Contents

[Chapter 3: Design 1](#_Toc10497661)

[Structural design 1](#_Toc10497662)

[Class Diagram: 1](#_Toc10497663)

[Dataflow Diagram (DFD): 2](#_Toc10497664)

[Behavioral design 6](#_Toc10497665)

[Activity Diagram: 6](#_Toc10497666)

[Sequence Diagram: 9](#_Toc10497667)

[Database design 12](#_Toc10497668)

[Data Dictionary: 12](#_Toc10497669)

[ER Diagram: 15](#_Toc10497670)

[System Architecture 18](#_Toc10497671)

[User Interface 20](#_Toc10497672)

[Prototype: 20](#_Toc10497673)

# Chapter 3: Design

## Structural design

### Class Diagram:

**Introduction**

Class diagram is structural diagram which is static structure that shows the system process by defining the classes, attributes and operation use in it. Class diagram will define the relationship between object that were used in system.

**Purpose of using class diagram:**

* It helps to provide the static configuration of classifier in the system.
* It is helpful for developer and other member too.
* It can be used to model the system from business viewpoint.

**Notation Used in class diagram:**

1. **Class Name:**

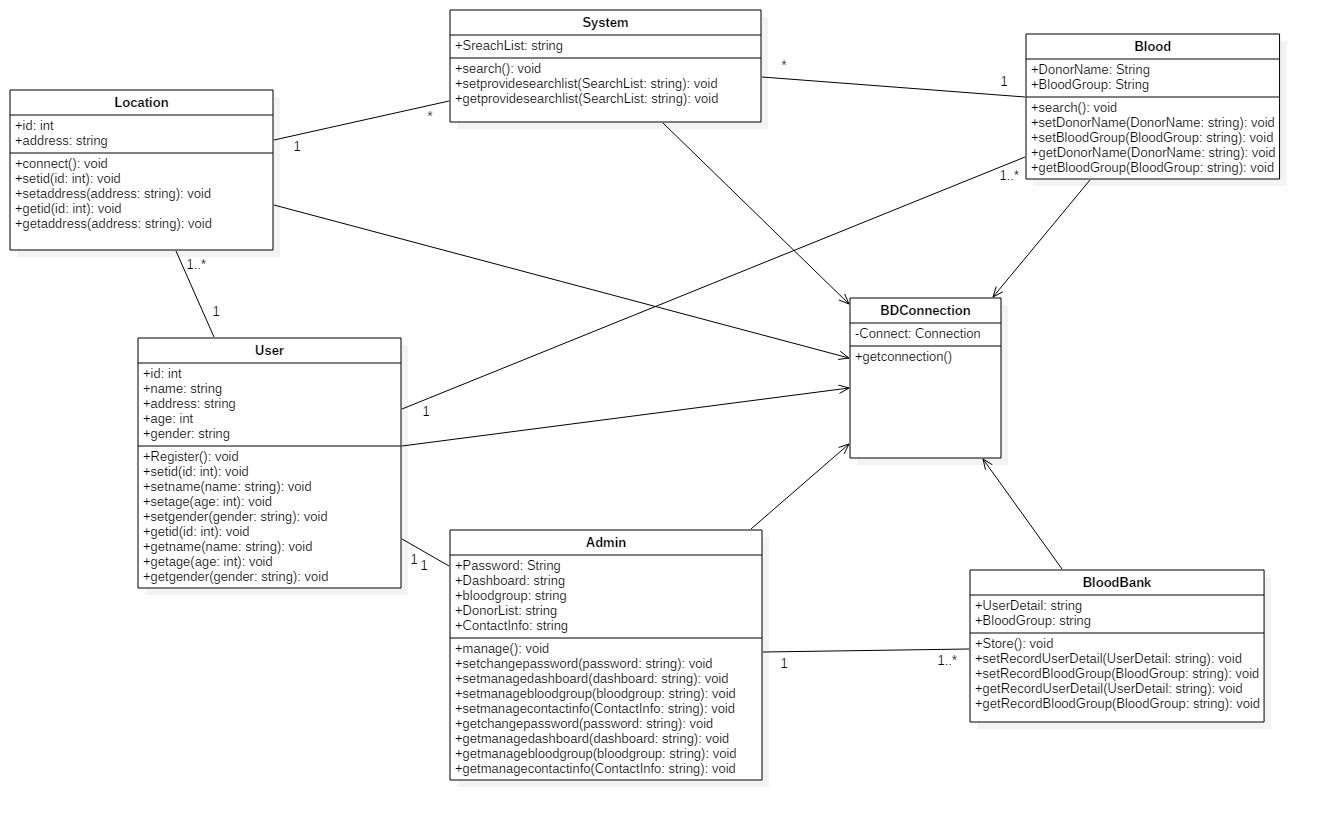
* The name of class perform in the first part.
* It can be extract for the scenario’s noun.

1. **Class Attributes:**

* Attributes are shown in the second part.
* The attributes type are shown after the colon.
* Attributes record into associate variables in program.

1. **Class operations:**

* It helps to provide services to class. And it comes in the third part in diagram.
* Operation record into class method in program.
* The return type of a process is displayed after the colon at the end of the method signature.



**Explanation**

This is final class diagram for my project which is mentioned above. There are five classes generated with the help of noun. I have used different operations for this classes. Also I have used setter and getter for methods for this classes. They all are interconnected with each other. I have used one to many relation for the classes.

### Dataflow Diagram (DFD):

**Introduction**

Dataflow diagram are the process which involves in the system to transfer data from the input to file storage. It can be dived into two parts that is logical dataflow diagram and physical dataflow diagram.

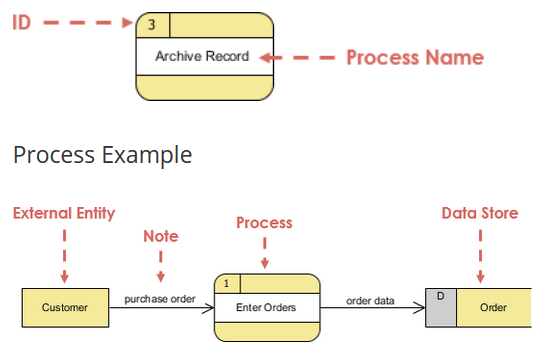
**Purpose of using Dataflow diagram:**

* Logical information flow system.
* Determining the physical system development requirement.
* Maintaining the manual and automated system requirement.

**Notation used in DFD:**

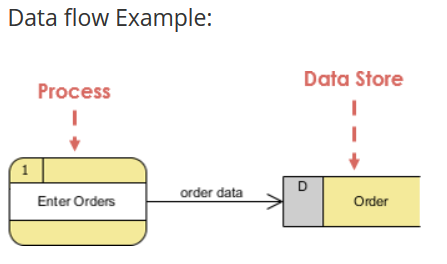
1. **Process:**

* This is represent by the rounded rectangle shape.
* A process receives input data and produces output with a different content or form.
* Processes are given ID for easy references.



