**Requirements Specifications**

**<P11>:<Blood Donating System>**

**<team member names & ids>**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 22100082 | Muhammad Saad |
| 22100221 | Ahmad Aslam |
| 22100224 | Muhammad Bilal |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction & system actors | 5 | 5 |
| Use case diagram | 10 | 10 |
| Use case descriptions | 20 | 12 |
| Class diagram | 20 | 12 |
| Sequence diagram | 20 | 17 |
| State diagram | 5 | 5 |
| Non-functional requirements | 5 | 5 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template | 5 | 5 |
| Late submission penalty | -20 |  |
| **Total** | **100** | **81** |
| Review | 20 |  |
| **Grand Total** |  |  |

**Table of Contents**

1. Introduction 4
2. System Actors 5
3. Use Cases 8
   1. Use Case Diagram 8
   2. Description of Use Cases 8
4. Class Diagram 25
   1. Diagram 25
   2. Description 25
5. Sequence Diagrams 26
6. State Diagrams 27
   1. Diagram details 27
   2. Diagram 27
7. Non-functional Requirements / Quality Attributes 29
8. Who Did What? 32
9. Review checklist 32

# Introduction

Blood Donating System is an Android, iOS and web-based app that provides a platform to blood donors and receivers for successful blood donation. It is going to make blood donation accessible to the one in need. The potential users of the app are blood donors, NGOs and blood recipients.

The overall objectives of the app are:

* Expedite the process of blood donation.
* Encourage NGOs to provide blood donors to the system.
* Through a system of ratings, points and rewards, encourage more and more donors to use the app.
* Provide an easy communication between blood donors and blood receivers.
* Make blood donation easily accessible to the one in need.

In the app, a recipient is going to ask for a donation of blood. The request will be pushed to a newsfeed where a donor will willingly respond to the request. A chat module will open between the donor and the receiver where they can communicate regarding the blood donation. On successful blood donation, the donor will be rewarded with points.

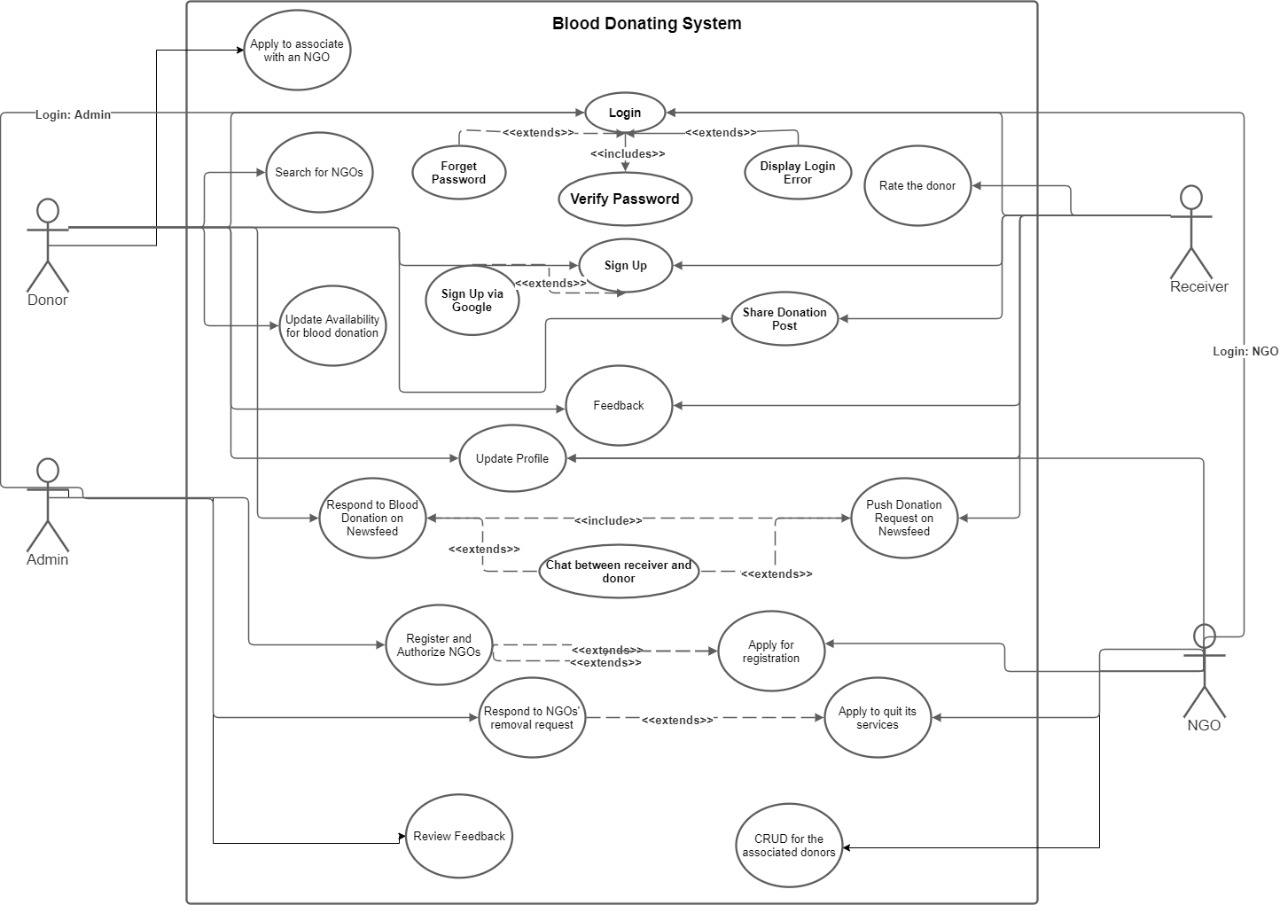
The app is going to be android, iOS and web-based. The main technologies that we are going to use for our app are Ruby on Rails, React and React Native.

# System Actors

|  |  |  |
| --- | --- | --- |
| **Actor** | **Description** | |
| Admin | Admin has full knowledge and control of the system. Admin acts as a moderator and has direct access to the whole system. | |
| **Functionalities** | | |
| Admin can:   * Register NGOs * Authorize NGOs * Sign in to his/her account * View feedbacks/complaints/reports regarding the app * View logs in order to detect any system failure or error * Log out of the app * Make internal changes in the app. | | |
| NGOs | NGOs will provide blood donors to the system which will in result assist in blood donation. In result of their services, they will receive more points, ratings and popularity in the app. | |
| **Functionalities** | | |
| NGOs can:   * Register their accounts * If they are registered, sign in their accounts * Provide data about their blood donors to the systems and provide its blood donors their login details. * Create, Read, Update and Delete data of their blood donors * Notify blood donors about blood donations request according to their eligibility and location * Respond to association requests if a donor wants to join that NGO * View user activity of their blood donors * Edit their own profile * If they want to drop their services, request the admin to remove them from the system | | |
| Blood donors | Blood donors are the actual main service provider to our system. They are going to donate blood to someone in need of blood. On fulfilment of a request, they will receive rewards and points. | |
| **Functionalities** | | |
| Blood donors can:   * Sign up their accounts * If they are registered, sign in their accounts * Respond to blood requests on newsfeed * Chat with a blood receiver whose request they have accepted * View/Edit their profile * Request an NGO to join with it * Look up NGOs according to their names * Log out of their accounts * View requests of blood receivers | | |
| Blood receiver | | They are the beneficiary of our service as they are the one who are in need of donation of blood. |
| **Functionalities** | | |
| Blood receivers can:   * Sign up their accounts * If they are signed up, sign in their accounts * Push notification for blood request on newsfeed * Chat with blood donors * View/Edit their profile * View notifications regarding the blood donors who have accepted their request | | |

# Use Cases

## Use Case Diagrams



## Description of Use Cases

[I have reviewed some use cases. You should look at the use case description template provided and update the use case details accordingly. There are issues with description, and alternate and exception paths of many use cases. See my comments below for selected use cases.]

