

PHARMACY MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Science degree in Computer Science

by

Sivaprakash M (39290095)

Varunkumar S (39290120)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI-600119

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sivaprakash M (39290095)** and **Varunkumar s(39290120)** who carried out the project entitled as “**Pharmacy management system**” under our supervision from Dec 2021 to March 2022.

Internal Guide

Dr.Malini Deepika

Head of the Department

Dr.L.Lakshmanan M.E,Ph.d

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, Sivaprakash M (39290095) and Varunkumar S (39290120) hereby declare that the Project Report entitled “**Pharmacy management system**”done by me under the guidance of Dr.Mrs.Malini Deepika is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer science.

DATE:10.03.2022

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr./Mr./Ms for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Pharmacy Management System application to help pharmacist to manage pharmacy in the systematic ways. Pharmacy Management System can make the work easier by giving the details of medicine when its name is entered. A computer gives the details of the medicine like rate of medicine, and expiry date of the medicine. It becomes very difficult in big medical stores to handle the details of all the medicines manually, so by using this pharmacy management system We can maintain the records of all the medicines. It is fed with the information whenever new medicines are brought and it is provided with expire date with search option. When we entire the name of medicine it gives the details of medicine .One of the most important responsibilities of pharmacy management is to supervise and manage the pharmacy employees in order to ensure healthy working relationships and outcomes. Each of these functions is critical to the pharmacy's operation and should be explained by the management.

However, most pharmacies faced problems such as insufficient service promotions, lack of coherence of pharmacy services in hospitals, poor drug information systems and the inconsistency of the pharmacy information management due to its manual processes. Now these are the problems that must be solved with this Pharmacy Management System Project Proposal.

TABLE OF CONTENTS

Chapter No.		Page No.
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1.	INTRODUCTION	1
	1.1. Overview of the project	1
2	LITERATURE SURVEY	3
3	AIM AND SCOPE OF PRESENT INVESTIGATION	6
	3.1 Aim of the project	6
	3.2 scope and objective	6
	3.3 System requirements	7
	3.3.1 Hardware requirements	7
	3.3.2 Software requirements	7
	3.4 Software used	8
	3.4.1 Python language	8
	3.4.2 Python characteristics	8
	3.4.3 Applications of python programming	9
	3.4.4 Opencv package	10
	3.5 Anaconda navigator	11
	3.5.1 Anaconda	11
	3.5.2 Applications of anaconda	11
	3.5.3 Pycharm IDE	12
	3.5.4 SQLite 3	13
	3.5.5 New features of Anaconda 5.3	13
	3.5.6 Tkinter	15
4	EXPERIMENTAL OR MATERIAL METHODS	20
	4.1 Design methodology	20
	4.1.1 Existing system	20
	4.1.2 Proposed system	20

	4.2 Module description	21
	4.3 System implementation	22
	4.4 Architecture diagram	22
	4.4.1 Application architecture diagram	25
5	RESULTS AND PERFORMANCE ANALYSIS	27
	5.1 Homepage	27
	5.2 Login page	28
	5.3 Add new customer	28
	5.4 Add new product	29
	5.5 Delete product	29
	5.6 Modify product	30
	5.7 Expiry check	30
	5.8 Billing system	31
	5.9 Check revenue	31
6	CONCLUSION AND FUTURE ENHANCEMENTS	32
	6.1 Conclusion	32
	6.2 Future enhancements	32
	REFERENCES	33
	APPENDIX	34

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO.
5.1	Homepage	27
5.2	Login page	28
5.3	Add new customer	28
5.4	Add new product	29
5.5	Delete product	29
5.6	Modify product	30
5.7	Expiry check	30
5.8	Billing system	31
5.9	Check revenue	31

LIST OF ABBREVIATIONS

GUI - Graphical user interface
IDE - Integrated development environment
DF - Data flow
OS - Operating system
API - Application programming interface
OOP - Object oriented programming
CMS - Content management system.

CHAPTER 1

1. INTRODUCTION

1.1 Overview of the project

The main aim of the project is the management of the database of the pharmaceutical shop. This project is insight into the design and implementation of a Pharmacy Management System. This is done by creating a database of the available medicines in the shop. The primary aim of pharmacy management system is to improve accuracy and enhance safety and efficiency in the pharmaceutical store. The aim of this project is to develop software for the effective management of a pharmaceutical store. We have developed this software for ensuring effective policing by providing statistics of the drugs in stock. The database is then connected to the main program by using interconnection of the Visual Basic program and the database already created. Pharmacy management system is useful to maintain correct database by providing an option to update the drugs in stock. This is pharmacy management system; it is used to manage most pharmacy related activities in the pharmacy.

Pharmacy management system is a management system that is designed to improve accuracy and to enhance safety and efficiency in the pharmaceutical store. This program can be used in any pharmaceutical shops having a database to maintain. It is a computer based system which helps the Pharmacist to improve inventory management, cost, medical safety etc. The software used can generate reports, as per the user's requirements. Using this pharmacy management system user is also able to generate report within a specified period of time. The system allows the user to enter a manufacturing and expiry date for a particular product or drug during opening stock and sales transaction. The software can print invoices, bills, receipts etc. It can also maintain the record of supplies sent in by the supplier. The system will also give report showing the list of products expiry after a specified date before the product eventually expires. The system services and goals are established by consultation with system user. It also

involves manual entry upon arrival of new batches of drugs and upon drug movement out of the pharmacy for a certain period. Pharmacy management system is being build. Pharmacy management system is robust, integrated technology. every month, the pharmacist may want to generate report for the movement of drugs in and out of the pharmacy, getting information about the drugs e.g. expiry date, date purchased, number of drug type left, location of a drug in the pharmacy. Pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. This pharmacy management system is user friendly.

The purpose of this Pharmacy Management System project is to improve the maintenance and manipulation of the drugs in the medicals. The pharmacy management system will be used to minimize the time and resource by maintaining the details of the drug systemically so that the data can be used in possible quickest time. While the resource which is minimized are workforce, money, papers, etc. The system is user-friendly and will help the pharmacist. This Pharmacy Management System will reduce the burden on pharmacist and will make the system efficient by providing the more accurate details about drugs in the medical.

