

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI-590018**



**A MINI PROJECT REPORT ON**

**“STOCK CONTROL MANAGEMENT SYSTEM”**

**Submitted By**

**BHOOMIKA M      4RA20IS005**

**MUKTHI A      4RA20IS0013**

To the Visvesvaraya Technological University during the academic year 2022-23 in  
partial fulfilment for the award of

**DATABASE MANAGEMENT SYSTEMS**

**(V SEMESTER)**

**In**

**INFORMATION SCIENCE ENGINEERING**

**Under the Guidance of**

**Mrs. SHRUTHI H S**

**Assistant Professor Dept. of CSE**

**Rajeev Institute of Technology**

**Hassan**



**RAJEEV INSTITUTE OF TECHNOLOGY**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**HASSAN-573201**

**2022-2023**

# RAJEEV INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, Belagavi and Approved by AICTE,  
New Delhi) Hassan-573201, Karnataka, India.



## CERTIFICATE

This Certified that the project work entitled “**STOCK CONTROL MANAGEMENT SYSTEM**” is carried out by **BHOOMIKA M** [4RA20IS005] and **MUKTHI A** [4RA20IS013] respectively, a bonafide students of **RAJEEV INSTITUTE OF TECHNOLOGY**, Hassan in partial fulfillment for the subject **DATABASE MANAGEMENT SYSTEMS** in **INFORMATION SCIENCE AND ENGINEERING** of the Visvesvaraya Technological University, Belagavi during the year 2022-2023. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said Degree.

**Mrs. SHRUTHI H. S**

Assistant Professor

Dept. of CSE

RIT HASSAN

**Dr. ARJUN B C**

Head of the Department

Dept. of ISE

RIT HASSAN

**Dr. MAHESH P. K**

Principal

RIT HASSAN

**Name of Examiners**

**Signature of Examiners with date**

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

## **DECLARATION**

We **BHOOMIKA M**, and **MUKTHI A**, students of Fifth Semester B.E, Department of Information Science and Engineering, Rajeev institute of technology, Hassan declare that the Mini Project entitled “**STOCK CONTROL MANAGEMENT SYSTEM**” has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in Bachelor of Engineering in Information Science and Engineering from Visvesvaraya Technological University, Belagavi during the academic year 2022-2023.

**BHOOMIKA M [4RA20IS005]**

**MUKTHI A [4RA20IS0013]**

Place: Hassan

Date:

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance crowned our efforts with success.

We would like to profoundly thank to our Management of RIT & our president Dr. **Rachana Rajeev** for providing such a healthy environment.

We express our deep and sincere thanks to our Principal **Dr. Mahesh P K**, Rajeev Institute of Technology for his encouragement.

We are grateful to **Dr. Arjun B C**, Head of the Department of ISE for providing a good working environment and for his constant support and encouragement.

We wish to express our gratitude to **Mrs. Shruthi H S**, Assistant Professor, Department of CSE for guiding and encouragement that leads us to complete this work.

We would like to thank the faculty members and supporting staff of the Department of ISE, for providing all the support for completing the Project work. Finally, we are grateful to our parents and friends for their unconditional support and help during the course of our Project work.

**BHOOMIKA M (4RA20IS005)**

**MUKTHI A (4RA20IS0013)**

## **ABSTRACT**

A stock control management system (or stock system) is the process by which you track your goods throughout your entire supply chain, from purchasing to production to end sales. It governs how you approach stock control management for your business. Stock control management is a system that gives clear cut information about sales and stocks to an admin. In the database the admin can add, delete, and update the information accordingly. The system also provides search option to figure out the details of the product be like its stock, sales, supplier etc. Spreadsheets, hand-counted stock levels and manual order placement have largely been replaced by advanced inventory tracking software. A stock control management system can simplify the process of ordering, storing, and using stock by automating end-to-end production, business management, demand forecasting and accounting. The key features of this project will be stock tracking, Order management, provides supplier and products details, billing facilities and many more.

# TABLE OF CONTENTS

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
	<b>ABSTRACT</b>	ii
	<b>LIST OF TABLES</b>	iv
	<b>LIST OF FIGURES</b>	v
<b>1.</b>	<b>INTRODUCTION</b>	
	1.1 PROBLEM STATEMENT	1
	1.2 INTRODUCTION TO DBMS AND SQLITE	1
	1.3 INTRODUCTION TO FRONTEND SOFTWARE	2
	1.4 PROJECT REPORT OUTLINE	4
<b>2.</b>	<b>REQUIREMENT SPECIFICATION</b>	5
	2.1 SOFTWARE REQUIREMENTS	5
	2.2 HARDWARE REQUIREMENTS	5
<b>3.</b>	<b>OBJECTIVE OF THE PROJECT</b>	6
<b>4.</b>	<b>IMPLEMENTATION</b>	7
	4.1 ER DIAGRAM	8
	4.2 MAPPING OF THE ER SCHEMA TO RELATIONS	10
	4.3 MAPPING OF ER DIAGRAM TO SCHEMA DIAGRAM	14
	4.4 CREATION OF TABLES	15
	4.5 INSERTION OF TUPLES	17
	4.6 CREATION OF TRIGGERS	19
	4.7 CREATION OF STORED PROCEDURES	19
<b>5.</b>	<b>FRONT END DESIGN</b>	
	5.1 FRONT END CODE	20
	5.2 CONNECTIVITY TO DATABASE	21
<b>6.</b>	<b>TESTING</b>	
	6.3 TEST CASES FOR THE PROJECT	23
<b>7.</b>	<b>RESULTS</b>	
	7.1 SNAPSHOTS	24
<b>8.</b>	<b>CONCLUSION</b>	32
<b>9.</b>	<b>REFERENCES</b>	33

## **LIST OF TABLES**

<b>TABLEN.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
4.1	Employee Table	15
4.2	Category Table	15
4.3	Register Table	15
4.4	Product Table	16
4.5	Supplier Table	16
4.6	Trigger Insert	19
6.3	Test Cases	23

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
4.1	ER diagram of STOCK CONTROL MANAGEMENT SYSTEM	9
4.3	Schema Diagram of Stock control Management System	14
7.1	User Login	24
7.2	User Registration	25
7.3	Admin Dashboard	26
7.4	Employee Details	27
7.5	Category Interface	28
7.6	Product Section	29
7.7	Sales And Stocks	30
7.8	Billing System	31



# CHAPTER 1

## INTRODUCTION

Stock control management is a system that gives clear cut information about sales and stocks to an admin. In the database the admin can add, delete, update the information accordingly. The system also provides search option to figure out the details of the product be like its stock, sales, supplier etc.

### 1.1 PROBLEM STATEMENT

The main aim of the “Stock control management system” is to make the interface easy and convenient for the admin to do all the operation on employee and stocks data.

### 1.2 INTRODUCTION TO DBMS AND SQLITE

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update, and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

SQLITE is a relational database management system (RDBMS) contained in a python Library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program. SQLite3 can be integrated with Python using sqlite3 module, which was written by Gerhard Haring. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards.

```
#!/usr/bin/py

thonimport

sqlite3

con = sqlite3.connect('test.db')
```

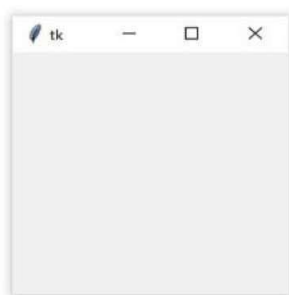
## 1.3 INTRODUCTION TO FRONT END SOFTWARE

Python is a scripting terminology or language like PHP, Perl, Ruby and many more. It can be used for web development. Python language was used to do website dynamic. if you want to begin composing web programs in Python, you can either use GUI or use one of its many web app frameworks.

This is one of the most popular GUI tools you can use with Python, it is a lightweight tool, but it does have quite a bit of functionality behind it. First off, Tkinter comes preinstalled with Python, so you don't have to install any dependencies (unless there are additional packages you want to add). Also, it does have a bit of a learning curve, but from my experience there is a lot you can do with this package. Here is an example that shows what the window looks like after importing the package:

```
Python >>>
>>> window = tk.Tk()
```

When you execute the above code, a new window pops up on your screen. How it looks depends on your operating system:



(a) Windows















(b) macOS



(c) Ubuntu

## Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table –

Sr.No.	Operator & Description
1	Button  The Button widget is used to display buttons in your application.
2	Canvas  The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	Checkbutton  The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	Entry  The Entry widget is used to display a single-line text field for accepting values from a user.
5	Frame  The Frame widget is used as a container widget to organize other widgets.
6	Label  The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	Listbox  The Listbox widget is used to provide a list of options to a user.
8	Menubutton  The Menubutton widget is used to display menus in your application.
9	Menu  The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	Message  The Message widget is used to display multiline text fields for accepting values from a user.
11	Radiobutton  The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	Scale  The Scale widget is used to provide a slider widget.

