

CHAPTER 1

INTRODUCTION

Blood bank management system is for creating and managing records. It provides users and programmers with a systematic way to create, retrieve, update and manage data DBMS makes it possible for end users to create, read, update and delete details in a database. The objective and scope of my project is to record the details and various activities of user. It will simplify the task and reduce the paper work specific support will also be provided at key points within the academic calendar. The system is very user friendly and it is anticipated.

The main features of the management system includes information retrieval facility for the users from anywhere in the form of obtaining statistical information about blood storage, donor details, patients details, the system allows the updating facility to the authorized person. JAVA was used to create front end for the system and SQL used for the back end. The main goal of the Blood Bank Management System project is to monitor Blood Bank data, Blood stock, Donor List. It manages all the Blood Bank, Donor, Blood stock data. The project is entirely administrative and therefore access is guaranteed only to the administrator.

1.1 FEATURES

Registered User:

- User needs to first give his username and password on the login portal.
- The user now can enrol as patient with the information.
- The patient gets registered in the patient portal.
- All the patients' information is stored in patient table.
- We can view, insert, and update the details.
- Donor's information is stored in Donor's table.
- The packets of blood donated, blood group all are stored in donor's table.
- The blood table consists of bloods that are present in the bank.
- Increment and decrement of blood packets are been updated.
- Billing of blood is made.

1.2 Objectives

- Blood Bank Management System (BBMS) is designed to store, process, retrieve and analyse information concerned with the administrative and inventory management within a blood bank.
- Features include product oriented data input, inventory control reports, product utilization reports, rapid retrieval of individual patient reports.
- Blood banks play an important role in the process of collecting blood and managing blood stocks, approving blood requests, updating donations and updating available blood types.
- Raster's web-based BBMS will address the issues and problems encountered in collecting information about donors, blood camps, inventories of blood bags, and blood transfusion services, etc, including donor screening, inventory management, blood ordering, blood usage review and compatibility testing.
- Blood Bank Management system will greatly increase the safety and quality of the blood supply as well as provide logistics data for the optimal supply chain management.

1.2 Problem statement

Scarcity of rare blood group, Unavailability of blood during the emergency, Less awareness among people about blood donation and blood transfusion and Deaths due to lack of blood during operations.

1.3 Proposed solution

- Blood bank lists the availability of blood bags at given time.
- It manages the information of its blood donor.
- Alerts for blood requirement from registered donors.

CHAPTER 2

ANALYSIS AND SYSTEM REQUIREMENT

2.1 Existing system

The operation of the blood bank still now is maintained in the manual system. It creates room for errors as the data is entered manually by the persons. Maintaining the stock of blood and the daily transactions without computerization also poses a challenge. Unavailability of blood during emergency. Less awareness among people about blood donation and blood transfusions.

2.2 Hardware and Software Requirements

2.2.1 Hardware Requirements:

- I/O devices
- Monitor
- Keyboard
- Internet Connection

2.2.2 Software Requirements:

- Front end : Java Back end : MYSQL
- Operating system: Windows 10
- Web server: Eclipse, MYSQL server 8.0, mysql connector java8.0.28

2.3 Project Objective and Project Scope

2.3.1 Project Objective

Blood Bank Management System (BBMS) is designed to store, process, retrieve and analyse information concerned with the administrative and inventory management within a blood bank. Features include product oriented data input, inventory control reports, product utilization reports, rapid retrieval of individual patient reports. Blood banks play an important

role in the process of collecting blood and managing blood stocks, approving blood requests, updating donations and updating available blood types.

Raster's web-based BBMS will address the issues and problems encountered in collecting information about donors, blood camps, inventories of blood bags, and blood transfusion services, etc, including donor screening, inventory management, blood ordering, blood usage review and compatibility testing. Blood Bank Management system will greatly increase the safety and quality of the blood supply as well as provide logistics data for the optimal supply chain management.

2.3.2 Project Scope

The proposed Blood Bank Management System help the Blood Bank Admin to easily monitor the blood requests and users database. The proposed system takes a systematic approach of how to bridge the gap between Recipients, Donors, and Blood Banks.

