

STARTUP - HEALTHY

Submitted by

P S KRISHNAN NAMPOOTHIRY (Reg No:32019895021)

Under the supervision of

RESHMA ANN MATHEWS

(Assistant Professor, Department of Computer Science)

COLLEGE OF APPLIED SCIENCE, PERISSERY



PROJECT REPORT

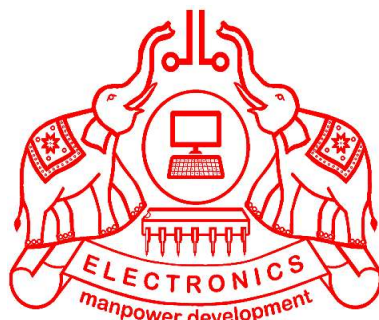
*Submitted in partial fulfilment of the
Requirements for the award of
BSC (Computer Science) degree of
University of Kerala*

2022

COLLEGE OF APPLIED SCIENCE PERISSERY, CHENGANNUR

(Affiliated to University of Kerala)

(Managed by Institute of Human Resource Development)



Certificate

This is to certify that the project report entitled “**STARTUP - HEALTHY**” is the Bonafede record of the Project work done by **Mr P S KRISHNAN NAMPOOTHIRY (Reg No:32019895021)** under our guidance and supervision, towards partial fulfilment of the requirements for the award of the Degree of BSC (Computer Science) of the University of Kerala.

Name &Signature of the Guide

Name &Signature of the Principal

Name &Signature of the HOD

Name &Signature of the External

ACKNOWLEDGEMENT

I hereby take the opportunity to thank **Asst Prof. Jaison D Joseph** Head of Department, **Asst Prof. Reshma Ann Mathews**, Project Coordinator of the Computer Science department, College of Applied Science, Perissery, for their valuable points and guidance throughout the course of my project. Last but not leaving the one, I would like to thank my parents for their motivation and my friends who gave me their valuable contributions regarding my topic and the encouragement through the preparation of project. Above all I would like to thank God for his abundant blessings without which this wouldn't be possible.

P S Krishnan Nampoothiry

TABLE OF CONTENTS

| | |
|--|-----|
| ABSTRACT..... | i |
| LIST OF FIGURES..... | ii |
| LIST OF TABLES..... | iii |
| 1. INTRODUCTION..... | 1 |
| 1.1 AIM | 2 |
| 1.2 OBJECTIVE..... | 2 |
| 1.3 ABOUT THE PROJECT..... | 2 |
| 2. SYSTEM STUDY AND ANALYSIS..... | 3 |
| 2.1 EXISTING SYSTEM..... | 4 |
| 2.2 NEED FOR THE SYSTEM..... | 4 |
| 2.3 PROPOSED SYSTEM..... | 5 |
| 2.4 SCOPE OF THE PROJECT..... | 5 |
| 2.5 PROBLEM DEFINITION..... | 5 |
| 2.6 FEASIBILITY ANALYSIS..... | 6 |
| 3. REQUIREMENT ANALYSIS & SPECIFICATION..... | 8 |
| 3.1 SOFTWARE REQUIREMENT SPECIFICATION..... | 9 |
| 3.2 HARDWARE REQUIREMENT SPECIFICATION..... | 10 |
| 3.3 SOFTWARE SELECTION..... | 10 |
| 4. SOFTWARE DESIGN AND DEVELOPMENT..... | 18 |
| 4.1 INTRODUCTION..... | 19 |
| 4.2 INPUT DESIGN..... | 20 |
| 4.3 OUTPUT DESIGN..... | 21 |
| 4.4 MODULE DESCRIPTION..... | 22 |
| 4.5 DESIGN METHODOLOGY..... | 22 |
| 4.5.1 DATA FLOW DIAGRAMS..... | 25 |
| 4.5.2 E-R DIAGRAMS..... | 29 |
| 4.6 DATA BASE DESIGN..... | 31 |
| 4.6.1 TABLES..... | 32 |
| 5 CODING AND TESTING..... | 37 |
| 5.1 INTRODUCTION..... | 38 |
| 5.2 CODING..... | 38 |
| 5.3 UNIT TESTING..... | 43 |
| 5.4 INTEGRATION TESTING..... | 43 |
| 5.5 VALIDATION TESTING..... | 43 |
| 5.6 FINAL SYSTEM TESTING..... | 44 |
| 6 SYSTEM IMPLEMENTATION AND MAINTENANCE..... | 46 |
| 6.1 SYSTEM SECURITY MEASURE..... | 47 |
| 6.2 IMPLEMENTATION DETAILS..... | 47 |
| 6.3 MAINTENANCE..... | 48 |
| 7 FUTURE ENHANCEMENT..... | 49 |
| 8 CONCLUSIONS..... | 51 |
| 9 APPENDIX..... | 53 |
| 10 BIBLIOGRAPHY..... | 58 |

ABSTRACT

A comprehensive solution for real time surveillance, care and support for people. This portal is a one stop platform for the public to avail emergency services and information related to Covid Pandemic and ensure transparency and quality in public services and welfare measures and to register the vaccination slots including covid vaccines and other vaccines (Polio, Hepatitis, etc.). Our project to be developed on Web Application and Android Application for users. The Covid pandemic has created a global health crisis that has had a deep impact on the way we perceive our world and everyday lives. The disease is more likely to cause symptoms on older adults and those with underlying health condition. A World-Wide virus attack has shut down cities, big and small and even entire countries. By using this application, you can get frequent updates related to current pandemics (Covid, Omicron) and Health care centres and can connect with the health experts.

LIST OF FIGURES

| | |
|--|----|
| Figure 4.1 : Level 0 DFD – Context Diagram of Startup – Healthy..... | 27 |
| Figure 4.2 : Level 1 DFD – Admin..... | 28 |
| Figure 4.3 : Level 1 DFD – Hospital..... | 28 |
| Figure 4.4 : Level 1 DFD – User..... | 29 |
| Figure 4.5 : ER Diagram Model..... | 29 |
| Figure 4.6 : ER Diagram of Startup – Healthy..... | 31 |
| Figure 9.1 : Admin Panel..... | 54 |
| Figure 9.2 : Graph View..... | 54 |
| Figure 9.3 : Containment Zone..... | 55 |
| Figure 9.4 : News & Guidelines..... | 55 |
| Figure 9.5 : Login Form..... | 56 |
| Figure 9.6 : New Registration..... | 56 |
| Figure 9.7 : User Home..... | 56 |
| Figure 9.8 : Covid Care Centres..... | 56 |
| Figure 9.9 : News & Guidelines..... | 57 |
| Figure 9.10: Emergency Services..... | 57 |
| Figure 9.11: Hospital Home..... | 57 |
| Figure 9.12: Vaccine Details..... | 57 |

LIST OF TABLES

| | |
|---|----|
| Table 3.1 : Software Specification..... | 9 |
| Table 3.2 : Hardware Specification..... | 10 |
| Table 4.1 : bank..... | 32 |
| Table 4.2 : availability..... | 33 |
| Table 4.3 : containment..... | 33 |
| Table 4.4 : customer..... | 33 |
| Table 4.5 : feedback..... | 34 |
| Table 4.6 : hospital_bed_rqst..... | 34 |
| Table 4.7 : hospitaller..... | 34 |
| Table 4.8 : login..... | 35 |
| Table 4.9 : services..... | 35 |
| Table 4.10 : vaccination..... | 35 |
| Table 4.11 : vaccin request..... | 36 |

1. INTRODUCTION

1.1. AIM

The main aim of this projects is to get connected with the Health Sector in one tap and act as a one stop platform to collect any vaccines and to collect the data's related to the pandemic in real – time

1.2 OBJECTIVE

The main objective of this project is:

- To make all the processes more efficient, speedy and secure by automating the system.
- Allow all users to book any type of vaccines irrespective of age.
- Can collect all the details about the pandemic in real-time
- Easily find out the nearby containment zones
- It helps to minimize time and effort.

1.3. ABOUT THE PROJECT

- Excluding Arogyasethu there is not even a single app to book vaccines, but using Arogyasethu you can book the vaccines only for covid pandemic.
- This Application for Health Services is a progressive step in the field of Health Sector
- Any user can make use of this website to locate and communicate with the health services in their vicinity and can locate the containment zones nearby
- This website uses innovative technology that connects you with a Health Services at the tap of a button

2. SYSTEM STUDY AND ANALYSIS

The first step in the system study includes analysis of the system. System analysis involves studying the way an organization currently receives and process data to produce information with the goal of determining how to make it work better. System analysis includes both a preliminary and a detailed stage. During preliminary analysis the analysis takes a quick look at what is needed and whether it benefits the perceived want. Detailed analysis includes an in depth look at what is wanted and contains more refined cost and benefits studies. The preliminary analysis begins when someone perceives a problem, modifications to existing, repairs to an existing system or demands an entirely new system. The analyst summarizes the gained modifications, including personal requirements and potential benefits of the new system in formal report called the preliminary report. Detailed analysis expands the preliminary efforts to include the complete analysis of all possible alternative solutions to the problem and complete expansion of what appears to be the most practical solution.

2.1. EXISTING SYSTEM

- There is only a single app named Arogyasethu. But it can't predict the current details of covid and also some other functionalities are provided in the websites, if we need to acquire rather details about the current covid and omicron details we need to browse the webpages
- The details about the pandemic are shattered on multiple websites
- Arogyasethu
- <https://www.worldometers.info/coronavirus/>

DISADVANTAGES:

- Doesn't get any idea about the upcoming covid status
- Can't book other vaccines
- We need to contact with the hospitals to collect information's about other vaccines
- We are not easily able to find out the current status of the district (Is in Red alert or yellow alert something)
- We can't find out the covid care centers easily
- We can't find out the availability of bed, ventilators, ICUs etc. easily

2.2. NEED FOR THE SYSTEM

- Convenience to use an app.
- Can be able to book any type of vaccines in one app.
- A wide range of services available in one app, anytime.
- Secure paperless store management.
- User friendly services.
- 24/7 support

2.3. PROPOSED SYSTEM

Can be able to predict the covid status of upcoming days using trend analyzing algorithm We can book covid vaccines as well as other vaccines like polio and others We can acquire all the information's about the vaccines without visiting the hospitals All the needs like Book vaccines, Covid updates and all others related to current pandemic situation comes under an umbrella. We can get real time information's about the alerts given to the districts. We can easily find out the information's related to covid care centers. We can get the details of the availability of bed, ventilators, ICUs etc. on hospitals and also, we can book them.

