TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction to DBMS

Database is a collection of related data. DBMS came into existence in 1960 by Charles. Again in 1960 IBM brought IMS-Information management system. In 1970 Edgor Codd at IBM came with new database called RDBMS. In 1980 then came SQL Architecture Structure Query Language. In 1980 to 1990 there were advances in DBMS e.g. DB2, ORACLE. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database.

- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.

- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents.

Metadata (meta data, or sometimes meta information) is "data about data", of any sort in any media. An item of metadata may describe a collection of data including multiple content items and hierarchical levels, for example a database schema. In data processing, metadata is definitional data that provides information about or documentation of other data managed within an application or environment. The term should be used with caution as all data is about something, and is therefore metadata.

A database management system (DBMS) is a collection of programs that enables users to create and maintain database. The DBMS is a general-purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.

Defining a database specifying the database involves specifying the data types, constraints and structures of the data to be stored in the database. The descriptive information is also stored in the database in the form database catalogue or dictionary; it is called meta-data. Manipulating the data includes the querying the database to retrieve the specific data. An application program accesses the database by sending the queries or requests for data to DBMS. The important function provided by the DBMS includes protecting the database and maintain the database.

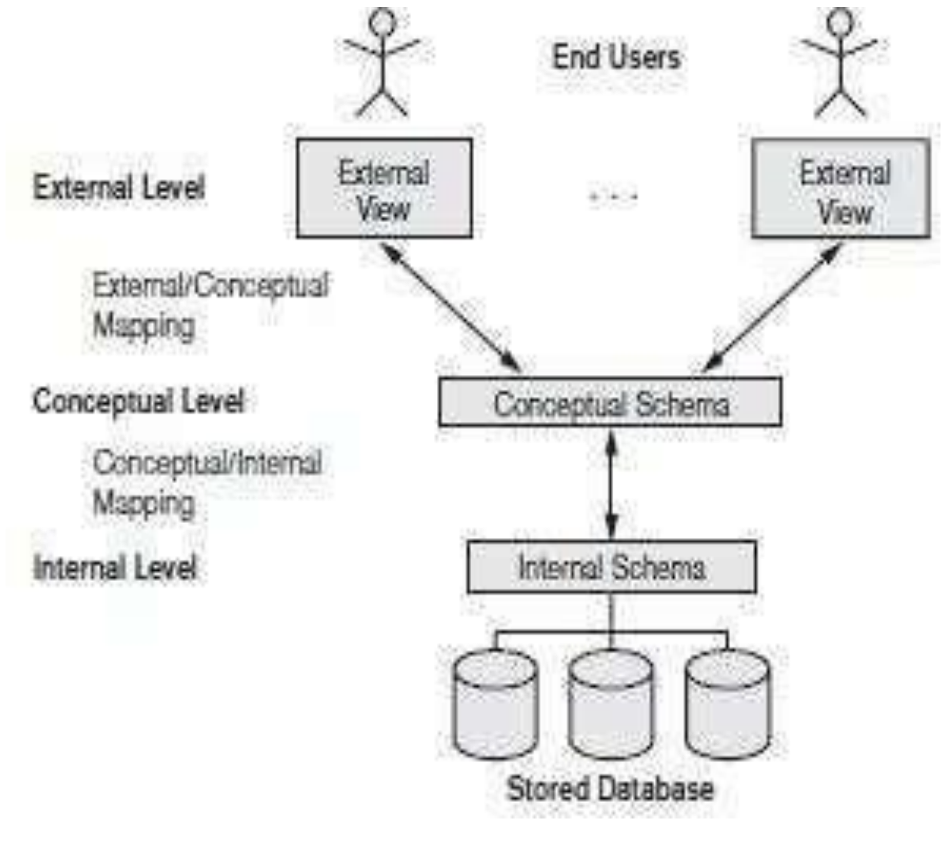


Figure 1.1: Three Schema Architecture

The Figure 1.1 shows the Three Schema Architecture of Database Management System. The architecture has three levels

• **External level:**

The external level is the view that the individual user of the database has. This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users. In general, the end users and even the application programmers are only interested in a subset of the database.

• **Conceptual level**:

The conceptual view is the informational model of the enterprise and contains the view of the whole enterprise without any concern for the physical implementation. The conceptual view is the overall community view of the database and it includes all the information that is going to be represented in the database.

• **Internal level:**

The internal view is the view about the actual physical storage of data. It describes what data is stored in the database and how.