CHAPTER 3

SYSTEM DESIGN AND MODELLING

3.1 Preliminary design

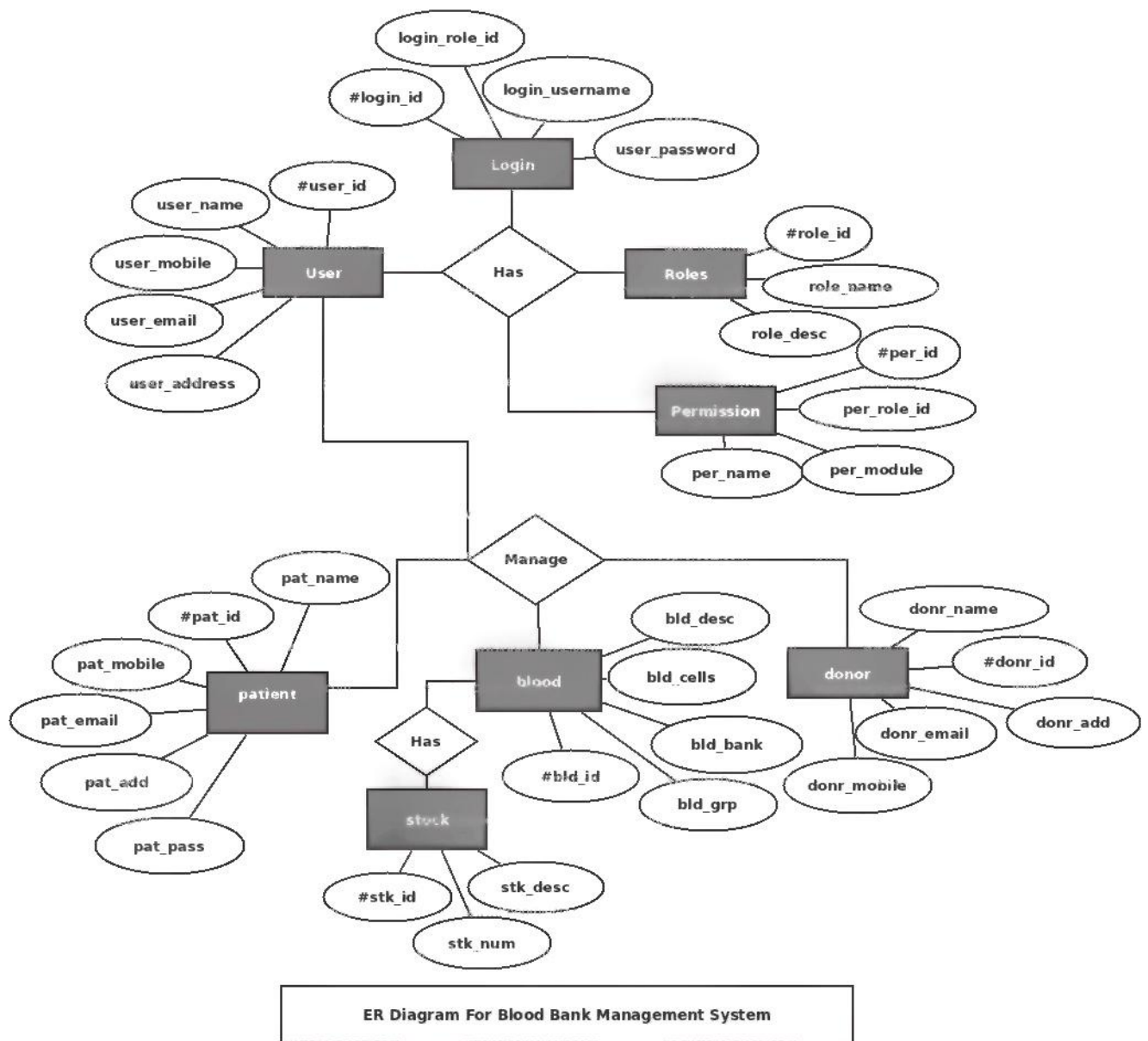
System design is an abstract representation of a system component and their relationship and which describe the aggregated functionally and performance of the system. It is also the overall plan or blueprint for how to obtain answer to the question being asked. The design specifies various type of approach.

