

BLOOD DONATION MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED IN FULFILMENT OF THE
DATABASE MANAGEMENT SYSTEMS

THIRD YEAR ENGINEERING

in

COMPUTER ENGINEERING

of

SAVITRIBAI PHULE PUNE UNIVERSITY

By

Vinay Valson (7358)

Yatharth Mathur (7360)

Suhrid Sabhlok (7344)



ARMY INSTITUTE OF TECHNOLOGY

PUNE



ARMY INSTITUTE OF TECHNOLOGY, PUNE
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

It is hereby certified that Vinay Valson, Yatharth Mathur and Suhrid Sabhlok of TE Computer Engineering class (section B) have completed a project titled, “BLOOD DONATION MANAGEMENT SYSTEM” under my guidance towards the fulfilment of the requirements for the Third Year Computer Engineering of Savitribai Phule Pune University during the academic year 2019-20.

Prof. Yogita Hambir

Department of Computer Engineering

Army Institute of Technology, Pune

Prof. Dr. S. R. Dhore

Head of Dept. of Computer Engg.

AIT, Pune

Prof. Dr. B. P. Patil

Principal

AIT, Pune

SYNOPSIS

Blood donation management system is a project model based on a database of blood donors which would make it easy for patients, hospitals and doctors to quickly get in contact with nearby blood donors. The model allows users to register on the website, as a doctor, blood donor or hospital. The aim of any such website is to make a database where information about blood donors is easily available to hospitals and doctors only. This would provide information like which regions should be targeted for blood donation camps or which blood group is lacking in hospital and to keep a continuous and well-maintained supply chain of blood to hospitals in order to reduce the deaths caused by lack of blood in hospitals.

The project uses HTML, CSS, JavaScript and Bootstrap for the generation of front end while the connectivity has been done using Flask which is well known and widely use framework of python. The backend uses MongoDB as a database management system. The project also uses Pymongo which is a python-based tool distribution.

This report discusses the result of the work done in the development of "Blood donation management system" on "HTML, CSS, Javascript, Bootstrap" Front-end Platform and "Mongodb" as back-end Platform with Flask being used to connect the two modules.

At the development of an application Flask and Pymongo provides a good connecting facility between all pages, also the back-end MongoDB is most important to save all the data related to the application.

The objectives of creating the Blood donation management system platform is as follows: -

- Create a database of blood donors
- Make blood donors easily available to hospitals
- Maintain privacy of blood donors

CONTENTS

1. Requirement Gathering and Scope finalization
2. Database Analysis and Design
 - I. Design Entity Relationship Model
 - II. Relational Model
 - III. Database Normalization
3. Implementation Details
4. Front End: HTML/CSS/JavaScript
5. Backend: MongoDB
6. Database Connectivity: Python Flask web framework
7. Testing: Data Validation

REQUIREMENT GATHERING

We are going to perform the project on UBUNTU platform so we need the OS as UBUNTU. Any version of Ubuntu preferably Ubuntu 18.04 or Ubuntu 16.0 would suffice the necessary operating system requirements.

The system should have minimum RAM of 4GB as well as minimum storage capacity of 100GB.

The various software, programs, packages and libraries used in the project should also be installed in the system including MongoDB, Flask and required compilers and editors for HTML, CSS, JavaScript and Bootstrap.

SCOPE OF PROJECT

The scope of the project is clear to give a simple and attractive application to simplify the work as well as to reduce the efforts while doing it offline or we can say by doing it with old methods.

