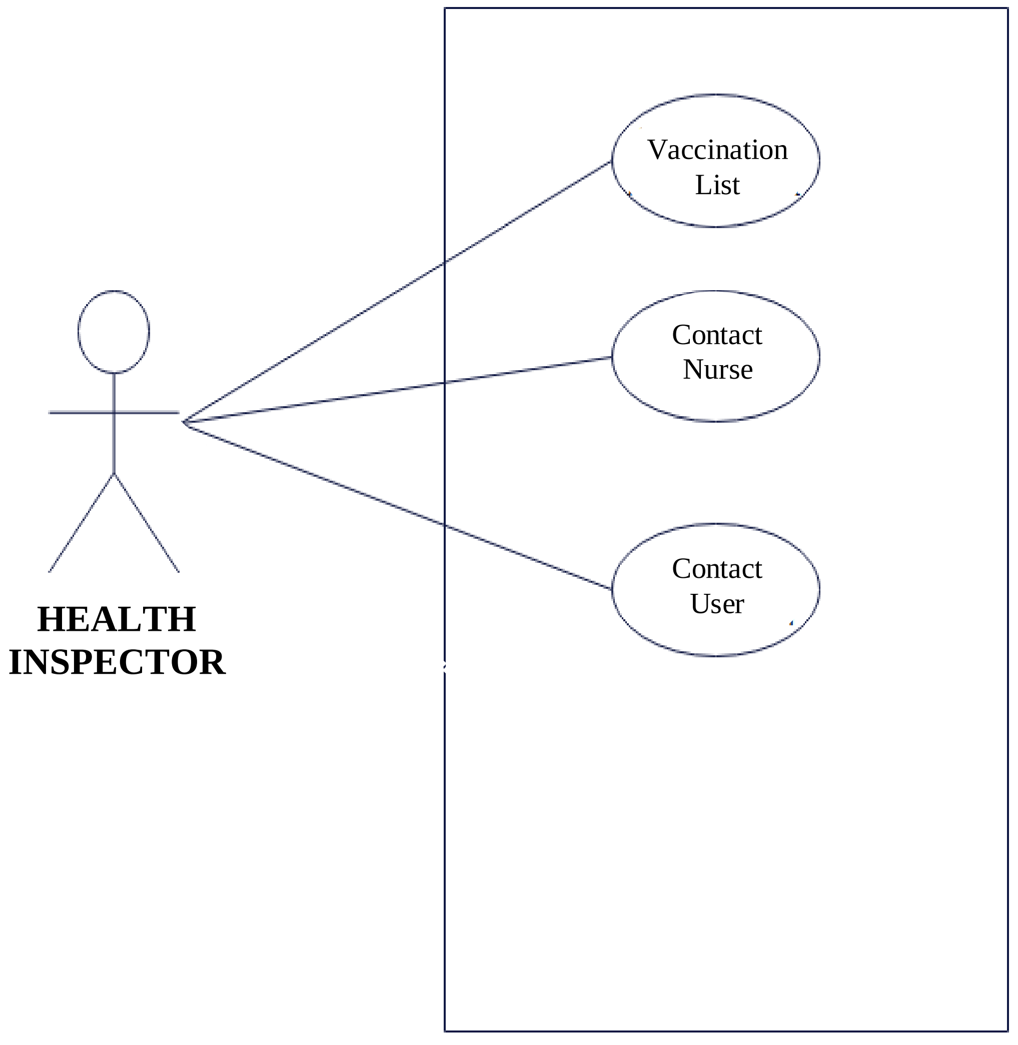

Give Vaccination
Notifications
Document Submission
Chat
NURSE

USER



Vaccinator

HEALTH INSPECTOR



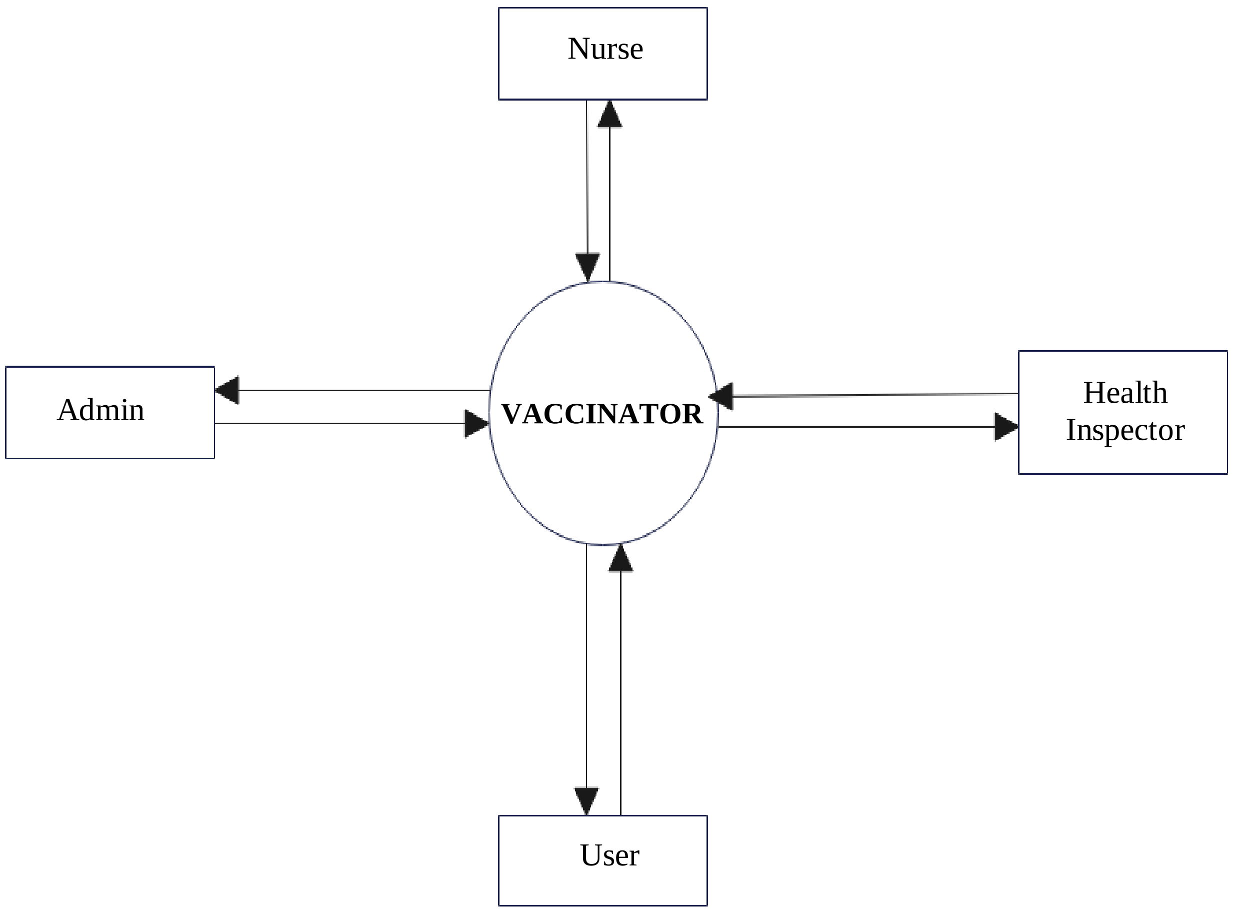
4.4 DATA FLOW DIAGRAM

Data flow diagrams are most commonly used way of documenting the processing of data in software system. As the name suggest they are the pictorial way of showing flow of data into around and out of the system.