CHAPTER 2

2. LITERATURE SURVEY

1. Medicines management

According to Western Pac Surveil Response J. (2015), Medicines management is the entire process of how medicines are selected, procured, delivered, prescribed, administered and reviewed to optimize the contribution they make to producing informed and desired outcomes of patient care. In the Philippines, medicines management occurs at all levels of government and is different during emergency and non-emergency times.

Health service delivery in the Philippines has been repeatedly disrupted as a result of disasters and emergencies – particularly so after Typhoon Haiyan in November 2013. The objectives of this study were to document existing policies for medicines management in the Philippines during emergency and non-emergency periods and to assess the public sector medicines management system in Haiyan-affected areas during the response.

2. Log-in

According to Dan Brian (2011) a lot of the websites we visit nowadays are testing and requiring that we create so-called “strong” passwords. Unfortunately, it is not a practice that every site accommodates, whether because it has not update, edits code to meet these most recent standards, or simply because it lacks the resources to do so. It is then up to the user to create its own personal “Security Best Practices”.

3. RECORDS MANAGEMENT

According to Patricia Faulhaber (2015), Record management is a systematic, organized, planned and controlled process of managing or tracking the life cycle of records. A record can be a tangible paper object or it can be in digital or electronic form. Records can be kept on financial, medical, informative, formal documents, office documents, payroll, government forms and emails among hundreds of other types of records. Record management addresses the three phases of the life cycle of records: The creation or the receipt of a record; the maintenance, safe storage, retrieval, or general use of a record; the disposal of a record.

4. Computerized Payroll System

According to Grace Ferguson (2017), Computerized systems offer a number of advantages, including increased accuracy and speed. Computerized payroll systems can round employee work hours into quarter-hour segments and accurately calculate the total hours worked and to be paid, thereby saving time spent on manual calculations. These systems calculate all pay frequencies, such as weekly, biweekly, semimonthly, based upon the input data received. The similarities of the proposed system to the Computerized Payroll System are almost the same in terms of Automated System, Connectivity, Platforms (PC or Android), and Payroll.

5. Computer Based System

As stated in the book of Passon (2011), to improve their efficiency and reduce their cost of operation. Reports are more timely and accurate. Computers have a large capacity to store data and tremendous speed to manipulate and recall data to the format a user wished to use which introduction of microcomputer computation of business is more affordable (Espinosa, n.d.). The similarities of the proposed system to the Computer Based System are almost the same in terms of Automated System, Connectivity, Platforms (PC or Android), Point of Sale, and Payroll.

6. SYSTEM ANALYSIS AND DESIGN

Cashman stated in her book “System Analysis and Design” (2011) that transaction processing system data generated by day-to-day business operations. Transaction processing are efficient because they process a set of transaction related commands as a group rather than individually to protect data integrity however, TP systems ensure that if any single element of a transaction fails, the system does not process the next transaction.

7. INVENTORY

According to Hold (2010) article, “Inventory refers to stocks of anything necessary to do business”. The U.S Small Business Administration publication describes what constitutes successful inventory management balancing cost versus benefits of inventory, including: Maintaining a wide assortment without sacrificing service; Keeping stock low without sacrificing performance; Obtaining lower prices by making volume purchases; and Maintaining an adequate inventory without an excess of obsolete items.

CHAPTER 3

3. AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF THE PROJECT

One of the most important responsibilities of Medical Shop management is to supervise and manage the pharmacy employees in order to ensure healthy working relationships and outcomes. Each of these functions is critical to the pharmacy's operation and should be explained by the management. However, most pharmacies faced problems such as insufficient service promotions, lack of coherence of pharmacy services in hospitals, poor drug information systems and the inconsistency of the pharmacy information management due to its manual processes. Now these are the problems that must be solved with this Pharmacy Management System Project Proposal.

3.2 SCOPE AND OBJECTIVE

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Pharmacy Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.

3.3 SYSTEM REQUIREMENTS

3.3.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The minimal hardware requirements are as follows,

1. Processor : Pentium IV
2. RAM : 4GB ram
3. Processor : 2.4GHZ
4. Main memory : 4GB

3.3.2 Software Requirements

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. The minimal software requirements are as follows,

1. **Front end** : Python
2. **Back end** : SQL lite 3
3. **IDE** : Pycharm
4. **Operating system** : Windows 10

3.4 SOFTWARE USED

3.4.1 Python Language

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications.

It has interfaces to many OS system calls and libraries and is extensible to C or C++.

Many large companies use the Python programming language include NASA, Google, YouTube, Bit Torrent, etc. Python programming is widely used in Artificial

Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class

will teach you python from basics.

3.4.2 Python Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's.
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-cod

- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code.
- In Python, since there is no compilation step, editing, debugging and testing is fast.

3.4.3 Applications of Python Programming

Web Applications:

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing:

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: Earthy for earth science, Astray for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes:

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygmy (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming:

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

3.4.4 OpenCV Package

Python is a general-purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

And the support of NumPy makes the task easier. NumPy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from NumPy arrays. So whatever operations you can do in NumPy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

3.5 ANACONDA NAVIGATOR:

3.5.1 Anaconda:

Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning.

The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source packages long with free community support.

What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.

3.5.2 Applications Of Anaconda

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

- 1.JupyterLab
- 2.Jupyter Notebook
- 3.Qt Console
- 4.Spyder
- 5.Glueviz
- 6.Orange3
- 7.RStudio
- 8.Visual Studio Code
- 9.Pycharm IDE

3.5.3 Pycharm IDE:

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive [Python](#), [web](#), and [data science](#) development.

PyCharm is available in three editions:

- *Community* (free and [open-sourced](#)): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.
- *Professional* ([paid](#)) : for professional Python, web, and data science development, including code assistance, refactorings, visual debugging, version control integration, remote configurations, deployment, support for popular web frameworks, such as Django and Flask, database support, scientific tools (including Jupyter notebook support), big data tools.
- *Edu* (free and [open-sourced](#)): for learning programming languages and related technologies with integrated educational tools.

Supported languages:

PyCharm supports the following versions of Python:

- **Python 2:** version 2.7
- **Python 3:** from the version 3.6 up to the version 3.11

Besides, in the *Professional* edition, one can develop Django , Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to Settings | Plugins or PyCharm | Preferences | Plugins for macOS users, to find out more or set them up during the first IDE launch).