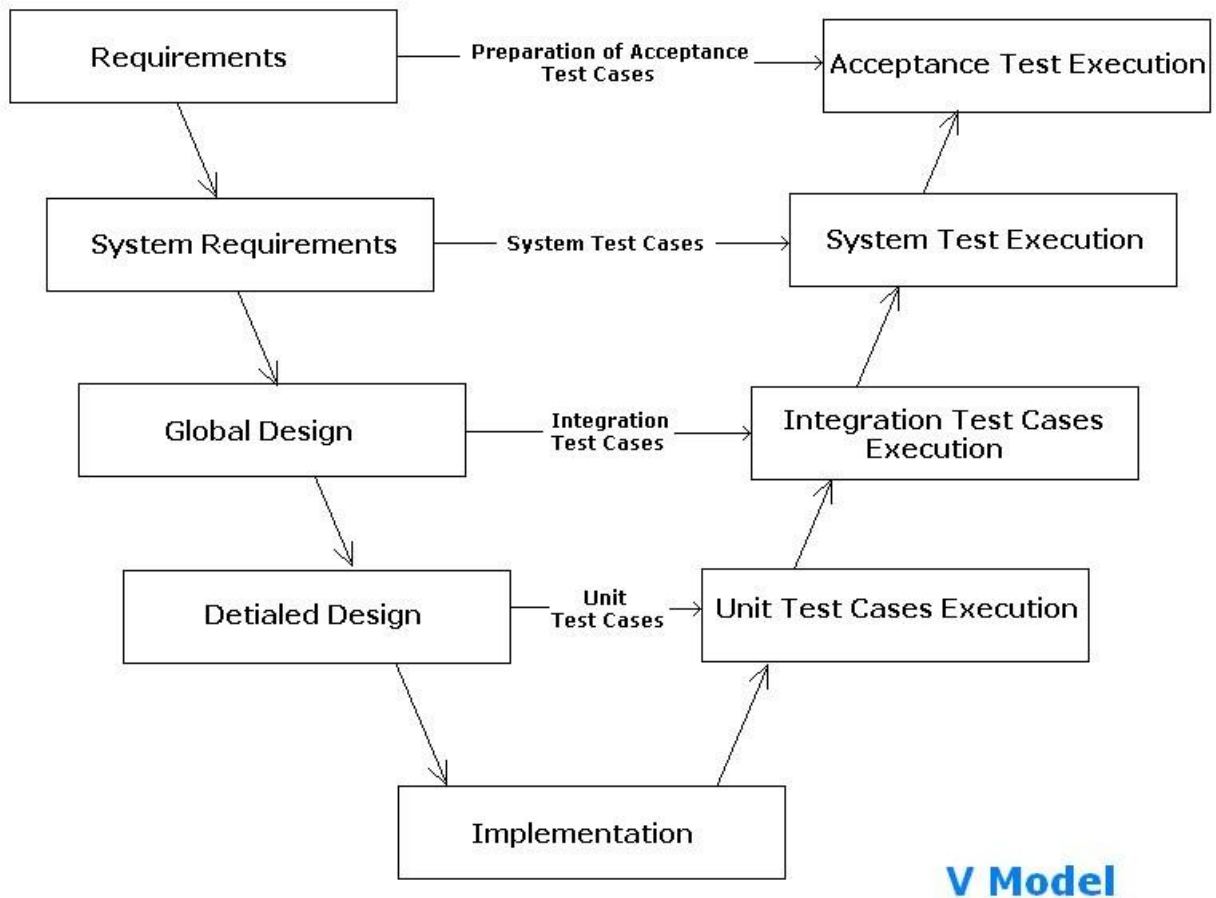
In this application we are able to save database of all products present on the site.

In this we can get access to any necessary information regarding donors just at the click of a mouse. This makes communication between donors and hospitals easy. During emergency situations it would be very crucial for such systems to exist.

Although the initial cost of developing computerized databases and portals would be low, the long run effects would be on the high expenditure on database management and security towards protecting the data and scalability. It would also be easy for people to eventually link their Aadhar cards or medical cards to such databases. Basically making the access to data to authorised personnel only like hospitals would be the main concern of such a system.