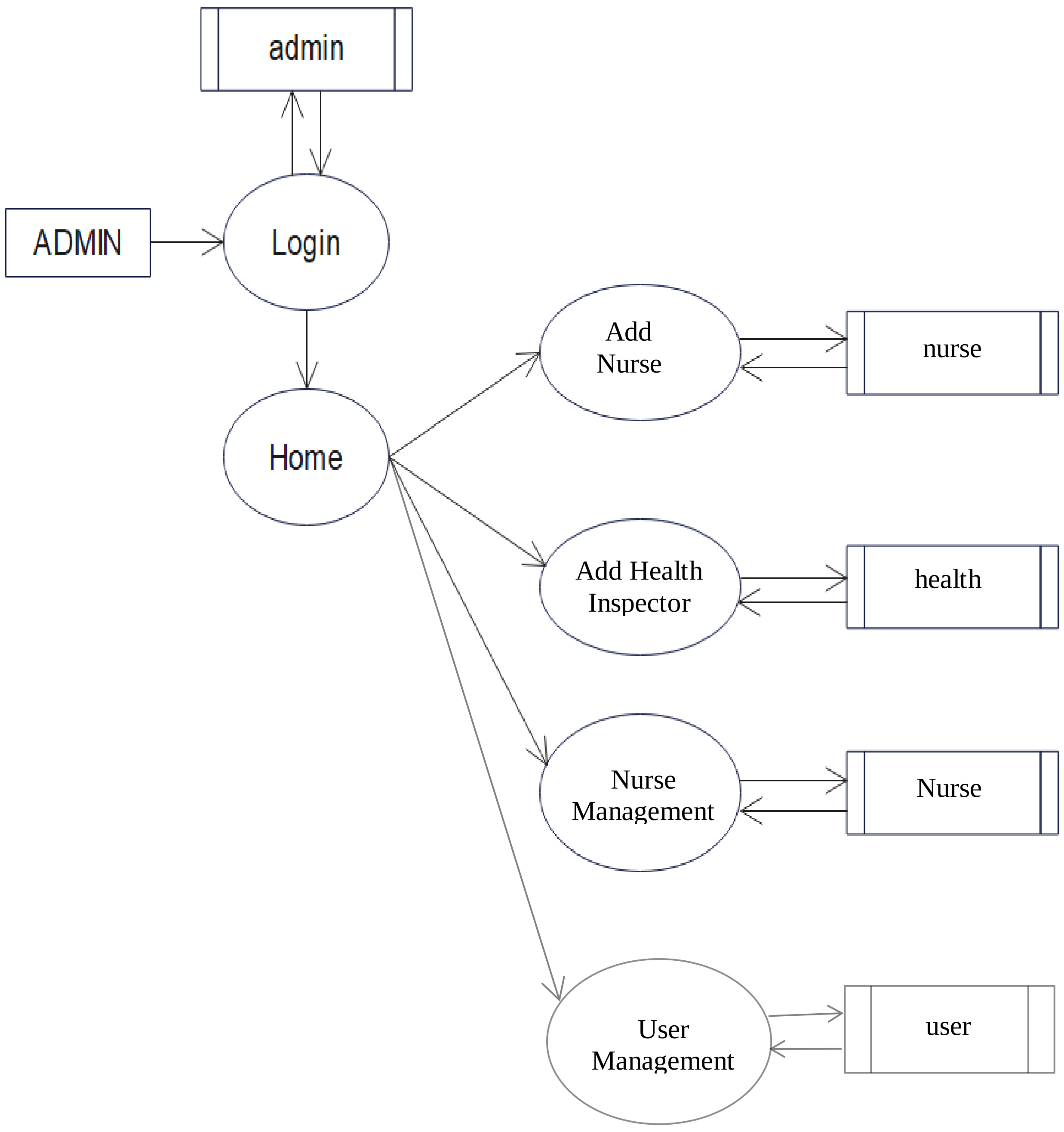
Symbols used

○	Used to represent a process in the system
▯▭▯	Used to represent a data store or table
▭	Represents an External entity or an input or output file
⟶	Represent the flow of data and its direction

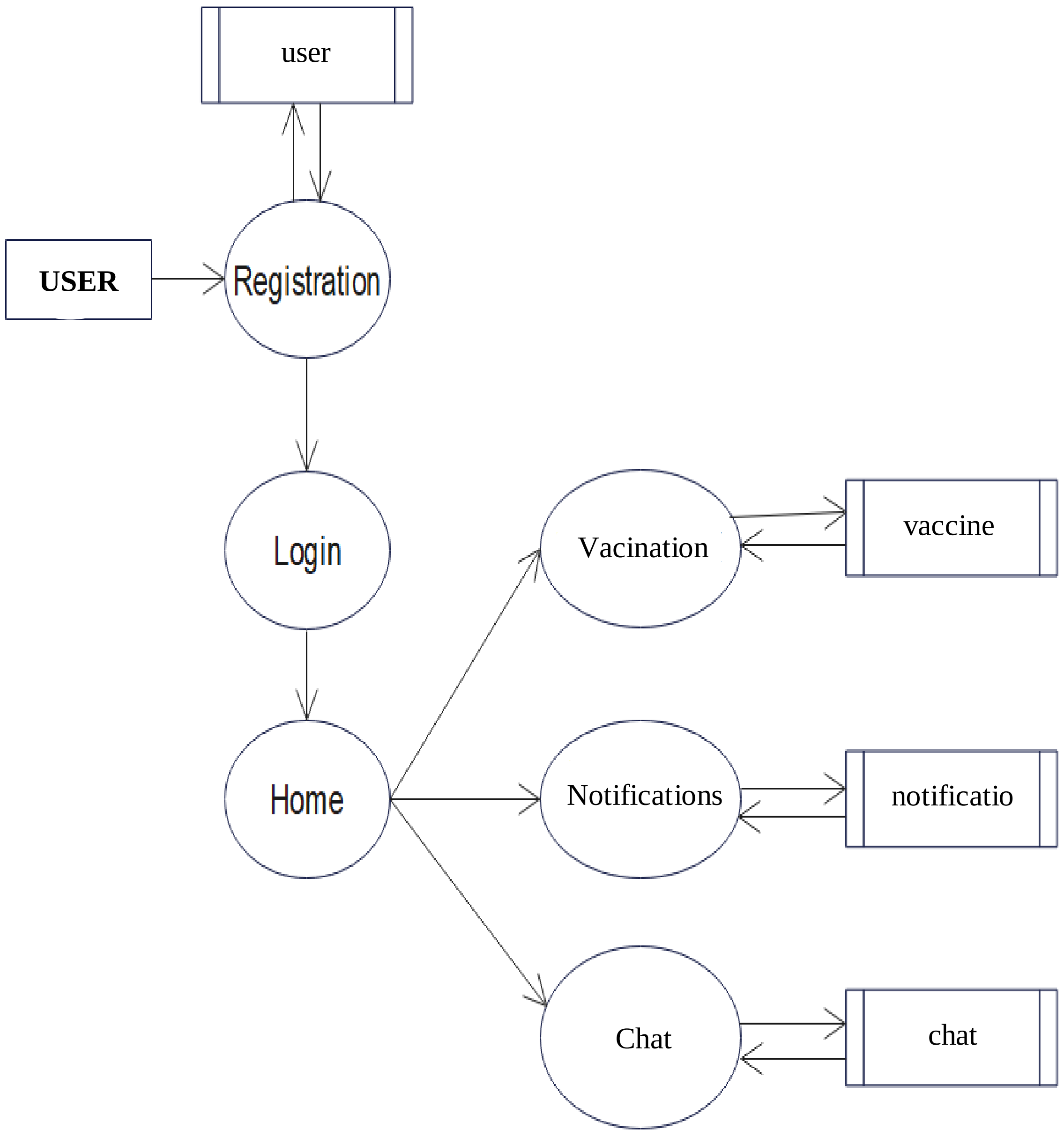
LEVEL 0 – CONTEXT DIAGRAM [VACCINATOR]