3.5.4 SQLite3

Python SQLite3 module is used to integrate the SQLite database with Python. It is a standardized Python DBI API 2.0 and provides a straightforward and simple-to-use interface for interacting with SQLite databases. There is no need to install this module separately as it comes along with Python after the 2.5x version.

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The `sqlite3` module was written by Gerhard Häring. It provides an SQL interface compliant with the DB-API 2.0 specification described by [PEP 249](#), and requires SQLite 3.

SQLite version 3.0 introduces important changes to the library, including:

- A more compact format for database files.
- Manifest typing and BLOB support.
- Support for both UTF-8 and UTF-16 text.
- User-defined text collating sequences.
- 64-bit ROWIDs.
- Improved Concurrency.

3.5.5 New Features of Anaconda 5.3

Compiled with Latest Python release: Anaconda 5.3 is compiled with Python 3.7, taking advantage of Python's speed and feature improvements.

Better Reliability:

The reliability of Anaconda has been improved in the latest release by capturing and storing the package metadata for installed packages.

Enhanced CPU Performance:

The Intel Math Kernel Library 2019 for Deep Neural Networks(MKL 2019) has been introduced in Anaconda 5.3 distribution. Users deploying TensorFlow can make use of MKL 2019 for Deep Neural Networks. These Python binary packages are provided to achieve high CPU performance.

New packages are added:

There are over 230 packages which has been updated and added in the new release.

Work in Progress:

There is a casting bug in Numpy with Python 3.7 but the team is currently working on patching it until Numpy is updated.

3.5.6 Tkinter

This Tkinter tutorial introduces you to the exciting world of GUI programming in python. Tkinter is pronounced as tea-kay-inter. Tkinter is the Python interface to Tk, which is the GUI toolkit for Tcl/Tk.

Tcl (pronounced as tickle) is a scripting language often used in testing, prototyping, and GUI development. Tk is an open-source, cross-platform widget toolkit used by many different programming languages to build GUI programs. Python implements the Tkinter as a module. Tkinter is a wrapper of C extensions that use Tcl/Tk libraries. Tkinter allows you to develop desktop applications. It's a very good tool for GUI programming in Python.

Tkinter is a good choice because of the following reasons:

- Easy to learn.
- Use very little code to make a functional desktop application.
- Layered design.
- Portable across all operating systems including Windows, macOS, and Linux.
- Pre-installed with the standard Python library.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

1. Tkinter Fundamentals

- Tkinter Hello, World! – show you how to develop the first Tkinter program called Hello, World!
- Window – learn how to manipulate various attributes of a Tkinter window
- including title, size, location, resizable, transparency, and stacking order.
- Tk Themed Widgets – introduce you to Tk themed widgets.
- Setting options for a widget – learn various ways to set options for a widget.
- Command Binding – learn how to respond to events using command bindings
- Event Binding – show you how to use the bind() method to bind an event of a widget
- Label – learn how to use the Label widget to show a text or image on a frame or window
- Button – walk you through the step of creating buttons.
- Entry – learn how to create a textbox using the Entry widget.

2. Layout Managements

Geometry managers allow you to specify the positions of widgets inside a top-level or parent window.

- pack – show you how to use the pack geometry manager to arrange widgets on a window.
- grid – learn how to use the grid geometry manager to place widgets on a container.
- place – show you how to use the place geometry manager to precisely position widgets within its container using the (x, y) coordinate system.

3. Ttk & Tkinter Widgets

Tkinter provides you with some commonly used widgets, which allow you to start developing applications more quickly.

- Text – show a multi-line text input field.
- Scrollbar – learn how to link a scrollbar to a scrollable widget e.g., a Text widget.

- Scrolled Text – show you how to create a scrolled text widget that consists of Text and vertical scrollbar widgets.
- Separator – use a separator widget to separate fields.
- Checkbox – show how to create a checkbox widget.
- Radio Button – learn how to use radio buttons to allow users to select one of a number of mutually exclusive choices.
- Combo box – walk you through the steps of creating a combo box widget.
- List box – show you how to display a list of single-line text items on a Listbox.
- Paned Window – show you how to use the Paned Window to divide the space of a frame or a window.
- Slider – learn how to create a slider by using the Tkinter Scale widget.
- Spin box – show you how to use a Spin box.
- Size grip – guide you on how to use the Size grip widget to allow users to resize the entire application window.
- Label Frame – show you how to group related widgets in a group using the Label Frame widget.
- Progress bar – show you how to use the progress bar widget to give feedback to the user about the progress of a long-running task.
- Notebook – guide you on how to use the Notebook widget to create tabs.
- Tree view – walk you through the steps of creating tree view widgets that display tabular and hierarchical data.
- Frame – learn how to use the Frame widget to group other widgets.

4. Tkinter Examples

Tkinter example – show you how to build a simple application that converts a temperature from Fahrenheit to Celsius.

5. Object-Oriented Programming with Tkinter

- Creating an object-oriented window – learn how to define an object-oriented window.
- Creating an object-oriented frame – show you how to define an object oriented Frame.
- Developing a full Tkinter object-oriented application – show you how to develop a full Tkinter object-oriented application.
- Switching between frames – guide you on how to switch between frames in a Tkinter application.

6 .Dialogs and Menus

- Displaying a message box – show you how to display various message boxes including information, warning, and error message boxes.
- Displaying a Yes/No Dialog – show you how to use the askyes no() function to display a yes/no dialog.
- Display an OK/Cancel Dialog – show you how to use the askok cancel() function to display an OK/Cancel dialog.
- Display a Retry/Cancel Dialog – show you how to use the askretry cancel() function to display a Retry/Cancel dialog.
- Show an Open File Dialog – display an open file dialog to allow users to select one or more files.
- Displaying the Native Color Chooser – show you how to display the native color-chooser dialog.
- Menu – learn how to add a menu bar and menus to a window.
- Menu button – show you how to the Menu button widget.
- Option Menu – Walk you through the steps of creating an Option Menu widget that provides a list of options in a drop-down menu.

