

Blood Donation Management System



December 16, 2023

Group Number: 37

Phase 1

Members

|  |  |
| --- | --- |
| Name | ID |
| Hassain Alsayhah | 202028180 |
| Mohammed Almubarak | 202024880 |
| Feras Alhasmi | 202031280 |

Table of Contents

[Phase 1 work: 2](#_Toc153643955)

[EER Model: 2](#_Toc153643956)

[Relational Schema: 3](#_Toc153643957)

[Assumptions: 4](#_Toc153643958)

[Semantic requirements that cannot be captured in the EER model: 5](#_Toc153643959)

[Team members Contribution in Phase 1 6](#_Toc153643960)

[Phase 2 work: 7](#_Toc153643961)

[How we implemented Phase 2: 7](#_Toc153643962)

[tools and resources that you used: 16](#_Toc153643963)

[problems that we faced: 17](#_Toc153643964)

[What we learned for the project: 17](#_Toc153643965)

[percentage completion of each required operation: 18](#_Toc153643966)

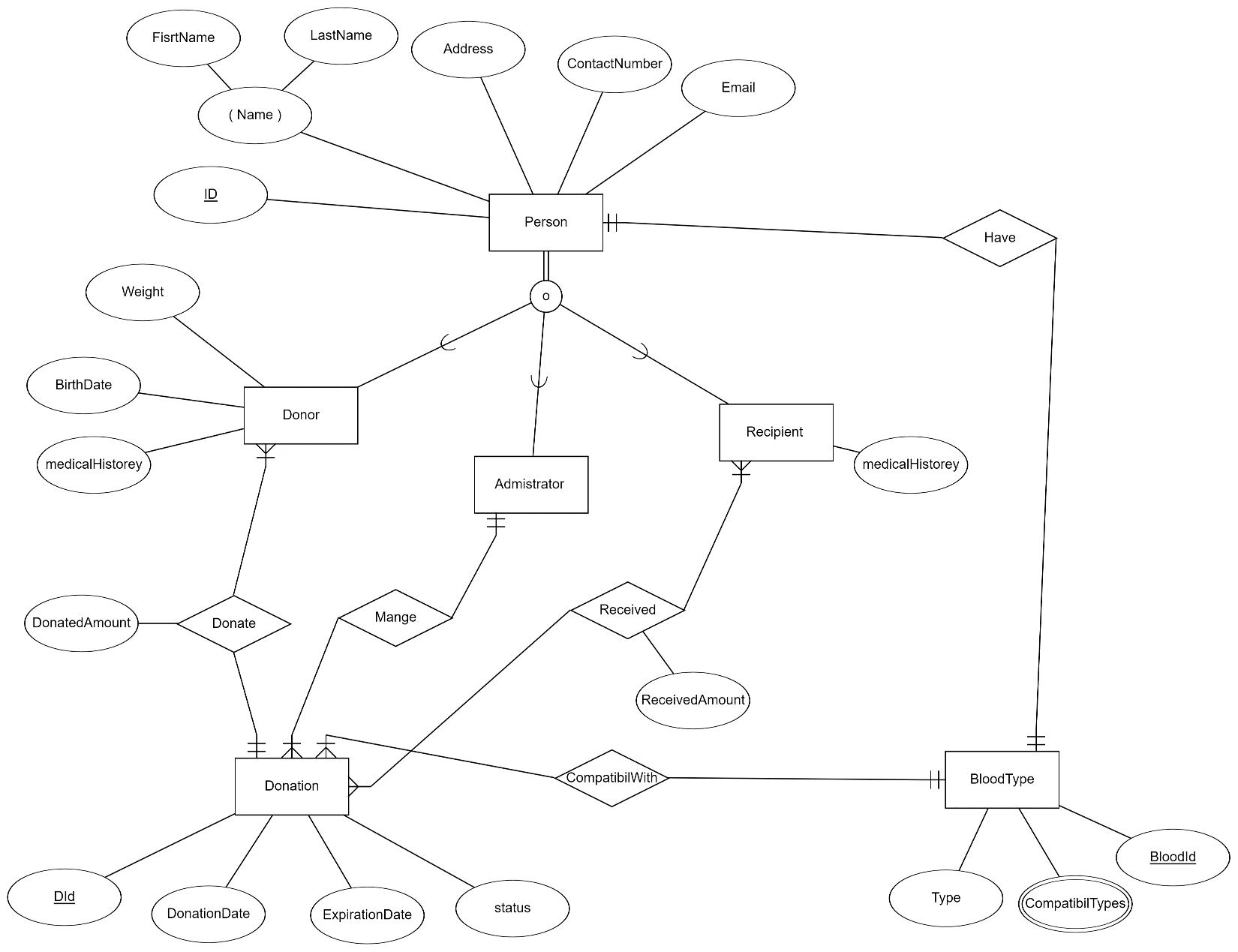
[extra things done: 19](#_Toc153643967)

