VISVESHVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

Mini Project Report
On

"Pharmacy management"

Submitted by

KOMAL S

In partial fulfillment of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Guide:
Madhura
Dept. of CSE

HOD:
Dr. Piyush Kumar pareek
Prof. & Head,
Dept. of CS&E

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

EAST WEST COLLEGE OF ENGINEERING

Yelahanka New Town,Bangalore-560064

2017-18

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mention of the people who made it possible and support had been a constant source of encouragement which crowned my efforts with success.

I am grateful to **Mr Ravi kiran sir** for providing me an opportunity to pursue my academics in this auspicious college and provide me an opportunity to present this Mini project report.

I am deeply indebted and i would like to express my sincere thanks to our beloved Principal **Dr. Ramesh B R, sir** for providing me an opportunity to do this Project.

My sincere thanks and special gratitude to my HOD **Dr. Piyush kumar pareek, HOD** of CS&E, EWCE for his guidance, constant encouragement and wholehearted support.

Finally I would like to express my sincere thanks to all the staff members of CS&E dept, EWCE for their valuable guidance & support.

KOMAL S

1EE15CS012

EAST WEST COLLEGE OF ENGINEERING

Yelahanka New Town, Bangalore-560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **komal s** of 5th semester B.E (Computer Science & Engineering) has completed the Mini Project report on **"pharmacy management"** in the partial fulfillment of MINI PROJECT prescribed by **Visveshvaraya Technological University** during the academic year 2017-18.

Signature of Guide
Madhura mam
Dept. of CS&E,
EWCE, Bangalore

Signature of HOD
Dr. Piyush Kumar Pareek
Prof. & Head,
Dept. of CS&E,
EWCE, Bangalore

External Signature

1.

2.

ABSTRACT

For any organisation, profile record keeping is very imperative to keep a track of various aspecs such as name of the customer, administration details, staff details etc. So as to manage such records on organisational level such as pharmacy companies and their outlets on various places, a large amount of information is equired to be monitered. The records of very individual then must be centrally tractable.The solution can be easily formed in digital database creation and management.

This project is insight into the design and implementation of a Pharmacy Management System. The primary aim of is to improve accuracy and enhance safety and efficiency in the pharmaceutical store. Today management is one of the most essential features of all form. Management provides sophistication to perform any kind of task in a particular form. This is pharmacy management system; it is used to manage most pharmacy related activities in the pharmacy.

CONTENTS

Chapter ONE

CHAPTER 1

INTRODUCTION TO DATABASE MANAGEMENT SYSTEM

1.1 Introduction

Databases and database technology have a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what a database is. Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database. The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted.

A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a generalpurpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types,

structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

1.2 HISTORY OF DBMS

A Database Management System allows a person to organize, store, and retrieve data from a computer. It is a way of communicating with a computer's "stored memory." In the very early years of computers, "punch cards" were used for input, output, and data storage. Punch cards offered a fast way to enter data, and to retrieve it. Herman Hollerith is given credit for adapting the punch cards used for weaving looms to act as the memory for a mechanical tabulating machine, in 1890. Much later, databases came along.

Databases (or DBs) have played a very important part in the recent evolution of computers. The first computer programs were developed in the early 1950s, and focused almost completely on coding languages and algorithms. At the time, computers were basically giant calculators and data (names, phone numbers) was considered the leftovers of processing information. Computers were just starting to become commercially available, and when business people started using them for real-world purposes, this leftover data suddenly became important.

Enter the Database Management System (DBMS). A database, as a collection of information, can be organized so a Database Management System can access and pull specific information. In 1960, Charles W. Bachman designed the Integrated Database System, the "first" DBMS. IBM, not wanting to be left out, created a database system of their own, known as IMS. Both database systems are described as the forerunners of navigational database.

By the mid-1960s, as computers developed speed and flexibility, and started becoming popular, many kinds of general use database systems became available. As a result, customers demanded a standard be developed, in turn leading to Bachman forming the Database Task Group. This group took responsibility for the design and standardization of a language called Common Business Oriented Language (COBOL). The Database Task Group presented this standard in 1971, which also came to be known as the "CODASYL approach."

The CODASYL approach was a very complicated system and required substantial training. It depended on a "manual" navigation technique using a linked data set, which formed a large network. Searching for records could be accomplished by one of three techniques:

- Using the primary key (also known as the CALC key)
- Moving relationships (also called sets) to one record from another
- Scanning all records in sequential order

Eventually, the CODASYL approach lost its popularity as simpler, easier-towork-with systems came on the market.

Edgar Codd worked for IBM in the development of hard disk systems, and he was not happy with the lack of a search engine in the CODASYL approach, and the IMS model. He wrote a series of papers, in 1970, outlining novel ways to construct databases. His ideas eventually evolved into a paper titled, A Relational Model of Data for Large Shared Data Banks, which described new method for storing data and processing large databases. Records would not be stored in a free-form list of linked records, as in CODASYL navigational model, but instead used a "table with fixed-length records."

