Application for Blood Bank Management System

Milestone: Project Report

Group 2
Jwalit Shah

(732-209-2041)

shah.jwa@northeastern.edu

Percentage of effort contributed by Jwalit Shah - 100%

Signature of Jwalit Shah

Submission Date - 10 December 2022

USE CASE STUDY REPORT

Group 2

Name - Jwalit Shah

Executive Summary

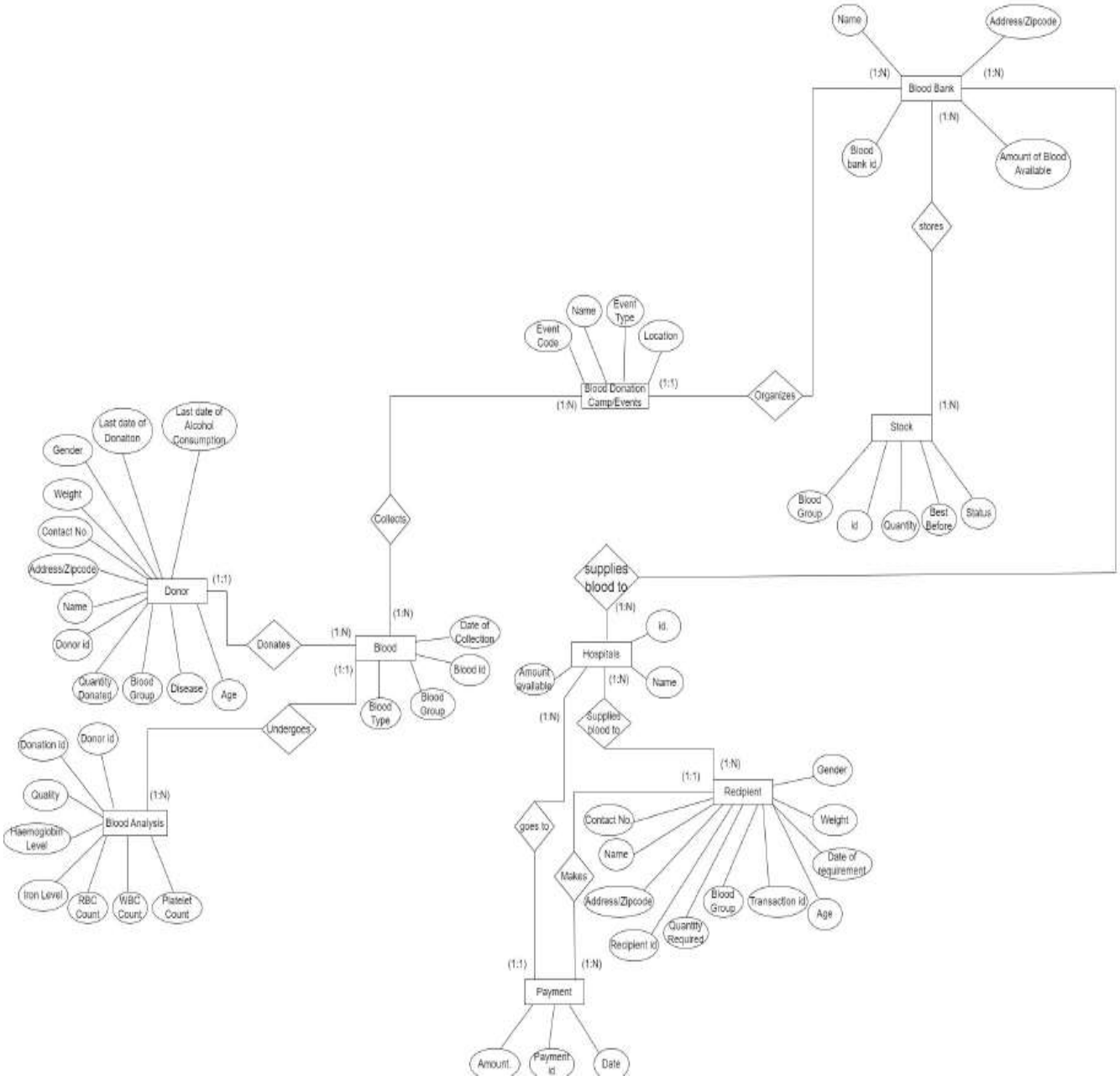
The primary purpose of this study is to develop and implement a relational database which serves as an easy source of information about the extensive network of blood banks across India in order to simplify and automate the process of searching blood. This can be achieved by designing a web based application which stores, processes, retrieves and analyzes the information concerned with administrative and inventory management involved in the entire chain of events. This implementation minimizes the cost and time of the labor invested in accessing and storing the vast information by 50%. The database modeling was completed by gathering previous data from hospitals and Blood banks as per their respective locations, that were traced by zipcodes. The Conceptual Models (EER Diagram and the UML Class Diagram) were modeled, followed by mapping of conceptual model to a logical model (relational) with the required primary and foreign keys. This database was then implemented in MySQL using the SQL Workbench 8.0 software, with a portion of it implemented in NoSQL using the MongoDB Shell Script. The databases were successfully created and connections to Python were implemented to enhance the analytics capabilities. Several Visualizations were created using Python to track the available amount of blood with each hospital and to check the current stock in Blood Bank. Using donor and recipient information from the database, both of them are mapped to each other as per the requirement.This improved the old system and also increased the efficiency of database.It also gives public an opportunity to get more information about donors and vice versa.

