1. **ACKNOWLEDGEMENT**

Every work that one completes successfully sends on the constant encouragement, good will and support of the people around. I, hereby, avail this opportunity of express my heartfelt gratitude to a number of people who extended their valuable time, full support and co operation and developing this project.

It gives me great pleasure in submitting this project entitled **“Being Mother”,** developed at myself.

I convey my heartfelt gratitude to my college, ―**Shantilal shah Engineering Collage”** and **“Being Mother”** for giving me this precious opportunity to work in the industry for the real time project.

I am highly indebted to **Dr. M.G.Bhatt , Principal, Shantilal Shah Engineering Collage** who had been constant source of inspiration and drive throughout the object. He has escorted me personally. I am greatly benefited by his guidance, suggestions and encouragement. We must make special mention of **Being Mother**, our project in charge for their co-operation & assistance in solving technical problem.

I owe the success of the project to my Project Guide, **Miss Disha Sanghani,** who was a tremendous supporter and an eager teacher, for providing excellent guidance for this project. She is one of the major sources behind the success of the project. We are wholeheartedly thankful to her for giving us her value able time & attention & for providing us a systematic way for completing our project in time.

We are also very thankful to respective principal M.G.Bhatt sir and our professors who gave us an opportunity to present this project.

2. **ABSTRACT**

* Complete user privilege management authentication and authorization.
* Sign in, sign up and forgot password utilities for information posted and end users as well.
* Different categories and sub categories of contents available in the application and those contents can be viewed and categories will be…..

* Lady’s health during pregnancy
* Lady’s health after pregnancy
* Child’s health record
* Child’s growth meter
* Food
* Vaccine
* Medical care
* Exercise
* Chat between Users
* Feedback
* For any category there will be features or information like…
* Photos
* Videos
* Details
* Section for the related news and related list as well
* Proper email notifications about different things.
* Mobile notifications about vaccine and other.
* Admin side management and admin can manage
* Photos
* Videos
* Can make any user or information activated or deactivated
* Can control information of the application
* Proper data syncing will be there
* All settings aspects
* Informational filtering
* Dynamic contact us – for contacting the admin
* Report abuse features
* Different forms for information posting by users and that information will be reviewed by admin and then can be published
* Privacy policy
* Terms of usage
* Complete unit and integration
* Deployment of complete application

3. **PROJECT PROFILE**

**Project Title**   **:** Being Mother

**Project Type** **:** Web Based Application

**Objective**  **:** To Develop a Web Based Application

**Technology**  **:** Android

**Purpose**   **:** B.E.IT, (7th Semester)

**Developed By**  **:** Devani Reshmakumari J. (120430116024)

Gadani Nipa C. (120430116062)

**Project Aim:**

Being Mother is designed and develop to provide better knowledge to the pregnant lady or the mother about the pregnancy and the baby, she can aware of all information and needs related to pregnancy and a new born baby like food, medical needs, vaccine, for baby etc.

**Project Definition:**

Being Mother is an android application, main aim of which is to provide better knowledge about pregnancy and new born baby to mother and to aware the mother about the basic needs of baby and pregnant lady.

**Project Scope:**

Software scope describes the function and features that are to be delivered to end-user. This project provide details like daily news, mother and baby food and health details, new updates about food, vaccine, health information. System have administration authentication, admin can change the module, update modules.

**Being Mother Modules Details**

Problem Decomposition Elaboration with Justification (Module wise).

1. **Home Screen Module**

* Mother
* Health
* Food
* Calendar
* Tools
* Baby
* Health
* Food
* Calendar
* Tools
* Profile
* Chat
* Upload Photos
* More
* Setting
* FAQ
* Feedback
* About Us

1. **Admin Module**

* Mother
* Upload Videos
* Create Photo Gallery
* Update Photo Gallery
* Edit Information Act
* Baby
* Upload Videos
* Create Photo Gallery
* Update Photo Gallery
* Edit Information Act
* Account
* Change Root Password
* Change First Level Password
* Logout
* Home Page Module deals with all the Details of All Screen related activity like Mother Module, Baby Module, Profile Module, Account Module etc.
* Admin is head of the system which basically deals with all our modules.

