

**Name: GONDRALA SAI SHARANYA**

**USN: 1NH18IS035**

**Title: BLOOD BANK DATABASE USING GUI**

## CHAPTER 1

### INTRODUCTION

Blood bank is a bank of blood or gathering of blood components as a result of donation which is preserved for future use. All aspects of blood bank are managed by the software which is called as blood bank database management system.

### MOTIVATION FOR THE PROJECT

It has been developed to override the problems prevailing in practicing the manual system. No formal knowledge is needed for the user to use this system. Thus it proves to be user friendly. It is error free, secure and fast management system.

Every organization has to maintain some amount of data so this database can be used by each and every organization for better utilization of resources.

DBMS essentially serves as an interface between database and end users or application programs, ensuring that data is properly organized and easily accessible.

This system enables administrators to create, insert, store, delete, view and perform DML and DDL commands on the records of blood bank database.

This database contains the details of blood bank which has donors, recipients, blood bank employees, blood product and also has connections with hospitals so that in case if an hospital needs blood in an emergency situation, it can contact to particular blood bank.

### PROPOSED SYSTEM

- Python programming language is used here as front end which is the most portable language used in GUI applications.
- We use Database management system as back end to store data using Sqlite3 which is a inbuilt module in python.
- We are using Graphical User Interface using a module of python called Tkinter.
- We can perform DDL and DML commands on the database tables.

**OBJECTIVES**

- The important objective is to create a system which helps them to complete their work in a simple way using computer which out the use of paper.
- It helps us to know the different components that exist in a blood bank.
- It is designed to store, process, retrieve and analyze information concerned with the administrative and account management within a blood bank.
- Searching becomes easier.
- Easily and efficiently stored and accessible.

**ADVANTAGES OF DATABASE**

- Today the database is used everywhere and all walks of life. The database is becoming the backbone of all the softwares from standalone, client-server, mainframe, supercomputers etc.
- The results of database are presented in form of reports or graphical representation, which are simple and easy to understand.
- Shows the whole description of business.
- Manage the information of all entities in a systematic manner.
- Addition and deletion of records are simpler and easier to perform on the database.
- Provides searching facilities based on various factors such as donors, blood products, recipients etc.

## CHAPTER 2

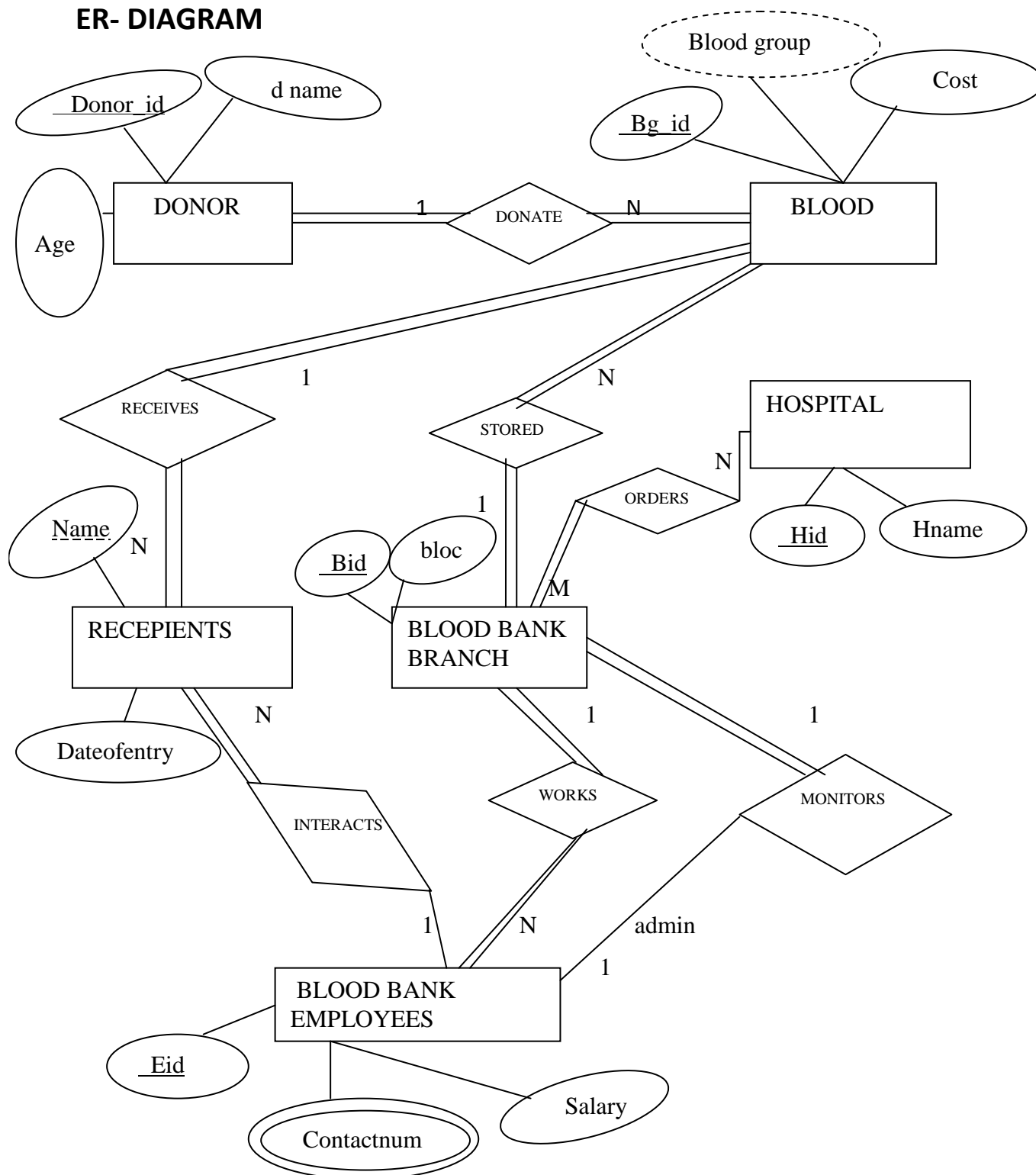
### ANALYSIS AND DESIGNING

#### HARDWARE REQUIREMENTS

- Any processor can be used which is above 500 MHz
- RAM is of 4GB
- Input Devices used are Standard keyboard and Mouse
- Output Devices used are VGA and High Resolution Monitor

#### SOFTWARE REQUIREMENTS

- Python Programming Language(GUI) is used as front-end and SQL as back-end in this project
- Python interpreter is used here
- db browser for SQLite software is required for storing data in database tables
- Operating system is Windows 10
- GUI is implemented using tkinter module of python

**ER- DIAGRAM**

## CHAPTER 3

### IMPLEMENTATION

#### INTRODUCTION TO DBMS

Data can be defined as the small volumes of facts and figures. Information can be defined as the collection of data. Database can be defined as the large volumes of facts and figures and that logically related data organized in a way easy to accessed, managed and updated.