CHAPTER 2

SYSTEM DESIGN AND METHODOLOGY

2.1 System Architecture

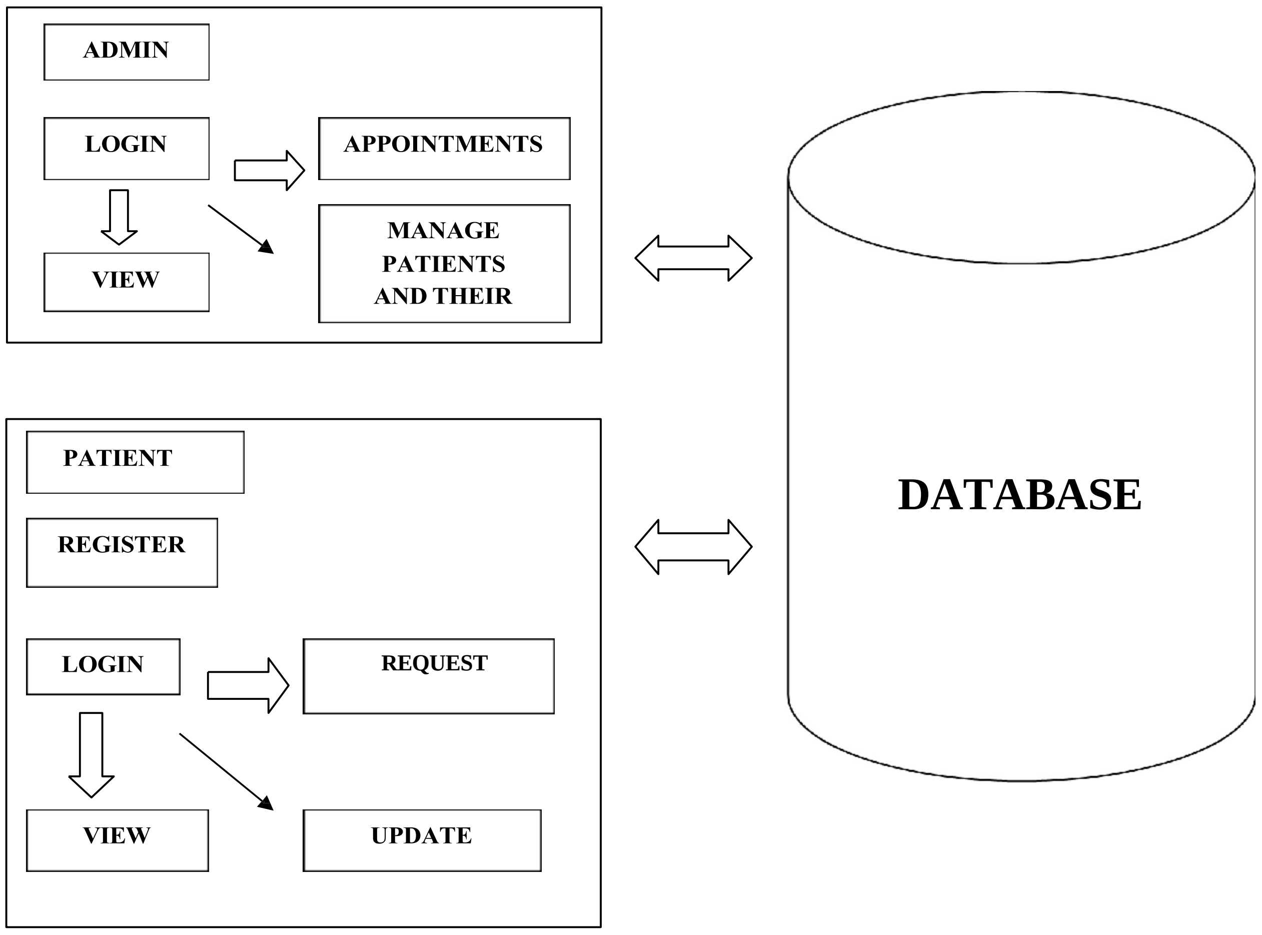


Figure 2.1: System Architecture of Blood Bank Management System

The above figure 2.1 describes the system architecture of Blood bank Management System. The architecture consists of a centralized database, which can be accessed by admin and normal users or patients. Administrative access is required for the admin, which is implemented through login module by which the admin can login with the registered userId and password.

Once login is successful, the admin can manipulate the updating or deleting patients users and their services, adding, updating and deleting categories .

If the admin fails to login, the admin gets a popup message that the userId and password are incorrect.

The admin needs to enter valid user Id and password once again.

A patient first needs to register in the website by filling up valid user Id,name, email,
address and password.

Username should be unique and if it is already been taken then a popup message is displayed that the username is already taken.

Password confirmation should be made by retyping the password again. If the password and password confirmation do not match then it produces a popup message stating that retype the password again.

On successful registration, patient can login through valid userId and password. On successful login, patient is directed to services list where patients can view services of their interest and then add to list, select appointment time, make payment and request the.
Patient can also update their password and address.

2.2 ER DIAGRAM

An entity-relationship model describes inter-related things of interest in specific domain of knowledge. An ER module is composed of entity types and specifies relationships that can exist between instances of those entity types. It is a data modeling technique that graphically illustrates an information systems entity and the relationship between those entities.

- This document is an entity-relationship diagram, or “ERD,” for a system to manage Blood bank Management System.
- An ERD is a model that identifies the concepts or entities that exist in a system and the relationships between those entities.
- An ERD is often used as a way to visualize a relational database: each entity represents a database table, and the relationship lines represent the keys in one table that point to specific records in related tables.
- ERD may also be more abstract, not necessarily capturing every table needed within a database, but serving to diagram the major concepts and relationships.
- This ERD is of the latter type, intended to present an abstract, theoretical view of the major entities and relationships needed for management of electronic resources.
- It may assist the database design process for an e-resource management system, but does not identify every table that would be necessary for an electronic resource management database

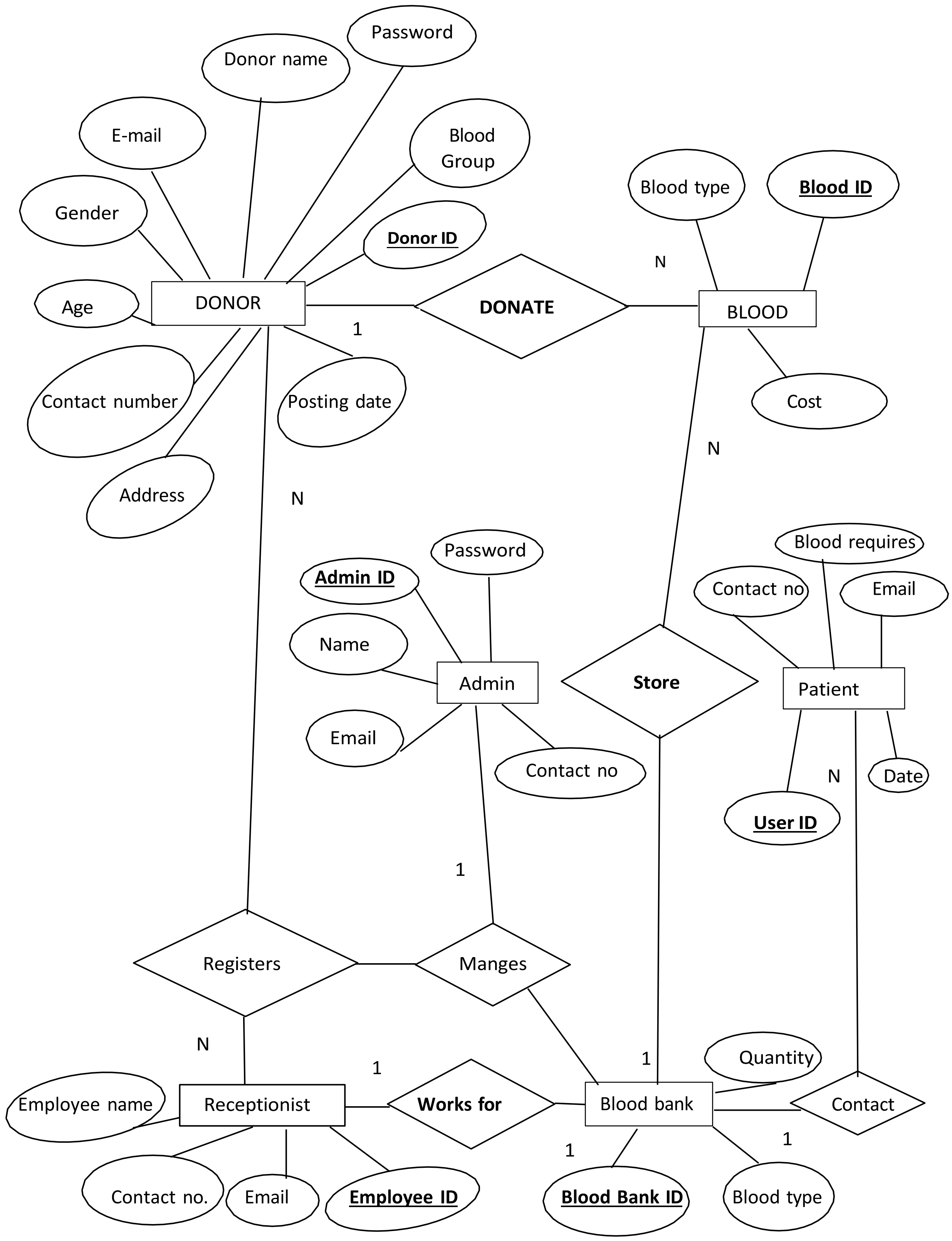


Figure 2.2: ER Diagram

2.3 SCHEMA DIAGRAM

A database schema is a skeleton structure that represents the logical view of the entire database. It defines tables, views and integrity constraints.

