BNet: A Multi-Platform Bot Framework with Cost-Optimized Dual-Layered Screening for Enhanced Blood Donation via Social Media Integration

In recent years, online social groups and communities have expanded rapidly with the growth of social networking sites. Users actively create, share, and circulate urgent blood donation requests within these groups during critical moments. However, locating these urgent requests quickly amid abundant streams of messages is challenging for donors. Prior research indicates that existing Blood Donation Systems are primarily app-based with limited reach. This reveals a gap in effectively managing blood donation requests within social media groups. In this work, we developed BNet, a multi-platform bot architecture to broaden the donor base. We also designed a dual-layered screening system for the cost-effective management of blood donation requests across social media. We conducted a study with 114 users, demonstrating that auto-screening and geo-location-based notifications significantly accelerate donor response time. We also noted that user satisfaction was greatly increased by the intuitive interface and ease of slash command usage. Based on these insights, we propose future research directions for cross-platform bot design and its application in information retrieval.

CCS Concepts: • **Information Retrieval (IR)-> Human Computer Interaction (HCI).**;

Additional Key Words and Phrases: social networking sites, blood donation systems, multi-platform bot architecture, dual-layered screening

ACM Reference Format:
. 2024. BNet: A Multi-Platform Bot Framework with Cost-Optimized Dual-Layered Screening for Enhanced Blood Donation via Social Media Integration. 1, 1 (October 2024), 23 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the digital era, the growth of online social groups and communities has accelerated due to the expansion of community-oriented and social networking sites (SNSs) [8]. Users actively share, generate, and circulate urgent blood donation requests within groups during critical moments [15]. The scattered nature of data on social media groups makes extracting significant information a formidable challenge [12]. The dynamic and overlapping nature of online social communities adds complexity to the problem [20]. Users often belong to multiple communities with various priorities [20]. It creates a tangled web of data that is difficult to navigate [20]. The lack of effective filtering and ranking mechanisms leads to the mixing of relevant and irrelevant messages [6]. So, It becomes challenging for users to locate crucial blood donation requests on time. The unrestricted and heterogeneous nature of information generation on SNSs has outpaced the development of intelligent systems that can effectively organize and notify potential donors about relevant requests [12].

AI-enabled agents or bots have rapidly gained popularity in healthcare, e-commerce, and banking [18]. They operate as notification systems and provide timely updates and alerts [36]. Conversational bots engage users to enhance information accessibility [11]. Additionally, bots can effectively replace information desks in banks and assist potential clients with inquiries regarding loan options, account opening procedures, and other general information [36]. In such circumstances,

Author's Contact Information:

integrating bots with social media groups is highly necessary for efficiently extracting blood donation requests and notifying potential donors.

Prior research shows that existing Blood Donation Systems (BDSs) primarily rely on app-based solutions [28, 33, 42]. In the saturated app marketplace, users face an overwhelming array of options—over 2 billion apps available on Android alone, with Apple offering a comparable selection [9]. Despite this vast ecosystem, user engagement remains minimal. Nielsen reports that the average number of apps actively used per person has stabilized at just 27 over recent years [9]. App-based platforms present several challenges like the need for installation, regular updates, and significant smartphone storage usage [9]. These standalone systems also struggle with user retention and reduce their effectiveness for time-sensitive tasks like blood donation requests. Millions of people across diverse groups spend time, share news, and seek help on Online Social Networks (OSNs) like Facebook, Telegram, Whatsapp, and Discord. People prefer social media to initiate donation requests due to the lack of a firm BDS compared to using apps [3]. As shown in Table 1, OSNs offer unparalleled opportunities for quickly reaching large audiences. Integrating a bot with these platforms ensures timely outreach and simplifies tasks by avoiding the need for a separate app for a single task. To the best of our knowledge, there has been no exploration of incorporating bots into social media groups to facilitate blood donation initiatives. To address this gap, we ask the following research questions in this work:

- **RQ1:** How can a multi-platform bot be designed to seamlessly integrate with OSNs to accelerate donor response and broaden the donor network?
- **RQ2:** How can a cost-efficient framework be developed to precisely filter blood donation messages from extensive message streams to minimize operational costs?
- **RQ3:** How can a bot serve diverse demographic groups for blood donation and ensure that users perceive its integration as convenient across social media groups?