I. Introduction

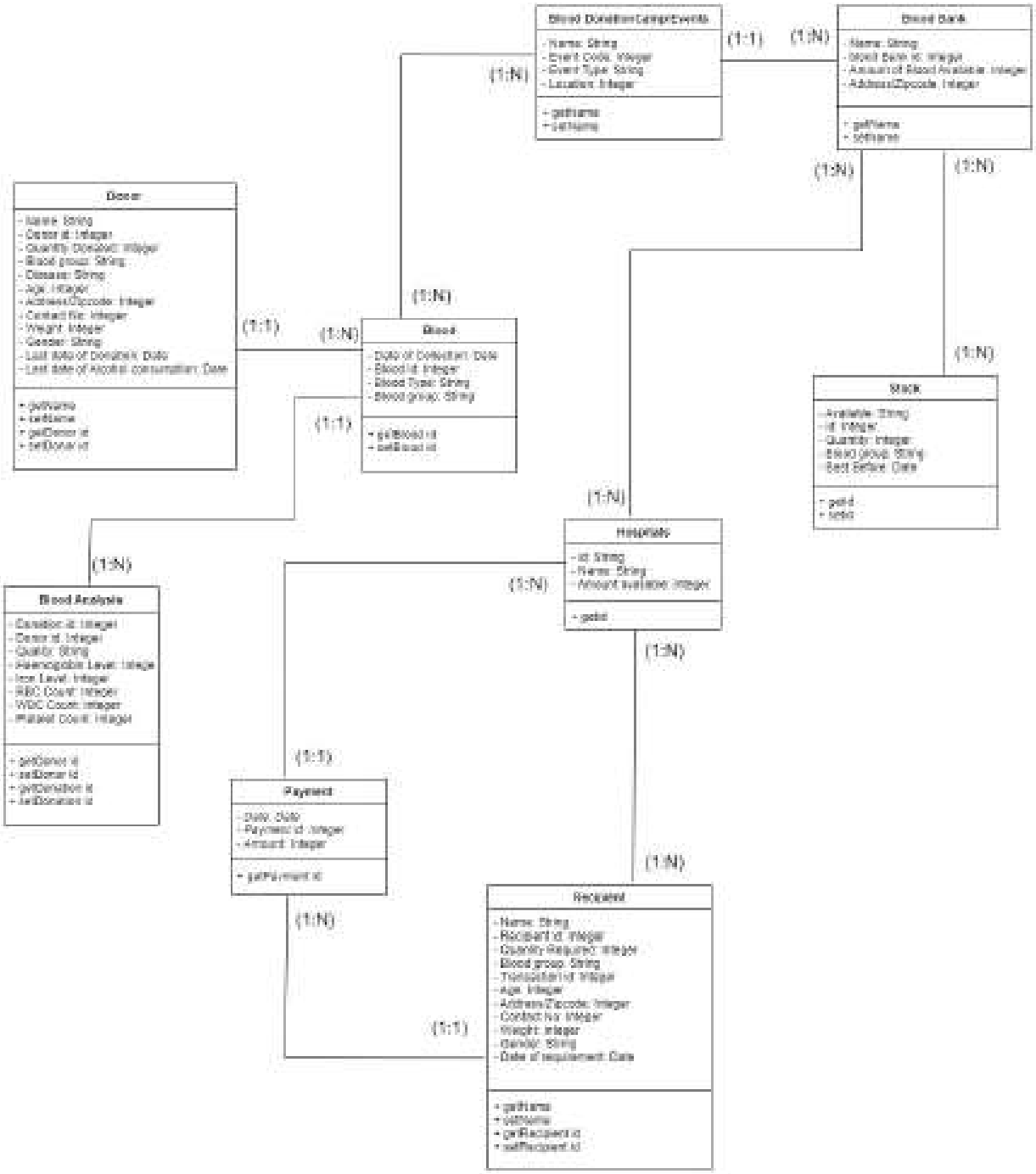
With the advent of Technology, the world is taking rapid strides in the Healthcare sector to provide efficient and cost-effective solutions to the existing problems. Blood donation is a process wherein a donor voluntarily donates his blood, which could be used for future transfusions at hospitals for medical treatments. Donation mainly categorizes into 2 types of viz Whole blood donation (blood is drawn directly from the body) and donation of specific components of the blood such as red blood cells, white blood cells, plasma, and platelets. Blood banks often participate in the process of collecting bloods and other procedures such as managing stocks, approving blood, approving blood requests, and updating donor information.Furthermore, they collaborate with the hospitals and the donors to provide the collected blood to the hospitals as per their requirement. The COVID-19 pandemic in a way, gave a harsh reality check to the existing Medicare system in India. The prolonged issues of many patients in backlog due to inadequate number of beds in the hospital got highlighted during the second wave of COVID-19 when numerous people across the country suffered as they could not get the desired blood in time for their respective treatments. Also, the ratio of number of doctors to the number of patients is quite low. Hence, Blood banks maintain an extensive network of blood donors and recipients to fulfill the medical requirement when the need arises. To ensure quick and easy management for the same, it is imperative to design a web-based application which stores, processes,retrieves and analyzes the information concerned with administrative and inventory management involved in the entire chain of events.

