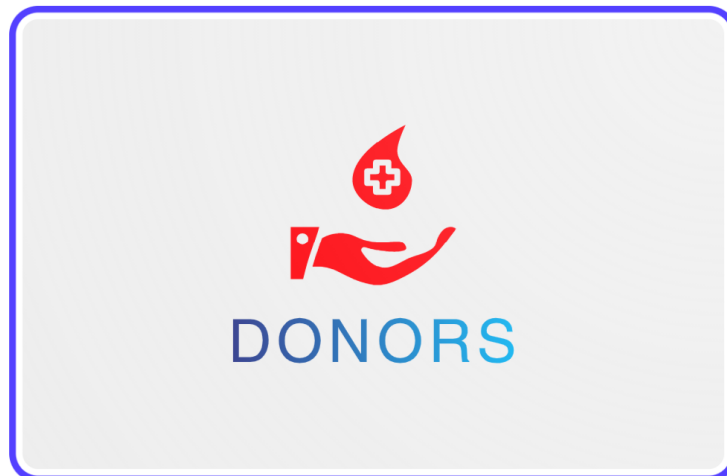


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
SPRING 2021**



**LIFESAVERS
DONORS**

**AAWAJ BHAIKAJEE
BIKASH SHRESTHA
LAURA ADHIKARI
MANISH SHRESTHA
SAYANA THAPA**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	04.05.2021	AB	document creation
0.2	04.09.2021	AB, BS, LA, MK, ST	rough diagrams drafts
0.3	04.12.2021	AB, BS, LA, MK, ST	computer generated diagrams
1.0	04.16.2021	AB, BS, LA, MK, ST	final review
1.1	04.16.2021	AB, BS, LA, MK, ST	final draft

CONTENTS

1	Introduction	5
2	System Overview	6
2.1	User Presenatation Layer	6
2.2	Server Layer	6
2.3	Cloud Service Layer	6
3	Subsystem Definitions & Data Flow	7
4	USER PRESENTATION LAYER	8
4.1	USER INTERFACE	8
4.2	USER AUTHENTICATION	8
4.3	USER INFORMATION	9
5	SERVER LAYER	11
5.1	SYSTEM API	11
5.2	GOOGLE MAP API	12
6	GOOGLE CLOUD SERVICE LAYER SUBSYSTEMS	13
6.1	DATABASE	13
6.2	GOOGLE PLAY STORE	14

LIST OF FIGURES

1	A simple architectural layer diagram	6
2	A simple data flow diagram	7
3	User Interface Diagram	8
4	User Authentication Diagram	9
5	User Information Diagram	10
6	System API Diagram	11
7	Google Map API subsystem Diagram	12
8	Database Diagram	13
9	Google Play Store Diagram	14

LIST OF TABLES

2	User Interface interfaces	8
3	User Authentication interfaces	9
4	User Information interfaces	10
5	System API interfaces	12
6	Subsystem interfaces	12
7	Database interfaces	14
8	Google Play Store interface	15

1 INTRODUCTION

"Donors" provides a platform for the people who actively donate their blood to the donation centers and hospitals. Also, it helps blood banks and hospitals to request for specific blood types to the donors. Donors is an android application where a hospital or any organizations needing blood for their patients or for any other applications request blood donations through it. On the other side, users get the notification/message on their phone. When the users acknowledge the message, they can either accept or deny it. They can accept it by scheduling for the donation on the locations where they find it easy, or they can deny it if they do not want to donate for the time being. After donating blood, users' records will be recorded in the database. Later, hospitals can go through each donor's profile to check their information, like the number of times the donor's donated blood. Scheduling will be a lot easier because time and place where the donation is taking place will be explicitly shown to the users and they can just simply view and schedule. At any time they can cancel it too. Another feature is that the user can search the blood donation centers on their own and schedule instead of waiting for the requests.