IBM had invested heavily in the IMS model, and wasn't terribly interested in Codd's ideas. Fortunately, some people who didn't work for IBM "were" interested. In 1973, Michael Stonebraker and Eugene Wong (both then at UC Berkeley) made the decision to research relational database systems. The project was called INGRES (*Interactive Graphics and Retrieval System*), and successfully demonstrated a relational model could be efficient and practical. INGRES worked with a query language known as QUEL, in turn, pressuring IBM to develop SQL in 1974, which was more advanced (SQL became ANSI and OSI standards in 1986 1nd 1987). SQL quickly replaced QUEL as the more functional query language.

RDBM Systems were an efficient way to store and process structured data. Then, processing speeds got faster, and “unstructured” data (art, photographs, music, etc.) became much more common place. Unstructured data is both nonrelational and schema-less, and Relational Database Management Systems simply were not designed to handle this kind of data.

1.3 CHARACTERISTIC OF DATABASE APPROACH

A number of characteristics distinguish the database approach from the much older approach of programming with files. In traditional file processing, each user defines and implements the files needed for a specific software application as part of programming the application. For example, one user, the grade reporting office, may keep files on students and their grades. Programs to print a student’s transcript and to enter new grades are implemented as part of the application. A second user, the accounting office, may keep track of students’ fees and their payments. Although both users are interested in data about students, each user maintains separate files— and programs to manipulate these files—because each requires some data not avail able from the other user’s files. This redundancy in defining and storing data results in wasted storage space and in redundant efforts to maintain common up-to-date data. In the database approach, a single repository maintains data that is defined once and then accessed by various users. In file systems, each application is free to name data elements independently. In contrast, in a database, the names or labels of data are defined once, and used repeatedly by queries, transactions, and applications. The main characteristics of the database approach versus the file-processing approach are the following:

- Self-describing nature of a database system
- Insulation between programs and data, and data abstraction
- Support of multiple views of the data
- Sharing of data and multiuser transaction processing.

1.3.1 Self-Describing Nature of a Database System

A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data. The information stored in the catalog is called meta-data, and it describes the structure of the primary database. The catalog is used by the DBMS software and also by database users who need information about the database structure. A general-purpose DBMS software package is not written for a specific database application. Therefore, it must refer to the catalog to know the structure of the files in a specific database, such as the type and format of data it will access. The DBMS software must work equally well with any number of database applications—for example, a university database, a banking database, or a company database—as long as the database definition is stored in the catalog. In traditional file processing, data definition is typically part of the application programs themselves. Hence, these programs are constrained to work with only one specific database, whose structure is declared in the application programs. For example, an application program written in C++ may have structure or class declarations, and a COBOL program has data division statements to define its files. Whereas fileprocessing software can access only specific databases, DBMS software can access diverse databases by extracting the database definitions from the catalog and using these definitions.

1.3.2 Insulation between Programs and Data, and Data Abstraction

In traditional file processing, the structure of data files is embedded in the application programs, so any changes to the structure of a file may require changing all programs that access that file. By contrast, DBMS access programs do not require such changes in most cases. The structure of data files

is stored in the DBMS catalog separately from the access programs. We call this property program-data independence. In some types of database systems, such as object-oriented and object-relational systems users can define operations on data as part of the database definitions. An operation (also called a function or method) is specified in two parts. The interface (or signature) of an operation includes the operation name and the data types of its arguments (or parameters).The implementation (or method) of the operation is specified separately and can be changed without affecting the interface. User application programs can operate on the data by invoking these operations through their names and arguments, regardless of how the operations are implemented. This may be termed program-operation independence. The characteristic that allows program-data independence and program-operation independence is called data abstraction. A DBMS provides users with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented. Informally, a data model is a type of data abstraction that is used to provide this conceptual representation. The data model uses logical concepts, such as objects, their properties, and their interrelationships, that may be easier for most users to understand than computer storage concepts. Hence, the data model hides storage and implementation details that are not of interest to most database users.

1.3.3 Support of Multiple Views of the Data

A database typically has many users, each of whom may require a different perspective or view of the database. A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored. Some users may not need to be aware of whether the data they refer to is stored or derived. A multiuser DBMS whose users have a variety of distinct applications must provide facilities for defining multiple views.