## **1.4 PROJECT REPORT OUTLINE**

The report is arranged in the following way:

Chapter 1: Introduction to data base management system and sqlite3 module in python.

Chapter 2: Requirement specification of hardware and software.

Chapter 3: Objective of the Project.

Chapter 4: Implementation of ER diagram and its description

Chapter 5: Front End Design, connecting to database using Python, Front end code of the Project

Chapter 6: Testing of project by different cases, its process and testing objectives

Chapter 7: Result of the Project

## **CHAPTER 2**

### **REQUIREMENT SPECIFICATION**

#### **2.1 HARDWARE REQUIREMENTS**

- Processor : Pentium IV or above
- RAM : 2GB or above
- Hard Disk : 3GB or more
- Output Device : Monitor VGA of 1024X768 screen resolution
- Input Device : Keyboard and mouse

#### **2.2 SOFTWARE REQUIREMENTS**

- Database : SQLite
- Tools : Microsoft visual studio code
- Operating System : Windows 10
- Programming Lang. : Python

## CHAPTER 3

### OBJECTIVE OF THE PROJECT

The main objective of this application is to:

1. Automate the complete operations of billing and managing stock. The business or the companies maintain hundreds of thousands of records. Also searching should be very faster so they can find required details instantly.
2. To develop a graphical user-based interface to facilitate the co-ordination between supply of products and the customer. This system makes conveniently available good customer experience, not having enough stock to fulfill orders you've already payment for can be a real negative, with this stock control management system the problem regarding the stocks can be easily resolved.
3. This will also serve in improving cash flow; cash flow can be monitored and recorded.
4. The interface avoids shrinkage, purchasing too much of the wrong stock and/or not storing it correctly can lead to it becoming 'dead', spoiled, or stolen.
5. Optimizing fulfillment, stock that's put away and stored correctly can be picked, packed, and shipped off to customers more quickly and easily.
6. The system will provide the admin to look at the details of the employees working, sales details, billing of customer, products or stocks that are actively sold, and inactive goods. Also, the supplier details who are engaged in contribution of goods and products that the business is accustomed to.
7. This system allows the add employees, products, categories and supplies. It also allows to modify the records and the admin can alter all the system data. The employees can only view the billing section to make sales and to generate and print bills of customers.
8. The Stock control Management System project report contains information related to sales and stocks such as Customer Bills, Products, Supplier Details, Employee Details, and Category Details etc.

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 ER DIAGRAM

The following ER Diagram shows the entity relationships of STOCK CONTROL MANAGEMENTSYSTEM.

Let the attributes of the EMPLOYEE entity be E\_ID, NAME, EMAIL, GENDER, CONTACT, DOB, DOJ, ADDRESS, UTYPE, SALARY, PASS

Similarly, let the attributes of PRODUCT entity be PID, SUPPLIER, CATEGORY, NAME, PRICE, QTY, STATUS

REGISTER entity attributes be ENAME, USERTYPE, USERID, UPASSCATEGORY entity attributes be CID, NAME

And finally, SUPPLIER entity attributes be INVOICE, NAME, CONTACT, DESC

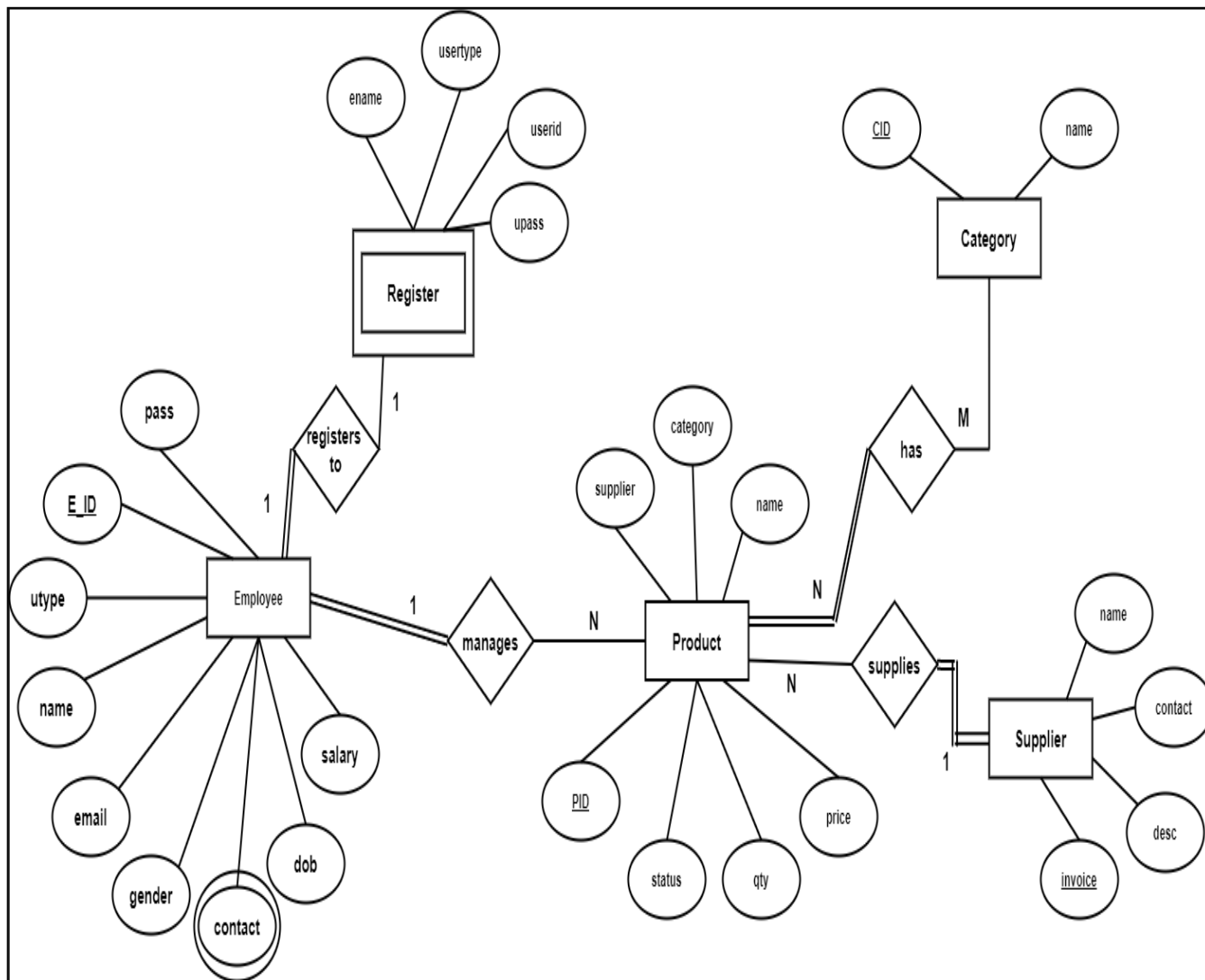
Let the 1<sup>st</sup> relationship be REGISTERS TO between EMPLOYEE and REGISTER entities i.e., Employee registers to Registration with cardinality ratio 1:1, Partial Participation from Employee and Total Participation from Register.

2<sup>nd</sup> relationship be MANAGES between EMPLOYEE and PRODUCT entities i.e. Employee manages Product with cardinality ratio N:1, Total Participation from Product and Partial Participation from Supplier.

3<sup>rd</sup> relationship be HAS between PRODUCT and CATEGORY entities i.e. Product has Category with cardinality ratio M:N, Total Participation from Product and Total Participation from Category.

The last relationship be SUPPLIES between SUPPLIER and PRODUCT entities i.e. Supplier supplies Product with cardinality ratio 1: N, Total Participation for, Supplier and Partial Participation from Product.

E\_ID, PID, CID, INVOICE are primary keys for the respective entities as they have only unique values. Register is the weak entity in the respective ER DIAGRAM as it has no unique values or primary key.



