TEXTILE SHOP MANAGEMENT SYSTEM

MINI PROJECT – I

Submitted by

PRASATH S

19MSS027

Under the Guidance of

ANU P

Assistant Professor

Department of Software Systems

In partial fulfillment of the requirements for the award of the degree of

MASTER OF SCIENCE IN SOFTWARE SYSTEMS

(Five years Integrated Course)

of Bharathiar University

DEPARTMENT OF SOFTWARE SYSTEMS

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous college - Affiliated to Bharathiar University
Accredited with 'A'grade by NAAC (3rd Cycle)
College with Potential for Excellence
(Status Awarded by the UGC)
Star College Status Awarded by DBT - MST
An ISO 9001:2015 Certified Institution
Coimbatore - 641 014

(December 2020)

DEPARTMENT OF SOFTWARE SYSTEMS

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous college - Affiliated to Bharathiar University
Accredited with 'A'grade by NAAC (3rd Cycle)
College with Potential for Excellence
(Status Awarded by the UGC)
Star College Status Awarded by DBT - MST
An ISO 9001:2015 Certified Institution
Coimbatore - 641 014

CERTIFICATE

This is to certify that this project work entitled **"SPORTS MANAGEMENT SYSTEM"** is a bonafide record of work done by **SELVAGNAPATHY K(19MSS041)** in partial fulfillment of the requirements for the award of Degree of **Master of Science in Software Systems** (Five years Integrated Course) of Bharathiar University.

Faculty Guide

Head of the Department

Submitted for Viva-Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SELVAGANAPATHY K(19MSS041)**, hereby declare that this project work entitled

"SPORTS MANAGEMENT SYSTEM" is submitted to PSG College of Arts &Science, Coimbatore in partial fulfillment of the requirements for the award of the degree of Master of Science in Software Systems, is a record of original work done by me under the supervision and guidance of **Ms Jerlin Adaikala Sundari J MCA MPhil , Assistant Professor,** Department of Software Systems, PSG College of Arts &Science, Coimbatore.

This report has not been submitted by me for the award of any other Degree/ Diploma/ Associate ship/ Fellowship or any other similar degree to any other university.

Place: Coimbatore **(SELVAGANAPATHY K)**

Date: **(18MSS011)**

ACKNOWLEDGEMENT

My venture stands imperfect without dedicating my gratitude to a few people who have contributed a lot towards the victorious completion for my project work.

I would like to thank **Thiru L. Gopalakrishnan, Managing Trustee, PSG & Sons Charities**, for providing me a prospect and surroundings that made the work possible.

I take this opportunity to express my deep sense of gratitude to **Dr. T. Kannaian, Secretary**, PSG College of Arts &Science, Coimbatore for permitting and doing the needful towards the successful completion of this project.

I express my deep sense of gratitude and sincere thanks to **Dr. D. Brindha M.Sc., M.Phil., Ph.D.**, **MA (Yoga).,** Principal, PSG College of Arts &Science, Coimbatore for her valuable advice and concern on students.

I am very much thankful to **Dr. A. Anguraj, M.Sc., M.Phil., Ph.D**., Vice Principal (Academics), **Dr.Jayanthi M M.Com., MBA., M.Phil., Ph.D.,**Vice Principal (Student Affairs), **Prof. M Umarani, MBA, M.Phil., Faculty-In-Charge (Student Affairs)**, for their support towards my project.

I sincerely thank **Dr. K.V.Rukmani., M.C.A., M.E, Ph.D.,** Head, Department of Software Systems for her whole hearted help to complete this project successfully by giving valuable suggestions.

I convey my heartiest and passionate sense of thankfulness to my project guide **Ms Jerlin Adaikala Sundari J MCA MPhil , Assistant Professor**, Department of Software Systems, for her/his timely suggestion which had enable me in completing the project successfully.

I am grateful to _____, external guide, _____ , for giving me an expert's insight into the project and for sparing time in his/her hectic agenda, for planning and giving me invaluable guidance and suggestions.

This note of acknowledgement will be incomplete without paying my heartful devotion to my parents, my friends and other people, for their blessings, encouragement, financial support and the patience, without which it would have been impossible for me to complete the job.

(COMPANY CERTIFICATE ORIGINAL)

(SYNOPSIS)

The Textile shop management is a report that highlights the importance of maintaining the textile industry. This report can easily focus on how the working of textile shop takes place easily. The Textile shop management system report can easily emphasize how the textile shop works. Most of the works at the textile shop is easily automated using some automation.

All the discounts that are available at the textile shops are easily highlighted through this report without any issue.

TABLE OF CONTENTS

1.INTRODUCTION

The Textile management system application is developed for managing the textile shop. The idea of textile shop development is how to manage the textile shop in a good manner or we can say managing the textile shop well from which people can get profit or just stay out from the difficulties, how the things is proper in the shopping mall, what is the input in the shopping mall and what is the output how to track the goods are available there or which is sort.

All this is auto track by the application from which there will be no any difficulties facing by the management after all there are certain report generation based on the shopping mall daily turnover, monthly turnover etc

1.1 PROJECT OBJECTIVE

The system is fabricated around the overall objectives of providing a flexible, user-specific work environment within management defined limits. It aims to execute better task control, coherence and efficient time utilization along with stringent data security and integrity. Textile Management System goes beyond conventional office automation to provide strategic management analysis, planning and decision support tools based on the critical routine and operational variables prevalent in the textile work culture.

1.2 PROJECT OVERVIEW

The Textile Management System is designed to allow the industry to keep track of all employee details, textile details, product details, banquet details and agent details. It keeps tracks of active employees as well as employees who have left the textile industry.

1.3 MODULES

- USER MODULE
- ITEM MODULE
- PURCHASE MODULE
- REPORT MODULE

1.3.1 MODULES DESCRIPTION

- USER: This module contains add the new dealer details. It Includes are User id, name, date of birth, address, mobile number, e-mail id, password and so on.
- ITEM: This module contains Add the new item it includes item name, id, date, unit price, manufacturing date details maintained. And view the item details, edit the item details, delete the unwanted item details maintained.
- PURCHASE: This module contains are user purchase the item details maintained. It includes are item name, purchasing date, no of quantity, rate details and so on.
- REPORT: This Module Contains two sub Module. They are a. User Report- user can view their past purchase history and b. Admin Report- Admin can view the user history and purchase history.