II. Conceptual Data Modeling

(1) EER Diagram



(2) UML Class Diagram



III. Mapping Conceptual Model to Relational Model

Donor (Donor_id, first_Name, Last_Name, Zip code,Gender,Weight,Last_Donation_date, Last_alcohol_consumption, Quantity_donated, Donor_Blood_Group, Disease, Age)

Here, Donor id is the primary key

Donor: Donor id in relation to Donor: NULL not allowed, on delete/update cascade.

Analysis (Donation id, Donor id, Quality, Haemoglobin Level, Iron Level, RBC Count, WBC Count, Platelet Count, Blood id)

Here, Donation id is the primary key

Donor: Donor id in relation to Donor: NULL not allowed, on delete/update cascade.

Blood (Blood Type, Blood Group, Collection date, Blood id, Donor id, Event Code)

Here, Blood id is the primary key.

Donor: Donor id in relation to Donor: NULL not allowed, on delete/update cascade.

Events: Event Code in relation to Events: NULL not allowed, on delete/update cascade.

Blood banks: Blood bank id in relation to Blood bank: NULL not allowed, on delete/update cascade.

Events (Event Code, Event Name, Event Type, Location, Blood bank id)

Here, Event Code is the primary key.

Events: Event Code in relation to Events: NULL not allowed, on delete/update cascade.

Blood banks: Blood bank id in relation to Blood bank: NULL not allowed, on delete/update cascade.

Hospitals (Hospital ID, Hospital Name,Amount available,blood bank id, Payment id,recipient id)

Here, Hospital ID is the primary key.

Blood banks: Blood bank id in relation to Blood bank: NULL not allowed, on delete/update cascade.