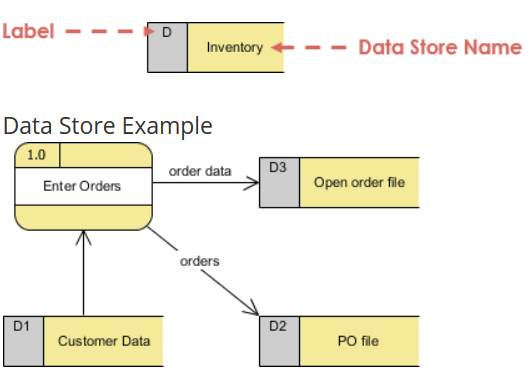
1. **Data flow:**

* They represent by straight line.
* A data-flow is a route for data to transfer from one part of the information system to another.
* Straight lines with received arrows are input data flow.
* Straight lines with leaving arrows are output data flows.

****

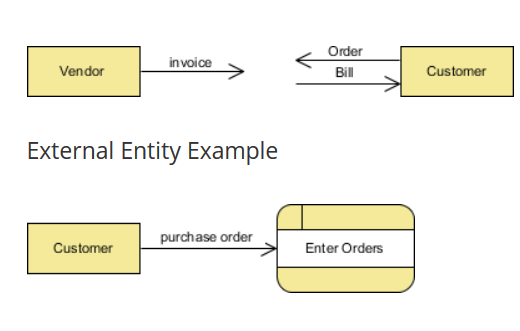
1. **Data store:**

* A data store is used in a data-flow diagram to represent a location when the system must keep information because one or more methods need to use the kept information in a future.
* Data can be stored into the data store, which is shown by an outgoing arrow
* Data can be read from a data store, which is shown by an incoming arrow.