7. Tkinter Themes and Styles

- Changing the ttk theme – how to change the default ttk theme to the new one.
- Modifying ttk style – show you how to change the appearance of widgets by modifying or extending the ttk style.
- Understanding ttk elements – help you understand ttk elements and how to use them to change the appearance of widgets.
- Modifying the appearance of a widget based on its states – show you how to dynamically change the appearance of a widget based on its specific state.

8. Advanced Tkinter Programming

- Tkinter MVC – structure a tkinter application using the MVC design pattern.
- Tkinter validation – show you how to use the Tkinter validation to validate user inputs.
- Tkinter & Matplotlib – show you how to display a bar chart from the matplotlib in Python.

9. Tkinter Asynchronous Programming

- Scheduling a task with the `after()` method – how to use the `after()` method to schedule a task that will run after a timeout has elapsed.
- Developing multithreading Tkinter Applications – show you how to use the `threading` module to develop a multithreading Tkinter application.
- Displaying a progress bar while a thread is running – walk you through the steps of connecting a progress bar with a running thread.

CHAPTER 4

4.1 EXPERIMENTAL OR MATERIAL METHODS

4.1 DESIGN METHODOLOGY

4.1.1 Existing System:

Currently, the medical works are based on the manual process, and each work is maintained in the paper. The details of purchasing drugs, audits, sell reports maintained on the paper while anyone can enter into the system and can make changes in these reports, so it is not a safe method to keep the information on the paper. The pharmacist faces problem in searching the products from the self as it is not an easy method to remember about the place of each medicine. There no Pharmacy Management System which can alert the pharmacist about the end of the drugs.

4.1.2 Proposed System:

The design of the pharmacy management system is based on the computer which will simplify the maintenance of the information, accessible and efficient. The Pharmacy Management System will provide the information about the end of the drugs in the medical so that the physician can order them drugs before the end. The pharmacist and nurses will get more accurate results at the time sell, about the details of the use of medicines and the dosages so that the system will become more reliable to use than the present system. The records of each work will be secure as to access the information the user must have to provide the ID and password in the system.

4.2 MODULE DESCRIPTION

Module 1: Pharmacy Information Management:

is a multi-functional system that helps pharmacists to keep track of medicine supplies and organize them. The modules aids in the reduction of medication errors, the improvement of patient safety, the reporting of drug usage, and the tracking of expenses.

Module 2 : Medicine Management:

To manage medicines modules will assess the need for and use of medication, the patient's response to medication, and the patient's level of understanding of the drug and how to take it with the patient.

Module 3 : Categorize Medicine Information:

Categorizing the drugs available in the pharmacy will be much easier for the admin through the help of this module. This will do the monitoring and checking of the medicine information to identify its category.

Module 4 : Manage Sales and Stocks:

This module will help the Pharmacist with the sales and stocks management that includes ordering, storing, tracking, and monitoring stock levels as well as monitoring their revenue.

Module 5: Monitor Medicine Orders:

is used to keep track of dates and events throughout the process chain, from placing an order with an external vendor to presenting goods in a store or receiving goods in a distribution center.

Module 6: Generate Processes Reports:

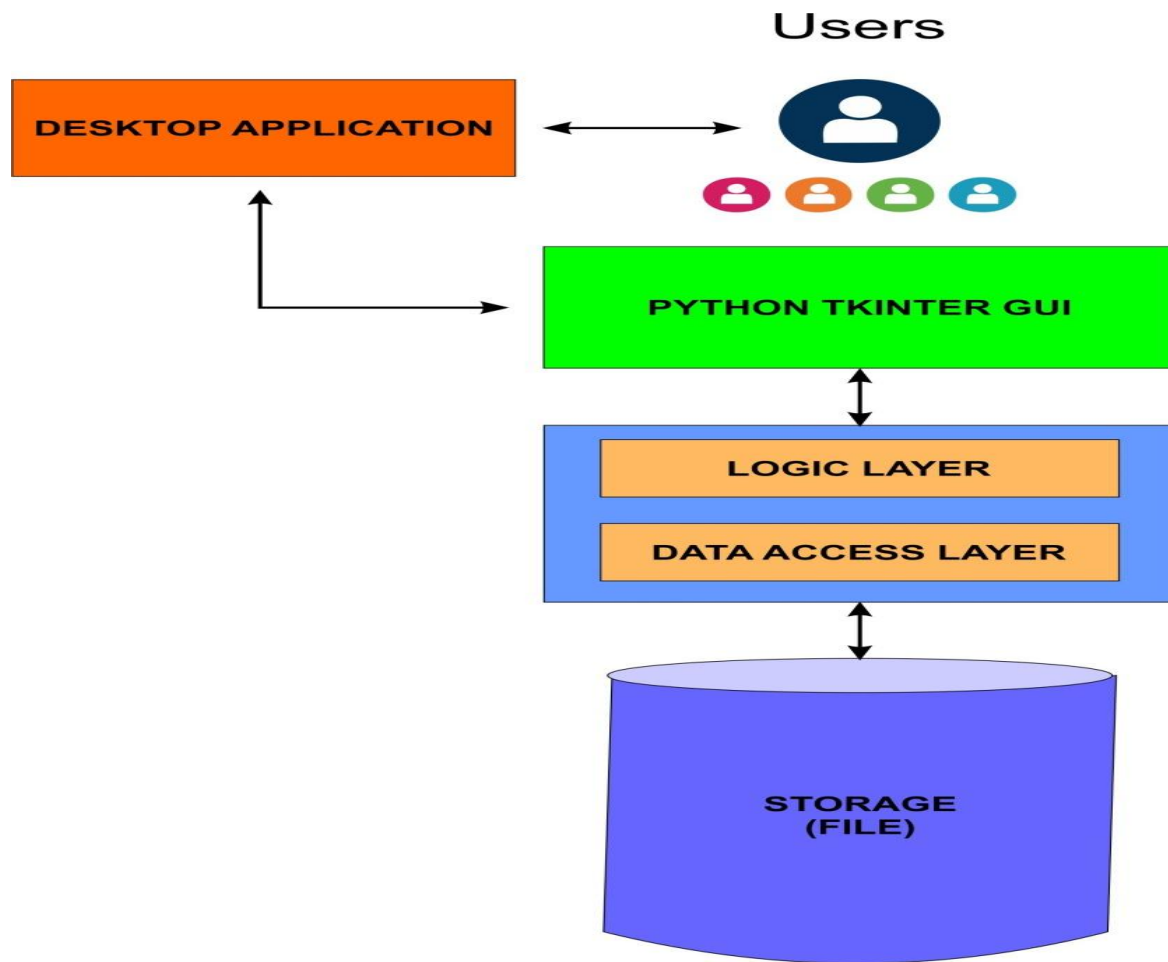
In all organization or business, reports are very essential. To help the admin in these matters, this module generates the transaction reports to keep track of the pharmacy activities.