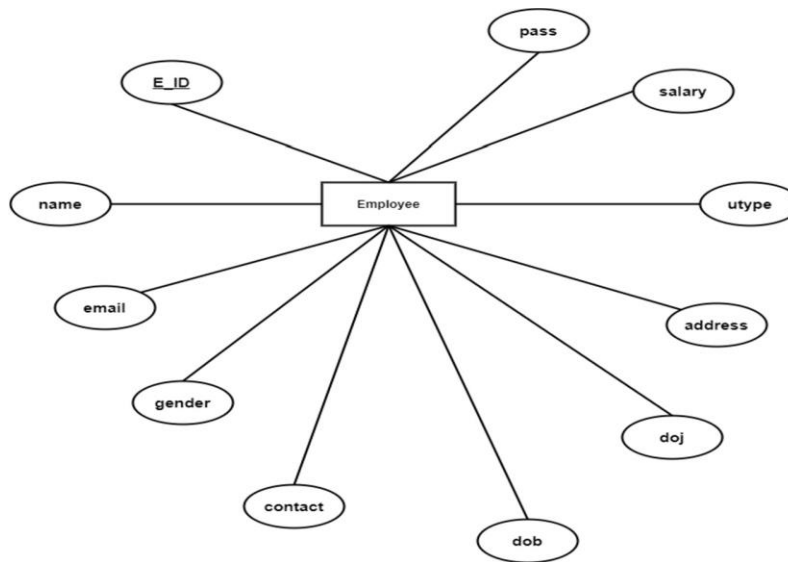
**Fig 4.1 ER DAIGRAM OF STOCK CONTROL MANAGEMENT SYSTEM**



## 4.2 MAPPING OF THE ER SCHEMA TO RELATIONS

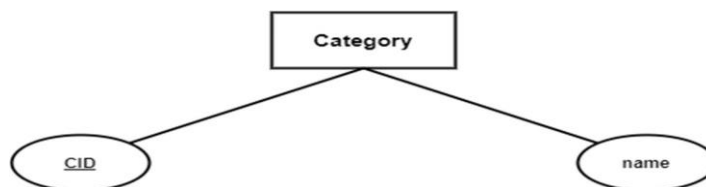
**Mapping:** - The conceptual/internal mapping defines the correspondence between the conceptual view and the store database. It specifies how conceptual record and fields are represented at the internal level. There could be one mapping between conceptual and internal levels.

### STEP 1: MAPPING OF REGULAR ENTITY TYPE



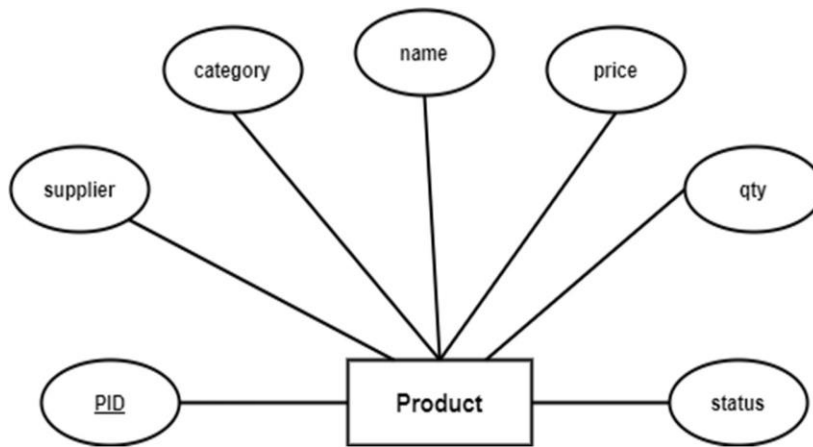
EMPLOYEE

<u>E_ID</u>	NAME	EMAIL	CONTACT	GENDER	DOB	DOJ	UTYPE	PASS	SALARY
-------------	------	-------	---------	--------	-----	-----	-------	------	--------



