**Vaccination Alert**

**SOFTWARE ENGINEERING PROJECT REPORT**



**UNIVERSITY OF DELHI, NEW DELHI**

**Submitted By:-**

**Aashima-17044570004**

**Bhawana-17044570007**

**B.Sc (H) COMPUTER SCIENCE**

**IV SEMESTER**

**MATA SUNDRI COLLEGE FOR WOMEN**

**NEW DELHI**

**ACKNOWLEDGEMENT**

It is our privilege to express our sincerest regards to our project coordinator, **Ms. Ashema Hasti**, for her valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our faculty **Ms. Ashema Hasti** for encouraging and allowing us to present the project on the topic **“VACCINATION ALERT”** at our department premises for the partial fulfilment of the requirements leading to the award of B.Sc. degree.

We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our career. Last but not the least we express our thanks to our friends for their cooperation and support.

AASHIMA(17044570004)

BHAWANA(17044570007)

**CERTIFICATE**

This is to certify that the Software Engineering project report titled **“VACCINATION ALERT”** is the work carried out by Bhawana and Aashima, students of B.Sc.(H) Computer Science - 4th Semester, Mata Sundri College For Women, University of Delhi under the supervision of **Ms. Ashema Hasti, Assistant Professor, Department of Computer Science, Mata Sundri College For Women.**

This report has not been submitted to any other institution for the award of any other degree/diploma.

**Ashema Hasti**

**(Project Guide)**

**ABSTRACT**

Vaccination Alert software is designed for providing facility to parents and increasing the vaccination rate. The topic is chosen because of decreasing rate of vaccination day by day. According to a survey, children are facing health issues due to lack of vaccination because of the busy schedule of parents and due to this they forget about their infants vaccination. To increase the vaccination rate, we started this project. This software is different from other existing softwares as it consumes small memory and also caretaker can consult with the doctor. Here, caretaker is the guardian or the parent. For this, we choose a methodology, and according to that caretaker’s register on this software and fill all the details asked by the software. This will complete in two steps, first one is for providing the caretaker’s details and second for infant details. After registration, they receive time to time alert messages for their infants vaccination. Also, in case if caretaker face any issue, s/he can consult with doctor and for this doctor will take a little fee. After doing all this work, we achieved our goal.

Also, this software is available only in english language and due to this, limited users can use this software. In future, we’ll make available this software in hindi and English, both the languages.

**LIST OF FIGURES USED IN THE PROJECT**

|  |  |  |
| --- | --- | --- |
| **FIGURE\_NO** | **NAME** | **PAGE No** |
| 1.1 | INCREMENTAL MODEL | 13 |
| 3.1 | CONTEXT LEVEL DFD | 26 |
| 3.2 | LEVEL – 1 DFD | 27 |
| 3.3 | LEVEL – 2 SIGN UP | 28 |
| 3.4 | LEVEL – 2 LOGIN | 28 |
| 3.5 | LEVEL – 2 UPDATE | 29 |
| 4.6 | LEVEL – 2 PAYMENT | 29 |
| 3.7 | USE CASE DIAGRAM | 32 |
| 7.1 | HOME PAGE | 59 |
| 7.2 | LOGIN PAGE | 59 |
| 7.3 | REGISTRATION PAGE | 60 |
| 7.4 | INFANT’S DETAILS | 60 |
| 7.5 | CARETAKER PROFILE | 61 |
| 7.6 | DOCTOR PROFILE | 61 |
| 7.7 | VACCINATION CHART | 62 |
| 7.8 | UPDATE DETAILS | 62 |
| 7.9 | AVAILABILITY OF DOCTORS | 63 |
| 7.10 | PAYMENT | 63 |

**LIST OF TABLES USED IN THE PROJECT**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **NAME** | **Page NO** |
| 3.1 | DOMAIN | 30 |
| 3.2 | CARETAKER’S CREDENTIALS | 30 |
| 3.3 | INFANT’S CREDENTIALS | 30 |
| 3.4 | DOCTOR’S CREDENTIALS | 31 |
| 4.1 | ADMIN | 40 |
| 4.2 | DOCTOR | 40 |
| 4.3 | CARETAKER | 41 |
| 5.1 | PROJECT SCHEDULING | 43-44 |
| 5.2 | TIMELINE CHART | 44-45 |
| 5.3 | SIZE ESTIMATION | 47-48 |
| 5.4 | FUNCTION POINT COMPLEXITY | 48 |
| 5.5 | TDI RESPONSES | 49 |
| 5.6 | COCOMO II COMPLEXITY WEIGHTS | 51 |
| 5.7 | PRODUCTIVITY RATE FOR OBJECT POINT COUNTS | 52 |
| 5.8 | RISK ANALYSIS | 56-57 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SNO** | **TOPIC** | **Page No** |
|  | PROBLEM STATEMENT | 8 |
|  | SOFTWARE LIFECYCLE MODEL | 11 |
|  | SOFTWARE REQUIREMENTS  SPECIFICATION | 22 |
|  | CONTEXT LEVEL DIAGRAM | 24 |
|  | DFD LEVEL 1 | 25 |
|  | DFD LEVEL 2 | 26 |
|  | DATA DICTIONARY | 28 |
|  | USE CASE DIAGRAM | 30 |
|  | USE CASE DESCRIPTION | 31 |
|  | DATA DESIGN | 38 |
|  | ESTIMATION AND SCHEDULING | 41 |
|  | FUNCTION POINT METRICS | 45 |
|  | COCOMO MODEL | 48 |
|  | RISK ANALYSIS | 54 |

**PROBLEM STATEMENT**

Nearly 2 Million children under the age of 5 years die every year in India .The Indian Academy of Pediatrics(IAP) estimates that over 50 percent of these are vaccine preventable. In order to prevent the infants from hazardous diseases such as small pox , hepatitis B , tetanus ,etc. we’ve taken this initiative .

This app is easy and convenient to use and it maintains the data of users and the main motto of our vaccination model is to provide convenience to the parents. The desired impact of this app is to ensure that the children receive all the mandatory vaccination on time.

On getting registered with this app, it automatically reminds or lets the person know about their babies vaccination and in case, if parents wants to consult the doctor then can do that too through this app.

Vaccination model is specially designed for maintaining the health of babies. So, parents can register on this software and once they register and fill all the details, they’ve no need to worry about remembering the dates of their child’s vaccines, they automatically receive reminder notifications.

The VACCINATION ALERT app provides 3 reminder alerts for the caregiver in a week that the vaccinations are due. The app requires data of both the child and the parent or caregiver. It’s the responsibility of the registering caregiver to provide the correct information at the time of signing up. The app uses the information provided to schedule the reminder alert till the child is 12 years old.

**ROLE PLAYERS**

1. Admin

2. Doctors

3. Caretaker

**Admin’s functionality :-**

* **Login/Logout –** Admin has to login to access and manage the software.
* **Upload instructions –** Admin will upload precautions which is common for all the users.
* **Generate vaccination chart –** Admin will generate the vaccination chart suggested by the government of India.
* **Add/Remove Doctors –** Admin can add or remove doctors on the basis of their qualification.
* **Update –** Admin can update any kind of changes.
* **Total billing/month –** Admin will keep a check on payment status.

**Doctor’s Functionality:-**

* **Login/Logout –**Doctor has authority to login/logout to the system and manage the software.
* **Generate -**  Doctor can generate infants record.
* **View infants record –** To check infants growth, doctor can view infants record.
* Caretaker can consult with the doctor by checking their availability.

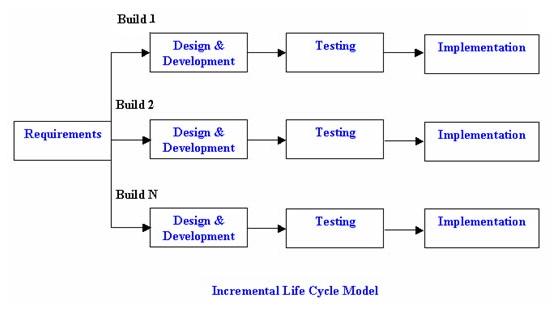
**Caretaker’s Functionality:-**

* **Register –** New caretakers can register to the software.
* **Login/Logout –** Registered Caretakers will simply to the software with their username and password.
* **Provide information –** Caretaker will give all the details of their child like age, weight, height, any disabilities etc.
* **Print-** Caretaker can print the health records of their infants.
* **Check availability–** Caretaker can check the availability of doctors.
* **Request –** If doctor is not available and caretaker want to talk to doctor very urgently, s/he can make request to doctor for consultancy.
* **Payment-** Caretaker has to pay doctor’s fee.
* **Update-** After a particular interval of time, caretaker will update their infant record such as weight, height etc.
* **Receive Notifications**
* **Contact us**

**SOFTWARE LIFECYCLE MODEL**

Vaccination Alert App follows Incremental model.