**Sign Up:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 1 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows the user (blood donor/receiver) to register into the system |
| **Pre-conditions** | | Users must have a valid email account/phone number. |
| **Post-conditions** | | The user is successfully registered into the system. |
| **Step #** | **Typical Course of Action** | |
|  | The user opens the web app or mobile app. | |
| 1 | The system will ask email account or phone number | |
| 2 | After entering email or phone, Users will receive a code through email/phone number, respectively [These are multiple steps. Both user actions and system actions are mixed up. User actions and system responses should be mentioned in separate steps as mentioned in the use case template.] | |
| 3 | The system asks for the code. | |
| 4 | Upon successful entry [It should be clear who enters data and what is entered], the system will prompt to set up a strong password (combination of digits, characters and alphabets) for their account. | |
| 5 | User will click “Register” button and system will show pop up message accordingly. | |
|  | **Alternate Courses of Action** | |
|  | User can use SSO (sign up with google or Facebook) to register. [At which step of the above typical course of action, this step triggers?] | |

**LOGIN:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 2 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows a user (blood donor/receiver) to Login into the system |
| **Pre-conditions** | | Users must have registered email or phone number. |
| **Post-conditions** | | The user is successfully Logged into the system. |
| **Step #** | **Typical Course of Action** | |
| 1 | The user opens the web app or mobile app. | |
| 2 | The system will ask email account or phone number | |
| 3 | System will check for email/phone and password combination. In case of in correct input system will show abstract error message. | |
| 4 | Upon successful entry, user will be redirected to the news feed. | |
|  | **Alternate Courses of Action** | |
|  | User can use SSO (login with google or Facebook) to login. | |
|  | **Exception Path** | |
|  | User can click forget password button, if he/she has forgotten password. Then course of Action for “Use Case-3” will be followed. | |

**Forget Password:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 3 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows the user (blood donor/receiver) to reset their password. |
| **Pre-conditions** | | Users must have registered email or phone number.  User have clicked on the “forget password” button and redirected to new screen. |
| **Post-conditions** | | The user has successfully reset their password and password for the respective user is updated in the system. |
| **Step #** | **Typical Course of Action** | |
| 1 | The system will ask user to enter the email/phone. | |
| 2 | System will check for the email. If entered email/phone does not exist then system will show an error message. | |
| 3 | Upon correct entry of the email/phone, user will be asked to click the link in the email or enter code sent to phone. | |
| 4 | After entering the correct code or clicking the link in the email. User will be asked to enter new strong password. | |
| 5 | System will show correct message and will redirect user to login screen. | |

**Sign out:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 4 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows the user (blood donor/receiver) to logout from their account. |
| **Pre-conditions** | | Users must have logged in to their account. |
| **Post-conditions** | | The user has successfully logged out from the account |
| **Step #** | **Typical Course of Action** | |
| 1 | User will click the profile tab. | |
| 2 | Dropdown will appear. | |
| 3 | Upon clicking the log out option in the dropdown, user will be redirected to the newsfeed. | |
| **Step #** | **Exception Path** | |
| 1 | If user clear the web-browser cache or app memory in the mobile then user will automatically be signed out. | |

**Update Profile:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 5 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows the user (blood donor/receiver) to keep their profile up-to-date. |
| **Pre-conditions** | | Users must have Logged in to their account. |
| **Post-conditions** | | User profile is successfully updated. |
| **Step #** | **Typical Course of Action** | |
| 1 | Upon clicking the profile button, drawer containing the user information will appear. | |
| 2 | User will click on the edit icon. | |
| 3 | User will be redirected to the expanded view editable profile will appear. | |
| 4 | User will also be able to update their password by click the “change Password” link. | |
| 5 | User will be asked to enter current and new password. | |
| 6 | After entering new inputs, like new name or profile picture, users will click save button to successfully update their profile. | |
| 7 | User will be redirected to main newsfeed page. | |

**Feedback:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 6 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows the user (blood donor/receiver) to give feedback regarding the app. |
| **Pre-conditions** | | Users must have Logged in to their account. |
| **Post-conditions** | | User’s feedback is successfully recorded. |
| **Step #** | **Typical Course of Action** | |
| 1 | Feedback button will be available in the footer of every screen. | |
| 2 | User will click the feedback button. | |
| 3 | After giving the ratings and any suggestions user will click submit button to give feedback. | |
| 4 | User will be redirected to the previous state. | |
| 5 | User will be able to give more than one feedbacks. | |

**Post Blood donation requests:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 7 |
| **Actor** | | Blood receiver |
| **Purpose** | | The use case allows the user (blood receiver) to post blood donation request. |
| **Pre-conditions** | | Users must have Logged in to their account. |
| **Post-conditions** | | Blood donation request is successfully created and displayed on the newsfeed. |
| **Step #** | **Typical Course of Action** | |
| 1 | User will click the Create request button. | |
| 2 | System will ask user to enter all required details of the request. Details will include location, blood group type etc. | |
| 3 | After entering all the required details, user will click the post button. | |
| 4 | The donation request will be pushed on the newsfeed. | |

**Receive and Respond To blood donation request:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 8 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows a user (blood donor/receiver) to receive and respond blood donation request. |
| **Pre-conditions** | | Users must have Logged in to their account. |
| **Post-conditions** | | User is successfully able to receive blood donation request.  User is successfully able to respond (accept) blood donation request. |
| **Step #** | **Typical Course of Action** | |
| 1 | When the blood donation request will be posted, system will send notification to nearby eligible blood donors. | |
| 2 | Upon receiving the notification, Blood donor can click the accept button on the notification to accept the blood donation request. | |
| 3 | System will send notification to receiver about blood donor acceptance. | |
| 4 | System will add blood donor and receiver in the chats of each other | |
| 5 | Blood donor and receiver will use chat to share further details. | |
| **Step #** | **Alternate Course of Action** | |
| 1 | On the news feed all the nearby blood donation requests will be visible. | |
| 2 | User will click on the accept button to accept request. | |
| 3 | Blood donor and receiver will chat to share further details. | |

**Chat:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 9 |
| **Actor** | | Blood donor and receiver |
| **Purpose** | | The use case allows the users (blood donor and receiver) to chat with each other. |
| **Pre-conditions** | | Users must have Logged in to their account.  Blood donor must have accepted the donation request of the receiver. |
| **Post-conditions** | | Users are successfully able to chat |
| **Step #** | **Typical Course of Action** | |
| 1 | After accepting the request, a chat module will open up between the donor and the receiver. | |
| **Step #** | **Alternative Course of Action** | |
| 1 | On clicking the chat button, all previous and existing chat boxes will open up. | |
| 2 | Upon clicking on one of the available contacts, the existing chat drawer will open up between the users. | |