[tasks done by each group member: 19](#_Toc153643968)

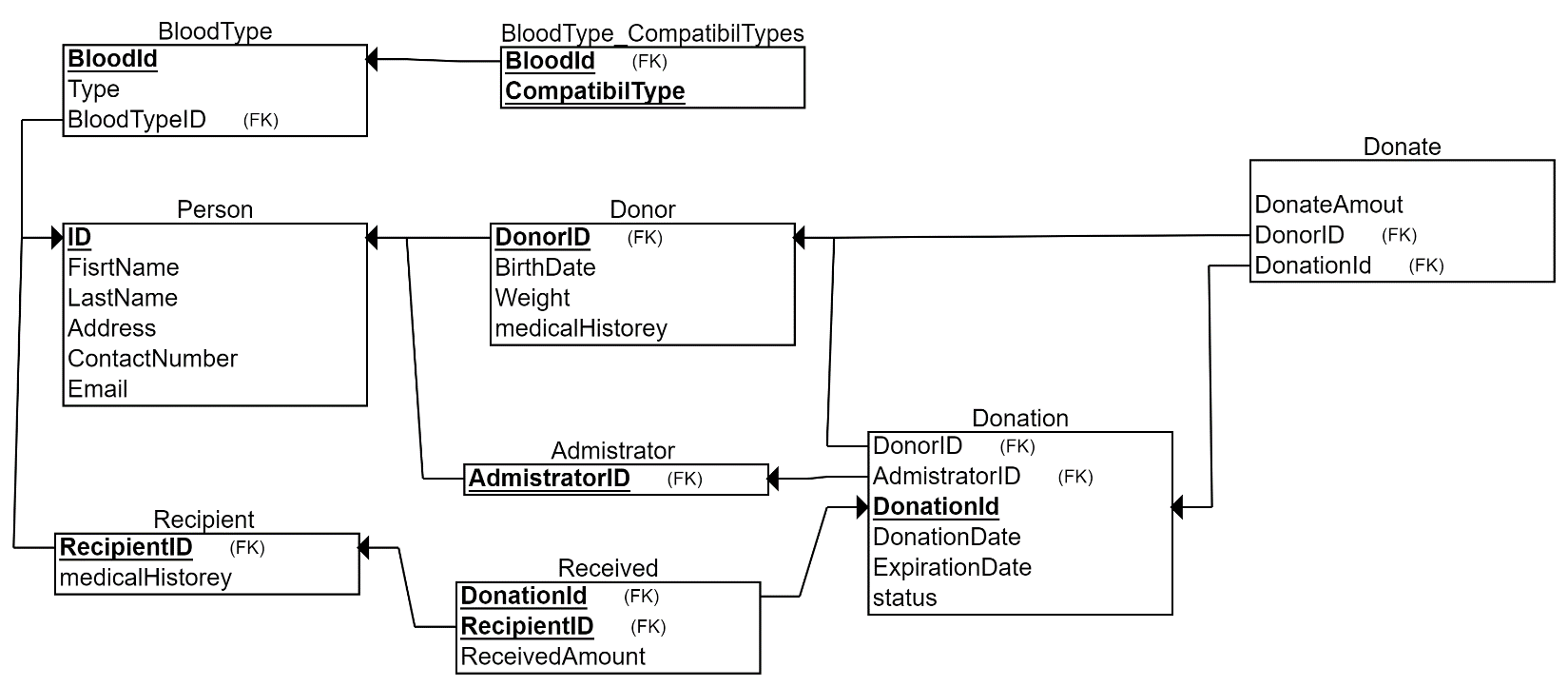
[Suggestions to improve ICS 321 future projects: 19](#_Toc153643969)

# Phase 1 work:

## EER Model:



## Relational Schema:



## Assumptions:

1- Each person has a unique ID.

2- Each person has a first name and a last name.

3- Each person has a blood type

4- A person can be a donor, a recipient, or an administrator at the same time.

5- A Donor has a weight.