1.3.4 Sharing of Data and Multiuser Transaction Processing

A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time. This is essential if data for multiple applications is to be integrated and maintained in a single database. The DBMS must

include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct. For example, when several reservation agents try to assign a seat on an airline flight, the DBMS should ensure that each seat can be accessed by only one agent at a time for assignment to a passenger. These types of applications are generally called online transaction processing (OLTP) applications. A fundamental role of multiuser DBMS software is to ensure that concurrent transactions operate correctly and efficiently. The concept of a transaction has become central to many database applications. A transaction is an executing program or process that includes one or more database accesses, such as reading or updating of database records. Each transaction is supposed to execute a logically correct database access if executed in its entirety without interference from other transactions. The DBMS must enforce several transaction properties. The isolation property ensures that each transaction appears to execute in isolation from other transactions, even though hundreds of transactions may be executing concurrently. The atomicity property ensures that either all the database operations in a transaction are executed or none are.

APPLICATIONS OF DBMS

Applications where we use Database Management Systems are:

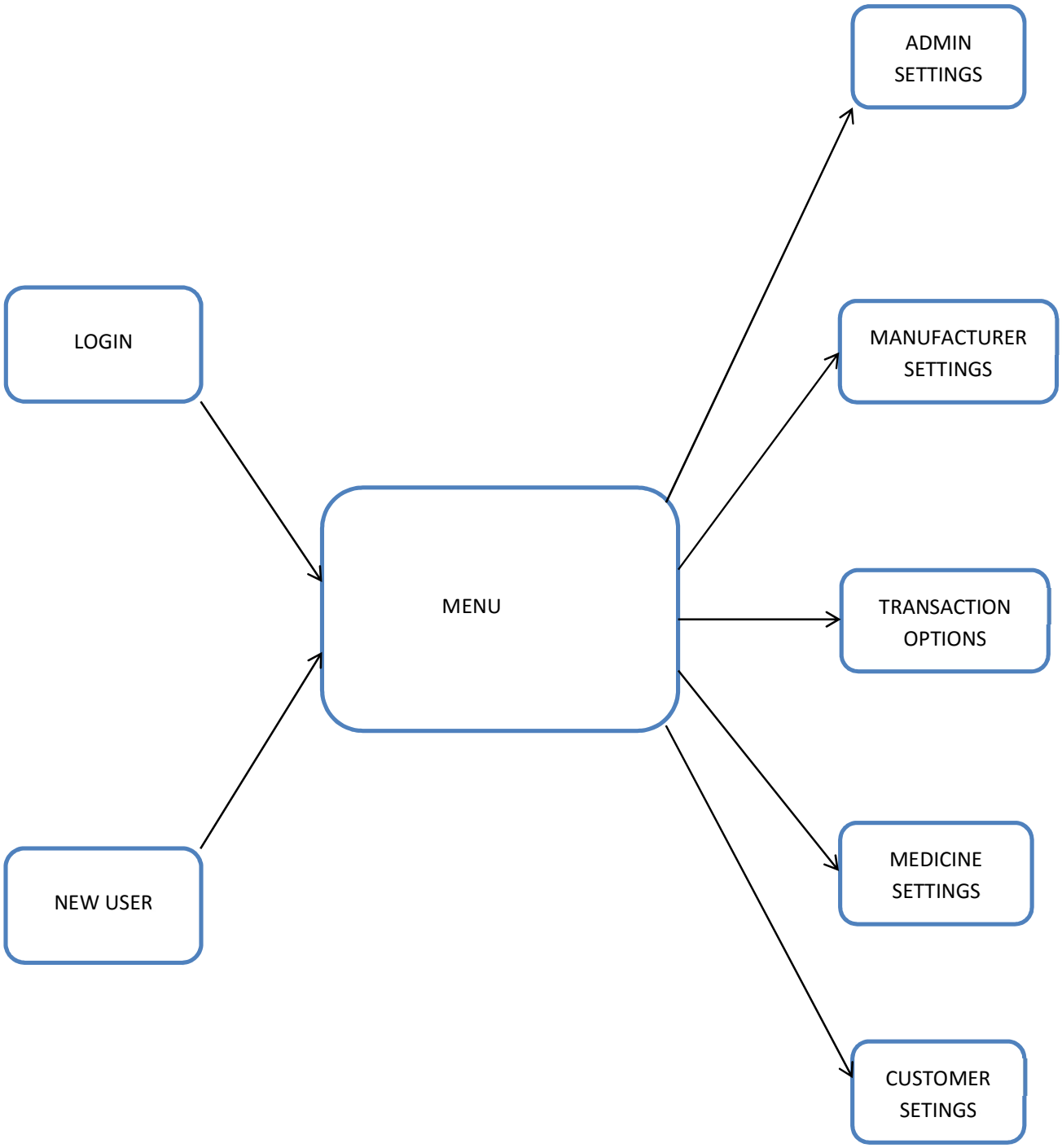
- **Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry**: Where it is a manufacturing unit, warehouse or distribution center, each one needs a database to keep the records of ins and outs. For example distribution center should keep a track of the product units that supplied into the center as well as the products that got delivered out from the distribution center on each day; this is where DBMS comes into picture.
- **Banking System**: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
- **Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.