**Share:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 10 |
| **Actor** | | Blood donor/receiver |
| **Purpose** | | The use case allows a user (blood donor/receiver) to share post on other platforms like WhatsApp and Facebook. |
| **Pre-conditions** | | Users must have Logged in to their account.  The post is already up on the newsfeed. |
| **Post-conditions** | | User is successfully able to share the post. |
| **Step #** | **Typical Course of Action** | |
| 1 | When blood donation request is visible on the newsfeed, user can share the request on the other social media platforms | |
| 2 | User will click on the share button. | |
| 3 | A text message containing all the request details (e.g. location, blood group type) will be created by the system. | |
| 4 | Use will copy the generated message and share it on other platforms like Facebook and WhatsApp. [Will this sharing be done through the system or will the system only generate the text? There appears to be more steps here than you have described.] | |

**Search and Associate with NGO:**

|  |  |
| --- | --- |
| **Use Case ID:** | 11 |
| **Actor** | Blood donor |
| **Purpose** | The use case allows the user (blood donor) to send onboarding request to NGO. |
| **Pre-conditions** | Users must have Logged in to their account. |
| **Post-conditions** | User will successfully be able submit onboarding request to the NGO. |
| **Step #** | **Typical Course of Action** |
| 1 | User will search the NGO by entering NGO name/location in the search bar. |
| 2 | NGO profile will appear. |
| 3 | System will ask actor for details like recent blood donation date and required documents like health certificate. |
| 4 | User will provide all the details. |
| 5 | User will click associate button to send association request to NGO. |

**View and update eligibility.**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 12 |
| **Actor** | | Blood donor |
| **Purpose** | | The use case allows a blood donor to view and update his/her eligibility of donating request |
| **Pre-conditions** | | User must be logged in to the account |
| **Post-conditions** | | Blood donor is able to successfully update eligibility. |
| **Step #** | **Typical Course of Action** | |
| 1 | When blood donor donates, the system will confirm from receiver whether donor has donated the blood. | |
| 2 | Blood receiver will respond either yes or no. | |
| 3 | If the response is “yes” system will automatically update blood donor eligibility. | |
| **Step #** | **Alternate Course of Action** | |
|  | Blood donor will also be able to update his/her eligibility. In case blood donor has participated in blood donation outside the app then he will update eligibility himself. | |

**Register:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 13 |
| **Actor** | | NGO |
| **Purpose** | | The use case allows the NGO to Submit registration request. |
| **Pre-conditions** | | NGO is not already registered in the system. |
| **Post-conditions** | | NGO is successfully able to submit request. |
| **Step #** | **Typical Course of Action** | |
| 1 | NGO will go to the web-app and click the \*register as an NGO\* | |
| 2 | NGO will be redirected to new page containing NGO registration form. | |
| 3 | NGO will provide all the details required in the form. For example: NGO will provide name of founder, NGO’s contact number, NGO’s email address, Registration certificate etc. | |
| 4 | After providing all the required information, NGO will submit registration application. | |
| 5 | Success pop up message will appear. | |

**Create donor:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 14.1 |
| **Actor** | | NGO |
| **Purpose** | | The use case allows the NGO to create a blood donor account. |
| **Pre-conditions** | | NGO must have blood donor onboard. |
| **Post-conditions** | | NGO will successfully be able to create a blood donor account and notify that donor. |
| **Step #** | **Typical Course of Action** | |
| 1 | NGO will click the Add blood donor button. | |
| 2 | System will ask for the details like email address and current eligibility(optional). | |
| 3 | Upon entering the details NGO will click the submit button. | |
| 4 | System will send a link to the specified email using which user will be able to set password for the account. | |

**Read donor:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 14.2 |
| **Actor** | | NGO |
| **Purpose** | | The use case allows the NGO to view blood donor account. |
| **Pre-conditions** | | NGO must have blood donor onboard. |
| **Post-conditions** | | NGO will successfully be able to view a blood donor account |
| **Step #** | **Typical Course of Action** | |
| 1 | NGO will search the blood donor. | |
| 2 | From the search results NGO will be able to click and view blood donor’s profile. | |
| **Step #** | **Alternative Course of Action** | |
| 1 | NGO will click the “onboard donors” button to view all the blood donors associated with the NGO. | |
| 2 | By clicking any of the blood donor, NGO will be able to view blood donor’s profile. | |

**Update donor:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 14.3 |
| **Actor** | | NGO |
| **Purpose** | | The use case allows the NGO to update blood donor account. |
| **Pre-conditions** | | NGO must have blood donor onboard. |
| **Post-conditions** | | NGO will successfully be able to update a blood donor account |
| **Step #** | **Typical Course of Action** | |
| 1 | NGO will search fort the blood donor. | |
| 2 | From the search results, NGO will be able to click and view blood donor's profiles. | |
| 3 | NGO will make updates like eligibility update and then click save to update the blood donor’s profile | |
| **Step #** | **Alternative Course of Action** | |
| 1 | NGO will click the “onboard donors” button to view all the blood donors associated with the NGO. | |
| 2 | By clicking specific blood donor, NGO will be able to open blood donor’s profile. | |
| 3 | NGO will make updates like eligibility update and then click save to update the blood donor’s profile | |

**Delete donor:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 14.4 |
| **Actor** | | NGO |
| **Purpose** | | The use case allows a NGO to terminate blood donor association. |
| **Pre-conditions** | | NGO must have blood donor onboard. |
| **Post-conditions** | | NGO will successfully be able to terminate association with a blood donor. |
| **Step #** | **Typical Course of Action** | |
| 1 | NGO will search fort the blood donor. | |
| 2 | From the search results NGO will be able to click and view blood donors profile. | |
| 3 | NGO will click the remove button. | |
| 4 | System will ask reason for the termination (optional). NGO will hit confirm button to remove association for the specific blood donor. | |
| 5 | System will send notification to the blood donor about the association removal. | |

**Respond to onboarding request:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 15 |
| **Actor** | | NGO |
| **Purpose** | | The use case allows the NGO to respond to onboarding requests. |
| **Pre-conditions** | | NGO must be registered to the system  The donor must have applied for the association request. |
| **Post-conditions** | | NGO is able to onboard blood donor. |
| **Step #** | **Typical Course of Action** | |
| 1 | Upon receiving the onboarding request, NGO will go through all the details, like health certificate. [How are onboarding requests received? Manually or through the system?] | |
| 2 | NGO will approve the onboarding request after reviewing details. | |
| 3 | Blood donor will be notified about the respond. | |
| **Step #** | **Exceptional Course of Action** | |
| 1 | If there is any problem in the details or documents. NGO will reject the onboarding request with message containing brief reason of rejection. [You must always mention the step in typical course of action where the exception would occur.] | |

**NGO Account Termination:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 16 [] |
| **Actor** | | NGO |
| **Purpose** | | The use case allows the NGO to request account termination. |
| **Pre-conditions** | | NGO must be registered to the system |
| **Post-conditions** | | NGO is successfully able to submit termination request. |
| **Step #** | **Typical Course of Action** | |
| 1 | In the NGO’s profile, NGO will click the Terminate Account button. | |
| 2 | An application form will appear. System will ask for some details like reason for termination. | |
| 3 | NGO will submit termination request by clicking the submit button. | |
| 4 | Request will be submitted and pop-up message will appear. | |