* The Home Screen contains several information about the application like Mother’s Health, food, and other information, Baby’s health, food, and other information and Profile etc.

4. **PROJECT MANAGEMENT**

**4.1 Project Planning and Scheduling:**

Projects planning and Scheduling donned through like first gathering all data and requirement of the project and then making planning for how to describe that data and how to manage that data. All required data is gathered from institute like mother’s health information, baby’s health information, food information, vaccine information. Then planning is done how to describe and manage that all data which provide more benefits to all pregnant lady, and mother or father.

**4.2 Time Line Chart**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **July &**  **August** | | | | **September & October** | | | | **November & December** | | | | **January &**  **February** | | | | **March & April** | |
| **Domain Understanding** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Analysis** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Learning**  **Process** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Design** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| |  | | --- | | **Coding Testing** | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Documentation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Final**  **Documentation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Fig 4.1 Time Line Chart

**4.3 Risk Management:**

**Technology Risk:**

The database used in the system cannot process as many transactions per second as expected. Software components that should be reused contain defect that limit their functionality.

**People Risk:**

Pregnant lady and Baby’s parents have to ready for new changes their response towards to the system have to positive.

**Requirements Risk:**

Changes to requirement that require major design rework are proposed. Customers fail to understand the impact of requirement change.

**Estimation Risk:**

The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

**4.4 Process Model: The Increment Model**

We have completed our project by increments as listed below:

**Core Application**:

The basic functionalities we have approached are to analysis our project.

**Increment1**:

We have implemented form designing modules as our first increment.

**Increment2**:

Then we have expanded the user requirements to the query level by allowing him/her to fire queries on current system.

**Increment3**:

Finally, we were toughly concentrating for the testing and validation in modules, we succeeded at the end by imposing them on**.**



Fig 4.2 Incremental model

The Incremental model encompasses the following activities:

1. **System/information engineering and modeling**

System engineering and analysis encompass requirements gathering at the system level with a small amount of top-level design and analysis. Information engineering encompasses requirements gathering at the strategic business level and at the business area level.

1. **Software requirements analysis**

The requirements gathering process is intensified and focused specifically on software. Software requirement analysis encompasses understanding the information domain for the software as well as required function, behavior, performance and interfacing. Requirements for both the system and the software are documented and reviewed with the customer.

1. **Design**

Software design is actually a multi-step process that focuses on four distinct attributes of a program: data structure, software architecture, interfaces representation and procedural detail. The design process translates requirements into a representation of the software that can be accessed for quality before coding begins.

1. **Code generation**

Code-generation phase translates the design into a machine-readable form.

1. **Testing**

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals; that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

1. **Maintenance**

Software maintenance applies to following phases in the existing program change in software due to errors. Change in software because the software must be adapted to accommodate changes in its external environment. Change in software when the customer requires functional or performance enhancements.

5. **SYSTEM REQUIREMENT STUDY**

**5.1 USER CHARACTERISCS:**

There are three users that dealing with the system:

1. Admin

2. Mother & Father

This is user which may be technical or non-technical by nature. He/she may not be known to surfing Application and downloading etc. or may be highly sophisticated software developer too. We are to concentrate about those of non-technical ones. We are providing GUI clicks on links or buttons would make their task easier and checkbox and image-buttons which would specify the purpose of it, itself. Admin and mother & father user is responsible for maintaining database and updating in system.

**5.2 Hardware and Software Requirement:**

**1. Platform : JAVA**

**2. Client OS : Android**

**3. Client Authoring : XML, JAVA**

**4. Code Behind : Android, Java, XML**

**5. Web Server : PdaNet**

6. **SYSTEM ANALYSIS**

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of an interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through the various processing that the inputs phase through in the organization.

A detailed study of these processes must be made by various techniques like Interviews, Questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now, the existing system is subjected to close study and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as a proposal. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

**6.1 Why is System Analysis Necessary?**

During system analysis some activities are carried out, such as problem definition, design, and implementation.