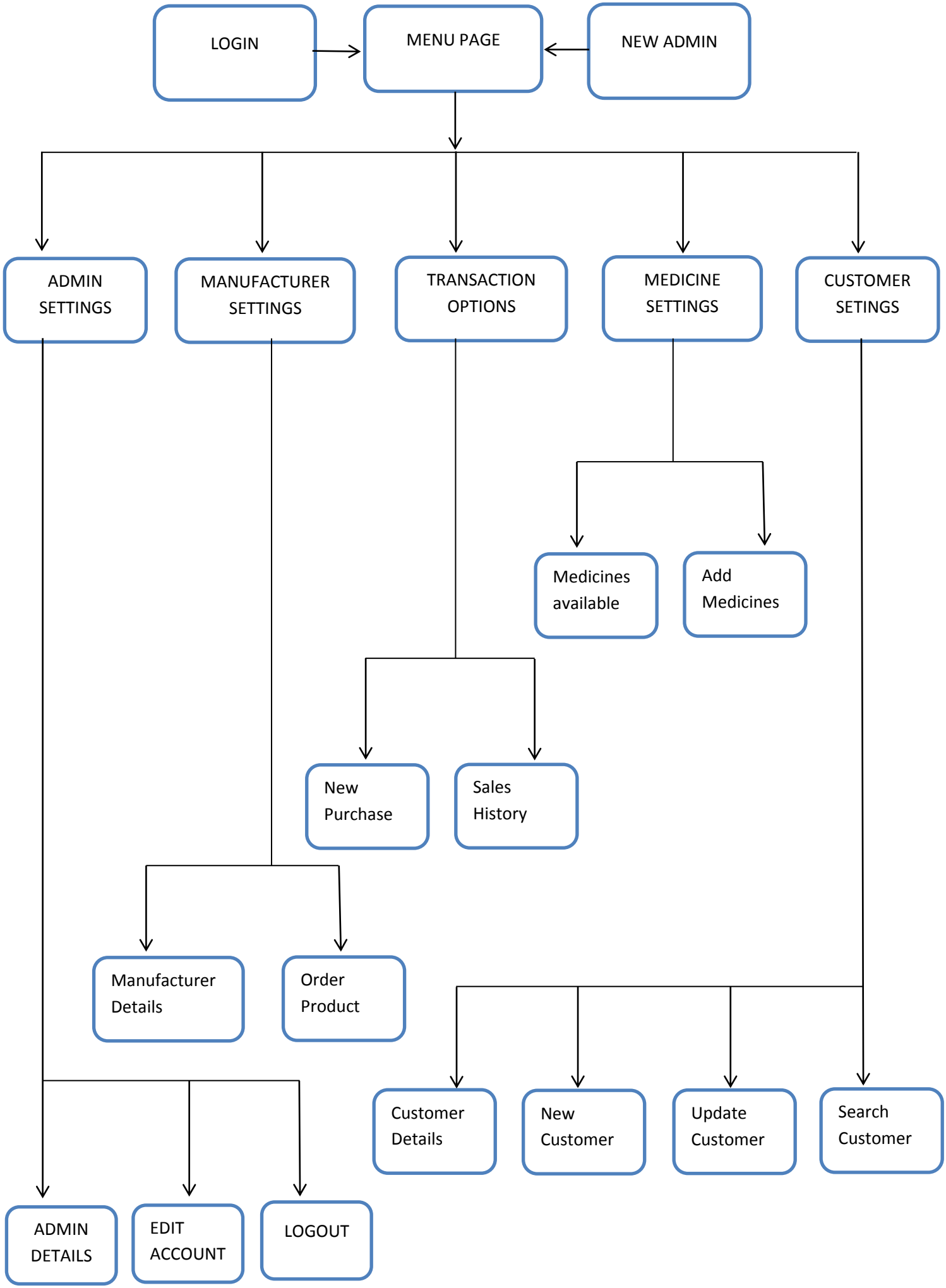
- **Online shopping**: You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

CHAPTER 2

SCHEMA AND ER DIAGRAM

E-R DIAGRAM


ADMIN SETTINGS
LOGIN
MANUFACTURER SETTINGS
MENU
TRANSACTION OPTIONS
MEDICINE SETTINGS
NEW USER
CUSTOMER SETINGS


LOGIN
MENU PAGE
NEW ADMIN
ADMIN SETTINGS
MANUFACTURER SETTINGS
TRANSACTION OPTIONS
MEDICINE SETTINGS
CUSTOMER SETINGS
Medicines available
Add Medicines
New Purchase
Sales History
Manufacturer Details
Order Product
Customer Details
New Customer
Update Customer
Search Customer
ADMIN DETAILS
EDIT ACCOUNT
LOGOUT

2.4 DATABASE DESIGN

This is a shared collection of data that are related or files that are to meet the immediate need of authorized users. These data may be in form of text, numeric, date or encoded images.