To the best of our knowledge, we are the first to introduce a system that automatically collects and parses messages seeking blood donations while identifying matching donors from our database. Additionally, we provide a user-friendly interface for new donors to register easily. Our system features a centralized control mechanism that extracts these messages from localized networks and disseminates them to suitable donors based on geolocation and availability. We also report a quantitative analysis of activity indicators before and after the adoption of BNet. Furthermore, We provide a qualitative assessment of how integrating the bot into SNSs reduces donor response time and expands the donor base. Our contributions in this paper include:

(1) We introduce an innovative multi-platform bot architecture that leverages a centralized database to enhance donor outreach and engagement and broaden the donor base.
(2) We develop a cost-optimized dual-layered screening architecture that employs advanced algorithms to efficiently filter blood donation requests from extensive message pools. This sequential approach first utilizes fastText framework, followed by GPT-4o-mini model to improve performance and parsing. We also analyze how dual-layer filtering reduces bot costs for users by enabling efficient integration on free OSNs.
(3) We provide empirical evidence demonstrating how the multi-platform bot approach can significantly reduce donor response time and ensure the timely visibility of critical messages for potential donors. We also identify the factors that impact user satisfaction levels.

Table 1. User engagement and group affiliation on SNSs [10, 13, 14, 34, 35]

Platform	Number of Users	Group Size	Per Person Group Affiliation	Time Spent Per Day
WhatsApp	2 billion+	1,024	5-10 groups	17 hr 18 min
Telegram	900 million+	200,000	Multiple channels and groups	3 hr 50 min
Discord	200 million+	250,000 per server	Multiple servers	8 hr
Facebook	3.05 billion+	250	Several group chats	15 hr 27 min

2 RELATED WORK

In this section, we review prior research on existing systems and challenges in BDSs and examine how such systems have addressed information elicitation and efficient user engagement. Additionally, we present the limitations posed by the lack of standardized design practices for multi-platform bots in SNSs.

2.1 Existing systems and challenges in BDSs

Prior works demonstrate notable progress in creating both web-based and mobile-based BDSs. A real-time, intelligent, and rational recommendation system is designed using sentiment analysis of user feedback, donor response rates, and current geo-location information [25]. They introduce a cross-platform application for blood collection and distribution. Cloud computing and mobile computing-based solutions have been developed to create a BDS utilizing cutting-edge information technologies [1]. A web application is further used to manage informative data published through the mobile application and notifications about current blood needs are sent [16]. The software system is comprised of three main components: a central server, a mobile application for donors operating on iOS and android platforms. However, these discrete applications may face challenges like low user engagement and ineffective outreach strategies.

Smart strategies for encouraging blood donation include donor classification using data mining and gamification. Algorithms like CART, Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Logistic Regression can predict donor behavior and enhance outreach efforts [33]. Sofia et al. [27] analyze the features of BDS available on Google Play Store, Apple App Store and Blackberry App World. They evaluate 169 free blood donation applications, primarily focused on the Android platform. Their findings reveal that 60% of the systems fail to notify users about blood requests. Main problems in existing BDS include limited iOS and Windows availability, installation failures, lack of authentication, geographical restrictions, missing notifications, minimal social media sharing and inadequate donar engagement features [27].

2.2 Lack of design standards for multi-platform bots in SNSs

Bots are software applications designed to perform automated, unsupervised and repetitive tasks efficiently. Telegram bots deliver regular notifications or updates such as vaccine availability, stock information, scheduled intervals and so on [27]. They can also integrate with external services like Gmail, IMDB, GitHub, and Spotify with the Wiki bot and enable direct Wikipedia searches within Telegram [27]. Additionally, they allow businesses and individuals to accept payments and streamline transactions [27]. Organizations can develop custom tools for client interaction and query resolution [27]. Telegram bots also deliver location-based services that offer personalized content and interactions based on user [27]. Discord bots help manage servers, control spam, play music and provide information about users [2]. Messenger bots serve multiple purposes such as giving weather updates, providing entertainment, appointment scheduling and order tracking [37]. However, none of these bots are designed to extract urgent blood-related messages from large message pools and identify potential donors by location on SNSs. Additionally, deploying a bot typically depends on specific platform architecture