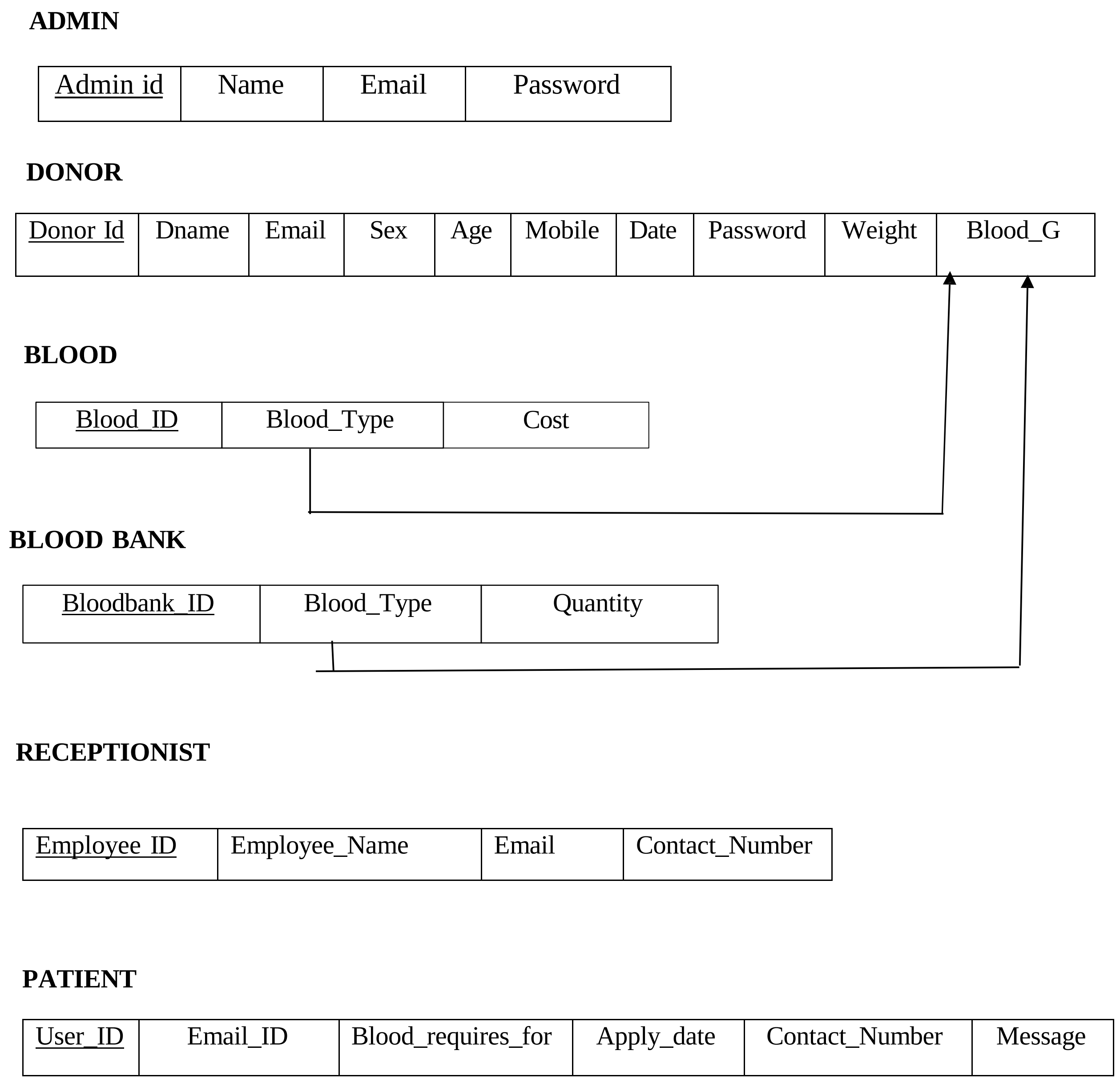

ADMIN
Admin id | Name | Email | Password
DONOR
Donor Id | Dname | Email | Sex | Age | Mobile | Date | Password | Weight | Blood_G
BLOOD
Blood_ID | Blood_Type | Cost
BLOOD BANK
Bloodbank_ID | Blood_Type | Quantity
RECEPTIONIST
Employee ID | Employee_Name | Email | Contact_Number
PATIENT
User_ID | Email_ID | Blood_requires_for | Apply_date | Contact_Number | Message

Figure 2.3: Schema Diagram

2.4 USE CASE DIAGRAM

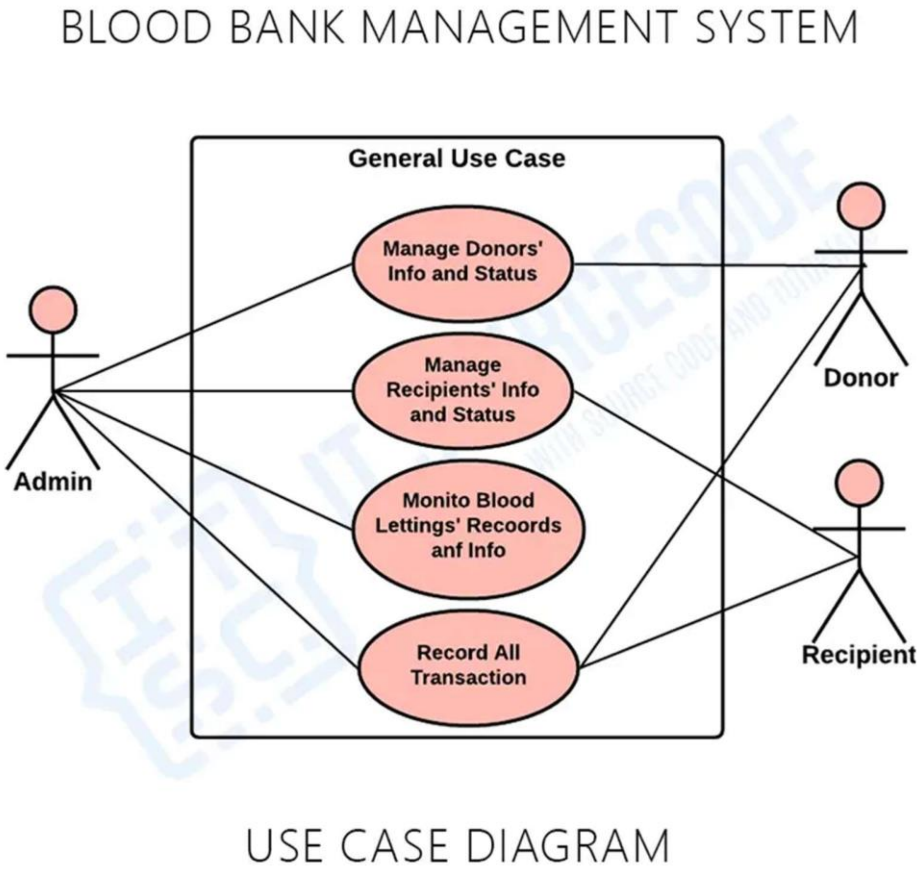

BLOOD BANK MANAGEMENT SYSTEM
General Use Case
Manage Donors' Info and Status
Manage Recipients' Info and Status
Monito Blood Lettings' Recoords anf Info
Record All Transaction
Admin
Donor
Recipient
USE CASE DIAGRAM

Figure 2.4: Use Case Diagram

Patient:

Here the patients books for an appointment and receives the blood provided by the blood bank**.**

Recipient:

They provide the services as per the requirement of the patient

Admin:

The admin has the control to accept and reject appointments, enquire about the patient details, services updation .

2.5 ALGORITHM:

Stored procedure

STEP 1: BEGIN

STEP 2: DELIMITER $$

STEP 3: CREATE DEFINER=`root`@`localhost` PROCEDURE `request_list` () NOSQL

```
SELECT
A  request_id,a.patient_id,s.service_name,CONCAT(c.phone_no,'
,’c.email_id,' ,c.state,' ,',c.city,' ) AS Address,a.number of services as NUM

 FROM  appointment a  INNER  JOIN services s  ON s.service_id=a.sevice_id
 INNER JOIN patient c ON c.patient_id=a.patient_id

 WHERE  a.appointment_status='APPOINTMENT_CONFIRMED'  REQUEST
 BYp.time_stamp           ASC$$

   DELIMITER ;
```

STEP 4: END

Stored procedure is a set of sql statement . The procedure is created by the name appointment list in the created definer `root`@`localhost` with select all the attributes of appointment table by giving appointment id as condition .