ADMINISTRATOR LOGIN TABLE:

Field Name	Field Type	Field Length	Description
Admin_Username	Varchar	(15)	Admin Username
Password	Varchar	(15)	Admin password

Table 2.1 Administrator table

Table 2.1 above serves as a repository for administrator details, that will be able to have access to the entire details on the application. It is a table that stores the administrator login details.

DRUG INFORMATION TABLE

Field Name	Field Type	Field Length	Description
Drug_Name	Varchar	(15)	Drug Name
Drug_ID	Varchar	(7)	Drug Number in stock
Manufacturer	Varchar	( )	Manufacturer
Batch_No	Varchar	( )	Batch Number
Production_Date	Date/Time	( )	Production Date
Expiry_Date	Date/Time	( )	Expiry Date

Table 2.2 Drug information table

Table 2.2 shows the information about the drugs in the pharmacy and is being queried from the database on the drug registration page to show all the drugs

Table 2.3 above serves as a repository for supervisor details, which will be able to have access to the required details on the application as stipulated by the administrator. It is a table that stores the supervisor login details.

CUSTOMER INFORMATION TABLE

Field Name	Field Type	Field Length	Description
CUST_NAME	Varchar	( ) (6)	Staff Name Registration number
REGNO	Int		
CUST_SEX	Varchar	(6)	Staff Sex
NATIONALITY	Varchar	( )	Nationality
STAFF_STATUS	Varchar	(15)	Staff Status
DOB	Date	( )	Date Of Birth
PHONE_NO	Int	(11)	Phone Number
ADDRESS	Varchar	( )	Contact Address
STATE	Varchar	( )	State
EMAIL	Varchar	( )	Email
USERNAME	Varchar	( )	Username
PASSWORD	Varchar	(15)	Password

TABLE 3.5 STAFF INFORMATION TABLE

Table 3.5 above consist of information of the list and information of the employed staff in the pharmacy .The information of staff on the required field can be seen from the table 3.5 above.

CHAPTER 3

RESOURCE REQUIREMENTS

3.1 Software requirement:-

- Front End tools: netbeans,java.
- Back End tools: Sql database
- Browser that support Java
- MySQL database

3.2 Hardware requirement:-

- CPU: Pentium processor and above
- RAM: 2 GB
- HDD: 40 GB

CHAPTER 4

Description of Tools and Technologies

4.1 MySQL Database

MySQL is a Relational Database Management System (RDBMS). MySQL server can manage many databases at the same time. In fact, many people might have different databases managed by a single MySQL server. Each database consists of a structure to hold the data and the data itself. A database can exist without data, only a structure, be totally empty, twiddling its thumbs and waiting for data to be stored in it.

Data in a database is stored in one or more tables. You must create the data-base and the tables before you can add any data to the database. First you create the empty database. Then you add empty tables to the database.

Database tables are organized like other tables that you're used in rows and columns. Each row represents an entity in the database, such as a customer, a book, or a project. Each column contains an item of information about the entity, such as a customer name, a book name, or a project start date. The place where a particular row and column intersect, the individual cell of the table, is called a field. Tables in databases can be related. Often a row in one table is related to several rows in another table. For instance, you might have a database containing data about books you own. You would have a book table and an author table. One row in the author table might contain information about the author of several books in the book table. When tables are related, you include a column in one table to hold data that matches data in the column of another table

4.2 What is MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

- MySQL is a database management system.

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL is a relational database management system.

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- MySQL software is Open Source.

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. The MySQL Database Server is very fast, reliable, and easy to use.

MySQL Seerver was originallu developed to handle large databases must faster than existing solutions and has been successfully used in highly demaning producion environments for several years. Although under costant development, MySQL Server today offers a rich and usefu set of functions. Its connectivity, speed, and security make MySQL Server highly suitable for accessing databases on the internet.

- MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and liraries, administrative tools, and a wide range of application programming interface(APIs).