2 SYSTEM OVERVIEW

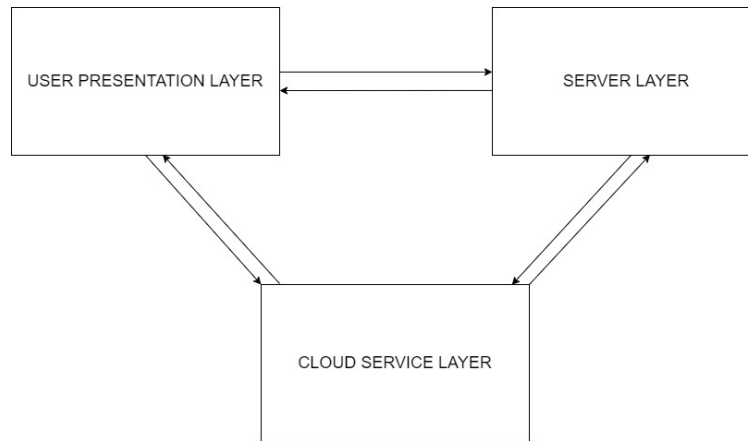


Figure 1: A simple architectural layer diagram

2.1 USER PRESENTATION LAYER

User presentation layer provides a user interface to the user which serves as the data translator for the network. It responds to the server request from the application layer and makes the information readable to the application layer. Also it makes sure the information is displayed properly in the receiving side.

2.2 SERVER LAYER

Server layer helps to communicate between the user layer and the cloud service layer (also known as the database layer). So, it acts as a bridge or an interface between these two layers. This layer can be considered as the heart of the application since the whole application is dependent and controlled by this layer. All the system APIs, and location API(google map) are the parts of the Server Layer.

2.3 CLOUD SERVICE LAYER

Cloud Service Layer serves as storing all the information of the users like username, password, blood type, blood pressure etc. as well as locations of different blood donation centers in a database as well allowing users to download the application from the Google Play Store.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