**Register and Authorize:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 17 |
| **Actor** | | Admin |
| **Purpose** | | The use case allows the admin to respond to NGO’s registration request. |
| **Pre-conditions** | | Admin must log in to admin panel.  NGO must have applied for the registration. |
| **Post-conditions** | | Request is approved or declined and NGO is successfully notified accordingly. |
| **Step #** | **Typical Course of Action** | |
| 1 | Admin will receive NGO registration request. | |
| 2 | Admin will go through all the details, like Registration certificate, provided by NGO. | |
| 3 | Admin will verify the details. | |
| 4 | Admin will grant access for the NGO and send an email to NGO’s email. | |
|  | **Exceptional Course of Action** | |
|  | In case admin need further details, Admin will ask NGO for more details via email. | |
|  | In case of fake details, Admin will decline the request and email will be sent. | |

**Terminate NGO:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 18 |
| **Actor** | | Admin |
| **Purpose** | | The use case allows admin to receive and respond NGO’s termination request. |
| **Pre-conditions** | | Admin must log in to admin panel. |
| **Post-conditions** | | Request is approved or declined and NGO is successfully notified accordingly. |
| **Step #** | **Typical Course of Action** | |
| 1 | Admin will receive NGO termination request. | |
| 2 | Admin will go through all the details, like reason for termination, provided by NGO. | |
| 3 | Admin will terminate NGO account and access of the NGO to system will be remove. | |
| 4 | Admin will send email about the termination of the account. | |
|  | **Exceptional Course of Action** | |
|  | If NGO want to terminate their account due to some system related problem, then admin will negotiate with NGO and take action accordingly. | |

**View system Log:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 19 |
| **Actor** | | Admin |
| **Purpose** | | The use case allows admin to view system logs. |
| **Pre-conditions** | | Admin must login to admin panel. |
| **Post-conditions** | | Admin is successfully able to view system log. |
| **Step #** | **Typical Course of Action** | |
| 1 | Admin will click the system log button. | |
| 2 | All the system log will be visible. System log will also contain log about the user activity. | |
| 3 | Admin will observe system logs containing potential error or warning messages. | |
| 4 | If needed, Admin will take action according to the logs. | |
|  | **Exceptional Course of Action** | |
|  | If system is not generating any logs, admin will identify and resolve problem. | |

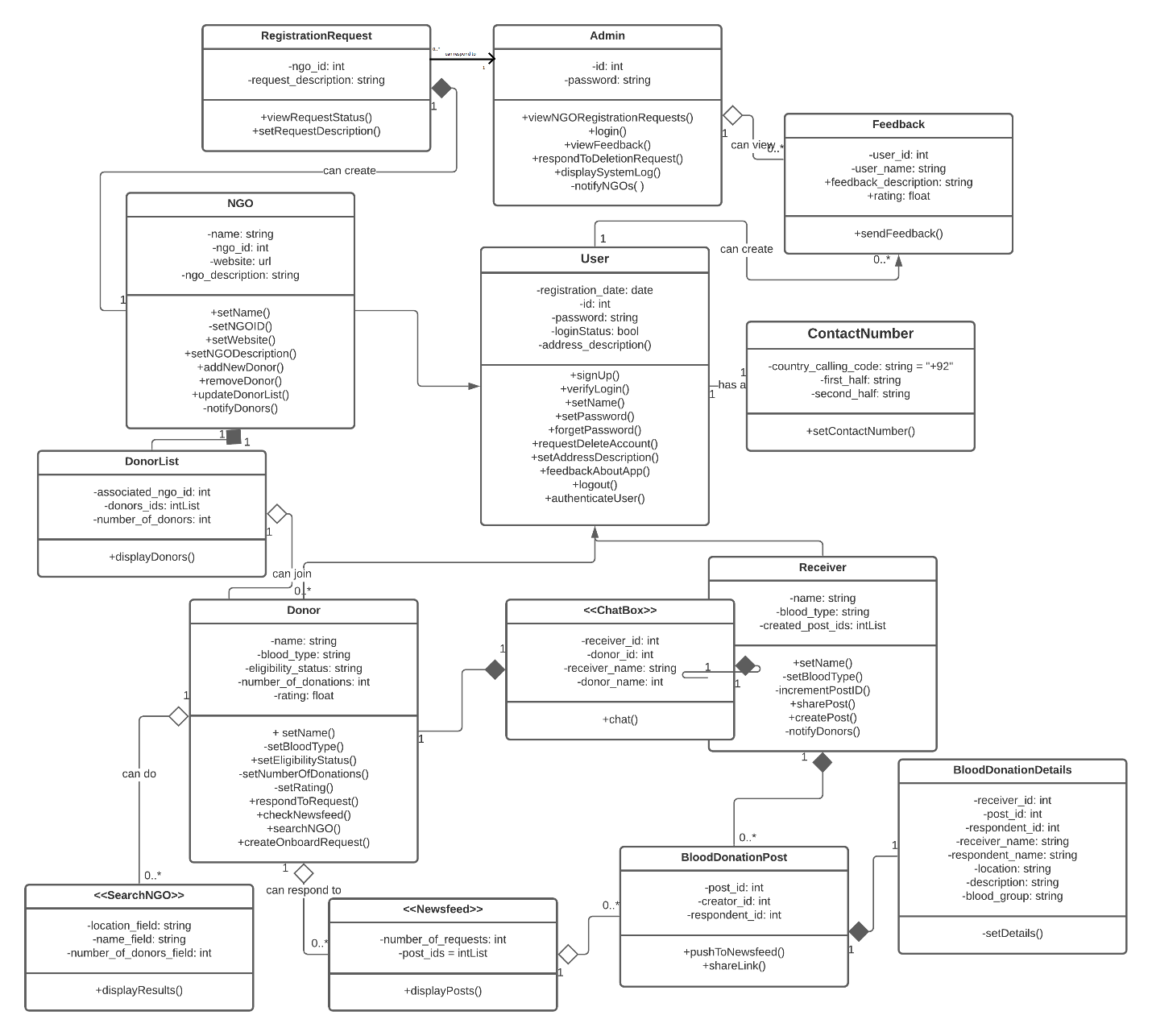
**View feedback:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | | 20 |
| **Actor** | | Admin |
| **Purpose** | | The use case allows admin to view analysis of feedback and make useful updates to the system. |
| **Pre-conditions** | | Admin must log in to admin panel. |
| **Post-conditions** | | Admin is successfully able to view feedback. |
| **Step #** | **Typical Course of Action** | |
| 1 | Admin will click the system view feedback button | |
| 2 | System will show basic analysis like average rating and graphs. | |
| 3 | Admin will also go through comments and make updates to the system if needed. | |

# Class Diagram

* Missing classes for security and authorization. Authorization must be implemented in a way that access rights can be granted or revoked on need basis.
* Some relationships are not clear. How is the “User” related to “Receiver” and “Donor”? Isn’t “Admin” a user? You have mentioned some details in the description, but your diagram should use standard notation to make the relationship between classes clear.
* Why does a “Registration Request” contain an “NGO”?
* In general, relationship between classes are not clear. You should rethink your system’s class model and draw a diagram using standard UML notation.

