MIDCARE SYSTEM

Project Team Members

Names	Role
Ahmed Ali	FE + BE + Mobile + DevOps
Hassan Alaa	Frontend Development
Mohamad Al-Kafrawy	Frontend Development
Mahmoud Abdel-Latif	DevOps
Aly Eldin Ahmed	Backend Development
Mohamed Adel	Backend Development

Agenda

1. Introduction
2. Project Description
3. System Features
4. Technology Stack
5. Interface Requirements
6. Project Approach, Timeline

Introduction

The Problem Statement

Current Problems

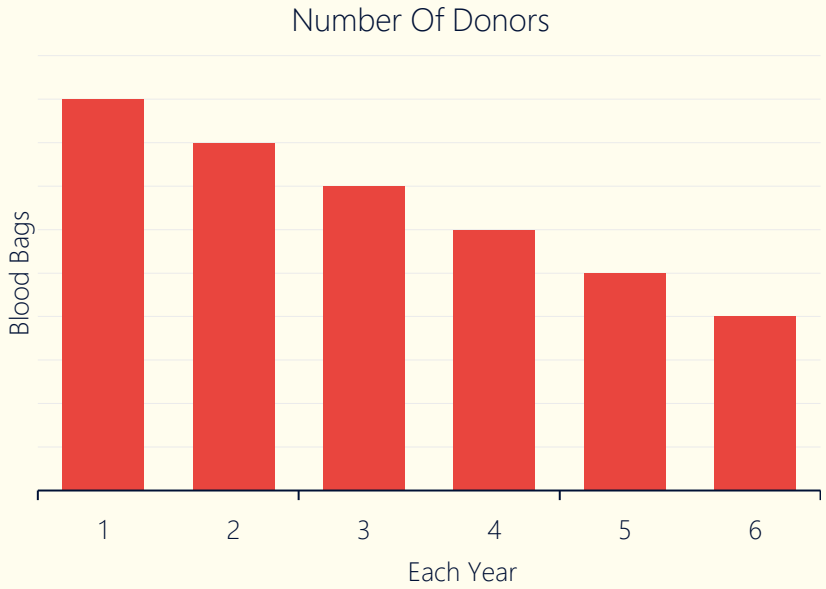
بيان إحصائي بأعداد الحملات وأكياس الدم المجمعة منذ عام 1999 وحتى شهر يوليو عام 2019			
المسلسل	السنة	عدد أكياس الدم	عدد الحملات خلال العام
1	1999	12847	216
2	2000	15016	277
3	2001	13734	235
4	2002	19752	239
5	2003	18530	257

Reference: https://www.afnci.org.eg/?page_id=120

The Problem Statement

Main Problems

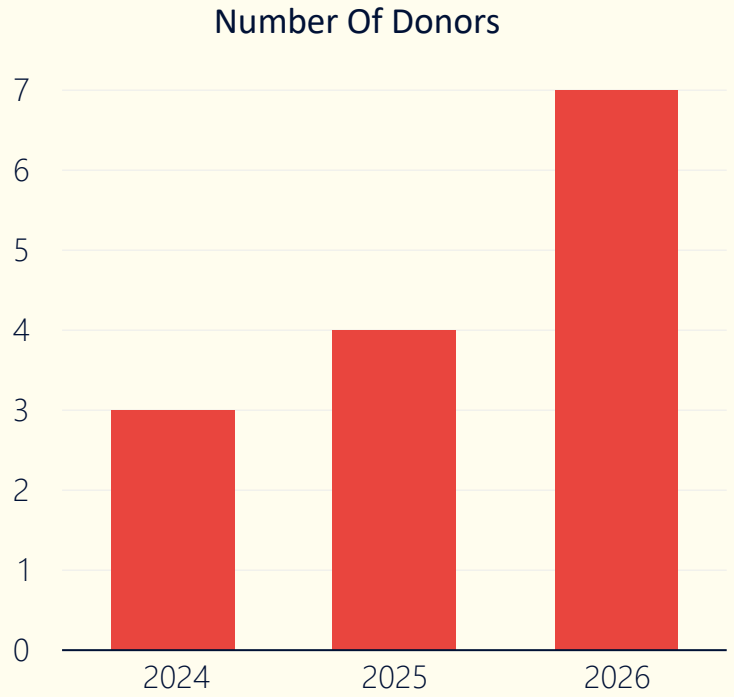
- Declining number of blood donors despite increasing demand
- Difficulties in matching blood supply with demand
- limited engagement of potential donors.
- The lack of user-friendly platforms discourages potential donors from participating actively in donation campaigns
- increase the problem of blood shortages in healthcare institutions.



Digital Solution Future Impact

Application aimed at solving the problems

- Increase in the number of blood donors due to improved accessibility and awareness
- Enhanced engagement of donors through user-friendly features and incentives
- Timely and efficient matching of blood supply with demand, leading to reduced shortages.
- Improved coordination and communication between donors, recipients, and healthcare institutions.



Three questions represent the introduction of the system presentation

1

What is the purpose of the application ?

2

What is the application system services?

3

Who is the intended audience ?

What is the purpose of the application ?