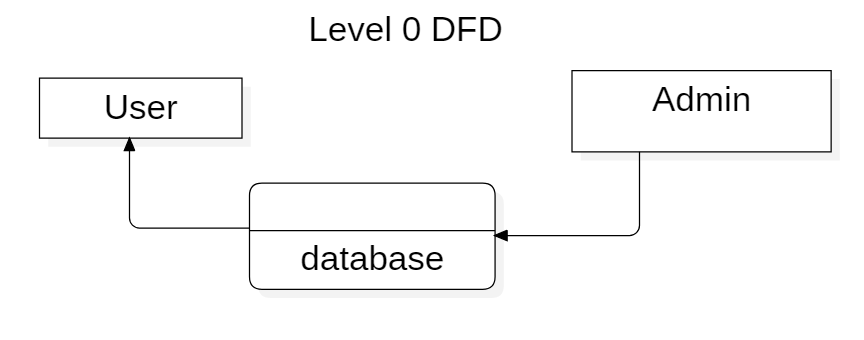


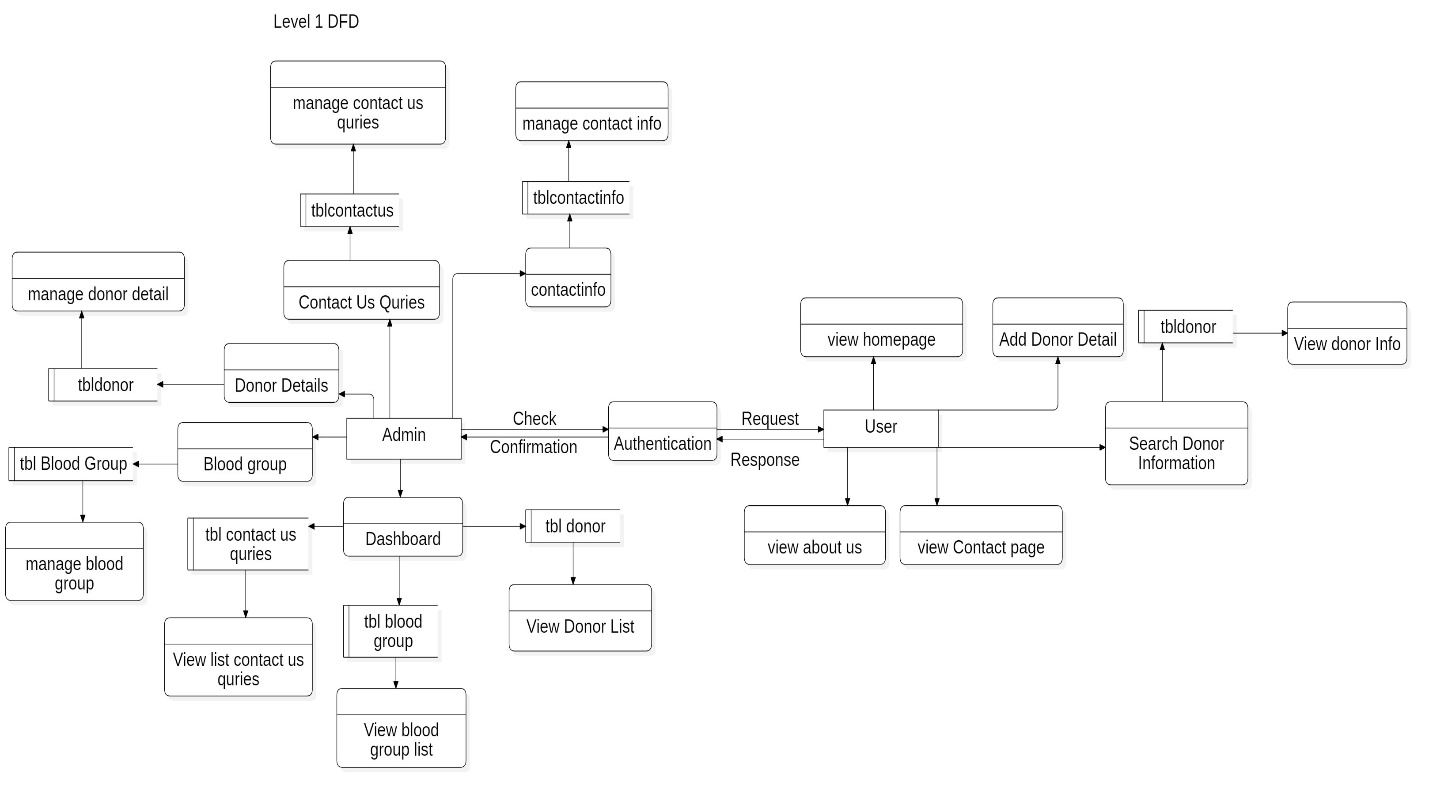
1. **External Entity:**

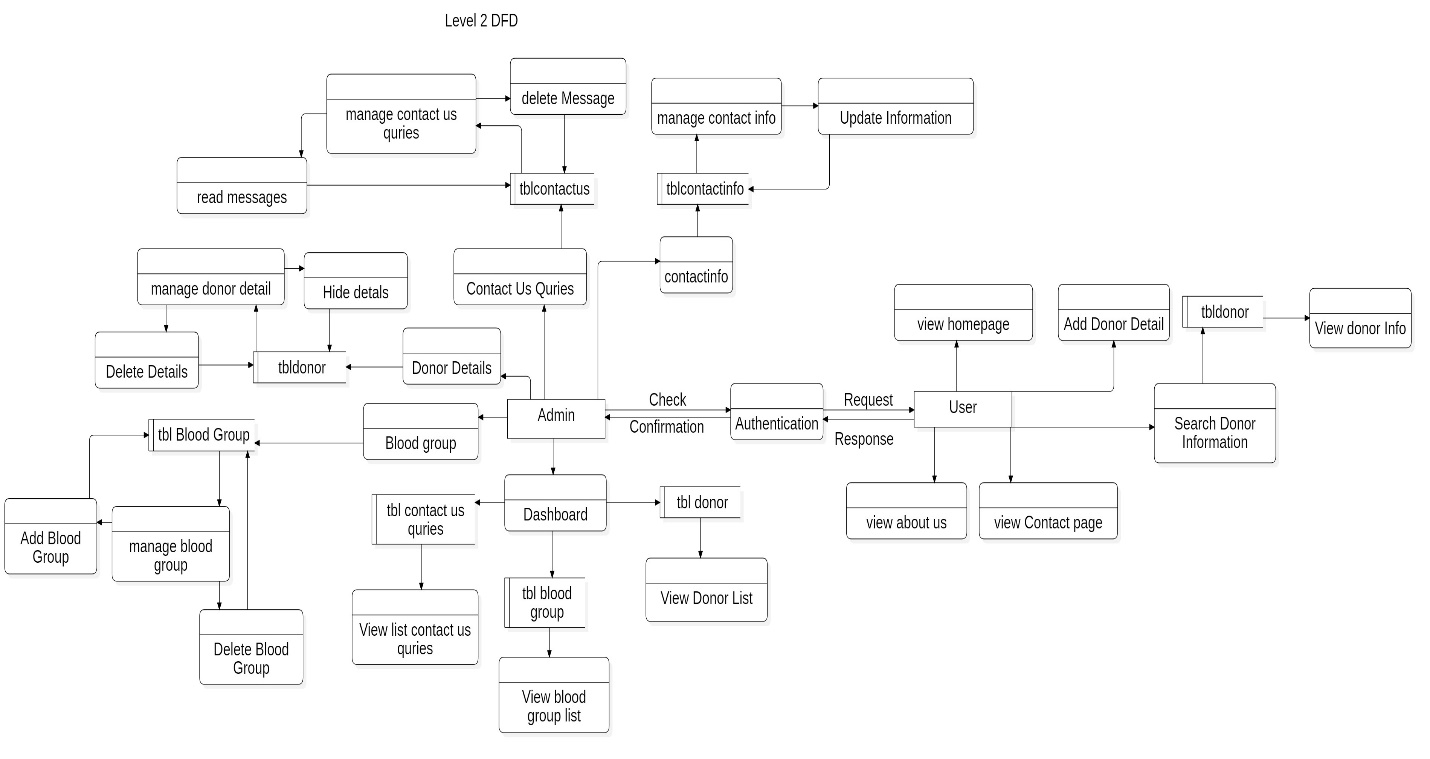
* They represented by rectangle shape.
* They either send data or receive data.
* They do not process data.



**Diagram:**

****





**Explanation:**

I have create DFD In three level that is level 0, level 1 and level 2. In level 0 I have just shown the user database and admin. After then I have come to level 1 DFD in where user and admin are two entity for DFD. In user entity, user has to send request for authentication after response user can use application to look homepage, add donor details, about us page, contact us page and search the required blood through location and blood group and these all data will be store their respected database table. After that in admin entity can see all the donor information, message and store blood group in the dashboard. Admin can manages donor information, pages, contact info, and send messages and their data will be store in the respected database. After completing level 1 DFD, I have created level 2 DFD in where user can only use application to look homepage, add donor details which will be store in tbldonor in database table, about us page, contact us page and search the required blood through location and blood group. After that in admin entity can see all the donor information, message and store blood group in the dashboard. Admin can manage donor details that is admin can delete or hide the donor details from the tbldonor table in the database. Admin also manage blood group that is admin can update or delete blood group from tblblood from the database table. Admin can update contact us information from tblcontact table in the database. Also admin can manage send message from user that is read or delete the message from tblcontactus table in the database. in this way I have completed the Dataflow diagram for my application.

## Behavioral design

### Activity Diagram:

**Introduction:**

Activity diagram is another main behavioral diagram in UML Design to define dynamic features of the system. Activity diagram is basically an advanced form of flow chart that forming the flow from one activity to another activity.

**Purpose of using Activity diagram:**

* Candidate use cases can be identify through the identification of system workflow.
* Model of workflows can be identify.
* Model of complex workflow in operation on object can be identity.

**Notation Used in Activity diagram:**

**UML Notation:**

**Activity –** It used to represent set of action.

****

**Action –** A task to be performed.

****

**Control Flow –** It shows the sequence of execution.



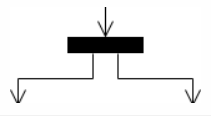
**Initial Node – it will start set of action or activities.**



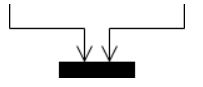
**Activity Final Node – it will stop all control of flow and object flow in the action.**



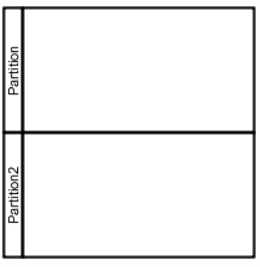
**Fork Node – It will split behavior into set of parallel flow of action.**



**Join Node – it will merge all split behavior from set of parallel flows of action.**

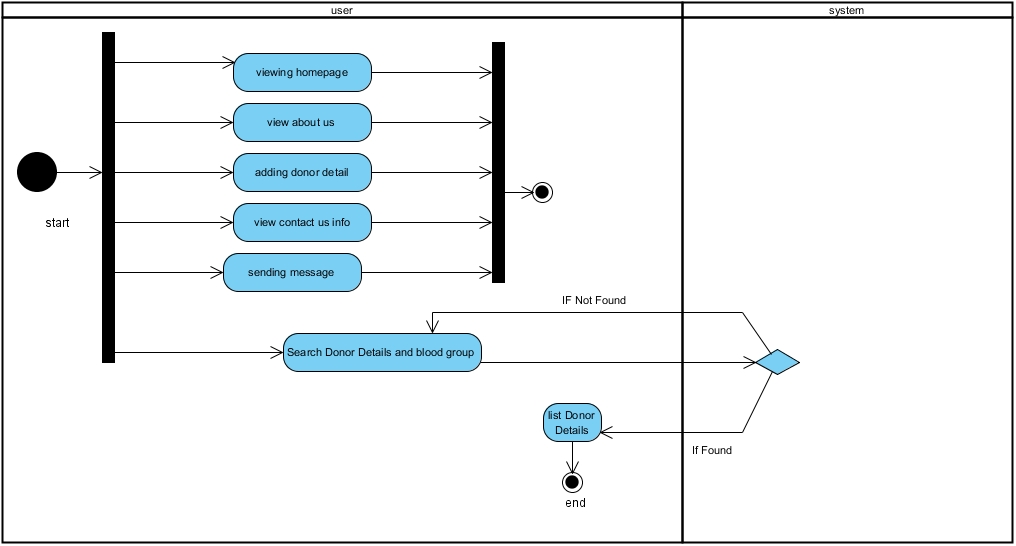


**Swimlane and Partition -** A way to collection actions executed by the same actor on an activity diagram or to collection actions in a single line.