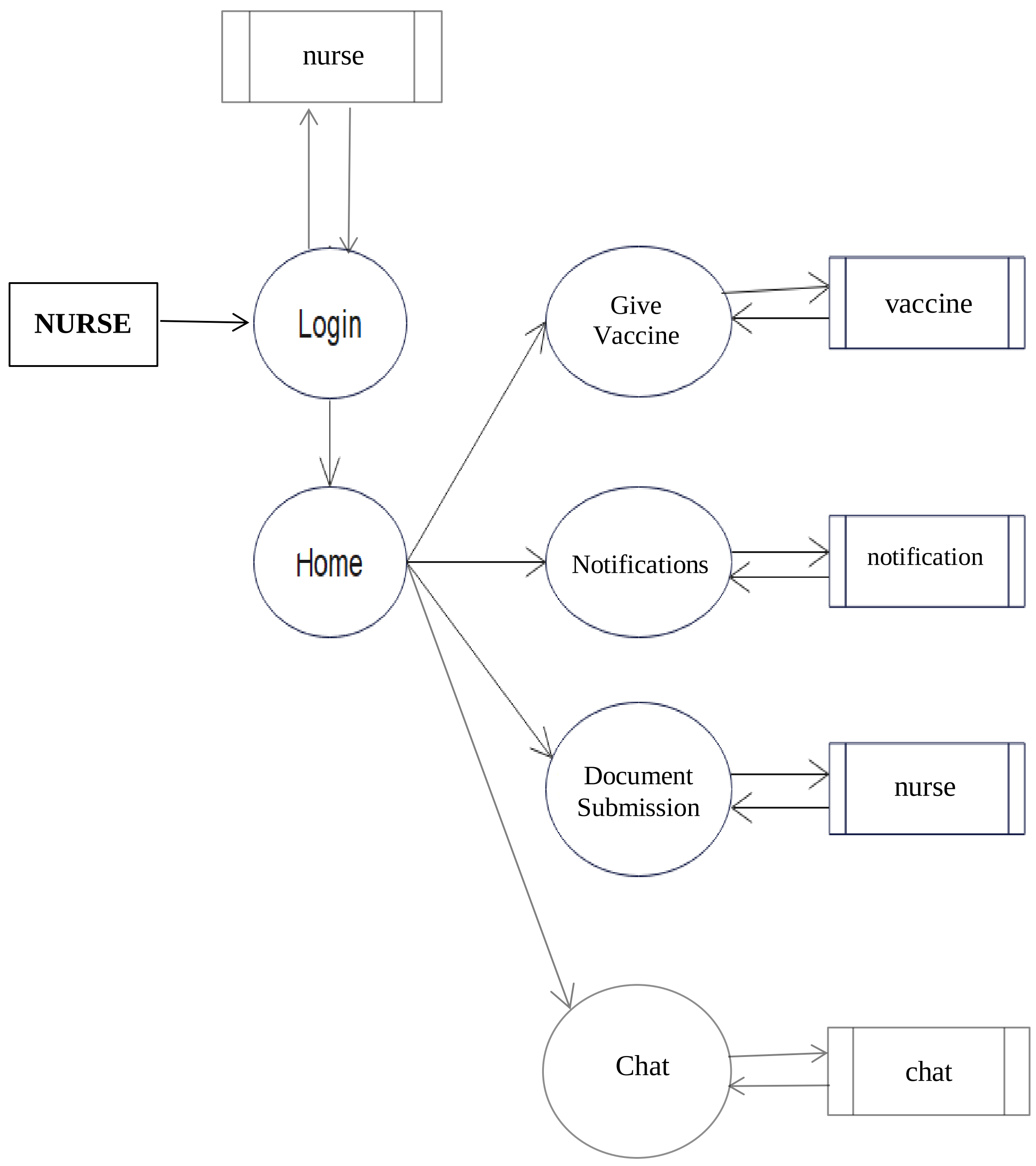
Level 1 ADMIN



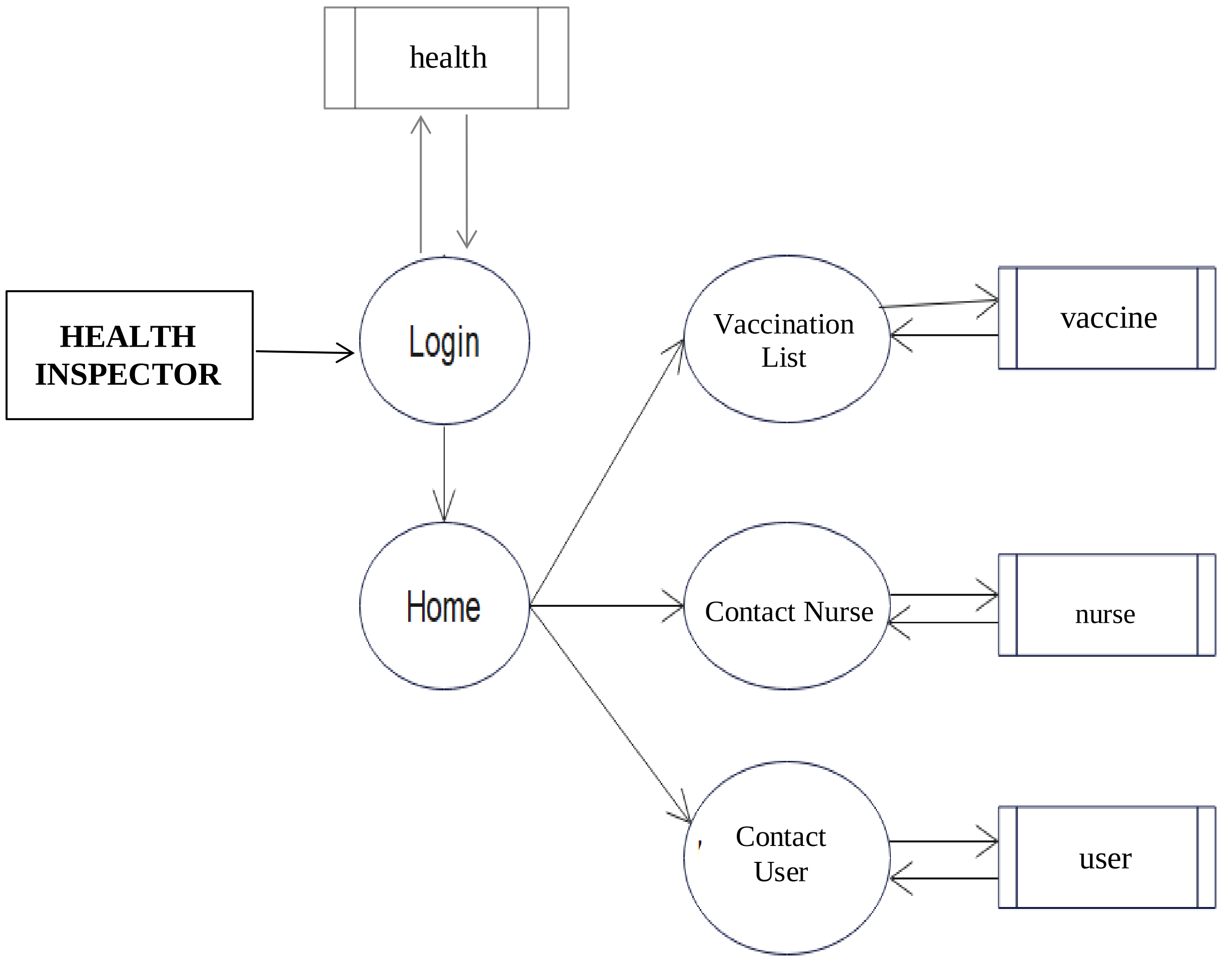
LEVEL 1 USER


user
USER
Registration
Login
Home
Vacination
vaccine
Notifications
notificatio
Chat
chat

LEVEL 1 NURSE



LEVEL 1 HEALTH INSPECTOR



4.5 STRUCTURE CHART

A structure chart is a chart which shows the breakdown of a system to its lowest manageable levels. It is a design tool that pictorially shows the relationship between processing modules in a computer software. Describes the hierarchy of components modules and the data are transmitted between them. This chart is used in structured programming to arrange the program modules in a tree structure. Each module is represented by a box, which contains the module name. The tree structure visualizes the relationships between the modules. A structure chart is a top-down modular design tool, constructed of squares representing the different modules of the system, and lines that connect them. The lines represent the connection or ownership between activities and sub activities as they are used in organizational charts. A structure chart depicts

- The size and complexity of the system
- The number of readily identifiable function and modules within each function.
- Whether each identifiable function is a manageable entity or should be broken down into smaller components.

Symbols used in structure chart are

- A rectangle represents the module
- Arrow represents the connection.

A structure chart depicts

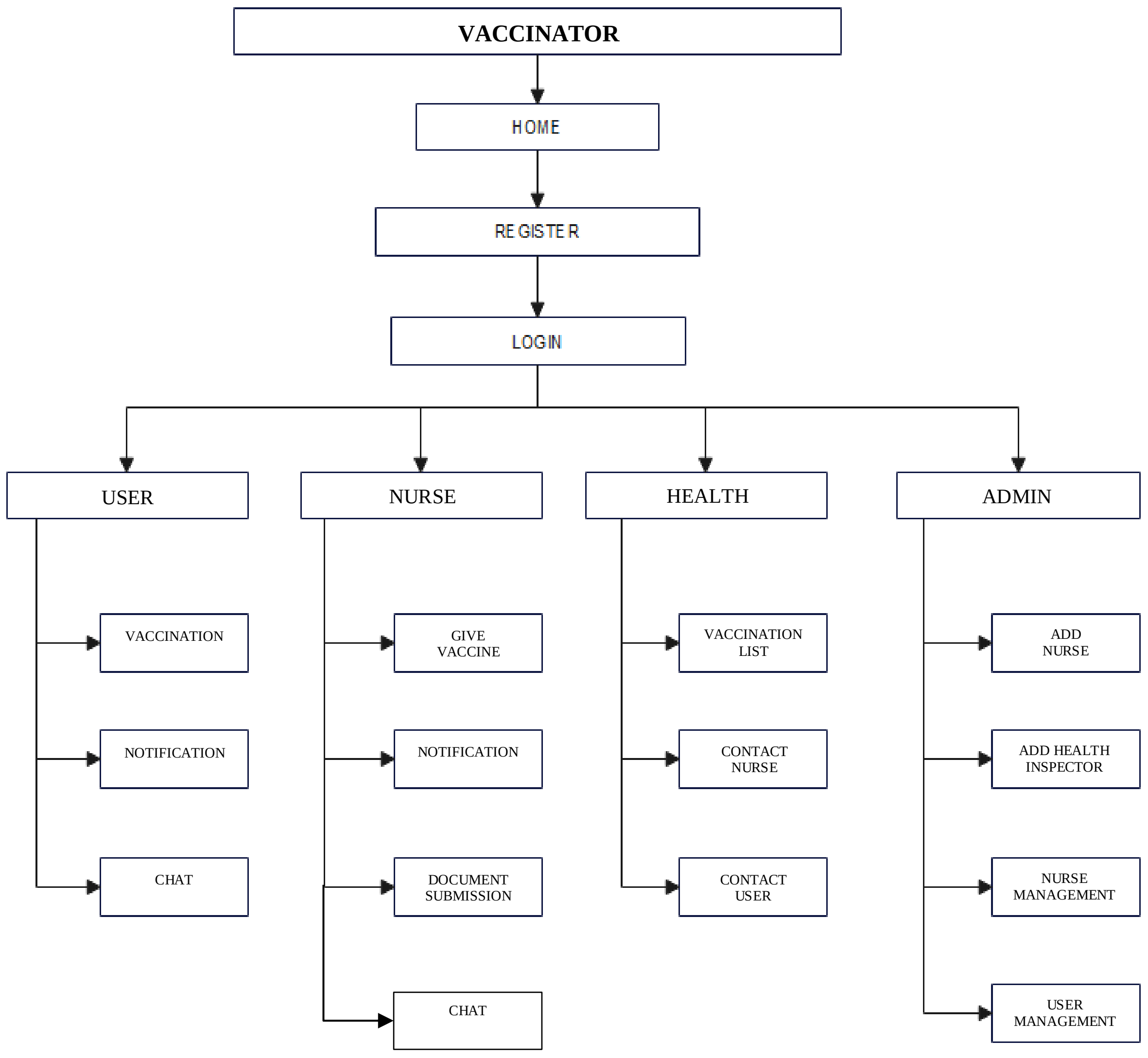
- The number of readily identifiable function and modules within each function.
- The size and complexity of the system
- Whether each identifiable function is a manageable entity or should be broken down into smaller components.