Payment (Payment id, Amount, Payment date, Recipient id)

Here, Payment id is the primary key.

Recipient: Recipient id in relation to Recipient: NULL not allowed, on delete/update cascade.

Recipient (Recipient id, Transaction id, first_Name, Last_Name, Zip code, Gender, Weight, Date of requirements, Quantity required, Blood Group, Age, Contact No)

Here, Recipient id and Transaction id are the primary keys

Blood Bank (Blood bank id, Blood bank Name, Zip code, Quantity Available)

Blood bank: Blood bank id in relation to Blood bank: NULL not allowed, on delete/update cascade.

Stocks (Stock id, Blood Group, Blood available,Status, best before, Blood bank id)

Here, Blood Group is the primary key.

IV. Implementation of Relational Model via MySQL and NoSQL

My SQL Implementation

The database was created in MySQL workbench using the basic CREATE statements. After specifying the attributes the Primary Key constraints, Null value constraints were given.

Queries and their Outputs

1) **Present the Recipients (first name, last name, recipient id) who need blood group A+ (Simple Queries)**

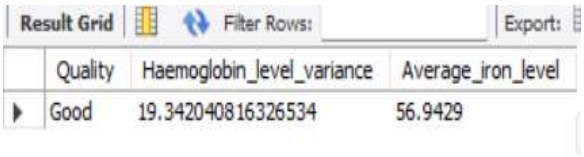
```
Select First_name,Last_name,
Recipient_id
from recipient
where Recipient_blood_group='A+' ;
```

Result Grid | Filter Rows:

First_name	Last_name	Recipient_id
Victor	Hamill	512873
Sierra	Keeling	576171
Emmett	O'Conner	3960149
Laury	Towne	23492237
Raven	Wolf	46330176
Vallie	Mosciski	689069251

2) **Calculate the variance in Haemoglobin level and the Iron level of samples which are deemed Good (Queries with Aggregate Functions)**

```
select Quality,
variance(Haemoglobin_level)
as Haemoglobin_level_variance,
avg(Iron_level)
as Average_iron_levelfrom analysis
where Quality='Good';
```

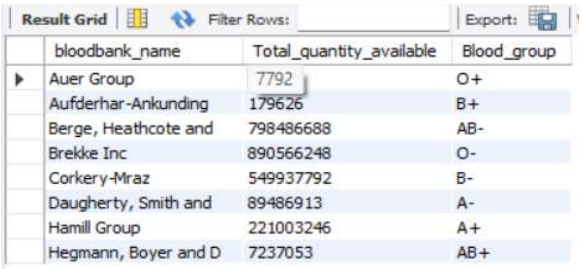


Result Grid | Filter Rows: | Export:

Quality	Haemoglobin_level_variance	Average_iron_level
Good	19.342040816326534	56.9429

3) **Find the Blood Bank Names and the Amount of Blood (of different blood groups) they have in stock (Queries with Group By, JOINS)**

```
Select b.bloodbank_name,
sum(s.Blood_available)
asTotal_quantity_available,s.Blood_gro
up  from Blood Banks b JOIN stocks s
ON s.Bloodbank_id=b.Bloodbank_id
Group by Blood_group
```



Result Grid | Filter Rows: | Export:

bloodbank_name	Total_quantity_available	Blood_group
Auer Group	7792	O+
Aufderhar-Ankunding	179626	B+
Berge, Heathcote and	798486688	AB-
Brekke Inc	890566248	O-
Corkery-Mraz	549937792	B-
Daugherty, Smith and	89486913	A-
Hamill Group	221003246	A+
Hegmann, Boyer and D	7237053	AB+

4) **List out the recipient names, hospital names and the corresponding amount they paid to the hospital in Descending order (Queries with Order By, Multiple JOINS)**

```
select r.recipient_id,r.first_name,
r.last_name,p.amount,
h.Hospital_name
from recipient r
JOIN payment p on
r.recipient_id=p.recipient_id JOIN
hospitals h on
p.payment_id=h.payment_id
order by p.amount desc;
```