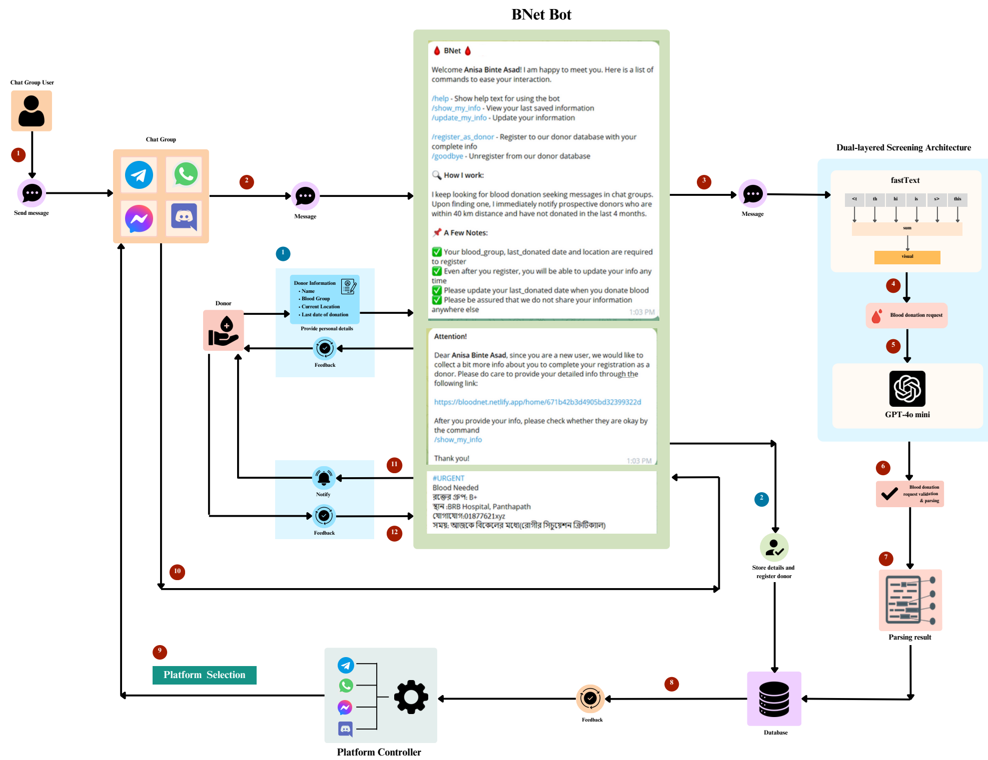


Fig. 1. This figure presents the BNet architecture. Firstly, users register from different platforms. Secondly, the bot continuously filters blood donation requests through a dual-layered screening architecture consisting of fastText and GPT-4-mini. After parsing messages, a notification is sent to the nearest donor in the database. Finally, donors can provide feedback through the bot.

[26]. It drives up both development and maintenance costs [26]. Blood donation requests require timely access to as many potential donors as possible. Therefore, we analyze the performance of a multi-platform bot for collecting data across various domains. We also explore the effects of multi-platform interface design.

3 BNet

Messages are frequently circulated on SNSs, sharing requests for blood donations through texts, posts and stories. However, most of these messages lack a consistent structure that complicates the automation of donor matching based on multifaceted criteria. Rapid dissemination is critical for timely responses. However, these messages typically reach only a limited audience within specific communities due to social media clustering. Donors often overlook them amid a vast pool of content. To address this, we firstly propose a cost-optimized dual-layered screening architecture for detecting blood-related requests in groups. Another key challenge is efficiently notifying potential donors based on their location and availability. Our system employs a centralized multi-platform control system to identify compatible donors from our database based on geo-location. In this section, we provide a comprehensive overview of the key features and functionalities of BNet. Figure 1 shows the system architecture of BNet.

3.1 Chatgroup end

The bots are initially integrated into chat groups with the explicit consent of both users and administrators. The bot serves two purposes. Firstly, it encourages group members to register as donors by providing a direct link to the registration inbox. Secondly, it polls for messages in the group continuously and looks for ones that seek blood donations.

3.2 Donor registration end

When users engage with our bot via direct messaging to register as donors, they are redirected to a centralized registration web application. This interface systematically collects a data set $D = \{\text{blood\_group}, \text{current\_location}, \text{last\_donation\_date}\}$. We use browser geo-location to get accurate latitude and longitude, with explicit user consent before data collection. Upon submission, the dataset D is stored with the corresponding chat platform ID, allowing efficient user notifications for future blood donation requests. The interface allows users to update their information at any time. This ensures accurate tracking of their last donation date for better record maintenance.

The implementation of the donor registration as a single point of access was a matter of design choice that was inspired by the fact that it would otherwise be cumbersome to craft such input mechanisms in each of the bot's chatting interfaces separately.