## Diagram



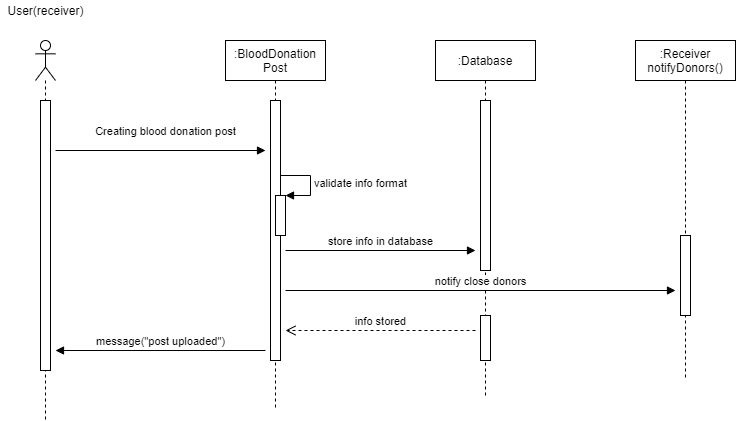
## Description

* **User** is the base class for other users: Donor, Receiver, and NGO. All the attributes under the user class are used by the derived ones with additional functions and attributes of their own. User class has an object named ContactNumber.
* **ContactNumber** class is used to type and encode contact number correctly. It consists of 3 parts, country calling code (with default value of +92: Pakistan), first half(03xx) and second half(-xxxxxxx). So, a complete contact number looks like: 03XX-XXXXXXX
* **Donor** is a derived class from user. The details of donor include his/her blood type, eligibility status (if he/she is eligible to donate blood), name, awards and other attributes. Donors will mainly respond to blood requests on newsfeed and communicate with the requestors.
* **Receiver** is a derived class from user which includes attributes like name, blood type and number of posts. Receivers can create blood donation requests and push them on newsfeed.
* **Chatbox** is the interface between donor and receiver when the donor has accepted the request of the receiver. It is meant to ease the communication between the users.
* **BloodDonationPost** are the posts that are created by receivers. Each post contains **BloodDonationDetails** that include the details regarding a blood donation.
* **Newsfeed** is the interface to show pending requests created by receivers.
* **SearchNGO** is the interface mainly used by donors to lookup NGOs. The interface includes filters that the donor can utilize in order to look up NGOs.
* **NGO** is a derived class from User, with more attributes and more functions NGO ID, website (if exists), description, and a donor list.
* Each NGO has a **DonorList** where its associated donors can be created, read, updated or removed from the donor.
* An NGO can request by sending a **RegistrationRequest** to the admin. After the acceptance of registration request, the NGO will be granted the CRUD operations for its donor list.
* Users like donors, receivers, and (maybe) NGOs can provide their **Feedback** by accessing that class.
* **Admin** is a super-user that can view feedback, check system logs, register NGOs, consider NGOs’ termination request, and other crucial operations.

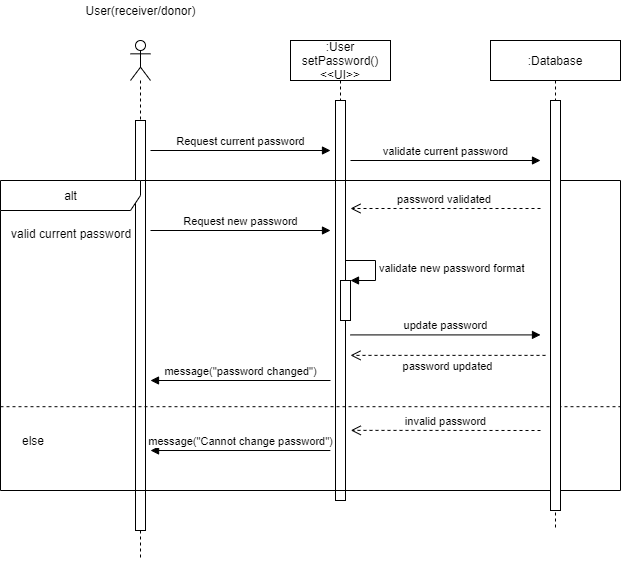
# Sequence Diagrams

**5.1 Create Blood Donation Request**

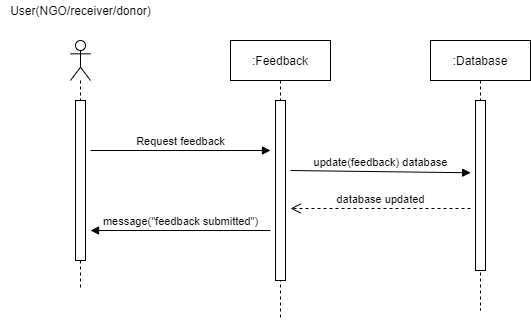
Why is there a gap between the call “store info in the database” and the return “info stored”? Shouldn’t these both be done in the same activation box?