2. SYSTEM ANALYSIS

System study is classified into two types

- Existing system and
- Proposed system

2.1 EXISTING SYSTEM

The system, which is followed at present, it is manual system. Important drawback of existing system is time factor. It will not help the management to solve the problem in time

Drawbacks of Existing System:

- ➔ Manual Work
- ➔ Security of information is low
- ➔ A lot of time consumed
- ➔ Needs a lot of manpower
- ➔ Frequent occurrence of error
- ➔ Calculations are difficult

Our **Textile Management System** provides a easy way to purchase the clothes through the on line. It's the simplest way to purchase dresses in the network

2.2 PROPOSED SYSTEM

- The drawbacks which are faced during existing system, can be eradicated by using the proposed system, the main objective of the existing system is to provide an Electronic User-friendly Interface
- The systems which are proposed, Now computerizes all the details that are maintained manually
- Once the details are fed into the computer There is no need for various persons to deal with
- Separate sections.
- Only a single person is enough to maintain all the reports.

Benefits of the proposed system:

➔ Large volumes of data can be stored with case
➔ Security is assured
➔ Maintenance of file is flexible
➔ Records Stored are updated now and then
➔ Stored data and procedures can be easily edited
➔ Reports can be generated with case
➔ Accurate calculations are made
➔ Less manpower required

3.SYSTEM CONFIGURATION

3.1 HARDWARE SPECIFICATION

- Processor : x86 compatible processor with 1.7 GHz Clock Speed
- Hard disk : 20 GB or greater
- Ram : 512 MB or greater
- Keyboard : 104 Standard keys
- Mouse : 2/3 button. Optical/ Mechanical.

3.2 SOFTWARE SPECIFICATION

- Operating Systems : Windows 2000/XP/Vista
- Front End Software : HTML, CSS & BOOTSTRAP
- Back End Database : MySQL 8.0
- Server-Side Script : XAMPP 7.3.25
- IDE : Visual Studio Code 1.56

4.SOFTWARE DESCRIPTION

FILE DESIGN

This chapter is about the Software languages and the tools used in the development of the project. The Primary Languages are PHP and MYSQL.

FEATURES OF PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.PHP is an interpreted scripting language that is embedded within an HTML web page in order to add dynamic processing to that page.

PHP is supported by a wide range of commercial and open-source web servers, including Red Hat Linux, and can also be installed as an Apache module. Its widespread availability and its relative simplicity mean that it is an excellent way to introduce dynamic features into your web pages. As it is an open, non-proprietary standard, PHP developers are not restricted by the limitations imposed by some commercial suppliers of server-side scripting software, neither do they have to purchase expensive licenses in order to use it.

You may already be familiar with 'client-side' scripting languages such as JavaScript. If you include JavaScript in your page, then the JavaScript code is downloaded to the client's browser and executed there. PHP is different in that it is strictly a 'server-side' scripting language

- this means that the PHP is always processed by the web server before the requested page is served to the browser. The PHP tags in the page are replaced by generated HTML strings and the client's browser then displays the HTML without any knowledge of the underlying PHP code at all.

The syntax of the language is similar to C, so anyone who is familiar with the C programming language, (or Perl or Java, for that matter) should be able to master PHP scripting quickly and without too much difficulty. Object Orientated Programming extensions have been introduced with the latest release of PHP which allow you to use objects within a PHP script.

PHP can be used to do anything that any CGI program can do, such as:

- Collect and process form data
- Generate dynamic page content
- Send and receive cookies

One of PHP's biggest strengths is its ability to interface with databases. PHP supports a wide range of databases, including proprietary (such as Sybase and Oracle) and open-source (such as MySQL and PostgreSQL). The complete list is growing all the time. PHP can also communicate with other processes using a variety of standard protocols.

The main disadvantage of PHP is that it is an interpreted language and therefore there is inevitably a overhead in processing a page of PHP script. However, no PHP code is ever downloaded to the client's browser so there is never any question of a user 'stealing' your PHP code and adapting it for his own purposes.

PHP is a very powerful program that can access files, execute commands and open network connections on the server. These features would make anything being run on the server insecure by default. Because PHP is specifically designed to operate on web servers then it is intrinsically more secure that 'general purpose' languages such as C or Perl. PHP has a number of different configuration options which give the web server manager the ability to set precisely the level of security that is needed for the situation.

UNIQUE FEATURES OF PHP

PHP language has support features of other languages like c, Perl and etc. It also has some unique features of its own. Some of them are listed below in this article.

1. In PHP there is no need to specify data type for variable declaration. Rather, it can be determined at the time of execution depends on the value of the variable. So that, PHP is called as loosely typed language.
2. PHP provides cross platform compatibility, unlike some other server side scripting language.
3. PHP has set of pre-defined variables called superglobals which will be start by _. Some of the examples are, $_GET, $_POST, $_COOKIE, $_SESSION, $_SERVER and etc. So, any variable except superglobals, that are start with _ will cause error.
4. PHP programming structure includes variable variables; that is, the name of the variable can be change dynamically.
5. This language contains access monitoring capability to create logs as the summary of recent accesses.
6. And then, it includes several magic methods that begins with __ character which will be defined and called at appropriate instance. For example, __clone() will be called, when the clone keyword is used.
7. Predefined error reporting constants are available to generate a warning or error notice. For example, when E_STRICT is enabled, a warning about deprecated methods will be generated.
8. PHP supports extended regular expression that leads extensive pattern matching with remarkable speed.
9. And then, properties like, nowdocs and heredocs are used to delimit some block of context which should not be sent for parsing.
10. Since PHP is a single inheritance language, the parent class methods can be derived by only one directly inherited sub class. But, the implementation of traits concept, reduce the gap over this limitation and allow to reuse required method in several classes.