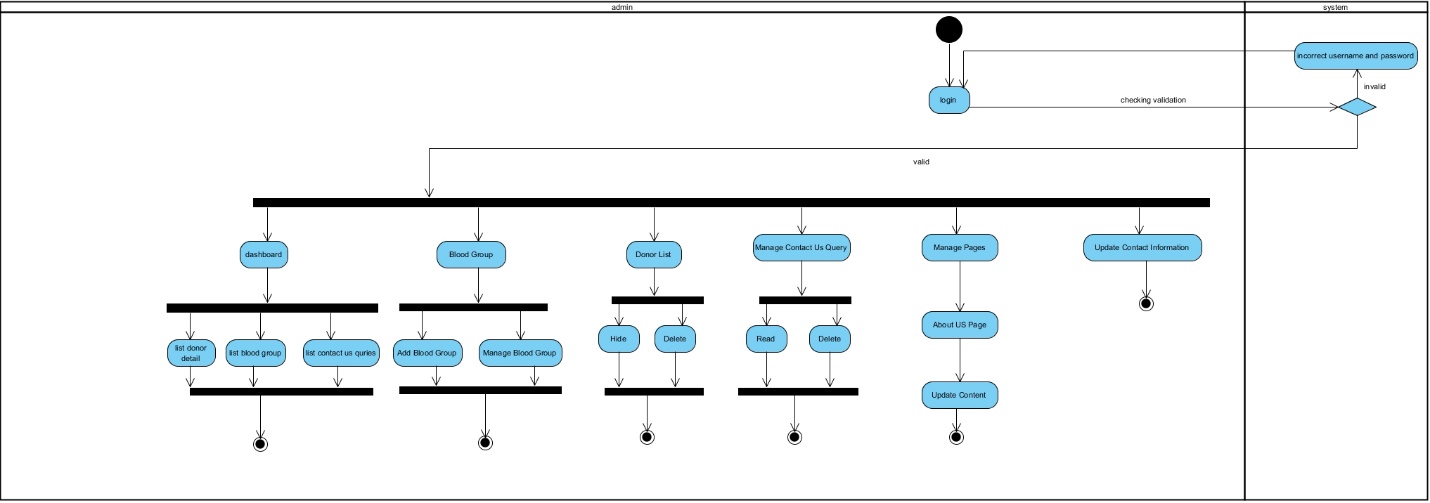


**Diagram:**

**For user:**

****

**For admin:**

****

**Explanation:**

For activity diagram I have separated two section for application that is for user and admin.

In user section user can view home page, about us page, contact us page, and search. In the search page, user has to enter blood group and location and then system will verify the data and If found it will listed the blood donor details and if not found it will send message not found.

After that admin part will come in where admin can do all the system manages. In admin section admin has to login into system which will be verify by the system. If system found the valid details it will open dashboard for the admin and if username and [password is incorrect then it will send message that invalid username and password. After login into system admin will manage the donor details that it delete and hide. Also admin will delete and add blood group. After that about us page will updated. Then at last contact info will be updated. Finally my activity for this application will complete.

### Sequence Diagram:

**Introduction:**

The diagram which interact with system to carry out the detail how operations performed is known Sequence diagram. They capture the relations between object in the perspective of a cooperation.

**Purpose of using Sequence Diagram:**

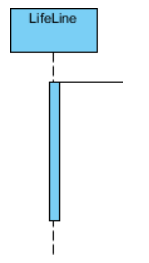
* Model high-level communication between active objects in a method.
* Model the communication between objects within a cooperation that understands an action.
* Model the relations between object cases within a cooperation that understands a use case.

**Notation Used in Sequence diagram:**

**Lifeline -** A lifeline represents an individual participant in the Relations.

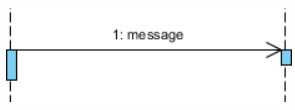


**Activations -** A thin rectangle on a lifeline) represents the period through which an element is acting an process.

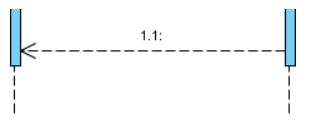


**Call Message -** A message defines a specific communication between Lifelines of an Interaction.

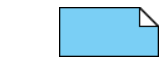
Call message is a kind of message that represents an invocation of operation of target lifeline.



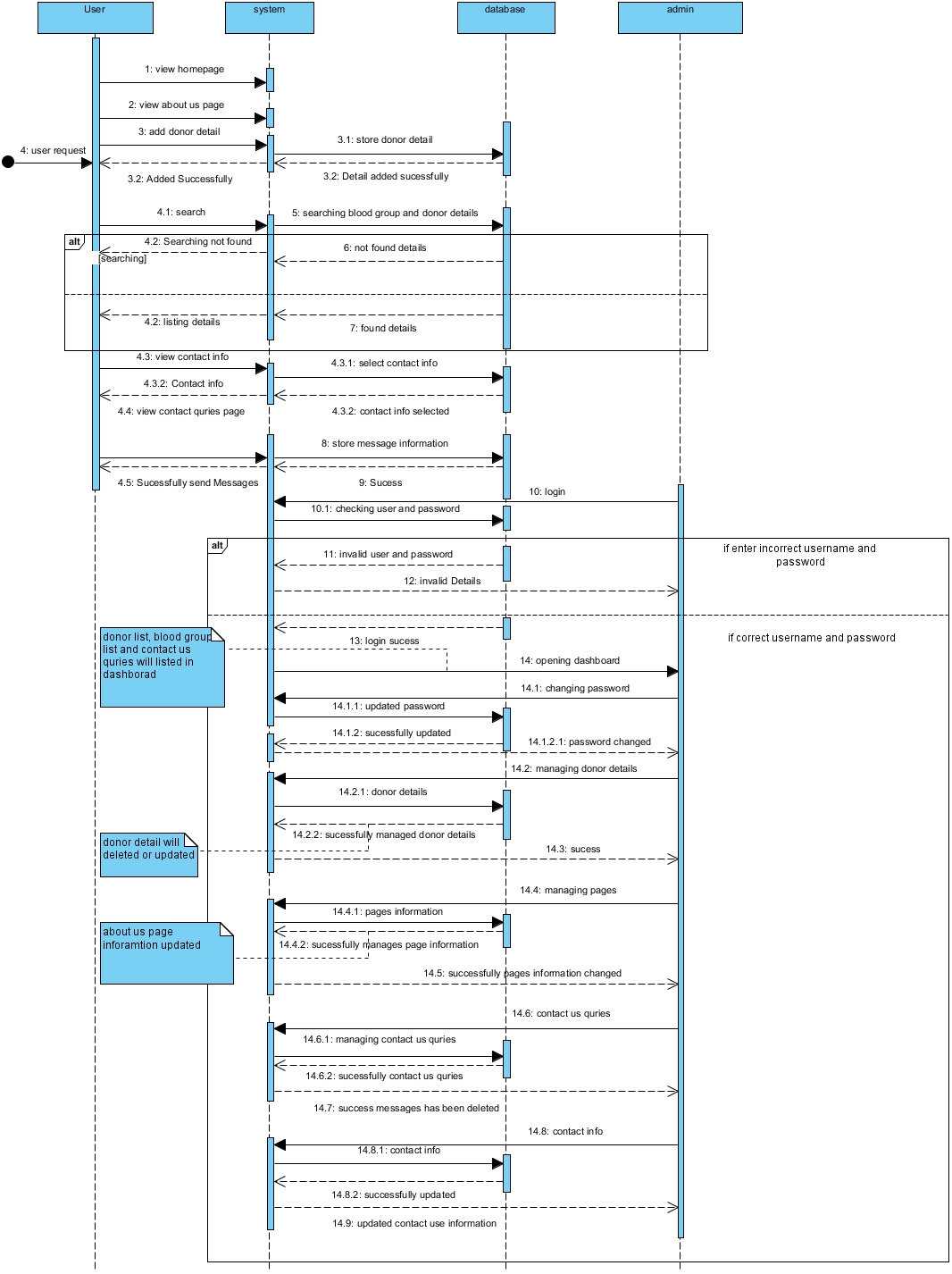
**Return Message -** Call message is a kind of message that characterizes call of process of mark lifeline.