Database design is one of the most important factors to keep in mind if you are concerned with application performance management. By designing your database to be efficient in each call it makes and to effectively create rows of data in the database, you can reduce the amount of CPU needed by the server to complete your request, thereby ensuring a faster application.

3.2 ER diagram

- **Entity-relationship diagram:** This depicts relationship between data objects. The attribute of each data objects noted in the entity-relationship diagram can be described using a data object description.
- **Relationship:** Data objects are connected to one another in a variety of different ways. We can define set of object relationship pairs that define the relevant relationships.
- **Cardinality ratio:** The data model must be capable of representing the number of occurrences of objects in a given relationship. The cardinality of an object relationship pair is 1:N, 1:1, N:N, N:1.

Figure 3.1.1 describes the ER diagram of Cricket record Management System. It has entities namely seeds names , fertilizers names & ware house details The entities have attributes which are primary, foreign and composite attributes. The primary attributes are underlined.



3.2 Schema diagram

Database schema is described as database connections and constraints. It contains attributes. Every database has a state instances represent current set of database with values. There are different types of keys in a database schema.

A primary key is a table column that can be used to uniquely identify every row of the table. Any column that has this property, these columns are called candidate key. A composite primary key is a primary key consisting of more than one column. A foreign is a column or

combination of columns that contains values that are found in the primary key of some table. All the attributes of each table are interconnected by foreign key which is primary key in another column and composite key. Primary key cannot be null. The fact that many foreign key values repeat simply reflects the fact that its one-to-many relationship. In one-to-many relationship the primary key has the one value and foreign key has many values.

Figure 3.1.2 is a Schema diagram of blood bank Management System which has five tables i.e., login, patient, donor, blood and billing where each table contain attributes some with primary key, foreign key.

In the patient table we have five attributes i.e., patient id (primary key), patient name, patient mobile no, patient email, purpose.

In the blood table we have four attributes i.e., blood group, number of packets, rate , date.

In the donor table we have four attributes i.e., donor name, donor blood group , donor mobile number, number of packets.

In the billing table we have patient name(foreign key), blood group, number of packets , amount.

LOGIN

ID	USERNAME	PASSWORD
----	----------	----------

PATIENT

PAT_ID	PAT_NAME	PAT_NO	PAT_EMAIL	PURPOSE
--------	----------	--------	-----------	---------

BLOOD

BLOODGROUP	NO_OF_PACKETS	RATE	DATE
------------	---------------	------	------

DONOR

DONOR_NAME	DONOR_BGRP	DONOR_MOB	NO_OF_PACKETS
------------	------------	-----------	---------------

BILLING

PAT_NAME	BLOOD GROUP	NO_OF_PACKETS	AMOUNT
----------	-------------	---------------	--------

3.3 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. Let's discuss about anomalies first then we will discuss normal forms with examples. **Anomalies in DBMS** There are three types of anomalies that occur when the database is not normalized. These are – Insertion, update and deletion anomaly.

1. First normal form (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values.

3.3 Login table

User name	password
prathi	P123

First normalization form

CHAPTER 4

IMPLEMENTATION

4.1 Implementation Operations

Searching Details : This is for the user in blood bank management system. If the patient needs blood then the blood bank system will refer the donor who gave the blood packets, the donor and the patients details are stored in the system.

Adding Accessories: Here donor is added newly whenever blood is donated.

4.2 SQL Statement:

4.2.1 Insert Statement:

- Insert into patient values(pat_id int, pat_name varchar, pat_no int, pat_email varchar, purpose varchar);
- Insert into blood values(bloodgroup varchar, no_of_packets int, rate int, date date);
- Insert into donor values(donor_name varchar, donor_bgrp varchar, donor_mob int, no_of_packets int);

Pseudo code:

1. Start system
2. Click login button
3. Enter the username and password to login.
4. Connect to database.
5. Query database to know whether user credentials are correct
6. If not, deny access and return login page with an error message
7. If correct, check if credentials for administrator
8. If yes, allow login
9. Set admin session, re-direct administrator to home page
10. If no, allow login set user session
11. Re-direct user to user home page

4.3 Updating information

1. Click on the update button on patient , donor dashboard
2. Login as patient or donor
3. Give the needed information
4. The already given values will be updated

4.4 View information

All the given information can be viewed in the form of table by the view button.