DATABASE MANAGEMENT SYSTEM

Following the technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the CODASYL model (network model), implemented in a number of products such as IDMS. The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

MySQL

MySQL is an open-source relational database Management System (RDBMS).
MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a homebrewed lexical analyzer. MySQL works on many system platforms,
including Linux, macOS, Microsoft Windows, NetBSD. MySQL is offered under two different editions: the open-source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.
Major features that are available in MySQL are a broad subset of ANSI SQL 99, as well as extensions, Cross-platform support, Stored procedures, using a procedural language that closely adheres to SQL/PSM, Triggers, Cursors, Updatable views, Online DDL when using the InnoDB Storage Engine. Many programming languages with languages specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio and the JDBC driver for Java. In addition, an ODBC interface called MySQL Connector/ODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion.

XAMMP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.
XAMPP's ease of deployment means a XAMPP stack can be installed quickly and simply on an operating system by a developer, with the advantage a number of common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami.

5.SYSTEM DESIGN

5.1 DATA FLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

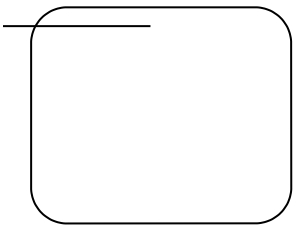
In the DFD, there are four symbols

1. **A square** defines a source(originator) or destination of system data

2. An **arrow** identifies data flow. It is the pipeline through which the information flows

3. A **circle or a bubble** represents a process that transforms incoming data flow into outgoing data flows.

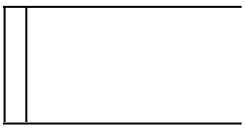
4. An **open rectangle** is a data store, data at rest or a temporary repository of data



Process that transforms data flow.