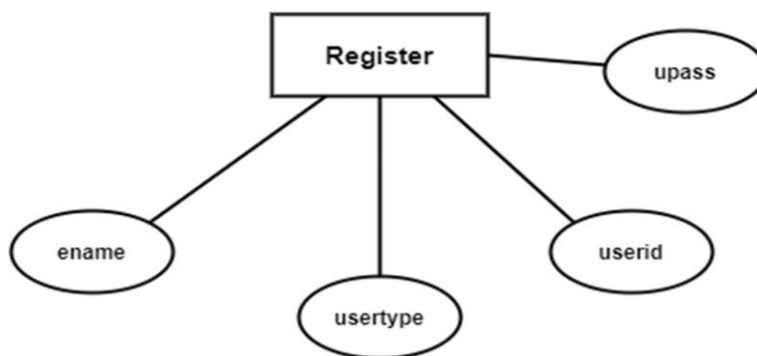
CATEGORY

<u>CID</u>	NAME
------------	------



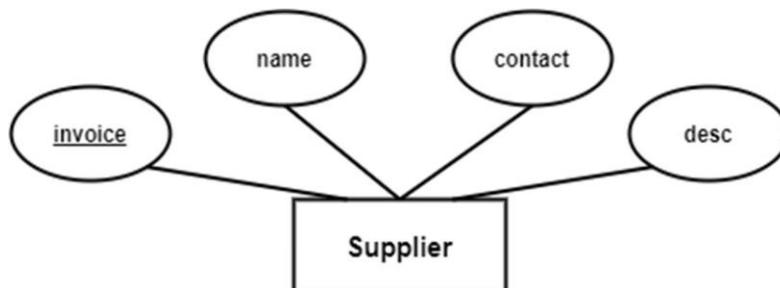
PRODUCT

<u>PID</u>	SUPPLIER	CATEGORY	NAME	PRICE	QTY	STATUS
------------	----------	----------	------	-------	-----	--------



REGISTER

ENAME	USERTYPE	USERID	UPASS
-------	----------	--------	-------



SUPPLIER

<u>INVOICE</u>	NAME	CONTACT	DESC
----------------	------	---------	------

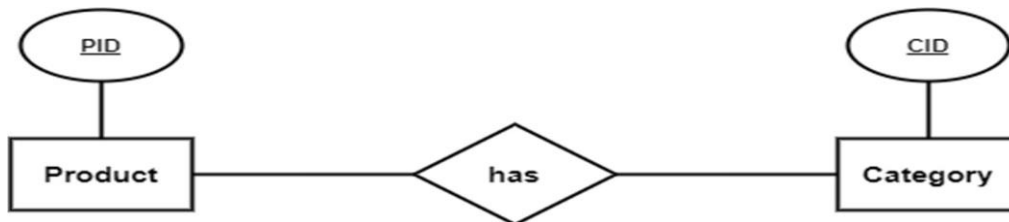
## STEP 2: MAPPING OF WEAK ENTITY TYPES

One Employee Registers only once



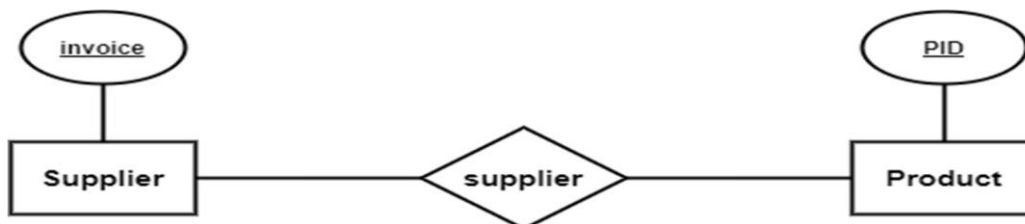
## STEP 3: MAPPING OF 1:1 RELATIONS

One Product has One Category



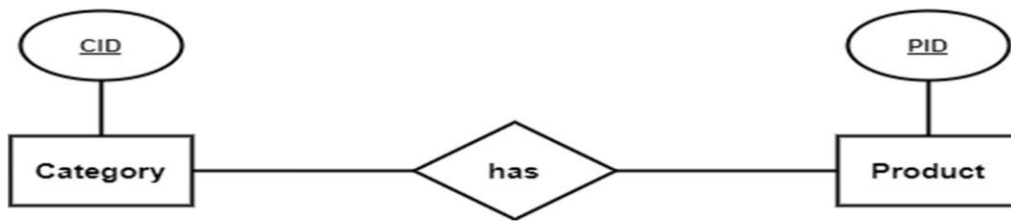
## STEP 4: MAPPING OF N:1 RELATIONS

One Supplier supplies different Products

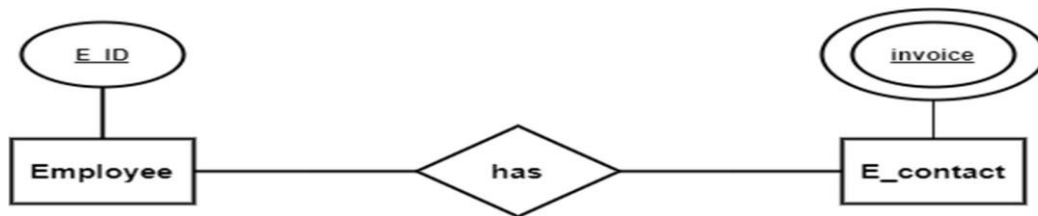


## STEP 5: MAPPING OF M: N RELATIONS

Many Categories has many Products



## STEP 6: MAPPING OF MULIT-VALUED ATTRIBUTES



## STEP 7: MAPPING OF N-ARRAY RELATIONS

No N-array relational types

## 4.3 MAPPING OF ER DIAGRAM TO SCHEMA DIAGRAM

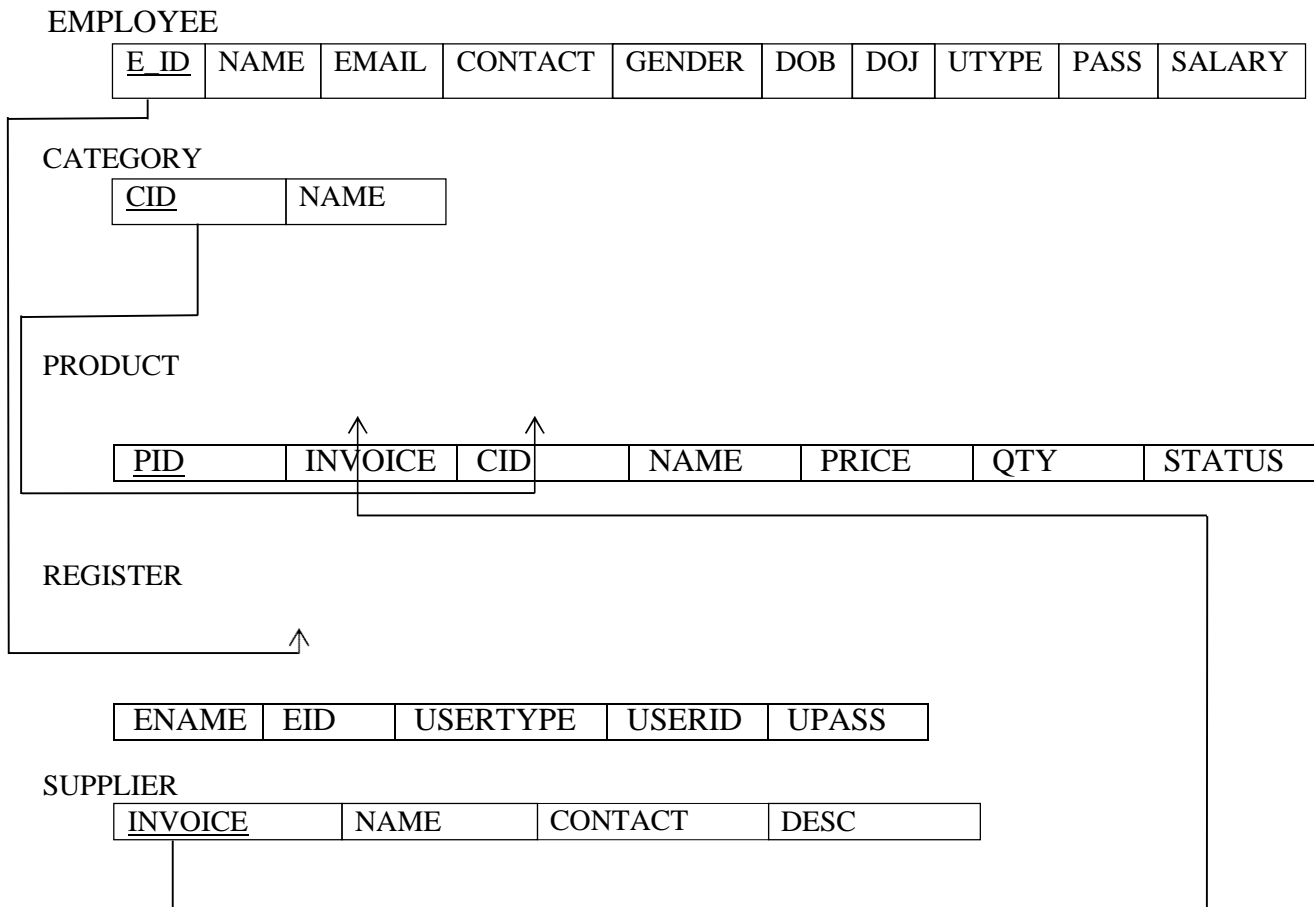


Fig 4.3 Schema Diagram of Stock control Management System

## 4.4 CREATION OF TABLES

1. CREATE TABLE employee (eid INTEGER PRIMARY KEY AUTOINCREMENT, name text, emailtext, contact text, dob text, doj text, utype text, address text, salary text)

employee	CREATE TABLE employee(eid INTEGER PRIMARY KEY AUTOINCREMENT,name text,email text,gender text,contact text,dob text,doj text,utype text,address text,salary text)	
eid	INTEGER	"eid" INTEGER
name	text	"name" text
email	text	"email" text
gender	text	"gender" text
contact	text	"contact" text
dob	text	"dob" text
doj	text	"doj" text
utype	text	"utype" text
address	text	"address" text
salary	text	"salary" text

**Table 4.1 Employee Table**

2. CREATE TABLE category (cid INTEGER PRIMARY KEY AUTOINCREMENT, Name text)

category	CREATE TABLE category(cid INTEGER PRIMARY KEY AUTOINCREMENT,name text)	
cid	INTEGER	"cid" INTEGER
name	text	"name" text

**Table 4.2 Category Table**

3. CREATE TABLE product (pid INTEGER PRIMARY KEY AUTOINCREMENT, supplier text, categorytext, name text, price text, qty text, status text)

product	CREATE TABLE product(pid INTEGER PRIMARY KEY AUTOINCREMENT,supplier text,category text,name text,price text,qty text,status text)	
pid	INTEGER	"pid" INTEGER
supplier	text	"supplier" text
category	text	"category" text
name	text	"name" text
price	text	"price" text
qty	text	"qty" text
status	text	"status" text

**Table 4.3 Product Table**

4. CREATE TABLE register (eid INTEGER PRIMARY KEY AUTOINCREMENT, ename text, usertype text, userid text, upass text)

register		CREATE TABLE register(eid INTEGER PRIMARY KEY AUTOINCREMENT,ename text,usertype text,userid text,upass text)
eid	INTEGER	"eid" INTEGER
ename	text	"ename" text
usertype	text	"usertype" text
userid	text	"userid" text
upass	text	"upass" text

**Table 4.4 Register Table**

5. CREATE TABLE supplier (invoice INTEGER PRIMARY KEY AUTOINCREMENT, Name text,contact text, desc text)

supplier		CREATE TABLE supplier(invoice INTEGER PRIMARY KEY AUTOINCREMENT,name text,contact text,desc text)
invoice	INTEGER	"invoice" INTEGER
name	text	"name" text
contact	text	"contact" text
desc	text	"desc" text

**Table 4.5 Supplier Table**

TOTAL TABLES CREATED ARE: -

- 1)EMPLOYEE
- 2)CATEGORY
- 3)PRODUCT
- 4)REGISTER
- 5)SUPPLIER

Tables (6)	
category	CREATE TABLE category(cid INTEGER PRIMARY KEY AUTOINCREMENT,name text)
employee	CREATE TABLE employee(eid INTEGER PRIMARY KEY AUTOINCREMENT,name text,email text,gender text,contact text,dob text,doj text,uhype text,address text,salary text)
product	CREATE TABLE product(pid INTEGER PRIMARY KEY AUTOINCREMENT,supplier text,category text,name text,price text,qty text,status text)
register	CREATE TABLE register(eid INTEGER PRIMARY KEY AUTOINCREMENT,name text,uhype text,userid text,upass text)
sqlite_sequence	CREATE TABLE sqlite_sequence(name,seq)
supplier	CREATE TABLE supplier(invoice INTEGER PRIMARY KEY AUTOINCREMENT,name text,contact text,desc text)