Incremental Model combines both linear and iterative flow. In iterative model, instead of starting with full specification of requirements, it‘s is divided into it’s sub processes and each process will linearly work.

**Figure 1.1** Incremental Model

This is the basic flow of incremental model.

The first increment is a core or the base in which the basic requirements are introduced. And then, in further increments, the desired requirements will meet.

Vaccination Alert Software is using incremental model because the basic functionality is fixed but there are chances of adding or removing features or we can say requirements can change in future.

Also this is cost efficient.

Using this model, we can make updation if there is any need due to change in government rules etc. Also at present, this software is available only in English language but in future, it will be available in other languages also. So, using the incremental model the software’s functionality can be expanded in later after the software release.

**INTRODUCTION**

Vaccination is the most effective method of preventing infectious diseases such as polio, measles, and tetanus from much of the world. Childhood vaccination is usually provided as a routine service in maternal-child health clinics or other health facilities.

Parents need to do everything possible to make sure their children are healthy and protected from preventable diseases. Children should receive the vaccinations they need at the right age during scheduled or drop-in clinic visits. Outbreaks of preventable diseases occur when many parents decide not to vaccinate their children or when they forget about their child’s vaccination. If children are not vaccinated, they can spread disease to other children who are too young to be vaccinated or to people with weakened immune systems. Because of advances in medical science, child can be protected against more diseases than ever before.

**Vaccination Alert** is an android application for tracking baby’s vaccination. This android application works by taking the information about the child like date of birth, gender, city, any kind of inability, etc. It sends the notifications to the users to remind their baby’s vaccination due.

**1.1 PURPOSE**

* To provide alert for upcoming vaccination with its price based on birth date of the baby according to immunization chart prescribed by government of India.
* It also helps the customer to find the address and contact number of nearby hospitals for treatment.
* Keep a track of your child’s growth by storing information like his height and weight to get graphical details about his growth.
* To provide description of vaccination along with its side effects if any.
* Quick view summary report of child's completed and upcoming vaccines.
* Send a message regarding polio vaccination date and location of nearest polio booth.
* **The purpose of this software is to increase the vaccination rate.**

**1.2 SCOPE**

The software will be used as an application that serves to peoples having new born babies. The intention of making this software is to increase the vaccination rate. Parents become careless sometimes and most often they forget about vaccination. This app is serving this facility to parents.

**1.3 METHODOLOGY**

This project is based on the database, Android based and web based techniques. To keep the records in database it uses MySQL software, which is one of the best and the easiest database to keep information. This project uses Java as the front-end software which is an Android based Programming. The user’s details are stored in a database like their name, date of birth and gender, contact, etc. This enables the admin of the application to control and keep track of the number of users. It identifies information to include in a text message reminder, conduct requirements gathering to build a text message feature and plan for implementation of the SMS feature. Group discussion has to be conducted to determine effectiveness of strategies and methods for sending immunization reminders and determine appropriate message content, the frequency of sending messages, and the message preferences for missed and upcoming appointments.

**1.4 ACRONYMS AND ABBREVIATIONS**

* **SRS:** SOFṬWARE REQUIRENMENT SPECIFICATION
* **DFD:** DATA FLOW DIAGRAM
* **Info:** Information
* **H/W:** Hardware
* **S/W:** Software
* **Sign up:** Creating new user
* **Login:** Logging in Existing user
* **Addr :** Address
* **Ph No. :** Phone Number
* **Expr :** Experience
* **INdb :** Infant’s database
* **DOCdb :** Doctor’s database

**1.5 REFERENCES**

* R.S Pressman, Software Engineering : A Practitioner’s Approach, Mc Graw-Hill, Edition-7 (2010)
* P. Jalote, an Integrated Approach to Software Engineering, Narosa publication house, Edition -3 (2011).
* https://www.cdc.gov/vaccines/imz-managers/laws/index.html

cdc – Centers for disease controls and prevention.

**1.6 ADVANTAGE AND DISADVANTAGES:**

**Advantages:**

* **Vaccination Alert** app is a native app. Also you can access it on browser.
* It’s user-friendly.
* Very less memory is required.
* It decreases the overhead of Parents via sending reminder to them.
* In this software, Caretaker can consult with doctors also.
* At present it’s available in English, but very soon it’ll be available in English and Hindi, both the languages.

**Disadvantages:**

* Parents should have knowledge of operating mobile phones on which this app can run.
* Availability of Internet is must.
* Admin has to manually keep updating the information by entering the details it the system.
* Doctor will be available online only.

**1.7 OVERVIEW**

Our project **Vaccination** **Alert** includes registration of parents having infants, storing their details into the system, getting alert about vaccination, etc.

Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. Caretaker can also consult with the doctor via checking their availability. Once the caretaker get registered on the software, for further entering in the Vaccination Alert System, username and password is sufficient. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

**THE OVERALL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

Vaccination Alert attempts to find a solution forthe parents specially for those who forget abouttheir babies vaccines. This product increases thevaccination rate also.

The main functionalities of this application is:

* Register to the software.
* Login/Logout.
* Generate Vaccination Chart for users.
* Send Vaccination Alert to the users.
* Send message for updating infants records.
* Provide Doctors.

**2.1.1 SYSTEM INTERFACES**

This system is designed to be transparent to itsusers and hence all the complexity is hidden fromthe user, i.e., user has no need to take care aboutthe internal working. The user will interact withsystem using the GUI.

* ***User Interfaces***
* This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.
* The **protocol used** shall be **HTTP**.
* The Port number used will be 80.
* There shall be logical address of the system in IPv4 format.
* **Hardware Interfaces**

No specific hardware is required. The app needs just a software compatible hardware on which app can run, i.e., Android mobile, etc.

* **Software Interfaces**
* **Operating System:** We have chosen windowsoperating system for its best support, performanceand user friendliness.
* **Database:** To save the records of the users andtheir details, SQL database is used.

**2.1.2 OPERATIONS Of CARETAKER**

The basic operations of the Vaccination Alert appare described as follows:

• The caretaker when using the application for thefirst time has to register with the application.

• The caretaker after registering, can login to theapplication with his username and password.

• The caretaker will be initially shown with the formwherein he has to enter the details, about hisinfants age, gender, height, weight, anydisabilities, any allergy, etc. Also caretaker has tofill it’s own contact info, address, etc.

• The system will then generate VaccinationChart for the caretakers.

• The system will evaluate the time of infantsvaccines and send SMS alert accordingly.

• Caretaker can check the availability of the doctors.

• Caretaker will make request to the doctor forconsultation, incase if the doctor is not available and caretaker needs doctor urgently.

• The caretaker can logout of the application at anytime by clicking on the logout button.

**HERE CARETAKER IS THE GUARDIAN OR THE PARENT.**

**SPECIFIC REQUIREMENTS**

**3.1 Performance requirements**

* **Response time**- Response time will be minimum. System should response within 0.5 seconds atmost.
* **Capacity**-The system must support 500 people at a time.

**3.2 Safety Requirements**

If there is any damage to a wide portion of the database due to any kind of failure, such as a disk crash, the recovery method restores a past copy of the database, So that users data will not lose. Also nobody can change system’s internal records except the system administrator.

**3.3 Security Requirements**

1. Want take the responsibility of failures due to hardware malfunctioning.
2. Warranty period of maintaining the software would be one year.
3. Additional payments will be analyzed and charged for further maintenance.
4. If any error occur due to a user’s improper use. Warranty will not be allocated to it.
5. No money back returns for the software.
6. User’s data must be secure.

**3.4 Software system attributes**

**3.4.1 Correctness:** Software should be correct in all aspects, also it should meet all the caretaker’s requirements.

**3.4.2 Completeness:** Software system should be complete.

**3.4.3 Availability:** The system shall be available all the time. Means caretaker can access the software anytime.

**3.4.4 Usability:** Software can be used again and again without any distortion.

**3.4.5 Accessibility:** Administrator and many other users can access the system but the access level, vary from user to user means admin, caretaker and doctor, is controlled for each user according to their work scope.