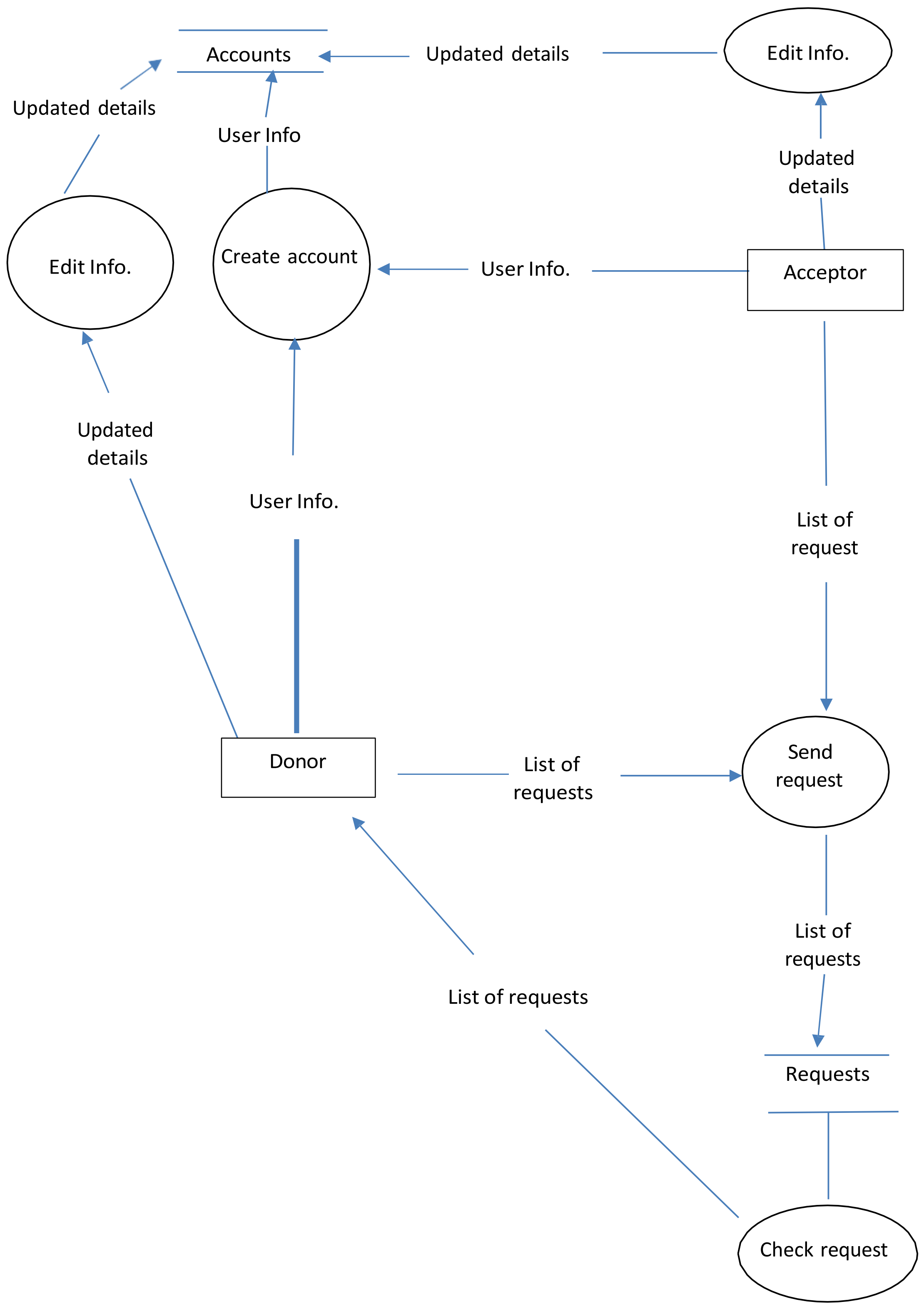


Figure 2.5: Data flow diagram

CHAPTER 3

SYSTEM IMPLEMENTATION

System Requirements specification:

Hardware Requirements:

1. Processor: Intel Core i5
2. RAM: 4GB and above
3. Hard Disk: 10GB and above.
4. Keyboard: Standard Keyboard.
5. Monitor: 15.6 inch.
6. Mouse: Optical Mouse.

Software Requirements:

System Software: Windows 7,10,11.
Application Software: Visual Studio Code Software.

Programming Languages:

1. Front-end: HTML, CSS, Framework (Bootstrap).
2. Back-end: MySQL or Python (For connecting DB to Front-end), Framework (Django for python Based web Framework).

Introduction to Django:

With Django, you can take web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

It's free and open source.

Django was designed to help developers take applications from concept to completion as quickly as possible.

IMPLIMENTATION

Admin page :

```
{% extends 'blood/adminbase.html' %}
{% block content %}
{% load widget_tweaks %}
<style>
.xyz{

display: table;
margin-right: auto;
margin-left: auto;
}
</style>
<br><br>
<div class="container">

<div class="row">
<div class="col-sm-3">
<div class="card bg-light">
<div class="card-body">
<div class="blood">
<h2>A+ <i class="fas fa-tint"></i></h2>
</div><br><br>
<div>
{{A1.unit}}ml
</div>
</div>
</div>
</div>
<div class="col-sm-3">
<div class="card bg-light">
<div class="card-body">
<div class="blood">
<h2>B+ <i class="fas fa-tint"></i></h2>
</div><br><br>
<div>
{{B1.unit}}ml
</div>
</div>
</div>
</div>
<div class="col-sm-3">
<div class="card bg-light">
<div class="card-body">
<div class="blood">
```

```
<h2>O+ <i class="fas fa-tint"></i></h2>
</div><br><br>
<div>
{{O1.unit}}ml
</div>
</div>
</div>
</div>
<div class="col-sm-3">
<div class="card bg-light">
<div class="card-body">
<div class="blood">
<h2>AB+ <i class="fas fa-tint"></i></h2>
</div><br><br>
<div>
{{AB1.unit}}ml
</div>
</div>
</div>
</div>
</div>

<div class="row">
<div class="col-sm-3">
<div class="card bg-light">
<div class="card-body">
<div class="blood">
<h2>A- <i class="fas fa-tint"></i></h2>
</div><br><br>
<div>
{{A2.unit}}ml
</div>
</div>
</div>
</div>
<div class="col-sm-3">
<div class="card bg-light">
<div class="card-body">
<div class="blood">
<h2>B- <i class="fas fa-tint"></i></h2>
</div><br><br>
<div>
{{B2.unit}}ml
</div>
</div>

{% endblock content %}
```