## 4.5 INSERTION INTO TUPLES

### I) Employee

INSERT INTO EMPLOYEE VALUES

(101,'gaayana','gaayu@gmail.com','Female',9481041406,'04-11-2000','09-11-2021','Employee','abc...',70000);

INSERT INTO EMPLOYEE VALUES

(102,'jayanth','jay@gmail.com','Male',8945672345,'27-12-2001','09-11-2021','Admin','xyz...',90000);

INSERT INTO EMPLOYEE VALUES

(103,'bhavya','bhavz@gmail.com','Female',7894625678,'13-01-2002','09-11-2021','Employee','pqr...',90000);

	eid	name	email	gender	contact	dob	doj	utype	address	salary
1	101	gaayana	gaayu@gmail.com	Female	9481041406	04-11-2000	09-11-2021	Employee	abc...	70000
2	102	jayanth	jay@gmail.com	Male	8945672345	27-12-2001	09-11-2021	Admin	xyz...	90000
3	103	bhavya	bhavz@gmail.com	Female	7894625678	13-01-2022	09-11-2021	Employee	pqr...	90000

### II) Category

INSERT INTO CATEGORY VALUES (14,'television');

INSERT INTO CATEGORY VALUES (15,'mobile');

INSERT INTO CATEGORY VALUES (16,'laptop');

	cid	name
1	14	television
2	15	mobile
3	16	laptop

### III) Product

INSERT INTO PRODUCT VALUES (7,'vaishnavi','mobile','vivo',20000,50,'Active');

INSERT INTO PRODUCT VALUES (8,'gunashree','television','sony',70000,100,'Active');

INSERT INTO PRODUCT VALUES (9,'prachi','laptop','dell',60000,30,'Active');

	pid	supplier	category	name	price	qty	status
1	7	vaishnavi	mobile	vivo	20000	50	Active
2	8	gunashree	television	sony	70000	100	Active
3	9	prachi	laptop	dell	60000	30	Active



#### IV) Register

```
INSERT INTO REGISTER VALUES (101,'gaayana','Employee','gaayu',4112000);
INSERT INTO REGISTER VALUES (102,'jayanth','Admin','jay',27122001);
INSERT INTO REGISTER VALUES (103,'bhavya','Employee','bhavz',13012002);
```

	eid	name	utype	userid	upass
1	101	gaayana	Employee	gaayu	4112000
2	102	jayanth	Admin	jay	27122001
3	103	bhavya	Employee	bhavz	13012002

#### V) Supplier

```
INSERT INTO SUPPLIER VALUES (11,'gunashree',9867542356,'apbd...');
INSERT INTO SUPPLIER VALUES (12,'vaishnavi',9567836723,'bcthz...');
INSERT INTO SUPPLIER VALUES (13,'prachi',9345678912,'guxz...');
```

	invoice	name	contact	desc
1	11	gunashree	9867542356	apbd...
2	12	vaishnavi	9567836723	bcthz...
3	13	prachi	9345678912	guxz...

## 4.6 CREATION OF TRIGGERS

```
import sqlite3
def Trigger():
    try:
        sqliteConnection = sqlite3.connect('ims.db')
        cursor = sqliteConnection.cursor()
        print("Connected to SQLite")
        sql_trigger_query = """ CREATE TRIGGER insert_trigger AFTER INSERT ON employee
        BEGIN
            INSERT INTO register (eid,name,utype)
            VALUES (NEW.eid,NEW.name,NEW.utype);
        END;"""
        cursor.execute(sql_trigger_query)
        sqliteConnection.commit()
        cursor.close()
    except sqlite3.Error as error:
        print("Failed to delete record from sqlite table", error)
    finally:
        if sqliteConnection:
            sqliteConnection.close()
            print("the sqlite connection is closed")
Trigger()
```

insert\_trigger CREATE TRIGGER insert\_trigger AFTER INSERT ON employee BEGIN INSERT INTO register (eid,name,utype) VALUES (NEW.eid,NEW.name,NEW.utype); END

Table 4.6 Trigger Insert Table

## 4.7 CREATION OF STORED PROCEDURE

As SQLITE does not support the stored procedure feature of MySQL or other SQL tools, the alternative for the same are user-defined functions for add and update which use classes and objects and are called often.

```
def add(self):
    con=sqlite3.connect(database='ims.db')
    cur=con.cursor()
    try:
        if self.var_emp_id.get()=="":
            messagebox.showerror("Error","Employee ID must be entered",parent=self.root)
        else:
            cur.execute("Select * from employee where eid=?", (self.var_emp_id.get(),))
            row=cur.fetchone()
            if row!=None:
                messagebox.showerror("Error","This Employee ID is already assigned, try different",parent=self.root)
            else:
                cur.execute("Insert into employee(eid,name,email,gender,contact,dob,doj,utype,address,salary) values(?,?,?,?,?,?,?,?,?,?)", (
                    self.var_emp_id.get(),
                    self.var_name.get(),
                    self.var_email.get(),
                    self.var_gender.get(),
                    self.var_contact.get(),
                    self.var_dob.get(),
                    self.var_doj.get(),
                    self.var_utype.get(),
                    self.txt_address.get('1.0',END),
                    self.var_salary.get()
                ))
                con.commit()
                messagebox.showinfo("Success","Employee Details Added Sucessfully",parent=self.root)
                self.show()
    except Exception as ex:
        messagebox.showerror("Error",f"Error due to: {str(ex)}",parent=self.root)
```

## CHAPTER 5

### FRONT END DESIGN

#### 5.1 CONNECTIVITY TO DATABASE

Graphical user interfaces would become the standard of user-centered design in software application programming, providing users the capability to intuitively operate computers and other electronic devices through the direct manipulation of graphical icons such as buttons, scroll bars, windows, tabs, menus, cursors, and the mouse pointing device. Many modern graphical user interfaces feature touchscreen and voice- command interaction capabilities.

#### PYTHON: USING MODULES TO CONNECT DATABASES

- Python can access other applications by calling in their respective modules, one of the applications that can be used as database is DB Browser or SQLITE.
- Another advantage of using such language is to create an user based application that can be run on any system as a fully developed software.
- Support for using the python standard GUI toolkit Tkinter and SQLITE are described at:-

<https://docs.python.org/3/library/tkinter.html>

<https://www.sqlite.org/docs.html>

#### PROCESS OF CREATING DATABASE CONNECTION USING SQLITE FROM PYTHON

- Create a file in any source code editor with python installed,
- Import the sqlite3 module,
- Write a code to create the respective database,
- In the main file using the GUI, import the module and connect the database,
- This creates connection between the application and SQLITE database.

## CREATING AND ACCESSING A DATABASE

- Importing the SQLite module, using *import sqlite3*,
- Defining a *create\_db* function to call and create tables,
- *sqlite3.connect(database=r'database\_name.db')* is stored into a variable to connect the DB Browser to the application,
- Using the *con.cursor()* method to use the database created,
- *cur.execute("CREATE TABLE IF NOT EXISTS .... ")* method is used to create the tables in SQLite,
- Finally, to commit the table or to make the creation of table we use *cur. commit ()* method.
- This will successfully create a database with specified entities and attributes.
- Then we call the filename to create the file successfully.

## 5.2 FRONT END CODE

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development" blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

SQLite is often faster than a client/server SQL database engine in this scenario. Database requests are serialized by the server, so concurrency is not an issue. Concurrency is also improved by "database sharing": using separate database files for different subdomains. For example, the server might have a separate SQLite database for each user, so that the server can handle hundreds or thousands of simultaneous connections, but each SQLite database is only used by one connection.

# **CHAPTER TESTING**

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## **6.1 TESTING PROCESS**

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

## **6.2 TESTING OBJECTIVES**

The main objectives of testing process are as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

## 6.3 TEST CASES

The test cases provided here test the most important features of the project.