2.4. SCOPE OF THE PROJECT

The purpose of an application in healthcare system is to develop and automated tool for identifying and disseminating relevant pandemic situations and to provide a better care. In this system we can access to all the information related to current pandemic situation and to connect with the health experts and provide an easy way to access the emergency services. We need to find out the importance of vaccines. Most importantly we are need to be more vigilant.

A comprehensive solution for real time surveillance, care and support for people. This portal is a one stop platform for the public to avail emergency services and information related to Covid Pandemic and ensure transparency and quality in public services and welfare measures and to register the vaccination slots including covid vaccines and other vaccines (Polio, Hepatitis, etc.). even if we came across to met with another outbreak, its very easy to convert this system as a complete guide to prevent the widespread of that pandemic. Since this is an android application, most of the peoples are using an android device hence everyone can easily connect with the system very easily and get aware about the pandemic crisis and become more vigilant.

2.5. PROBLEM DEFINITION

Start-up Healthy observed that WHO (world health organization) has recently updated the case definitions based on the current information available and will be revised as new information accumulates. Making covid information and guidance from the World Health Organization as Emergency services (Ambulance, Police, containment Zone). Ambulance Services for remote areas and life-threatening emergencies. Effective communication system between the covid care centers, ambulances and the emergency facilities in the hospitals with having vaccination booking as well as bed availability

2.6. FEASIBILITY STUDY

The main objective of the feasibility study is to test the technical, operational and economic feasibility of developing a computer system. The proposed system must be evaluated from a technical, operational and economic feasibility of developing of a computer. In any Project, feasibility analysis is a very important stage; here the project is checked for its feasibility. Any project may face scarcity in resources, time or work force. Hence all these are to be studied in detail and a conclusion should be drawn whether the project under consideration is feasible or not. This analysis is a test of the proposed project, regarding its work ability, impact on users and clients and resources management. Feasibility and risk involved are inversely related to each other. The main objective of feasibility is to test technical, operational and economic feasibility of a project. The proposed system must be evaluated from a technical view point first, and if technically higher impact on the customer and service provider must be accessed, if compatible social and technical system can be devised. Then the system must be tested for the economic feasibility. Since if the proposed system is not either Economically, Technically, socially feasible, it cannot be easily implemented the proposed system. Feasibility study of a candidate system can be broadly classified as follows:

2.6.1. ECONOMIC FEASIBILITY

The proposed system is economically feasible. The economic feasibility is the most important and used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better result along with increased efficiency. It is also known as Cost benefit analysis. If the benefit is more than the cost then the system is approved. A simple economic analysis which gives the actual comparison of the cost and benefits are much meaningful in this case. It benefits outweigh costs, and then the decision is made to design and implement the system, Otherwise, further justifications or alternations in the proposed system will have to made if it is to have a change of being approved.

The given system can be developed under optimal expense with the available hardware and software. Besides it is a good economic to invest in such a kind of software from the project manager's point of view as the benefits over weighs the costs. The resources need to run the above project should be less in cost and highly reliable so that there might be no hanging and minimum level of expense to implement the software. Economic feasibility is the most frequently used method for evaluating the effectiveness of a candidate system.

2.6.2. OPERATIONAL FEASIBILITY

The system operation is the longest phase in the development life cycle of a system so operational feasibility should be given much important. The users of the system don't need through training on the system. All they are expected to know to operate the system is the basic net surfing knowledge. It has a user-friendly interface. Operational Feasibility is considered with the working of the system after its installation. The company has good record of development, installation, and maintenance of systems for its clients. So, this system can be installed in the client environment and the company will help in the main-tenancy of the system in future. Proposed projects are beneficial only if they can be turned into information system that will meet the organizations operating requirements simply stated, this test of feasibility asks if the system will work when it is developed and installed. The present system is easily understandable and maintenance and working of a new system needs fewer human efforts. The proposed project is beneficial to the user and is user friendly. So, the system can be judged operationally feasible.

2.6.3. TECHNICAL FEASIBILITY

The project requires the system to be functional and multi-user one. It should be based on specified technology. The system under study must be portable and platform independent. It should be compatible with all kind of existing system in industry and should not provide any overhead to the user. Technical feasibility sends on the existing system and to what extent it can support the proposed system. Updating and viewing the previous editions are tedious in the existing system. Implementation of proposed system does not require saving of the existing configuration of the system.

Technical Feasibility centers around the existing system and do what extend it can support the proposed addition. Technical considerations evaluate existing hardware and software. This involves financial considerations to accommodate technical enhancements. Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed designs of the system, making it difficult to access issue like performance, cost, etc. A number of issues have to be considered while doing a technical analysis.

2.6.4. SOCIAL FEASIBILITY

The feasibility study determines whether the system will work in the place of existing system. Here the proposed system is user friendly and there is no special training for using the system. It provides the users more effective and efficient way to utilize this system easily. Therefore the proposed system is socially feasible.

In social feasibility the acceptance of the people is considered regarding the product to be launched. It describes the effect on users from the introduction of the new system considering whether there will be a need for retraining the workforce

3. REQUIREMENT ANALYSIS & SPECIFICATION

A software requirement specification (SRS), a requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition, it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints) The software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements, we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer.

3.1 SOFTWARE SPECIFICATION

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. We require much different software to make the application which is in making to work efficiently. It is very important to select the appropriate software so that the software works properly.

| | |
|----------------------|------------------------------------|
| Operating System | 64-bit Microsoft® Windows® 8/10/11 |
| Programming Language | Java, Android |
| RDBMS | MYSQL |
| Web Server | Apache Tomcat & Glassfish Server |
| Scripting language | JSP |

Table 3.1

3.2 HARDWARE SPECIFICATION

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and capacity requirements are also important. Below are some of the hardware that is required by the system.

| | |
|----------------|--|
| Main Processor | Intel core i3 or above |
| RAM | 8 GB or Above |
| Keyboard | Standard 108 keys |
| Mouse | 3D Optical mouse |
| Monitor | 15" Standard |
| Hard disk | 10 GB of available disk space minimum or above |

Table 3.2

3.3 SOFTWARE SELECTION

It is very important to select appropriate software so that the software works properly.

FRONT END: ANDROID

Android is a Linux based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers. Android is open source and Google releases the code under the Apache License This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wire- less carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications that extend the functionality of devices, written primarily in a customized version of the Java programming language. Android is an open-source mobile operating system that combines and builds up on parts of many different open-source projects.

Android's kernel is based on the Linux kernel and has further architecture changes by Google outside the typical Linux kernel development cycle. Android does not have a native Window System nor does it support the full set of standard GNU libraries, and this makes it difficult to port existing Linux applications or libraries to Android. Android's user interface is based on direct manipulation using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. Android devices boot to the home screen, the primary navigation and information point on the device, which is similar to the desktop found on PCs. Android home screens are typically made up of amplicons and widgets; amplicons launch the Assoc- acted app, whereas widgets display live, auto updating content such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen. Graphical user interfaces Web frameworks Multimedia Database Networking

ABOUT JAVA

Java is a general-purpose computer-programming language that is concurrent, class- based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers “write once, run anywhere” (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling, a Canadian, at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its original features from Smalltalk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plugin for applets).

The latest version is Java 11, released on September 25, 2018. Java 11 is a currently supported long-term support (LTS) version (“Oracle Customers will receive Oracle Premier Support”); Oracle released for the “legacy” Java 8 LTS the last “public update”, which is free for commercial use, in January 2019.

Oracle will still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) “highly recommend that you uninstall older versions of Java”, because of serious risks due to unresolved security issues. Since Java 9 is no longer supported, Oracle advises its users to “immediately transition” to Java 11. Oracle extended support for Java 6 ended in December 2018.

ABOUT ANDROID

Android is a Linux-based operating system designed primarily for touch screen mobile devices such as Smartphone and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005. Android is open source and Google releases the code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers,

Wireless carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications (“apps”) that extend the functionality of devices, written primarily in a customized version of the Java programming language.

Android Studio is Android’s official IDE. It is purpose-built for Android to accelerate your development and help you build the highest-quality apps for every Android device.

Android Studio’s Instant Run feature pushes code and resource changes to your running app. It intelligently understands the changes and often delivers them without restarting your app or rebuilding your APK, so you can see the effects immediately.

The code editor helps you write better code, work faster, and be more productive by offering advanced code completion, refactoring, and code analysis. As you type, Android Studio provides suggestions in a dropdown list. Simply press Tab to insert the code

The Android Emulator installs and starts your apps faster than a real device and allows you to prototype and tests your app on various Android device configurations: phones, tablets, Android Wear, and Android TV devices. You can also simulate a variety of hardware features such as GPS location, network latency, motion sensors, and multi- touch input.

Android Studio includes project and code templates that make it easy to add well- established patterns such as a navigation drawer and view pager. You can start with a code template or even right-click an API in the editor and select Find Sample Code to search for examples. Moreover, you can import fully functional apps from GitHub, right from the Create Project screen.

Android Studio provides a robust static analysis framework and includes over 280 different lint checks across the entirety of your app. Additionally, it provides several quick fixes that help you address issues in various categories, such as performance, security, and correctness, with a single click.

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit.

ADT (Android Development Tools) is the software used to develop android apps. It basically encases Eclipse IDE, which is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment... The latest version comes with ADT plug-in preinstalled and bundled to the IDE.