Donor page :

```
{% extends 'donor/donorbase.html' %}
{% block content %}
{% load widget_tweaks %}
{%load static%}
<head>
<!-- Font special for pages-->
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i"
rel="stylesheet">

<!-- Main CSS-->

<link href="{% static "css/main.css" %}"rel="stylesheet" media="all">
</head>
<div class="page-wrapper bg-gra-03 p-t-45 p-b-50">
<div class="wrapper wrapper--w790">

<div style="margin-left: 70px;" class="card card-5">
<div class="card-heading">
<h2 class="title">DONATE BLOOD</h2>
</div>
<div class="card-body">
<form method="POST" autocomplete="off" enctype="multipart/form-data">
{% csrf_token %}

<div class="form-row">
<div class="name">Blood Group</div>
<div class="value">
<div class="input-group">
<div class="rs-select2 js-select-simple select--no-search">
<select name="bloodgroup" class="form-control">
<option disabled="disabled" selected="selected">Please Select</option>
<option>O+</option>
<option>O-</option>
<option>A+</option>
<option>A-</option>
<option>B+</option>
<option>B-</option>
<option>AB+</option>
<option>AB-</option>
</select>
<div class="select-dropdown"></div>
</div>
</div>
</div>
</div>
{% endblock content %}
```

Patient page :

```
{%load static%}
{% load widget_tweaks %}
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.css">

<!-- Font special for pages-->
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,80
0i" rel="stylesheet">

<!-- Main CSS-->
<link href="{% static "css/main.css" %}"rel="stylesheet" media="all">
<style>

</style>
</head>
<body>
{% include "blood/navbar.html" %}




<div class="page-wrapper bg-gra-03 p-t-45 p-b-50">
<div class="wrapper wrapper--w790">
<br><br><br>
<div class="card card-5">
<div class="card-heading">
<h2 class="title">Patient Login</h2>
</div>
<div class="card-body">
<form method="POST">
{% csrf_token %}
<div class="form-row">
<div class="name">Username</div>
</body></html>
```

Urls.py :

```
from django.urls import path

from django.contrib.auth.views import LoginView
from . import views
urlpatterns = [
    path('patientlogin',
LoginView.as_view(template_name='patient/patientlogin.html'),name='patientlogin'),
    path('patientsignup', views.patient_signup_view,name='patientsignup'),
    path('patient-dashboard', views.patient_dashboard_view,name='patient-dashboard'),
    path('make-request', views.make_request_view,name='make-request'),
    path('my-request', views.my_request_view,name='my-request'),
]
```

Settings.py :

```
Django settings for bloodbankmanagement project.

Generated by 'django-admin startproject' using Django 3.0.5.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(_file_)))
TEMPLATE_DIR = os.path.join(BASE_DIR,'templates')
STATIC_DIR=os.path.join(BASE_DIR,'static')
MEDIA_ROOT=os.path.join(BASE_DIR,'static')

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '+zy!9k=9pql5gz9bkqjore)k6r!%w0atk(@(!(!zvp5e(t2i8n'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'widget_tweaks',
'blood',
'donor',
'patient',
]
```

```
MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
CSRF_COOKIE_SECURE=False
ROOT_URLCONF = 'bloodbankmanagement.urls'

TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [TEMPLATE_DIR,],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]

WSGI_APPLICATION = 'bloodbankmanagement.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
}
}


# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators
```

CHAPTER 4

RESULTS AND SNAPSHOTS

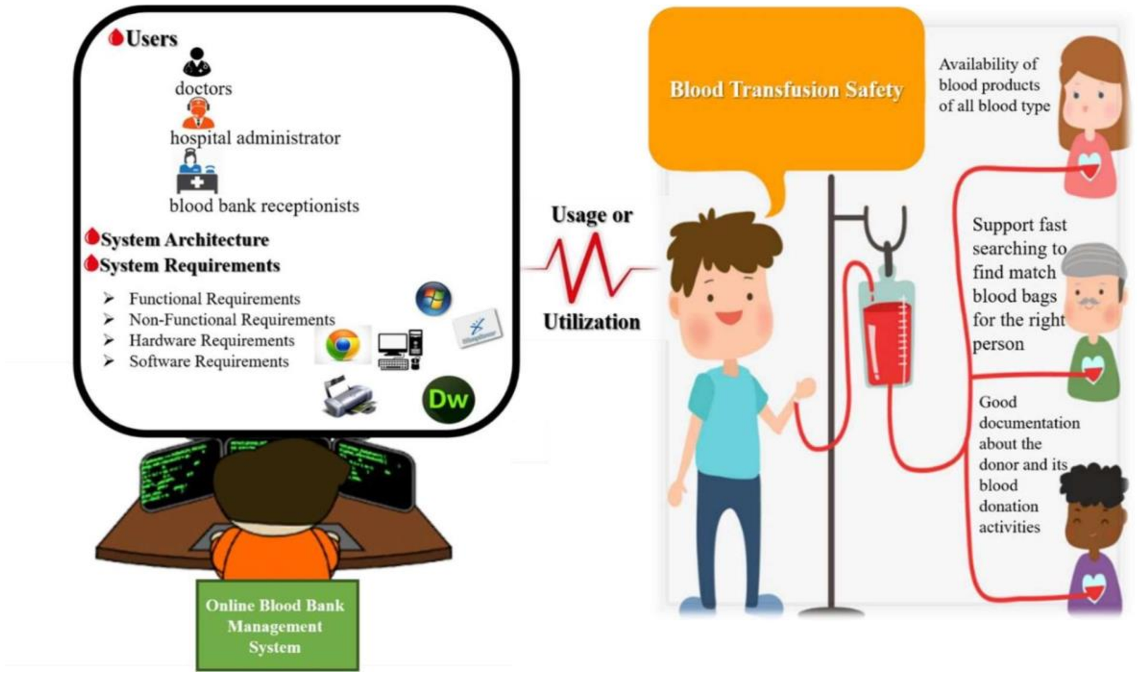
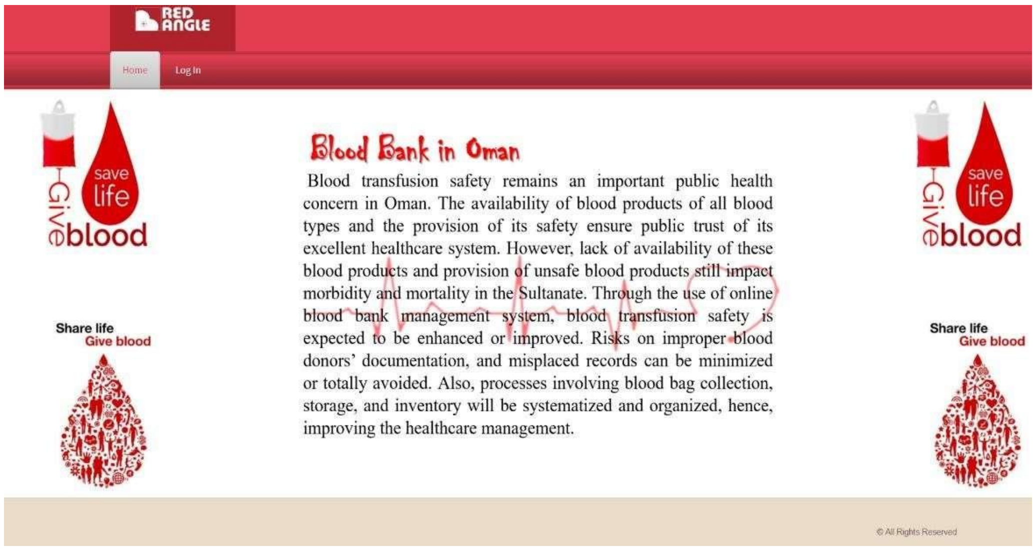