Source or Destination of data

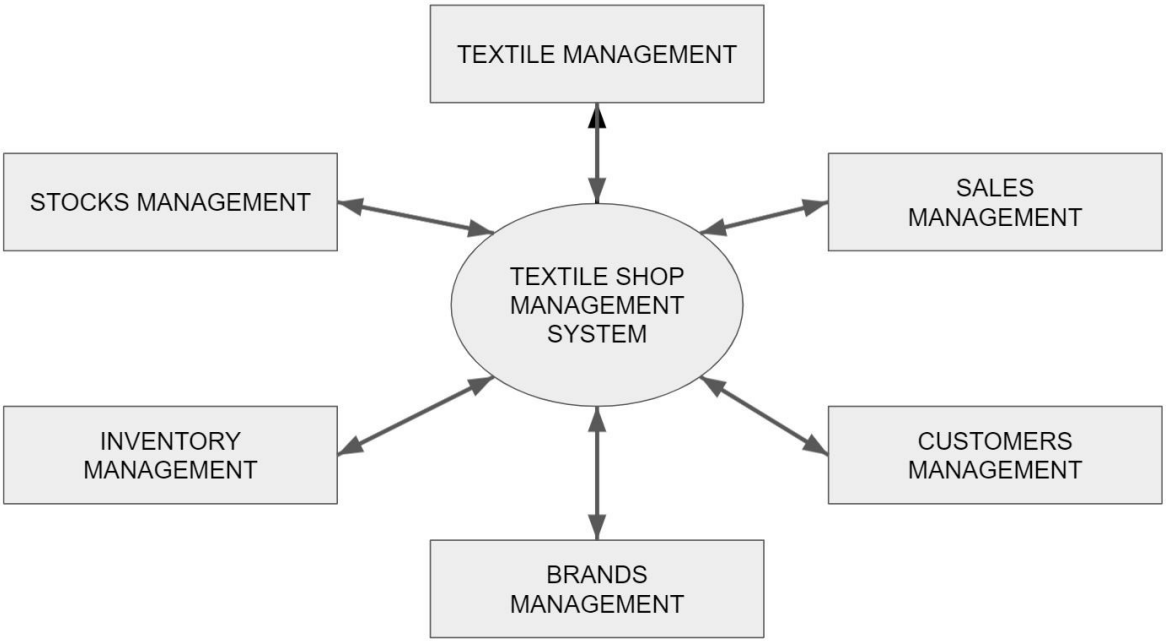
Data flow



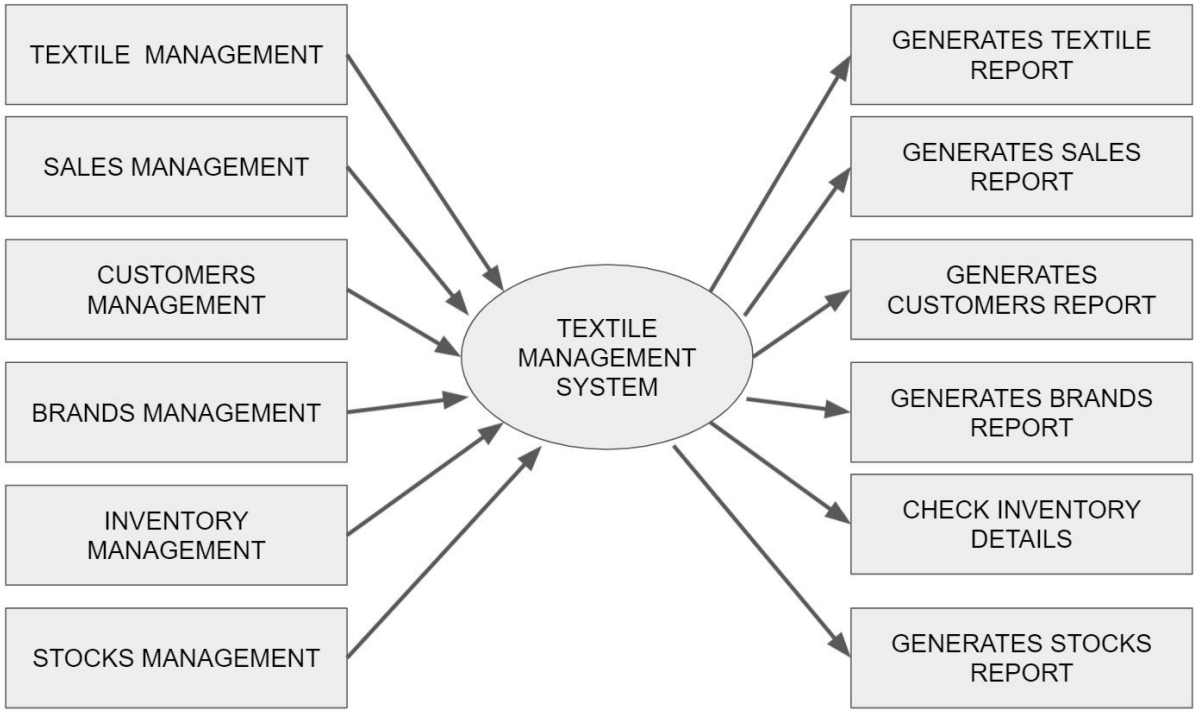
Data Store

A. DATA FLOW DIAGRAM

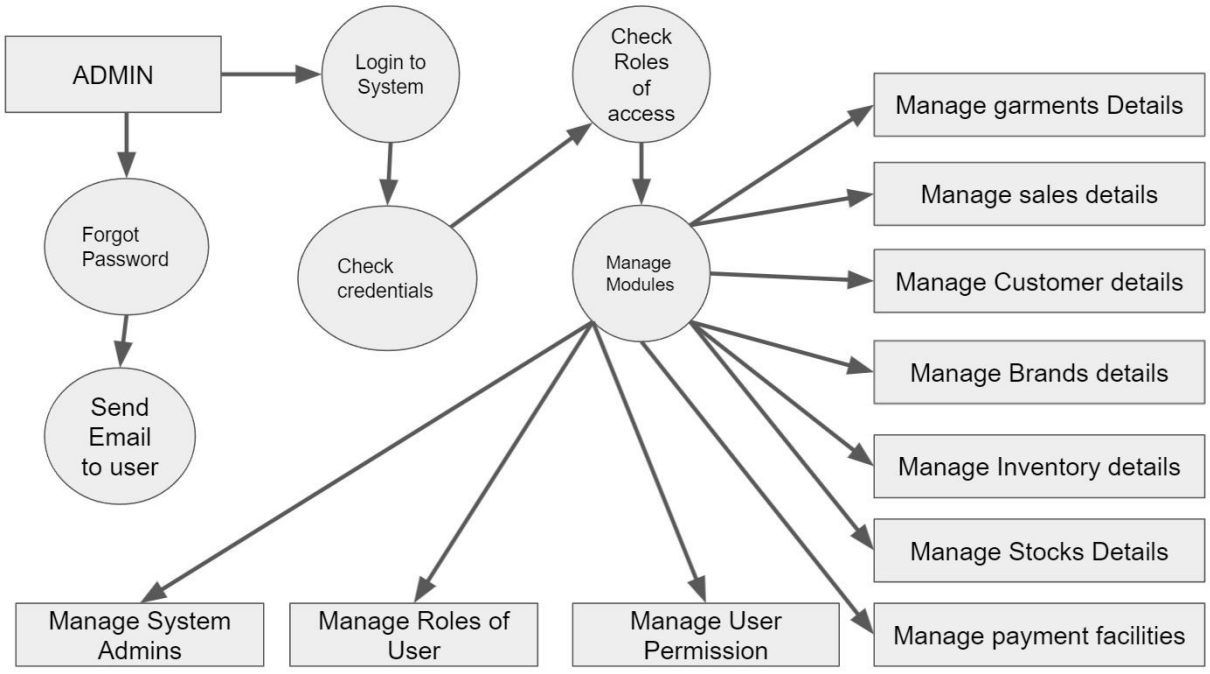
LEVEL – 0 DFD DIAGRAM FOR TEXTILE MANAGEMENT SYSTEM



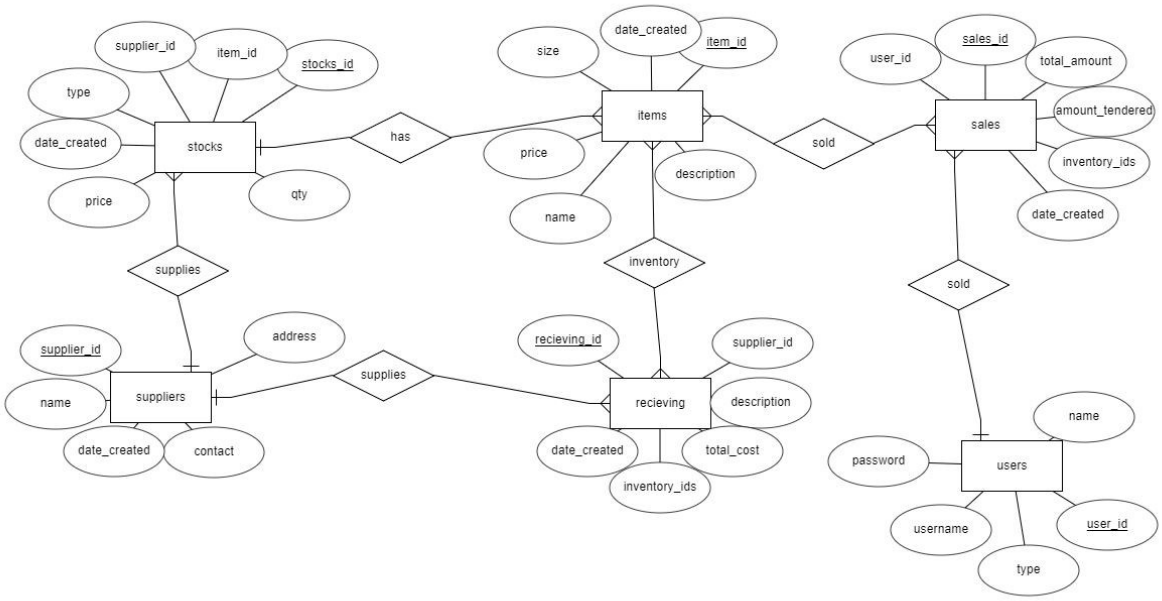
LEVEL – 1 DFD DIAGRAM FOR TEXTILE MANAGEMENT SYSTEM



LEVEL – 2 DFD DIAGRAM FOR TEXTILE MANAGEMENT SYSTEM



5.2 ER DIAGRAM



5.3.TABLE DESIGN

i)User Login:

Description : This table is to Store the Login details of the User.