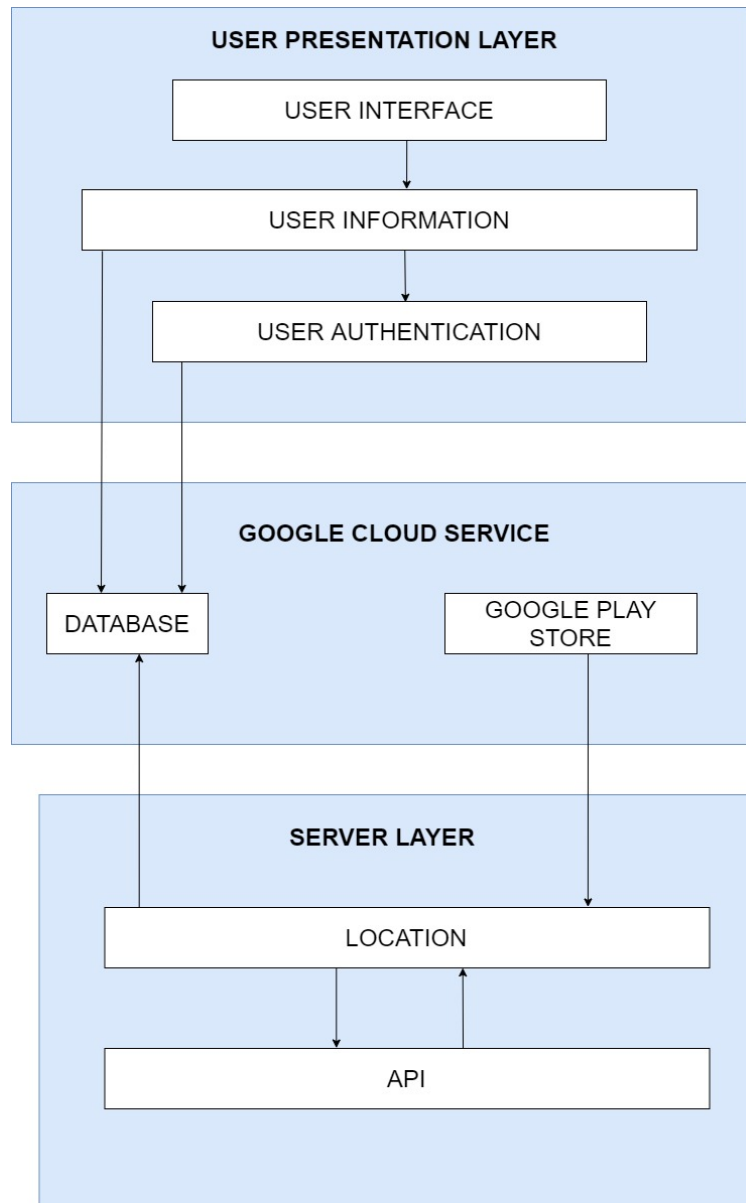


Figure 2: A simple data flow diagram

4 USER PRESENTATION LAYER

4.1 USER INTERFACE

The user interacts with the system through an android mobile application. The first interaction would be the login page where the user interacts with the system by providing the login information. The overall android application is an interactive system, therefore at each instance the user is able to see different buttons and click on them and the system in turn would provide corresponding response.

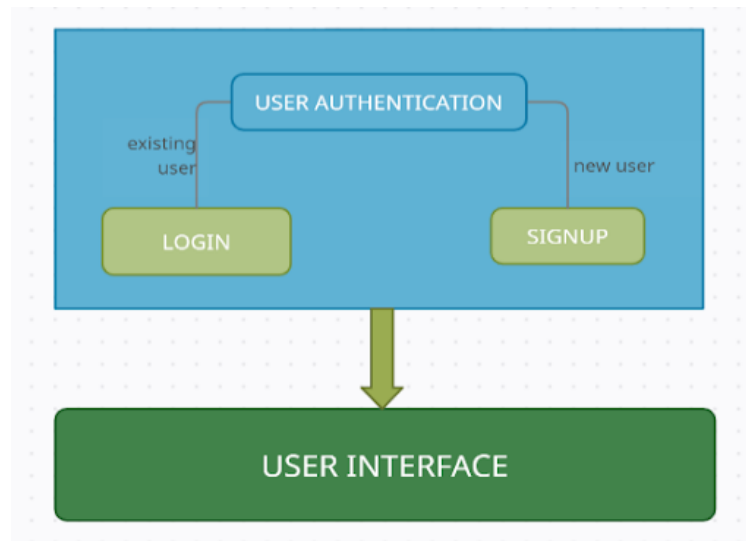


Figure 3: User Interface Diagram

4.1.1 ASSUMPTIONS

Assumption is made that the user is connected to the internet at all times. The interface would interact better with proper android API devices and location as well as notification turned on.

4.1.2 RESPONSIBILITIES

The subsystem is responsible to display proper response to each user interaction.

4.1.3 SUBSYSTEM INTERFACES

Table 2: User Interface interfaces

ID	Description	Inputs	Outputs
#1	Displays corresponding response to each user's touch on the screen	Requests/ Touch	Display a page/message or interactive page

4.2 USER AUTHENTICATION

This is the subsystem which is used to validate the users. It takes the user credentials and sends them to the server layer for authentication purposes. If all the credentials are valid or meet the requirements of the application, users will have access to the application. The type of user is also determined by this subsystem. Functions available within the app will differ as per the type of the users.

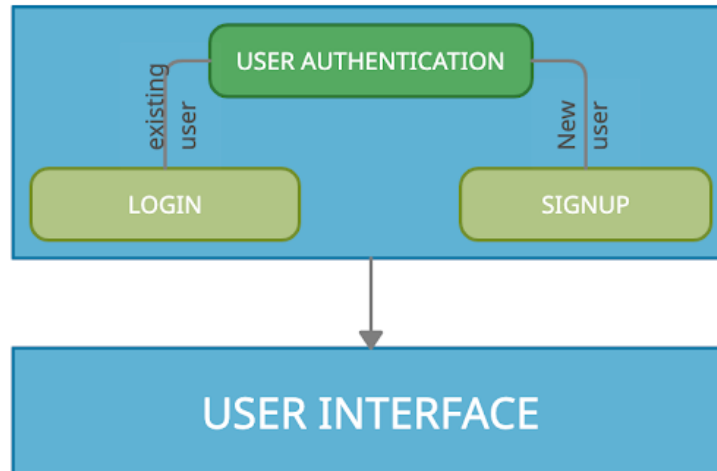


Figure 4: User Authentication Diagram

4.2.1 ASSUMPTIONS

User authentication is required each time to access the application. User information and the user interface are only provided after authenticating the user. After authentication, users can access their information and use available functions within the app through the simple user interface.