Users
doctors
hospital administrator
blood bank receptionists
System Architecture
System Requirements
Functional Requirements
Non-Functional Requirements
Hardware Requirements
Software Requirements
Online Blood Bank Management System
Usage or Utilization
Blood Transfusion Safety
Availability of blood products of all blood type
Support fast searching to find match blood bags for the right person
Good documentation about the donor and its blood donation activities

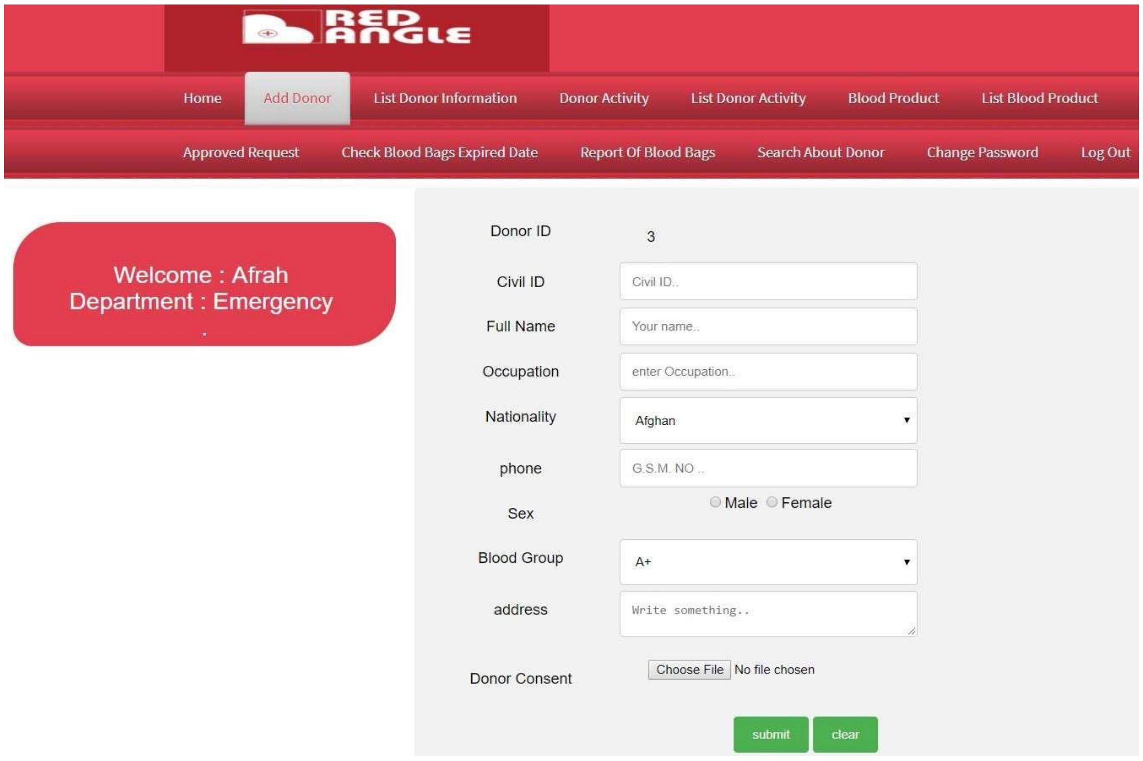
Figure 4.1: Conceptual Framework



Blood Bank in Oman

Blood transfusion safety remains an important public health concern in Oman. The availability of blood products of all blood types and the provision of its safety ensure public trust of its excellent healthcare system. However, lack of availability of these blood products and provision of unsafe blood products still impact morbidity and mortality in the Sultanate. Through the use of online blood bank management system, blood transfusion safety is expected to be enhanced or improved. Risks on improper blood donors' documentation, and misplaced records can be minimized or totally avoided. Also, processes involving blood bag collection, storage, and inventory will be systematized and organized, hence, improving the healthcare management.

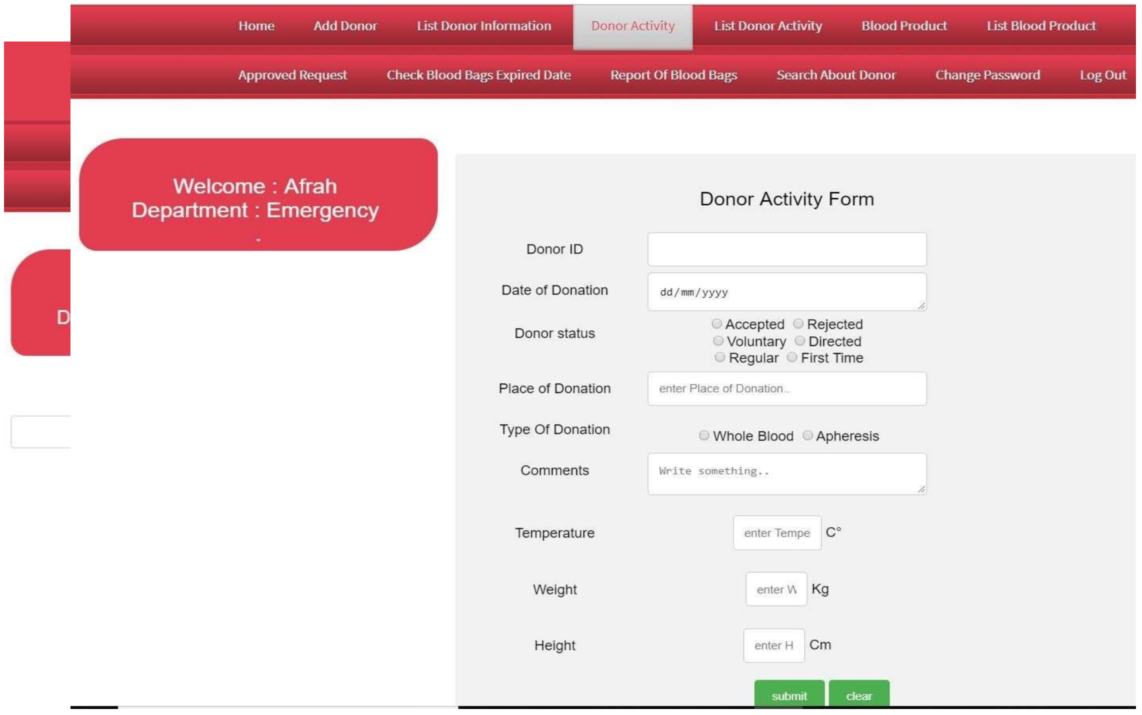
Figure 4.2: Homepage for Online Blood Bank Management System



Welcome : Afrah
Department : Emergency

Donor ID	3
Civil ID	Civil ID..
Full Name	Your name..
Occupation	enter Occupation..
Nationality	Afghan
phone	G.S.M. NO ..
Sex	Male Female
Blood Group	A+
address	Write something..
Donor Consent	Choose File No file chosen