4.3 System Implementation

The application is going to be constructed through pycharm IDE for GUI, Python for OOP and SQLite3 for database management. It is designed to improve the accuracy, enhance safety and efficiency in the pharmaceutical store. It is a computer based system which helps the pharmacist to improve inventory management, cost, medical safety. The homepage with a banner explains the app's monologue with a huge title front. users can log in to their accounts by entering their username and password. Once the user log in to application , they can see the features of this application and users can use this based on their requirements. Users can log out anytime they want and their activity inside the application will be completely secure. The application has a neat user interface for better interaction and it works smoothly.

4.4. ARCHITECTURE DIAGRAM

Architecture is defined as the art and science of designing buildings and structures. A wider definition would include within this scope the design of any built environment, structure or object, from town planning, urban design, and landscape architecture to furniture and objects.



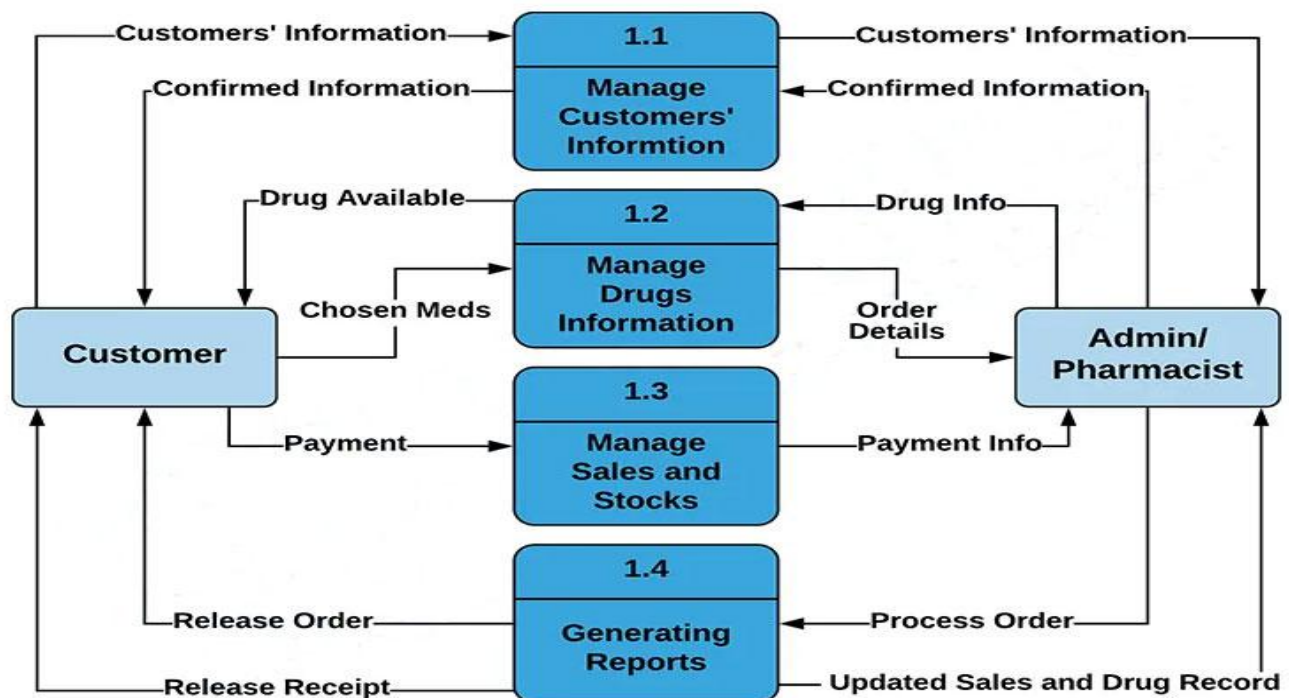
Explanation:

When user open this desktop application, it automatically connects to the python Tkinter GUI it's a graphical user interface. Tkinter is a GUI library of Python which is designed to help Python developers create system programs with a comprehensible user interface. Tkinter can easily be called the de facto standard GUI of Python. Apart from being an open-source GUI library, it is popular for its sheer simplicity and interactive graphical user interface. Then it connects to the Logic layer it's a two way connection. It is used to do calculations inside the application and also used to store full source of this

Project.Logic layer handles all of the business rules, calculations and actual logic within your application that makes it actually "do" things and it may often use some of the objects retrieved from your 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 of some kind, such as an entity-relational database. Then data taken by data access layer send the information to the admin based on the user need, then admin can easily able to maintain stocks without any loss and also able to avoid unnecessary losses. Then admin easily take invoice bill based on user needs.

4.4.1. Data flow diagram

The Data Flow Diagram (DFD) represents the flow of data and the transformations in Pharmacy management system. These transformations occurs as data enters and exits a system.In the DFD, input, processing, and output are used to represent and define the overall system.



Pharmacy Management System DFD – is created or should be created to determine the flow of data inside a system. This Pharmacy Management System will guide the programmer regarding with what to do and handle in building the system. To give you more ideas about this Pharmacy Management System .

- Managing Customer
- Managing Drugs Information
- Managing Sales and Stocks
- Generating Reports

data flow diagram level 0

In this data flow diagram you will see the general process done in Pharmacy management. This will also serve as a guide as you go through the deeper processes of the Pharmacy management system data flow diagrams.

As you see, when you build the levels of data flow diagrams, the connections of the transactions and data also broadens and gets more specific.

data flow diagram level 1

With being knowledgeable about the DFD level 1 of the Pharmacy Management System, you will know then its broaden context terms. In addition to that, this may also serve as your reference on how the inputs or data fed on the system. Then you will be also informed about the outputs that the system gives. These processes shown in the DFD were all based on the concept of Pharmacy Management System.

data flow diagram level 2

DFD level 2 let's you know the ideas on where does the data inputs goes and inputs comes within the Pharmacy management system. Considering the the dataflow levels mentioned above, you can determine well the work of breaking the processes into more specific manner.