****

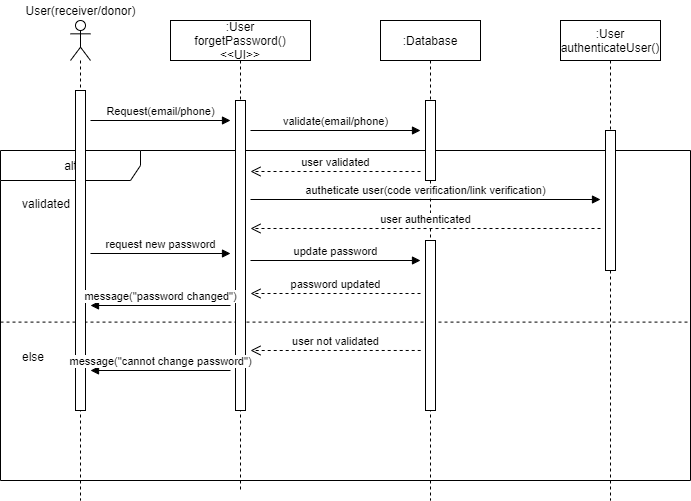
**5.2 Change Password**

****

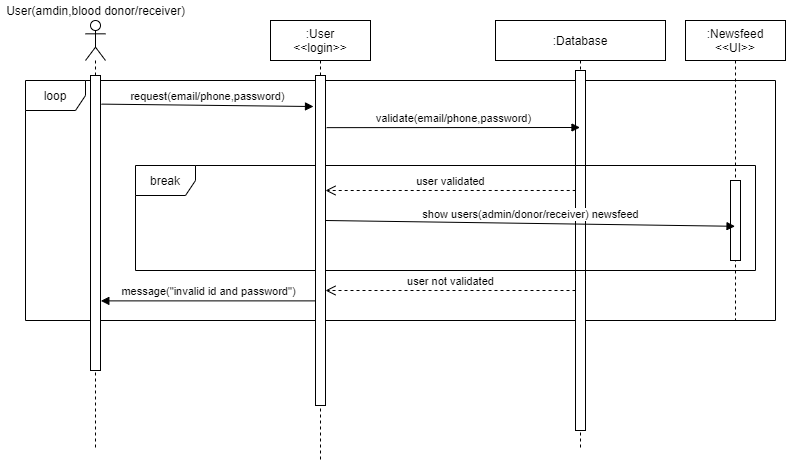
**5.3 Feedback**

****

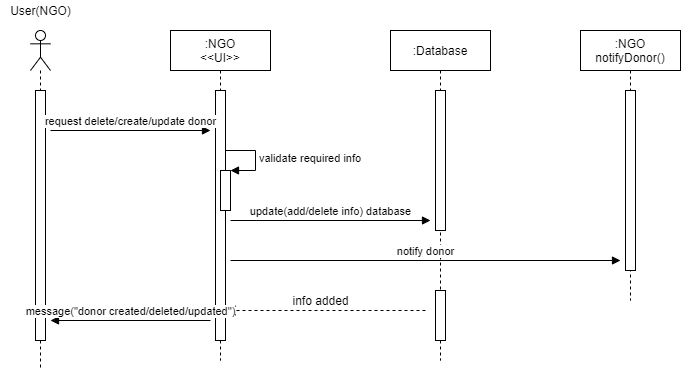
**5.4 Forget Password**

****

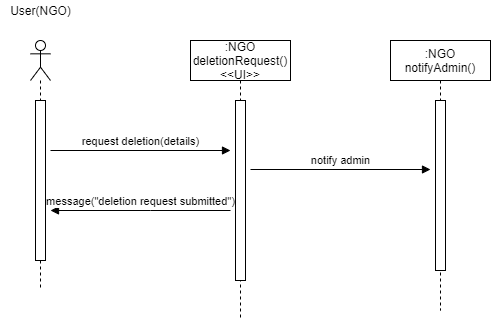
**5.5 Login**

****

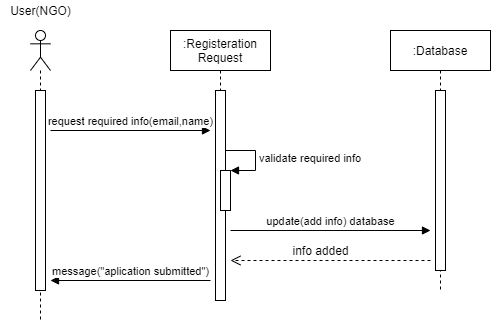
**5.6 Create, Update, Delete operations for NGO**

****

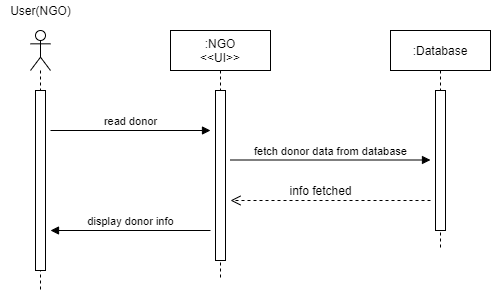
**5.7 NGO Account Termination Request**

****

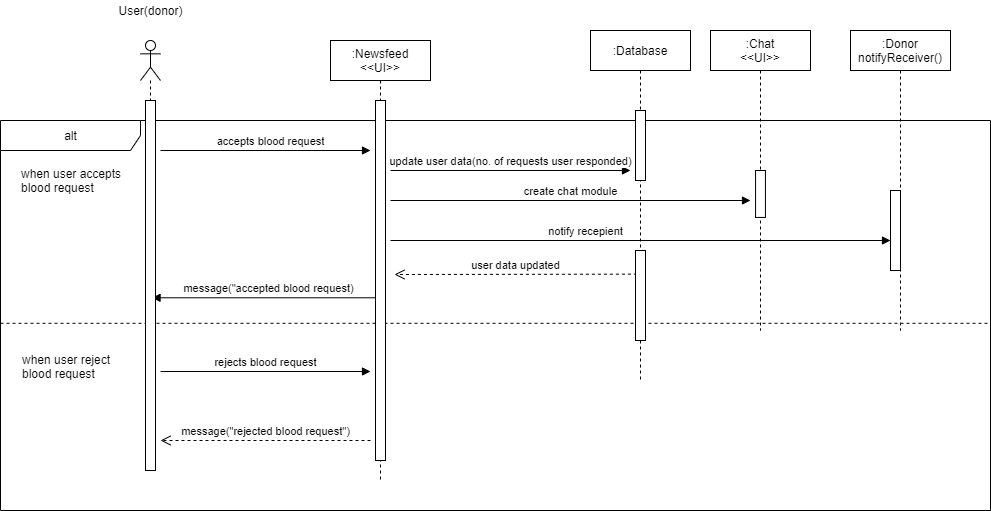
**5.8 NGO Registration Request**

****

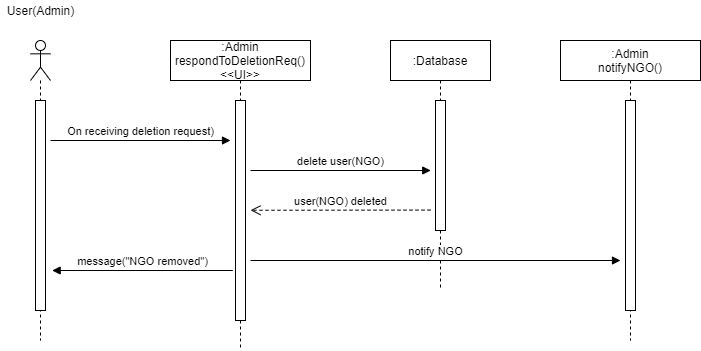
**5.9 Read Operation for NGO**

****

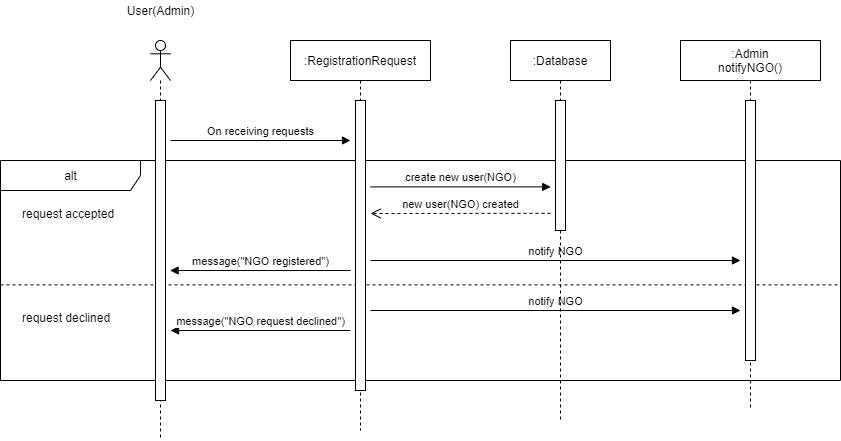
**5.10 Respond to Blood Request**

****

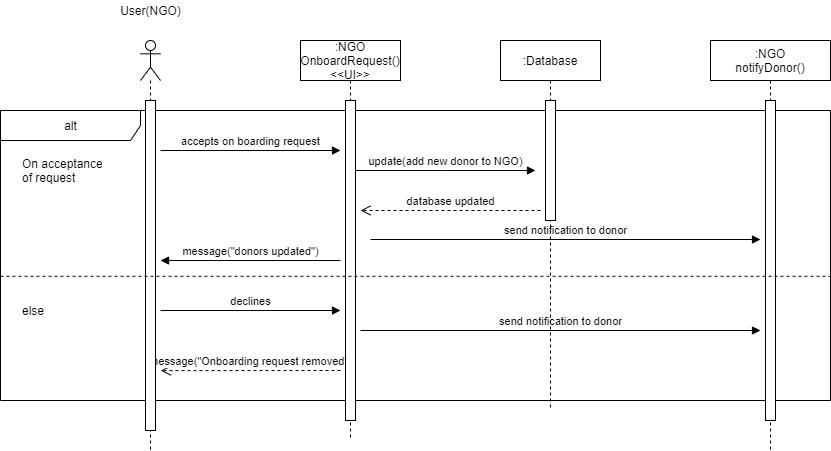
**5.11 Respond to NGO’s Deletion Request**