Application programming interface (API) specifies how some software components should interact with each other. In practice in most of the cases an API is a library that usually includes specification for routines, data structures, object classes, and variables

| Name | Internal codename | Version number(s) | Initial release date | stable API level |
|------------------------|-------------------|-------------------|----------------------|------------------|
| <u>Android 1.0</u> | N/A | 1.0 | September 23, 2008 | 1 |
| <u>Android 1.1</u> | Petit Four | 1.1 | February 9, 2009 | 2 |
| <u>Android Cupcake</u> | Cupcake | 1.5 | April 27, 2009 | 3 |
| <u>Android Donut</u> | Donut | 1.6 | September 15, 2009 | 4 |
| <u>Android Eclair</u> | Eclair | 2.0 | October 27, 2009 | 5 |
| | | 2.0.1 | December 3, 2009 | 6 |

| | | | | |
|-----------------------------------|--------------------|---------------|-------------------|----|
| | | 2.1 | January 11, 2010 | 7 |
| <u>Android Froyo</u> | Froyo | 2.2 – 2.2.3 | May 20, 2010 | 8 |
| <u>Android Gingerbread</u> | Gingerbread | 2.3 – 2.3.2 | December 6, 2010 | 9 |
| | | 2.3.3 – 2.3.7 | February 9, 2011 | 10 |
| <u>Android Honeycomb</u> | Honeycomb | 3.0 | February 22, 2011 | 11 |
| | | 3.1 | May 10, 2011 | 12 |
| | | 3.2 – 3.2.6 | July 15, 2011 | 13 |
| <u>Android Ice Cream Sandwich</u> | Ice Cream Sandwich | 4.0 – 4.0.2 | October 18, 2011 | 14 |
| | | 4.0.3 – 4.0.4 | December 16, 2011 | 15 |
| <u>Android Jellybean</u> | Jellybean | 4.1 – 4.1.2 | July 9, 2012 | 16 |
| | | 4.2 – 4.2.2 | November 13, 2012 | 17 |
| | | 4.3 – 4.3.1 | July 24, 2013 | 18 |
| <u>Android KitKat</u> | Key Lime Pie | 4.4 – 4.4.4 | October 31, 2013 | 19 |
| | | 4.4W – 4.4W.2 | June 25, 2014 | 20 |
| <u>Android Lollipop</u> | Lemon Meringue Pie | 5.0 – 5.0.2 | November 4, 2014 | 21 |
| | | 5.1 – 5.1.1 | March 2, 2015 | 22 |

| | | | | |
|----------------------------|----------------------|-------------|-------------------|----|
| <u>Android Marshmallow</u> | Macadamia Nut Cookie | 6.0 – 6.0.1 | October 2, 2015 | 23 |
| <u>Android Nougat</u> | New York Cheesecake | 7.0 | August 22, 2016 | 24 |
| | | 7.1 – 7.1.2 | October 4, 2016 | 25 |
| <u>Android Oreo</u> | Oatmeal Cookie | 8.0 | August 21, 2017 | 26 |
| | | 8.1 | December 5, 2017 | 27 |
| <u>Android Pie</u> | Pistachio Ice Cream | 9 | August 6, 2018 | 28 |
| <u>Android 10</u> | Quince Tart | 10 | September 3, 2019 | 29 |
| <u>Android 11</u> | Red Velvet Cake | 11 | September 8, 2020 | 30 |
| <u>Android 12</u> | Snow Cone | 12 | October 4, 2021 | 31 |

An API specification can take many forms, including an International Standard such as POSIX, vendor documentation such as the Microsoft Windows API, the libraries of a programming language, e.g., Standard Template Library in C++ or Java API.

Google APIs can be downloaded from Google Code, Google's site for developer tools, APIs and technical resources. The Google Data API] allow programmers to create applications that read and write data from Google services. Currently, these include APIs for Google Apps, Google Analytics, Blogger, Google Base, Google Book Search, Google Calendar, Google Code Search, Google Earth, Google Spreadsheets, Google Notebook, and Picasa Web Albums.

SDK (Software Development Kit or “devkit”) is typically a set of software development tools that allows for the creation of applications for a certain software package software framework, hardware platform, computer system, video game console, operating system, or similar development platform. It may be something as simple as an application programming interface (API) in the form of some files to

interface to a particular programming language or include sophisticated hardware to communicate with certain

Embedded system. Common tools include debugging aids and other utilities often presented in an integrated development environment (IDE). In the latest version of ADT, the android SDK adds on to the IDE automatically as soon as you unzip and load the IDE.

Android Virtual Device (AVD) manager enables us to launch virtual android devices/ emulators in our PC and run the app in the emulator, and at the same time we can track and debug each app activity from the Logcat in our IDE.

The major advantages of Android Studio are:

- Faster Deployment of Fresh Builds.
- More Accurate Programming.
- The newly introduced emulator is 3 as faster in CPU, RAM, & I/O in comparison to its predecessor.
- Promoting is an important component of the app marketing, and Android Studio 2.0 takes it to a new high

ABOUT MySQL

MySQL is a relational database management system (RDBMS) which is more than 11 million institutions. The program runs as a server providing multi-user access to a number of databases.

MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now a subsidiary of Sun Microsystems, which holds the copyright to most of the code base. The project's source code is available under terms of the GNU General Public License, as well as under a variety of proprietary agreements.

SELECTION OF OPERATING SYSTEM

Windows10 an Overview

Windows 10 includes improved network, application and Web services. It provides increased reliability and scalability, lowers your cost of computing with powerful, flexible management services, and provides the best foundation for running business application.

Network Data Security

Network data can be protected on the wire or at the network interface. Securing data at the network requires a firewall to proxy services and mediate connections between the internal network, (LAN) and external network (Internet). This is the purpose of Proxy Server.

Internet Protocol Security

Internet Protocol Security (Ipsec) is a framework of open standards for ensuring secure private communications over Internet Protocol networks, using cryptographic security services.

INTERNET EXPLORER WEB BROWSER

INTERNET EXPLORER(IE) makes it easier to get the most from the world wide web, whether we are searching for new information or browsing your favorite website. And built-in intelligence technology can save us time completing web tasks, such as automatically completing web addresses and form for, and automatically detecting your network and connection status.

ADVANTAGES OF IE

When we start typing a frequently used web address in the address bar, a list of similar appears that you can choose from. And if a web page address is wrong, IE can search for similar addresses to try to find a match.

Search for websites by clicking the search button on the toolbar.

In the search bar type a word or phrase that describes what you are looking for.

Go to other web pages similar to the one you are viewing without even doing a search. Just use the show related sites feature. Browse through the list of web pages you recently visited by clicking the history button on the tool bar.

4. SOFTWARE DESIGN AND DEVELOPMENT

4.1 INTRODUCTION

System design is an interactive process through which requirements are transmitted to a “blue print” for constructing the software initial; the blue print depicts a holistic view of software that is design is represented at a high-level abstraction a level that can be directly traced to specific data, functional and behavioral requirements. System design is the solution to the creation of a new system. This is the important aspect made up of several steps. System design is the process of developing specifications for a candidate system that meet the criteria established in the system analysis. Major step in system design is the preparation of the input forms and output reports in a form applicable to the users.

The main objective of system design is to use the package easily by a computer operator. System design is the creative act of invention, developing new inputs, a database, off-line files, method, procedures and output for processing business to meet an organization objective. System design-built information gathered during the system analysis. As design interaction occurs subset-quant refinement leads to design representation at much lower level of abstraction. System design is a creative art of inventing and developing input, data bases, off line files, method and procedures, for processing data to get meaningful output that satisfies the organization objectives. Through the design phase consideration to the human factor, that is inputs to the users will have on the system.

Some of the main factors that have to be noted using the design of the system are:

- Practicability
- System must be capable of being operated over a long period of time and must have ease of use.
- Efficiency
- Make better use of available resources. Efficiency involves accuracy, timeliness and comparability of system output.
- Cost
- Aim of minimum cost and better results.
- Security
- Ensure physical security of data.

4.2 INPUT DESIGN

Input design is the process of converting user-oriented input to a based format. Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. The goal of designing input data is to make data entry as easy, logical and free from errors. When we approach input data design; we design the data source documents that capture the data and then select the media used to enter them into computer.

User-friendly screen format can reduce the burden on end users, who are not highly proficient in computers. An important step in input design stage is a design of source document. Source document is the form in which the data can initially capture. The next step is the design of the document layout. In the layout organizes the document by placing information, where it will be noticed and establishes the appropriate sequence of items.

User interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the user- oriented inputs into the computer-based format. Input design is a part of overall system design, which requires very careful attention. If data going into the system is correct, then the processing and output will magnify these errors. Thus, the designer has a number of clear objectives in the different stages of input design

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that input is acceptable to and understand by the user.

Space 1cm Inaccurate input data is most common cause of data processing errors. If poor input design-particularly where operators must enter data from source documents-permits bad data to enter a computer system, the outputs produced are of little value. The input design process was initiated in the study phase were, as a part of the feasibility study:

1. Input data were found to be available for establishing and maintaining master and transaction files and for creating output records
2. The most suitable types of input media, for either off-line or on-line devices were selected after a study of alternative data capture techniques

The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. This data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right messages for the user at the right time are also considered for development for this project.

4.3 OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. Efficient and intelligent output design improves the system's relationship and helps user decision-making.

In the output design it is determine how the implementation is to be played for immediate need and also the hardcopy output. A major form of input is a hardcopy from the printer. Print- outs should be designed around the output requirement of the user. Printers, CRT screen display are the examples for providing computer-based output. The output design associated with the system includes the various reports of the table generations and query executions.

A quality output is one, which meets the requirements of end user and presents the in- formation clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision –making. The objective of the output design is to convey the information of all the past activities, current status and to emphasis important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output design is one of the, most important features of the information system. The log- ictal design of an information system is analogous to an engineering blue print of an automobile. It shows the major features and how they are related to one another. The outputs, inputs and databases are designed are in this phase. At the beginning of the output design various types of outputs such as external, internal, operational, and interactive and turnaround are defined. Then the format, content, location, frequency, volume and sequence of the outputs are specified. The content of the output must be defined in detail. The system analysis has two specific objectives at this stage.

- To interpret and communicate the results of the computer part of a system to the users in a form, which they can understand, and which meets their requirements
- To communicate the output design specifications to programmers in a way in which it is unambiguous, comprehensive and capable of being translated into a programming language.

4.4 MODULE DESCRIPTION

The main modules present in the system are:

1. Admin
2. User
3. Hospital

Admin Module

The admin module is responsible for the overall management of this system. The admin has the only power to add containment zones and Emergency services. The admin also has access to all the details of the other modules. The Admin Module can view all the details of covid status both graphically and digitally. The admin can view the news and guidelines provided by the government authorities. Controls the feedback section The admin also has access to verify the payments

User Module

In this module, the user can get the real-time covid data in both graphically and digitally. The user can be able to locate the nearby covid care centers easily by accessing the location services and seek the bed requests on the hospital. Can be able to view the news and guidelines provided by the government and can access the emergency services. Can book vaccination not only for covid. And make payment for the private hospitals They can add their bank for payments and add feedback on these services.