CHAPTER 5

Database Connection

5.1 Creating Database Connection

- JAVA provides built-in database connectivity for a wide range of databases -

  MySQL, PostgreSQL, Oracle, Berkeley DB, Informix, Lotus Notes, and more

- Use either mysql_connect or mysql_pconnect to create database connection
- mysql_connect: connection is closed at end of script (end of page)
- mysql_pconnect: creates persistent connection -connection remains even after end of the page
- Connect to the MySQL server ○ $connection = mysql_connect("localhost",

  $username, $password);

- Access the database ○ mysql_select_db("databasename", $connection);

- Perform SQL operations ○ Example: $result = mysql_query ($query, $connection)
- Disconnect from the server ○ mysql_close($connection);

5.2 Architecture used (4-TIER architecture)

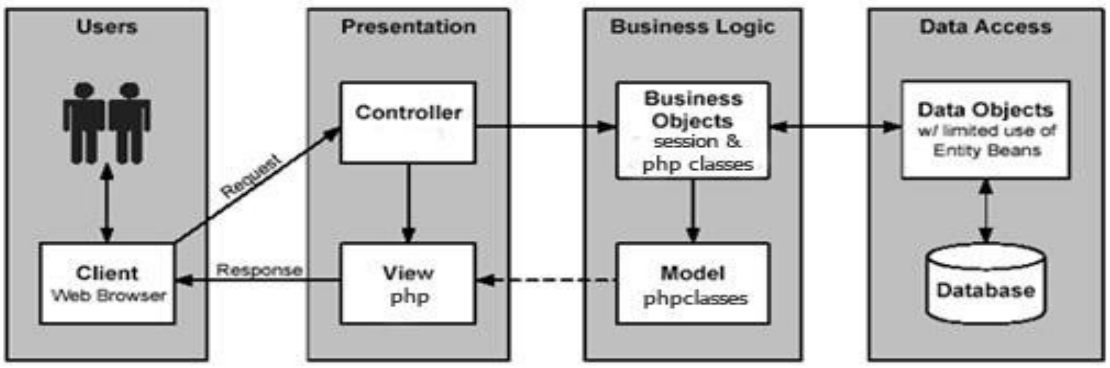


Fig: Architecture used

Four Tier architecture is a client–server architecture in which presentation, application processing, and data management functions are physically separated. Four-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application.

5.2.1 Presentation layer

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing and shopping cart contents. It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system's GUI).

5.2.2 Business layer

Business layer or domain logic is the part of the program that encodes the real-world business rules that determine how data can be created, stored, and changed. It is contrasted with the remainder of the software that might be concerned with lower-level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program.

5.2.3 Data access layer

A Data Access Layer (DAL) in computer software, is a layer of a computer program which provides simplified access to data stored in persistent storage.

For example, the DAL might return a reference to an object (in terms of object-oriented programming) complete with its attributes instead of a row of fields from a database table. This allows the client (or user) modules to be

created with a higher level of abstraction. This kind of model could be implemented by creating a class of data access methods that directly reference a corresponding set of database stored procedures. Another implementation could potentially retrieve or write records to or from a file system. The DAL hides this complexity of the underlying data store from the external world.

5.2.4 Control layer

The control layer is responsible for communication between business and presentation layer. It connects the logic and data with each other and gives a better connectivity and separation between layers.

CHAPTER 6

IMPLEMENTATION

SNAPSHOTS

HOME PAGE

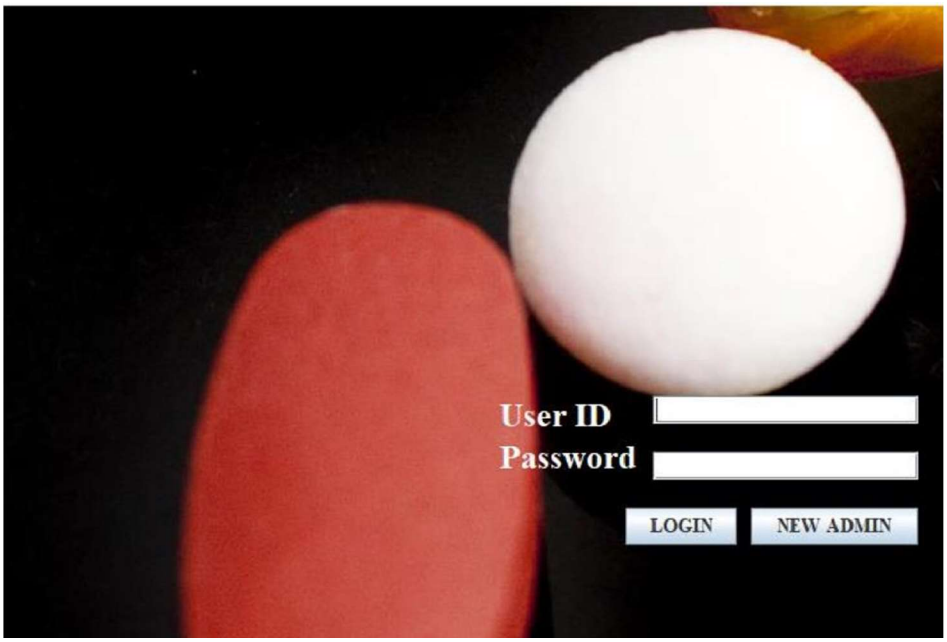
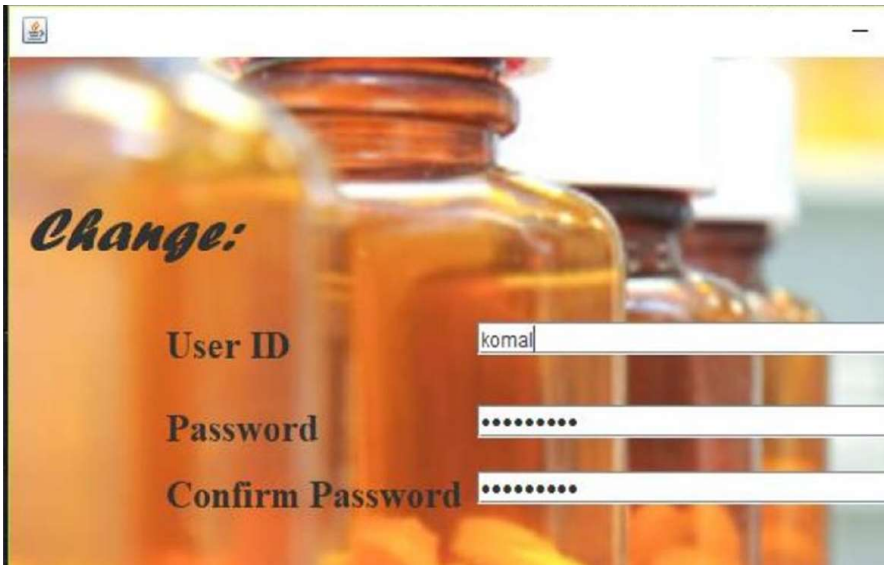