4.2.2 RESPONSIBILITIES

The main responsibility of this subsystem is to validate users and determine the type of users. Once the user is validated, users will be given access to the application.

4.2.3 SUBSYSTEM INTERFACES

Table 3: User Authentication interfaces

ID	Description	Inputs	Outputs
#1	Login	Validate username and password	Give access or "Invalid" message
#2	Sign Up	Sign Up button	Redirect to sign up page

4.3 USER INFORMATION

It is the information provided by the user on the interface for authentication.

4.3.1 ASSUMPTIONS

The information provided by the user is not incorrect. The user information is being provided by the associated user.

4.3.2 RESPONSIBILITIES

The user information provided by the user is correct and valid. It meets up all the requirements for being a valid entry (like matching the cases, special characters,length, etc.

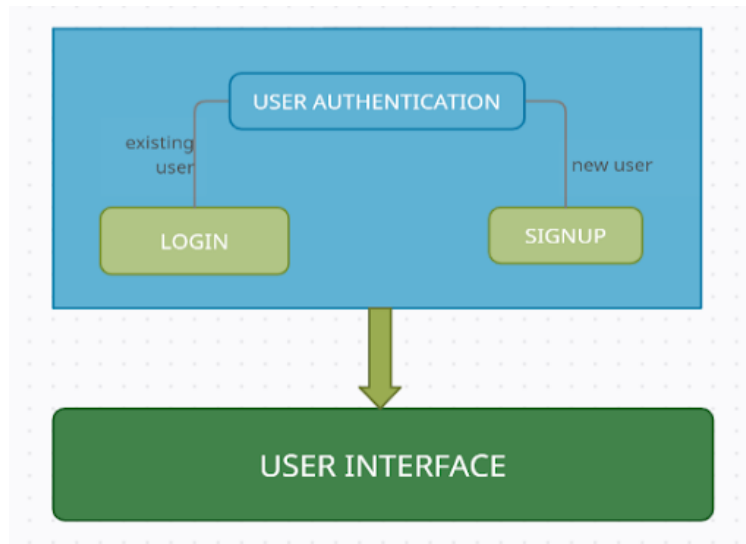


Figure 5: User Information Diagram

4.3.3 SUBSYSTEM INTERFACES

Table 4: User Information interfaces

ID	Description	Inputs	Outputs
#1	Gets input and validates the input	username and password, blood group blood pressure	Prompt error if invalid input The valid information is visible

5 SERVER LAYER

5.1 SYSTEM API

It is the most important subsystem of the application which includes all the necessary APIs required to communicate between the different layers of the system. This subsystem acts as the interface between the user layer and the database layer. It takes the input from the users and stores the data in the database. Then the desired output is also displayed to the users via this layer.

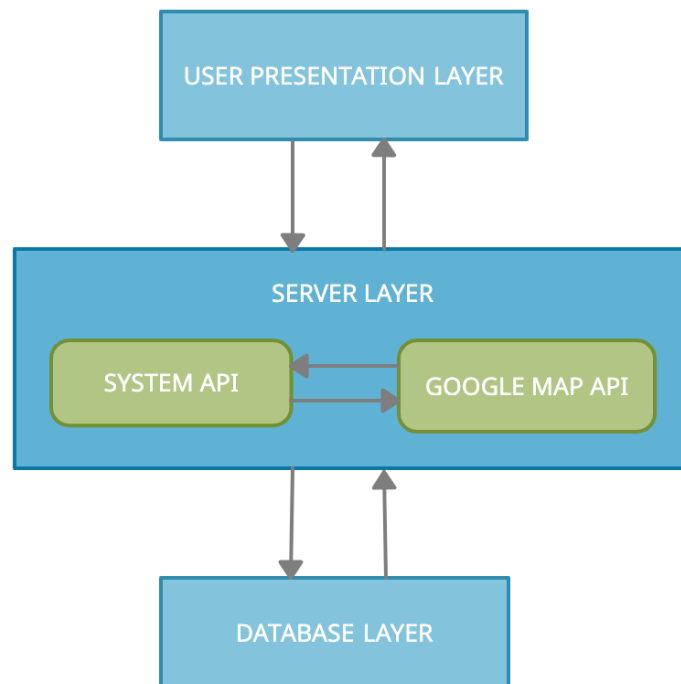


Figure 6: System API Diagram

5.1.1 ASSUMPTIONS

System API will communicate all the layers as well as google map API within the server layer. There is a two-way data flow between these layers. All the buttons and functions of the application use system API to access and store data in the database layer. Also, the database layer needs the system API to receive and send data. System API will also have two-way data flow with the google map API.