Hospital Module

In this module, the user can update the availability of the services provides by the hospitals in real-time. The vaccination details of the entire system is added through this module. They can be able to pass a complaint to the admin very easily

4.5 DESIGN METHODOLOGY

The design of this project is divided into three phases as described below:

- User Interface design: In this phase the UI or the user interface of the project is developed. That is, the designing of the web pages which the user will use to interact.
- Database design: The database is considered to be the main pillars of every project. In our application, database is used to store user details such as name, age etc. Database here is also used to keep information about the emails sent or received or in draft. The complete proposed design of the database is shown in the ER-diagram below. This ER-diagram shows all the tables with all fields and also relationship between different tables.

- System design: In this phase a complete flow diagram of the working system is designed. This flow diagram will show the details of all the events like actions to be performed for an event.
- After completion of the design phase, we will now start to implement our project. Design methodology refers to the development of a system or method for a unique situation.

Today, the term is most often applied to technological fields in reference to web design, software or information systems design. Various degree programs involve design methodology, including those in the graphic and digital arts. Therefore, the system design includes the following types of design.

- **Level-Oriented Design**

In the level-oriented design approach, there are two general or broad strategies that can be used. The first strategy starts with a general definition of a solution to the problem then through a step by-step process produce a detailed solution (this is called Stepwise Refinement). This is basically dependent on the system requirements and is a top-down process. The other strategy is to start with a basic solution to the problem and through a process of modeling the problem, build up or extend the solution by adding additional features (this is called design by composition).

- **Data Flow-Oriented Design**

In the data flow-oriented design approach, which is often called Structured Design, information flow characteristic is used to derived program structure. In the data flow-oriented approach, emphasis is on the processing or operations performed on the data. Design is information driven. Information maybe represented as a continuous flow that is transformed, as it is processed from node to node in the input-output stream. As software can ideally be represented by a data flow diagram (DFD), a design model that uses a DFD can theoretically be applied in the software development project. The data flow-oriented approach is especially applicable when information is processed without hierarchical structure. A DFD can be mapped into the design structure by two means – transform analysis or transaction analysis. Transform analysis is applied when the data flow in the input-output stream has clear boundaries. The DFD is mapped into a structure that allocates control to three basic modules – input, process and output. Transaction analysis is applied when a single information item causes flow to branch along one of many paths. The DFD is mapped to a substructure that acquires and evaluates a transaction; another substructure controls all the data processing actions based on a transaction. A few examples of structured design or data flow-oriented design methodologies are Structured Analysis and Design Technique (SADT), Systematic Activity Modeling Method (SAMM) and Structured Design (SD).

• Data Structure-Oriented Design

The data structure-oriented design approach utilizes the data structures of the input data, internal data (for example databases) and output data to develop software. In the data structure-oriented approach, the emphasis is on the object, which is the data. The structure of information, called data structure, has an important impact on the complexity and efficiency of algorithms designed to process information. Software design is closely related to the data structure of the system, for example, alternative data will require a conditional processing element, repetitive data will require a control feature for repetition and a hierarchical data structure will require a hierarchical software structure. Data structure-oriented design is best utilized in applications that have well defined, hierarchical structure of information. As both data flow and data structure-oriented design approaches are based on considerations in the information domain, there are similarities between both approaches. Both depend on the analysis step to build the foundation for later steps. Both attempts to transform information into a software structure; both are driven by information. In data structure-oriented design information structure are represented using hierarchical diagrams; DFD has little relevance; transformation and transaction flows are not considered. Data structure-oriented design have a few tasks – evaluate the characteristics of the data structure, represent the data in its lowest form such as repetition, sequence or selection, map the data representation into a control hierarchy for software, refine the control hierarchy and then develop a procedural description of the software. Some examples of the data structure-oriented design approach are the Jackson System Development (JSD) and the Data Structured Systems Development (DDSD) which is also called the Warnier-Orr methodology.

• Object Oriented Design

The object-oriented design approach is unique in its usage of the three software design concepts: abstraction, information hiding and modularity. Objects are basically a producer or consumers of information or an information item. The object consists of a private data structure and related operations that may transform the data structure. Operations contain procedural and control constructs that may be invoked by a message, that is, a request to the object to perform one of its operations. The object also has an interface where messages are passed to specify what operation on the object is desired. The object that receives a message will then determine how the requested operation is to be performed. By this means, information hiding (that is, the details of implementation are hidden from all the elements outside the object) is achieved. Also objects and their operations are inherently modular, that is, software elements (data and process) are grouped together with a well-defined interface mechanism (that is, messages). Object oriented design is based on the concepts of: objects and attributes, classes and members, wholes and parts. All objects encapsulate data (the attribute values that define the data), other objects (composite objects can be

defined), constants (set values), and other related information. Encapsulation means that all of this information is packaged into a single name and can be re-used. The object-oriented design is rather new and as such it is still evolving even at this present moment. Object oriented design encompasses data design, architectural design and procedural design. By identifying classes and objects, data abstractions are created; by coupling operations to data, modules are specified and a structure for the software is established; by developing a mechanism for using objects (for example, passing of messages) interfaces are described.

4.5.1 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) or a bubble chart is a graphical tool for structured analysis. It was De Macro in 1978 and Gene and Carson in 1979 who introduced DFD. DFD models a system transforms the data and creates, output data-flows which go by using external entities from which data flows to a process which to other processes or external entities or files. Data in files many also flow to processes as inputs.

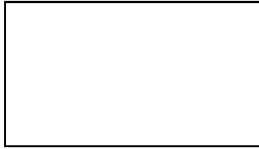
There are various symbols used in DFD. Bubbles represent the process. Named arrows indicate the dataflow. External entities are represented by rectangles and are outside the system such as vendors or customers with whom the system interacts. They either supply or consume data. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled in with a descriptive name. Process names are further identified with a number.

DFD can be hierarchically organized, which help in partitioning and analyzing large systems. As a first step, one Data Flow Diagram can depict an entire system. Which gives the system overview? It is called Context Diagram of level 0 DFD. The context diagram can be further expanded. The successive expansion of the DFD from the context diagram that giving more details is known as levelling of DFD. Thus, of top-down approach is used, starting with an overview and then working out the details.

The main merit of DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used, and where the result flow. The data flow diagram of Co-operative Bank Management System has been represented as a hierarchical DFD context level DFD was drawn first. Then the processes were decomposed into several elementary levels and are represented in the order of importance.

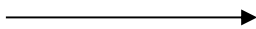
BASIC DFD SYMBOLS

Square:



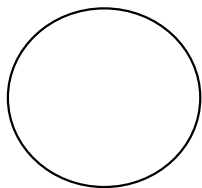
is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the context of data flow model

Arrow:



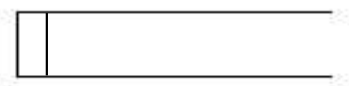
A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a processor and from data store or process. An arrow lines depicts the flow, with arrowhead pointing in the direction of flow

Circle:



A Process represents transformation where incoming data flows are changed into outgoing data flows.

Open rectangle:



A data is a repository of data that is to be stored for use by one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and doesn't alter it, the arrowhead goes only from the store to the process. If process alters the details in the store, then a double-headed arrow is used.

Rules for drawing Data Flow Diagram:

Establish the context of the data flow diagram by identifying all of the net input and output data flows.

- Select a starting point for drawing the data flow diagram
- Give meaningful labels to all data flow lines.
- Label all processes with action verbs that relate input and output data flows.
- Omit insignificant functions routinely handled in the programming process.
- Don't include control or flow of control information.
- Don't try to put too much information in one data flow diagram. Try to plan for the number of levels.
- Be prepared to start over.