**Note -** A note (comment) gives the capacity to attach different comments to elements. A comment carries no semantic force, but may contain information that is useful to a user.



**Diagram:**

****

**Explanations:**

In this sequence diagram I have created four lifeline and they are user, system database and admin. At first user request to view home page, about us page, contact us page then system will response to their request and open that request pages. After that user will add donor details then it will be send to system and system send that data into database. Then user search will required blood group a then system will search it and if found it will display the list and if not found then it will display not found message to the user. After then admin lifeline will come in where first admin have to login into system if the user and password is correct then it will open dashboard in where list of donor detail, list of contact query and list of blood group will be shown. If not correct username and password then it will show invalid detail. After login admin can manage the blood groups, manage pages, updated contact info and contact query. This process was shown in above diagram. This how I have create sequences diagram for my project

## Database design

### Data Dictionary:

**Introduction**

It is a record of data elements in a database or data model with detailed description of its format, relationships, meaning, source and usage.

**Elements of Data Dictionary**

Data dictionary basically made of number rows and column in where data element are rows and their attributes as columns.

Data Dictionary contains of two essential element and they are:

1. List of tables (or entities).
2. List of columns (or fields, or attributes).

**DBMS data dictionary**

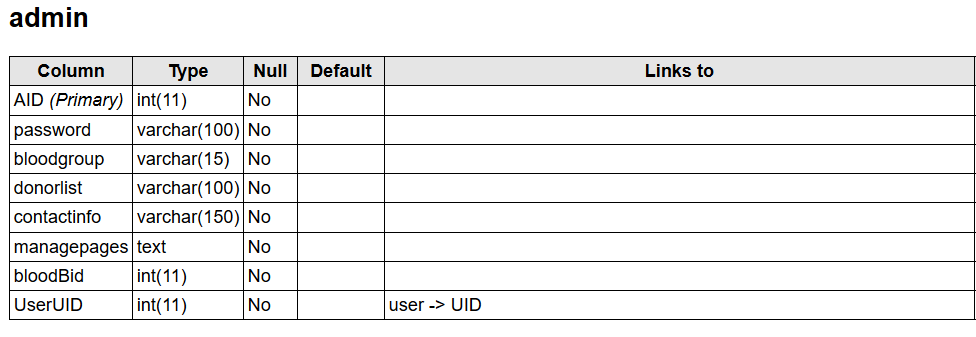
Relational database engines support much more explanation of data models and provide this info through their data dictionaries. This information is:

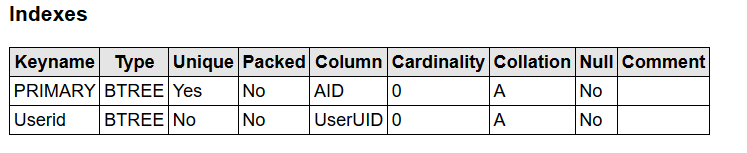
* Data type of presented in the columns.
* Colum’s default values.
* Nullable column.
* Relationship of tables (Like foreign key).
* Unique values that presented in the column (like primary key etc.).

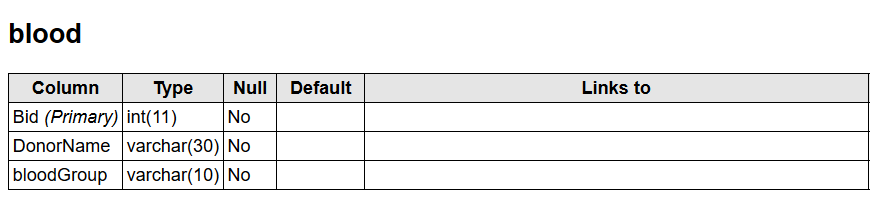
**Purpose of Data Dictionary:**

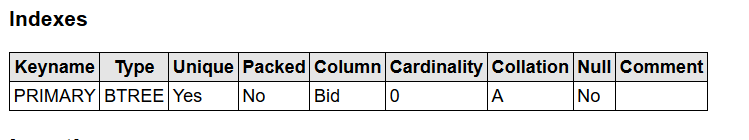
* Data Dictionary in database systems (DBMS)
* Data modeling
* Documentation