**Entity**:-Table in database which is a collection of data organized in form of rows and columns.

**Records**:-A single entry in database is called as record.

**Tuple**:-Rows within the tables are also called as tuples.

**Attributes**:-Columns in the tables are also called as attributes or fields.

There are different types: Atomic, composite, single-valued, multi-valued, stored, derived attributes.

**Domain**:-Set of values given for an attribute in a column is called as domain.

Database provides the subsequent features:

1. Allows multiple user interface
2. Controlled data redundancy
3. Provides backups and recovery
4. Easy to develop applications
5. Provides data integrity
6. Data security by enforcing security constraints

The most traditional database management system was flat file model within which all the details of database was embedded in a single large file with all the disadvantages like redundancy, lack of concurrency, less efficient in the use of space for storing data etc.

To overcome these disadvantages, Relational database model was developed in which we are able to create separate tables for different entities with all the attributes in form of rows and columns. This makes querying much easier in this database. It is much user friendly and programmer friendly and dominant. Oracles, Sybase, MS-SQL server, informix etc are the popular relational database management systems.

## RELATIONSHIP

Relationship describes the relations existing between the entities. There are 3 types of relationships:

- Binary relationship
- Recursive relationship
- Ternary relationship

In binary relationship, two entities participate in the relationship.

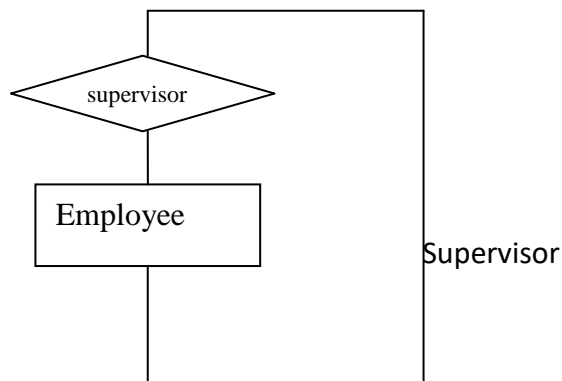


The binary relationship in blood bank database can be explained by the above example.

The donor is the one who donates blood to the blood bank so there exists a binary relationship which is nothing but 2 different entities are participating in the relationship.

In recursive relationship, entity is related with itself in the relationship.

Example:



The recursive relationship in this database is explained by the above example.

Supervisor supervises all the employees in the blood bank, in turn a supervisor is also an employee.

## STRUCTURAL CONSTRAINTS

Cardinality ratio: It is the maximum number of relationship instances of an entity that may participate in. There are 4 types:

1:1 - one-to-one ratio

1:N - one-to-many ratio

N:1 - many-to-one ratio

M:N - many-to-many ratio

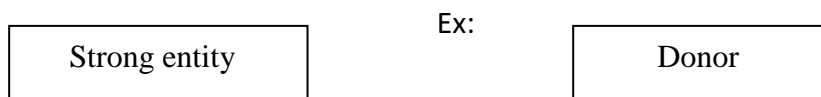
Participation constraint: It is the minimum number of relationship instances of an entity that may participate in. There are 2 types:

1. Total participation: All entities are involved in a relationship. Represented by double line.
2. Partial participation: Not all entities are involved in a relationship. Represented by single line.

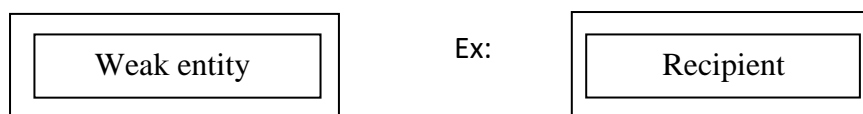
## ER-DIAGRAM

This diagram describes the relationship between the entities. It is the visual representation of data.

STRONG ENTITY: Entity which has a key attribute is called a strong entity and it is represented by single lined rectangle. Key attribute is a unique and not null attribute.



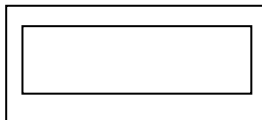
WEAK ENTITY: Entity which has a partial key attribute or do not have any attribute to make as key attribute is called weak entity and it is represented by double lined rectangle. Partial key attribute is not unique.



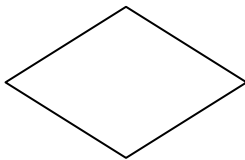


**REPRESENTATIONS FOR ER-DIAGRAMS:**SymbolMeaning

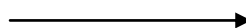
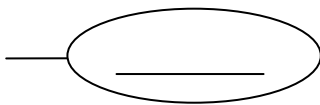
Strong entity



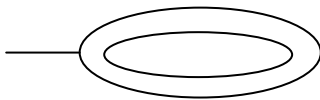
Weak entity



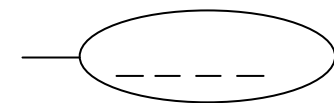
Relationship



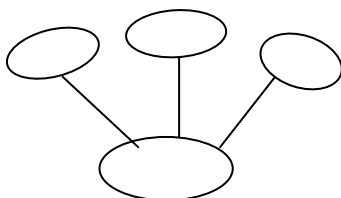
Key attribute



Multi-valued attribute



Partial key attribute



Composite attribute



Derived attribute

## keys

keys are used to represent or identify the relation between the database tables. It is a set of one or more columns whose values are unique when combined among the occurrences in a table.

There are different keys: Candidate keys, primary keys, alternate keys, super keys, foreign keys etc.

**Candidate key:** This key uniquely identifies any database record without referring to any other data.

**Primary key:** It has a unique and not null value and it is the one among candidate keys.

**Foreign key:** This key is a primary key of another table in that particular database.

**Alternate key:** Candidate keys except primary key is called as an alternate key.

**Super key:** Set of columns which uniquely identifies rows can be defined as super key.

## QUERYING

Action of extracting the data from the database can be defined as Querying.

## BENEFITS OF QUERYING

Queries help to save time and money for business. Queries are questions that you ask the database that to help you find specific information. This gives you the opportunity to request materials from the database that contains multiple tables. There are numerous benefits in database that come because of a business using query in their daily tasks. They are comprised of multiple data views, using interactive languages to make it easier to learn and understand.

A query helps us to make fewer mistakes when performing on databases. If the information is correct it helps minimize the possibility of an error occurring. If the queries bring up the information which is not correct, you can fix them. It acts as a time saver in order to fetch a specific item easily and within a short period of time.

## INTRODUCTION TO SQL

SQL is Structured Query Language, which is a dbase language includes creation, fetching rows, modifying rows and deletion. It helps us to store, manipulate and retrieve the data In relational database model.