FIELD	DATA TYPE	SIZE	CONSTRAINT
User_id	int	30	primaryKey
name	varchar	200	Not null
Username	varchar	200	Not null
password	Varchar	20	Not null
type	tinyint	1	Not null

ii) Items:

Description : This table is to Store the item details.

FIELD	DATA TYPE	SIZE	CONSTRAINT
Item_id	int	30	primaryKey
Item_code	varchar	100	Not null
name	varchar	200	Not null
description	varchar	200	Not null
size	Varchar	10	Not null
price	float	10	Not null
Date_created	datetime		Not null

iii) Stocks:

Description : This table is used to store the Stocks available

FIELD	DATA TYPE	SIZE	CONSTRAINT
Stocks_id	int	30	primaryKey
Item_id	varchar	100	Foreign Key
type	tinyint	1	Not null
qty	int	30	Not null
price	Float	10	Not null
Date_created	datetime		Not null
supplier_id	varchar	30	Foreign Key

iv) Suppliers:

Description : This table is used to store the Supplier details

FIELD	DATA TYPE	SIZE	CONSTRAINT
Supllier_id	int	30	primaryKey
name	varchar	200	Not null
address	Varchar	200	Not null
conatct	Varchar	50	Not null
Date_created	datetime		Not null

V) System Settings :

Description : This table is used to store System settings

FIELD	DATA TYPE	SIZE	CONSTRAINT
id	int	30	primaryKey
name	varchar	200	Not null
email	varchar	200	Not null
conatact	Varchar	10	Not null
Cover_img	varchar	100	Not null
About_content	varchar	200	Not null

Vi)Sales :

Description : This table is used to store Sales details

FIELD	DATA TYPE	SIZE	CONSTRAINT
Sales_id	int	30	primaryKey
User_id	int	30	Foreign Key
Total_amount	float	20	Not null
Amount_tendered	int	30	Not null
Inventory_ids	Varchar	10	Foreign Key
Date_created	datetime		Not null

vii)Receiving:

Description : This table is used to store receiving product details

FIELD	DATA TYPE	SIZE	CONSTRAINT
recieving_id	int	30	primaryKey
Supplier_id	int	30	Foreign Key
description	varchar	200	Not null
Total_cost	Float	10	Not null
Inventoty_ids	varchar	10	Foreign Key
Date_created	datetime		Not null

5.4 INPUT DESIGN

The input design is the link that ties the Information system into the world of its users. It is a process of converting user-originated inputs to a computer based format. Input data are collected and organized into a group of similar data. Once identified, appropriate input media are selected for processing.

The goal of designing input data is to make entry easy, logical and free form errors. In input data design, we design source document that capture the data and then select the media used to enter them into the computer. The input forms are developed in a user-friendly way so that a layman also can easily understand everything. Menus are provided to users and different icons are designed so the proposed system design looks decorative.

Input design is the part of the overall system design. Source documents initiate a processing cycle as soon as they are entered into the system through the keyboard. A source should be logical and easy to understand.

Objectives of input Design:

- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

5.4.1. DATABASE DESIGN

If you do not know, a database is a place to store information used by software applications. For example, you could have a web page with a list of companies and all their locations with contact information for each location. Or a banking application on your computer to sort and manage your check book. In both cases, it makes sense to store the data in one piece of software called a database. The database has a structure and rules about how to add, edit, delete, and read data stored in the database.

Databases also reside in many different places. Some databases exist on only your computer. Other databases have their data shared and divided across hundreds or thousands of databases located in many data centres all over the world.

NoSQL database systems are today an effective solution to manage large data sets distributed over many servers. A primary driver of interest in NoSQL systems is their support for next-generation web applications, for which relational DBMSs are not well suited. These are OLTP applications for which data have a structure that does not fit well in the rigid structure of relational tables, access to data is based on simple read write operations, scalability and performance re important quality requirements, and a certain level of consistency is also desirable. NoSQL technology is characterized by a high heterogeneity, which is problematic to application developers. Currently, database design for NoSQL systems is usually based on best practices and guidelines.

For example, a NoSQL database works best for applications with massive amounts of data where most activity involves reading data from the database with some writing of data to the database. Reading is less intensive than writing because writing data to a database requires tracking when a database table is open. NoSQL databases tend to be on multiple machines and, in some cases, machines in multiple data centres. Keeping data in sync is comparatively easier and less complex with NoSQL databases. Even in cases where large data sets are not involved, some developers prefer the easier interactions between their code and a NoSQL database.

DATA MODELS

The first step in any database design is the creation of a data model. The model distils all the functionality requirements for an application into data collections, for example, products, customers, and suppliers for an ecommerce site, as well as properties and relationships between these collections of data

There are several risks data models help limit or avoid:

Business processes sometimes can be duplicated in the database structure, creating problems if a process changes. A good data model provides flexibility independent from any process.

Needless duplicated table in multiple locations within the same database. This is a big issue in relational databases.

Data models for related applications differ for no reason. Ideally, a data model takes into account other applications used by the business or individual.

Data might be difficult to extract or share with other software applications. If data sharing is important, a data model should ensure data can be extracted easily.

Database and data models typically are represented as graphs. Early stages of development, however, use business requirements and functional specifications to clarify the system a data model must represent and support. In some cases, for example, health care or finance, there may be examples of data models used widely which are adopted or adapted.

The data model also is one of several factors in the decision about what database management system (DBMS) to use, relational or NoSQL.

NOSQL DATABASE DESIGN

Key-value pairs are the main feature of these databases. Keys are names or unique ID numbers and values range from simple data to documents to columns of data to structured lists (arrays) of key-value data. Each row in a NoSQL table includes the key and its value. The design of NoSQL databases depends on the type of database, called stores:

Document Stores pair each key identifier with a document which can be a document, key-value pairs, or key-value arrays.

Graph Stores are designed to hold data best represented by graphs, interconnected data with an unknown number of relations between the data, for example, social networks or road maps.

Key-Value Stores are the simplest type with with every bit of data stored with a name (as key) and its data (value).

Wide Column Stores are optimized for queries across large data sets.

