Functional requirement:

1. Donor account system:
2. The system should let donor create, login and change account (user id and password)
3. The system should let donor upload his/her personal information
4. The system should let donor declares intents to donate.
5. The system should display the donation history to the donor.
6. The system should take into account donor’s donation history when this donor declares intents to donate.
7. The system will notify the appropriate donor when new request appears.
8. The system will also notify the donor if he/she is chosen to donate blood.
9. Requester account system:
10. The system should let requester donor create, login and change account (user id and password)
11. The system should let requester upload his/her personal information
12. The system should let requester upload his/her detailed request information.
13. The system will notify the requester when it find the appropriate donor.
14. The system will display the status of the request.

Non-Functional requirements:

1. Operational Requrements
   1. An admin web interface should be provided to manage users.
   2. System should have user friendly UI.
   3. System should be able to connect to backend database.
   4. System should handle invalid input on both backend and frontend.

2. Performance requirements:

1. System should allow more then 100 users online at the same time.

2. System should be avaliable 24/7.

3. Security Requirement

1. Backend should check for invalid input.

2. Backend should handle database error and should filter input for potential abuses (e.g. SQL inject, XSS, CSRF)

3. User should not be able to access the services that they don't have access to. (e.g. Donor should not be allow to use Requester's API)

4. User's password should be stored in hashed format, not plaintext.