CHAPTER 5

TESTING

This chapter gives the outline of all the testing methods that are carried out get a bug of free application. Quality can be achieved by testing the product using different techniques at different phases of the project development.

5.1 Testing process

Testing is an integral part of software development. Testing process, in a way 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. In some cases, test cases are done based on the system requirements specified for the product/software, which is to be developed.

5.2 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 an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

5.3 Levels of Testing

Different levels of testing are used in the testing process; each level of testing aims to test different aspects of the system. The basic levels are unit testing, integration testing, system testing and acceptance testing.

5.3.1 Unit testing

Unit testing focuses verification effort on the smallest unit of software design the module. The software built, is a collection of individual modules. In this kind of testing exact flow of control for each module was verified. With detailed design consideration used as a guide, important control paths are tested to uncover errors within the boundary of the module

5.3.2 Integration testing

The second level of testing is called integration testing. In this, many class-tested modules are combined into subsystems, which are then tested. The goal here is to see if all the modules can be integrated properly. We have identified and debugged.

5.3.3 System testing

Here the entire application is tested. The reference document for this process is the requirement document, and the goal is to see IF the application meets its requirements. Each module and component of ethereal was thoroughly tested to remove bugs through a system testing strategy. Test cases were generated for all possible input sequences and the output was verified for its correctness.

5.4 Applications of System:

The system will perform the following tasks:

1. Ability to manage the information of its blood donor for blood requirement from registered donors.
2. Auto-check if the person donated blood in the last 3 months. Allows good documentation about the donor and their blood donation activities.
3. Allows good documentation about the donor and their blood donation activities.
4. Support fast searching to find match blood bags for the right person.
5. Effectively manage blood camps.

CONCLUSION:

This proposed Blood Bank Management System gives a reliable platform for both donors and acceptors. The BMMS is a web-based application that helps to minimize human errors and problems pertaining to data redundancy.