Figure 4.1 showing HOME PAGE

Figure 4.1 shows the select login page. It is the first interface that appears on the screen when the application is being loaded. This interface displays the name of the application and some other information about the software. The page consist of logins that exist for several other levels in the application. They consist of administrator, supervisor and staff login.

CHANGE PASSWORD:



ADMIN MAIN MENU

Figure 4.5 Showing Admin Menu Module

Figure 4.5 above shows the Admin module menue. It list activities that can be performed by the administrator. The Administrator is the person responsible for the upgrade, management on the software. The administrator is given an unlimited access as to performing the operations in the pharmacy.

MANUFACTURER DETAILS

Figure 4.6 above shows the manufacturer details which including the list of manufacture name, the address, the drug manufactuered and the phone number. the details given here can be changed by the administrator.

PLACING ORDER

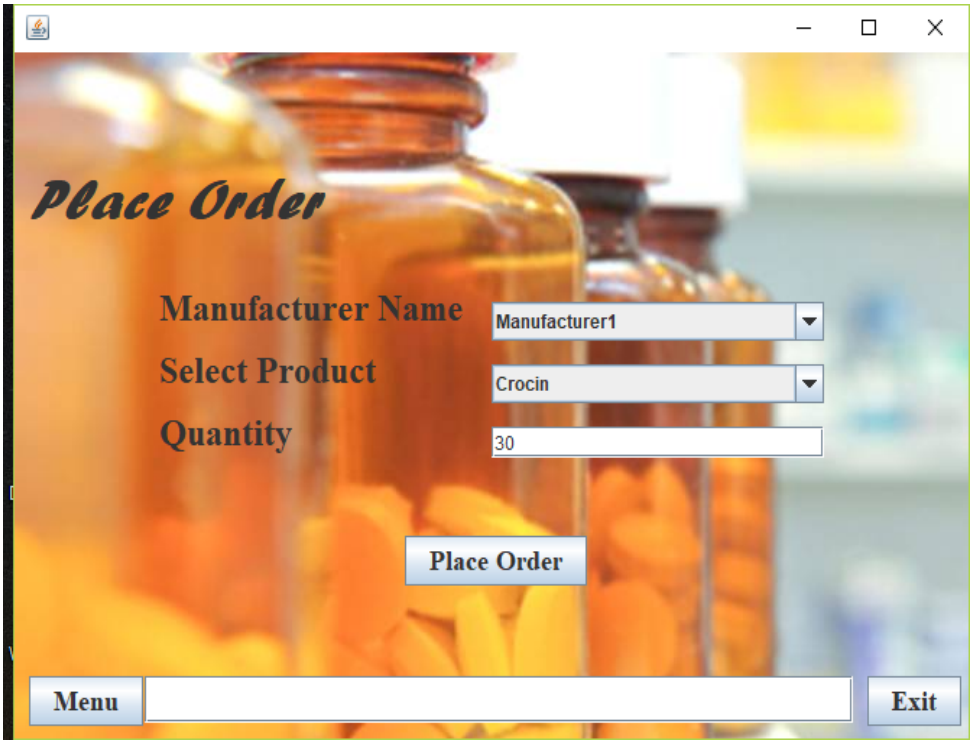
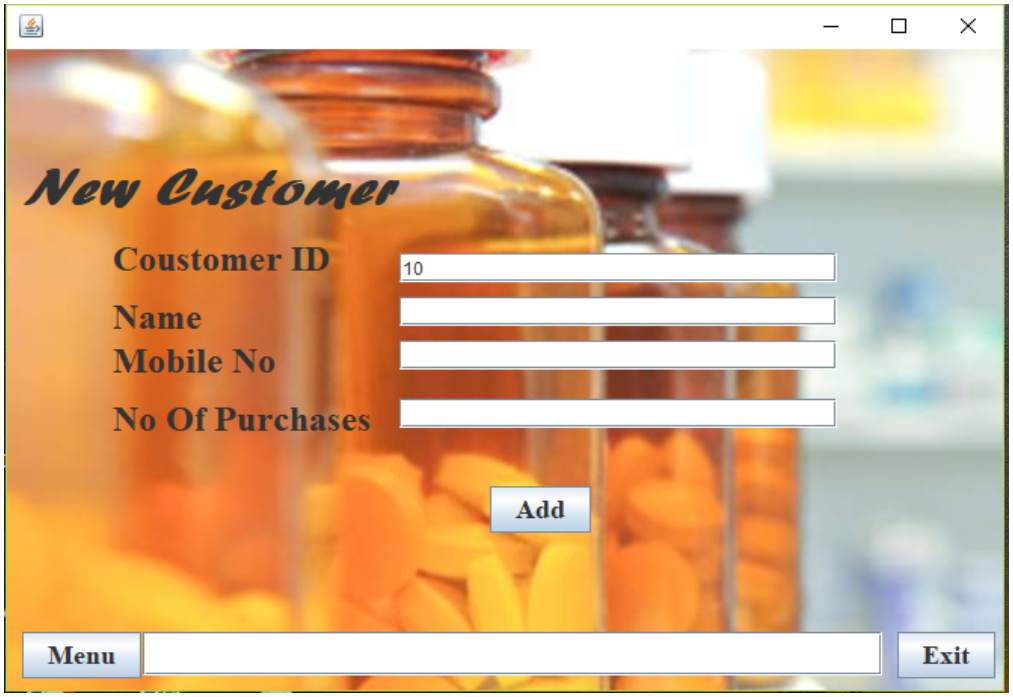