Figure 4.3: Add Donor Page

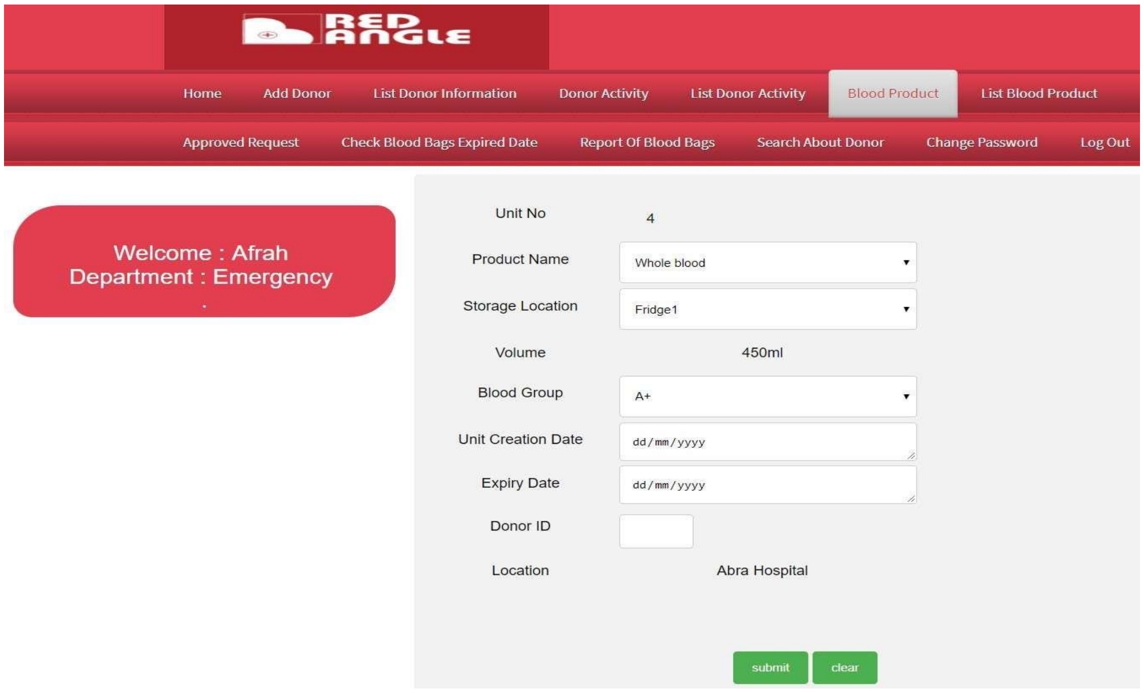


Welcome : Afrah
Department : Emergency

Donor Activity Form

Donor ID	
Date of Donation	dd/mm/yyyy
Donor status	Accepted Rejected Voluntary Directed Regular First Time
Place of Donation	enter Place of Donation..
Type Of Donation	Whole Blood Apheresis
Comments	Write something..
Temperature	enter Tempe C°
Weight	enter W Kg
Height	enter H Cm

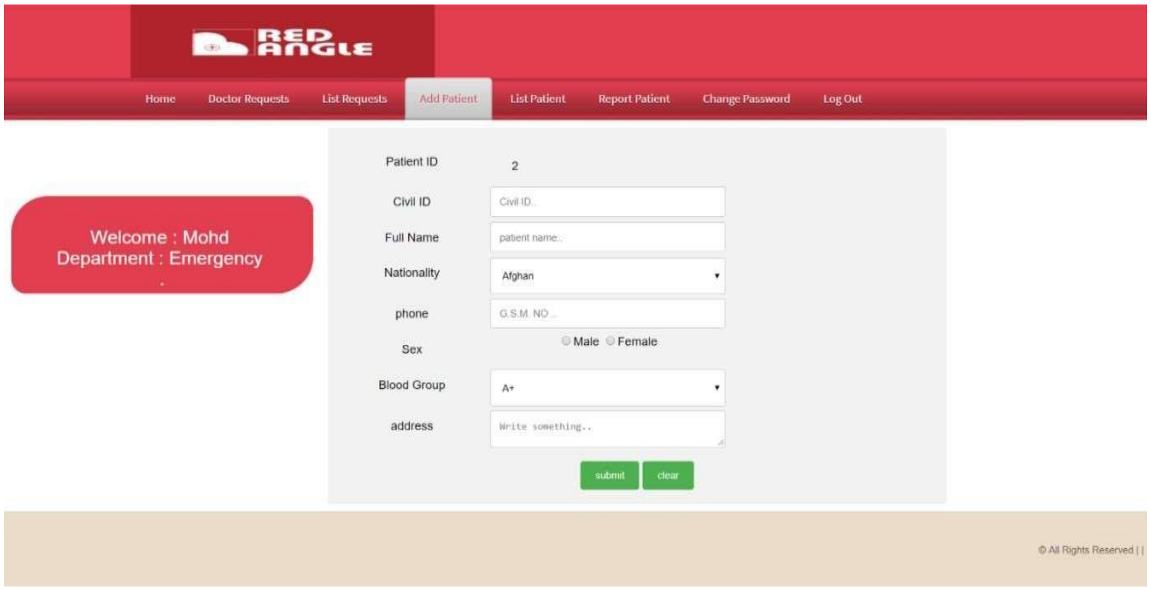
Figure 4.4: Donor Activity Page



Welcome : Afrah
Department : Emergency

Unit No	4
Product Name	Whole blood
Storage Location	Fridge1
Volume	450ml
Blood Group	A+
Unit Creation Date	dd/mm/yyyy
Expiry Date	dd/mm/yyyy
Donor ID	
Location	Abra Hospital

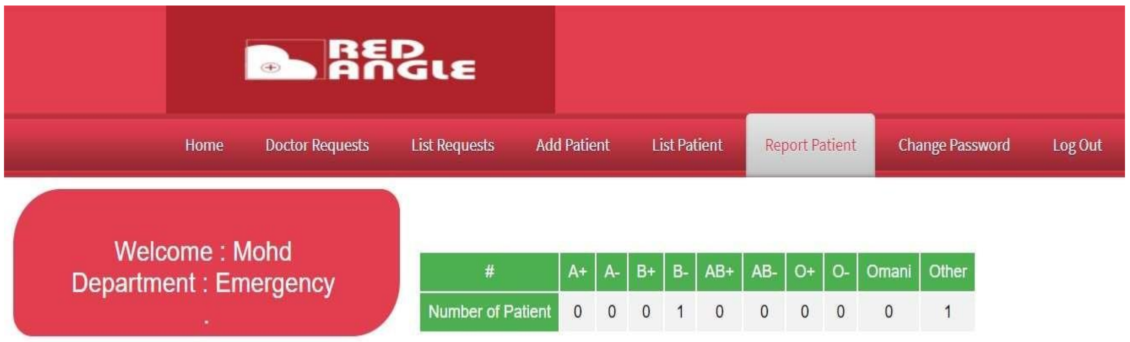
Figure 4.5: Blood Product Page



Welcome : Mohd
Department : Emergency

Patient ID	2
Civil ID	Civil ID..
Full Name	patient name..
Nationality	Afghan
phone	G.S.M. NO...
Sex	Male Female
Blood Group	A+
address	Write something..