1. What is the problem?
2. How to solve it?
3. What are the technical factors required?
4. What are its features and limitations?

If a system is designed without considering above activities then the system will not work properly to the satisfaction of user requirement.

So system analysis is necessary to design perfect and accurate system, which satisfies the user requirement.

**6.2 Problems & Weaknesses of The Current System:**

The existing application is dynamic. Over application has provide to automatic ringing vaccine alarm but other application has not provide this system.

**6.3 Requirement of New System:**

In order to make the application dynamic and more interactive it‘s provide more better GUI. We have tried to include a database link to our application. Hence the recruiters have been provided with the facility to post their eligibility criteria. Provision has also been made to display the latest events and announcements associated with online.

**6.4 Feasibility Study:**

**Operation Feasibility:**

The application is having a high degree of operational feasibility. It is also very user –friendly and even any non-technical people go though. It is providing links & menus such a ways that anyone can easily get for what one is searching for by a few clicks. It is having better adaptability & understandability.

**Technical Feasibility:**

The application provides user-friendly GUI and facility. Use language Android. The most secure and trusted back end SQL with its advance features.

**Schedule Feasibility:**

This project has stick to its regular schedule of time v/s work to do. In order to provide quality software built, we have planned to complete project‘s schedule that makes completion of final stage earlier then demanded. This is so because we want to add additional facility or correct errors if any remains in the application at the end after being used by user and thus we can achieve robust application.

**Economic Feasibility:**

Application need the computer to have high configuration as they provide so many facilities, they consume more main memory, etc. but, both the high cost factor because of font end and back end of our project has most have effectiveness features and thus they are how they are told. Front end android provide many special feature such as platform independence, compatibility with many languages, easy to built large application, etc. while SQL server is of course famous for its numerous features another user-friendly facilities.

**6.5 What is Android?**

Android is the mobile operating system developed by Google. Essentially, Google produces the software that runs almost every other mobile phone besides Apple’s iPhone. There are also some popular Android tablets as well. Android is a linax-based software system, and similar to Linux, is free and open source software. This means that other companies can use the Android operating system developed by Google and use it in their mobile devices (more on that later). The distinguishing factor of this brand is a kernal. Android hosts a central core, which essentially is a strip code that helps the software operate.

[Operating Systems](http://www.engineersgarage.com/articles/operating-systems-tutorial) have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

One of the most widely used mobile OS these days is ANDROID. Android does a software bunch comprise not only operating system but also middleware and key applications. Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android.

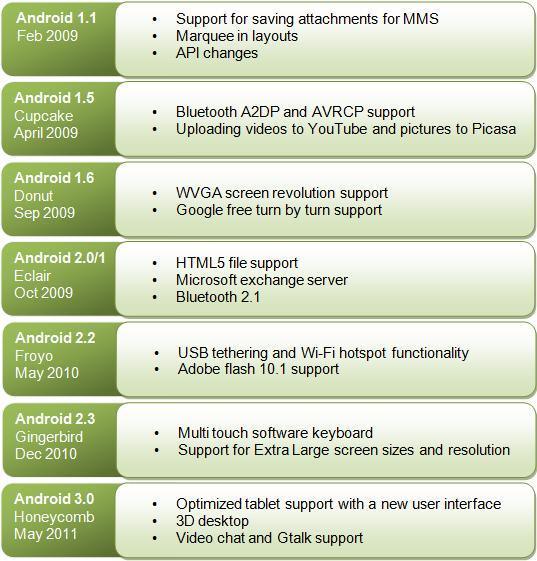


Fig. 6.5 Android Version

* + 1. **Features & Specifications**

**Android** is a powerful Operating System supporting a large number of applications in [Smart Phones](http://www.engineersgarage.com/articles/smart-phones). These applications make life more comfortable and advanced for the users. Hardware that supports Android is mainly based on [ARM architecture](http://www.engineersgarage.com/articles/arm-advanced-risc-machines-processors) platform. Some of the current features and specifications of android are:

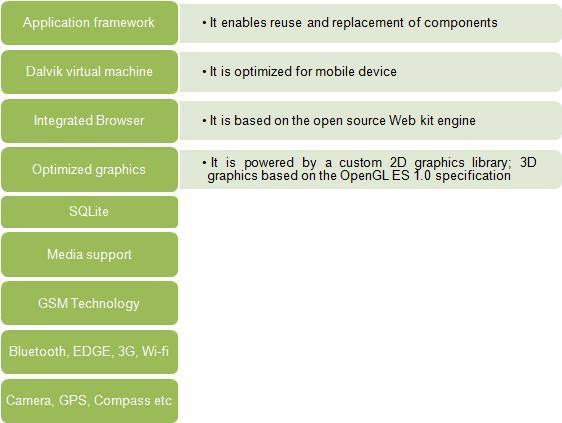


Fig. 6.5.1 Android Uses

Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides **Android SDK** (Software development kit).

**6.5.2. Applications**

These are the basics of Android applications:

•      Android applications are composed of one or more application components (activities, services, content providers, and broadcast receivers)

•      Each component performs a different role in the overall application behavior, and each one can be activated individually (even by other applications)

•      The manifest file must declare all components in the application and should also declare all application requirements, such as the minimum version of Android required and any hardware configurations required

•      Non-code application resources (images, strings, layout files, etc.) should include alternatives for different device configurations (such as different strings for different languages)

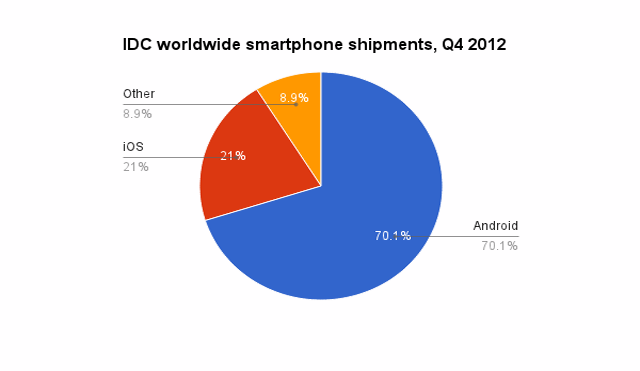


Fig. 6.5.2 Android User Chart

**7. SYSTEM DESIGN**

System design is the solution to the creation of a new system. This phase is composed of several systems. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design.

During logical design phase the analyst describes inputs (sources), out puts (destinations), databases (data sores) and procedures (data flows) all in a format that meats the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design.

The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

Specification must reflect the application to be handled by the system and include system objective, flowchart, input-output requirement, file structure and cost.

1. **Reliability:**

Software reliability brings up the concept of modularity. A package with a high degree of modularity has the capacity to operate in many machine configurations and perhaps across manufacture product lines.

This software with high potential can work or can be operated in based configuration computers.

1. **Functionality:**

It is a definition of the facilities, performance and other factors that the user requires in the finished product.

This software provides the facility of having the complete infrastructure of the house, etc. The performance of the software includes the flexibility of the software.

1. **Flexibility:**

One feature of the flexibility is adaptability, which is of the ease of extending the product. In the field of flexibility, the software covers all factors to be considered.

1. **Usability:**

The points to be considered while thinking of usability are portability and understandability.

While designing this software, we always try to make it more portable, should not consume more space as well. As we are using HTML a front-end, which is very user friendly, so it scores full marks in understanding.

1. **Security:**

Security control and access of software or data file should be maintained in this software.

1. **Performance:**

The language, in which the package is developed, should be well known or say should be aware to the programmers. The operating system may also affect its performance. These both factors are important for consideration. This software should be economical for buying and should be user friendly.

In our project, we have considered all above factors and developed such a project hat can run in all environments but may be supported by any API levels.

**7.1 Entity Relationship Diagram (ER Diagram)**

* **Entity Relationship Analysis**

There are three major abstractions to describe data.

|  |  |
| --- | --- |
| **Entities** | **: They are distinct things in the system.** |
| **Attributes** | **: They are meaningful instruction between objects.** |
| **Relationship** | **: They are properties of entities and relationships.** |

Entity-relationship is described by their dependence on each Other as well as by extent of relationship.