Symbols used in structure chart

A rectangle represents the module

Arrow represents the connection

STRUCTURE CHART



4.6 ENTITY RELATION DIAGRAM

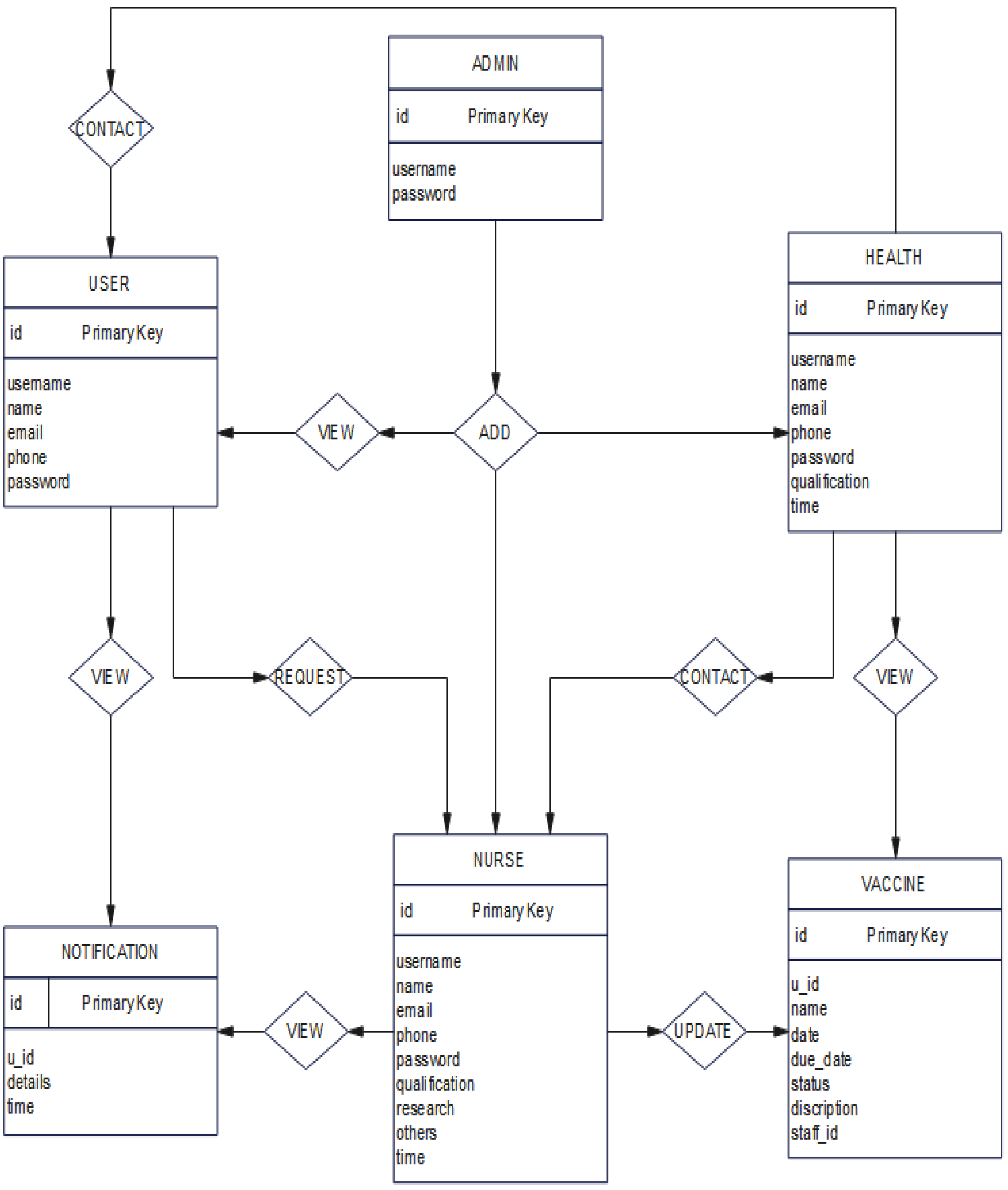
Introduction to ER Diagram

An entity-relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

In software engineering an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database. An E-R model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (*entities*) that are connected by lines (*relationships*) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: "*one building may be divided into zero or more apartments, but one apartment can only be located in one building*". Entities may be characterized not only by relationships, but also by additional properties (*attributes*), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute- relationship diagrams, rather than entity–relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

Vaccinator


ADMIN
id Primary Key
username
password
USER
id Primary Key
username
name
email
phone
password
HEALTH
id Primary Key
username
name
email
phone
password
qualification
time
CONTACT
VIEW
ADD
REQUEST
CONTACT
VIEW
NURSE
id Primary Key
username
name
email
phone
password
qualification
research
others
time
NOTIFICATION
id Primary Key
u_id
details
time
VIEW
UPDATE
VACCINE
id Primary Key
u_id
name
date
due_date
status
discription
staff_id

4.7 DATABASE DESIGN

A database is a collection of logically related data designed to meet the information needs of one or more users. It is a collection of records stored in a computer in a systematic way. The general theme behind a database is to integrate all the information. A database is an integrated collection of data and provides centralized access to the data.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database is to make the data access easy, inexpensive and flexible the user. The main objectives of designing a database are Data Integration, Data Integrity and Data Independence.

In the database, all the information is stored in the form of tables. A table is simply a way of storing data in rows and columns. In the system the data is stored in many tables. There are 8 tables used to store data. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data Independence

Design Methodology

A design methodology is the approach taken towards the design of database. It is the process of designing a database with a sound plan from the beginning. Database design is the process of converting business into tables and views. The most common hallmarks of a good database design are:

- Data storage needs have met.
- Date is readily available to the end-user
- Data is protected through database security
- Data is accurate and easy to manage
- Database accurately represents the business data
- Data can be easily backed up or recovered

- Overall database performance is acceptable
- Has minimum amount of redundant data stored

Database design is required to manage the large bodies of information. The management of data involves the definition of structure of storage of information and provisions of mechanism for the manipulation of information. For developing an efficient database, we will have to fulfill certain condition such as:

- Control redundancy
- Ease of use
- Data independence
- Accuracy and integrity
- Avoiding inordinate delays.
- Recovery from failure.
- Privacy and security.
- Performance

There are 6 major steps in the design process. The first five steps are usually done on a paper and finally the design is implemented:

- Identify the table and relationships
- Identify the data that is neither for each table and relationship
- Resolve the relationship
- Verify the design
- Implement the design

DESIGN CONSIDERATIONS

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

Identifying Keys:

Once we have drawn up 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.

Primary keys:

The primary key of a relation table uniquely identifies each record in the table. It can either be a 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 combination. This is because the primary key constitutes the main method of access to a record in the table and, as a rule; numeric keys do work and may be used when a suitable numeric key is to make absolutely sure that the field or fields you pick for the key will always be unique.

Foreign keys:

A foreign key is a key comprised of a field or multiple fields that to the primary key of another table.

IDENTIFY RELATIONSHIP

A relation 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 another tables. There are three possible types of relationship:

One-To-One-Relationships

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

One-To-Many-Relationships

In a One-To-Many-Relationship, any records in a table can relate to multiple records in a second table.

Many-To-Many-Relationships

A Many-To-Many Relationship, many records in a table can link too many records in the second table.

NORMALIZATION

Normalization of the data is a process of analysis the given relation schema based on their functional dependencies and primary keys to achieve the desirable properties of minimizing redundancy and minimizing the insertion, deletion and update anomalies. The normalization procedure provides:

- A frame work for analyzing relation schema based on their keys and on the functional dependencies among their attributes.
- A series of normal form test that can be carried out on individual relation schema so that the relational database can be normalized to any desired degree.

1NF

1NF is defined to disallow multi-valued attributes, composite attributes and their combinations. It states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.

2NF

2NF is based on the concept of full functional dependency. A functional dependency X -> Y is a fully functional dependency if removal of any attribute a form X means that the dependency does not hold any more; that is for any attribute A(-X,X(-X,X-{A})->Y.

A relation schema r is in 2NF if every nonprime attribute A in R is fully functionally depends on the primary key of R. the test for 2NF involves testing for functional dependencies whose left-hand side attribute are part of the primary key. If the primary key contains a single attribute, the test need not be applied at all.

In second normal form each row must contain associated field that describes an attribute of the entry that the table describes.

4.7.1 TABLES

1. TABLE : admin

DESCRIPTION : USED TO HOLD ADMIN DETAILS

Fieldname	**Data type**	**Size**	**Constraints**	**Description**
id	int	20	Primary key	Admin id
username	varchar	2000	Not null	User Name
password	varchar	100	Not null	Password

2. TABLE : health

DESCRIPTION : USED TO STORE HEALTH INSPECTOR DETAILS

Fieldname	**Data type**	**Size**	**Constraints**	**Description**
id	int	20	Primary key	Health Inspector id
username	varchar	2000	Not null	User Name
name	varchar	20	Not null	Name
email	varchar	30	Not null	Email
phone	varchar	15	Not null	Phone No
password	varchar	25	Not null	Password
qualification	varchar	200	Not null	Educational Qualifications
time	timestamp	6	Not null	Time Stamp

3. TABLE : nurse

DESCRIPTION: USED TO STORE NURSE DETAILS .

Fieldname	**Data type**	**Size**	**Constraints**	**Description**
id	int	20	Primary key	Nurse id
username	varchar	2000	Not null	User id
name	varchar	200	Not null	Name
email	varchar	30	Not null	Email
phone	varchar	15	Not null	Phone No
password	varchar	25	Not null	Password
qualification	varchar	200	Not null	Educational Qualification
research	varchar	2000	Not null	Research Topic
others	varchar	2000	Not null	Other Details
time	timestamp	6	Not null	Time Stamp

4. TABLE : notification

DESCRIPTION: USED TO STORE NOTIFICATIONS

Fieldname	**Data type**	**Size**	**Constraints**	**Description**
id	int	20	Primary key	Notification id
u_id	varchar	200	Not null	User id
details	varchar	2000	Not null	Notification Details
time	timestamp	6	Not null	Time Stamp

5. TABLE : user

DESCRIPTION: USED TO STORE USER DETAILS

Fieldname	**Data type**	**Size**	**Constraints**	**Description**
id	int	10	Primary key	User id
username	varchar	2000	Not null	User Name
name	varchar	25	Not null	Name
email	varchar	30	Not null	Email
phone	varchar	15	Not null	Phone No
password	varchar	20	Not null	Password

6. TABLE : vaccine

DESCRIPTION: USED TO STORE VACCINE DETAILD

Fieldname	**Data type**	**Size**	**Constraints**	**Description**
id	int	10	Primary key	ID
u_id	varchar	200	Not null	User id
name	varchar	200	Not null	Name
date	timestamp	6	Not null	Date
due_date	datetime	6	Not null	Due Date
status	int	200	Not null	Vaccine Status
discription	varchar	2000	Not null	Vaccine Description
staff_id	varchar	2000	Not null	Nurse id

ARCHITECTURAL DESIGN

Architectural Design comprise of the design and effective integration of various components of the software. This project is architecturally well designed. Since there is two part in the project, client side and server side, their architecture is kept consistent within each part. Each one has a unique procedure. The client focuses on the user interaction part, thus designed with an eye on user friendly designs. The interface is designed in such a way that the user has no need to search for important information that has to be conveyed. Instead proper design concepts are implemented to catch the attention of user. in the same way server side design has all the information that has to be informed with the server administrator. It has extensive interfaces that make software to be used in an effective way. The server administrator has protected access to the information in an effective way

User Interface Design

The user interface is the most important element of a computer based system or product. The user interface design process encompasses four distinct framework activities.