DIAGRAMS

LEVEL 0

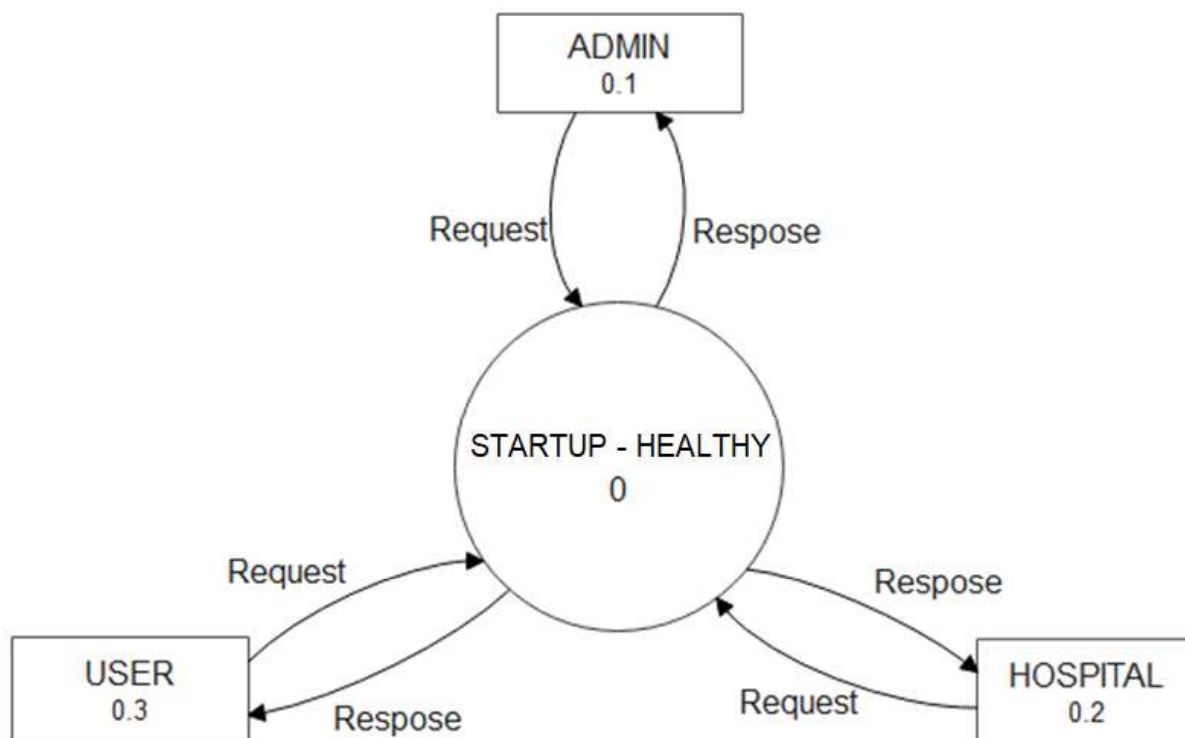


Fig. 4.1

LEVEL 1

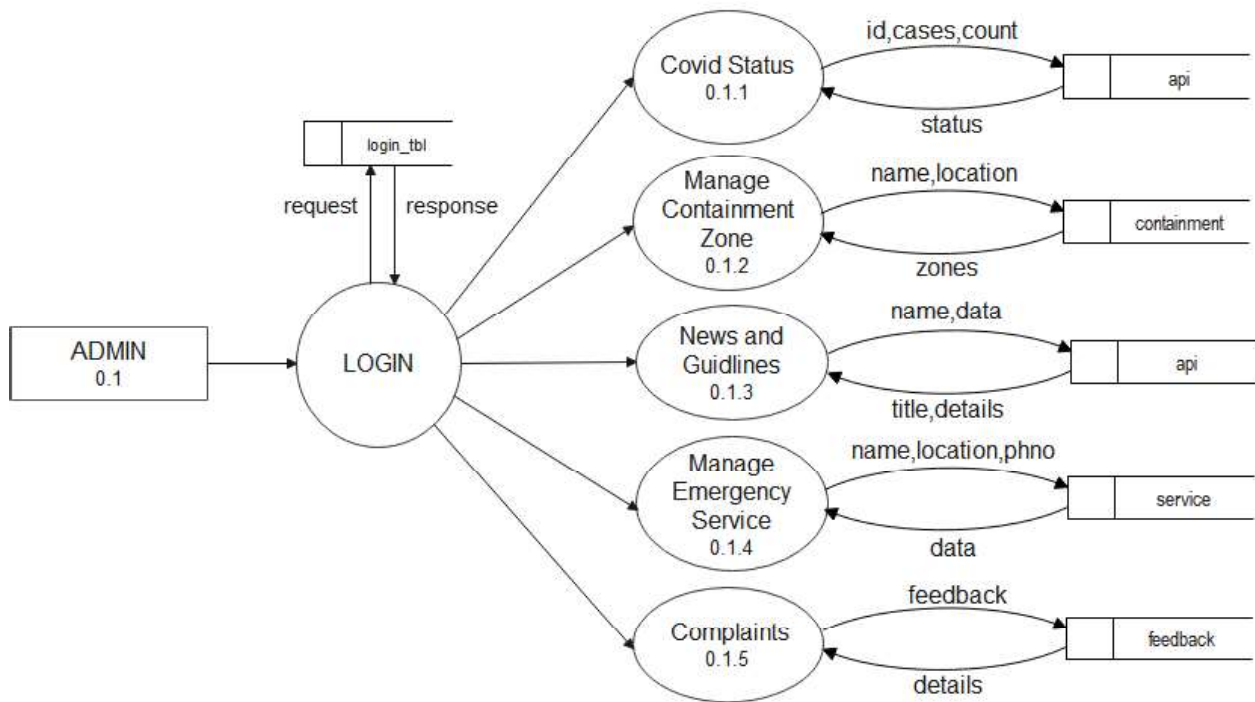


Fig. 4.2

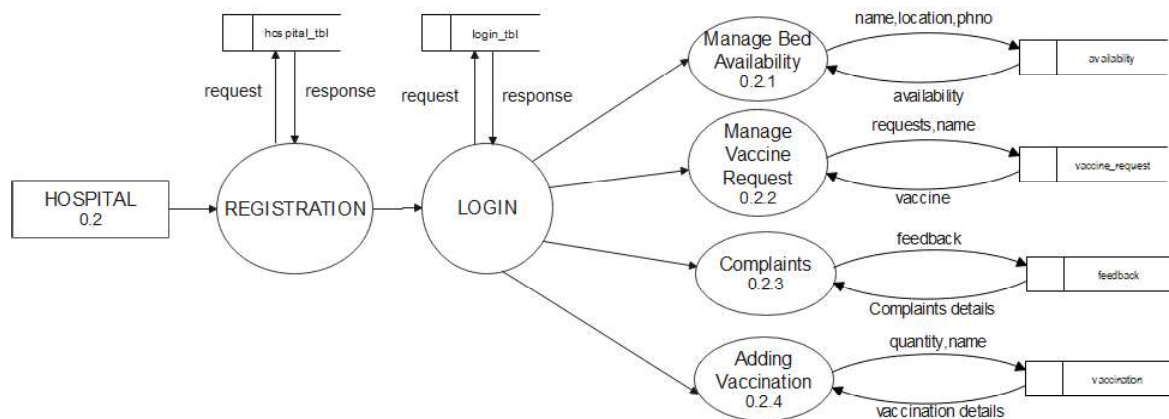


Fig 4.3

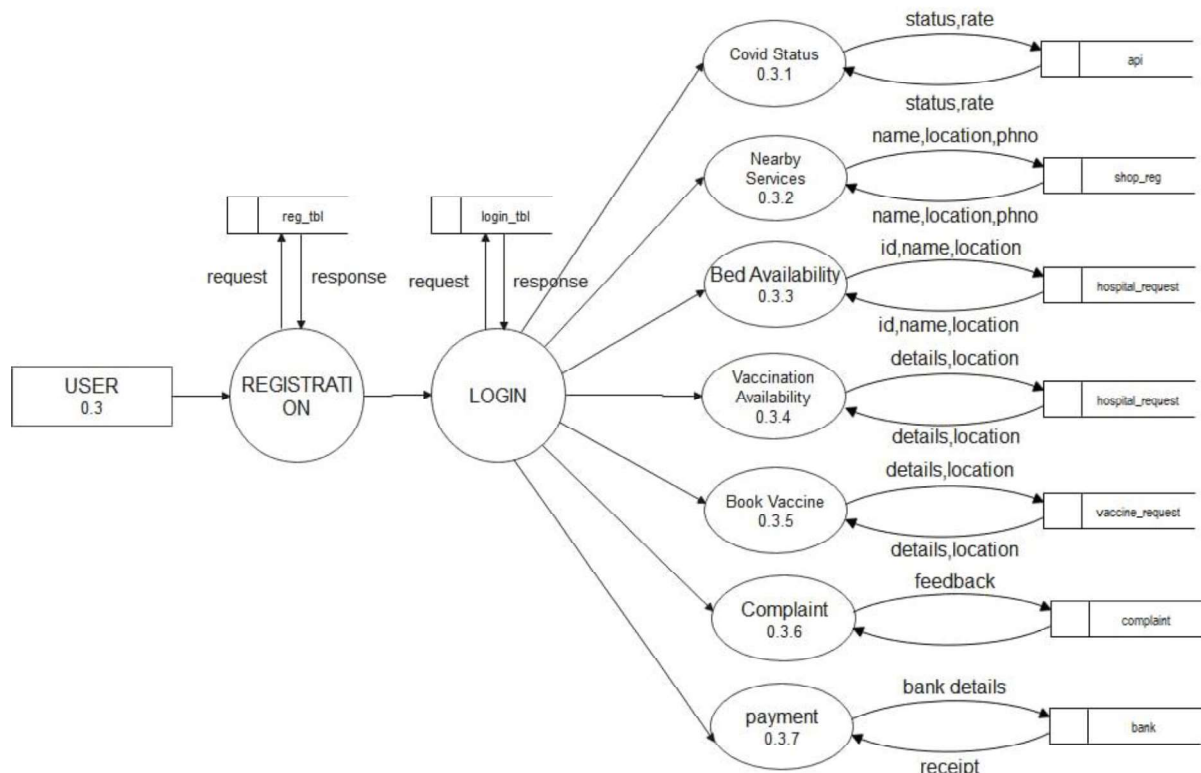


Fig. 4.4

4.5.2 ER DIAGRAM

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

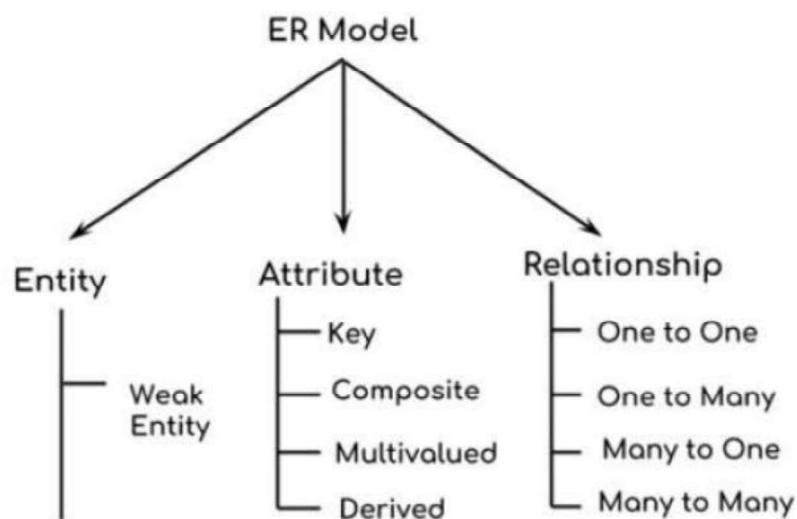


Fig. 4.5

Components of ER Diagram

ENTITY

An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

ATTRIBUTE

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

RELATIONSHIP

A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. There are four types of relationships:

1. One to One

When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.

2. One to Many

When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationships. For example – a customer can place many orders but an order cannot be placed by many customers.

3. Many to One

When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship. For example – many students can study in a single college but a student cannot study in many colleges at the same time.

4. Many to Many

When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationships. For example, a can be assigned to many projects and a project can be assigned to many students.