BBank “MIDCARE” system is a Blood Bank System Application and component in the broader healthcare and emergency response ecosystem. It acts as a bridge between donors, healthcare institutions, and individuals in need, facilitating a streamlined and efficient blood donation process. The application fits into the larger architecture by:

Purpose

Blood Bank System is
To develop a robust
blood donation application

Aims

Address challenges and
opportunities within the
blood donation ecosystem

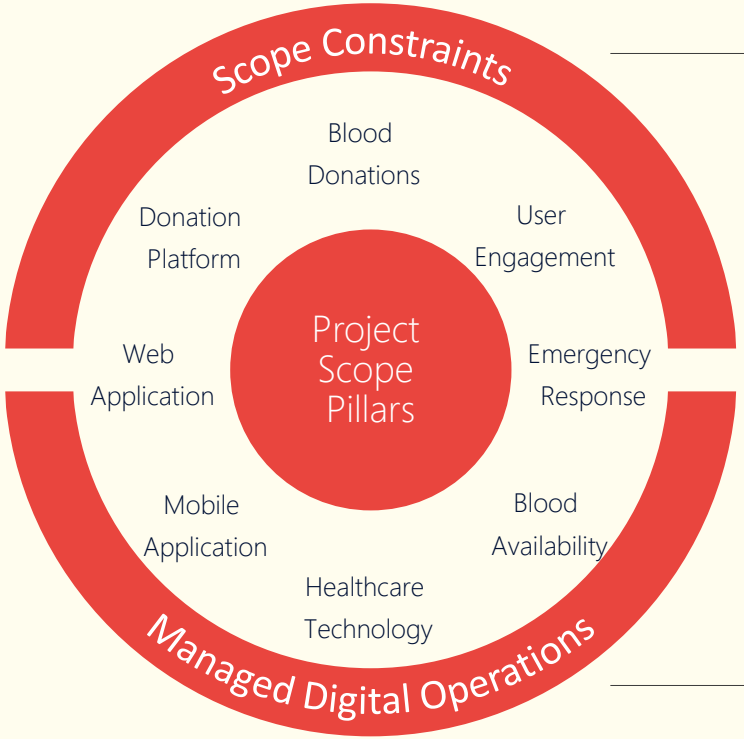
Goal

Improve the overall
blood donation Experience
for donors and recipients.

Application System Services

The Blood Bank System aims to increase the number of blood donations, enhance user engagement and retention, and improve emergency response and blood availability. The project falls within the technical field of healthcare technology and mobile & web application development. The end result will be the successful creation and deployment of a comprehensive blood donation application, serving as a platform connecting blood donors, recipients, and healthcare institutions.

Application System Services



Blood Donations
Encouraging donation participation

Emergency Response
Swift reaction during critical situations

Blood Availability
Ensuring an ample supply for medical needs

User Engagement
Ensuring active involvement

Audience and Expansion Plan

Audience:

Primary: Individuals, hospitals in Egypt.

Expansion: Middle East, Europe, USA.

Expansion Plan:

Start Point: Egypt

Next Step: Middle East

Final Target: Europe, USA

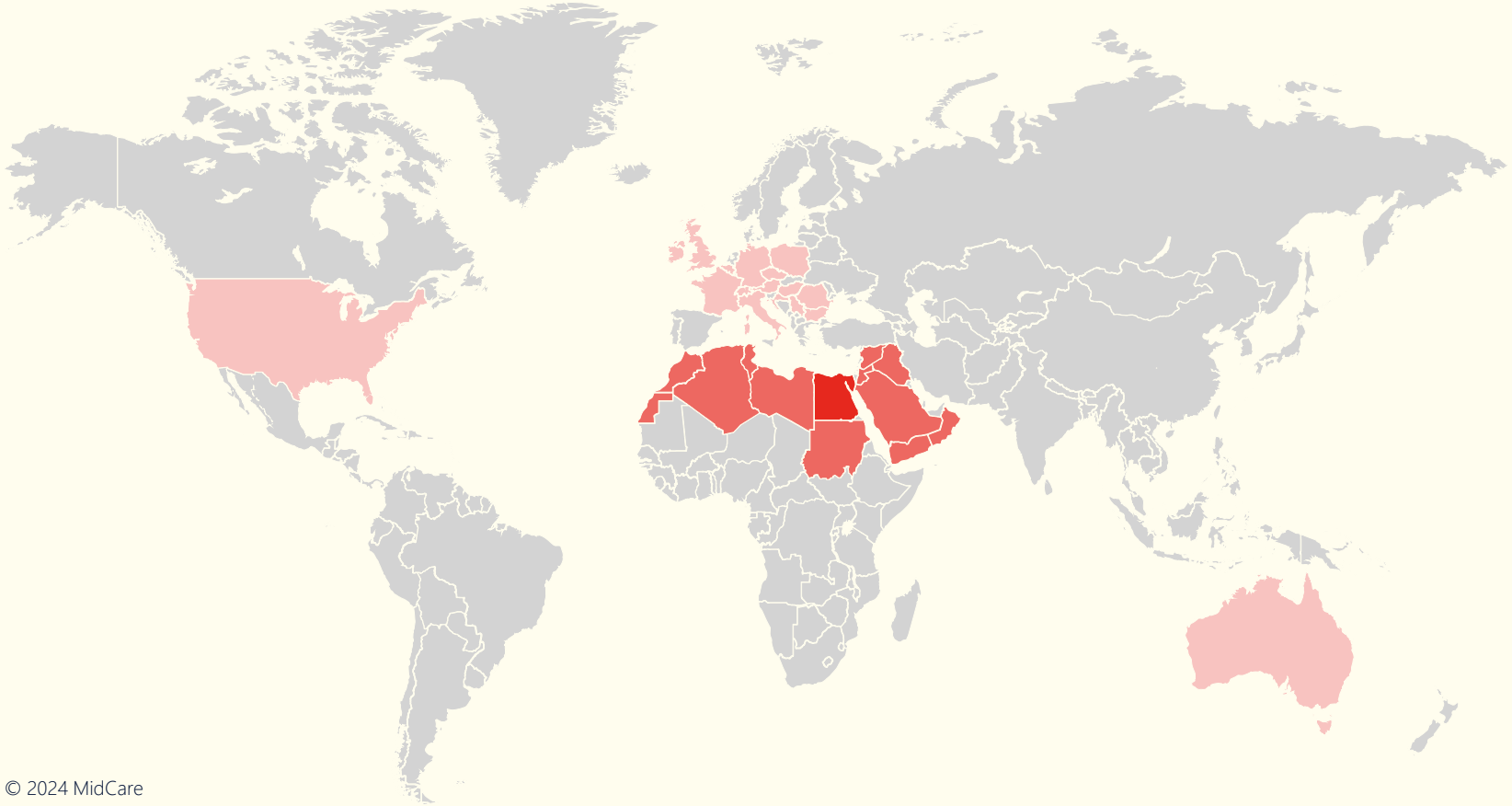
Approach:

Cultural Sensitivity: Prioritize local customs and languages.

Regulatory Compliance: Adhere to regional healthcare regulations.

Global Impact: Enhance emergency response and blood availability worldwide.

Priority Targeted Regions and audience



Project Description

Three questions represent the introduction of the system presentation

1

What is the main functions of the application ?

2

Who is the System Stockholders?

Main functions

What are the main features and functionalities of your software ?

1. User Navigation
2. Donation Ticket Management
3. Monetary Donations
4. Emergency Notifications
5. Donor Profiles
6. Appointment Scheduling

What technologies are utilized for implementing these functions ?

1. Geolocation Services
2. Database Management
3. Real-Time Notifications
4. Payment Gateway Integration

System Functions

User Navigation: Utilizing map integration to help users locate nearby hospitals and blood donation centers effortlessly.

Donation Ticket Management: Issuing digital donation tickets after blood donation, allowing users to store, retrieve, and transfer tickets securely.

Monetary Donations: Enabling users to contribute financially to specific cases requiring blood donation, seamlessly integrating with payment methods.

System Functions

Emergency Notifications: Allowing hospitals to broadcast emergency cases, immediately notifying users in the affected area.

Donor Profiles: Providing a platform for users to create, manage, and share their donor profiles, including personal information, blood type, and medical history

Appointment Scheduling: Allowing users to schedule appointments for blood donation with reminders to prevent missed opportunities.

Who is the System Stockholders

Blood Donors: Individuals interested in contributing blood either regularly or in response to emergency alerts.

Blood Recipients: Individuals in need of blood transfusions due to medical conditions or emergencies.

Healthcare Professionals: Medical personnel, including doctors and nurses, who interact with the system to manage and access donor information.

General Users: Individuals interested in staying informed about blood donation events, emergencies, and contributing financially to specific cases.

Nonfunctional Requirements

Performance Must be interactive	**Encrypted** Passwords Encrypted AES	**Safety** Application Should Provoide Safe Community	**Protection** safeguarded from hacking
Backup Maintained to recover	**Access Control** No unauthorized		

Safety and Security Requirements

- Performance Requirements
- Safety and Security Requirements.
- Proper and Encrypted Login Authentication.
- Protection of Sensitive Information.
- Secure Information Transmission.
- Access Control.
- Data Backup.

Software Quality Attributes

System Major Attributes To Produce Hight Quality Application

Fault Tolerant

Availability

Maintainability

Easy to Use

Reusability

Portability

Performance

Flexibility

Reliability

System Features

System Features

The Application System Features

1

Login/Sign Up Screens

Authentication & Authorization Integration with Backend with Firebase Mail + JWT Then Map to Dashboard

2

Donate Blood

Transfer Blood Ticket From User to Other

3

Purchase Tickets

If blood ticket exist on public hospital or private hospital pay fees of transfer or exchange.

4

My Tickets + Ticket Transfer

Get User Ticket with it's details and transfer ticket to another account

5

Monetary Donations

Donate Money to Users or Hospitals with & without context of blood donation.