- User, task and environment analysis and modeling
- Interface design
- Interface construction
- Interface validation

User, task and environment analysis and modeling

User interface design begins with the identification of user task and environment requirements. The initial analysis activity focuses on the project of the users who will interact with the system. Task analysis was done to find out the entire task that will be performed by the user. All the major tasks were considered and elaborated into sub-tasks. These sub-tasks that will accommodate user model, design model and the system perception. The user environment was also considered and an analysis model was create with information gather and the design process was started.

Interface Design

In this phase the goals and intention of each user was established. The object and actions were isolated to create a list of objects and actions. The source objects, the target objects and the impact of the each action were identified. Then the screen layout was performed. The design of web pages was carried out in the screen layout process.

The interface prototype is made from initial interface design. Once the interface prototype is established this is evaluated by the user and the designer. Further modifications are made based in the suggestion.

Procedural Design

A design methodology combines a systematic set of rules for creating a program design with diagramming tools needed to represent it. Procedural design is best used to model programs that have an obvious flow of data from input to output. It represents the architecture of a program as a set of interacting processes that pass data from one another. Design tools the two major diagramming tools used in procedural design are data flow diagrams and structure charts. Data Flow Diagram is a tools to help you discover and document the program`s major process. The following table shows the symbol used and what each represents. The design of an object- oriented software requires the definition of a multilayered software.

System design process

System design develops the architectural details required to build a system or product. The system design process encompasses the following activities:

- Partition the analysis model into subsystems.
- Identify the concurrency

5. CODING

INTRODUCTION TO CODING

The purpose of coding is to translate the software design into source code. Each component of the design is implemented as a program module. The end product of this phase is a set o program modules that have been individually tested. The main characteristic of good code is uniqueness, stability, expandability, uniform size and format, portability, simplicity and versatility. There are mainly two types of code design are top-down design and bottom up design. In top-down design, the implementation starts from the top of the hierarchy and proceeds to the lower level and in the bottom-up level, the development starts with implementing the module at the bottom of the hierarchy and proceeds through the higher levels until it reaches the top. The code should be easy to modify to reflect necessary changes in condition, characteristics and relationship of the encoded entities.

The goal of coding or programming activity is to implement the design in the best possible manner. The coding activity affects both testing and maintenance profoundly.

```
 <?php
session_start();
$user = $_POST['username'];
$pass = $_POST['password'];
$role = $_POST['role'];

include 'connection.php';

if($role == "admin")
{
   if(!empty($user)||!empty($pass)){
$sql = "SELECT * FROM admin WHERE username='$user' AND password='$pass'";
$res = mysqli_query($conn,$sql);
$count = mysqli_num_rows($res);

            if($count == 0)
            {
                    echo '<script language="javascript">';
                    echo 'alert("Username OR password dosent exist please check if role is
defined correctly in the drop down")';
                        echo '</script>';
                        header("location:index.php");
                        }
                else
                {
                        $_SESSION['username'] = $user;
                        $_SESSION['MODULE'] = $role;
                        header("location:ADMIN/index.php");
                }
        }
        else {
            echo 'Fill up all fields';
        }
      }
      else if($role == "nurse")
      {
        if(!empty($user)||!empty($pass)){
        $sql = "SELECT * FROM nurse WHERE username='$user' AND password='$pass'";
        $res = mysqli_query($conn,$sql);
        $count = mysqli_num_rows($res);

                if($count == 0)
                {
                echo '<script language="javascript">';
                echo 'alert("Username OR password dosent exist please check if role is defined
correctly in the drop down")';
                echo '</script>';
                header("location:index.php");
                }
                else
                {
                        $_SESSION['username'] = $user;
                        $_SESSION['MODULE'] = $role;
                        header("location:NURSE/index.php");
                }
            }
     else {echo 'Fill up al fields';}
    }
    else if($role == "health")
    {
       if(!empty($user)||!empty($pass)){
       $sql = "SELECT * FROM health WHERE username='$user' AND password='$pass'";
       $res = mysqli_query($conn,$sql);
       $count = mysqli_num_rows($res);

                if($count == 0)
                {
                        echo '<script language="javascript">';
                        echo 'alert("Username OR password dosent exist please check if role is
defined correctly in the drop down")';
                        echo '</script>';
                header("location:index.php");
                }
                else
                {
                        $_SESSION['username'] = $user;
                        $_SESSION['MODULE'] = $role;
                        header("location:HEALTH/index.php");
                }
            }
     else {echo 'Fill up al fields';}
    }
    else if($role == "user")
    {
       if(!empty($user)||!empty($pass)){
       $sql = "SELECT * FROM user WHERE username='$user' AND password='$pass'";
       $res = mysqli_query($conn,$sql);
       $count = mysqli_num_rows($res);
```

```
                if($count == 0)
                {
                echo '<script language="javascript">';
                echo 'alert("Username OR password dosent exist please check if role is defined
correctly in the drop down")';
                echo '</script>';

                }
                else
                {
                        $_SESSION['username'] = $user;
                        $_SESSION['MODULE'] = $role;
                        header("location:USER/index.php");
                }
            }
     else {echo 'Fill up al fields';}
     }
     ?>
```

```
     <?php
     include 'connection.php';
       if(isset($_POST['add']))
       {
         if($_POST['stpass']==$_POST['cstpass'])
         {
           $id=$_POST['username'];
           $name=$_POST['name'];
           $email=$_POST['email'];
           $phone=$_POST['phone'];
           $pass= $_POST['stpass'];
```

```
                if (!empty($id)|| !empty($name)||!empty($email)||!empty($phone)||!empty($pass))
                {

                 $sql= "INSERT INTO `vaccinations`.`user` (`username`, `name`, `email`, `phone`,
   `password`) VALUES ('$id', '$name', '$email', '$phone', '$pass');";
                    mysqli_query($conn, $sql);
                    $conn->close();
                    header('Location:index.php');
                }
              else

              {
                   echo 'Please fill up all fields';
                   header('Location:reistration.php');
              }
             }
             else{
               echo '<script type="text/javascript">alert("Password Miss match")</script>';
             }



          }


        ?>
        <html>
           <head>
           <meta name="viewport" content="width=device-width, initial-scale=1">
           <link                                                                  rel="stylesheet"
   href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"   integrity="sha384-
   Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
   crossorigin="anonymous">
        <style>
        .boy {
          margin: 0 auto;
          max-width: 800px;
```

```
  padding: 0 20px;
}

.container {
  border: 2px solid #dedede;
  background-color: #f1f1f1;
  border-radius: 5px;
  padding: 10px;
  margin: 10px 0;
  max-height: 100px;
}

.darker {
  border-color: #ccc;
  background-color: #ddd;
}

.container::after {
  content: "";
  clear: both;
  display: table;
}

.container img {
  float: left;
  max-width: 60px;
  width: 100%;
  margin-right: 20px;
  border-radius: 50%;
}

.container img.right {
  float: right;
  margin-left: 20px;
  margin-right:0;
```

```
}

.time-right {
  float: right;
  color: #aaa;
}

.time-left {
  float: left;
  color: #999;
}
</style>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand"href="#">Registration Gains you acess to our premium nurses who would guide you to your goals for complete vaccination.</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
</nav>
<div class="boy">
  <form form action="" method="post">
    <div class="form-group">
    <label for="exampleInputEmail1">Username</label>
    <input type="text" name="username"class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Must be unique" onkeyup="if (/[^|a-z0-9]+/g.test(this.value)) this.value = this.value.replace(/[^|a-z0-9]+/g,")" required>
    </div>
    <div class="form-group">
    <label for="exampleInputEmail1">Name</label>
    <input type="text" id="in" name="name"class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter Your Name" required>
```

```
              </div>
              <div class="form-group">
              <label for="exampleInputEmail1">Email</label>
              <input type="email" id="in" name="email"class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter Your Email" required>
              </div>
              <div class="form-group">
              <label for="exampleInputEmail1">Phone</label>
              <input type="number" id="in" name="phone"class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" pattern="[789][0-9]{9}" placeholder="Enter valid Indian phone no" required>
              </div>
              <div class="form-group">
              <label for="exampleInputEmail1">Password</label>
              <input type="password" id="in" name="stpass" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Password" required>
              </div>
              <div class="form-group">
              <label for="exampleInputEmail1">Conform Password</label>
              <input type="password" id="in" name="cstpass" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Conform Password" required>
              </div>
              <input value="submit"  type="submit"  name="add" id="bt"class="btn btn-primary">
              </form>
          </div>
          </div>
          </body>
          </html>
```

6. TESTING

INTRODUCTION

IEEE defines testing as a process of analyzing a software item to detect the difference between the existing and required conditions and evaluate the features of the features of the software item. The testing is the process of analyzing a program with the intention of finding errors. Software testing is a critical element of total quality assurance and represents the ultimate reviews on specification, design and coding. Testing present is an interesting anomaly for the software. Testing is the stage of implementation, which is aimed ensure that the system works efficiently before the real operation commences.