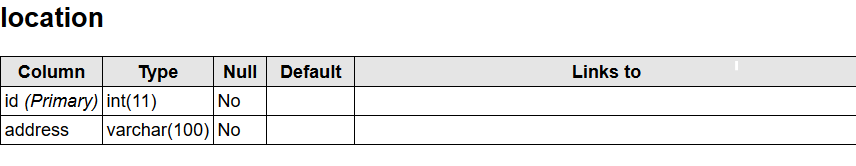
**Diagram:**

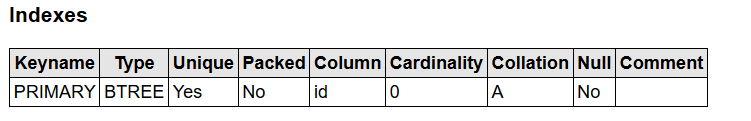
****

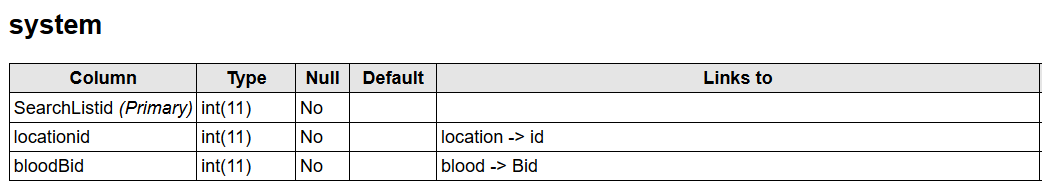
****

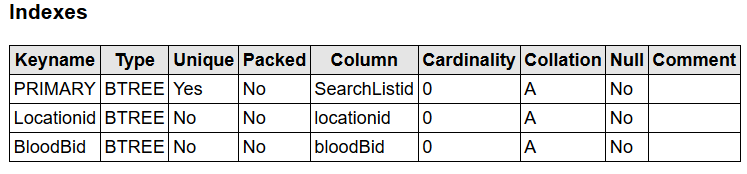
****

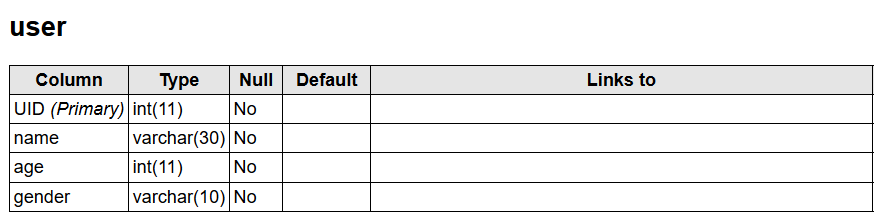
****

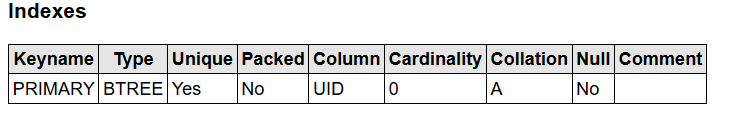
****

****

****

****

****

****

### ER Diagram:

**Introduction:**

Entity Relationship Diagram is known as ERD. It is a type of structural diagram for use in database design. In this diagram there different symbols and connectors were used which visualize two vital information they are:

* The major entities within system scope and
* The relationship between these entities

**Purpose of using ER diagram:**

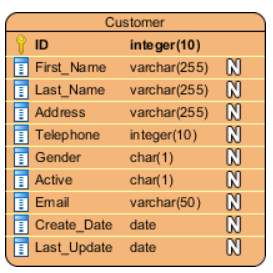
* It will helps to design the database structure.
* It will helps to manage entities, view their attributes and to identify the relation between these entities.
* ER diagram can determine the requirement of an information system by designing conceptual ERD which can represents the high level business object of the system.

**Notation used in ER Diagram:**

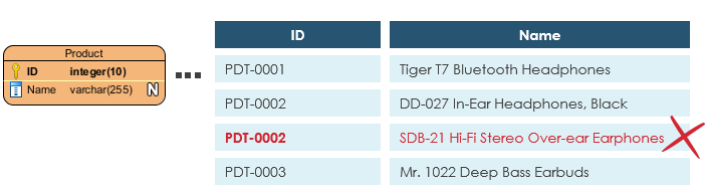
**Entity -** An ERD entity is a **definable thing or concept within a system.**

****

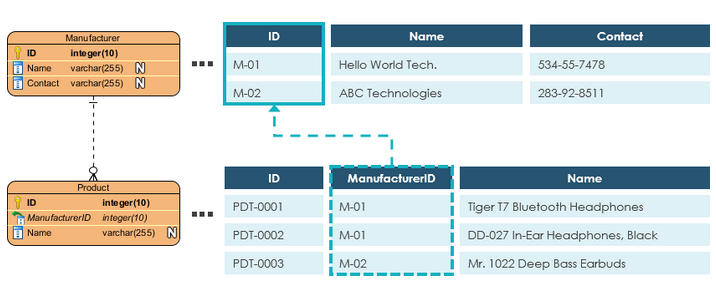
**Entity Attributes -** an attribute is a **property or characteristic of the entity that holds it**.



**Primary Key –** The special kind of entity attributes that are store a record in database table which define uniqueness in the table.

****

**Foreign Key –** Foreign key is referencing key to primary key in the table which is used to find the relations between entities.



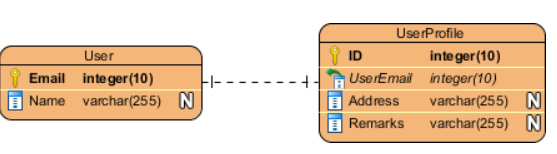
**Relationship –** it will help to manage the two entities linked with each other.

**Cardinality -** Cardinality defines the **possible number of existence in one entity which are linked to the number of existences in another.**

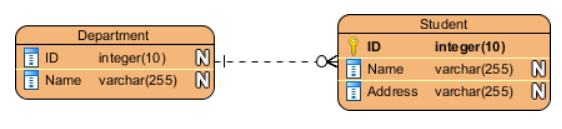
There are three main cardinality relations used which are mentioned below:

* One-to-one,
* One-to-many, and
* Many-to-many.

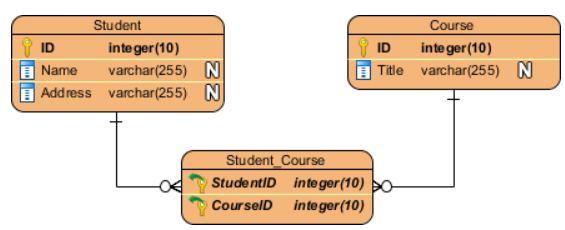
**One-to-One cardinality example:**

****

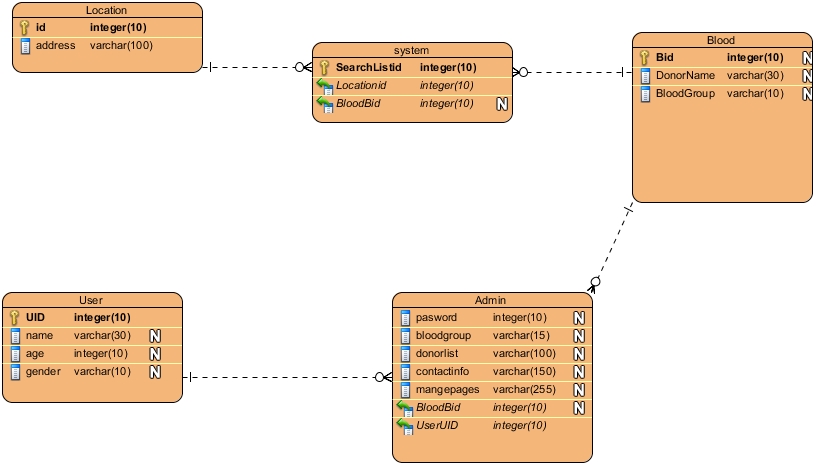
**One-to-Many cardinality example**

****

**Many-to-Many cardinality example**

****

**Diagram:**

****

**Explanation:**

This is ER diagram for my project. In this diagram I have created 5 entity which are location, system, blood, user, and admin. In this entity also there are different attributes uses inside the entity. I have given primary key for each entity. Along with there are foreign key also given for the required entity. I have use one – many relation for this diagram.