Figure 4.8 above shows the page of placing order. Agai, this can be done by the administrator operaing the application at that time. the page consists of manufacturer name from whom we have to place order, the drug name whose order has to be placed and the quantity of the drug required for order. The quantity here is in the amount of strips. One strip might contain about 10 tablets of the drug. This varies from manufacturer to another.

CUSTOMER DETAILS

Figure 4.9 and Figure 4.10 shows the details of the customer who have given their information of sae of storing it in the database. the customer whose details are saved in the database have advantages over customers whose details aren't saved. The customer could get a discount for being regular.



SOTWARE MAINTENANCE

Maintenance is a continuous process of making modifications and upgrading the application. This usually commences after the application has gone into use. There are two different ways by which this application can be maintained. They include:

- Additive or Enhancement maintenance**:** Business processes are dynamic. As business processes change, applications that support these processes must evolve to reflect these changes. Thus, for this application to perform optimally and to meet changing user requirements, it must be modified continuously.

- Corrective maintenance: This is required in the event that an error occurred when the application is in use. Corrections must be made to changes discovered that can cause malfunctioning of the system.

CHAPTER 7

FUTURE ENHANCEMENT

Designing this application (Pharmacy management system) is not an easy task. It all started from the requirement gathering and passes through so many other stages before completion.

Based on the benefits of this system and tremendous value it will add to customeruser satisfaction, the below recommendation will be considered;

It is recommended that the new system should be used with the necessary specifications of the system requirements and provision for an uninterrupted power supply should be made available throughout the hours of operation of the pharmacy to avoid power outage. There should also be basic computer knowledge for the users of the software.

It is recommended that the software be improved especially in areas of accounting as it will be of great impact to the development of retail pharmacy.

As of now, only the backbone of the system thatis the databse is being monitered. The scope of the project is inflationary. The system can be raised to run in real time with enhancements like storing customer prescriptions, adding and monitereing staff in the centre, sending and receing bill payments to customer on their phones, etc. It can be personalised, can have stratified interfaces thatis for staff access. It can be upgradded to a virtual portal for notification updates etc.

CONCLUSION

Effective implementation of this software will take care of the basic requirements of the pharmacy management system because it is capable of providing easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved.

In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the pharmacy management, whenever the idea arises.

BIBLIOGRAPHY

WE COLLECT OUR DATA FROM THE FOLLOWING SOURCES

www.google.com
www.programmingknowledge.com
www.asknerd.com
www.github.com

www.Mysql.com/downloads/tools/