Implementation is an important aspect of a system analyst's job is to make sure that the new design is implemented to the established standards. The term implementation has different meanings, ranging, from the conversion of a basic application, to a complete replacement of a computer system. Implementation used here is to mean the process of converting a new system design into an operational one.

6.1 TESTING STRATEGIES

A test strategy is an outline that describes the testing approach of the software development life cycle. It is created to inform project managers, testers, and developers about some key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment.

Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of testing are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used and occasionally, conceptual design documents may be referred to. Design documents describe the functionality of the software to be enabled in the upcoming release. For every stage of development design, a corresponding test strategy should be created to test the new feature sets.

6.1.1 UNIT TESTING

At the lowest levels unit testing where different modules are tested against the specification produced during design for the modules. Unit design is essential for the verification of the code produced during the coding phase, and hence the goal is to test the internal logic of the modules. Each program unit and form modules are tested rigorously during the development and again before interconnecting these modules. During the time program, unit testing emphasis is on the correctness of procedures program logic, control flow, etc.

6.1.2 INTEGRATION TESTING

In this level of testing many tested modules are combined into subsystems, which are then tested. The goal here is to see is the modules can be integrated properly, the emphasis being on testing interfaces between modules. The tested form modules are integrated and tested for correct interfaces, parameter passing, coordinated operation and such issues using simple data collected from the actual documents and files already available with the department. Top down integration strategy is employed for integration testing.

6.1.3 VALIDATION TESTING

Validation succeeds when the developed software function in a manner that can reasonably be expected by the user. The system is ensured that the software specification is satisfied. The user of non-developers tests it. System testing is stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. The principle, system providing is an ongoing activity throughout the project. The logical design and the physical design should be thoroughly and continually examined on paper to ensure that they will work when implemented Thus the system test in implication should be confirmation that all is correct and an opportunity to show the user that the system works.

7. SYSTEM IMPLEMENTATION

INTRODUCTION

Implementation is the process of converting of a new or revised system design into an operation one. An implementation plan is to be made before starting the actual implementation of the system. Implementation is the stage where the theoretical design is converted into a working system.

Implementation includes all those activities that take place to convert from the developed system to the working one. The new system may be totally new or it may be provide major modification to an existing system. Proper implementation is essential to provide a reliable system to meet users the new system, but improper installation will prevent it.

7.1 IMPLEMENTATION PLAN

For a successful implementation of the system, implementation plan is necessary. Its major elements include test plan, an equipment installation plan and launching plan.

The test plan provides for the preparation of the test and for testing the system is planned, structured manner. Training plan is necessary to ensure that persons who are associated with the computer related information system have necessary knowledge and skills. Equipment implementation activities are equipment installation and hardware checkout.

Launching is the process of initiating and performing all the physical operations that result directly in the turnover of the new system to the user. There are two parts of launching. The launching plan is implementing throughout the development phase into the operational phase. The launching plan includes procedural launching and file launching.

7.2 SYSTEM MAINTENANCE

The launching of the system, even though it is fully correct and complete is not the end of the matter. The system should be given proper security and maintenance in order to keep them efficient and up-to-date. The system security is for protection against fraud and disaster. To avoid unauthorized access, password protection is highly recommended while running this new system. The password as to be maintained directly and files have to be kept very confidential. Finally systems and programs that have been successfully implemented are usually subjected to continuous change. The system should made modification and improvement to meet changing conditions. The first class maintenance requires the cooperation of the people served by the system or programs and those reasonable for maintaining it.

7.3 MAINTENANCE ISSUE

Maintenance ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer decides a change in requirements. The software is characterized by the following activities. In this project considerable amount of time is spent in maintenance and monitoring.

CORRECTIVE MAINTENANCE

Corrective maintenance is to uncover the errors still exist after testing. During this maintenance work the user is asked to work on the system and if any error is reported.

ADAPTIVE MAINTENANCE

The adaptive maintenance is needed if the platform or the environment of the project to be change. For the project the language takes care of all these things.

PERFECTIVE MAINTENANCE

The third maintenance active is the perfective maintenance. The recommendation of new capabilities and modification of existing function and general enhancement are received from the user and proposed future enhancement.

PREVENTIVE MAINTENANCE

The preventive maintenance is to improve the future maintainability and reliability and to provide better basis for future enhancement.

7.4 IMPLEMENTATION AND MAINTANANCE

The definition of software maintenance can be given by describing four activities that are undertaken after the program is released for use. The first maintenance activity occurs since it is unreasonable to assume that software testing will uncover all errors in a large software system. The process of including the diagnosis and correction one or more errors is called corrective maintenance.

The second activity that contributes to definitions of maintenance occurs since rapid change in encountered in very aspects of computing. Therefore adaptive maintenance modifies software to properly interface with a changing environment.

The third activity involves the recommendations for new capabilities; modification to the existing function and general enhancement when the software is used. To satisfy request perceptive maintenance is performed.

The forth maintenance activity occurs when the software is changed to improve future maintainability or reliability. This is called maintenance. This project is prone to all type of maintenance activity and carried out quite well.

7.5 LIMITATIONS AND ENHANCEMENTS

Limitations:

- It is open discussion forum so that everyone uploads unwanted and wrong information so that it misleads the users.
- Quite inefficiency in querying details.

Enhancements:

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on object-oriented design, any further changes can be easily adaptable.
- Based on the future security issues, security can be improved using emerging technologies.
- Admin can get some transaction profit percentage.

8. CONCLUSION

CONCLUSION

By this project, we provide a Vaccination Management System for Kids, which is an advanced software solution designed to simplify and automate the management of vaccination processes for children. The system consists of four primary modules: Admin, Vaccination Provider, Vaccination Receiver, and Health Inspector. The Admin module acts as the central control hub, responsible for user role management and access privileges. The Vaccination Provider, who is a nurse or a designated healthcare professional, interacts with the system to input user details and provide registration numbers along with the Date of Birth (DOB) of the recipients.

Vaccinator offers a comprehensive and efficient solution for managing the vaccination process for children. By incorporating the Admin, Vaccination Provider, Vaccination Receiver, and Health Inspector modules, the system facilitates seamless coordination among various stakeholders, enhances data accuracy, and expedites the immunization process for children.

9.FUTURE ENHANCEMENTS

FUTURE ENHANCEMENT

The future enhancement of our project is to reduce the limitation to some extent. The enhancement may be required if there is any change in the available resources . Enhancement means adding, modifying or developing the code to support the changes in specification. Every module in the system is being developed carefully such that the future enhancement do not affect the basic performance of the system. In future we can add any links or services to the system very easily. In future we can add :

- Warehouse system, we could store the Vaccines.
- We could introduce a paid section where users have to pay a small fee to get their periodic vaccination alert.

10. APPENDIX

10.1 APPENDIX-A : SOURCE CODE

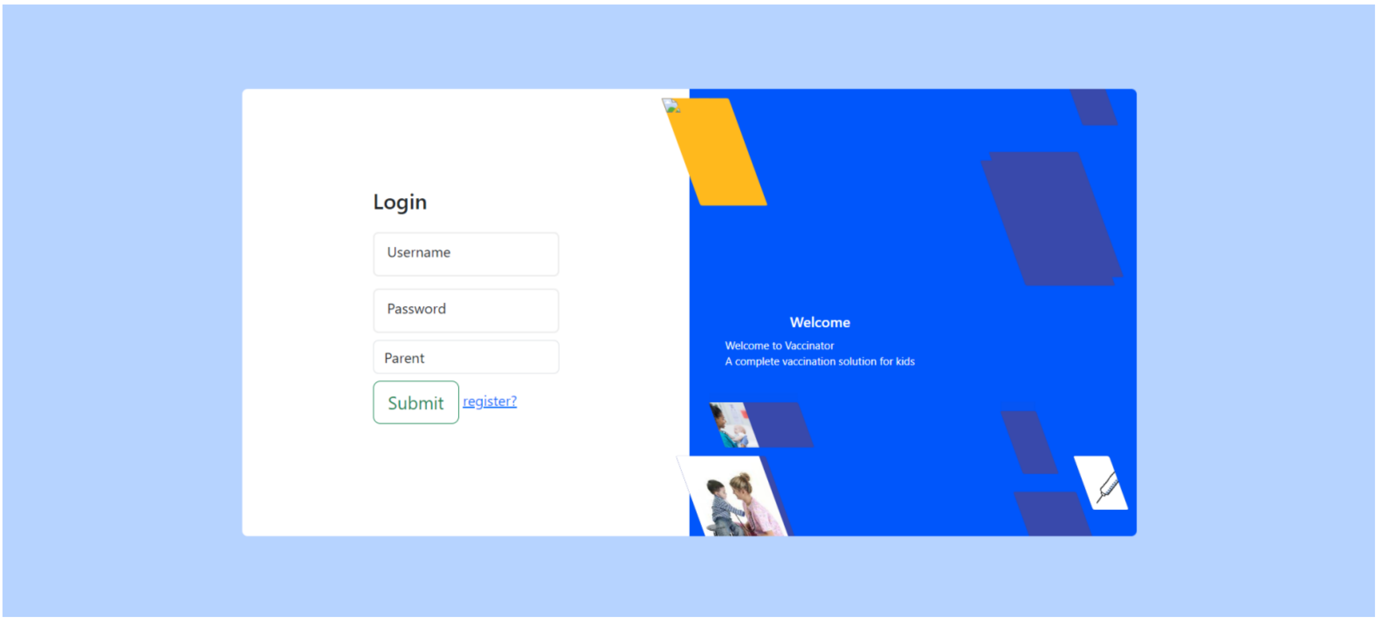
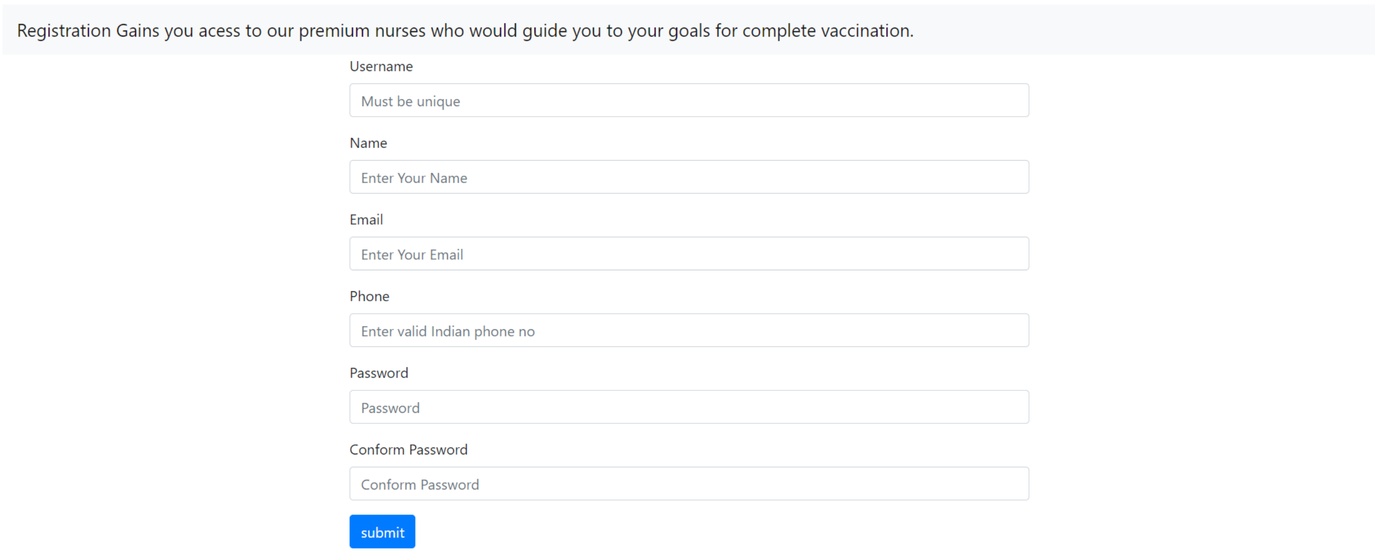