**Entity Dependency:**

Two types of dependency are common, in the first existence dependency; one entity is unable to exit in the database unless the other is first present. The existence of the second depends on the existence of the first.

**External of Dependency:**

The extent of dependency includes two interrelated concerns. The direction of the relationship and the type of association between them both can be represented graphically.

**7.1.1 Entity Relationship Diagram**

E-Mail

Password

Password

Date of Birth

BEING MOTHER

Password

User Name

Admin Login

Admin

Age

ID

User Name

Baby Name

ID

User ID

Last Name

First Name

Running Month

Name

E-Mail

Pregnant lady

Baby

User

Baby’s Date of Birth

Fig 7.1.1: E-R Diagram

**7.2 Data Flow Diagram**

The Data flow diagram is one of the most important modeling tools used by system analysts. The use of data flow diagram as modeling tools was populated by Demarco (1978) and Gane Sarson (1979) through their structured System analysis methodologies.

They suggested that a data flow diagram should be first tool used by the system analysts to model system components. These components are the system processes, the data used by the used by system analysts to model system these processes.

**Data Flow Diagram Symbols:**

**Data Flow:**

Data move in a specific direction from an origin to a destination in the form of documents, letter, telephone call, or virtually any other medium. The data flow is a ―packet of data.

**Processes:**

In this, people, procedures, or devices that use or produce (transform) data. The physical components are not identified.

**Source or Destination:**

External Sources or destination of data, which may be people, programs, organization or other entities interact with system but data are outside its boundary.

**Data Store:**

Data are stored or referenced by a process in the system. The data store may represent computerized or non-computerized devices. Each component in a data flow diagram is labeled with a descriptive name. Process names are further identified with a number that will be used for identification purpose. The number assign to a specific process does not represent the sequence of process. It is strictly used for identification and will take on added value to the components that make up a specific process.

**7.2.1: Data Flow Diagram**

Baby

Admin

Display

Information

Display informati Display

Information

User

Commands and data

Ask any Que.

&

Information

Pregnant lady

Provide Provide

Information Information

Baby related Provide

Information Information

Fig 7.2.1: Data Flow Diagram

**7.2.2: Data Flow Diagram of Pregnant lady**

Displaying screen

Interact with display

Pregnant Lady

Display information

Display

Select task

Displaying screen

Select document

Select task Provide

Document

Process downloading document

Information

Process downloading document

Provide information

Fig 7.2.2: Data Flow Diagram of Pregnant lady

**7.2.3: Data Flow Diagram of Baby**

Displaying screen

Interact with display

Baby

Display information

Display

Select task

Displaying screen

Select document

Select task Provide

Document

Process downloading document

Information

Process downloading document

Provide information

Fig 7.2.3: Data Flow Diagram of Baby

**7.3 Class Diagram**

The class model represents the static, structural data aspects of a system. The class model describes the structure of objects in a system- their relationships to other objects, their attributes, and their operations. The class model provides context for the state and interaction models.

**Objects:**

The purpose of class model is to describe objects. An object, abstract or thing with identify that has meaning for an application. Some objects have real world counterparts, while others are conceptual.

**Classes:**

A class describes a group of objects with the same properties, behavior, kinds of relationship and semantics. The choice of classes depends on the nature and is a matter of judgment. Objects grouped in class each objects knows its class.

**Operations and methods:**

An operation is a function or procedure that may be applied to or by objects in a class. Each operation has target objects as implicit arguments. The behavior of the operation depends on the class of its target. An object knows its class, and hence the right implementation of the operation.

**Links and association:**

A link is a physical or conceptual connection among objects. Most links relate two objects, but some links relate three or more objects. A link is an instance of an association.

An association is a description of a group of links with common structure and common semantics. An association describes a set of potential link in the same way that a class describes a set of potential objects. Links and association often appear as verbs in a problem statement.

**Generalization:**

Generalization is the relationship between a class (the superclass) and one or more variation of class (the subclass). Generalization organizes classes by their similarities and differences structuring description of objects.