DIAGRAM

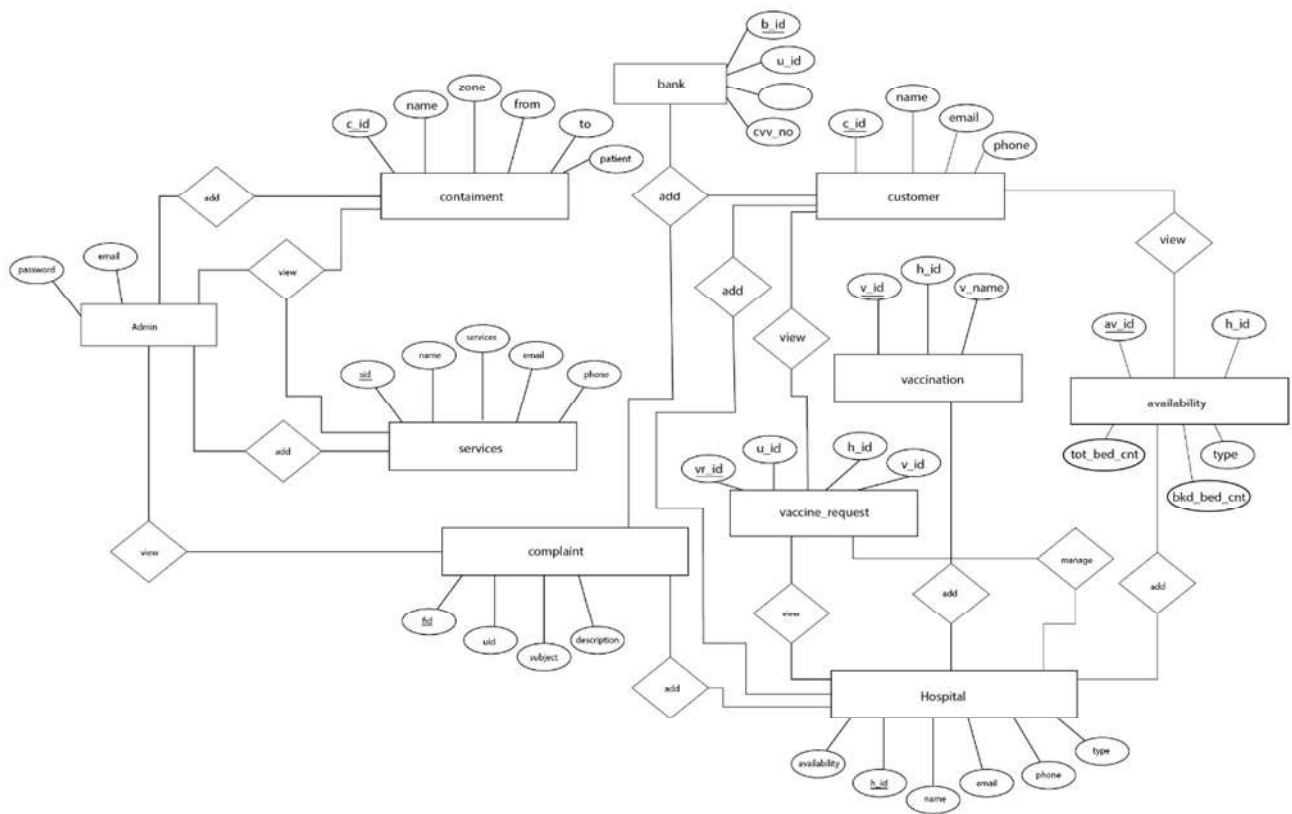


Fig. 4.6

4.6 DATABASE DESIGN

The Database design is the process of producing a detailed data model of a database. The logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributes for each entry. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model, these are the tables and views. In an object database, the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall data base applications within the database management system.

NORMALIZATION

The process of normalization is concerned with the transformation of the conceptual schema to a computer represent able form. Normalization reduces the redundancies and anomalies.

THE FIRST NORMAL FORM

First Normal form does not allow multivalued and composite valued attributes. It states that the domain of an attribute must include only atomic values and that value of any attribute in a table must be single value from the domain of that attribute.

THE SECOND NORMAL FORM

In Second Normal form, for relation where primary key contains multiple attributes, on key attribute should not be functionally dependent on a part of the primary key.

THE THIRD NORMAL FORM

In Third Normal form, relation should not have a non-key attribute functionally determined by non-key attribute. That is there should be no transitive dependency of a non-key attribute on the primary key.

4.6.1 TABLES

Table: bank

| Field | Type | Comment |
|----------|--------------------|-------------|
| bid | int (100) NOT NULL | PRIMARY KEY |
| urid | varchar (100) NULL | FOREIGN KEY |
| cardanol | varchar (100) NULL | |
| cvv_no | varchar (100) NULL | |
| pin | varchar (100) NULL | |
| balance | varchar (100) NULL | |
| status | varchar (100) NULL | |

Table 4.1

Table: availability

| Field | Type | Comment |
|-------------|--------------------|-------------|
| avid | int (20) NOT NULL | PRIMARY KEY |
| hid | varchar (20) NULL | FOREIGN KEY |
| tot_bed_cnt | varchar (100) NULL | |
| bkd_bed_cnt | varchar (100) NULL | |
| type | varchar (100) NULL | |

Table 4.2**Table: containment**

| Field | Type | Comment |
|-------------|---------------------|-------------|
| Cid | int (11) NOT NULL | PRIMARY KEY |
| name | varchar (200) NULL | |
| zone | varchar (100) NULL | |
| from | varchar (200) NULL | |
| to | varchar (100) NULL | |
| patients | varchar (200) NULL | |
| latitude | varchar (200) NULL | |
| longitude | varchar (200) NULL | |
| description | varchar (1000) NULL | |

Table 4.3**Table: customer**

| Field | Type | Comment |
|---------|--------------------|-------------|
| chid | int (20) NOT NULL | PRIMARY KEY |
| name | varchar (100) NULL | |
| age | varchar (50) NULL | |
| gender | varchar (100) NULL | |
| address | varchar (100) NULL | |
| phone | varchar (100) NULL | |
| Aadhar | varchar (100) NULL | |

| | | |
|----------|--------------------|--|
| email | varchar (100) NULL | |
| autojoin | varchar (100) NULL | |
| status | varchar (100) NULL | |

Table 4.4**Table: feedback**

| Field | Type | Comment |
|-------------|--------------------|-------------|
| fid | int (11) NOT NULL | PRIMARY KEY |
| aid | int (20) NULL | FOREIGN KEY |
| subject | varchar (20) NULL | |
| description | varchar (100) NULL | |
| rating | varchar (20) NULL | |
| status | varchar (10) NULL | |

Table 4.5**Table: hospital_bed_rqst**

| Field | Type | Comment |
|-----------|--------------------|-------------|
| hubristic | int (20) NOT NULL | PRIMARY KEY |
| hid | varchar (20) NULL | FOREIGN KEY |
| urid | varchar (20) NULL | FOREIGN KEY |
| bed_cnt | varchar (50) NULL | |
| type | varchar (50) NULL | |
| status | varchar (100) NULL | |

Table 4.6**Table: hospitaller**

| Field | Type | Comment |
|--------------|--------------------|-------------|
| hid | int (20) NOT NULL | PRIMARY KEY |
| name | varchar (100) NULL | |
| address | varchar (200) NULL | |
| phone | varchar (100) NULL | |
| type | varchar (100) NULL | |
| availability | varchar (100) NULL | |
| email | varchar (100) NULL | |
| autojoin | varchar (100) NULL | |

| | | |
|--------|--------------------|--|
| status | varchar (100) NULL | |
| flat | varchar (100) NULL | |
| flung | varchar (100) NULL | |

Table 4.7**Table: login**

| Field | Type | Comment |
|----------|--------------------|-------------|
| logic | int (11) NOT NULL | PRIMARY KEY |
| rigid | varchar (20) NULL | FOREIGN KEY |
| email | varchar (200) NULL | |
| password | varchar (200) NULL | |
| type | varchar (200) NULL | |
| status | varchar (200) NULL | |

Table 4.8**Table: services**

| Field | Type | Comment |
|-------------|---------------------|-------------|
| Sid | int (11) NOT NULL | PRIMARY KEY |
| name | varchar (200) NULL | |
| services | varchar (200) NULL | |
| email | varchar (200) NULL | |
| phone | varchar (200) NULL | |
| latitude | varchar (200) NULL | |
| longitude | varchar (200) NULL | |
| description | varchar (1000) NULL | |

Table 4.9**Table: vaccination**

| Field | Type | Comment |
|-------------|--------------------|-------------|
| void | int (11) NOT NULL | PRIMARY KEY |
| hid | varchar (100) NULL | FOREIGN KEY |
| name | varchar (100) NULL | |
| description | varchar (500) NULL | |
| type | varchar (100) NULL | |
| do | varchar (100) NULL | |

Table 4.10

Table: vaccine request

| Field | Type | Comment |
|----------------|--------------------|-------------|
| vied | int (11) NOT NULL | PRIMARY KEY |
| aid | int (11) NULL | FOREIGN KEY |
| hid | int (11) NULL | FOREIGN KEY |
| void | varchar (100) NULL | FOREIGN KEY |
| requested date | varchar (50) NULL | |
| status | varchar (30) NULL | |
| booking date | varchar (100) NULL | |
| booking time | varchar (100) NULL | |

Table 4.11

5. CODING AND TESTING

5.1 INTRODUCTION

The objective of the coding phase is to transform the design of a system into code in a high-level language and then to unit test this code. Good software development organizations normally require their programmers to adhere to some well-defined and standard style of coding called coding standards. Coding Standards-A coding standard gives a uniform appearance to the codes written by different engineers. It enhances code understanding. It encourages good programming practices. Testing a program consists of providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected. Aim of testing The aim of the testing process is to identify all defects existing in a software product. Some commonly used terms associated with testing are: Failure: This is a manifestation of an error (or defect or bug). Test case: This is the triplet [I,S,O], where I is the data input to the system, S is the state of the system at which the data is input, and O is the expected output of the system. Test suite: This is the set of all test cases with which a given software product is to be tested.