**3.4.6 Accuracy:** The system should be accurate and reliable.

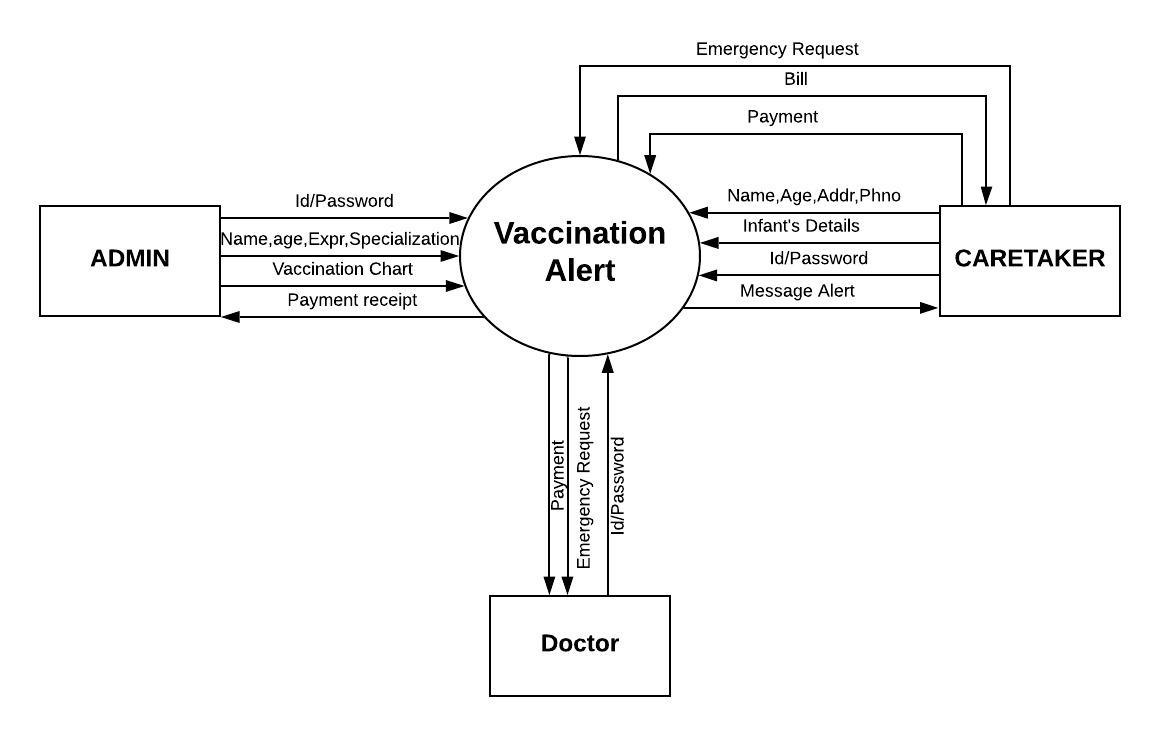
**3.4.7 Stability:** System should be stable.

The system output won’t change time to time. Same output will be given always for a given input.

**3.4.8 Maintainability and Modifiability:** It’s structure and style are such that if there is need to any change, that changes should easily made.

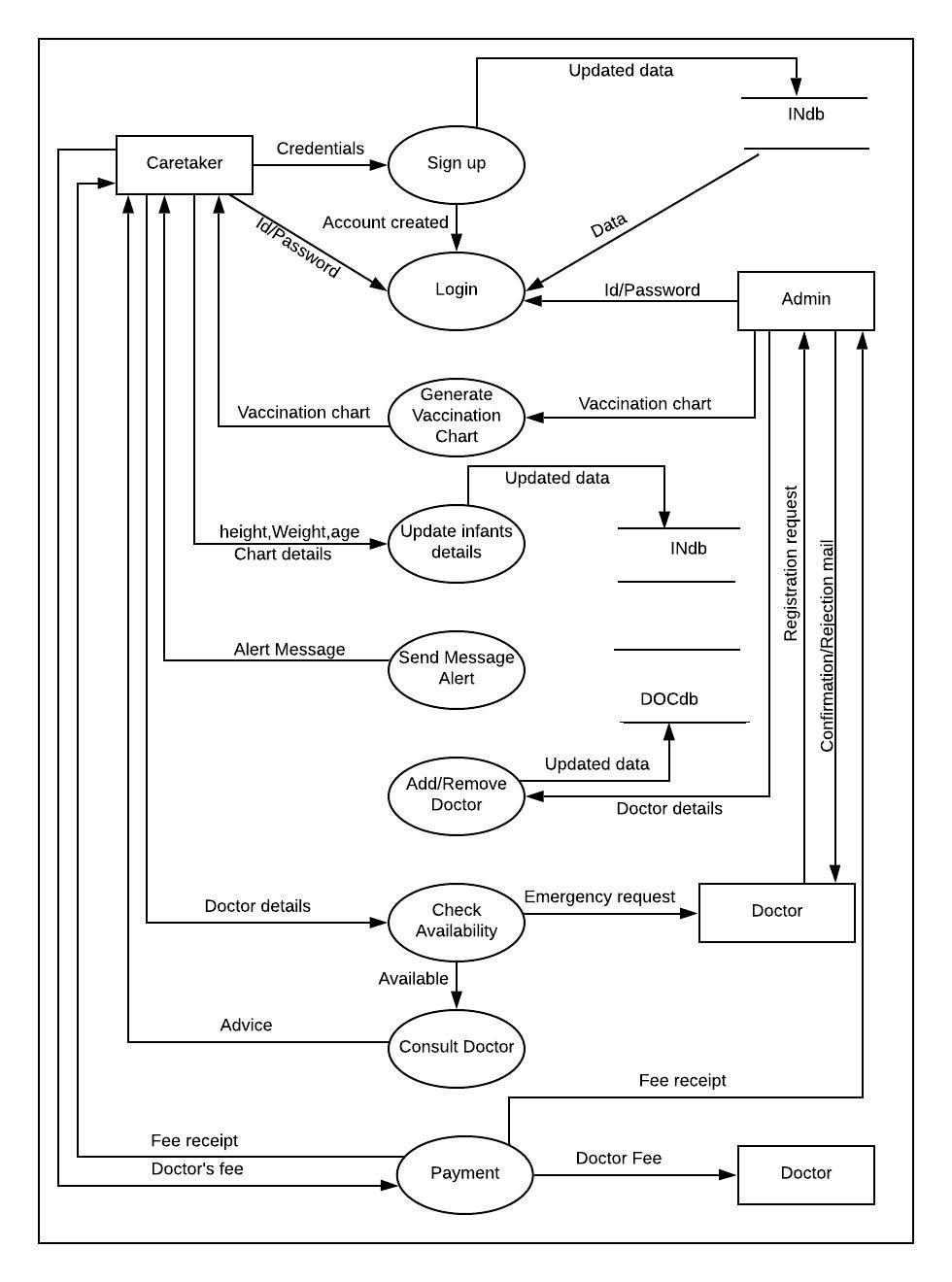
System should have ability to maintain, modify information and update fix problems of the system.