The presented level not only shows you the detailed processes of system, but also gives you precise destination of the data that flows in the system.

This DFD will also be your references as you make your own management system DFD levels 0, 1 and 2.

CHAPTER 5

5. RESULTS AND PERFORMANCE ANALYSIS

5.1. HOMEPAGE



This is the homepage of this desktop application. The homepage covers a navigation bar at the top with the title, new customer, modify, billing, add product, search, check revenue, delete product, expiry check and logout buttons embedded on it. The UI buttons are responds smoothly.

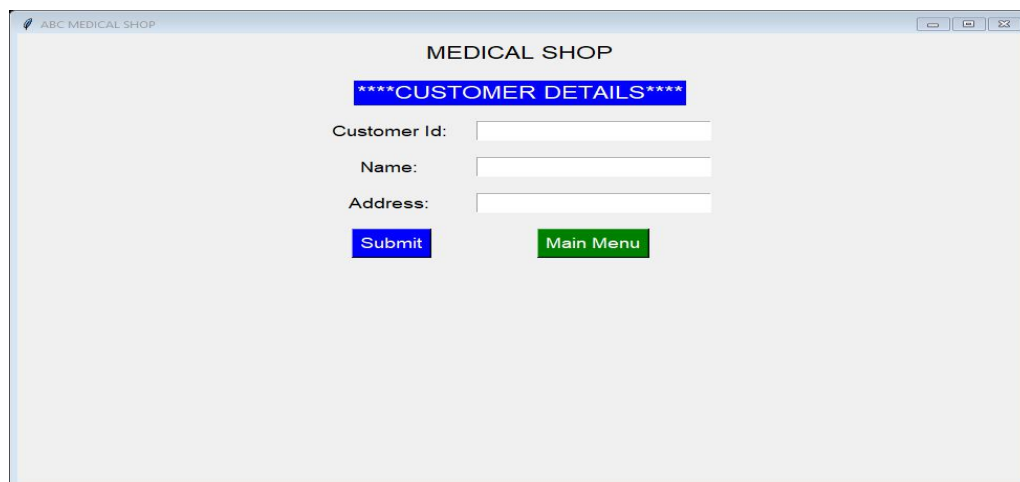
5.2. LOGIN PAGE



The screenshot shows a web browser window titled "ABC MEDICAL SHOP". The page content includes the title "ABC MEDICAL SHOP" and "PERUNGUDI" below it. There are two text input fields labeled "Username" and "Password". Below the "Username" field is a blue button labeled "Enter". Below the "Password" field is a red button labeled "Close".

The login page displays the text fields for both username and password. When admin enters the correct username and password it automatically redirect to Homepage

5.3. ADD NEWCUSTOMER



The screenshot shows a web browser window titled "ABC MEDICAL SHOP". The page content includes the title "MEDICAL SHOP" and a blue button labeled "****CUSTOMER DETAILS****". Below this button are three text input fields labeled "Customer Id:", "Name:", and "Address:". Below the "Customer Id:" field is a blue button labeled "Submit". Below the "Name:" field is a green button labeled "Main Menu".

This how new customers are created in this application by entering their new customer ID, name and address, then press the submit button to complete the process. It automatically save to database.

5.4. ADD NEWPRODUCT

ABC MEDICAL SHOP

Name: Type:

Quantity Left: Cost:

Purpose: Expiry Date:

Rack location: Manufacture:

Reset **Submit** **Refresh stock** **Main Menu**

Name	Type	Quantity Left	Cost	Purpose
1. Dolo 650	Tablet	613	9.46	Fever
2. Gelusil	Syrup	20	9.5	Gastric
3. Anacin	Tablet	71	3.0	Headache

This is how new products are created by entering their medicine name, quantity of that product, purpose of that medicine, rack location, type of the medicine, cost of the product, expiry date of that product and manufacturer of the medicine, at last click the submit button to finish the entire process.

5.5. DELETE PRODUCT

ABC MEDICAL SHOP

Delete a product from Stock

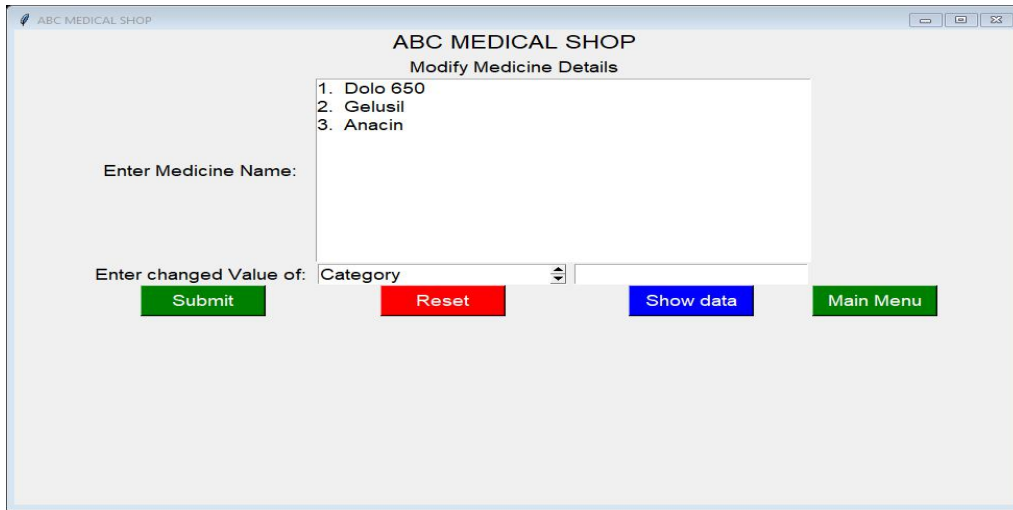
Product:

Qty. Exp.dt. Cost

Delete **Main Menu**

This is how admin can able to delete the unnecessary medicine by entering their product name and quantity of the medicine want to delete. By pressing the delete button to finish the process.

