**Title:** Blood Donation System (BDS)

**Team 9:**

Juan Tamayo

Bienvinido Onia

Nathan Luc

Karandeep Singh

**Miniworld**

Project Development Management System that allows access to every step of the developing process. It facilitate the user to access any previous version or data of the project to be used at the administrator’s discretion.

The Blood Donation System allows for easy lookup of blood types of particular patients, and provides an inventory of which blood types are available through the blood banks across the country. At first the blood donor registers to donate blood. The registration process includes providing a name, address, phone number, and a donor identification number. Next, the screening process occurs where a series of questions regarding health, travel, and other necessary information will be conducted, along with a health exam to ensure the donator is in fit shape to donate blood. If all is well, the donator will have his blood drawn and have the blood sent to a blood bank for further tests. A donator is only allowed to donate blood every 60 days.

**Purpose of Application/database and Intended Users**

The purpose of this application/database is to keep track of the software development process, and provide a safe and stable restore point in the event of code/software malfunction.

Intended users could include but not limited to: Project Advisors, Managers, Code Developers, QA .

The purpose of the blood donation system database is to keep track of the data for a particular Patient in need for a blood transfusion, a blood donor who is willing to donate their blood for the patient, and the data of a blood bank where the blood will be preserved for future patients. Intended users could include but not limited to: Doctors and

**Objects/Actors/Roles and plan functionality and operations**

The objects within this project are File, Version Control, public and private repositories, User Requests, commits, and logs. The actors in this project are the user or contributor. Users can play the role of team, team members, admins, QA, and developers.

A Contributor can play as an admin, QA, or a simple developer. A developer can make commits, but must request for push & pull requests from the admin. QA can review code and logs set by developers. Admins can manipulate any and all of the objects. Different teams can collaborate with admins, QAs, developers, and other contributors.

The objects will be the different blood types (A,B,O,AB), the location, and health of the patient. The actors are the patients, donors, doctors, and nurses.

The doctor will have information of the patients and the donor. The doctor will administer the blood donation and can have multiple patients. The doctor will have a location in order to administer the blood.

The nurse will also have the information of the patients and donors. The nurse will help with administering the donation. The nurse can help one doctor at a time and be assigned to multiple patients at a time.

The patient will have the blood type of the donor and will receive the blood. The patient will have one doctor and one nurse.

The donor will give blood to a patient and will have one doctor and may or may not have a nurse.

**Scenarios**

Alex wants to donate blood so he goes to a blood bank for extraction. He is asked to complete a set of electronic forms. These forms contain questions such, name, address, age, medical history, family history, blood type, identification number. After the form is completed, the data collected is compared with the criteria that will categorized the donor and determine whether his blood is fit or not to be stored on the blood bank. If the blood is fit, the donor’s information will be stored on the database the donor’s data and the location of where the blood is stored.

A doctor needs blood for one of his patients, he requests a nurse to locate the blood needed by looking for the location of a specific blood type. The nurse access the blood bank search engine. She inputs the blood type requested. After all options are presented to the nurse, she will obtain the information of the location of the blood and then proceed to obtain the blood. After pick up, the amount on stock on the location of pick up must be updated by subtracting the amount being picked up. The blood will be taken back to the doctor where it will be used on the patient.

Alex wants to create a new project called Project 1 for collaboration, so he uses the Not Git app and creates public repo. After creating the initial public repository, he is automatically given the admin role for the project. Developer Kevin wants to contribute to the project, so he can openly pull the project since it is public, and Kevin can make a push request to Alex to update any code that has recently been committed. Admin Alex has the control to either block or accept developer kevins push request to push the updated code to the master. Kevin can openly make local commits to the repository he as pulled, but since Alex is the admin, she can only allow the commits to become permanent. Kevin is also allowed to review logs and previous commits made in the repository.

Jane wants to create a new private project that she can only contribute to, so she uses the Not Git app and creates a private repository. Jane becomes the admin and the only contributor of the project. An unknown QA named Chad request access to see Jane’s code in order to review the code. But Jane does not know QA Chad so she rejects the request to see the code.

The developers of project 1 finishes the project and hires a team of QAs to optimize their code. The team of QAs sends a pull request to the admin of project 1 and the team successfully pulled the code. The team of QAs run the code numerous times and finds some bugs and review past commits and review the log to find the origin of the bugs. The team of QAs make minor changes in order to make the code work. The team commits and then sends a push request to the admin. The admin accepts the changes and the changes are made public and the logs are updated.