* 1. **DATA FLOW DIAGRAM (DFD)**
     1. **CONTEXT LEVEL DIAGRAM (DFD LEVEL – 0)**

****

**Figure 3.1** Context level DFD

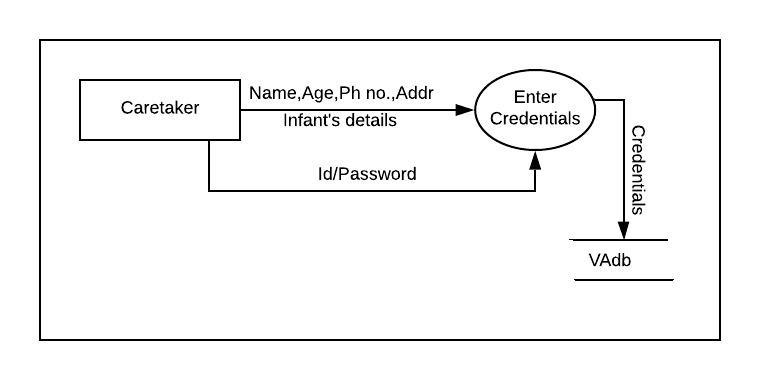
**3.5.2 DFD LEVEL -1**

****

**Figure 3.2** Level-1 DFD

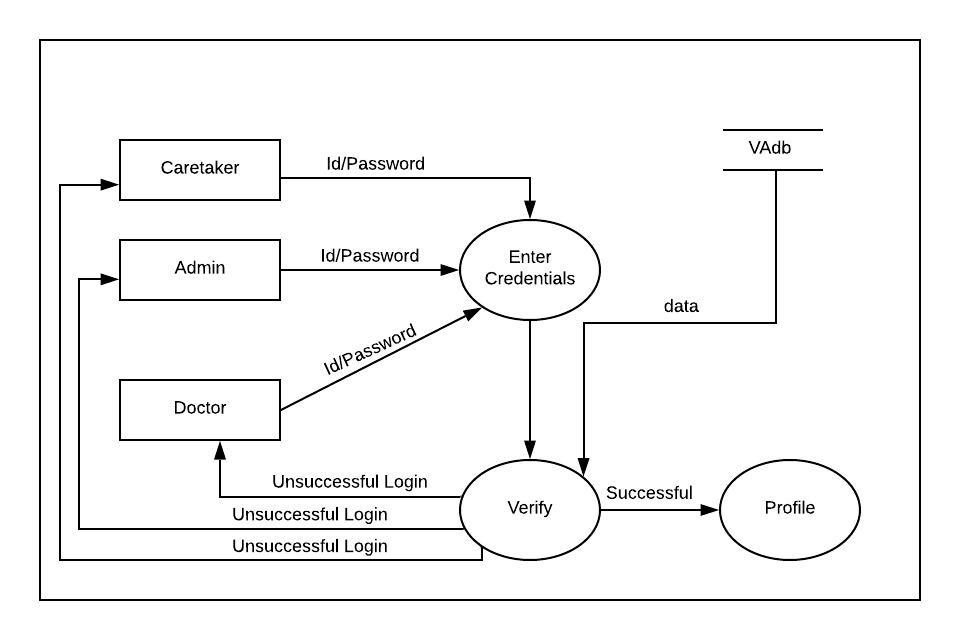
* + 1. **DFD LEVEL – 2**

**3.5.2.1 Sign up**

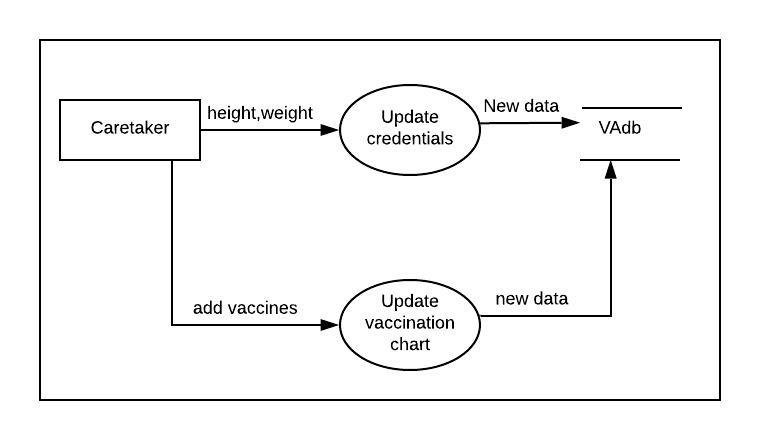
****

**Figure 3.3** DFD LEVEL-2(Sign up)

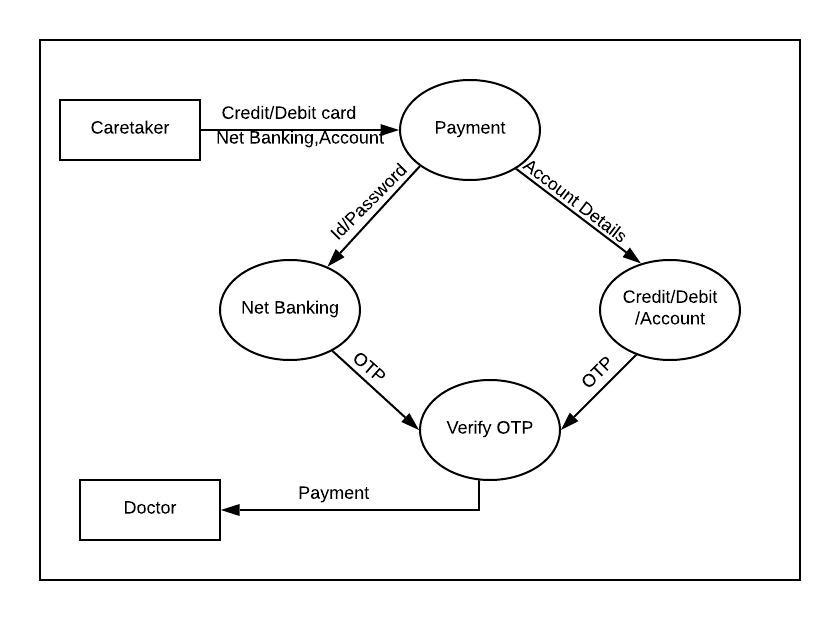
**3.5.2.2 Login**

****

**Figure 3.4** DFD LEVEL-1(Login)

**3.5.2.3 Update**

**Figure 3.5** DFD LEVEL-2(Update)

**3.5.2.4 Payment **

**Figure 3.6** DFD LEVEL-2(Payment)

**3.6 DATA DICTIONARY**

The dictionary is an organized list of all data elements that are linked to the system so that both user and analysts have a common understanding of inputs and outputs

**Domain**

|  |  |  |
| --- | --- | --- |
|  | legal\_Ch | [a-z| A-Z] |
|  | Digit | [0-9] |
|  | Special\_Ch | [@|$|#|+|-|,] |

**Table 3.1** Domain

**Caretaker’s credentials**

|  |  |  |
| --- | --- | --- |
|  | Caretaker’s Name | First\_name + (Middle\_name) + Last\_name |
|  | First\_name | {Legal Ch}\* |
|  | Middle\_name | {Legal\_Ch}\* |
|  | Last\_name | {Legal\_Ch}\* |
|  | LoginID | {Legal Ch + digit + Special Ch}\* |
|  | Password | {Legal\_Ch + Digit + Special\_Ch}\* |
|  | Mobile No. | {Digit}\* |
|  | Relation with Infant | {Legal Ch}\* |
|  | Address | House\_no + (Street) + City + State + Pincode |
|  | House\_no | {Legal\_Ch + Digit}\* |
|  | Street | {Legal\_Ch}\* |
|  | City | {Legal\_Ch}\* |
|  | State | {Legal\_Ch}\* |
|  | Pincode | {Digit}\* |

**Table 3.2** Caretaker’s credentials

Infant’s Credentials

|  |  |  |
| --- | --- | --- |
| 1. | Infant’s Name | First\_name + (Middle\_name) + Last\_name |
| 2. | First\_name | {Legal Ch}\* |
| 3. | Middle\_name | {Legal\_Ch}\* |
| 4. | Last\_name | {Legal\_Ch}\* |
| 5. | Date of Birth | {Digit + Legal Ch + Special Ch} |
| 6 | Gender | {Legal Ch} |
| 7. | Height | {Legal Ch + Digit} |
| 8. | Allergy | {Legal Ch + Special Ch} |
| 9. | Disability | {Legal Ch + Special Ch} |