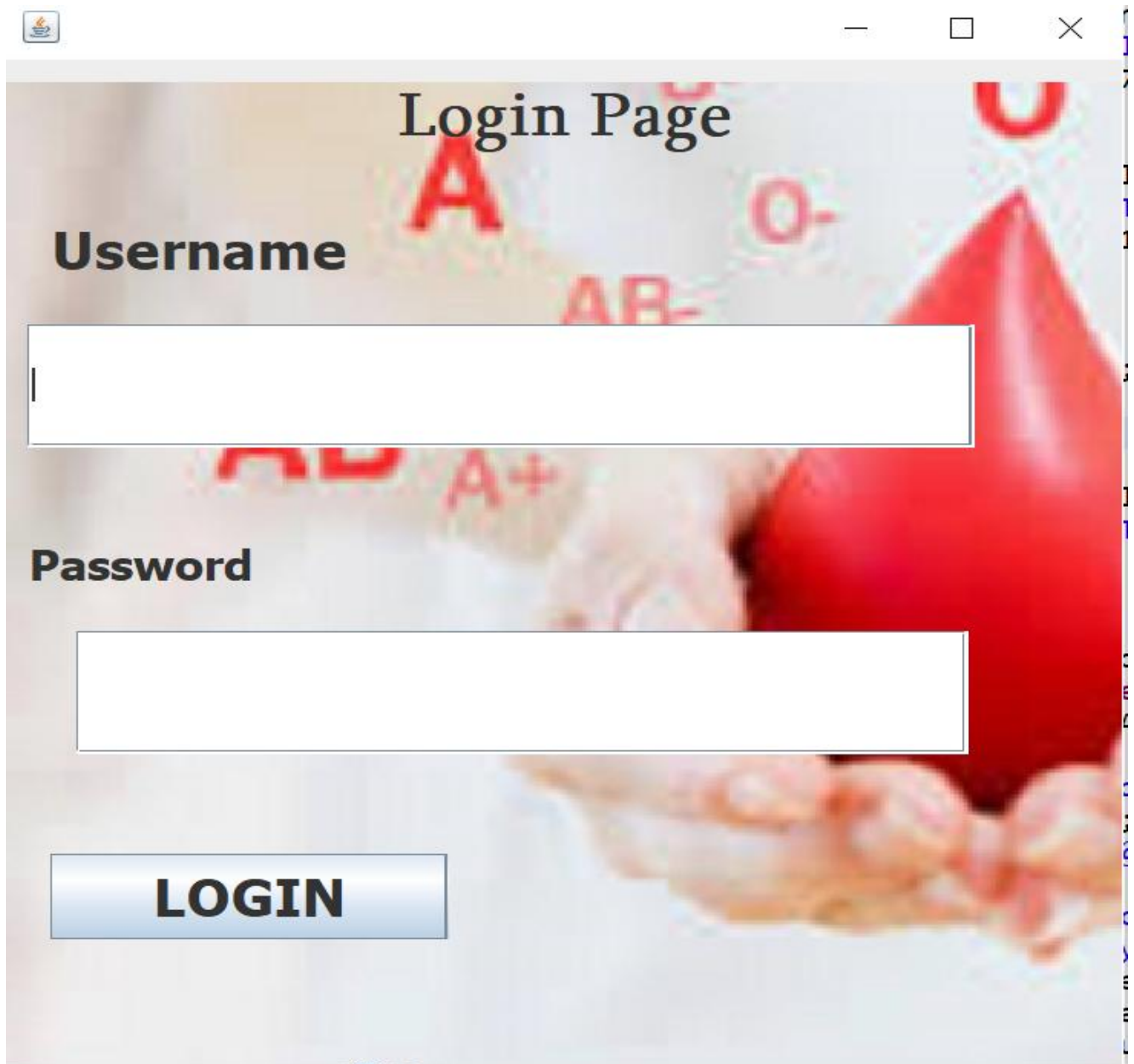
A majority of the donors were willing to be regular donors. The donors showed positive effects like a sense of satisfaction after the donation. Creating an opportunity for blood donation by conducting many blood donation camps may increase the voluntary blood donations.

This project is built to serve many advantages to the organisation. As everything is centralized we can combine many objects in order to perform effective analysis. Effective analysis of data can help a lot in medical field as many other objects and fields can be added to this system for the different blood groups.

REFERENCES

Text book

1. Fundamentals of Database System 5th Edition by Ramez Elmasri
2. Database Management System 3rd Edition by Raghu Ramakrishnan
3. Youtube
4. Introduction to java programming by Richard A Johnson