5.1.2 RESPONSIBILITIES

This subsystem is the control center of the application. This is the part which handles all the user requests and button events. It receives all the user inputs and other data, interprets the data, sends and retrieves data from the database layer, and displays the desired output to the users. It is the responsibility of the system API to necessary parameters to the google map API. It is also responsible for accessing and storing data in the database layer.

5.1.3 SUBSYSTEM INTERFACES

Table 5: System API interfaces

ID	Description	Inputs	Outputs
#1	Handles user requests and button events.	Requests	Response to the request or error message

5.2 GOOGLE MAP API

This subsystem is used to search and view locations. It will also show the distance between two locations by using the google map interface. It communicates with the system API which will help to access and store data in the database if required.

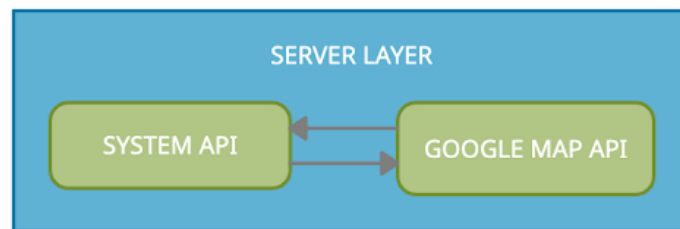


Figure 7: Google Map API subsystem Diagram

5.2.1 ASSUMPTIONS

With the help of google map interface, this subsystem will help to search for the locations and find the distance between the locations. It receives the user requests via system API and sends back the output to the system API. As per requirement, it could access data as well as store data in the database layer which will be done with the help of system API.

5.2.2 RESPONSIBILITIES

All the user requests regarding the locations which include search and view locations, pin locations, distance between two locations, etc. are handled by this subsystem.

5.2.3 GOOGLE MAP API INTERFACES

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Search for Locations	Search	Displays location to the user
#2	Calculate Distance	Request	Find the distance between two locations

6 GOOGLE CLOUD SERVICE LAYER SUBSYSTEMS

6.1 DATABASE

The donor's information throughout the application is stored in a database. Whenever the user registers his/her account, the login information is saved in a database including username and password. It is connected with the user information and user authentication subsystems in User Presentation Layer and with the Location subsystem in the Server Layer.