Result Grid | Filter Rows: | Export: | Wrap

recipient_id	first_name	last_name	amount	Hospital_name
53717026	Annabell	Romaguera	714732273	culpa
7266675	Sarah	Anderson	253711825	quibusdam
23	Alena	Rogahn	244853510	aperiam
322	Madisyn	Tromp	134177907	deserunt
5161665	Madisyn	Shanahan	96656013	nulla
181291	Reva	Padberg	53710525	quaerat
9020	Nyasia	Smith	43646210	harum
3806	Madonna	Nitzsche	40344857	veritatis
5	Katharina	Schulist	32394568	labore
647	Jennyfer	Kub	31827256	impedit
8368882	Reynold	Stehr	30801163	tempora
542895913	Yasmine	Schmidt	7656680	nam

5) **Find out the Blood samples which were collected from Universities and also list the event code (Nested Queries)**

```
select Blood_id,Event_code
from blood
where Event_code
IN
(select Event_code from events
where event_type='University');
```

Result Grid | Filter Rows:

Blood_id	Event_code
473725954	3
972	7
29498519	9
934	98
41408	77457
748384570	2899140
920774388	6420765
5	18261995
5217	43790016
1417	51396725
93	855754859

6) **Find the details of the recipient, hospital from which he receives blood and the blood bank which supplies to the hospital, and has made a payment in excess of 50,000(Correlated Queries)**

```
select
r.First_name,r.Last_name,b.bloodbank_i
d,h.hospital_id
from Blood Banks b
JOIN hospitals h on
b.bloodbank_id=h.bloodbank_idJOIN
recipient r on
h.recipient_id=r.recipient_id where
50000 < (select amount from payment p where h.recipient_id=p.recipient_id);
```

Result Grid | Filter Rows:

First_name	Last_name	bloodbank_id	hospital_id
Clark	Prohaska	7340	90220
Laury	Towne	99381055	24
Reynold	Stehr	51508372	84743
Julia	McLaughlin	4802	4906
Madisyn	Tromp	9473	65
Nyasia	Smith	678783	39
Jeff	Considine	7238	45228
Blanche	Robel	4565846	3150374
Alena	Rogahn	62	845719526
Reva	Padberg	1432861	55169
Toy	Bernier	81	257113
Jennyfer	Kub	324944	733167575

7) **Retrieve the Name of Blood Banks that conducted events with a Company (Queries with EXISTS)**

```
select Bloodbank_name
from Blood Banks b
WHERE EXISTS
(select *
from events e
where
b.bloodbank_id=e.bloodbank_id
and e.event_type='Company');
```

Result Grid | Filter

Bloodbank_name
Auer Group
Brekke Inc
Feil and Sons
Hamill Group
Hegmann, Boyer and D
Hermann-Schimmel
Howe Group
Kohler-Lueilwitz
Kuhlman, Littel and
Lesch, Bernhard and
Lindgren-McKenzie
Lubowitz LLC

NoSQL Implementation

1) db.Events.aggregate([{"$group" : {_id:"$Event_type", count:{$sum:1}}}])