****

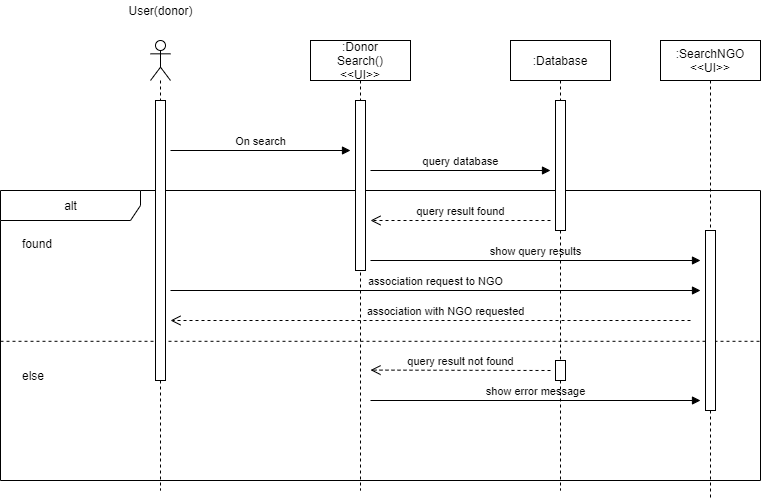
**5.12 Respond to NGO’s Registration Request**

****

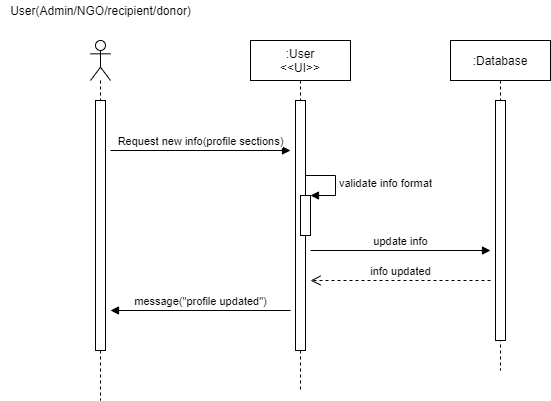
**5.13 Respond to Donor’s Onboarding Request**