There are other ways to describe the range of NoSQL databases available but these are the simplest and most comprehensive categories. And within each type of NoSQL database, functionality differs which can impact database design. For example, Mongo DB was evolved from the MySQL project, changing the data model from relational to NoSQL, yet retains most of the indexing, dynamic queries, and other useful features of relational databases.

Perhaps the key design difference between NoSQL and relational databases is the structure of data in each database. Relational databases require data be organized ahead of time. NoSQL databases can have their structure modified on the fly with little impact because they use key-value pairs; updating a data structure in NoSQL can involve adding additional data to the value of one or more keys while leaving other key-value pairs in the database untouched.

Design strategies for NoSQL databases depend on the type of database and the virtues (or negatives) of different data model techniques. Where relational databases have a user-centered approach, asking “What answers can I get from the database?”, NoSQL databases have an application-centered approach, asking “What questions do I have?”

This is a critical difference both in data structures as well as approaches to designing a database.

Configuring a database to provide specific answers entails lots of design and structure up front which limits future flexibility and makes future changes likely to be complicated. Configuring a

database to handle many possible questions, in contrast, results in a more flexible database design. Typically data is duplicated in many different places in a database to help answer questions with less effort. NoSQL database design uses a set of rules called BASE (basically available, soft-state, eventually consistent) to guide their design

NoSQL database data model techniques include:

- DE normalization puts all data needed to answer a query in one place, typically a single database table, instead of splitting the data into multiple tables.
- Aggregates use light or no validation of data types, for example, strings or integers.
- Joins are done at the application level, not as part of a database query. This requires more planning to match one type or set of data with another, for example, all examples of a product type (jeans) sorted by manufacturer in an online store.
- Indexes and key tables to identify and sort data quickly for retrieval.
- Tree structures can be modelled as a single data entity, for example, a comment with all its responses.

5.5. OUTPUT DESIGN

Output forms are also designed in a specific manner as per the user requirement. Results are formatted to enhance clarity. Depending on the user the system would generate appropriate output. The output forms are designed in such a way that the entire user required data is presented.

While designing an output, the system analyst must accomplish the following.

- Determine what information to present.
- Decide whether to display, print or speak information and select the output medium.
- Arrange the presentation of information in an acceptable form.
- Decide how to distribute the output to intended users

6.SYSTEM TESTING & IMPLEMENTATION

6.1.SYSTEM IMPLEMENTATION

The user may execute some or all Test Cases based on your mobile testing requirements. Test Cases are organized based on Mobile Testing Types.

- Functional Testing
- Performance Testing
- Security Testing
- Usability Testing
- Compatibility Testing

6.1.1. FUNCTIONAL TESTING

The functional testing of Mobiles normally consists in the areas of testing user interactions as well as testing the transactions. The various factors which are relevant in functional testing are

Type of application based upon the business functionality usages (banking, gaming, social or business)Target audience type (consumer, enterprise, education) Distribution channel which is used to spread the application (e.g. Apple App Store, Google play, direct distribution) The most fundamental test scenarios in the functional testing can be considered as:

To validate whether all the required mandatory fields are working as required. To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields. To validate whether the application works as per as requirement whenever the application starts/stops. To validate whether the application goes into minimized mode whenever there is an incoming phone call. In order to validate the same we need to use a second phone, to call the device. To validate whether the phone is able to store, process and receive SMS whenever the app is running. In order to validate the same we need to use a second phone to send sms to the device which is being tested and where the application under test is currently running. To

validate that the device is able to perform required multitasking requirements whenever it is necessary to do so. To validate that the application allows necessary social network options such as sharing, posting and navigation etc. To validate that the application supports any payment gateway transaction such as Visa, MasterCard, PayPal etc. as required by the application. To validate that the page scrolling scenarios are being enabled in the application as necessary. To validate that the navigation between relevant modules in the application are as per the requirement. To validate that the truncation errors are absolutely to an affordable limit. To validate that the user receives an appropriate error message like "Network error. Please try after some time" whenever there is any network error. To validate that the installed application enables other applications to perform satisfactorily, and it does not eat into the memory of the other applications. To validate that the application resumes at the last operation in case of a hard reboot or system crash. To validate whether the installation of the application can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors. To validate that the application performs auto start facility according to the requirements. To validate whether the application performs according to the requirement in all versions of Mobile that is 2g, 3g and 4g. To perform Regression Testing to uncover new software bugs in existing areas of a system after changes have been made to them. Also rerun previously performed tests to determine that the program behaviour has not changed due to the changes. To validate whether the application provides an available user guide for those who are not familiar to the app

6.1.2 PERFORMANCE TESTING

This type of testing's fundamental objective is to ensure that the application performs acceptably under certain performance requirements such as access by a huge number of users or the removal of a key infrastructure part like a database server.

The general test scenarios for Performance Testing in a Mobile application are:

To determine whether the application performs as per the requirement under different load conditions. To determine whether the current network coverage is able to support the application

at peak, average and minimum user levels. To determine whether the existing client-server configuration setup provides the required optimum performance level. To identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels. To validate whether the response time of the application is as per as the requirements. To evaluate product and/or hardware to determine if it can handle projected load volumes. To evaluate whether the battery life can support the application to perform under projected load volumes. To validate application performance when network is changed to WIFI from 2G/3G or vice versa. To validate each of the required the CPU cycle is optimization. To validate that the battery consumption, memory leaks, resources like GPS, Camera performance is well within required guidelines. To validate the application longevity whenever the user load is rigorous. To validate the network performance while moving around with the device. To validate the application performance when only intermittent phases of connectivity is required.

6.1.3 SECURITY TESTING

The fundamental objective of security testing is to ensure that the application's data and networking security requirements are met as per guidelines.

The following are the most crucial areas for checking the security of Mobile applications.