5.2 CODING

```
package com.syntax.startup_healthy;

import android.Manifest;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.syntax.startup_healthy.HOSPITAL.Hospital;
import com.syntax.startup_healthy.USER.User;

import java.util.HashMap;
import java.util.Map;
```

```

public class LoginActivity extends AppCompatActivity {
    EditText username, password;
    Button btnlog;
    TextView reg;
    String UNAME, PASS;
    AlertDialog.Builder builder;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
//        requestWindowFeature(Window.FEATURE_NO_TITLE);
//        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
//                               WindowManager.LayoutParams.FLAG_FULLSCREEN);

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        ActionBar actionBar = getSupportActionBar();
        actionBar.hide();

        int PERMISSION_ALL = 1;
        String[] PERMISSIONS = {
            Manifest.permission.WRITE_EXTERNAL_STORAGE,
            Manifest.permission.READ_EXTERNAL_STORAGE,
            Manifest.permission.ACCESS_FINE_LOCATION,
            Manifest.permission.ACCESS_COARSE_LOCATION,
            Manifest.permission.CAMERA,
            Manifest.permission.READ_SMS,
            Manifest.permission.SEND_SMS,
            Manifest.permission.READ_PHONE_STATE
        };

        if (!hasPermissions(this, PERMISSIONS)) {
            ActivityCompat.requestPermissions(this, PERMISSIONS, PERMISSION_ALL);
        }

        builder = new AlertDialog.Builder(this);

        username = (EditText) findViewById(R.id.input_Lemail);
        password = (EditText) findViewById(R.id.input_Lpassword);
        btnlog = (Button) findViewById(R.id.user_loginButton);
        reg = (TextView) findViewById(R.id.Userlink_register);

        reg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(),
RegistrationActivity.class));
            }
        });

        btnlog.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Validate();
            }
        });
    }

    @Override
    public void onBackPressed() {

        builder.setMessage("Do you want to close this application ?")
            .setCancelable(false)

```

```

        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                finish();
            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                // Action for 'NO' Button
                dialog.cancel();
            }
        });
    //Creating dialog box
    AlertDialog alert = builder.create();
    //Setting the title manually
    alert.setTitle("FuelSpot");
    alert.show();

    alert.getWindow().setBackgroundDrawable(new
    ColorDrawable(Color.parseColor("#FFFFFF")));
}

private void Validate() {

    UNAME = username.getText().toString();
    PASS = password.getText().toString();

    if (UNAME.isEmpty()) {
        username.requestFocus();
        username.setError("enter your email ID");
    } else if (PASS.isEmpty()) {
        password.requestFocus();
        password.setError("enter password");
    } else {
        login();
    }
}

private void login() {

    com.android.volley.RequestQueue queue =
    Volley.newRequestQueue(getApplicationContext());

    StringRequest request = new StringRequest(Request.Method.POST,
    Utility.SERVERUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("*****", response);

            if (!response.trim().equals("failed")) {

                String data = response;
                String respArr[] = data.trim().split("#");

                SharedPreferences.Editor editor =
                getSharedPreferences("SharedData", MODE_PRIVATE).edit();
                editor.putString("u_id", "" + respArr[0]);
                editor.putString("type", "" + respArr[1]);
                editor.commit();

                if (respArr[1].trim().equals("HOSPITAL")) {
                    startActivity(new Intent(getApplicationContext(),
                    Hospital.class));

                    Toast.makeText(getApplicationContext(), "Login
                    Successful", Toast.LENGTH_SHORT).show();
                } else if (respArr[1].trim().equals("CUSTOMER")) {
                    startActivity(new Intent(getApplicationContext(),

```

```

User.class));
        Toast.makeText(getApplicationContext(), "Login
Successful", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(LoginActivity.this, "Something Went
Wrong", Toast.LENGTH_SHORT).show();

    }

    } else {
        Toast.makeText(getApplicationContext(), "Login Failed",
Toast.LENGTH_SHORT).show();
    }
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

        Toast.makeText(getApplicationContext(), "my Error :" + error,
Toast.LENGTH_LONG).show();
        Log.i("My Error", "" + error);
    }
}) {
    @Override
    protected Map<String, String> getParams() {

        Map<String, String> map = new HashMap<String, String>();
//        SharedPreferences sp=getSharedPreferences("booking_info",
Context.MODE_PRIVATE);
        map.put("key", "login");
        map.put("U_name", UNAME);
        map.put("P_swd", PASS);

        return map;
    }
};
queue.add(request);
}

public static boolean hasPermissions(Context context, String... permissions)
{
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M && context
!= null && permissions != null) {
        for (String permission : permissions) {
            if (ActivityCompat.checkSelfPermission(context, permission) !=
PackageManager.PERMISSION_GRANTED) {
                return false;
            }
        }
    }
    return true;
}
}
}

```

SPLASH

```

package com.syntax.startup_healthy;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;

```



```
import android.os.Handler;
import android.view.Window;
import android.view.WindowManager;

public class SplashActivity extends AppCompatActivity {
    Handler handler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.activity_main);

        ActionBar actionBar = getSupportActionBar();
        actionBar.hide();

        handler=new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent=new Intent(getApplicationContext(), LoginActivity.class);
                startActivity(intent);
                finish();
            }
        },3000);
    }
}
```

5.3 UNIT TESTING

Developers typically do unit testing in order to trace out bugs in each module of the code. Unit testing is done in parallel with coding. It includes testing each function and procedures. Unit testing is also called as module testing. In module testing each module are tested for any possible logical error. They are also tested for specification to see if they are working as per the program should do and they are tested under various conditions. Each module is being tested thoroughly in order to discover pitfalls. Specification testing examines the specification what the program should do and how it should perform under various conditions. The testing will be done by entering data into different tables using forms. The data with less validation will be tested first. Whenever an error is encountered, an informative error message will be displayed which informs user about the type of error. After the completion of form testing the program will be tested.

The unit testing is done to identify:

- The image entries are in the correct format.
- No duplicate entries are present.
- To check whether it provide the required result.

5.4 INTEGRATION TEST

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). The task of the integration test is to check that components or software applications, interacts without error. Therefore, testing the data flow between 2 modules is integration testing.

5.5 VALIDATION TEST

The validation phase reveals the failures and the buds in the developed system. It will become to known about the practical difficulties the system faces when the operated in the true environment. Validation is the process of ensuring that user input is clean, correct, and useful

Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid email?
- Has the user entered text in a numeric field?

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the submit button. If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process which used to put a lot of burden on the server. Most often, the purpose of validation is to ensure correct user input. Validation can be defined by many different methods, and deployed in many different ways. Server-side validation is performed by a web server, after input has been sent to the server. Client-side validation is performed by a web browser, before input is sent to a web server.

5.6 FINAL SYSTEM TESTING

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems.

System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system. Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario. Verify thorough testing of every input in the application to check for desired outputs. Testing of the user's experience with the application. That is a very basic description of what is involved in system testing. You need to build detailed test cases and test suites that test each aspect of the application as seen from the outside without looking at the actual source code.

Different Types of System Testing:

- Usability Testing- mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives.
- Load Testing- is necessary to know that a software solution will perform under real-life loads.
- Regression Testing- involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
- Recovery testing - is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.

- Migration testing- is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
- Functional Testing - Also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.
- Hardware/Software Testing - IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing

6. SYSTEM IMPLEMENTATION AND MAINTENANCE

6.1 SYSTEM SECURITY MEASURES

Security of a system is a crucial task. It is a process of ensuring confidentiality and integrity of the OS. A system is said to be secure if its resources are used and accessed as intended under all the circumstances, but no system can guarantee absolute security from several of the various malicious threats and unauthorized access. Security of a system can be threatened via two violations:

Threat: A program which has the potential to cause serious damage to the system.

Attack: An attempt to break security and make unauthorized use of an asset.

Security violations affecting the system can be categorized as malicious and accidental. Malicious threats, as the name suggests are a kind of harmful computer code or web script designed to create system vulnerabilities leading to back doors and security breaches. Accidental Threats, on the other hand, are comparatively easier to be protected against. Example: Denial of Service DDoS attack.

Security is an important property of any software. Many applications are outsourced too where the application development lacks strong integration of software security. The growing need to address software security measures across development life cycle has been discussed here. Application Security can be seamlessly integrated in the SDLC by introducing specific steps or process within the development phases.

6.2 IMPLEMENTATION DETAILS

The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented. Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right. The implementation stage involves following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of method to achieve change over.
- Evaluation of the changeover method.

6.3 MAINTENANCE

System maintenance is a going activity, which covers a wide variety of activities including, removing program and design errors, updating documentation and test data and updating user support system maintenance is a catchall term used to describe various forms of computer or server maintenance required to keep a computer system running properly, it can describe network maintenance which could mean that servers are being physical repaired, replaced or mode. For the purpose of convenience, maintenance may be categorized into three classes they are:

6.3.1 CORRECTIVE MAINTENANCE

This type of maintenance implies removing errors in a program, which might have kept in the system due to faulty design or wrong assumption.

6.3.2 ADAPTIVE MAINTENANCE

In adaptive maintenance program functions are changed to enable the information system to satisfy the information needs of the user.

6.3.3 PERFECTIVE MAINTENANCE

In perfective maintenance means adding new programs or modifying the existing programs to enhance the performance of the information system. This type of maintenance under taken to respond to user addition needs which may be due to the changes within or outside of the organic- nation.

7. FUTURE ENHANCEMENT

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the advent of worldwide influence and shifting societal values, technological development is accelerating the transformation of the healthcare industry. Today we can clearly see which parts of health care delivery have failed and shown to be vulnerable.

This work can be extended by adding functionality such as:

1. 24/7 Connect with the Doctor:

Making an appointment with a doctor in a few clicks online is one of the solutions. So the users need not to wait for the appointments and also connect online so physical contact can be reduced

2. Marketing platform:

Like Amazon, Flipkart we can extend our work and can make marketing platform available for retailers and hospitals from where they can directly sell or purchase the medicines on their own.

3. Medical Records

Make the system to store the complete medical record of a user digitally and it makes the doctor easy to get the medical records of a user at his finger tap

8. CONCLUSIONS

This system timely updates the complete reference related to the current pandemic situation. With the help of this system the user can get all the information like the current TPR rate, Timely updates of the pandemic and to locate the nearest covid care centers. You can get easy access to an emergency service also. The system aims at improving the existing system and providing an efficient way for managing current pandemic situation. It is a time-saving as well as cost-efficient application. The proposed system can be used to reduce human efforts and easy collect and store the information related to the pandemic in digital form and also it can be useful for further researches. It makes us to control the pandemic with modern technology. Helps to get an aware of all the vaccinations.