### FEATURES OF SQL

- Allows users to create database tables and define data, insert values into the rows.
- Allows users to define data, operate on the data.
- Allows users to access or retrieve the data in database management system.
- Allows embedding with other languages using SQL modules and libraries.
- Allows delete and drop database tables.
- Allows us to create views, stored procedure, functions in a database and many more operations can be performed on the database tables.

There are two types of SQL commands:

1. DDL: Data Definition Language provides link between logical and physical structures of the database. It defines all physical characteristics of all records, provides data security measures, describes schema and subschema, indicates keys of the records etc.

Basic commands are: CREATE, ALTER, DROP

2. DML: Data Manipulation Language provides all kinds of data manipulation techniques like selection, insertion, deletion, modification, updation, retrieval, sorting and displaying data or rows. It provides for sovereignty of high level programming languages, facilitates the relationship between the records.

Basic commands are: INSERT, DELETE, UPDATE, GRANT, REVOKE, SELECT.

SELECT is the command used for horizontal fragmentation of data.

### SQL CONSTRAINTS:

These are the rules or conditions enforced on the data columns of the table.

5 constraints are: Foreign key, primary key, unique, not null, check.

**TKINTER AND SQLITE3 USING PYTHON**

- Python is a portable language which can run on any platform.
- It is a highly productive and easier language compared to other languages.
- Python is used for developing various graphical user interfaces, web sites, web applications.
- It allows us to focus on the internal functionality of application.
- It is the most efficient language with simplest syntax.
- Python provides two inbuilt modules sqlite3 and tkinter.
- Tkinter helps us to create graphical user interface and sqlite3 is used to create efficient database tables.

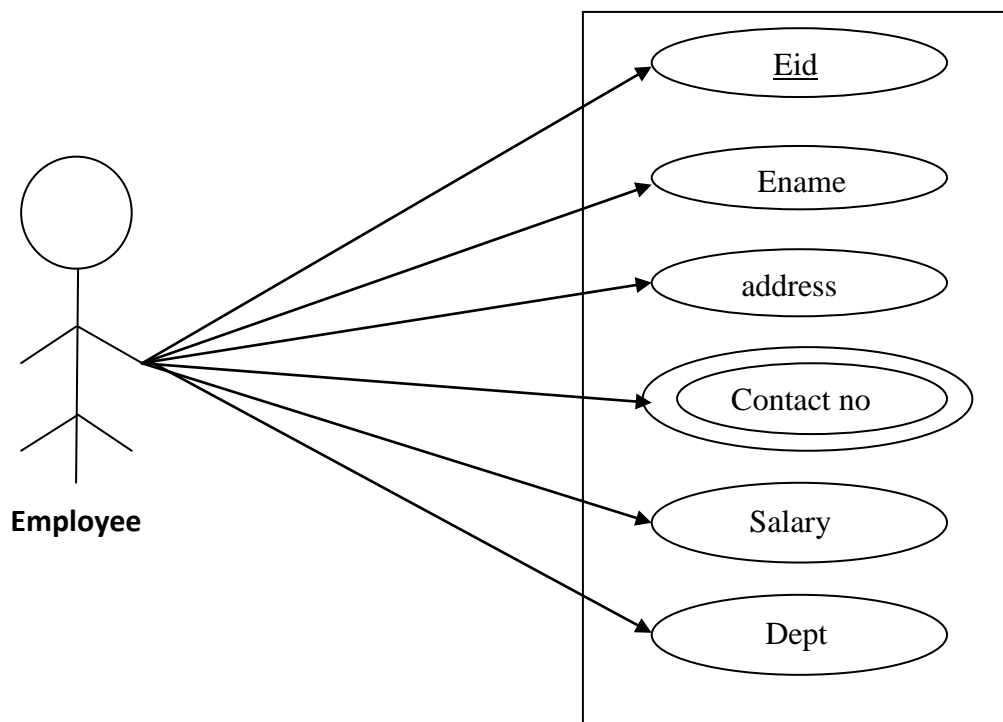
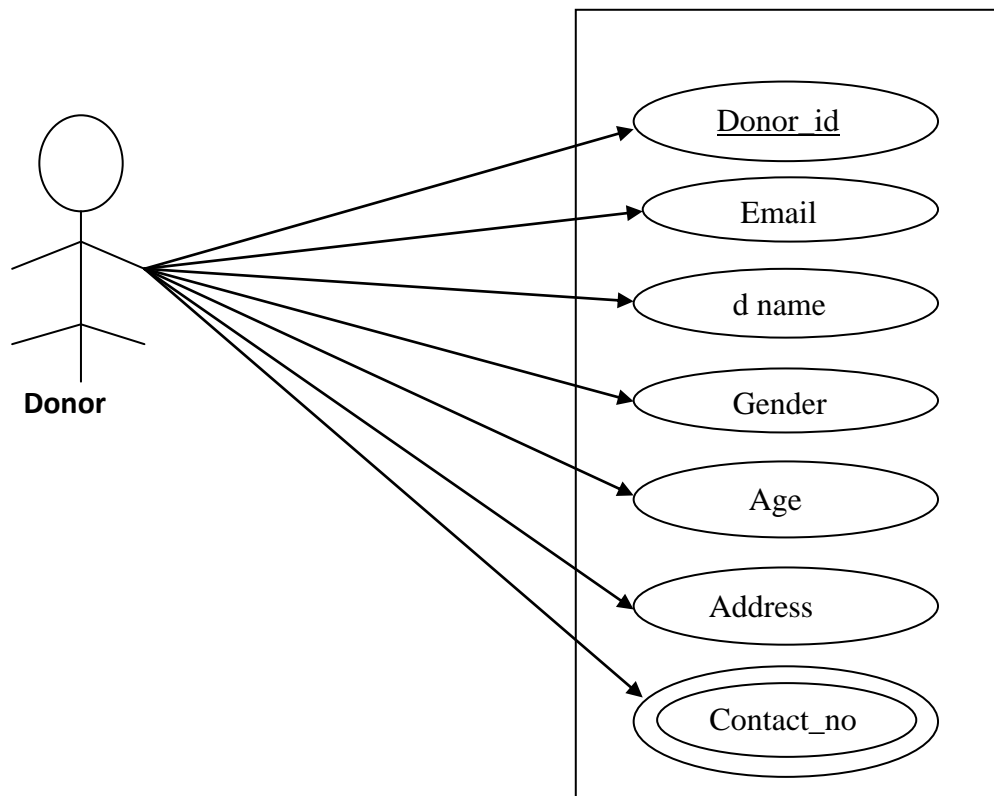
Syntax for importing the modules:

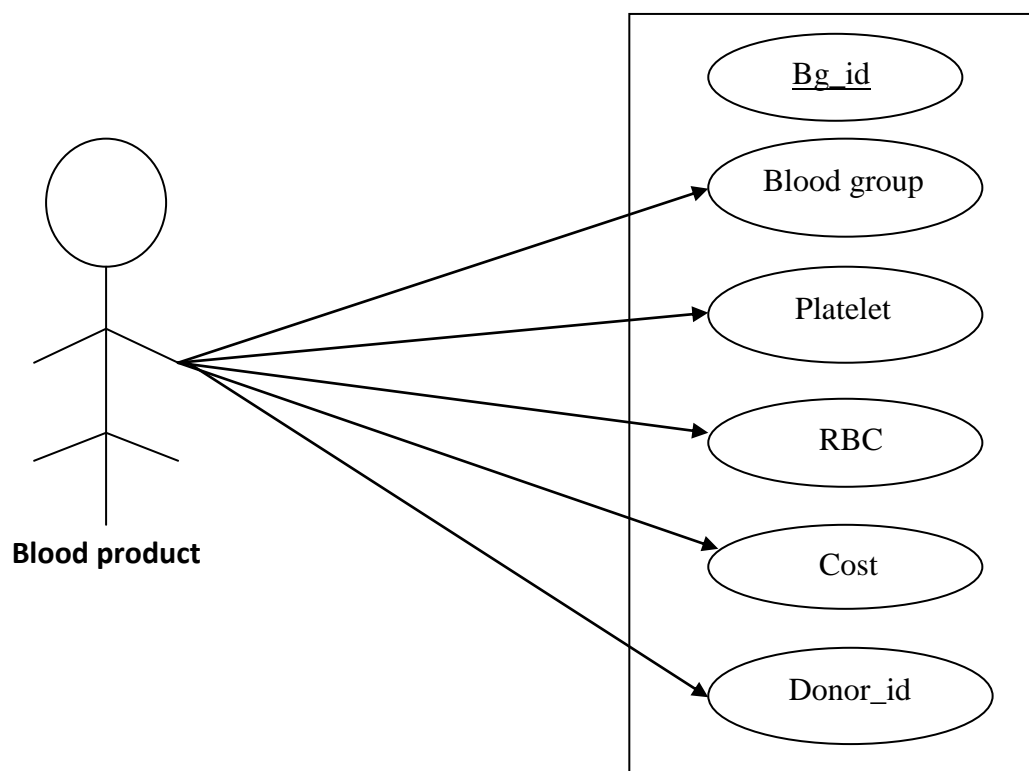
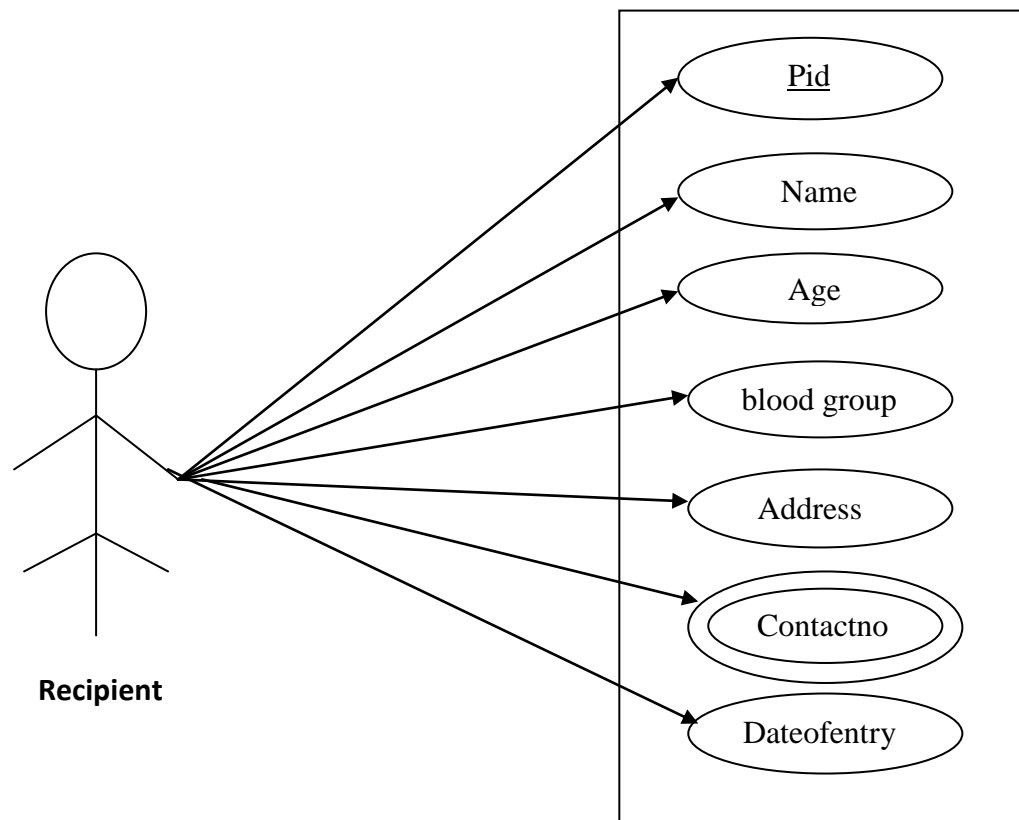
```
from tkinter import *  
import sqlite3
```

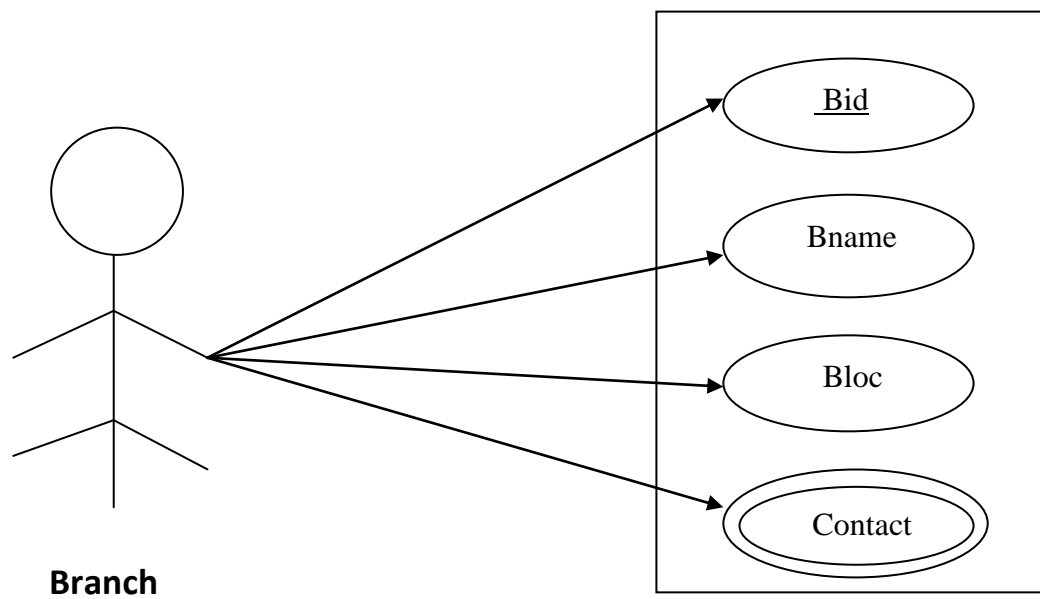
Tkinter module is the standard GUI library for python. It provides different features to change the font size, font style, set width and height, change background colour, create buttons with different layout using relief and change the border width of it, create text box using Label for user entry, Entry widget is used to receive a single line input from the user and many more widgets can be used.