**Table 3.3** Infant’s Credentials

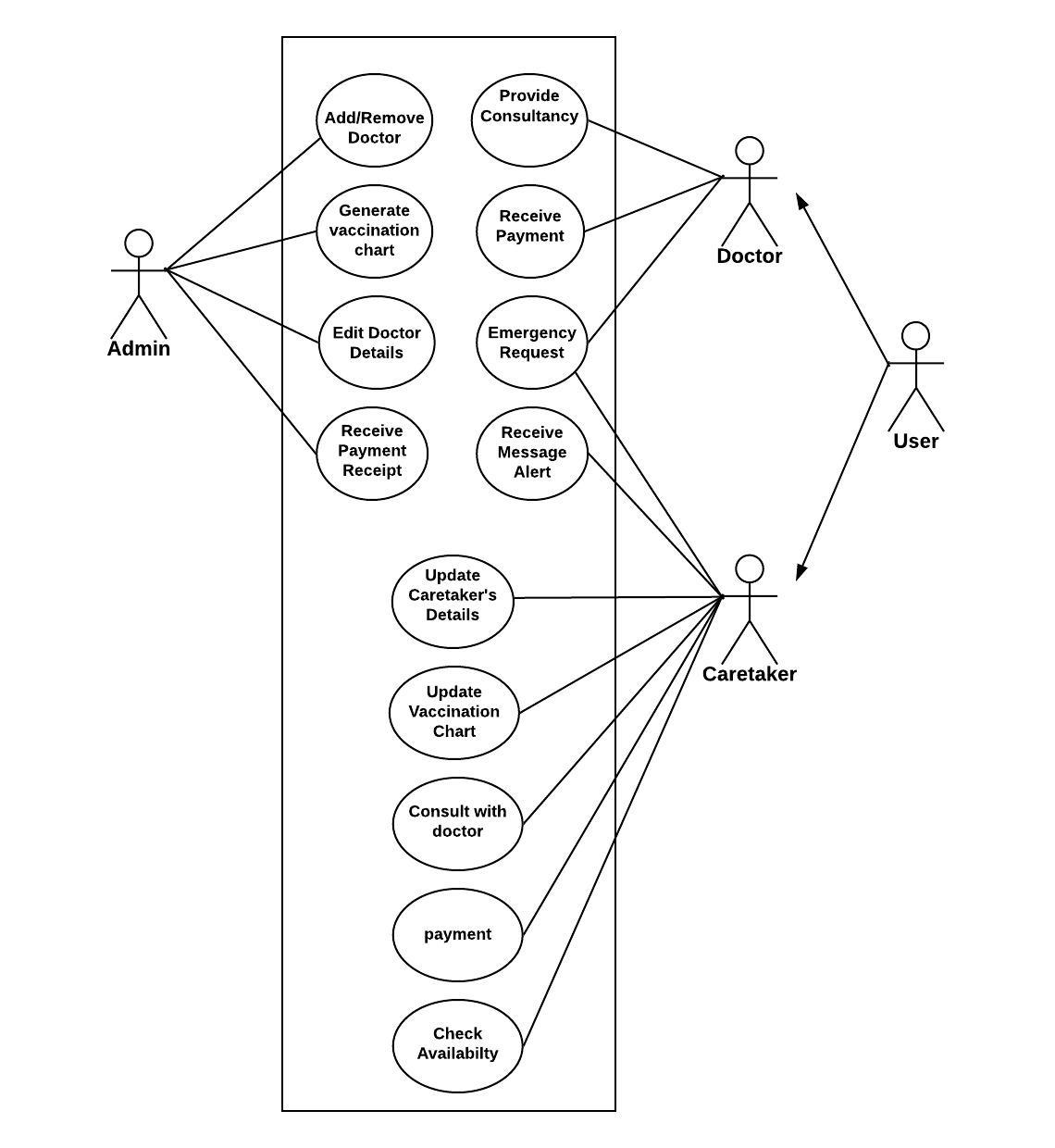
Doctor

|  |  |  |
| --- | --- | --- |
| 1. | Doctor’s Id | {Legal Ch + digit + Special Ch}\* |
| 2. | Password | {Legal Ch + digit + Special Ch}\* |
| 3. | Advice | {Legal Ch + digit + Special Ch}\* |
| 4. | Doctor’s Fee | {Legal Ch + Digit} |

**Table 3.4**  Doctor Credentials

**3.7 Use Cases**

**3.7.1 Use Case Diagram**



**Figure 3.7** Use Case Diagram

**3.7.2 USE CASE DESCRIPTION**

**Caretaker**

* **SignUp**

1. **Brief Description :** This use case describes how caretakers registers into the **‘Vaccination Alert’** system.
2. **Flow of events**
3. Basic flow
   * This use case starts when the caretaker wishes to register to the ‘Vaccination Alert’ system.
   * The system requests that the actor caretaker his/her infant’s (name, age, DOB, height, weight, medical history, any disabilities), email, Password, contact info, relation with infant, address.
   * The caretaker enters all the details asked by system.
   * The system stores the entered attributes in the database and the caretaker is then registered to the system and a confirmation mail is sent to the caretaker.
   * It’s a one-time process. After signing up/registration caretaker needs to enter only username and id to log in to the system again.
4. **Pre-conditions**

None

1. **Post conditions**

If the use case was successful, the caretaker is registered to the system. If not, the system state is unchanged.

1. **Extension points**

None

* **Login**

1. **Brief Description**

This use case describes how caretaker logs into the **‘Vaccination Alert’** system.

1. **Flow of events**
2. a) Basic flow

* This use case starts when the caretaker wishes to log in to the ‘Vaccination Alert’ system.
* The system requests that the caretaker enter his/her username and password.
* The caretaker enters his/her username and password.
* The system validates the entered name and password and the caretaker is then logged in to the system.

1. b) Alternative flow

If the caretaker enters an invalid name or password, the system displays an error message. The caretaker can use to either return to the beginning of the basic flow or cancel the login at the point where use case ends.

1. **Pre-conditions**

For login, caretaker must have an account created on the system before login.

1. **Post conditions**

If the use case was successful, the caretaker is logged in to the system. If not, the system state is unchanged.

Every caretaker has the access to the corresponding screens to his/her role.

1. **Extension points**

None

* **Update changes**

1. **Brief Description**

This use case describes how caretaker updates the desired changes in different fields in a caretaker’s records.

1. **Flow of events**

a) Basic flow

* 1.1 This use case starts when the caretaker needs to update the desired changes in different fields in it’s own records
* 1.2Caretaker update the infants details like as height, weight, etc. S/He also update the vaccination chart indicating the vaccination state.

1. **Special Requirements**

None

1. **Pre-conditions**

Login to app

1. **Post conditions**

If the use case was successful, the changes are updated in corresponding fields in the database. If not, the system state is unchanged.

1. **Extension points**

None

* **Generate Vaccination Chart**

1. **Brief Description**

This use case describes how an actor generates the **‘Vaccination Alert’** system.

1. **Actors**

Admin

Caretaker

1. **Flow of events**
2. a) Basic flow

This use case starts when admin generates vaccination chart. After registration, caretaker can view the vaccination chart.

1. **Special Requirements**

None

1. **Pre-conditions**

Before executing this use case, actors must have account created on this system.

For existing users, login must be done.

1. **Post conditions**

If the use case was successful, the vaccination chart is stored in the caretaker’s database and is available to the caretaker. If not, the system state is unchanged.

1. **Extension points**

None

* **Send SMS Alert**

1. **Brief Description**

This use case describes how SMS is sent to the caretaker for vaccination alert.

1. **Actors**

Caretaker

1. **Flow of events**
2. Basic flow
   * This use case starts when there is a time of infants vaccination.An Alert Message is sent to caretaker for vaccination.
3. Alternative flow
   * If user has not updated that vaccination is completed, message is again sent to the caretaker.
4. **Special Requirements**

None

1. **Pre-conditions**

None

1. **Post conditions**

If the use case was successful, the changes are updated in corresponding fields in the database. If not, the system state is unchanged.

1. **Extension points**

None

* + **Consult Doctor**

1. **Brief Description**

This use case describes how caretaker consults with doctor.

1. **Flow of events**

a) Basic flow

This use case starts when caretaker wants to consult with doctor.

1. **Special Requirements**

None

1. **Pre-conditions**

Login

1. **Post conditions**

If the use case was successful, caretaker will consult with doctor and take the proper guidance.

1. **Extension points**

None

* + **Check Availability**

1. **Brief Description**

This use case describes how user check availability of doctors.

1. **Flow of events**
2. Basic flow

* This use starts when caretaker wants to consult with doctor.
* Caretaker will check the availability of doctor whether he/she is available or not.
* Caretaker will simple consult with the doctor.

1. Alternative flow

If there is any emergency, and doctor is not available , caretaker can make request to doctor for consultance.

1. **Special Requirements**

None

1. **Pre-conditions**

Login

1. **Post conditions**

If the use case was successful, use will come to know whether doctor is available or not.

1. **Extension points**

None

**Admin**

* **Add/Remove Doctor**

1. **Brief Description**

This use case describes how an actor adds a doctor to the system to provide guidance to the actor.

1. **Flow of events**

a) Basic flow

* This use case starts when the actor needs to add a doctor to the system to provide guidance. The system requires that the actor enter doctor’s name and qualifications.
* The actor enters doctor’s name and qualifications.
* And as a result, doctor is added to the software.

1. **Special Requirements**

None

1. **Pre-conditions**

Doctor have to make request to admin for approval and registering to the system.

Login to the app.

1. **Post conditions**

If the use case was successful, the system generates a unique doctor id for the entered record. If not, the system state is unchanged.

1. **Extension points**

None

* **Payment**

1. **Brief Description**

Caretaker will pay fee to the doctor as consultancy fee.

1. **Actors**

Doctor

Caretaker

Admin

1. **Flow of events**
2. Basic flow

* Doctor will demand payment as a fee from caretaker.
* Caretaker will give doctor’s fee.
* A Recipt of Payment is also sent to admin.

1. **Special Requirements**

None

1. **Pre-conditions**

Login

1. **Post conditions**

None

1. **Extension points**

None

**Design**

1. Admin :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNo | Field Name | Data Type | Constraints | Description |
| 1. | Name | String | - | Gives the name of admin |
| 2. | Id | Alphanumeric | Primary Key | Id of admin |
| 3. | Password | Alphanumeric | - | Used for login |

**Table 4.1** Admin

1. Doctor :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNo | Field Name | DataType | Constraints | Description |
| 1. | Name | String | - | Gives the name of Doctor |
| 2. | Age | Numeric | - | Age of doctor |
| 3. | Gender | String | - | Gender of doctor, either male or female |
| 4. | Addr | Alphanumeric | - | Address of doctor |
| 5. | PhNo | Numeric | - | Contact Info |
| 6. | EmailId | Alphanumeric | - | Gives the mail id to contact doctor |
| 7. | LoginId | Alphanumeric | Primary Key | Unique id of doctor |
| 8. | Password | Alphanumeric | - | Used to login to the software |

**Table 4.2** Doctor

1. Caretaker

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNo | Field Name | Data Type | Constraints | Description |
| 1. | Caretaker’s Name | String | - | Contains Caretaker’s Name |
| 2. | LoginId | Alphanumeric | Primary Key | Unique Id of caretaker(infant) |
| 3. | Password | Alphanumeric | - | Used to Login to the software |
| 4. | PhNo | Numeric | - | Gives the Mobile Number of Caretaker |
| 5. | Relation with Infant | String | - | Gives detail about relation with infant. |
| 6. | Address | Alphanumeric | - | Gives the Address |
| 7. | Infant’s name | String | - | Gives the infants name |
| 8. | Date of Birth | Alphanumeric | - | Gives the Date of Birth(Age) |
| 9. | Gender | String | - | Contains Gender |
| 10. | Height | Alphanumeric | - | Contains Height |
| 11. | Weight | Alphanumeric | - | Contains Weight |
| 12. | Any Allergy | Alphanumeric | - | Gives Allergies |
| 13. | Any Disability | Alphanumeric | - | If child has any kind of disability. |