9. APPENDIX

ADMIN PANEL

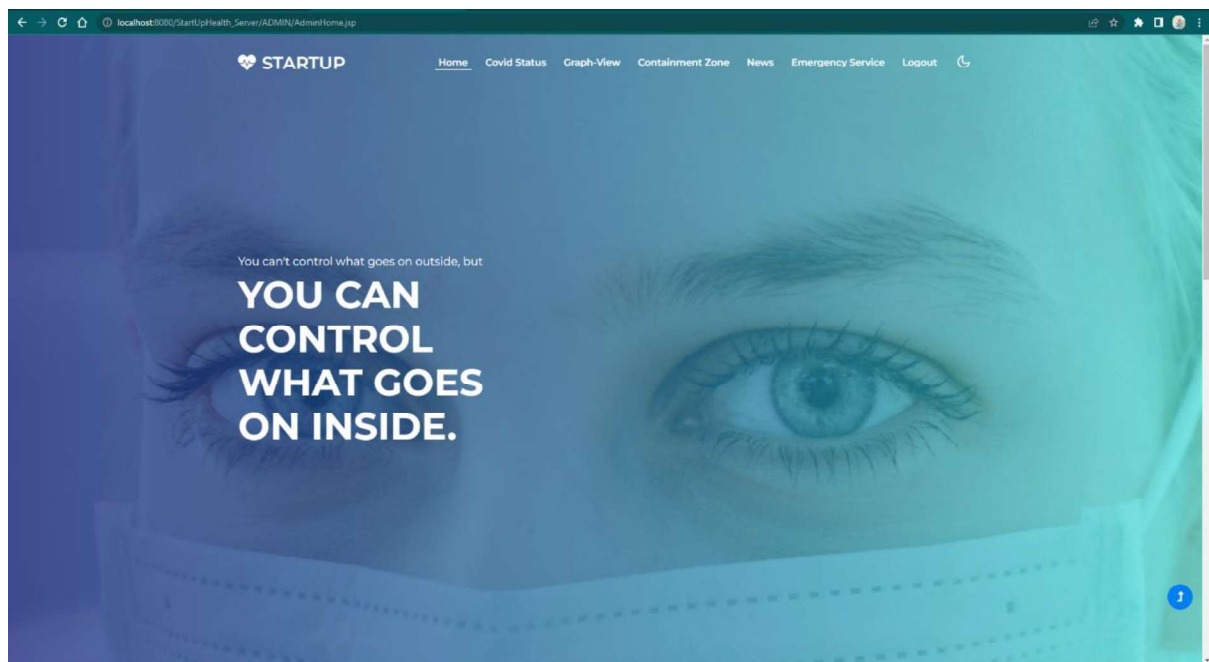


Fig. 9.1

GRAPH VIEW

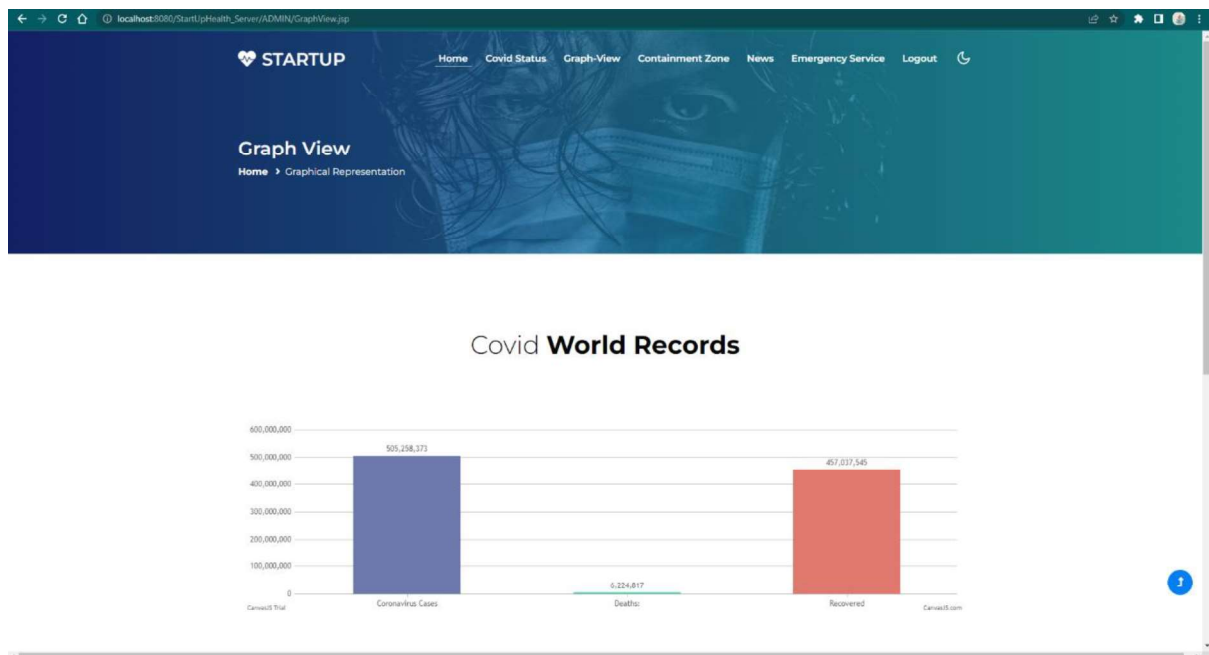


Fig. 9.2

CONTAINMENT ZONE

Fig 9.3

NEWS & GUIDELINES

Fig 9.4

LOGIN FORM

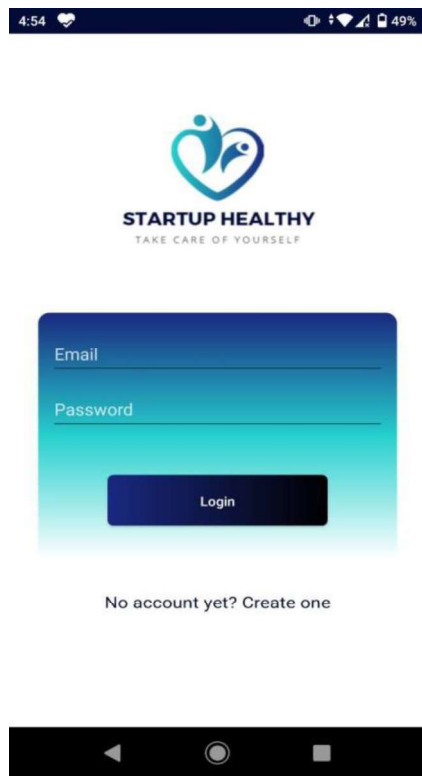


Fig. 9.5

USER HOME



Fig. 9.7

NEW REGISTRATION

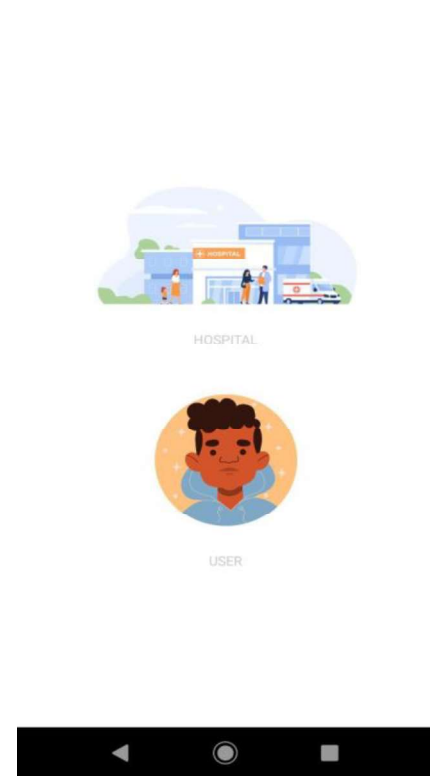


Fig. 9.6

COVID CARE CENTERS

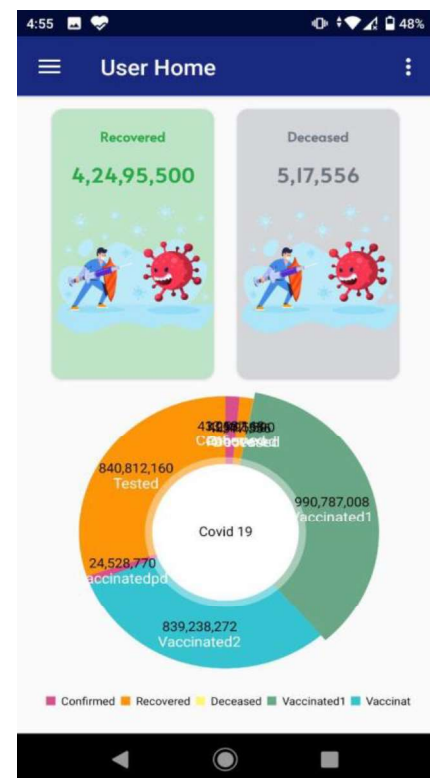


Fig 9.8

NEWS & GUIDELINESS



Fig 9.9

HOSPITAL HOME

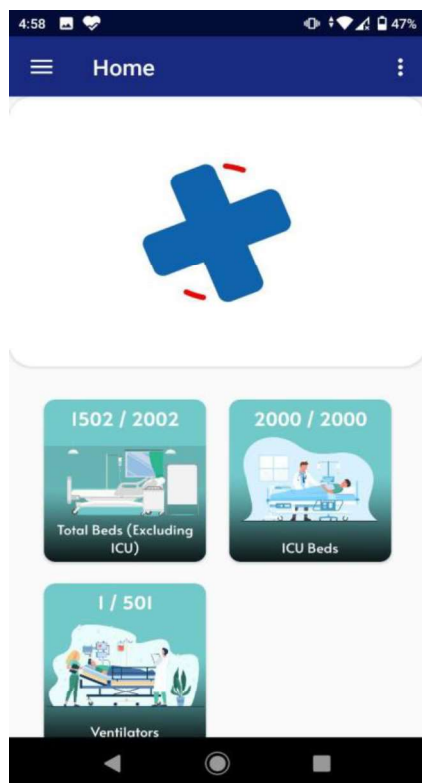


Fig 9.11

EMERGENCY SERVICES



Fig 9.10

VACCINE DETAILS

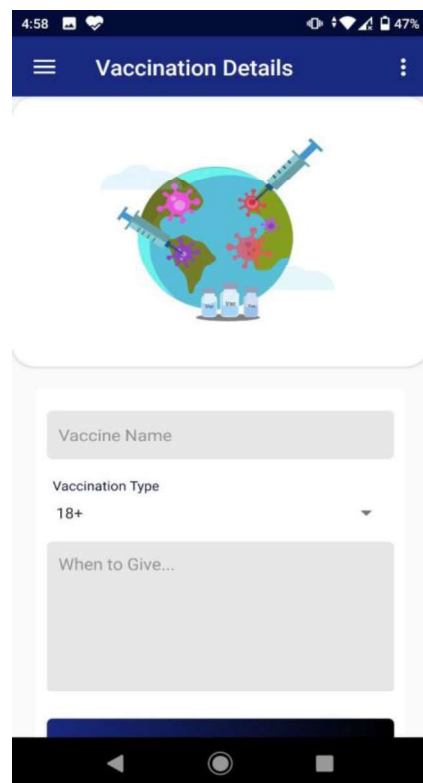


Fig 9.12

10. BIBLIOGRAPHY

- System Analysis and Design - Elias Mewed
- An integrated Approach of Software Engineering - Pankaj Jalote
- Internet and Web Technology - Raj Kamal
- Introduction to Database Management - Naveen Prakash
- Analysis and Design of Information System - V. Rajkumar
- Web Programming - Chris Bates