## System Architecture

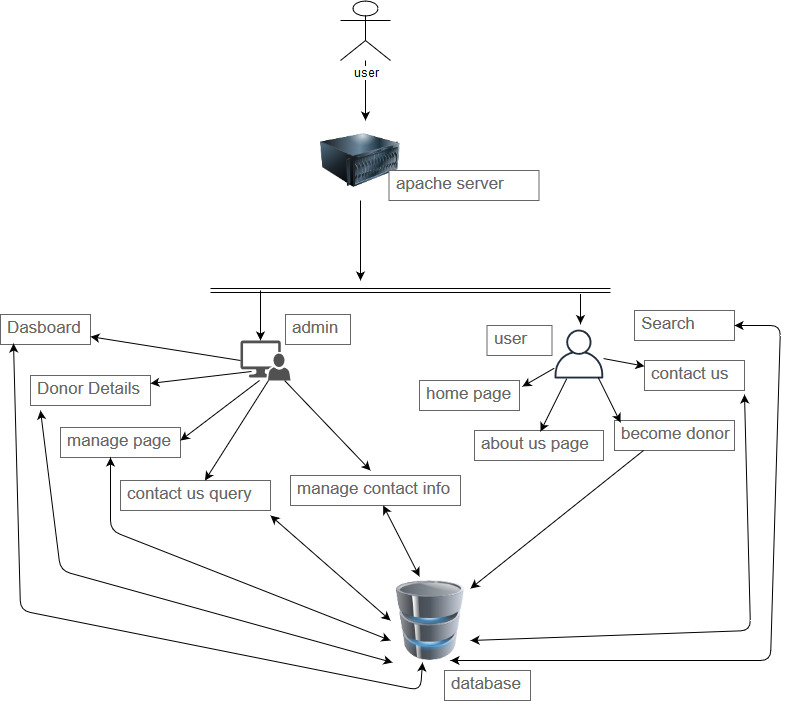
**Introduction**

The structural diagram which define the function of automated system is system architecture. It also define structure of the software. It includes elements such as services, layers, components, relationships, technologies, standards, principles, conventions and constraints. Architecture can be estimated built on business purposes in ranges such as cost, functionality, reliability, maintainability and operability.

**Purpose of using System Architecture**

* It will helps to know which server I have used in my software.
* It will helps me to map my hardware and software components.
* It helps me to know that my application will interact with human or not.

**Diagram**

****

In this system architecture design I have used 3 tier design. In this diagram user send request to sever for connection and after getting connection user will have two option that is admin and user. Also after getting connection user if user is client then client can use only user interface in where home page, about pages, become donor page, contact us and search option will be located. Or if the client is admin then he authority to use dashboard, managing donor details, manage page, manage contact us query and manage contact info. All the required data will be store in the database.

**Why I have used 3 Tier in system architecture:**

There are various benefits to using a 3-layer architecture including speed of development, scalability, performance, and availability.  As mentioned, modularizing various tiers of an application gives progress teams the capacity to progress and improve a software with more speed than developing a alone code base because a specific layer can be improved with least impact on the other layers. It can also help progress growth productivity by allowing teams to effort on their main capabilities. Scalability is another great benefit of a 3-layer architecture. By splitting out the different layers we can scale each freely conditional on the want at any given time.