6

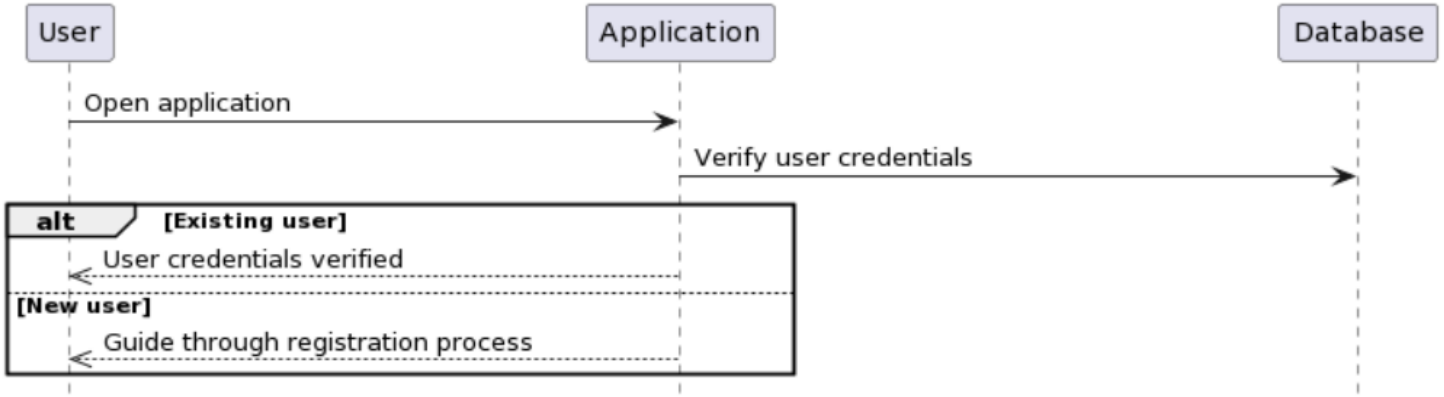
Notifications

Notify users that installed the application if emergency is happening nearby them.

System Features and Sequence

Login/Sign Up Screens

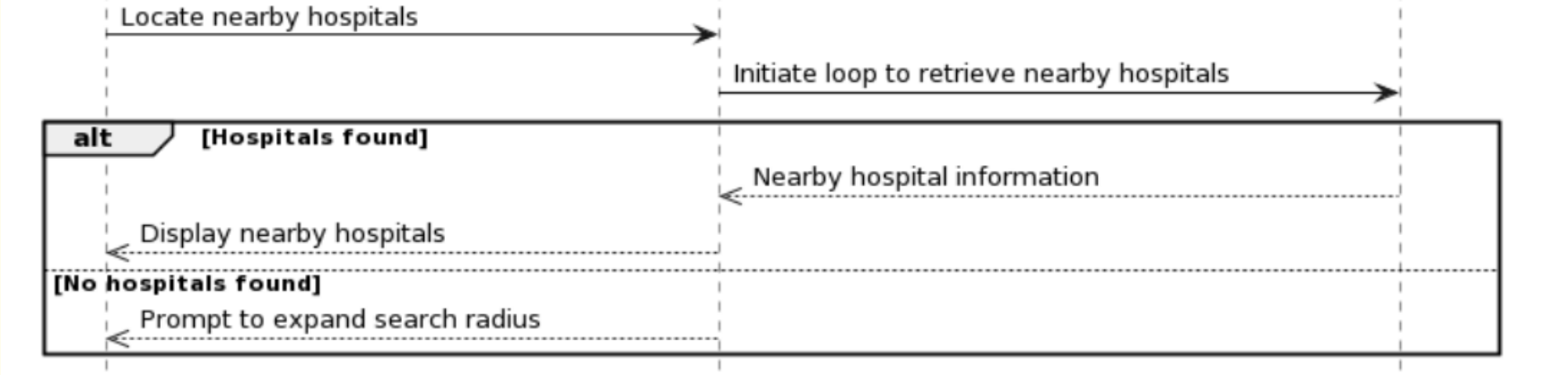
Login and Signup Integration With The Backend Using JWT For Authentication and Role-Base Authorization for Admin User and Regular User. With Integration With Firebase For Email Verification Mail and FCM for Notification.