**7.3.1: Home Screen Class Diagram with Class Name and Objects**

HH

View

Setting

FAQ

Feedback

About Us

**More**

View

Chat

Upload Photos

**Profile**

View

Health

Food

Calendar

Tools

**Baby**

View

Health

Food

Calendar

Tools

**Pregnant lady**

View

Mother

Baby

Profile

More

**Home**

**Admin Login**

User ID

Password

Login

Fig 7.3.1: Home Screen Class Diagram with Class Name and Objects

**7.3.2: Admin Panel Class Diagram with Class Name and Objects**

**Home**

**Photo Gallery**

**Info. Of User**

User ID

Password

User details

File

Path

InsertPhotoGallery

UpdatePhotoGallery

CreateAdminAccount

UpdateAdminAccount

View

**Information**

Food

Health

Other

Insert Information

Update Information

Fig 7.3.2: Admin Panel Class Diagram with Class Name and Objects

**7.3.3: Class Diagram with Relationship and Association**

Home Screen

Baby Home

Pregnant lady Home

Admin Login

Info. Of User

Photo Gallery

Information

Fig 7.3.3: Class Diagram with Relationship and Association

**7.4 State Diagrams:**

State diagrams express the state model. Each state diagrams show the state and events sequences permitted in a system for one class of objects. State diagrams refer to the other models. Actions and events in a state diagram become operations on objects in a class model. References between state diagrams become interaction in the interaction model.

**Events:**

An event is an occurrence at a point in time, such as user depress left button or flight departs for Chicago. Events often correspond to verbs in the past tense or onset of some condition. By definition, an event happens instantaneously with regards to the time scale of an application. Once event may logically precede or follow another or two events may be unrelated.

**States:**

A state is an abstraction of the values and links of objects. Sets of values and links are grouped together into a state according to the gross behavior of objects.

**Transitions and Conditions:**

A transition is an instantaneous change from one state to another. For example, when a called phone is answered, the phone line transition from the ringing state to the connected state. The transition is said to free upon to change from the source state to the target state. The origin and target of a transition usually are different states, but may be the same. A transition fires, when its event occurs. The choice of next state depends on both original state and the event received. An event may cause multiple objects to transitions; from a conceptual point of view such transition occurs concurrently.

**7.5 Iteration Diagram**

The interaction model is the third leg of the modeling tripod and describes interaction within a system. The class model describes the objects in a system and their relationship, the state model describes the life cycle of the objects, and the interaction model describes how the objects interact.

The interaction model describes how objects interact to produce useful results. It is a holistic view of behavior across many objects, whereas the state model is a reductionist view of behavior that examines each object individually. Both the state model and the interaction model are needed to describe behavior fully. They complement each other by viewing behavior from two different perspectives.

**Use case Diagram:**

Use cases describe how system interacts with outside actors. Each use case represents a piece of functionality that a system provides to its user. Use cases are helpful for capturing informal requirements.

**7.5.1 Use Case Diagram**

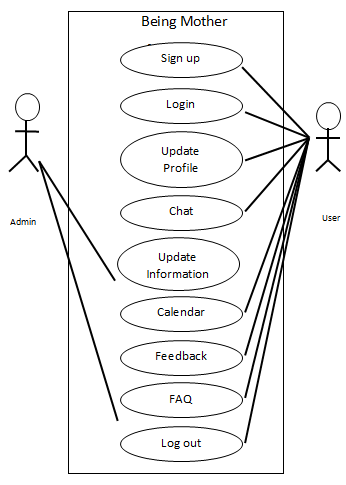


Fig 7.5.1: Use Case Diagram

**Sequence diagram:**

Sequence diagram provides more detail and show the message exchanged among a set of objects over time. Messages include both asynchronous signals and procedures calls. Sequence diagrams are good for showing the behavior sequence seen by user of system.

**7.5.2 Sequence Diagram**

: User

: System

: Admin

Log in

Display Info. Of Application

Update Profile

Display the Notification

Feedback

FAQ FAQ

Display to the FAQ Info.