**Table 4.3** Caretaker

**ESTIMATION AND SCHEDULING**

**5.1 Project Scheduling**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Work**  **Task** | **Planned**  **Start** | **Actual Start** | **Planned**  **complete** | **Actual complete** | **Assigned person** | **Effort**  **Allocated** |
| **PROBLEM STATEMENT** | **Jan W1** | **Jan W1** | **Jan W1** | **Jan W1** | **Aashima,**  **Bhawana** | **2 person**  **Per week** |
| **SOFTWARE MODEL** | **Jan W2** | **Jan W2** | **Jan W2** | **Jan W3** | **Aashima,**  **Bhawana** | **2 person**  **Per week** |
| **PROJECT SCHEDULING** | **Jan W2** | **Jan W2** | **Jan W3** | **Jan W3** | **Aashima,**  **Bhawana** | **2 person per week** |
| **SRS** | **Jan W3** | **Jan W3** | **Feb W1** | **Feb W1** | **Aashima,**  **Bhawana** | **2 person per week** |
| **CONTEXT LEVEL DIAGRAM** | **Feb W1** | **Feb W1** | **Feb W1** | **Feb W2** | **Aashima,**  **Bhawana** | **2 person per week** |
| **DFD LEVEL – 1** | **Feb W2** | **Feb W2** | **Feb W2** | **Feb W3** | **Aashima,**  **Bhawana** | **2 person per week** |
| **DFD LEVEL – 2** | **Feb W3** | **Feb W3** | **Feb W4** | **Mar W1** | **Aashima,**  **Bhawana** | **2 person per week** |
| **DATA DICTIONARY** | **Mar W1** | **Mar W1** | **Mar W1** | **Mar W1** | **Aashima, Bhawana** | **2 person per week** |
| **ER DIAGRAM** | **Mar W1** | **Mar W1** | **Mar W2** | **Mar W2** | **Aashima,**  **Bhawana** | **2 person per week** |
| **USE CASE DIAGRAM** | **Mar W3** | **Mar W3** | **Mar W3** | **Mar W3** | **Aashima** | **1 person per week** |
| **USE CASE DISCRIPTION** | **Mar W4** | **Mar W4** | **Mar W4** | **Mar W4** | **Bhawana** | **1 person per week** |
| **FUNCTION POINT MATRIX** | **Mar W3** | **Mar W3** | **Mar W3** | **Mar W4** | **Aashima** | **1 person per week** |
| **COCOMO MODEL** | **Apr W1** | **Apr W1** | **Apr W1** | **Apr W1** | **Aashima,**  **Bhawana** | **2 person per week** |
| **RISK ANALYSIS** | **Apr W2** | **Apr W2** | **Apr W2** | **Apr W2** | **Aashima,**  **Bhawana** | **2 person per week** |
| **TEST CASES** | **Apr W2** | **Apr W2** | **Apr W2** | **Apr W2** | **Aashima,**  **Bhawana** | **2 person per week** |

**Table 5.1** Project Scheduling

Jan-January W-Week

Feb-February Apr-April

Mar-March

**5.2 Timeline Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Month | January | | | | February | | | | March | | | | April | |  |
| Week | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| PROBLEM STATEMENT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SOFTWARE MODEL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PROJECT SCHEDULING |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SRS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CONTEXT LEVEL DIAGRAM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DFD LEVEL – 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DFD LEVEL – 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DATA DICTIONARY |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ER DIAGRAM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| USE CASE DIAGRAM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| USE CASE DISCRIPTION |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| FUNCTION POINT MATRIX |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| COCOMO MODEL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RISK ANALYSIS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| TEST CASES |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Table 5.2** Timeline Chart

**5.3 Size Estimation**

Project Size Estimation is an essential part of Project. It can be done in many ways:

* Lines of code (LOC)
* Number of entities in Entity-Relationship Diagram
* Total Number of Processes in DFD
* Function Point, etc.

**The size of this project is estimated via Function point or Function-Based Metrics.**

Information Domain Values for **Function-Based Metrics** are as follows:

* **Number of External Inputs (Eis) –** Count of external inputs related to data entering the system. Eachexternal input originates from user and often used to update Internal Logical Files (ILFs).
* **Number of External Outputs (Eos) –** Count of external output related to data exiting the system. Each external output is derived data within the application that provides information to the user. Basically, In this, external output refers to reports, screens, error messages, etc.
* **Number of External Inquiries (Eqs)** – External Inquiries is defined as data retrieval from system but this inquiries don’t change the system. This data is often retrieved from Internal Logical Files (ILFs).
* **Number of Internal Logical Files (ILFs) –** These files resides within the system and is maintained via external inputs. These actually contains data entered by the user, etc.
* **Number of External Interface Files (EIFs) –** These are logical files and resides outside the application but provide information to the application and used by our system.

**Size Estimation for this Project**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| External Inputs | User | Login Module | 1.1Username  1.2Password | | 2 |
| Signup/  Registration  Module | 2.1Caretaker’s Name  2.2Contact Info  2.3Address  2.4Email  2.5Infants Name  2.6Age  2.7DOB  2.8Height  2.9Weight  2.10Gender  2.11Any disability  2.12Any allergy | | 12 |
| Admin | Add/Remove Doctor | 3.1Name  3.2Age  3.3Gender  3.4Email  3.5Specialization  3.6Experience  3.7Language  3.8Contact Info | | 8 |
| Generate Vaccination Chart | 4.1Vaccination Chart | | 1 |
| External Outputs | | User/Caretaker | 1.1My Details  1.2Vaccination Chart  1.3Availability of Doctor | | 3 |
| Doctor | 2.1User’s Details  2.2Vaccination Chart of User | | 2 |
| Admin | 3.1Doctor’s Details  3.2Payment status | | 2 |
| Internal Logical Files | | 1.1User’s File  1.2Doctor  1.3Admin | | | 3 |
| External Logical Files | |  | | | 0 |
| External Inquiries | | Vaccination Chart  Availability of Doctor | | 2 | |
|  | | |

**Table 5.3** Size estimation

**Table 5.4** Function Point Complexity Weights

|  |  |  |  |
| --- | --- | --- | --- |
| Measurement parameter | Weighting factor | | |
|  | Simple | Average | Complex |
| Number of user inputs | 3 | 4 | 6 |
| Number of user outputs | 4 | 5 | 7 |
| Number of user inquiries | 3 | 4 | 6 |
| Number of files | 7 | 10 | 15 |
| Number of external interface | 5 | 7 | 10 |

|  |  |
| --- | --- |
| **Calculating Total Degree of Influence –TDI (∑fi)**  The fi(i= 1 to 14) are *TDI* based on following responses: | |
| 1. | Does the system require reliable backup and recovery? |
| 2. | Does the users data secure ? |
| 3. | How are distributed data and processing functions handled? |
| 4. | Did the user require response time or throughput? |
| 5. | Is Performance critical? |
| 6. | Does the system require online data entry? |
| 7. | Will the system run in existing operational environment? |
| 8. | Are the inputs, outputs, files, or inquiries complex? |
| 9. | Is the internal processing complex? |
| 10. | Is the code designed to be reusable? |
| 11. | Is the conversion and installation complex ? |
| 12. | How effective and/or automated are start-up, back-up, and recovery procedures? |
| 13. | Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations? |
| 14. | Is the application designed to facilitate change and ease of use by the user? |

**Table 5.5** TDI Responses

**Function point ( FP)  = UFP x VAF = Count Total \* (0.65 + (0.01 \*))**

UFP (Count Total) = Sum of all the complexities, given in question

VAF = Value Adjustment Factor i.e. 0.65 + (0.01 \* TDI),   
TDI = Total Degree of Influence of the 14 General System Characteristics.

Hence, To compute function points (FP), the following relationship is used:

**FP= count total \* [0.65 + 0.01 \* ∑(fi)]**

**TOTAL EXTERNAL INUPUTS =** 23

**TOTAL EXTERNAL OUTPUTS =** 7

**TOTAL LOGICAL INTERNAL FILES =** 3

**TOTAL EXTERNAL INQUIRIES =** 2

**TOTAL EXTERNAL INTERFACE FILES =** 0

Assuming all the parameters are of simple complexity,

Count total = {23 \* 3} + {7 \* 4} + {2 \* 3} + {3 \* 7} + {0 \* 5} = 124

**Considering all adjustment factors of simple influence = 14 \* 1 = 14**

FP = 124 \* [0.65 + (0.01 \* 14)]

= 124 \* [0.65 + 0.14]

= 124 \* 0.79

= 97.96

**Function-Point = 98**

**5.4 Cost Estimation (COCOMO II MODEL)**

The original COCOMO model became one of the most widely used and discussed software cost estimation models in the industry. It has evolved into a more comprehensive estimation model, called COCOMO II.