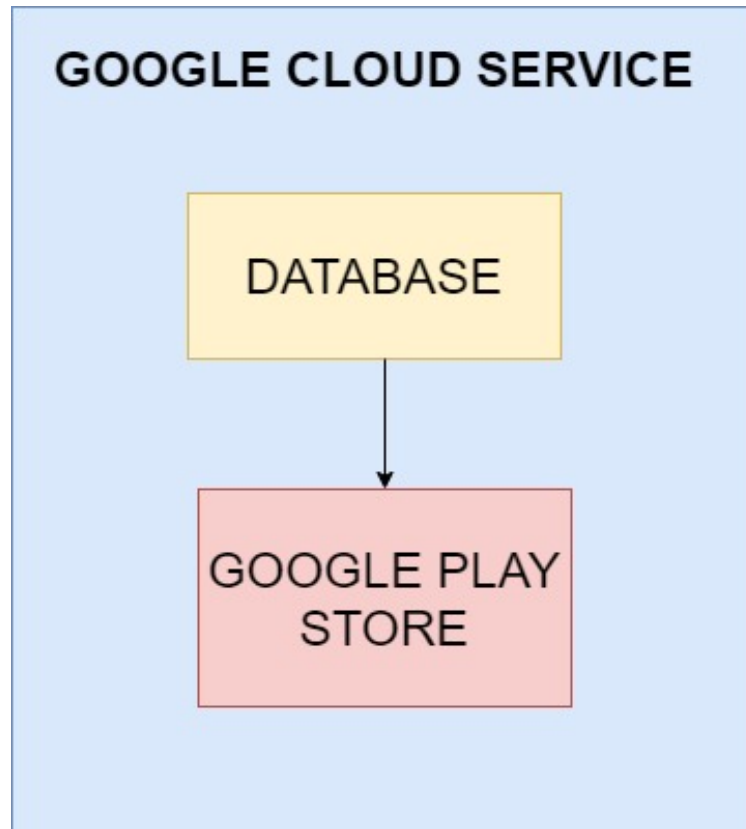


Figure 8: Database Diagram

6.1.1 ASSUMPTIONS

It can be assumed that the information in the database will update automatically whenever the information of the donor is updated. For example, when a donor will donate blood, the record of the number of blood donations by a user will increase by 1. Database will successfully update the record.

6.1.2 RESPONSIBILITIES

Right from the beginning, when a user creates an account, the information is saved on the database. Basic information like username and password is protected so that no unknown person can login to the account. The main data will be the blood type so that hospitals can easily locate the donor by the type of blood they need. Database will also update the records periodically and will keep track of the location of donation centers too.

6.1.3 SUBSYSTEM INTERFACES

Table 7: Database interfaces

ID	Description	Inputs	Outputs
#1	Get name of donor	Username	Store name
#2	Get password from donor	Password	Store password
#3	Get valid information of donor	Blood type Date of birth Blood pressure	Store valid information
#4	Update record of donor	Number of blood donation	Store records
#5	Save location of donation centers	Location	Store Locations

6.2 GOOGLE PLAY STORE

After the application is complete, users will be able to download it from the google play store.

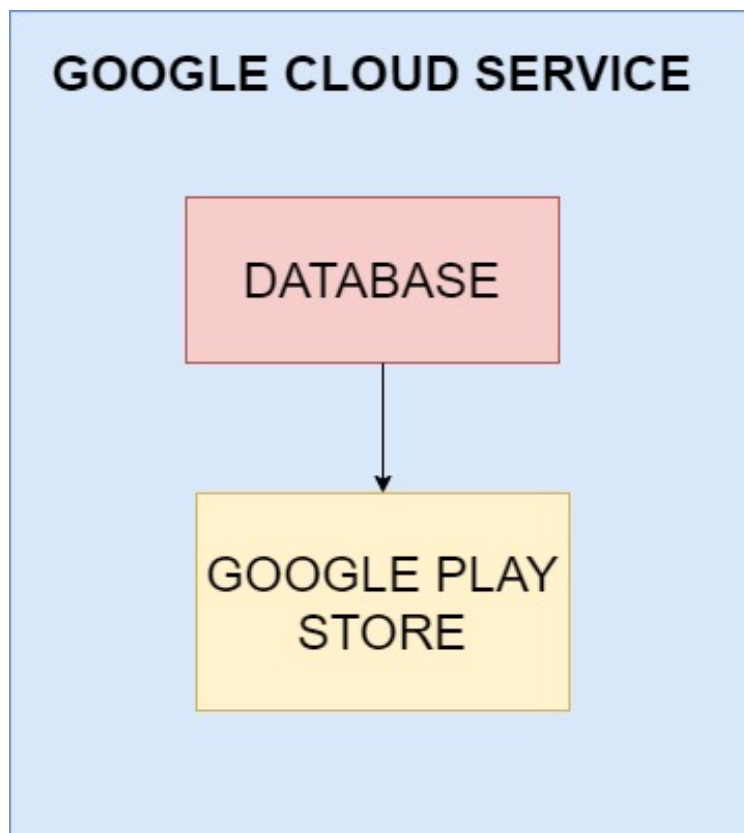


Figure 9: Google Play Store Diagram

6.2.1 ASSUMPTIONS

There will be no charge to download the application, it will be free to all the users.

6.2.2 RESPONSIBILITIES

The application should always be available to be downloaded by everyone from the google play store.

6.2.3 SUBSYSTEM INTERFACES

Table 8: Google Play Store interface

ID	Description	Inputs	Outputs
#1	Users will download the application	Download	Applications will be downloaded and installed on their phone.

REFERENCES