Log out Update Info.

Fig 7.5.2: Sequence Diagram

**Activity Diagram:**

Activity diagrams provide further detail and show the flow of control among the steps of a computation. Activity diagrams can show data flow as well as control flows. Activity diagram document the steps necessary to implement an operation or a business process referenced in a sequence diagram.

**7.5.3 Activity Diagram**

Login

Wrong ID and Password

Yes

More

Profile

Baby

Mother

Calander

Food

Feedback

FAQ

Health

Tools

Setting

About Us

Calander

Food

Tools

Health

Upload Photos

Chat

Log Out

Fig 7.5.3 Activity Diagram

**7.6 TABLE AND RELATIONSHIP**

**7.6.1 PREGNANT LADY SINGUP TABLE PRIMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| USER ID | VARCHAR(30) | Not Null |
| FIRST NAME | VARCHAR(50) | Not Null |
| LAST NAME | VARCHAR(50) | Not Null |
| USER NAME | VARCHAR(50) | Not Null |
| PASSWORD | VARCHAR(100) | Not Null |
| CONFIRM PASSWORD | VARCHAR(100) | Not Null |
| DATE OF BIRTH | INT | Not Null |
| RUNNING MONTH | INT | Not Null |
| E-MAIL | VARCHAR(100) | Not Null |

**7.6.2 BABY SINGUP TABLE PRIMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| USER ID | VARCHAR(30) | Not Null |
| USER NAME | VARCHAR(50) | Not Null |
| BABY NAME | VARCHAR(50) | Not Null |
| PASSWORD | VARCHAR(50) | Not Null |
| CONFIRM PASSWORD | VARCHAR(100) | Not Null |
| BABY’S DATE OF BIRTH | INT | Not Null |
| AGE | INT | Not Null |
| E-MAIL | VARCHAR(100) | Not Null |

**7.6.3 CATEGORY TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(30) | Not Null |
| TITLE | VARCHAR(50) | Not Null |
| DETAILS | VARCHAR(150) | Not Null |

**7.6.4 SUB CATEGORY TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(30) | Not Null |
| TITLE | VARCHAR(50) | Not Null |
| DETAILS | VARCHAR(150) | Not Null |

**7.6.5 PROFILE TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| NAME | VARCHAR(30) | Not Null |
| USER ID | VARCHAR(50) | Not Null |
| AGE | INT | Not Null |
| TYPE OF USER | VARCHAR(50) | Not Null |
| GENDER | VARCHAR(30) | Not Null |
| DATE OF BIRTH | INT | Not Null |
| BORN TIME OF BABY | TIME | Not Null |
| BABY NAME | VARCHAR(50) | Not Null |
| BABY HEIGHT | INT | Not Null |
| BABY WEIGHT | INT | Not Null |

**7.6.6 UPLOAD PHOTOS TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(50) | Not Null |
| USER ID | VARCHAR(50) | Not Null |
| TITLE | VARCHAR(100) | Not Null |
| DETAILS | VARCHAR(150) | Not Null |

**7.6.7 LIKE TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(50) | Not Null |
| PHOTOS ID | VARCHAR(50) | Not Null |
| USER ID | VARCHAR(50) | Not Null |

**7.6.8 FEEDBACK TABL PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(50) | Not Null |
| USER ID | VARCHAR(50) | Not Null |
| TYPE OF FEEDBACK | VARCHAR(80) | Not Null |
| MESSAGE | VARCHAR(500) | Not Null |

**7.6.9 VACCINE TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(50) | Not Null |
| USER ID | VARCHAR(50) | Not Null |
| VACCINE NAME | VARCHAR(80) | Not Null |
| VACCINE DATE | DATE | Not Null |

**7.6.10 VIDEO TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(50) | Not Null |
| NAME | VARCHAR(30) | Not Null |
| DETAILS | VARCHAR(80) | Not Null |

**7.6.11 FAQ TABLE PRYMARY KEY: ID**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Constraint** |
| ID | VARCHAR(50) | Not Null |
| USER ID | VARCHAR(30) | Not Null |
| ANSWER | VARCHAR(500) | Not Null |

**8.** **TESTING**

**Introduction:**

Testing is a process used to help identify the correctness, completeness and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software.