6- Each donation has a donated amount

7- The compatibility between the recipient and the donor's blood types can be determined by referring to a blood type table, which identifies the compatible types for transfusion validation.

8- Each donation has a donation date and Expiration Date.

9- Each donation has a status of either passed or failed.

10- The administrator can manage many donations, collecting, storing, distributing, etc.

11- The recipient has a medical history.

12- Each donation has a received amount.

13- Each blood type has a unique ID.

14- Each blood type has a compatible blood type or more.

## Semantic requirements that cannot be captured in the EER model:

1. Each donor must be:

At least 17 years old

Weigh at least 114 lbs.

Be free of major diseases.

1. Blood collection drives scheduling every 3 months.
2. Monitoring expiration dates of stored blood.
3. Incident tracking (capturing details of process failures).
4. Usage trend reports.

## Team members Contribution in Phase 1

|  |  |
| --- | --- |
| Name | Contribution |
| Hassain Alsayhah | 33.34 |
| Mohammed Almubarak | 33.33 |
| Feras Alhasmi | 33.33 |

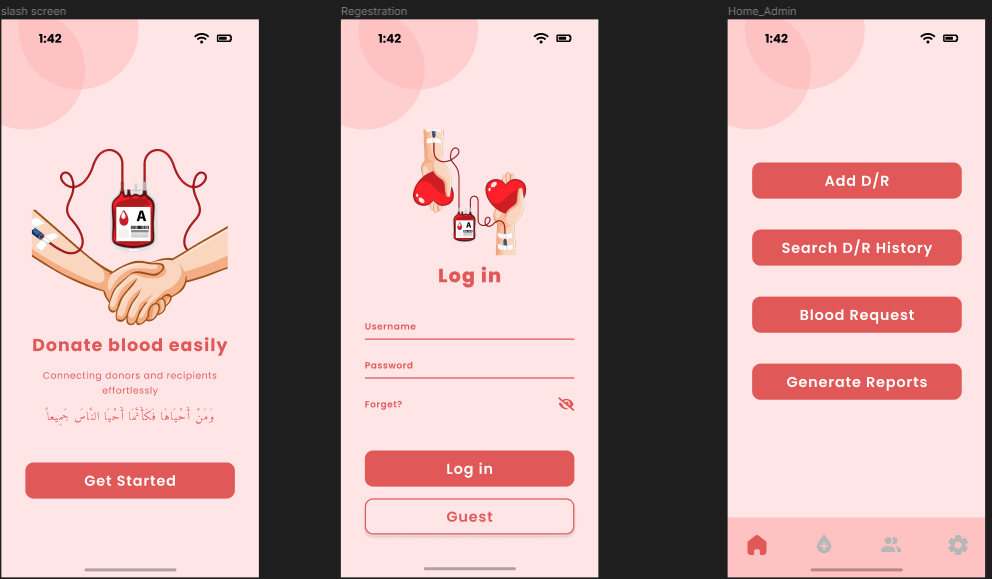
# Phase 2 work:

## Repository link:

<https://github.com/MoAlmubarak/ICS321-Database_project.git>

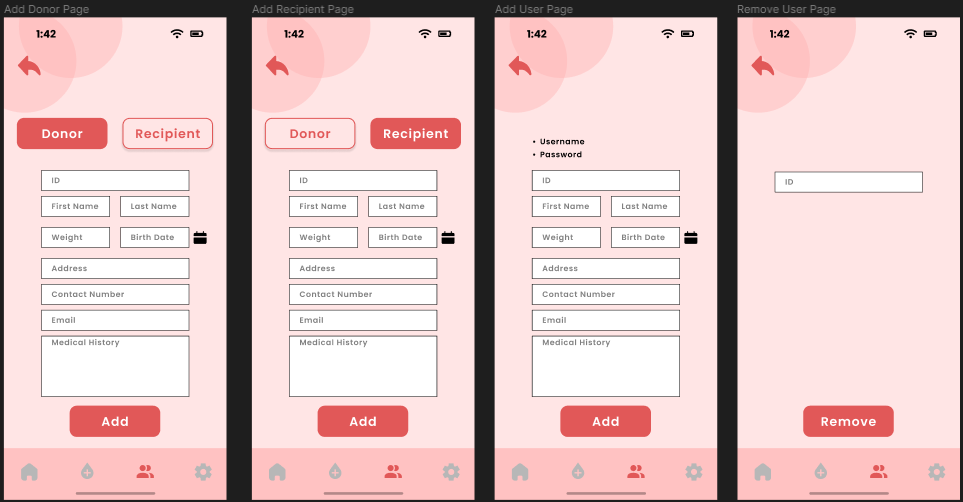
## How we implemented Phase 2:

For brainstorming the user interface and the user experience we used Figma you can see the work that we did in Figma below:



Screens screenshot of a phone

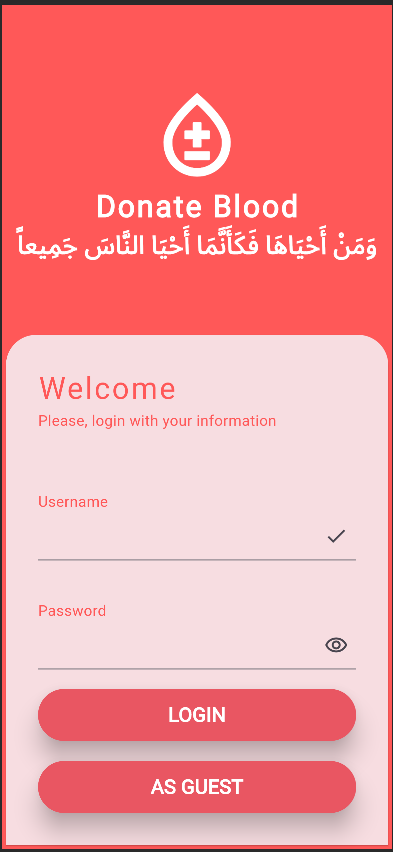
Description automatically generated



Screens screenshot of a screenshot of a phone

Description automatically generated

Below are the actual screens for our project



|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |

For the implementation of phase 2 we have used **Flutter framework** for the interface and for the database we used **Sqflite** which is a Flutter plugin for **SQLite.**

## tools and resources that you used:

* Figma (for brainstorming)
* Flutter (for development)
* Git and GitHub (for controlling the code versions)
* Teams (for communication)

## problems that we faced:

we have faced many problems during the journey of developing the project below I will highlight some of them:

* Sharing and controlling any code that we wrote we have solved this problem by using Git and GitHub
* Combining them database code with the frontend work
* Ambiguity in the requirements of the project we solved this problem by making our own assumptions.
* Recreating the exact scenes from Figma. We tried to do our best to be as close as possible to the Figma design.

## What we learned for the project:

We have learned a lot of things during developing the project. Below are some of them.

* We improved our experience of working with teams.
* Learnd a lot about database management and SQLite.
* Learned how to use version control systems like Git and how to use GitHub with it.
* Engaged and increased our knowledge in Figma.
* Learned a new framework for app development which is Flutter.

## percentage completion of each required operation:

|  |  |
| --- | --- |
| Operation | Percentage of completion |
| Functions of an Administration/Employee | |
| Add/Remove/Edit Donor/Recipient information. | 100 |
| Search for donor and recipient history. | 100 |
| Add/Remove/Edit system user information. | 100 |
| Process request for Blood for a recipient. | 100 |
| Initiate Blood Collection Drive in a given period. | 100 |
| Generate dashboards/reports using your system. | 80 |
| Sending appropriate notifications through email etc. | 70 |
| Functions **of a Donor/Recepients** | |
| Search for own (as donor or recipient) history. | 100 |
| Agree for blood donation/receiving (as donor or recipient). | 100 |
| Update personal information of Donor and Recipient including medical history. This should then be approved by system administrator before actual database update. | 100 |
| Do payments as charges only for receiving blood. | -------- |
| General Function (for all users) | |
| Login and Logout | 100 |
| Browse as Guest | 100 |
| Reports (as system output) | |
| List of all blood donations received in a week or a month. | 100 |
| List the aggregated amount available for each blood type. | 100 |
| List all Collection Drive and total blood collected during each drive. | 100 |
| All Payments that have been confirmed as completed. | -------------- |

## extra things done:

We tried to implement the best practices of software design in writing the code.

We allowed the user to make a request for blood donation.

## tasks done by each group member:

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Mohammed Almubarak | Feras Alhasmi | Hassain Alsayhah |
| Implemented the database. |  | ✔️ |  |
| Implemented the frontend. | ✔️ |  |  |
| Connected the database to the frontend. |  |  | ✔️ |

## Suggestions to improve ICS 321 future projects:

* Include a lab for the course.
* Give assignments as mini projects rather than a pig project at the end.
* Make the requirements of the project as detailed as possible.