To validate that the application is able to withstand any brute force attack which is an automated process of trial and error used to guess a person's username, password or credit-card number. To validate whether an application is not permitting an attacker to access sensitive content or functionality without proper authentication. To validate that the application has a strong password protection system and it does not permit an attacker to obtain, change or recover another user's password. To validate that the application does not suffer from insufficient session expiration. To identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities. To prevent from SQL injection related attacks. To identify and recover from any unmanaged code scenarios. To ensure whether the certificates are validated, does the application implement Certificate Pinning or not. To protect the application and the network from the denial of service attacks. To analyse the data storage and data validation

requirements. To enable the session management for preventing unauthorized users to access unsolicited information. To check if any cryptography code is broken and ensure that it is repaired. To validate whether the business logic implementation is secured and not vulnerable to any attack from outside. To analyse file system interactions, determine any vulnerability and correct these problems. To validate the protocol handlers for example trying to reconfigure the default landing page for the application using a malicious frame. To protect against malicious client side injections. To protect against malicious runtime injections. To investigate file caching and prevent any malicious possibilities from the same. To prevent from insecure data storage in the keyboard cache of the applications. To investigate cookies and preventing any malicious deeds from the cookies. To provide regular audits for data protection analysis. Investigate custom created files and preventing any malicious deeds from the custom created files. To prevent from buffer overflows and memory corruption cases. To analyse different data streams and preventing any vulnerabilities from these.

6.1.4 USABILITY TESTING

The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult application with many features. The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used.

To ensure that the buttons should have the required size and be suitable to big fingers. To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users. To ensure that the icons are natural and consistent with the application. To ensure that the buttons, which have the same function should also have the same colour. To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled. To ensure that the keyboard input can be minimized in an appropriate manner. To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration. To ensure that the contextual menus are not overloaded because it has to be used quickly. To ensure that the text is kept simple and clear to be visible to the users. To ensure that the short sentences and paragraphs are readable to the end users. To ensure that the font size

is big enough to be readable and not too big or too small. To validate the application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance. To validate that the closing of the application is performed from different states and verify if it re-opens in the same state. To ensure that all strings are converted into appropriate languages whenever a language translation facility is available .To ensure that the application items are always synchronized according to the user actions. To ensure that the end user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application's proceeding. Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

6.1.5 COMPATIBILITY TESTING

Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired.

The following are the most prominent areas for compatibility testing.

To validate that the user Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible. To ensure that the text is readable for all users for the application. To ensure that the call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed.

6.2.SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system and giving a user confidence in that the new system will work efficiently and effectively in the implementation stage. The stage consists of

- Testing a developed program with sample data
- Detection and correction of error
- Creating whether the system meets a user requirements
- Making necessary changes as desired by users
- Training user personal

The purpose of system implementation can be summarized as follows: making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the performing Organization (the transition).at a finer level of detail, deploying the system consists of executing all steps necessary to educate the consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operation is available and accurate, and validating that business function that interact with the system are functioning properly. Transitioning the system support responsibility involves changing from a system development to a system support and maintenance mode of operation, with ownership of the new system moving from the project team to the performing organization

A key difference between system implementation and all other phases of the lifecycle is that all project activities up to this point have been performed in safe, protected, and secure environments, where project issues that arise have little or no impact on day-to-day business operations. Once the system goes live, however, this is no longer the case. Any miscues at this point will almost certainly translate into direct operational and/or financial impacts on the performing Organization. It is through the careful planning, executions, and management of system implementation activities that the project team can minimize the like hood of these occurrences, and determine appropriate contingency plans in the event of a problem.

6.2.1 IMPLEMENTATION PROCEDURES

The implementation phase is less creative than system design. A system design may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase. The final report of the implementation phase includes procedural flowcharts,

records layouts and a workable plan for implementing the candidate system design into an operational design

6.2.2 LIST OF PROCESS

This phase consists of the following processes:

Prepare for System Implementation, where all steps needed in advance of actually deploying the application are performed, including preparation of both the production environment and the consumer communities

Deploy System, where the full deployment plan, initially developed during System Design and evolved throughout subsequent lifecycle phases, is executed and validation

Transition to Performing Organization, where responsibility for and ownership of the application are transitioned from the project Team to the unit in the performing Organization that will provide system support and maintenance.

6.2.3 SYSTEM MAINTENENCE

The maintenance Manual provide maintenance personal with the information necessary to maintain the system effectively. The manual provide the definition of the software support environment, the roles and responsibilities of maintenance personnel, and the regular activities essential to the support and maintenance of program modules, job streams, and database structures.

In addition to the items identified for inclusion in the Maintenance Manual, additional information may he provided to facilitate the maintenance and modification of the system. Appendices to document various maintenance procedures standards, or other essential information may he added to this document as needed.

7. CONCLUSION

The need for the Android based Medicare alert using geo-fencing sensor is to computerize the application processing and servicing the Patients request through automated modules is most necessary and now inevitable.

As we have already seen that the need cannot be emphasized for the further development of this system is only timely and helpful to medicine seeker, the system defined in the above script is up to date and caters to all kinds of request faced by the requirements to provide the better service to the patients, being developed in java it is also flexible modularized highly parameterized and hence can be easily deployed by any other application because of its componentized approach.

Based on the various parameters and properties files everything from the look and feel to the functionalities can be customized. Thus this project is developed from the beginning with reuse in mind and implicitly uses several design patterns. The architecture of this project is such that it suits the diverse and distributed nature of the Medicare Applications.

8.SCOPE FOR FUTURE ENHANCEMENT

The features provided by the (Medicare Alert System) are in no means comprehensive but by all means full filling all important functionalities of Medical shops services. Inclusion of further functionalities as days go by can be easily done because the project has been developed in a layered architecture.

Plug-in modules would easily add new features which change with the times and being performance oriented the project will not face any issues. It is also extensible and scalable as all applications should be thus it can be said that it will meet surges of huge employee and patient requests that may come up in the near future with the payment gateway implementation.

9.BIBLIOGRAPHY

BOOKS:

i) Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)**(By: Bill Philips & Brian Hardy)**

ii) Reto Meier, "Professional Android Application Development", Wiley Publishing Inc., Indianapolis, Indiana, 2009. Google Inc., "Using JSON with Google Data APIs", July 3, 2009.

iii) Crockford, Douglas, "Introducing JSON", May 28, 2009.