****

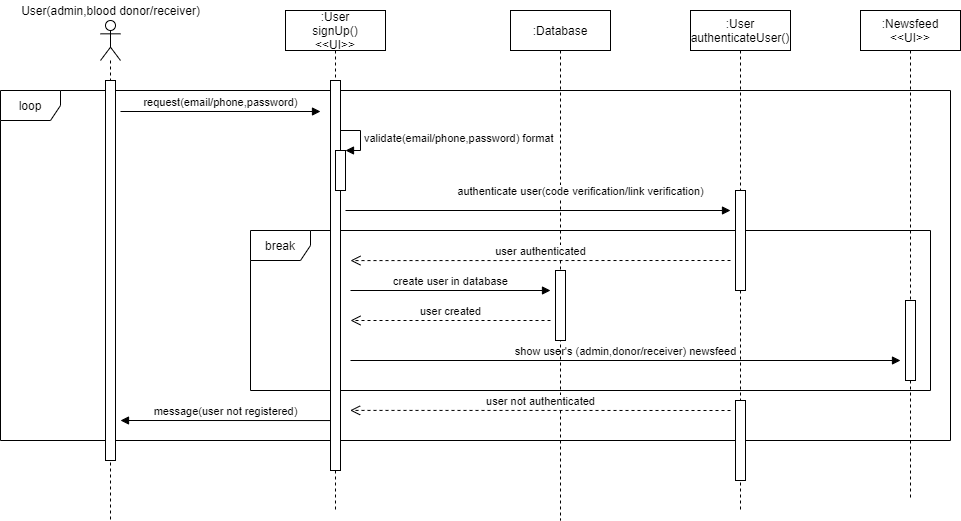
**5.14 Search NGOs**

****

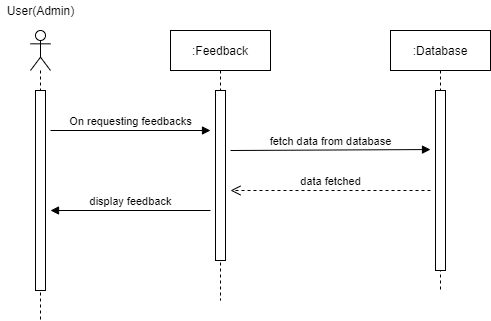
**5.15 Update Profile**

****

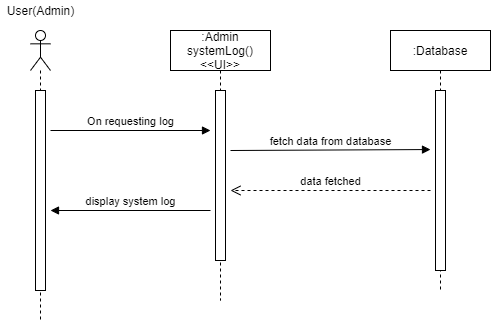
**5.16 Sign Up**

****

**5.17 View Feedback**

****

**5.18 View System Log**

****

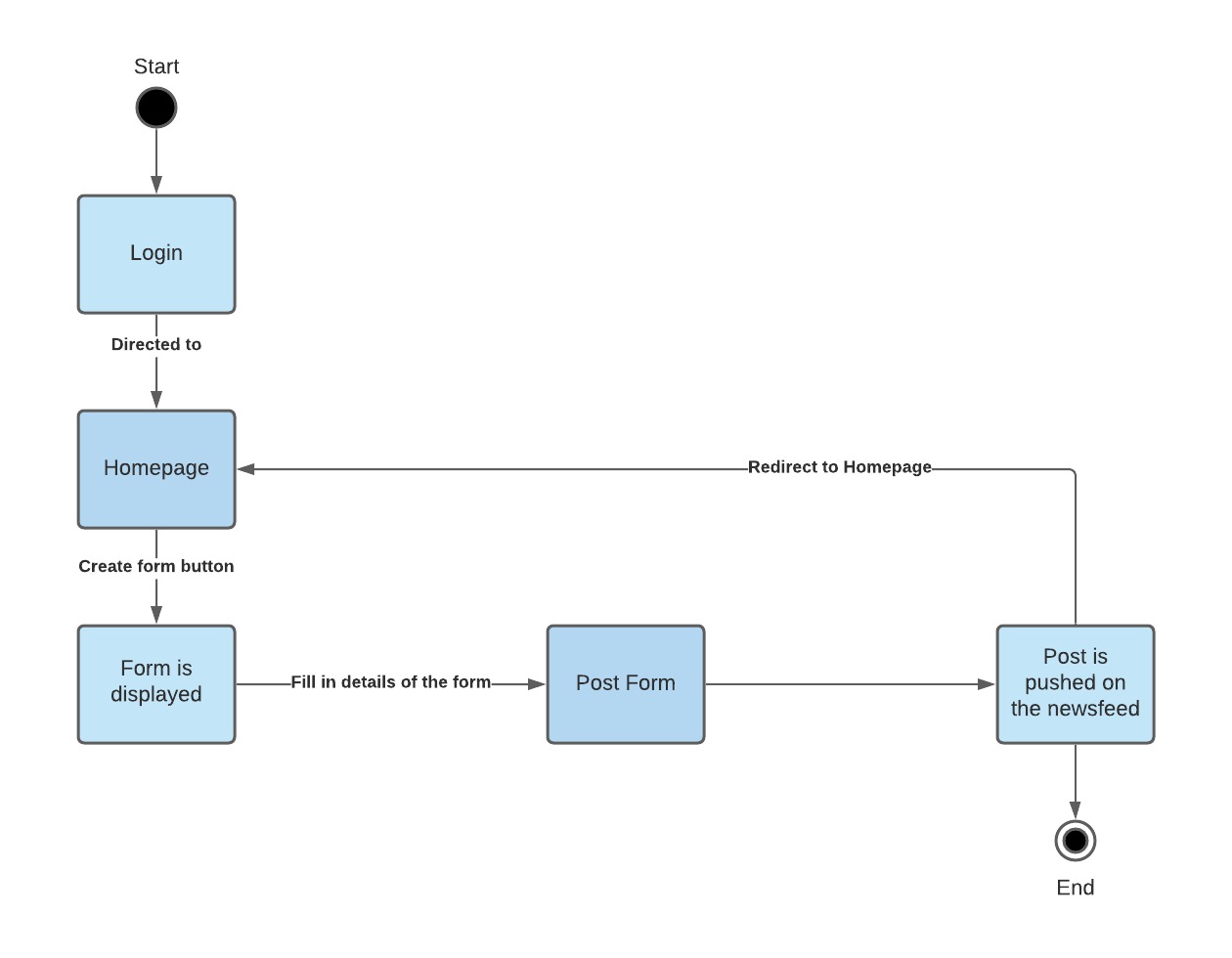
# State Diagrams

## Diagram details

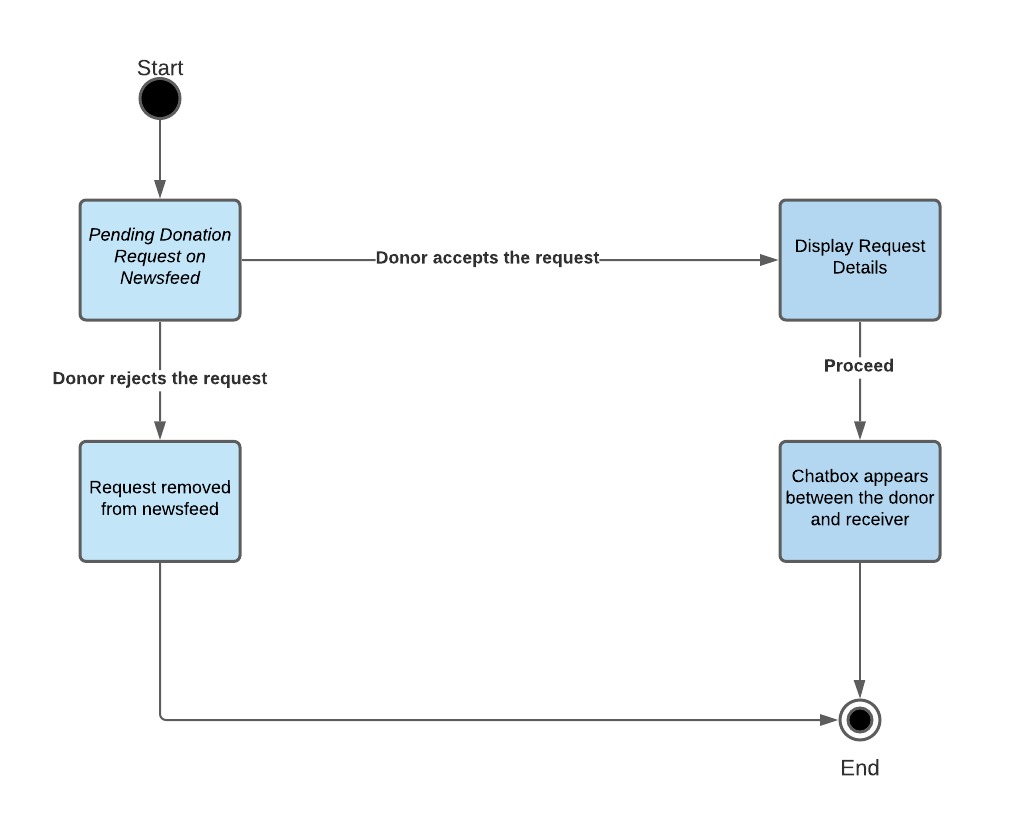
The objects for which the diagrams are drawn are of Receiver:createPost() and Donor:respondToRequest()

## Diagrams

***Receiver:createPost()***



***Donor:respondToRequest()***

******

# Non-functional Requirements / Quality Attributes

|  |  |
| --- | --- |
| **Sr#** | **Requirements** |
| 1 | App will be available for Android and IOS users. Moreover, web-based App will also increase the availability of the system. Moreover, MTBF (Mean Time Between Failures) will be more than 48 hours which means that system should not fail more than 4 times a week and MTTR (Mean Time To Repair) will be less than 5 hours which means system should be operational within 5 hours after the failure is detected. |
| 2 | Download size of the Application will not be more than 150 Mbs. |
| 3 | Any requested page should not take more than 7 seconds to load with average internet speed of 10 Mbps. |
| 4 | Upon entering the correct login details, user should be able to access the app within 5 seconds. |
| 5 | System should be scalable which means it should be capable of handling at least 10000 users simultaneously. |
| 6 | System will force user to create strong password, at least 8 character long containing combination of alphabet, number and character, to ensure account security. System will use Password authenticator to determine the strength of the given passwords based on the number of alphabets, characters and digits and display pop up message accordingly. |
| 7 | System will display correct and understandable error messages to user if user is doing something wrong. For Example: entering wrong blood group. |
| 8 | Any personal data of the user sent to server will be encrypted. This will ensure that the data is not usable in case of any data theft. |
| 9 | System will be able to correctly declare log about module crashing. |
| 10 | System will be robust against cyber security attacks. Input validation attacks like SQL injection, Buffer overflow and XSS attacks will be prevented using centralized validation approach rather than relying on client-side validation. |
| 11 | Current state of the database will be backed up bi-weekly and there will also be option for emergency backup in case on any security threat or hardware failure. |
| 12 | Reusability of the code will be ensured with different techniques like modularity, high cohesion and loose coupling. |
| 13 | Since the users of the app will be average internet user, usability of the system will be given prime importance. An average user will be able to post, search and create blood donations request within 5-steps. Moreover, use of consistency standards will help understanding the content easily but help and documentation will also be available. |
| 14 | For the early stages of the software launch portability of the software will be limited to only Android, IOS apps and chrome web-app. Later, the support for other web browsers like Safari and Firefox will be available. |
| 15 | Chat between the blood donor and blood receiver will be encrypted so that the users feel safe while sharing contact information in the chat. |

# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Muhammad Saad Azam | Class Diagram and description, Use case diagram, State diagrams |
| Muhammad Bilal | Sequence Diagrams, Introduction |
| Ahmad Aslam | Non-functional Requirements, Use Cases Description, System Actors |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Introduction | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |
| System Actors | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |
| Use Cases | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |
| Class Diagram | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |
| Sequence Diagrams | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |
| State Diagrams | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |
| Non-functional Requirements | Ahmad Aslam, Muhammad Saad, Muhammad Bilal |