Fig 8.1 : Login


Registration Gains you acess to our premium nurses who would guide you to your goals for complete vaccination.

Username

Must be unique

Name

Enter Your Name

Email

Enter Your Email

Phone

Enter valid Indian phone no

Password

Password

Conform Password

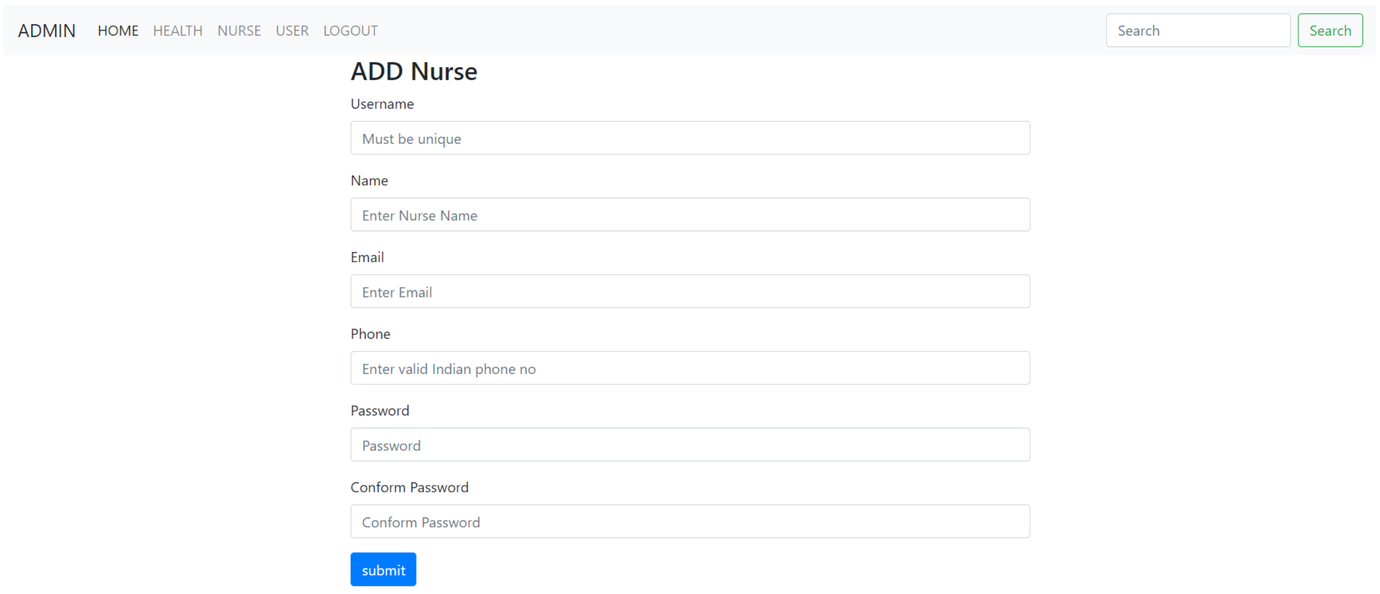
Conform Password

submit

Fig 8.2 : Register

ADMIN PAGES

ADMIN HOME



ADMIN HOME HEALTH NURSE USER LOGOUT Search Search

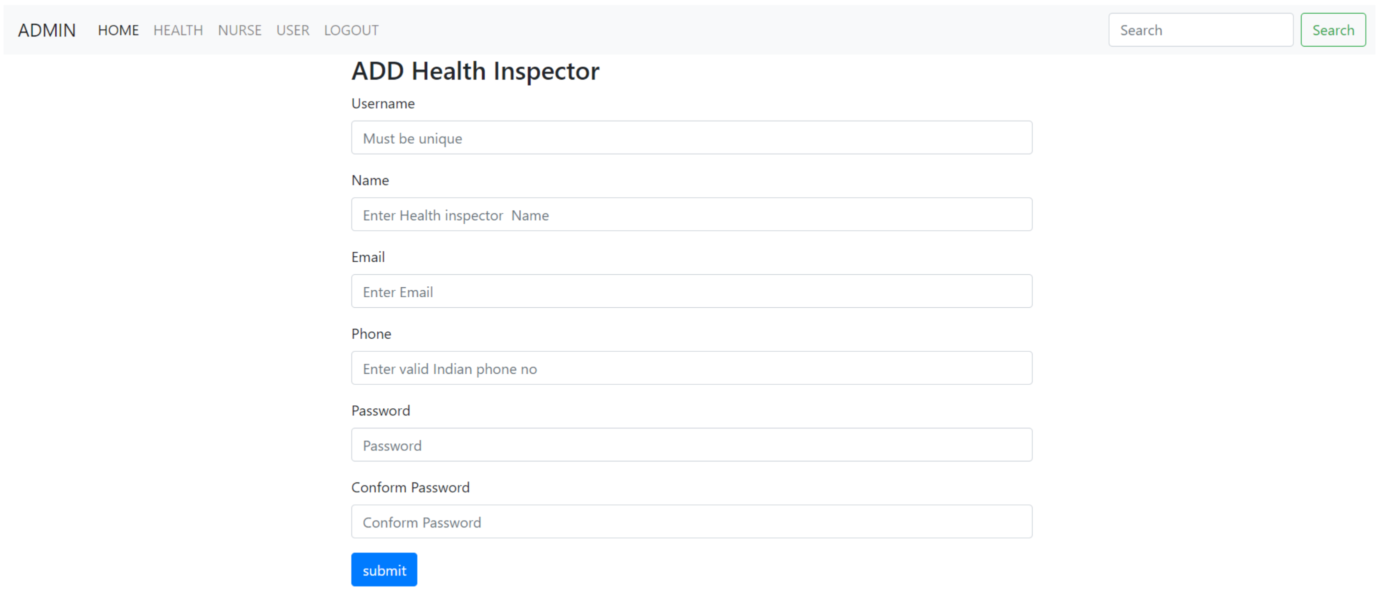
ADD Nurse

Username	Must be unique
Name	Enter Nurse Name
Email	Enter Email
Phone	Enter valid Indian phone no
Password	Password
Conform Password	Conform Password

submit

Fig 8.5 : Admin Home

ADD HEALTH INSPECTOR



ADMIN HOME HEALTH NURSE USER LOGOUT Search Search

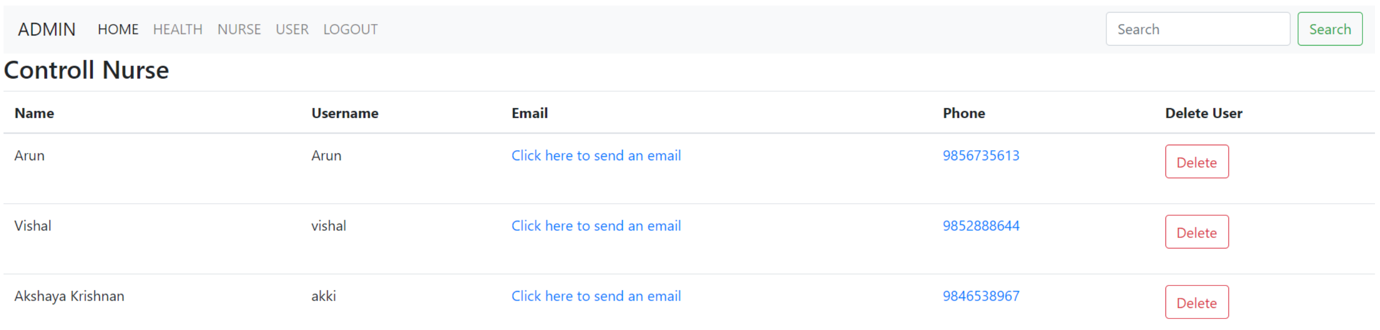
ADD Health Inspector

Username	Must be unique
Name	Enter Health inspector Name
Email	Enter Email
Phone	Enter valid Indian phone no
Password	Password
Conform Password	Conform Password