COCOMO II models require sizing information. Three different sizing options are available as part of the model hierarchy:-

* Object Points
* Function Points
* Lines Of Source Code

The COCOMO II application composition model uses object points.

Like function point, the **object point is an indirect software measure** that is computed using counts of the number of

**(1)** **screens** (at the user interface),

**(2)** **reports,**

**(3)** **components** likely to be required to build the application.

Each object instance (e.g., a screen or report) is classified into one of three complexity levels (i.e. ,simple ,medium, or difficult).

Once complexity is determined, the number of screens, reports, and components are weighted according to the table illustrated in Table 5.6 .

|  |  |  |  |
| --- | --- | --- | --- |
| **OBJECT TYPE** | **COMPLEXITIY WEIGHT** | | |
| **SIMPLE** | **MEDIUM** | **DIFFICULT** |
| **SCREEN** | 1 | 2 | 3 |
| **REPORT** | 2 | 5 | 8 |
| **3GL COMPONENT** |  |  | 10 |

**TABLE 5.6** COCOMO II Complexity Weights

The object point count is then determined by multiplying the original number of object instances by the weighting factor in the figure and summing to obtain a total object point count.

When component-based development or general software reuse is to be applied, the percent of reuse (%reuse) is estimated and the object point count is adjusted:

|  |
| --- |
| **NOP = (Object Point) \* [ (100 - %reuse) / 100 ]** |

where NOP = defined as new object points.

To derive an estimate of effort based on the computed NOP value, a **“productivity rate”** must be derived.

|  |
| --- |
| **PROD =** |

Table 5.7 presents the productivity rate for different levels of developer experience and development environment maturity. Once the productivity rate has been determined, an estimate of project effort is computed using

|  |
| --- |
| **ESTIMATED EFFORT =** |

In more advanced COCOMO II models,12 a variety of scale factors, cost drivers, and adjustment procedures are required.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Developer’s experience/capability | Very Low | Low | Normal | High | Very high |
| Environment maturity/capability | Very Low | Low | Normal | High | Very  high |
| PROD | 4 | 7 | 13 | 25 | 50 |

**TABLE 5.7** Productivity Rate For Object Point Counts

**COST ESTIMATION FOR THIS PROJECT**

**(1) SCREENS**

|  |  |
| --- | --- |
| 1. Home Page | 1. Login Page For Doctor |
| 1. Registration for Caretaker | 1. Doctor Profile |
| 1. Login For Caretaker | 1. Generate Payment |
| 1. Caretaker Profile | 1. View Infant record by Doctor |
| 1. Caretaker Update Details | 1. Add Prescription |
| 1. Doctors Availability | 1. Login Page For Admin |
| 1. Update Vaccination Chart | 1. Generate Bill |
| 1. Payment Receipt | 1. Update Doctor Details |
| 1. Payment By Caretaker | 1. Add/Remove Doctor |
| 1. View doctor By Admin. | 1. Add Suggestion for everyone |
| 1. Add Suggestion for everyone |  |

**(2) REPORTS**

1. Registered Successfully.
2. Details Successfully Updated.
3. Vaccination Chart Updated.
4. Consultation Successful.
5. Payment Successfully Made.
6. Refund Of Payment Successfully Made.
7. Doctor Added Successfully.
8. Doctor Removed Successfully.

**(3) 3 GL MODULES**

None

**TOTAL SCREENS =** 21

**TOTAL 3GL MODULES =**0

**TOTAL REPORTS =** 7

**CONSIDERING ALL OF THE ABOVE HAVE SIMPLE COMPEXITY, 0% OF COMPONENTS ARE REUSED AND TAKING THE DEVELOPER EXPERIENCE AND ENVIRONMENT MATURITY AS NORMAL.**

**PRODUCTIVITY RATE =**  = 7.

**OBJECT POINT** = {21 \* 2} + (3\*5) = 57

**ESTIMATED EFFORT =**  = = 8.14 **12 Person-Months.**

**RISK ANALYSIS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SNO. | RISK | Category | Probability  ( P ) | Impact  ( I ) | Exposure Risk  ( E = P\*I ) | RMMM Plan |
| 1. | Some team members leave the project In between | Technical Risk | 20% | 2 | 0.4 | Use Back Up staff members who know about the project |
| 2. | Delivery Deadline Tightened | Project Risk | 20% | 2 | 0.4 | Use Extra Staff members to meet the deadline |
| 3. | Less use of project than planned | Project Risk | 30% | 3 | 0.9 | Update the project as per the requirements |
| 4. | Losing all project data may be due to system crash or hard disk failure | Project Risk | 20% | 4 | 0.8 | Back Up the Project Online in every system and also on Internet using full security |
| 5. | Change in rules for vaccination  As per by Government | Programmatic Risk | 50% | 3 | 1.5 | Update the project as per the rules |

**Table 5.8** Risk Analysis

**RMMM plan for the project in detail (Risk Mitigation Monitoring and Management)**

**Mitigation**

The cost of the project would rise too much if the requirements are changed after the subsequent have commenced. To mitigate the risk we can make the constraint as put a forward deadline for proposing the changes.

**Monitoring**

While working on SRS, we should conduct multiple reviews to make sure that the requirements are well understood and not have to change later.

**Management**

In case, if there is no other option except make changes in the SRS, the development team must cease their work until changes are fixed.