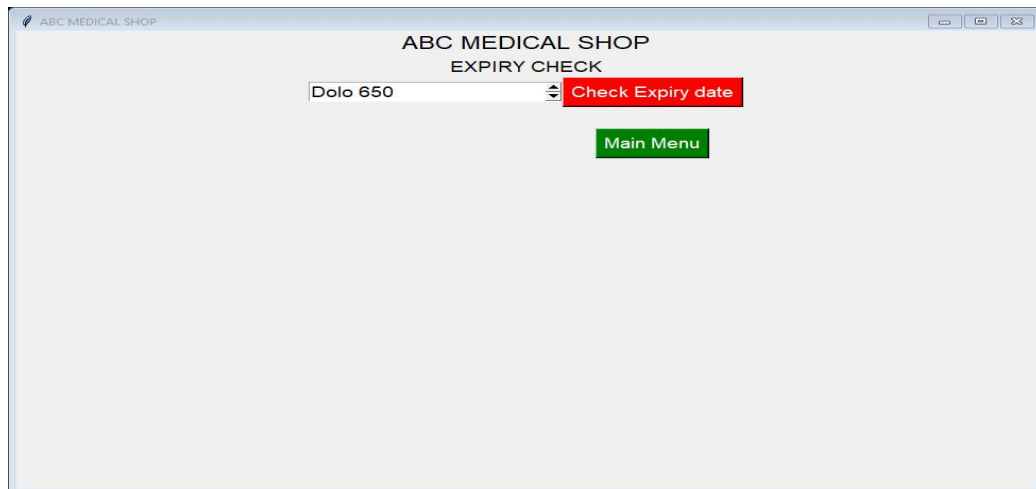
5.6. MODIFY PRODUCT



The screenshot shows a web application window titled "ABC MEDICAL SHOP" with a subtitle "Modify Medicine Details". Inside the window, there is a list of three items: "1. Dolo 650", "2. Gelusil", and "3. Anacin". Below this list, there is a text input field labeled "Enter Medicine Name:". To the right of this field is a dropdown menu currently showing "Category". Below the input field and dropdown, there are four buttons: "Submit" (green), "Reset" (red), "Show data" (blue), and "Main Menu" (green). The window also has standard OS window controls (minimize, maximize, close) in the top right corner.

This is how admin can able to modify the medicine details by entering their name of the medicine they want to change and enter changed value of that product then press submit button to finish the process.

5.7. EXPIRY CHECK



The screenshot shows a web application window titled "ABC MEDICAL SHOP" with a subtitle "EXPIRY CHECK". Inside the window, there is a dropdown menu currently showing "Dolo 650". To the right of this dropdown is a red button labeled "Check Expiry date". Below these elements is a green button labeled "Main Menu". The window also has standard OS window controls (minimize, maximize, close) in the top right corner.

This is how admin can able to check the upcoming expiry of the medicine, by entering their name of the medicine and press the button it will show the all about the medicine.

5.8. BILLING SYSTEM

The screenshot shows a software window titled "ABC MEDICAL SHOP". Inside, there is a section labeled "BILLING SYSTEM" in a green box. Below this, there are input fields for "Enter Name:", "Enter Address:", and "Customer ID (if available)". To the right of these fields are three empty text boxes. Below the input fields is a table with three columns: "SELECT PRODUCT", "RACK STOCK COST", and "QUANTITY". The "SELECT PRODUCT" column has a dropdown menu. Below the table are three buttons: "Add to bill" (blue), "Save Bill" (blue), and "Main Menu" (green).

This is how admin can able to print the invoice bill by entering customer name, customer address, customer ID if available, select the customer bought medicine name in the available stocks and select the quantity and press the save bill button to save the bill, if you have any source to print the bill it will automatically print.

5.9. CHECK REVENUE

The screenshot shows a software window titled "ABC MEDICAL SHOP". Inside, there is a section labeled "SEARCH FOR MEDICINE" in a green box. Below this, there are two lines of text: "Today: 28-1-2022" and "Total revenue: 0.00". Below these lines is a green button labeled "Main Menu".

This is how admin can able to check overall revenue of the day to reduce unwanted loss of money.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1. Conclusion

This Pharmacy management system database design was made based on managing pharmacy requirements. The system can encode both customers and medicines' information. Pharmacy admin can have access to the medicines status and information in terms of the number sold and remains to identify the inventories of sales and stocks. They can handle the data needed in managing pharmacy as well as the sales of medicines and customer records. Pharmacy management system is actually a desktop application which handle the essential data and save the data and actually about the database of a pharmacy and it's management. This software help in effectively management of the pharmaceutical store or shop. It provide the statistics about medicine or drugs which are in stocks which data can also be updated and edited. It works as per the requirement of the user and have options accordingly. It allow user to enter manufacturing as well as the expiry date of medicine placing in stock and for sales transaction. This software also have ability to print the bill and invoices etc. The record of suppliers supplies can also be saved in it. There are other function available too . The main purpose is effectively and easily handling of pharmacy data and it's management.

6.2. Future enhancements

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.

REFERENCES

- Python For Desktop Applications: How to develop, pack and deliver Python applications with TkInter by Tran Duc Loi.
- Python in A Nutshell: A Desktop Quick Reference, Third Edition by Alex Martelli , Anna Ravenscroft , Steve Holden.
- The Python Language Reference Manual (version 3.2) by Guido van Rossum, and Fred L. Drake, Jr. (Editor)
- Barbara Griggs.
Green Pharmacy: The History and Evolution of Western Herbal Medicine, Second Edition. Viking press, (1982). Pg 93-97.
- Charles E. Rosenberg, Morris J. Vogel.
The Therapeutic Revolution: Essays in the socialhistory of American Medicine, Second Edition. University of Pennsylvania press, (1979). Pg 174.
- Peter G Homan, Briony Hudson, Raymond C Row.
Popular Medicines: An illustrated
- Leslie G. Mathews.
History of Pharmacy in Britain. Edinburgh, E&S. Livingstone(1962). Pg 213-223.
- History, Fourth Edition. Trade paperback, Pharmaceutical press (2008). Pg. 127 .
- Stuart Anderson.
Making Medicines: A Brief History of Pharmacy and Pharmaceuticals,Sixth Edition. Jessica Kingsley publisher, (2005). Pg 214.
- Charles E. Rosenberg.
History of Medicine and Allied Sciences, Seventh Edition.University of Pennsylvania press, (2003). Pg 197.
- George A. Bender.
Great moments in Medicine, Cambridge university press (1967). Pg118.