System Features and Sequence

Dashboard Interface

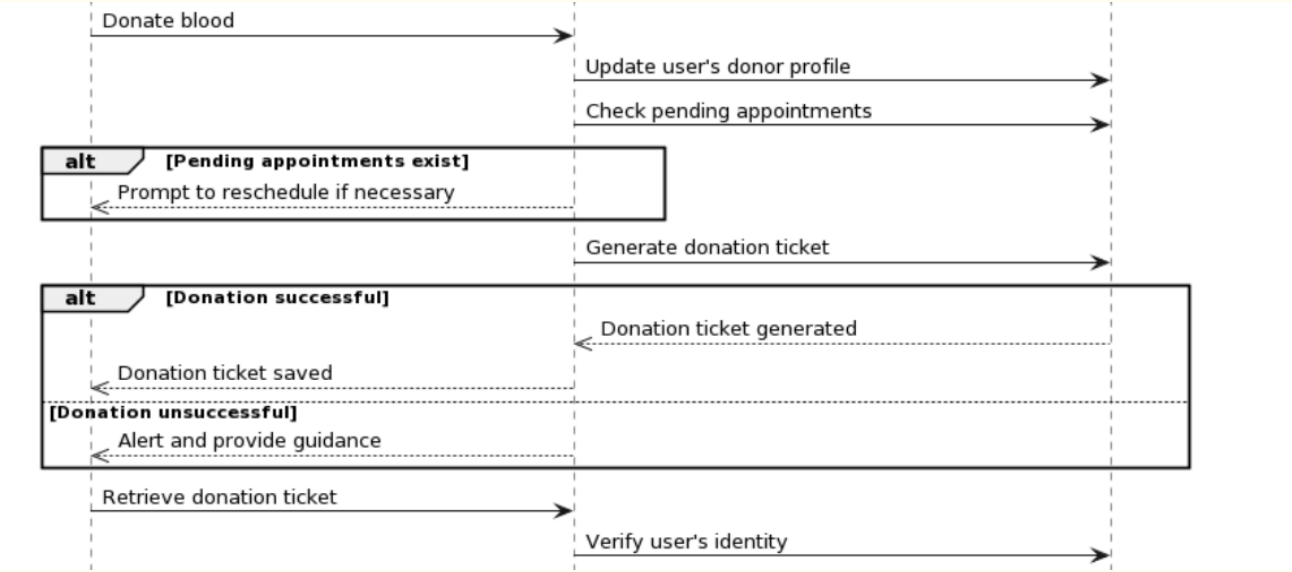
Once User Authentication Verified JWT Token Saved in User Client and Dashboard Load by hitting Dashboard Component Apis and with User Details to be displayed to the User in Main Screen.



System Features and Sequence

Blood Donation

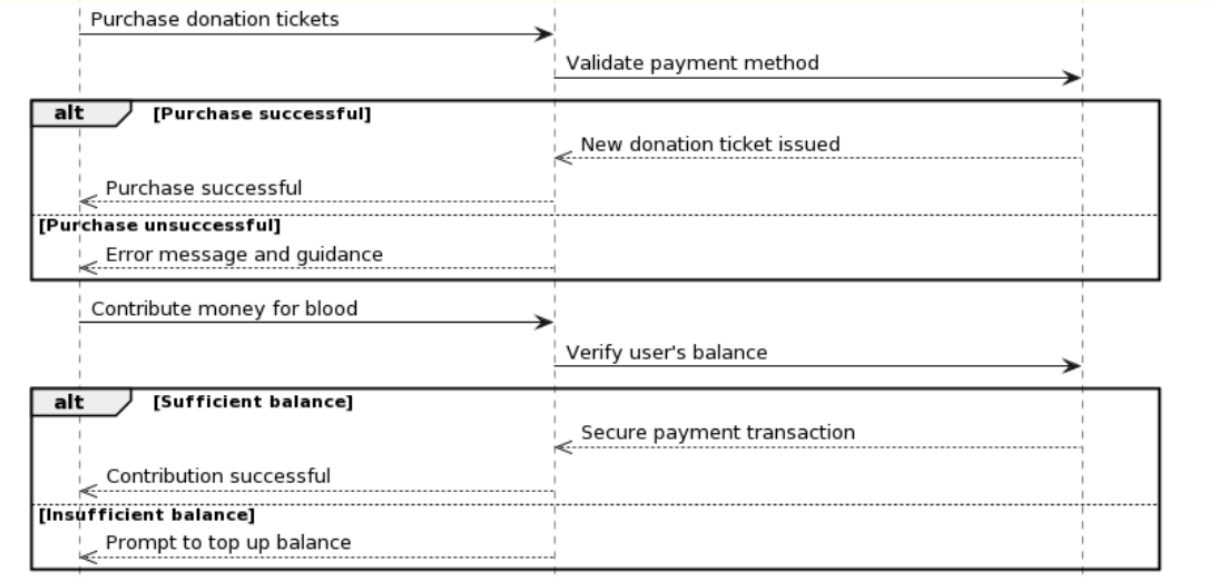
User can get an appointment with hospital for donation then he receive notification once it is possible to go to hospital, after donation the admin end from the hospital create new ticket then he assign It to the user.