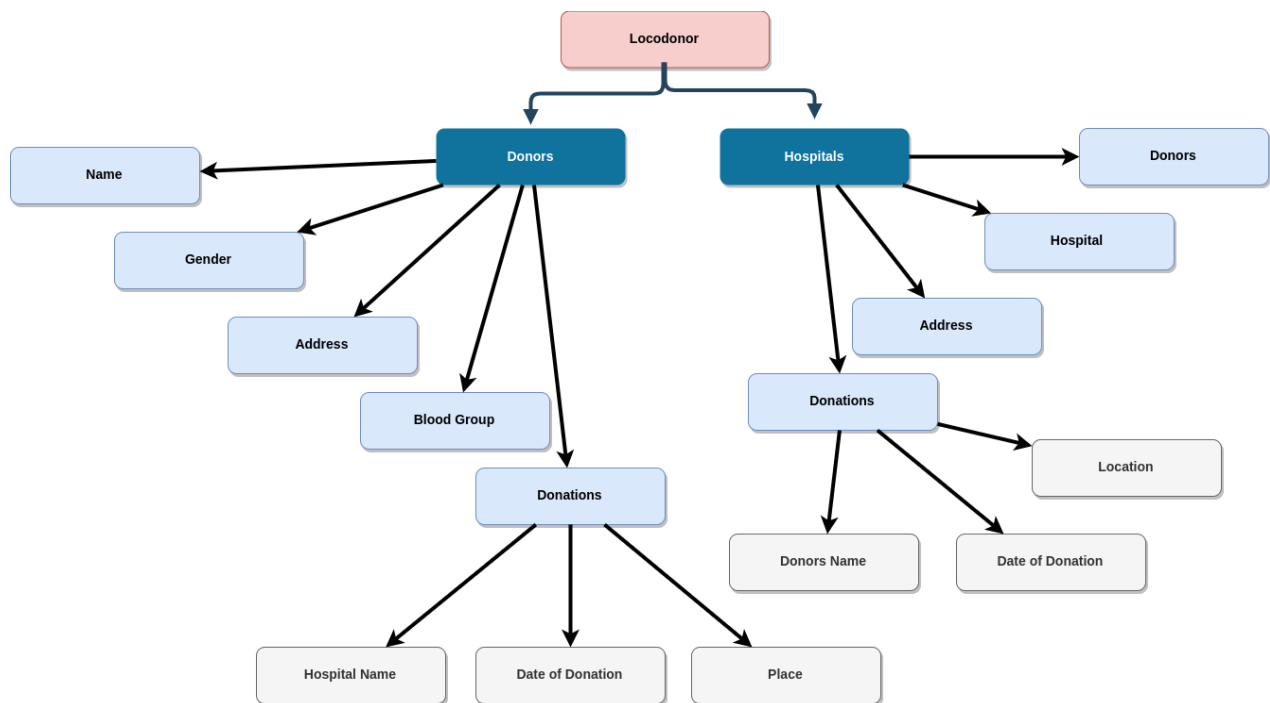
DATABASE ANALYSIS AND DESIGN

For the sake of this project the V-model was adopted: V- Model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development.



The various phases of the V-model are as follows:

- Requirements like BRS and SRS begin the life cycle model just like the waterfall model. But, in this model before development is started, a system test plan is created. The test plan focuses on meeting the functionality specified in the requirements gathering.
- The high-level design (HLD). The phase focuses on system architecture and design. It provides an overview of solution, platform, system, product and service/process. An integration test plan is created in this phase as well in order to test pieces of the software systems ability to work together.
- The low-level design phase is where the actual software components are designed. It defines the actual logic for each and every component of the system. Class diagram with all the methods and relation between classes comes under LLD. Component tests are created in this phase as well.
- The implementation phase is, again, where all coding takes place. Once coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.



IMPLEMENTATION DETAILS

The system was interfaced with Mongo DB at the background, the database holds the data and interactions that is going on in the portal. An extremely enriching and friendly welcome screen which is essential to facilitate a good user interface and experience.