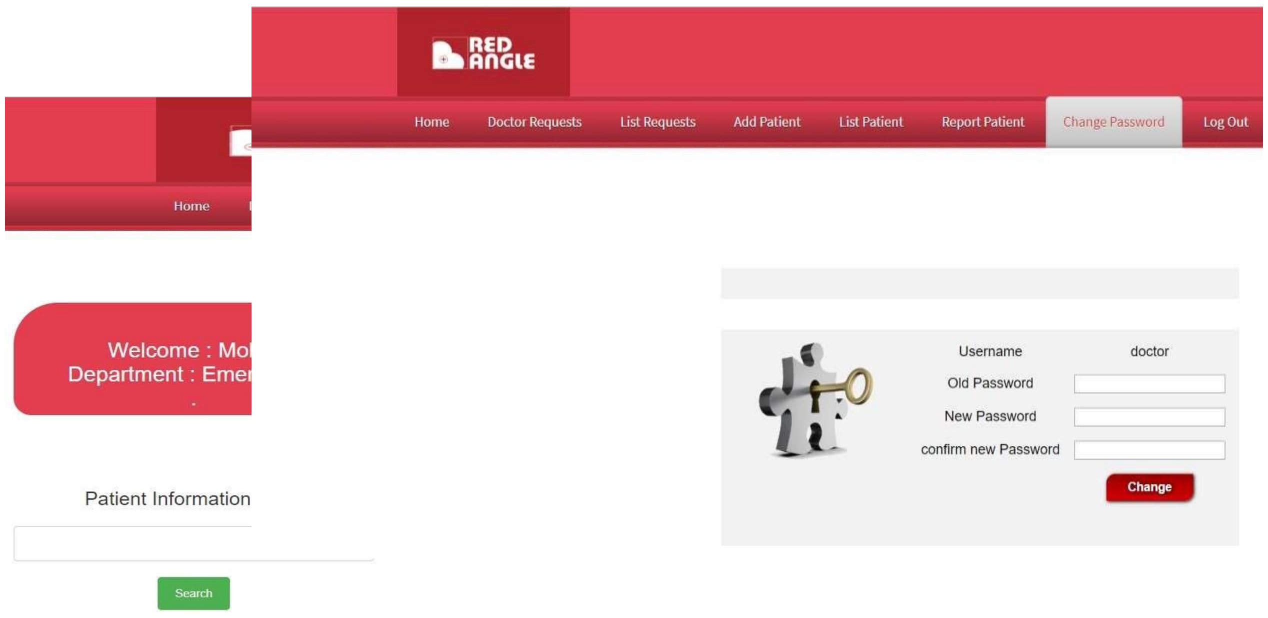
Figure 4.6: Add Patient Page



Welcome : Mohd
Department : Emergency

#	A+	A-	B+	B-	AB+	AB-	O+	O-	Omani	Other
Number of Patient	0	0	0	1	0	0	0	0	0	1

Figure 4.7: Report Patient Page



Welcome : Mo
Department : Emer

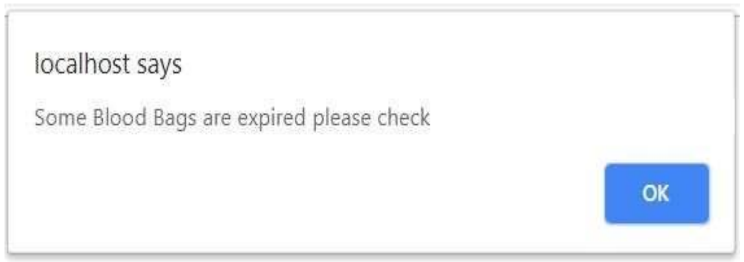
Patient Information

Username	doctor
Old Password	
New Password	
confirm new Password	

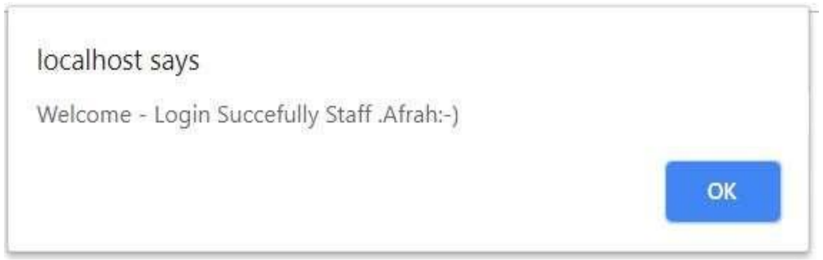
Figure 4.8: Change Password Page

Validations:

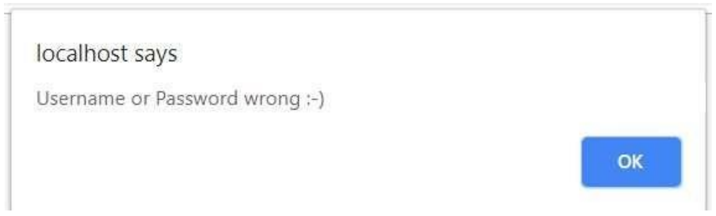
The researcher used a validation check to ensure data consistency and correctness in the system. One of the validation checks is the *required field validation*. Empty fields are not allowed. Also, the password should match. Likewise, it will alert expired blood bags.


localhost says

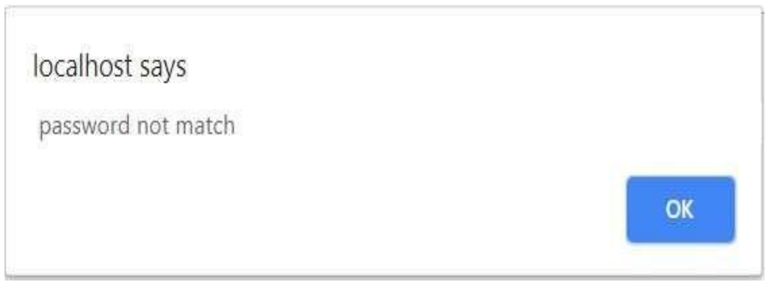
Some Blood Bags are expired please check


localhost says

Welcome - Login Succefully Staff .Afrah:-)


localhost says

Username or Password wrong :-)


localhost says

password not match

Figure 4.9: Validations

CHAPTER 5

CONCLUSION AND FUTURE WORKS

In conclusion, the implementation of the Blood Bank Management System offers a comprehensive solution for efficiently managing blood donation, storage, and distribution processes. Through the utilization of a well-structured database management system, the project ensures accurate record-keeping, streamlined operations, and enhanced accessibility to critical information.

By integrating user-friendly interfaces and robust security measures, the system facilitates seamless communication between donors, recipients, and healthcare professionals, ultimately contributing to the optimization of blood bank operations and the provision of timely and life-saving transfusions. With ongoing updates and improvements, this project stands poised to make a significant impact in the realm of healthcare management, underscoring the importance of leveraging technology to support humanitarian endeavors..

FUTURE WORKS:

In view of the future scope , the recommend that implementation of online blood bank management system in many hospitals . The Display of the Donors location on the website in case of emergency and track the nearest donor .The donors also will receive the message related to shortage of blood in the hospital and its blood group. To ensure that there will be better user engagement, user manuals and proper user training will be given here.Lastly, this study recommends that the system can be expanded by allowing donors to register online and be a system user, and these donors will be informed about the planned blood donation activities through the online.

BIBLIOGRAPHY

1. Rakrishnan, & amp; Gehrke,J.(2011). Database management systems. Boston: McGraw-Hill.
2. Teena, C.A, Sankar, K. and Kannan, S. (2014) “A Study on Blood Bank Management”
3. Kumar, R., Singh, S. and Ragavi, V.A. (2017) .“Blood Bank Management System”
4. Liyana, F (2017) .“Blood Bank Management System Using Rule-Based Method”
5. Monson-Haefel,R. (2007).J2EEWebservices. Boston,Mass.: AddisonWesley.
   Silberschatz, A.,Korth,H.F. ,&amp ; Sudarshan,S.(2011).
6. Database systems concepts. Estados Unidos: McGraw-Hill Companies,Inc.
7. Hanna P. (2002) : JSP 2.0 The Complete Reference , Second Edition
   McGrawHill Education.
8. https://www.w3schools.com/php/default.asp
9. https://www.mysql.com/
10. https://php/default.asp
11. https://www.mysqltutorial.org
12. https://www.apachefriends.org/download.html