The quality of the application can and normally does vary widely from system to system but some of the common quality attributes include reliability, stability, portability, maintainability and usability. Refer to the ISO standard ISO 9126 for a more complete list of attributes and criteria.

Testing helps is verifying and Validating if the Software is working as it is intended to be working. Thins involves using Static and Dynamic methodologies to Test the application.

Because of the fallibility of its human designers and its own abstract, complex nature, software development must be accompanied by quality assurance activities. It is not unusual for developers to spend 40% of the total project time on testing. For life-critical software (e.g. flight control, reactor monitoring), testing can cost 3 to 5 times as much as all other activities combined. The destructive nature of testing requires that the developer discard preconceived notions of the correctness of his/her developed software. Testing objectives includes:

1) Testing is a process of executing a program with the intent of finding an error. 2) A good test case is one that has a high probability of finding an as yet undiscovered error.

3) A successful test is one that uncovers an as yet undiscovered error.

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications. The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

**White Box Testing**

The purpose of any security testing method is to ensure the robustness of a system in the face of malicious attacks or regular software failures. White box testing is performed based on the knowledge of how the system is implemented.

White box testing includes analyzing data flow, control flow, information flow, coding practices, and exception and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities.

White box testing requires access to the source code. Though white box testing can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing during the unit testing phase.

White box testing requires knowing what makes software secure or insecure, how to think like an attacker, and how to use different testing tools and techniques. The first step in white box testing is to comprehend and analyze source code, so knowing what makes software secure is a fundamental requirement. Second, to create tests that exploit software, a tester must think like an attacker. Third, to perform testing effectively, testers need to know the different tools and techniques available for white box testing. The three requirements do not work in isolation, but together.

**Black Box Testing**

Black box testing is based on the software‘s specifications or requirements, without reference to its internal workings. Gray box testing combines white box techniques with black box input testing. This method of testing explores paths that are directly accessible from user inputs or external interfaces to the software. In a typical case, white box analysis is used to find vulnerable areas, and black box testing is then used to develop working attacks against these areas. The use of gray box techniques combines both white box and black box testing methods in a powerful way.

**Unit Testing**

Unit test case design begins after the high level design is approved by a Peer Technical Review. The unit test cases shall be designed to test the validity of the program's correctness. White box testing will be used to test the modules and procedures that support the modules. The white box testing technique *ignores* the function of the program under test and will focus only on its code and the structure of that code. To accomplish this, a Statement and Condition technique shall be used. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once:

1. Each decision statement in the program shall take on a true value and a false value at least once during testing.

2. Each condition shall take on each possible outcome at least once during testing.

3. Each program shall be tested for complexity based on McCabe's measurement.

**Test Strategy:**

The first step in planning white box testing is to develop a test strategy based on risk analysis. The purpose of a test strategy is to clarify the major activities involved, key decisions made, and challenges faced in the testing effort. This includes identifying testing scope, testing techniques, coverage metrics, test environment, and test staff skill requirements. The test strategy must account for the fact that time and budget constraints prohibit testing every component of a software system and should balance test effectiveness with test efficiency based on risks to the system. The level of effectiveness necessary depends on the use of software and its consequence of failure. The higher the cost of failure for software, the more sophisticated and rigorous a testing approach must be to ensure effectiveness. Risk analysis provides the right context and information to derive a test strategy.

**9.** **SCREEN SHORT**

**10.** **LIMITATION & FUTURE ENHANCEMENT**

**Limitation**

This application the limitation is the user can’t contact any doctors directly. In this application only the mother, father or pregnant lady can make account and use day to day calendar and other services.

**Future enhancement**

Future enhancement means new updating can done later in application, new requirement can be added in application.

**11.** **CONCLUTION**

The project report entitled ― “Being Mother” has come to its final stage of designing. We will try our level best to make the application as dynamic as possible. We have tried to make the diagrams that can be easily understood. Also provision is provided for future developments in the system.