APPENDIX A

Login Page

Username

Password

LOGIN

Figure A : Login Page

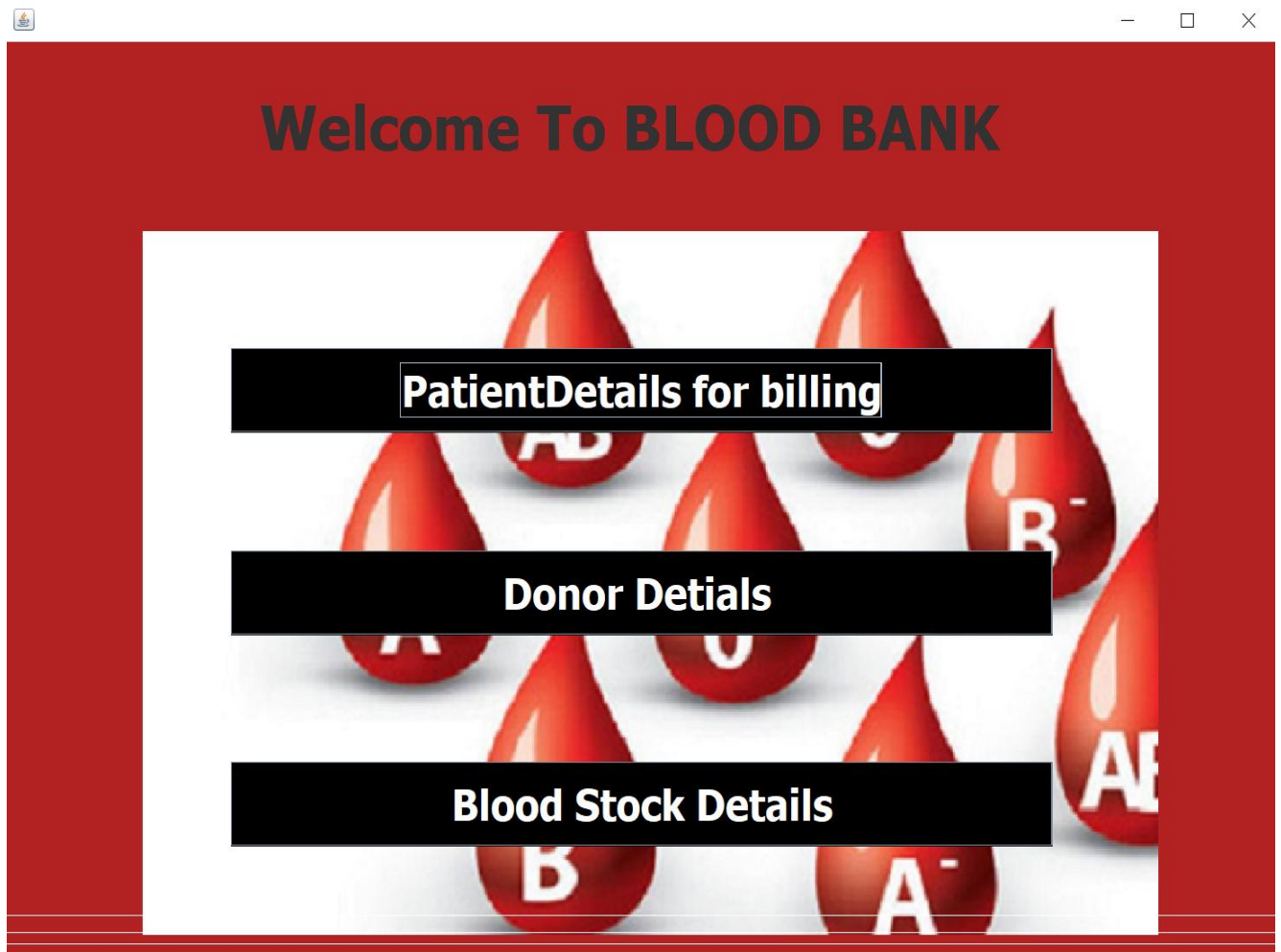


Figure B: Home Page

pat_id

pat_name

pat_mob

pat_email

purpose

bloodgrp

Packets

view

insert

update

back

New column	New column	New column	New column	New column
------------	------------	------------	------------	------------

Figure C: Patients registration

pat_id

pat_name

pat_mob

pat_email

purpose

bloodgrp

Packets

view

insert

update

back

pat_id	pat_name	pat_no	pat_email	purpose	bloodgroup	no_of_packet
1001	Aradya	886193	arday@gmail.com	Accident	O+	1
1002	Aishwarya	90000	test@gmail.com	low bp	O+	2
1003	kavya	123321	test1@gmail.com	high bp	O+	1
1004	komal	123321	test1@gmail.com	high bp	O+	3
1005	komala M	123321	test1@gmail.com	high bp	O+	2
1006	komala MNM	123321	test1@gmail.com	high bp	O+	2
1006	komala MNM	123321	test1@gmail.com	high bp	O+	2
1007	Aishwarya rai	90000	test@gmail.com	low bp	O+	3
1007	Aishwarya rai bac...	900009	test@gmail.com	low bp	O-	3
1008	adaivani	123321	test3@gmail.com	high bp	A+	4
1003	kavya	123321	test1@gmail.com	high bp	O-	3
1008	Priya	897654	priya@gmail.com	cancer	AB+	2
1009	divya	678967	divua@gmail.com	accident	A+	1
1010	sanju	897654	sanj@gmail.com	accident	O+	1
1011	rrr	765467	rrr@gmail.com	accident	O-	1