It mainly provides two methods: Tk() is a method used to create a separate window and mainloop() is a method which is used to run the application and the process keeps running until the window is not closed.

Sqlite3 provides different functionalities to create and manipulate database tables. Insertion, deletion, modification, updation, grant, revoke, alter, search and many other operations are provided to perform on the data of database tables.

**ATTRIBUTES STORED IN DATABASE**





## CHAPTER 5

### CODE SNIPPETS

The tables created in the database are as follows:

#### DONOR

```
def insertDonor(d_id,name1,email,gender,age,contact,address):
```

```
    cursor.execute('CREATE TABLE IF NOT EXISTS donor(Donor_id INT primary key, Fullname TEXT, Email TEXT, Gender TEXT, Age INT, Contact INT, Address TEXT)')
```

```
    cursor.execute('INSERT INTO donor(Donor_id, Fullname, Email, Gender, Age, Contact, Address)VALUES(?,?,?,?,?,?,?)',(d_id,name1,email,gender,age,contact,address,))
```

```
    conn.commit()
```

	Donor_id	Fullname	Email	Gender	Age	Contact	Address
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	301	sai	shar645@gm...	male	30	9532187991	Whitefield
2	302	Chandana	chandu40@g...	female	41	9872454791	Kadubisanahalli
3	303	Anup	anugowda23...	male	26	9900453791	Itpl
4	304	Varun	varunss20@g...	male	35	8720499101	Jayadeva
5	305	Gagana	Ganga67@gm...	female	52	8562749224	Somanahalli

In the above table, Donour\_id, Fullname, Email, Gender, Age, Contact, Address are the attributes of the entity donor. Donor\_id acts as primary key and Email is a candidate key. Contact is a multivalued attribute and rest all attributes are atomic.

#### BLOOD\_PRODUCT

```
def insertBlood(d_id,bg_id,bloodgroup,platelet,rbc,cost):
```

```
    cursor.execute('CREATE TABLE IF NOT EXISTS blood_product(Bg_id PRIMARY KEY, Donor_id, Bloodgroup TEXT, Platelet INT, RBC INT, Cost INT)')
```

```
    cursor.execute('INSERT INTO blood_product (Bg_id,Donor_id,Bloodgroup,Platelet,RBC, Cost) VALUES(?,?,?,?,?,?)',(d_id, bg_id, bloodgroup, platelet, rbc, cost,))
```

```
    conn.commit()
```



	Bg_id	Donor_id	Bloodgroup	Platelet	RBC	Cost
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	301	A+	123	4.2	1050
2	2	303	O-	267	3.2	600
3	3	302	AB+	245	3.1	750
4	4	301	A+	123	4.2	1050
5	5	304	B+	200	2.2	700
6	6	305	O+	128	2.9	960

In this table, Bg\_id, Donor\_id, Bloodgroup, platelet, RBC, cost are the attributes to the entity blood\_product in this database. Bg\_id is the primary key, Donr\_id is a foreign key which acts as a primary key to the table donor.

#### BRANCH

```
def branch(bid,bname,bloc,contact):
```

```
    cursor.execute('CREATE TABLE IF NOT EXISTS branch(Bid INT PRIMARY KEY,B_name
    TEXT,B_location TEXT,Contact_info INT)')
```

```
    cursor.execute('INSERT INTO branch(Bid, B_name, B_location, Contact_info)VALUES
    (?, ?, ?, ?)',(bid, bname, bloc, contact,))
```

```
    conn.commit()
```

	Bid	B_name	B_location	Contact_info
	Filter	Filter	Filter	Filter
1	1001	chanakya bb	jayadeva	8087654321
2	1501	Grace bb	sanjaynagar	9825658200
3	1354	Oceans bb	shivajinagar	7827536810
4	1295	Jeevaraksha bb	cunningham r...	9876543223
5	1753	Lions bb	Attiguppe	9876543648
6	1900	Swamy viveka...	Indiranagar	8035674388
7	1813	Bangalore bb	jayanagar	9828117610

In this table, we have Bid, B\_name, B\_location, Contact\_info as attributes to the entity branch. Contact\_info is a multivalued attribute. Bid is the primary key.

### RECIPIENTS

```
def recipients(pid,bg_id,name,age,blg,amtofblood,contct,date):
```

```
    cursor.execute('CREATE TABLE IF NOT EXISTS recipients(Pid INT PRIMARY KEY, Bg_id INT, Name TEXT, Age INT, Bldgrp TEXT, Amtofblood TEXT, Contactno INT, Dateofentry DATE)')
```

```
    cursor.execute('INSERT INTO recipients(Pid, Bg_id, Name, Age, Bldgrp, Amtofblood, Contactno, Dateofentry)VALUES(?,?,?,?,?,?,?)',(pid, bg_id, name, age, blg, amtofblood, contct, date,))
```

```
    conn.commit()
```

	Pid	Bg_id	Name	Age	Bldgrp	Amtofblood	Contactno	Dateofentry
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	501	5	anup	40	B+	500ml	9834678530	11.03.2019
2	502	4	Vindhya	40	A+	500ml	7643268392	12.03.2019
3	503	3	John	25	AB+	1000ml	9988246709	10.04.2019
4	504	4	Anjana	25	A+	1000ml	9862357932	17.12.2019

In this table, Pid, Bg\_id, Name, Age, Bloodgroup, Amtofblood, Contactno, Dateofentry are the attributes to the entity recipients. Contactno is a multivalued attribute, Bldgrp is the derived attribute. Pid is the primary key and Bg\_id the foreign key which acts as a primary key to the table blood\_product.