System Features and Sequence

Purchase Tickets

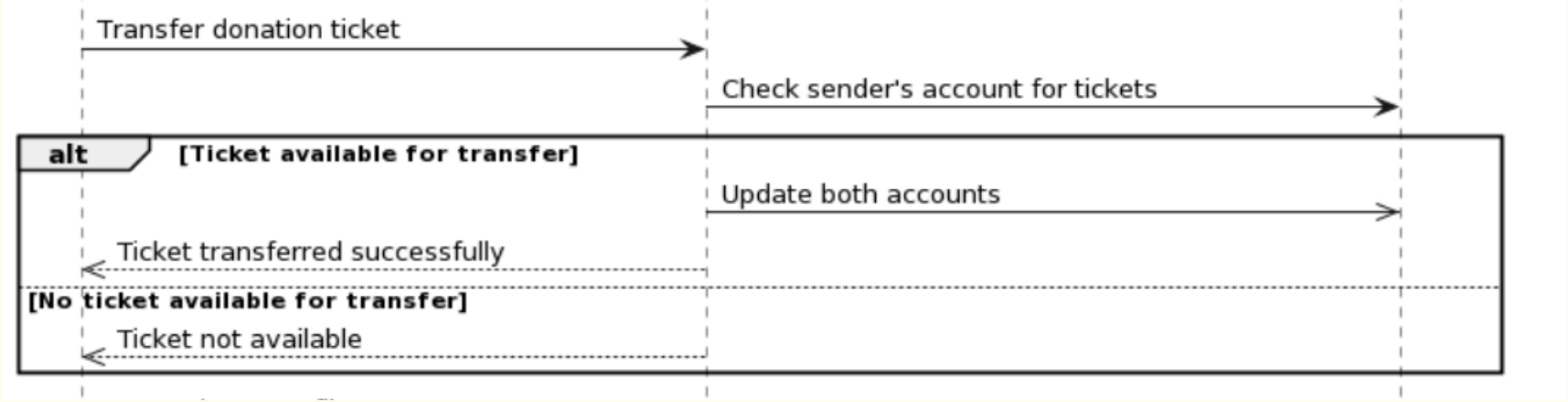
User can purchase a donation ticket to another user using the application that used for exchange the blood kind or any other fees needed.



System Features and Sequence

My Ticket – Ticket Transfer

User can see all his available tickets with details using the application by easy and friendly UI, then he can send a ticket to another user by his national number.



Application Class Diagram

Relation Snippet of User Entity

User:

User ↔ Role

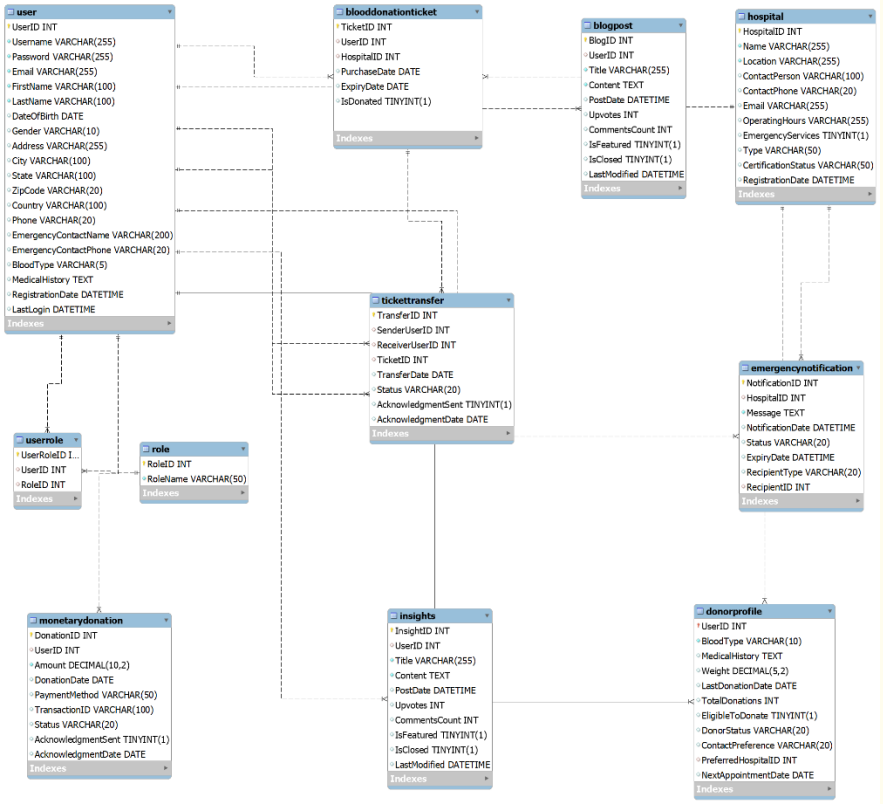
User ↔ Monetary Donation

User ↔ Donor Profile

User ↔ Blog Post & Insights

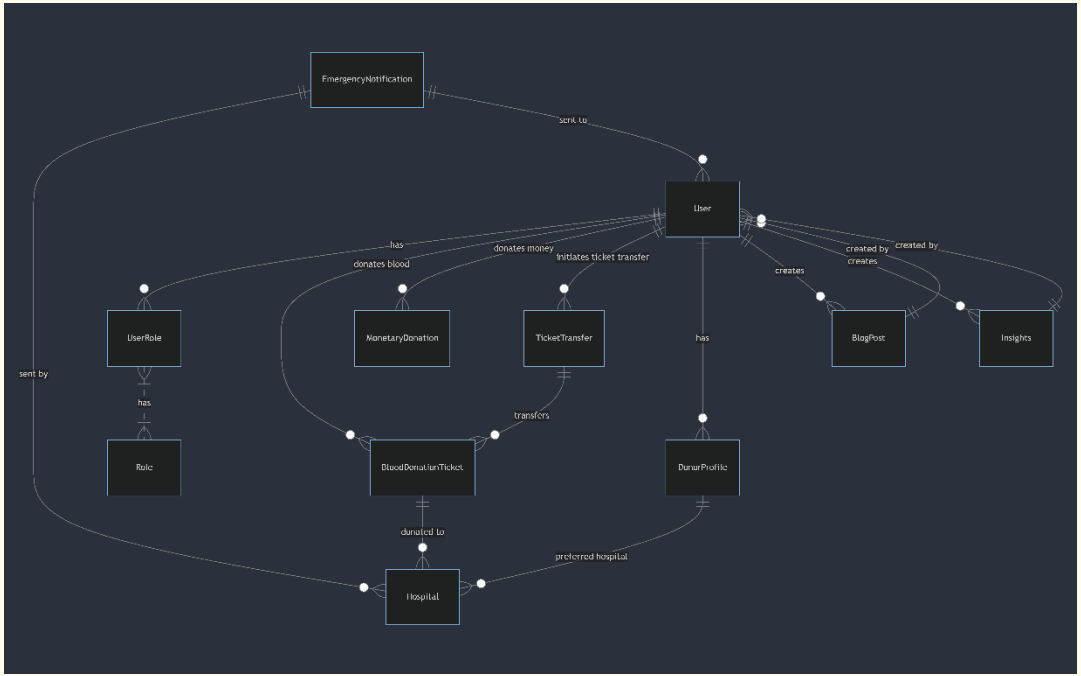
User ↔ Emergency Notification

User ↔ Ticket Transfer



Application Entity Relationship Diagram

ERD Snippet of User Entity



User:

User has Role

User makes Monetary Donation

User has Donor Profile

User creates Blog Post & Insights

User receives Emergency Notification

User transfers Ticket

Technology Stack

What is the used technologies for building the system ?

Technology list for each section.

Mobile Application

Database

Communication

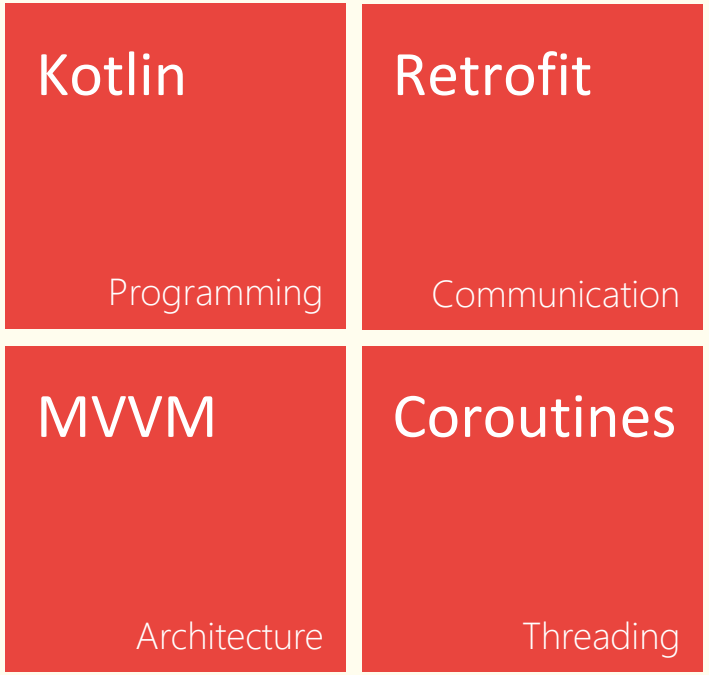
Web Application

DevOps

Mobile Development Stack

Used Technologies & Libraries

- Native Android
- Kotlin
- Glide
- Hilt
- Coroutines
- Viewmodel
- Markwon
- Prism4j



Web Development Stack

Used Technologies & Libraries

Html & CSS

User Interface Development

SQL

Only Showcase DB Relations & Diagrams

Firebase

Firebase FCM To Pass Notification

Spring || .Net

Backend Framework

Angular

Frontend Framework

TypeScript & JavaScript

For Frontend Development in the Admin end

Java || C#

For Backend Development

MongoDB

Backend Host

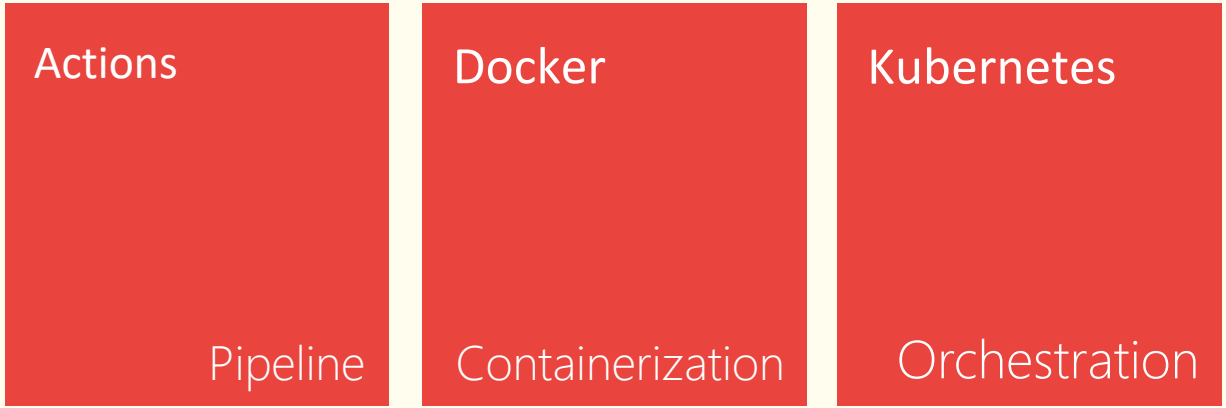
AWS

For Application Deployment

Cloud & DevOps Stack

Used Technologies & Libraries

- ArgoCD
- Helm
- Go Templates
- Nginx
- Ingress
- Services Discovery
- Docker Registry
- Kind Cluster