- The opening page
- One donor with the same id and password cannot register twice.
- All the details were entered at the time of SignUp are stored in the database management system and are referred to time and again for the purpose of reference.
- All details are kept secret.
- Hospitals can only register through contacting the site administrator for security reasons.
- The donors can see all the details of their past donations and will also be advised when they can donate the blood again.
- Hospitals can search all the nearby donors by specifying their location, including the blood type and all other necessary details.

This system is a web based system so the following are met:

- A PC (desktop, laptop, Tablet or other portable devices)
- Jupyter notebook for creating modules
- Database Design (MongoDb)
- For Interface (HTML,CSS, JavaScript, Bootstrap)
- Testing (Flask server, Django)
- Reporting Tool (error Reporting/log file)

TESTING (DATA VALIDATION)

- Donor Registration: Donors Registered at their convenient locations, at their own time whenever they want, then hospitals can use the portal to gather information like donors in there region, how much blood is there in their blood banks and whether they need more blood. For this donors need to create an account as well as register. Apon login donors are able to see their information and know when was the last time they have donated blood and where. This crucial since donors can donate blood only in a gap of 3 - 4 months. Since our primary concerns where security, username and id type of login was made to allow donors to access their information with information of previous blood donations and stats.
- Hospital Validation/Account Activation: The hospitals in contact officials of locodonor in order to get access to our database in order to maintain privacy and give medical practitioners to save lives. Since our primary concerns where security, username and id type of login is made to only allow hospitals to access donors information.
- We performed unit testing on the program where in each module was tested on Jupyter notebook before it was integrated with the web framework.
- The flask web frameworks testing was done on flask server to check whether
 - ❖ The GUI is suitable and eligible to the users.
 - ❖ The instructions provided on how the system is to be used is clearly understood
 - ❖ The webpage works smoothly and as intended.
- Then an acceptance testing was performed where the end users where our own batchmates. Many bugs were identified and rectified immediately.

FRONT END, BACK END AND CONNECTIVITY

1. Front end

- HTML
- CSS
- Java Script
- Bootstrap

2. Back end

- MongoDB

3. Connectivity

- Flask – A python web framework

The below image shows the python mongo connectivity

```
1  from pymongo import MongoClient
2  class Database:
3
4      def __init__(self,ip='localhost',db=None):
5          try:
6              conn = MongoClient(ip)
7              self.db = conn.local_donor
8              print('Connection successful.')
9          except:
10             print('Error in connection.')
11             self.db = None
12
13     def insert_hospital(self, hname, husername, hpassword, hadd, hcity, hstate):
14         print("inserting")
15         collection = self.db.Hospitals
16         a = collection.find( { 'username': { '$exists': True, '$in': [husername]
17         val = []
18         for x in a:
19             val = x
```

CONCLUSION AND FUTURE WORKS

In this project we propose a web based blood donation database which keeps all the necessary details of donors and the respective donations. This will help drastically in emergency situations if any patient needs blood immediately. To implement the technology known to man in systems like these, to improve our services is a necessity. Projects like this will help to automate an important sector of our Medical Services. By our Blood Donation Management System, the efficiency and the time saved would be significant. Our future plan is to make our system more effective and also we want to add some application to our system like SMS alert system, text message notification over mobile, facebook notification and help alerts and implement an android application.

This project, being reliable, convenient, user friendly, secure (as datasets are less revealed to user-end), has much more fields in future for development, where the models developed can be deployed. Also many of the technologies used in this project are easily scalable. This project will find its way into future where

- a) Time delay is to be quantized
- b) Automation is required
- c) Security and abstraction is required
- d) Emergency situations can be resolved quickly

REFERENCES

- <https://ieeexplore.ieee.org/document/8537351>
- <https://ieeexplore.ieee.org/document/8471988/versions>
- <https://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty#targetText=Software%20testing%20is%20the%20process,done%20during%20the%20development%20process.>