EMPLOYEE

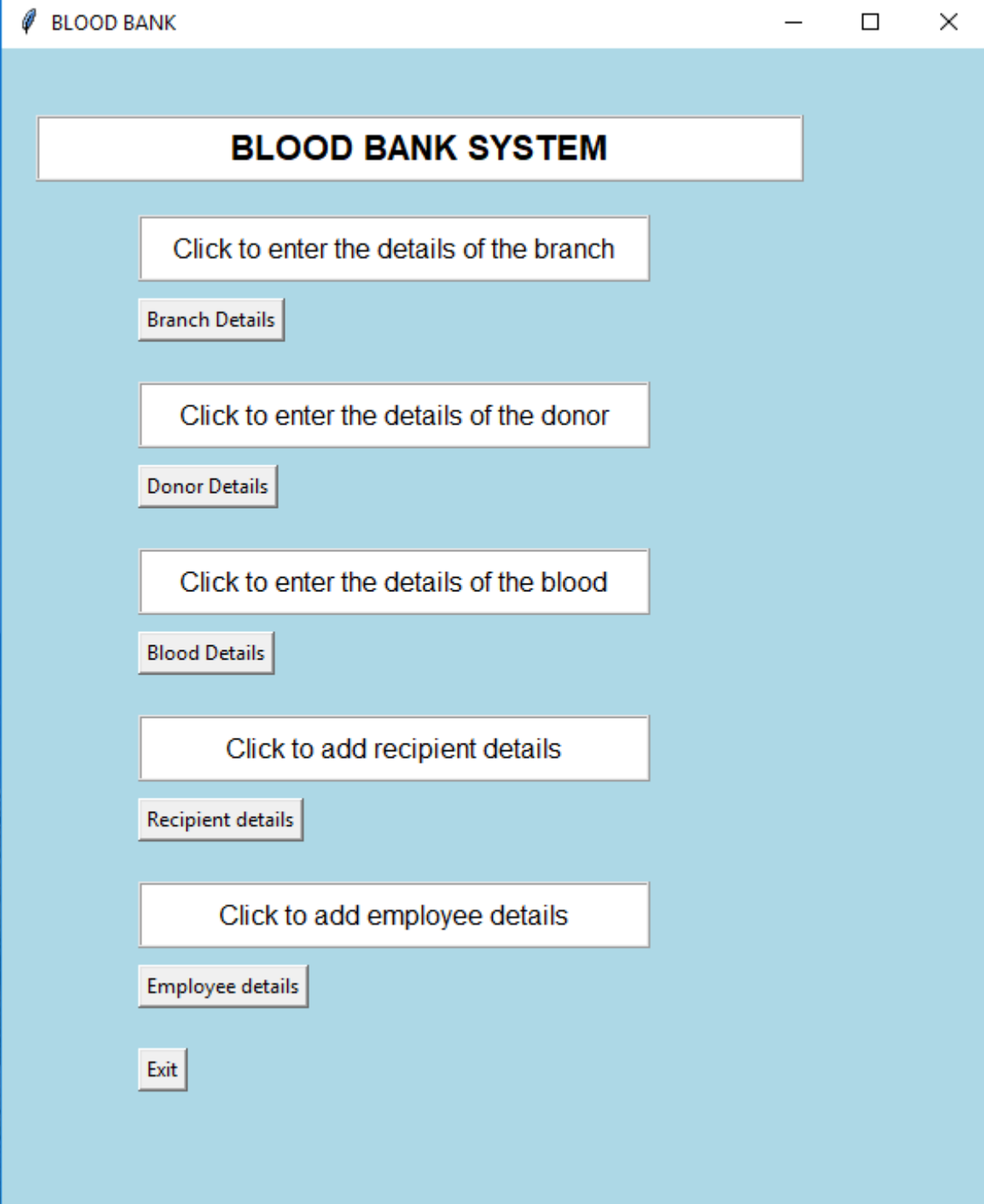
def employee(eid,bid,ename,adrs,contactdetails,salary,designation):

```
    cursor.execute('CREATE TABLE IF NOT EXISTS employee(Eid INT primary key,Bid,E_name
    TEXT,address TEXT,Contact_details INT,Salary INT,Designation TEXT)')
```

```
    cursor.execute('INSERT INTO employee(Eid, Bid, E_name, address, Contact_details,Salary,
    Designation)VALUES(?,?,?,?,?,?,?)',(eid,bid,ename,designation,adrs,contactdetails,salary,))
    conn.commit()
```

	Eid	Bid	E_name	address	Contact_details	Salary	Designation
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	800	1001	Anusha	Bellandur	9094328101	60700	Snr manager
2	801	1001	Anjali	KR puram	9929746189	54700	Supervisor
3	802	1501	Balu	Bellandur	9098642084	40700	Technician
4	803	1753	Chandana	Marathalli	900474327	40000	Lab Technician
5	804	1001	Deeraj	Marathalli	8653681458	14200	Technician

In this table, Eid, E\_name, Bid, Address, Contact\_details, Salary, Designation are the attributes to the entity Employee. Eid is the primary key and Bid is the foreign key which is a primary key to the branch relation(table). Contact\_details is a multivalued attribute. Remaining attributes are single valued in this relation.

**CHAPTER 4****EXPECTED OUTCOMES**

The screenshot displays a graphical user interface for a 'BLOOD BANK SYSTEM'. The window has a title bar with the text 'BLOOD BANK' and standard minimize, maximize, and close buttons. The main area has a light blue background. At the top, a white rectangular box contains the title 'BLOOD BANK SYSTEM' in bold black text. Below this, there are five main menu items, each consisting of a white rectangular button with black text and a smaller, light gray rectangular button directly beneath it. The menu items are: 'Click to enter the details of the branch' with a 'Branch Details' button; 'Click to enter the details of the donor' with a 'Donor Details' button; 'Click to enter the details of the blood' with a 'Blood Details' button; 'Click to add recipient details' with a 'Recipient details' button; and 'Click to add employee details' with an 'Employee details' button. At the bottom left of the main area, there is a single light gray button labeled 'Exit'.