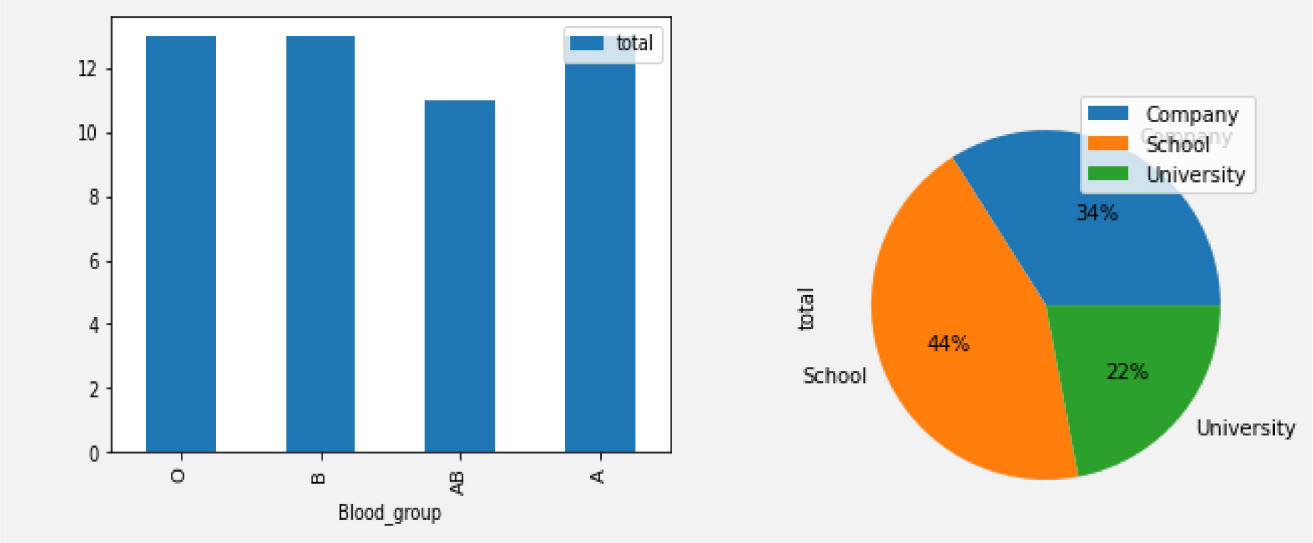
```
> db.Events.aggregate([{"$group" : {_id:"$Event_type", count:{$sum:1}}}])
< { _id: 'School', count: 22 }
  { _id: 'University', count: 11 }
  { _id: 'Company', count: 17 }
```

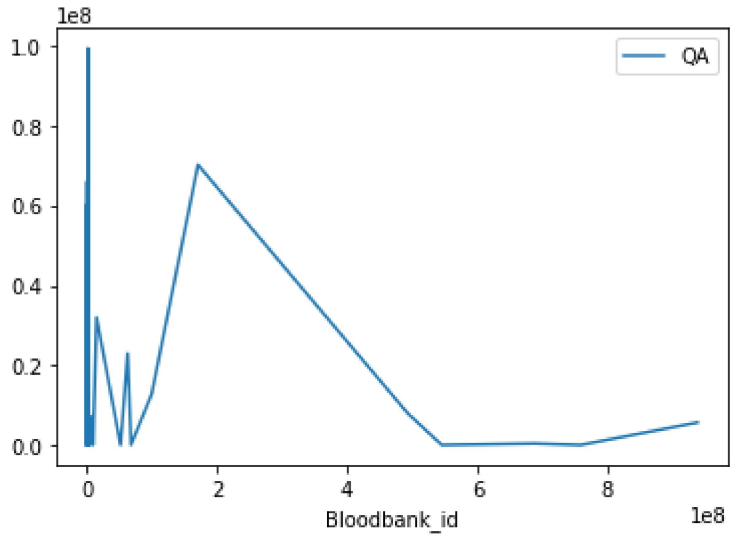
2) db.Analysis.findOne({ Quality : "Good" })

```
> db.Analysis.findOne({ Quality : "Good" })
< { _id: ObjectId("6387deef6156b0644764996c"),
    Donation_id: 0,
    Quality: 'Good',
    Haemoglobin_level: 3,
    Iron_level: 32,
    RBC_Count: 24,
    WBC_Count: 86,
    Platelet_Count: 2,
    Donor_id: 0,
    Blood_id: 0 }
```

V. Database Access via Python

The database is accessed using Python and Visualization of analyzed data is shown below The connection of MySQL to Python is done using mysql.connector, followed by cursor.execute to run and fetch all from query, followed by converting the list into a dataframe using pandas library and using matplotlib to plot the graphs for the analytics.





VI. Summary and Recommendation

Blood bank management system is a web based application, which is ready to use and implement across the country such that people can use it to access and trach information regarding the extensive network of Blood banks and Hospitals. Recipients are likely to find their respective donors more quickly than previous time. This will ensure easy access of information for all. From the database, people who have consumed alcohol during the last 15 days or donors who have a chronic disease are deemed ineligible to donate as contaminated blood is not feasible to donate. Further more, we can develop real time tracking of information regarding the availability of Blood in different hospitals or Blood banks. A series of customization features can also be added in order to get hands on insights of the already existing data and make predictions for the future.