DevOps Used Technologies

Containerization: Implementing Docker for containerization, enabling consistent application deployment across different environments.

Orchestration: Utilizing Kubernetes for orchestrating and managing Docker containers, ensuring efficient resource utilization and scalability.

Service Discovery: Utilizing Nginx and Kubernetes with the Nginx Ingress Controller Helm Chart to facilitate service discovery, enabling dynamic communication between applications within distributed environments.

Project Approach

Planning:

Define project scope and objectives.

Prioritize features and requirements.

Create a roadmap for development.

Iteration Planning:

Break down work into smaller tasks.

Assign tasks to team members.

Develop and test the features.

Regularly review progress and adjust as needed.

Collaborate closely with team members.

Project Approach

Project follow **Agile** Approach

Sprint

Each Sprint 2 Week

Scrum

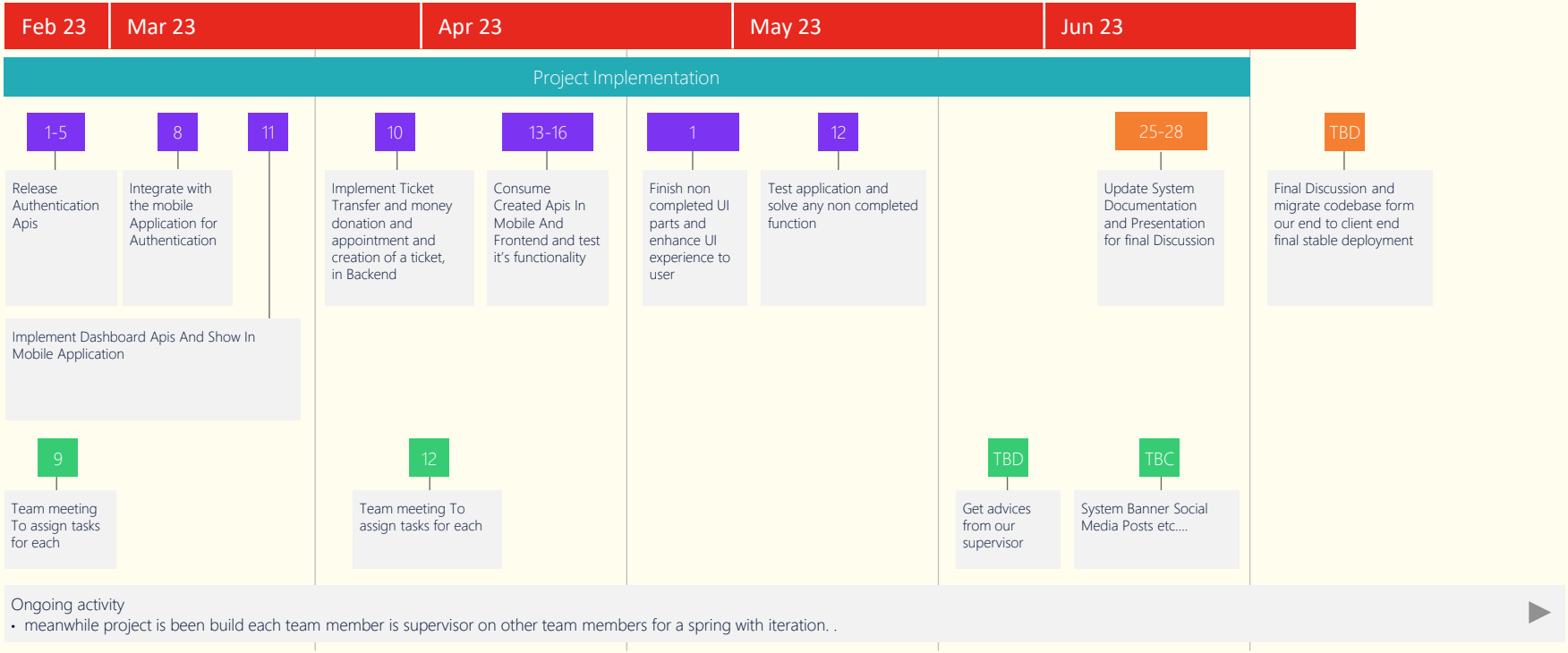
Each Member Scrum For a Sprint with Iteration

Meeting

Each Week a Meeting Update of Current Progress

Timeline/planning

Application Timeline for Second Term



Thank You