**Table 6.3 Test Cases**

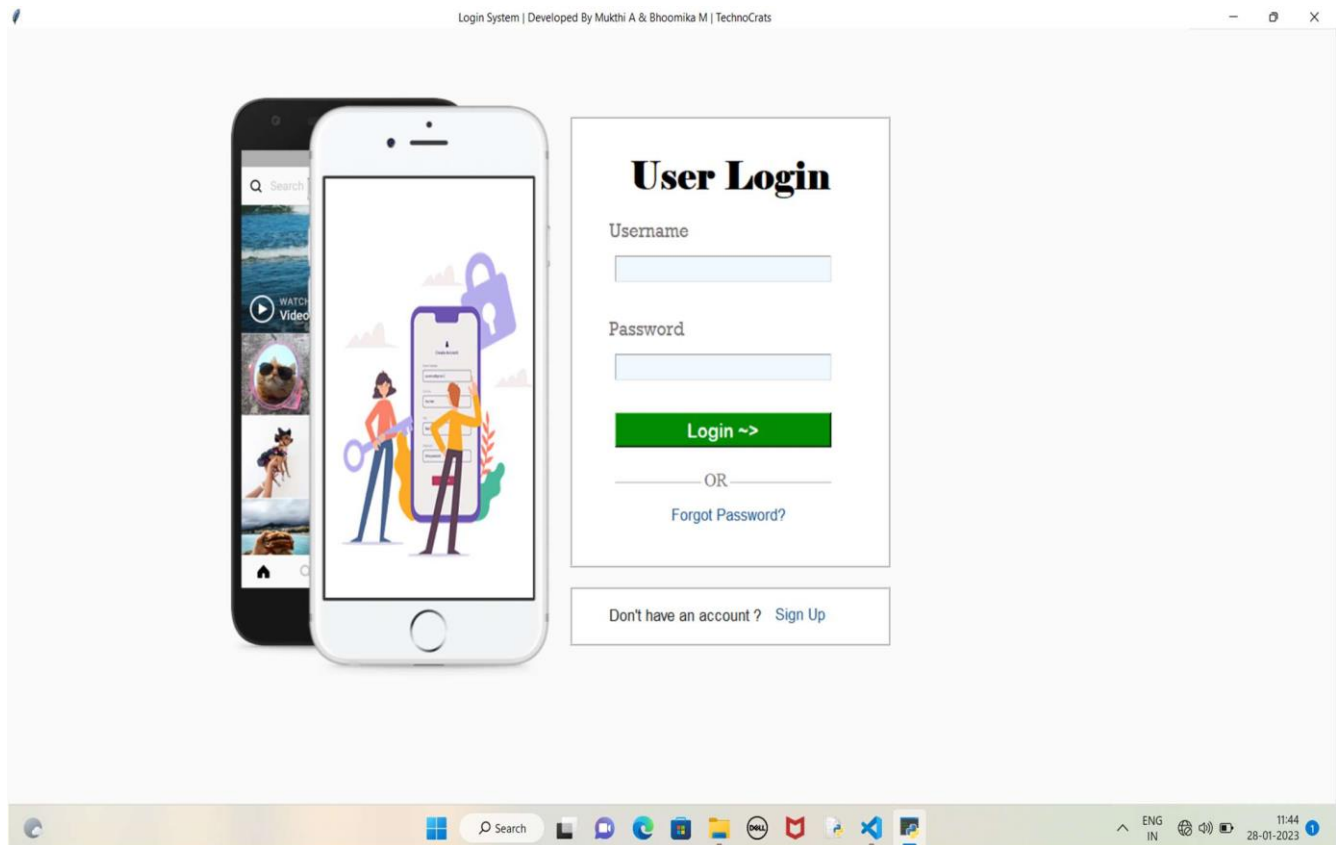
SL No	Test Input	Expected Results	Observed Results	Remarks
1	Insert a Record	New tuple Should be inserted	Query OK 1 row affected or inserted	PASS
2	Search a Record	Search from existing Records	Query OK 1 row affected or searched	PASS
3	Delete a record	Delete a Record	Query OK 1 row affected or deleted	PASS
4	Create Trigger	Trigger <del>actd</del>	Query OK Trigger created	PASS
5	Create Stored Procedure	Stored Procedure Created	Query OK Stored Procedure created	PASS

## CHAPTER 7

### RESULT

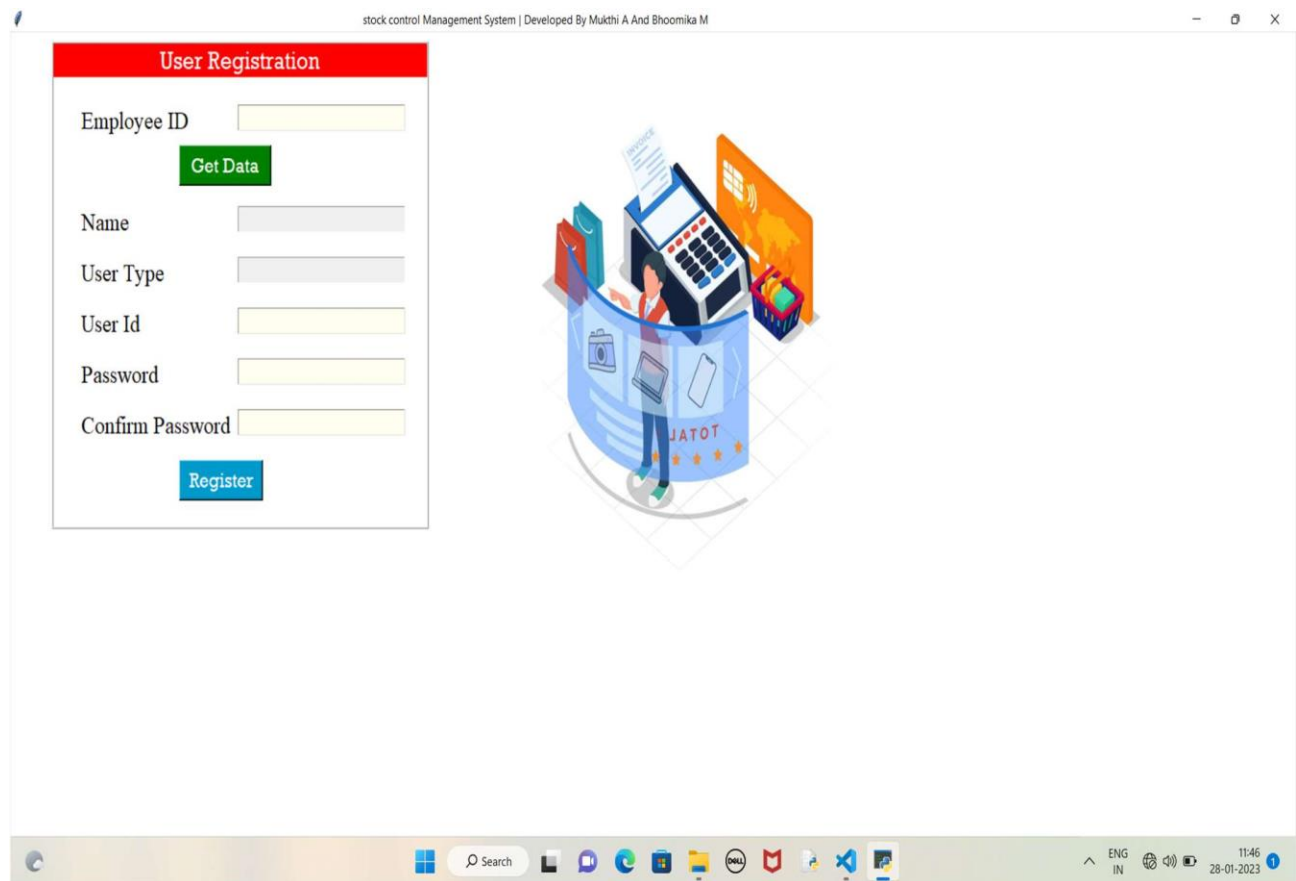
This section describes the screens of the “Stock Control Management System”. The snapshots are shown below for each module.

#### 7.1 SNAPSHOTS



**Fig 7.1 Snapshot of User Login**

This is the Login Interface of the application where the user has to input user ID and password to continue.



**Fig 7.2 Snapshot of User Registration**

The Registration Interface allows the user to get any employee details if the employee is not registered then they will be registered giving a new user ID and password.