**BLOOD BANK SYSTEM**

Click to enter the details of the branch

Branch Details

Click to enter the details of the donor

Donor Details

Click to enter the details of the blood

Blood Details

Click to add recipient details

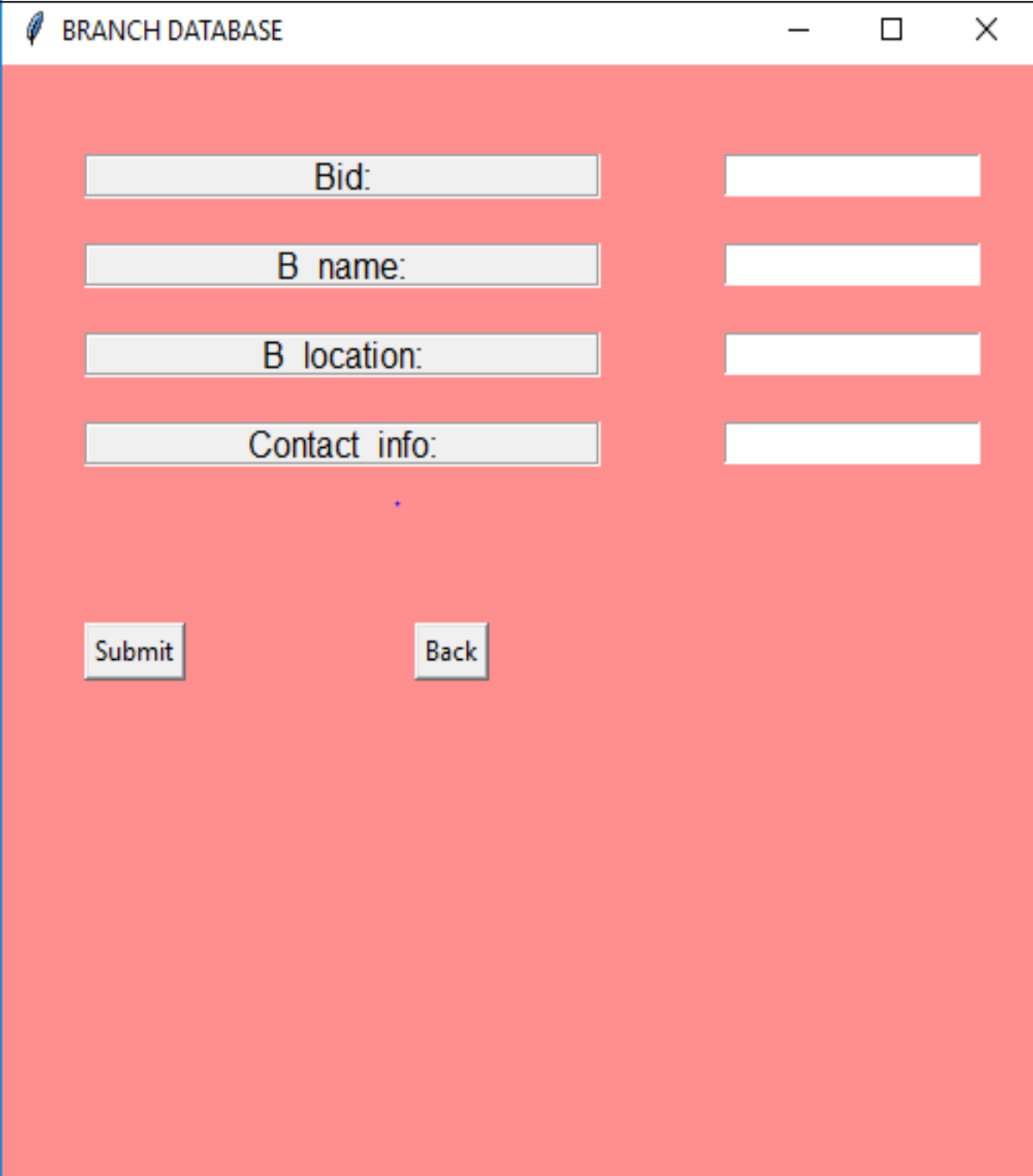
Recipient details

Click to add employee details

Employee details

Exit

To enter the details of the branch

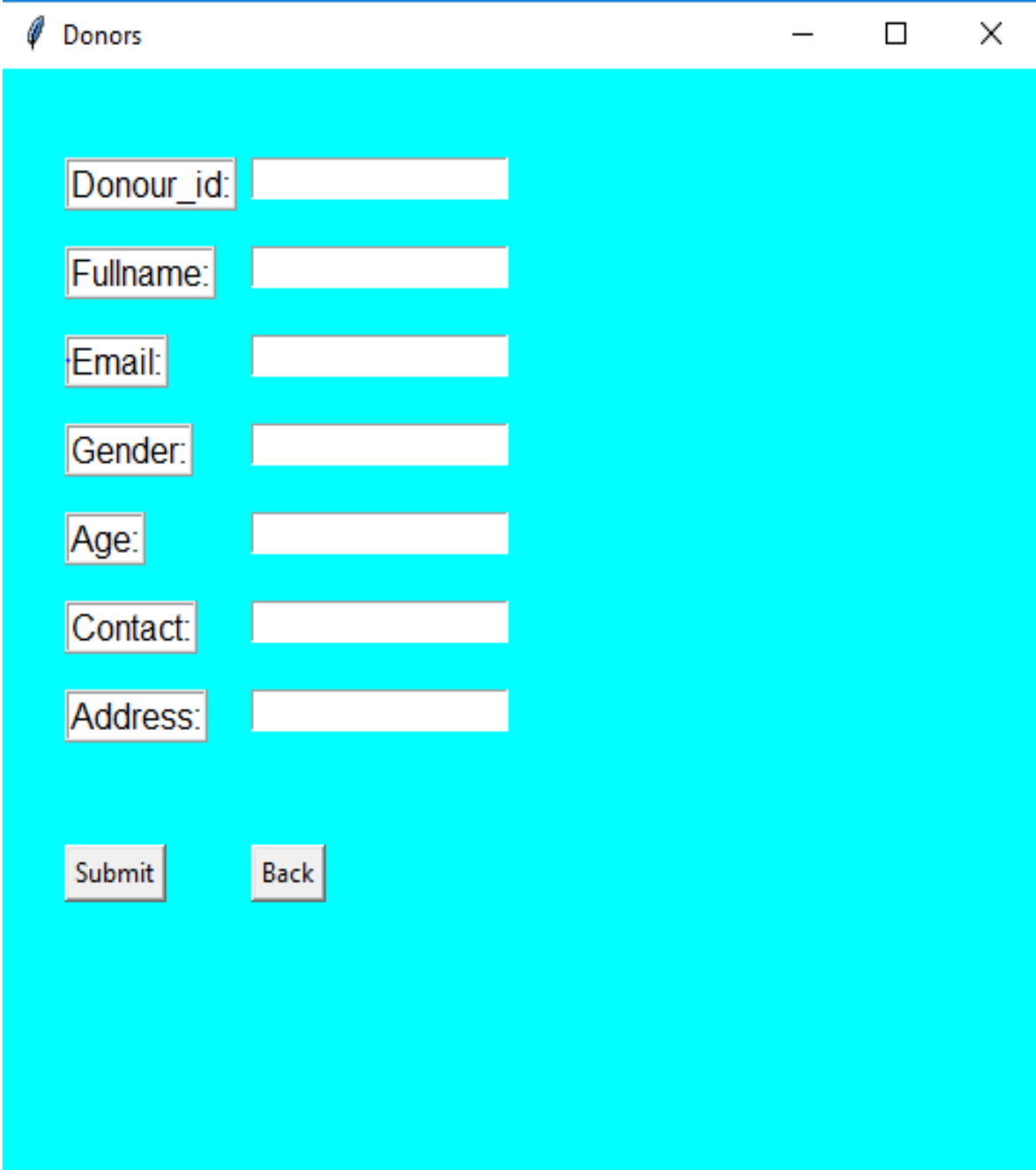


The screenshot shows a Java Swing window titled "BRANCH DATABASE" with a pink background. The window contains four input fields for entering branch details, each with a corresponding text label. The labels are "Bid:", "B name:", "B location:", and "Contact info:". Below the input fields are two buttons: "Submit" and "Back".