submit

Fig 8.6 : Add Health Inspector

NURSE MANAGEMENT

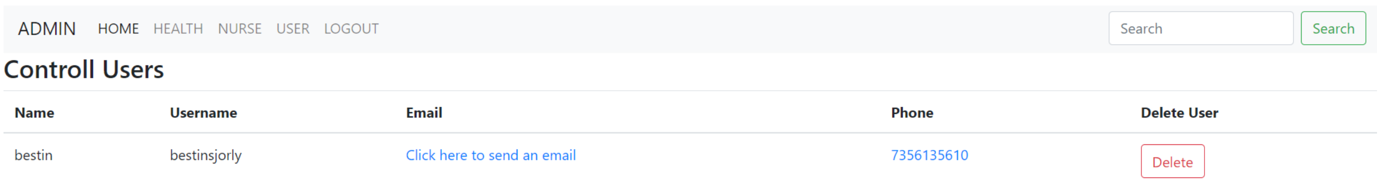


Controll Nurse

Name	Username	Email	Phone	Delete User
Arun	Arun	Click here to send an email	9856735613	Delete
Vishal	vishal	Click here to send an email	9852888644	Delete
Akshaya Krishnan	akki	Click here to send an email	9846538967	Delete

Fig 8.7: Nurse Management

USER MANAGEMENT



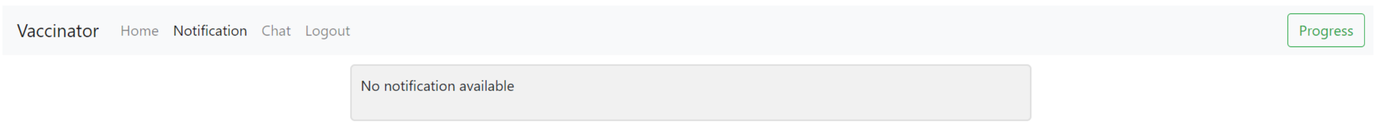
Controll Users

Name	Username	Email	Phone	Delete User
bestin	bestinsjorly	Click here to send an email	7356135610	Delete

Fig 8.8: User Management

USER PAGES

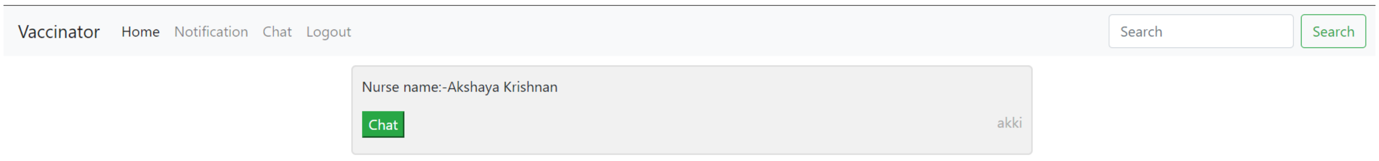
NOTIFICATIONS



No notification available

Fig 8.9: Notification

CHAT



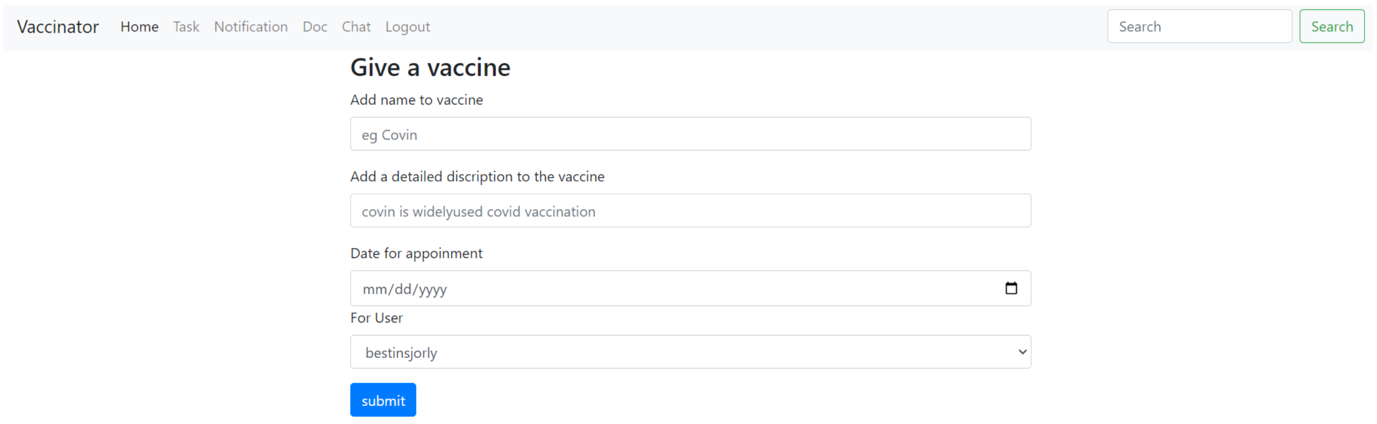
Nurse name:-Akshaya Krishnan

akki

Fig 8.10: Chat

NURSE PAGES

VACCINE



Give a vaccine

Add name to vaccine

eg Covin

Add a detailed discription to the vaccine

covin is widelyused covid vaccination

Date for appoinment

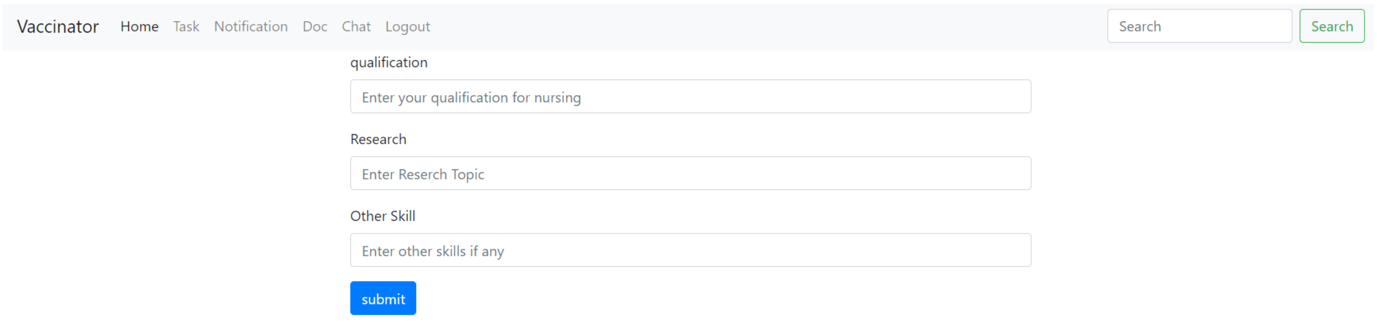
mm/dd/yyyy

For User

bestinsjorly

Fig 8.11: Vaccine

DOCUMENT SUBMISSION



qualification

Enter your qualification for nursing

Research

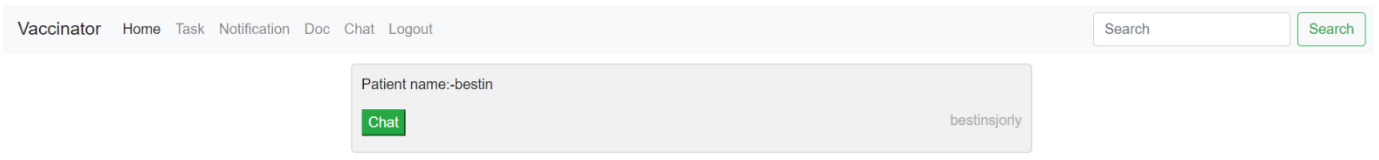
Enter Reserch Topic

Other Skill

Enter other skills if any

Fig 8.12: Document Submission

CHAT



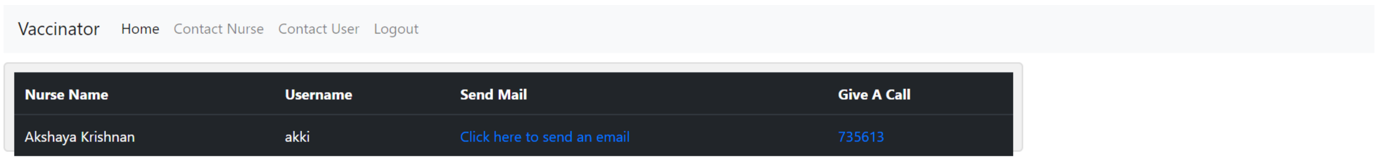
Patient name:-bestin

bestinsjorly

Fig 8.13: Chat

HEALTH INSPECTOR PAGES

CONTACT NURSE



Nurse Name	Username	Send Mail	Give A Call
Akshaya Krishnan	akki	Click here to send an email	735613

Fig 8.14: Contact Nurse

BIBILOGRAPY

1. Rajib Mall, ”Fundamentals of Software Engineering “ ,Third Edition 2000.

2. Elias M Award, “System Analysis and Design”, Galgotia publications, Second Edition 1985.

WEBSITES

- http://www.w3schools.com/php
- http://www.google.com