**Fig 7.3 Snapshot of Admin Dashboard**

This is the Admin Dashboard where the admin can view all the data from employee details to the sales and stocks of the business.

stock control Management System | Developed By Mukthi A And Bhoomika M

Search Employee

Select  Search

### Employee Details

Emp ID  Gender  Contact

Name  D.O.B  D.O.J

Email  Salary  User-Type

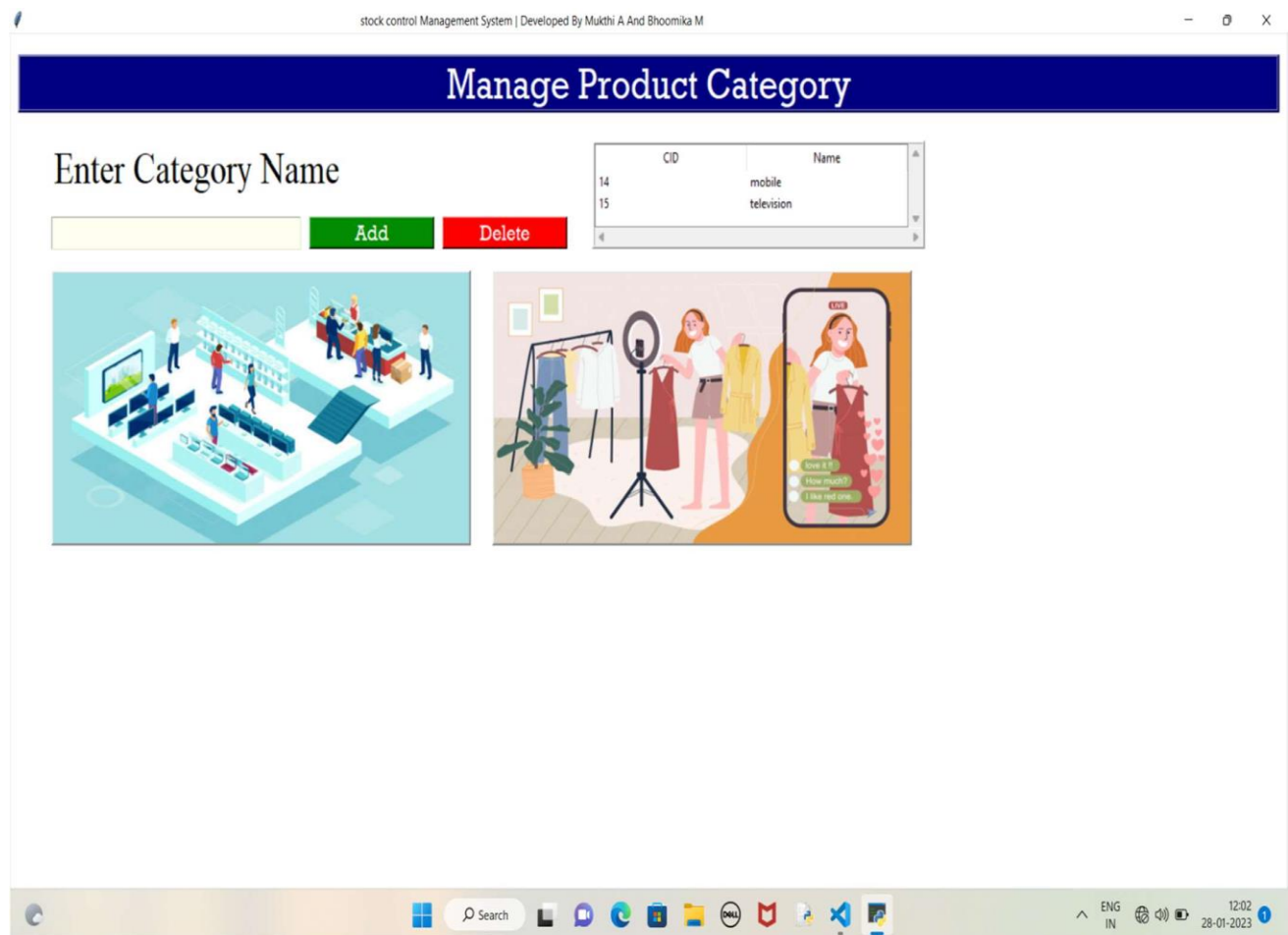
Address

Save Update Delete Clear

Emp ID	Name	Email	Gender	Contact	D.O.B	D.O.J	User-Type	Address	Salary
101	gaayana	gaayu@gmail.com	Female	9481041406	04-11-2000	09-11-2021	Employee	abc	70000
102	mukthi	mukthi@gmail.com	Female	9476289334	26-02-2002	02-11-2022	Admin	xyz	70000

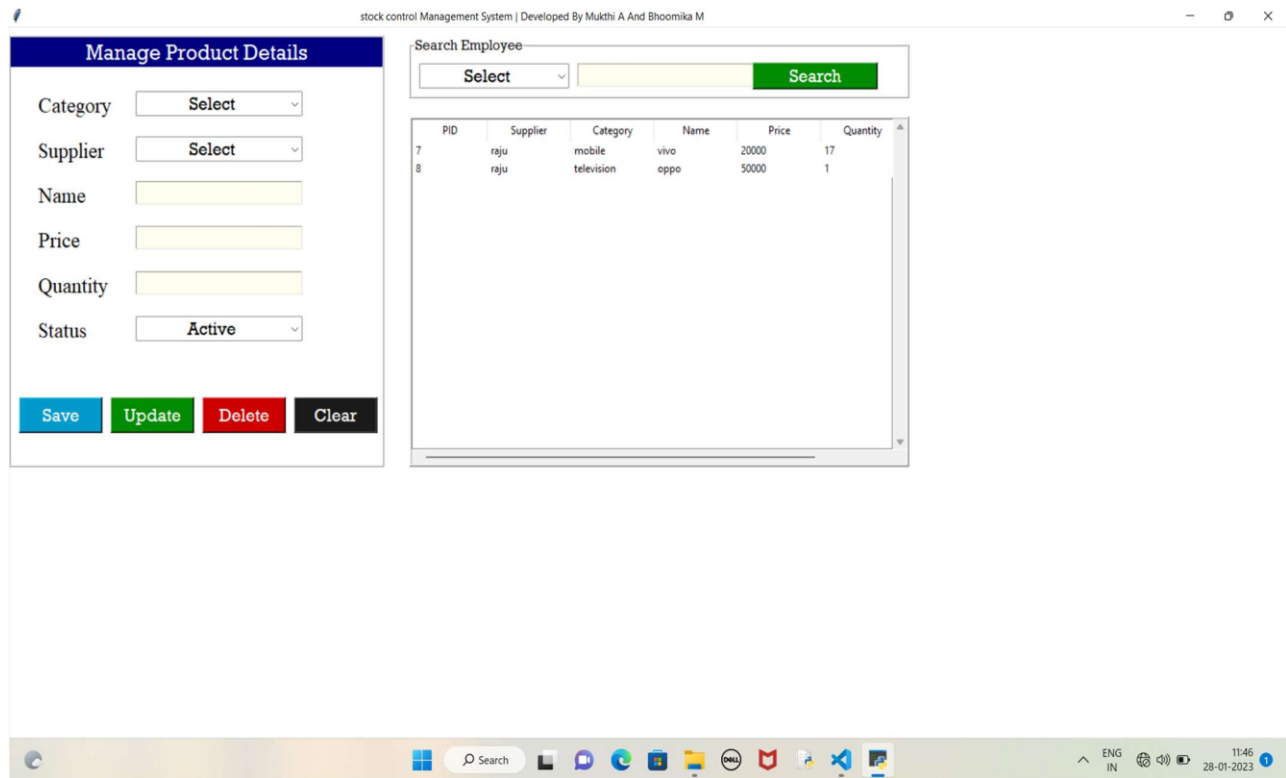
**Fig 7.4 Snapshot of Employee Details**

In this interface Employee details is saved, if there are any changes to be updated or when the Employee no longer exist the details can be deleted. Also, an employee can be quickly searched using either Name or Email or by Contact.