Field Label	Input Field
Bid:	<input type="text"/>
B name:	<input type="text"/>
B location:	<input type="text"/>
Contact info:	<input type="text"/>

Buttons: Submit, Back

To enter the details of the donor



Donors

Donour\_id:

Fullname:

Email:

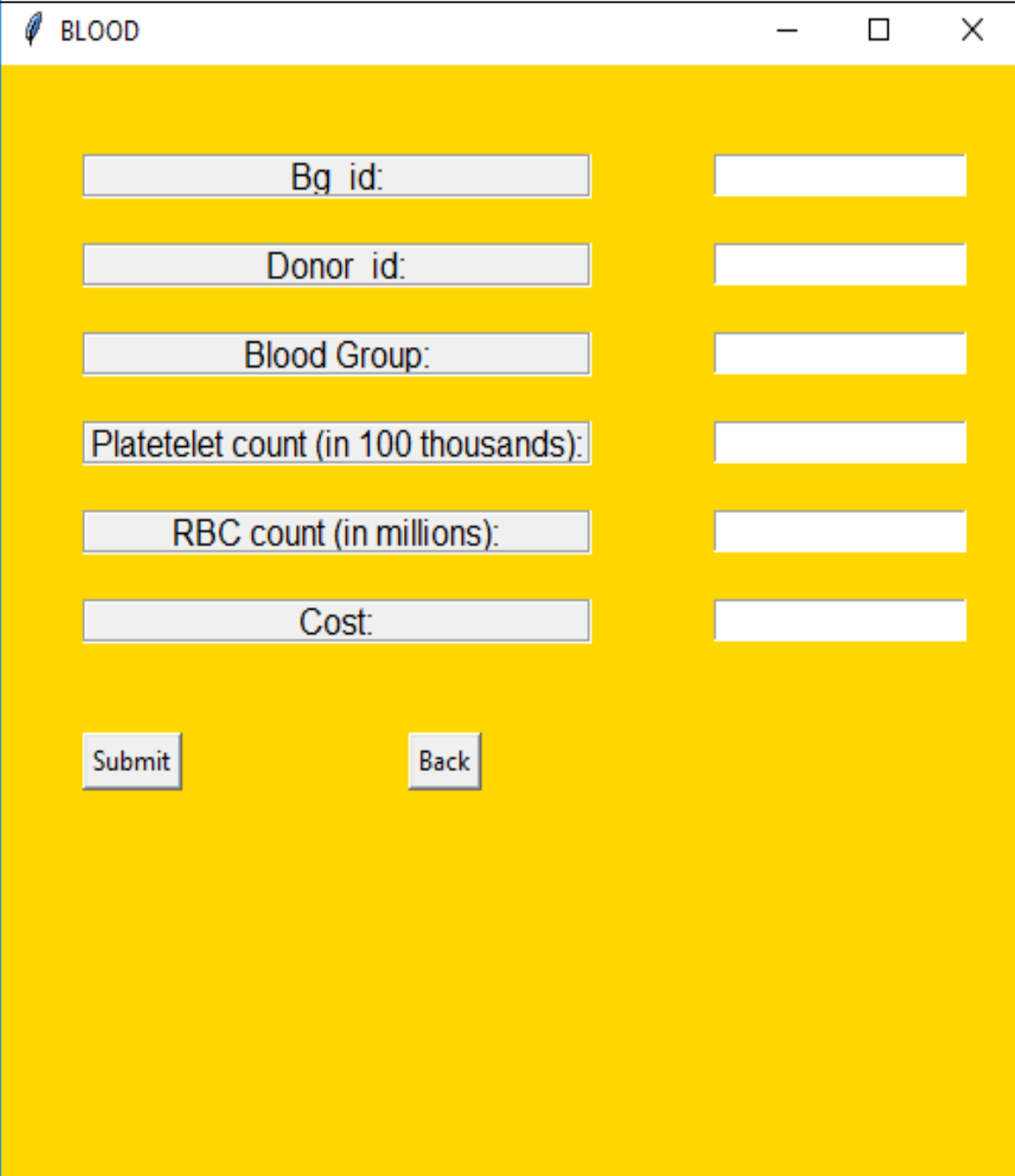
Gender:

Age:

Contact:

Address:

To enter the details of blood



The screenshot shows a Java Swing window titled "BLOOD" with a yellow background. The window contains six text input fields arranged in two columns. The labels for the fields are: "Bg id:", "Donor id:", "Blood Group:", "Platetelet count (in 100 thousands):", "RBC count (in millions):", and "Cost:". Below the input fields are two buttons: "Submit" and "Back".

Field Label	Input Field
Bg id:	<input type="text"/>
Donor id:	<input type="text"/>
Blood Group:	<input type="text"/>
Platetelet count (in 100 thousands):	<input type="text"/>
RBC count (in millions):	<input type="text"/>
Cost:	<input type="text"/>

Buttons:

To enter the details of Recipients



RECIPIENTS

Pid:

Bg\_id:

Name:

Age:

Bloodgrp:

Amtofblood:

Contactno:

Dateofentry:



To enter the details of blood bank employees



The image shows a graphical user interface (GUI) window titled "Employee DB". The window has a light green background and contains several input fields and two buttons. The input fields are arranged in two columns. The left column contains labels for "Eid:", "Bid:", "E name:", "address:", "Contact details:", "Salary:", and "Designation:". The right column contains corresponding empty text input boxes. At the bottom of the form, there are two buttons: "Submit" and "Back".

Eid:	<input type="text"/>
Bid:	<input type="text"/>
E name:	<input type="text"/>
address:	<input type="text"/>
Contact details:	<input type="text"/>
Salary:	<input type="text"/>
Designation:	<input type="text"/>
<input type="button" value="Submit"/>	<input type="button" value="Back"/>

## CHAPTER 6

It contains conclusion and bibliography (References)

### CONCLUSION

- The purpose of conducting this study and doing this project is to understand the significance of relational database.
- Databases were made to make our lives a little easier by helping us deal with the data in a simplified manner.
- Companies could keep up with substantial amounts of information in a singular place.
- Queries facilitate realize this knowledge quicker by punching in key words.
- This brings out smart good yield in an exceedingly work place.
- Python helps in understanding the significance of GUI by providing tkinter and and conjointly the importance of databases by sqlite3 modules.
- We include different screen layouts and other widgets with the help of Tkinter

## REFERENCES

- Using sql with tkinter examples: <https://www.GeeksforGeeks.com>
- Review of literature on blood bank database:
- <https://www.ukessays.com>
- [www.github.com](http://www.github.com)
- www.youtube.com for the usage of sqlite3
- www.wikipedia.org

## Books referred are:

- Raghu Ramakrishnan - "Database Management Systems",2003
- Charles severance - "Python for everybody"