Figure D:Patient details

donor_name

donor_bgrp

donor_mob

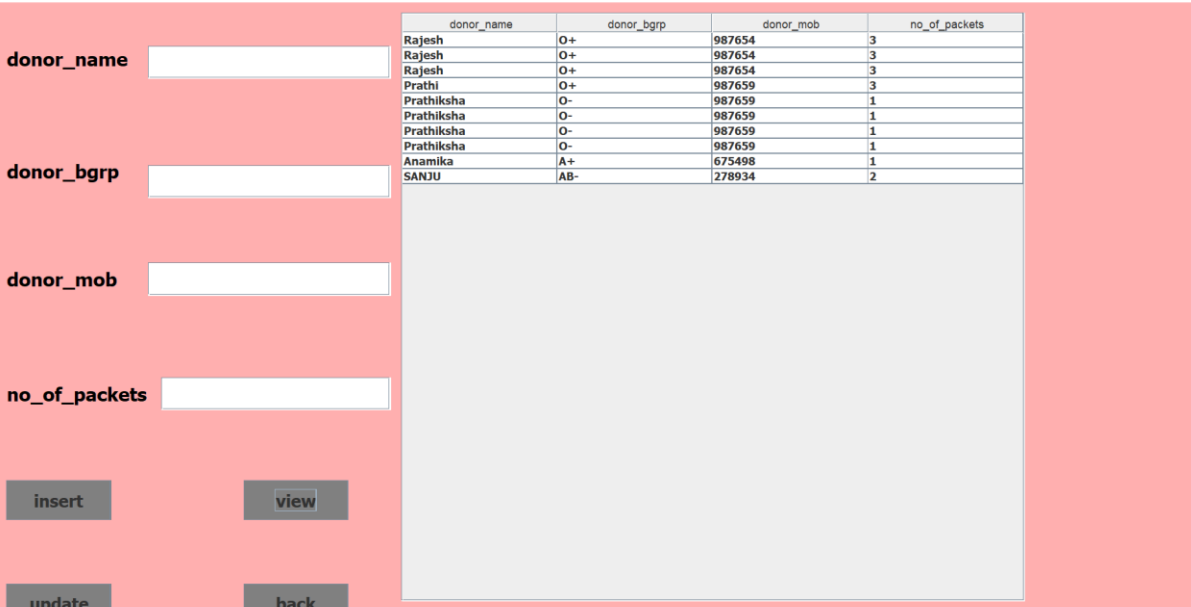
no_of_packets

insert view

update back

New column	New column	New column	New column
------------	------------	------------	------------

Figure E: Donor registrations



donor_name	donor_bgrp	donor_mob	no_of_packets
Rajesh	O+	987654	3
Rajesh	O+	987654	3
Rajesh	O+	987654	3
Prathi	O+	987659	3
Prathiksha	O-	987659	1
Prathiksha	O-	987659	1
Prathiksha	O-	987659	1
Prathiksha	O-	987659	1
Anamika	A+	675498	1
SANJU	AB-	278934	2

Figure F: Donor details

The screenshot displays a web application interface for a Blood Bank Management System. On the left, there is a dark sidebar with a light blue header. The sidebar contains four input fields with labels: "Bloodgroup", "no_of_pac...", "Rate", and "Date". Below these fields are three buttons: "view", "insert", and "update". The main content area on the right is white and features a table with four columns, each labeled "New column". The table is currently empty.

Figure G: Blood stock

The screenshot displays a web application interface for a Blood Bank Management System. On the left, there is a form with four input fields: 'Bloodgroup', 'no_of_pac...', 'Rate', and 'Date'. Below these fields are four buttons: 'view', 'insert', 'update', and 'back'. On the right, there is a table showing the current blood stock details.

bloodgroup	no_of_packets	rate	date
A+	5	700	2019-01-02
B+	5	600	2022-01-01
O+	5	600	2019-03-04
O-	5	600	2022-02-03
AB+	5	900	2022-04-05
B+	5	600	2022-01-01
O-	5	600	2022-02-03

Figure H: Blood stock details

BLOODBANK

Patient Name :rrr
Blood Group :O-
Number Of Packets:1
Amount :600

Join Hands to Help Others

Print Back

Figure I: Snapshot of Billing