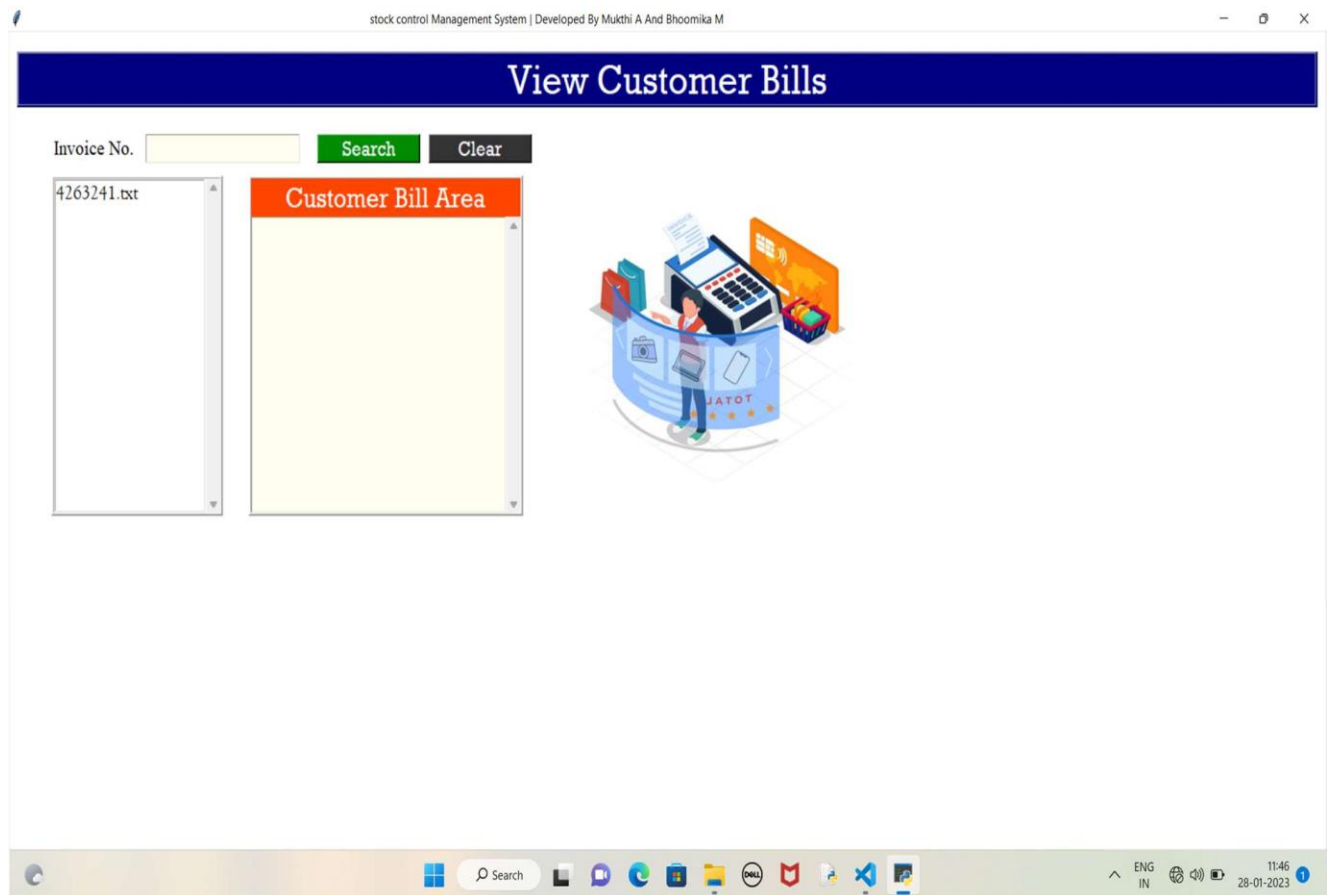
**Fig 7.5 Snapshot of Product Section**

The category section, here any product category is added or deleted from the database of the stock.



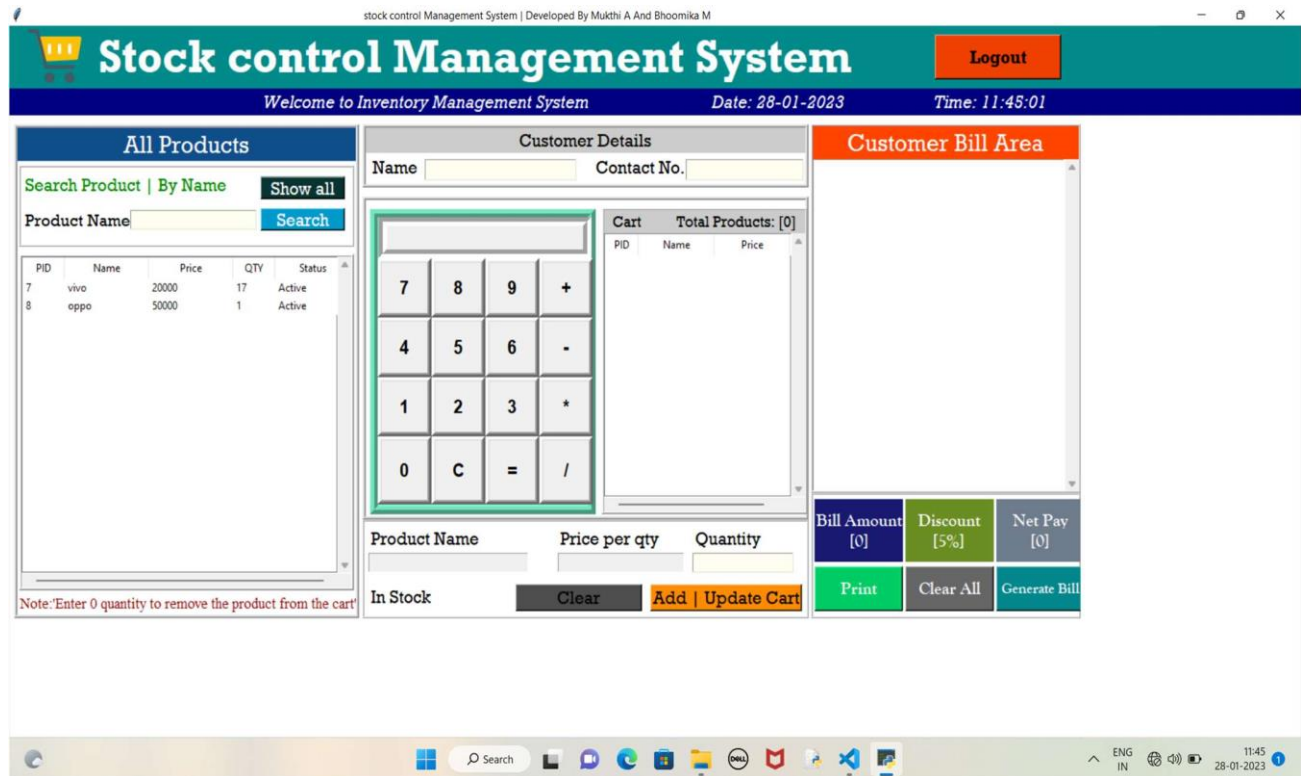
**Fig 7.6 Snapshot of Category Interface**

In the Product interface, the product details such as Category, Supplier, Name, Price, and Quantity are saved, updated or deleted. The products can also be searched using name and status of the stocks.



**Fig 7.7 Snapshot of Sales and Stocks**

In this section of Sales, the invoices of all the transitioned and sold-out products are viewed, and also specific bills can be accessed through their specific invoice nos. that are generated during the billing process.



**Fig 7.8 Snapshot of Billing System**

This interface is the Billing Interface where an employee doing the billing process can generate a customer bill, print it, view the active products from the inventory, an effective calculator to perform quick operations. A product can be added or updated corresponding to the customer requirement with ease.

## CHAPTER 8

### CONCLUSION

The Stock Control management system provides easier maintenance of various product's details and stocks. It allows simplified operation and is a time saving platform. The building blocks of this Major Project “STOCK CONTROL MANAGEMENT SYSTEM” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project,
- The importance of proper planning and an organized methodology,
- The key element of team spirit and co-ordination in a successful project.

The project also provides us the opportunity of interacting with our teachers and to gain from their best experience. The application has been completed successfully and tested with suitable test cases. It is user friendly. This is developed using Python with Tkinter and SQLite.

## BIBLIOGRAPHY

### REFERENCES

1. <https://www.geeksforgeeks.org/python-classes-and-objects/>
2. <https://docs.python.org/>
3. <https://www.youtube.com/playlist?list=PL4P8sY6zvjk76iWbcUQ2jDntGhmX3PLF4>
4. <https://www.geeksforgeeks.org/python-classes-and-objects/>
5. [https://www.w3schools.com/python/python\\_intro.asp](https://www.w3schools.com/python/python_intro.asp)
6. <https://www.youtube.com/watch?v=pd-0G0MigUA>
7. [https://www.tutorialspoint.com/sqlite/sqlite\\_python.html](https://www.tutorialspoint.com/sqlite/sqlite_python.html)