WEBSITES

1. *Introduction to Android:* http://developer.android.com/guide/index.html.
2. *Android API:* http://developer.android.com/reference/packages.html
3. *Java 7 API:* http://docs.oracle.com/javase/7/docs/api/
4. *Android f undamentals* http://developer.android.com/guide/ /fundamentals.html

10. APPENDIX

B.SAMPLE CODING

LOGIN PAGE

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginPage"
    android:background="@drawable/gradient"
    android:orientation="vertical">



    <ImageView
        android:layout_width="120dp"
        android:layout_height="80dp"
        android:layout_marginLeft="110dp"
        android:layout_marginTop="50dp"
        />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
    android:id="@+id/Uname"
    android:hint="Enter Your Username"
    android:drawableLeft="@drawable/ic_action_user"
    android:layout_marginTop="50px"
    android:textColor="#fff"

    />
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/passw"
    android:layout_marginTop="80px"
    android:drawableLeft="@drawable/ic_action_name"
    android:inputType="textPassword"
    android:textColor="#fff"
    android:hint="Enter yout Password"/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/logbtn"
    android:text="Login"
    android:layout_marginTop="50px"

    />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="New User ! Register Here."
        android:layout_marginLeft="100dp"
        android:layout_marginTop="50px"
        android:layout_marginBottom="6dp"
        android:textColor="#fff"
        android:id="@+id/Newuser"
        android:background="@null"
       />

</LinearLayout>

package com.example.user.medical;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Html;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
```

```
public class LoginPage extends AppCompatActivity {
TextView tv1;
    EditText ed1,ed2;
    Button login,pass,newuser;
    String uname,passed,dbedname,dbedpass;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_page);
        final FirebaseDatabase database = FirebaseDatabase.getInstance();
        ed1=(EditText)findViewById(R.id.Uname);
        ed2=(EditText)findViewById(R.id.passw);

        tv1=(TextView)findViewById(R.id.Newuser);



        findViewById(R.id.logbtn).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                uname=ed1.getText().toString();
                passed=ed2.getText().toString();
                DatabaseReference myRef = database.getReference("Users/");

                myRef.child(uname).addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
```

```
            Users user = dataSnapshot.getValue(Users.class);
            dbedname=user.dbusername;
            dbedpass=user.dbuserpassword;
            Bundle b1=new Bundle();
            b1.putString("name",dbedname);
            // Toast.makeText(MainActivity.this,"login Sucess",Toast.LENGTH_LONG).show();
            if(uname.equalsIgnoreCase(dbedname) && passed.equalsIgnoreCase(dbedpass))
            {
               Intent i1=new Intent(LoginPage.this,UserHomepage.class);
               i1.putExtras(b1);
               startActivity(i1);

            }
            else
            {
               Toast.makeText(LoginPage.this,"Invalid Credentials",Toast.LENGTH_LONG).show();


            }
         }

         @Override
         public void onCancelled(@NonNull DatabaseError databaseError) {

         }
      });

   }
});
```

```
    tv1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i2=new Intent(LoginPage.this,MainActivity.class);
            startActivity(i2);
        }
    });
```

PHP :

```
<!DOCTYPE html>
<html lang="en">
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <title>Login</title>

        <meta name="viewport" content="width=device-width, initial-scale=1">
<!--
===============================================================================
==========-->
        <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>
<!--
===============================================================================
==========-->
        <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
<!--
===============================================================================
==========-->
        <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
<!--
===============================================================================
==========-->
```

```
	<link rel="stylesheet" type="text/css" href="fonts/iconic/css/material-design-iconic-font.min.css">
<!--
===============================================================================================-->
	<link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
<!--
===============================================================================================-->
	<link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">
<!--
===============================================================================================-->
	<link rel="stylesheet" type="text/css" href="vendor/animsition/css/animsition.min.css">
<!--
===============================================================================================-->
	<link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
<!--
===============================================================================================-->
	<link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">
<!--
===============================================================================================-->
	<link rel="stylesheet" type="text/css" href="css/util.css">
	<link rel="stylesheet" type="text/css" href="css/main.css">
<!--
===============================================================================================-->
</head>
<body>


	<div class="limiter">
		<div class="container-login100">
```

```
<div class="wrap-login100">

<form action="login.php" method="post" class="login100-form validate-
form">
<img style="height:100px; width:100px;margin-
left:150px;"src="images/aa.jpg">
</span><BR/><BR/>
<center>VR THERE TO CARE FOR U</center>

<span class="login100-form-title p-b-34 p-t-27">
Log in
</span>

<div class="wrap-input100 validate-input" data-validate =
"Enter username">
<input class="input100" type="text" name="username"
placeholder="Username">
<span class="focus-input100" data-
placeholder="&#xf207;"></span>
</div>

<div class="wrap-input100 validate-input" data-validate="Enter
password">
<input class="input100" type="password"
name="password" placeholder="Password">
<span class="focus-input100" data-
placeholder="&#xf191;"></span>
</div>

<div class="contact100-form-checkbox">
<input class="input-checkbox100" id="ckb1"
type="checkbox" name="remember-me">
```

```
                    <label class="label-checkbox100" for="ckb1">
                        Remember me
                    </label>
                </div>

                <div class="container-login100-form-btn">
                    <button class="login100-form-btn">
                        Login
                    </button>
                </form>

                </div>

                <div class="reg">
                    <a class="txt1" href="reg.php">
                        New User Signup
                    </a>
                </div>

                <div class="text-center p-t-90">
                    <a class="txt1" href="#">
                        Forgot Password?
                    </a>
                </div>

            </div>
        </div>
    </div>
```

```
        <div id="dropDownSelect1"></div>


<!--
===============================================================================================
===========-->
        <script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
===============================================================================================
===========-->
        <script src="vendor/animsition/js/animsition.min.js"></script>
<!--
===============================================================================================
===========-->
        <script src="vendor/bootstrap/js/popper.js"></script>
        <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<!--
===============================================================================================
===========-->
        <script src="vendor/select2/select2.min.js"></script>
<!--
===============================================================================================
===========-->
        <script src="vendor/daterangepicker/moment.min.js"></script>
        <script src="vendor/daterangepicker/daterangepicker.js"></script>
<!--
===============================================================================================
===========-->
        <script src="vendor/countdowntime/countdowntime.js"></script>
<!--
===============================================================================================
===========-->
        <script src="js/main.js"></script>
```

```
</body>
</html>
```