**HOME PAGE**

**SAMPLE SCREENSHOTS**



DOCTOR LOGIN

ADMIN LOGIN

CARETAKER LOGIN

VACCINATION ALERT

**Figure 7.1** Home PAGE

**CARETAKER LOGIN PAGE** 

**Username :**

**LOGIN**

**Password:**

**New User? Register**

CARETAKER LOGIN

**Figure 7.2** Login Page

**CARETAKER REGISTRATION FORM**

**Figure 7.3** Registration Page

Enter Infants details

**Password :**

CARETAKER REGISTRATION FORM



Submit

INFANTS DETAILS

**Any disability :  : :**

**Any allergy :**

**Weight :  : :**

**Height : :**

**Gender : :**

**DOB :**

**NAME :**

**Mobile No. :**

**Address :**

**a**

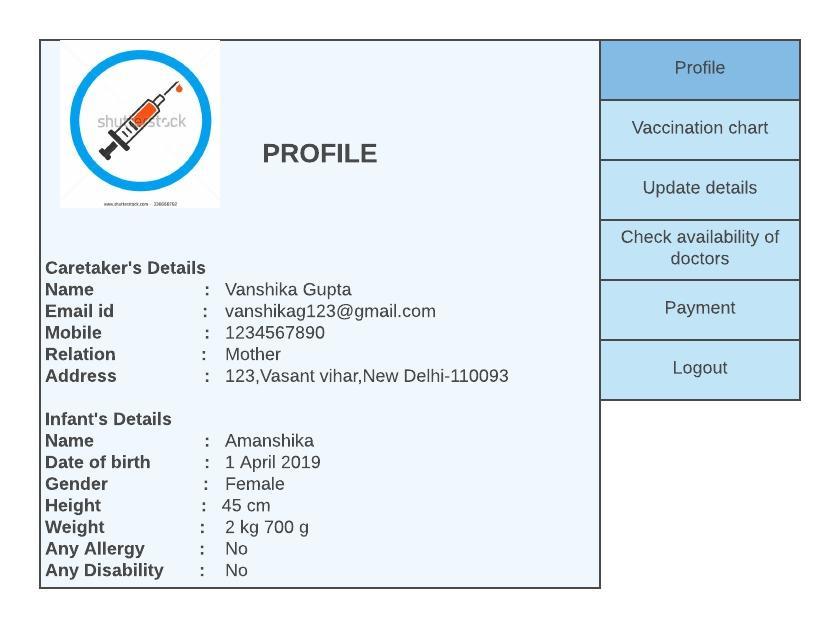
**Relation :**

**Email id :**

**NAME :**

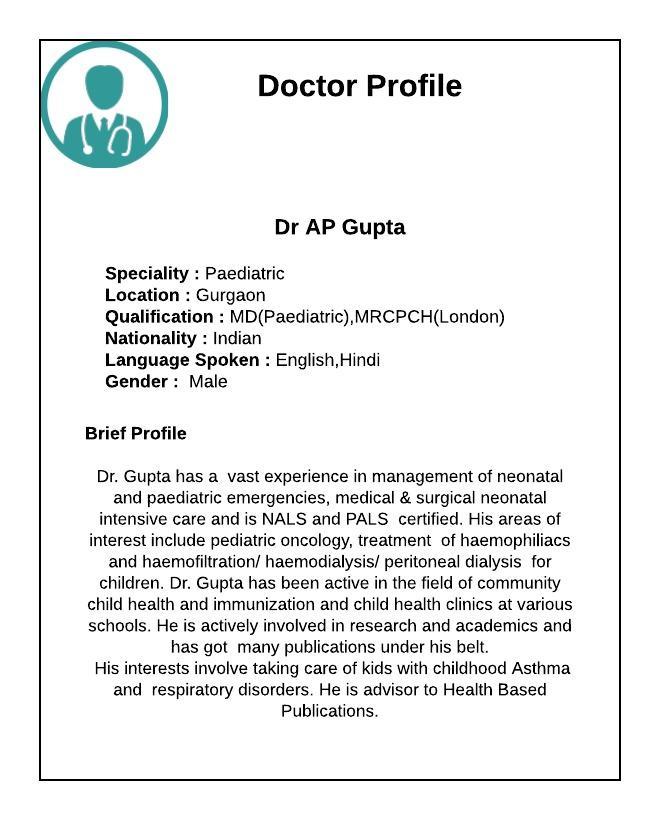
**Figure 7.4** Infant’s Details

**Caretaker’s Profile**

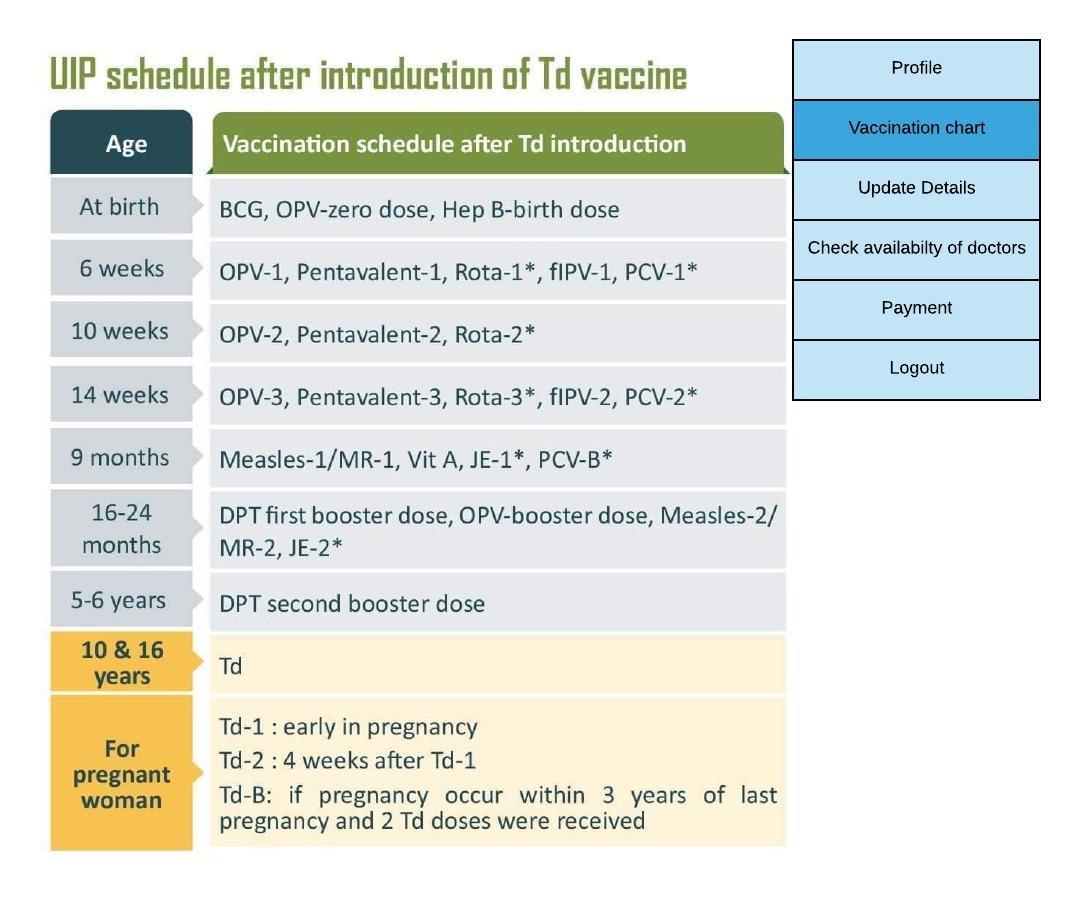


**Figure 7.5** Caretaker’s Profile

**Doctor’sProfile:**



**Figure 7.6** Doctor Profile



**Figure 7.7** Vaccination chart



Submit

Name :

Weight :

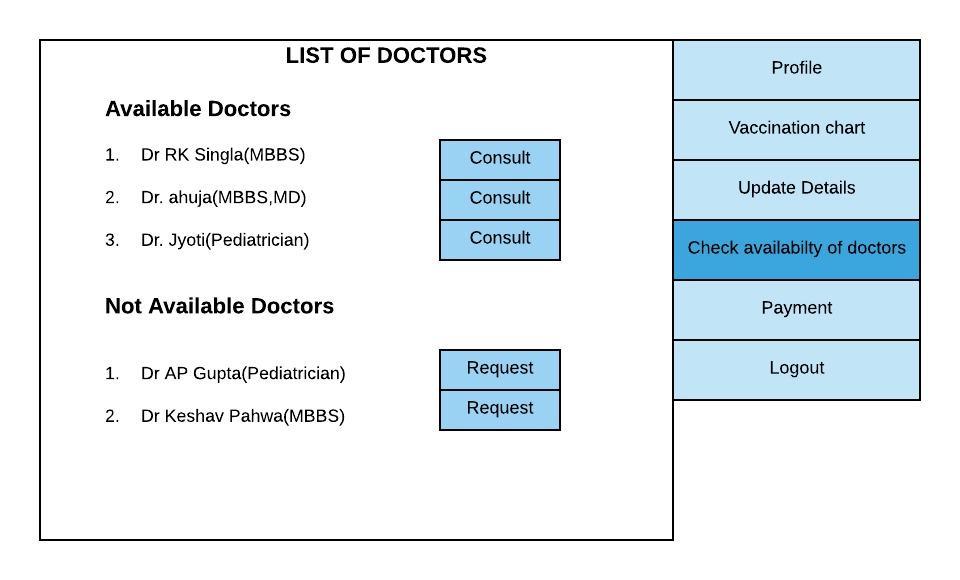
Height :

Any allergy :

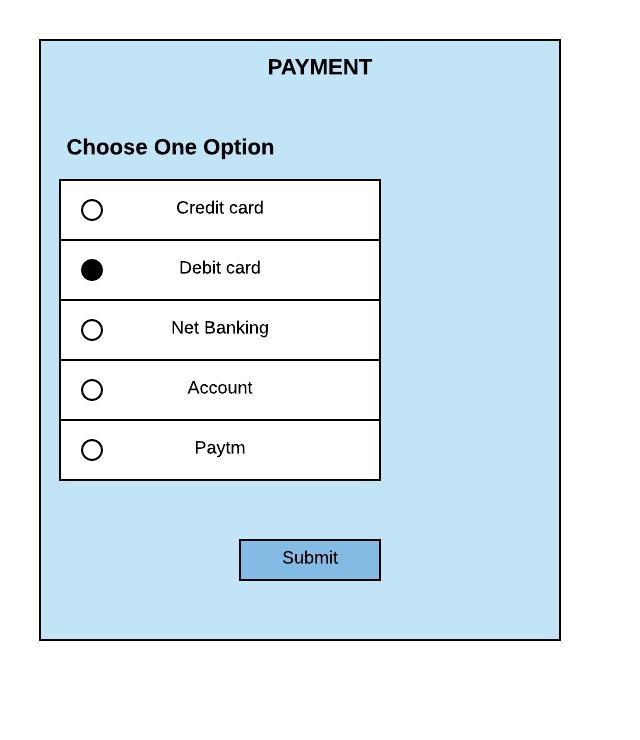
Any allergy :

Update Details

**Figure 7.8** Update details



**Figure 7.9** Availability of Doctors



**Figure 7.10** Payment



**RESULT AND CONCLUSION:**

**Vaccination Alert** application is designed to protect young children before they are likely to be exposed to potentially serious diseases and when they are most vulnerable to serious infection. **Creation of awareness about vaccination increases the rate of vaccination and thus prevents great reduction of vaccine preventable diseases.** It is an useful android application which can help a lot of rural people. The use of this application helps parents not to memorize the list of vaccinations to be given to their child. It also has an user-friendly interface and self-explanatory .The user of this application will not miss any of the vaccines and hence prevents the child from suffering any serious diseases in the future.

Apart from the vaccination notification, it allows the users to check the child’s growth(like height, weight) rather than visiting the hospital every week or month. It reduces the time of parents to search and visit the hospitals for vaccinating their child in case of any emergency.