APPENDIX

A . SOURCE CODE

```
from tkinter import *
from tkinter import font
import time
from datetime import datetime
import sqlite3
import ctypes
import random
from tkinter import messagebox
from tkinter import ttk

user32 = ctypes.windll.user32
xpos = int((user32.GetSystemMetrics(0) - 960) / 2)
ypos = int((user32.GetSystemMetrics(1) - 600) / 2)
f = "
flag = "
flags = "
login = sqlite3.connect("admin.db")
l = login.cursor()
c = sqlite3.connect("medicine.db")
cur = c.cursor()
columns = ('SI No', 'Name', 'Type', 'Quantity Left', 'Cost', 'Purpose', 'Expiry Date', 'Rack
location', 'Manufacture')
def open_win():
    global win, flag, ft, hft
    flag = 'win'
    win = Tk()
    win.geometry('960x600+{}+{}'.format(xpos, ypos))
    win.title('ABC MEDICAL SHOP')
```

```

ofrm=Frame(win)
hft = font.Font(size=18)
ft = font.Font(size=14)

Label(ofrm, font=hft,text="ABC MEDICAL SHOP").grid(row=0, column=0,
columnspan=3, padx= 10, pady= 10)

Button(ofrm, font=ft,text='New Customer', width=25, bg='green', fg='white',
command=newcustomer).grid(row=4, column=0, padx= 10, pady= 10)
Button(ofrm, font=ft,text='Add product ', bg='green', fg='white', width=25,
command=stock).grid(row=5, column=0)
Button(ofrm, font=ft,text='Delete product ', bg='red', fg='white', width=25,
command=delete_stock).grid(row=6,

column=0)

Button(ofrm,font=ft, text='Modify', width=15, bg='blue', fg='white',
command=modify).grid(row=4, column=1, padx= 10, pady= 10)
Button(ofrm,font=ft, text='Search', width=15, bg='blue', fg='white',
command=search).grid(row=5, column=1)
Button(ofrm,font=ft, text='Expiry Check', bg='red', fg='white', width=15,
command=exp_date).grid(row=6, column=1)

Button(ofrm, font=ft,text="Check Revenue", bg='blue', fg='white', width=20,
command=show_rev).grid(row=5,column=2, padx= 10, pady= 10)
Button(ofrm, font=ft,text='Billing', width=20, bg='blue', fg='white',
command=billing).grid(row=4, column=2)
Button(ofrm, font=ft,text='Logout', bg='red', fg='white', width=20,
command=again).grid(row=6, column=2)

ofrm.pack()

```

```

win.mainloop()
def delete_stock():
    global cur, c, flag, lb2, d, ft, hft, serials, medicines, details
    win.destroy()
    flag = 'd'
    d = Tk()
    d.geometry('960x600+{}+{}'.format(xpos, ypos))
    d.title('ABC MEDICAL SHOP')
    dfrm=Frame(d)
    hft = font.Font(size=18)
    ft = font.Font(size=14)

    Label(dfrm, font=hft, text='ABC MEDICAL SHOP').grid(row=0,
column=0,columnspan=5)
    Label(dfrm, font=ft, text='Delete a product from Stock').grid(row=1,
column=0,columnspan=5)
    Label(dfrm, font=ft, text='Product').grid(row=2, column=0)
    lb2=ttk.Combobox(dfrm,font=ft)
    lb2.grid(row=2,column=1)
    lb2.bind("<<ComboboxSelected>>", getmedicine)
    #Entry(dfrm, font=ft).grid(row=2, column=1)

    Label(dfrm, font=ft, text='Qty. Exp.dt. Cost ').grid(row=3,
column=1,columnspan=4)
    details=Label(dfrm,font=hft, text="", width=30, bg='white')
    details.grid(row=4, column=1,columnspan=4)
    Button(dfrm, font=ft,width=20, text='Main Menu', bg='green', fg='white',
command=main_menu).grid(row=5, column=3)
    Button(dfrm, font=ft,width=20, text='Delete', bg='red', fg='white',
command=delt).grid(row=5, column=1)

```

```
loadmedicine()  
#print(medicines)  
lb2['values']=medicines  
dfrm.pack()  
d.mainloop()
```

```
def loadmedicine():  
    global serials,medicines,lb2  
    serials=[]  
    medicines=[]  
    cur.execute("select *from medicine")  
    for i in cur:  
        serials.append(i)  
        medicines.append(i[1])  
    c.commit()
```

```
def getmedicine(evt):  
    global lb2,details,serials,sl2,sl1  
    data=serials[lb2.current()]  
    details['text']='{} - {} - {}'.format(data[3], data[6], data[4])  
    sl2=data[0]  
    sl1=data[0]
```

```
def getmedicinedetails(evt):  
    global lb2,details,serials,sl2  
    data=serials[lb2.current()]  
    details['text']='{} - {} - {}'.format(data[7], data[3], data[4])  
    sl2=data[0]
```

```
def sel_del(e):  
    global lb1, d, cur, c, p, sl2
```

```

p = lb1.curselection()
x = 0
sl2 = ""
cur.execute("select * from medicine")
for i in cur:
    # print(x, p[0])
    if x == int(p[0]):
        sl2 = i[0]
        break
    x += 1
c.commit()
# print(sl2)
Label(d, text=' ', bg='white', width=20).grid(row=0, column=1)
cur.execute('Select * from med')
for i in cur:
    if i[0] == sl2:
        Label(d, text=i[0] + '. ' + i[1], bg='white').grid(row=0, column=1)
c.commit()
def delt():
    global p, c, cur, d
    cur.execute("delete from medicine where serial=?", (sl2,))
    c.commit()
    loadmedicine()
def modify():
    global cur, c, accept, dfrm, flag, att, up, n, name_, win, st, col, col_n, mrows
    col = ('category', 'quantity', 'cost', 'purpose', 'expdate', 'location', 'manufacturer')
    col_n = ('Category', 'Quantity', 'Cost', 'Purpose', 'Expiry Date', 'Rack location',
'Manufacturer')
    flag = 'st'
    name_ = ""

```