## User Interface

### Prototype:

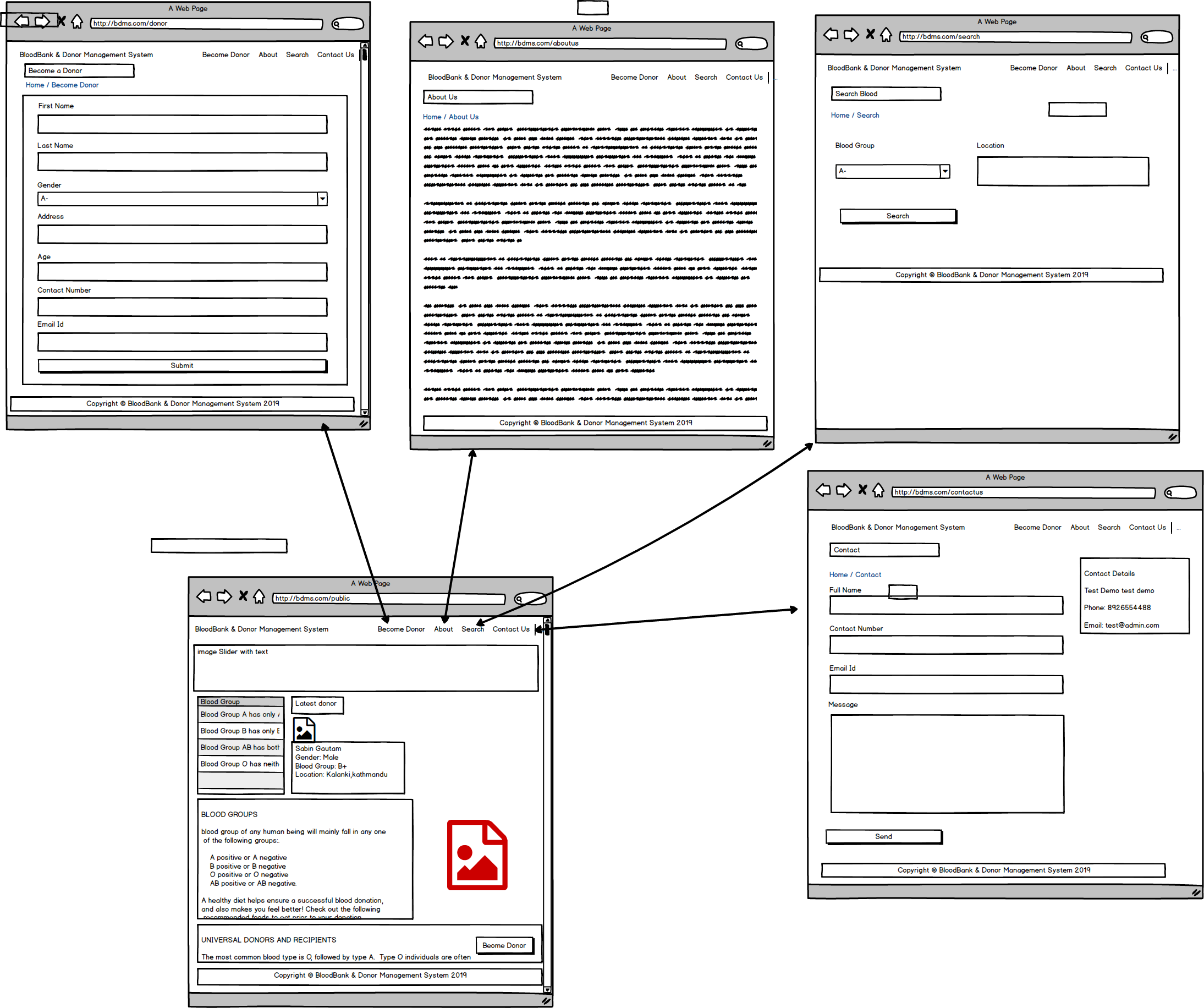
**Introduction:**

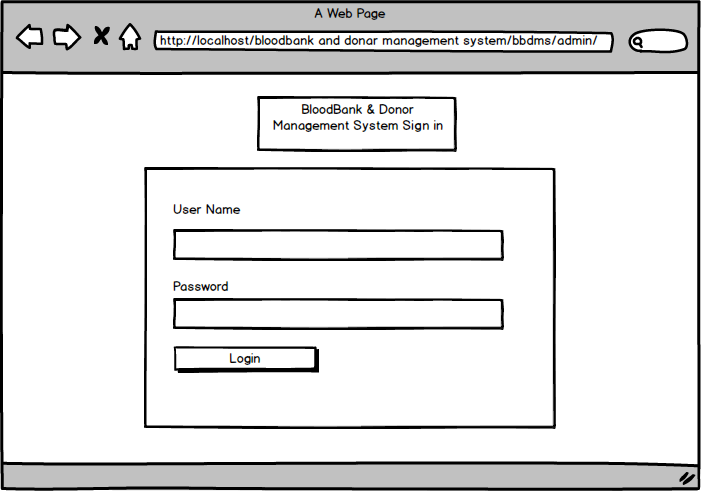
A prototype is a rudimentary working sample, model, mock-up or just a simulation of the actual product based on which the other forms (final product, and variations) are developed). The main motive behind prototyping is to validate the design of the actual product. Sometimes, creating a prototype is called materialization as it is the first step of transforming the virtual or conceptualized design into the real physical form.

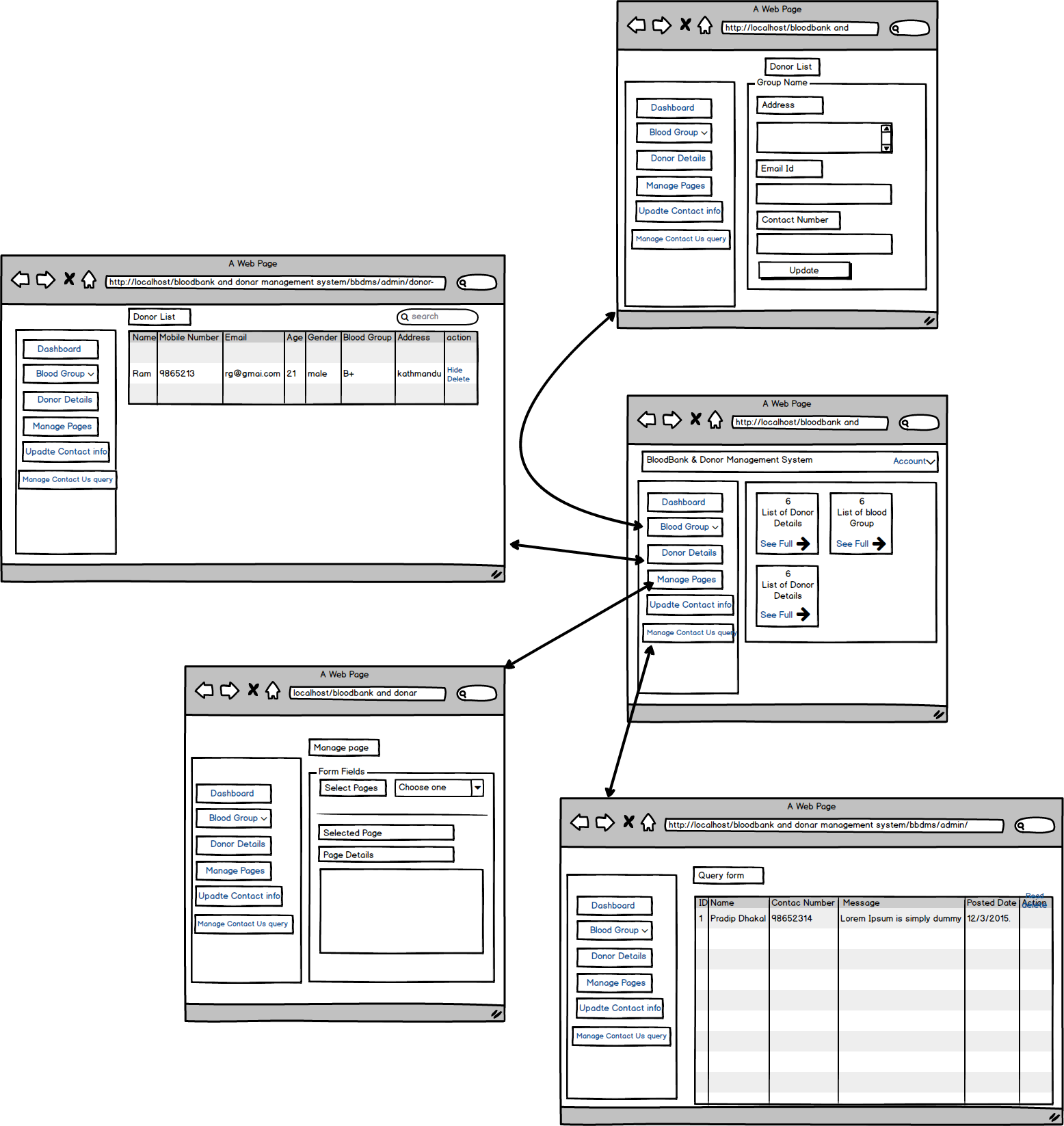
**Purpose of using Prototype:**

* It will validate the design of my project.
* It will provide logical property protection for my project.
* It will remove kinks in developed for my project.
* It will provide testing and refining for my project.

**Diagram:**

****

****

****

**Explanation**

Above mentioned diagram is the prototype of my project. There are two interfaces that is one for user and another is for admin. In user there is home page will open at first then user can go to different pages. And for admin there is login system that means admin has to login into system to use the interfaces. After login admin will open dashboard from that dashboard admin can find list of donor details, list of blood group, list of messages send from user. Also he can connect to different pages from that dashboard as shown in above picture.

# Conclusion

This is design section for my project. By performing this design I have come to how to create class diagram, activity diagram, sequences diagram and so…on. Also I have create prototype and data dictionary for my project. I learned to use different tools like star uml, visual paradgism and balsamiq. By using different tools I have completed my projects.