3.3 Proposed cost-optimized dual-layered screening architecture

In the first layer, we employ the fastText algorithm [40], developed by Facebook AI Research, to implement a Blood Donation Needed Message Classifier. This model is designed to detect blood donation requests in Bengali, English, and Banglish, handling multilingual and mixed-language texts, which is critical for the linguistic diversity of the messages in our dataset. The model architecture begins with an input layer where raw messages are processed. We then employ subword tokenization to break words into meaningful subwords to handle spelling variations and language-specific forms. The embedding layer maps these subwords into dense vector representations to capture their semantic meaning. An average pooling layer is applied to compute the average embedding across all subwords in the message. Next, we use a fully connected layer for feature extraction to learn higher-level patterns relevant to classification. Finally, a softmax layer is employed to predict the binary class—whether the message is related to a blood donation request or not. Figure 2 presents the overall architecture of dual-layered screening.

We also experiment with Logistic Regression using TF-IDF vectorization [32]. This approach transforms text messages into numerical features by capturing the frequency of unigrams and bigrams. We use L2 regularization to prevent overfitting [22]. We compare this with the fastText model. fastText proves to be the better approach due to its superior handling of multilingual and mixed-language texts. In the second layer, we utilize GPT-4o-mini to further filter out non-blood donation-related messages, ensuring only relevant content is allowed [5].

3.4 Optimized message parsing with few-shot learning

After detecting a message requesting blood donation, we first parse it into a predefined structure to extract key information. Formally, the message is parsed into a dataset $M_p = \{\text{blood\_group}, \text{bags\_needed}, \text{location}, \text{contact\_number}\}$, ensuring that all critical elements are captured for efficient processing. For the task of parsing, we employ the model named GPT-4o-min chosen for its balance between cost and accuracy.

To further enhance parsing precision, we apply the technique of Few-Shot Prompting [30]. In this approach, the model is exposed to a small number of examples, specifically two positive examples and one negative example to guide its predictions.

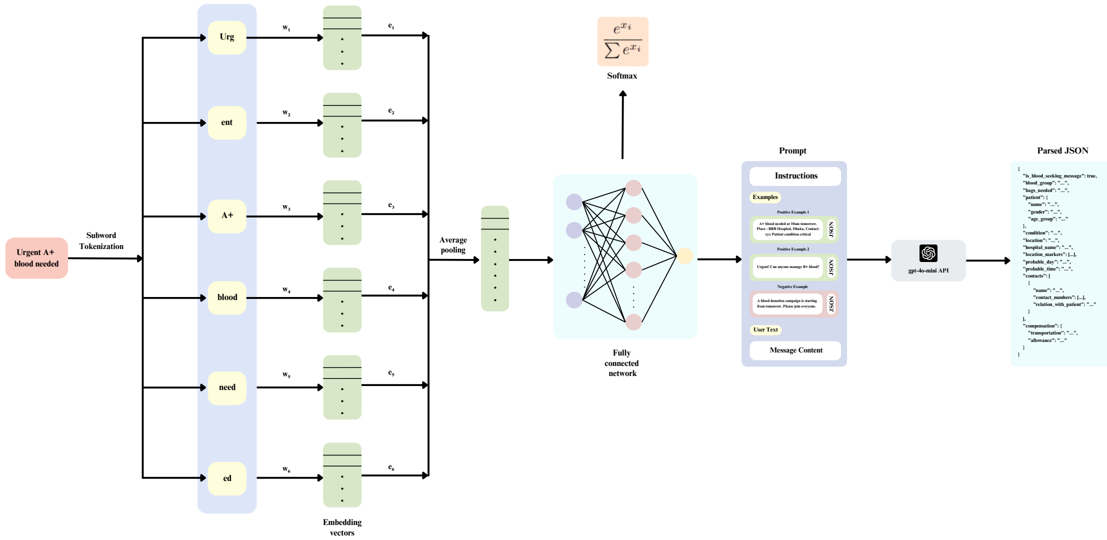


Fig. 2. This figure shows the dual-layered screening architecture of BNet. Firstly, raw messages are processed in the input layer. Then, subword tokenization and average pooling are applied to compute the average embedding across all subwords in the message. Next, a fully connected layer performs feature extraction and a softmax layer predicts the binary class. GPT-4o-mini is then used to further filter out and parse messages.

For a **positive example** message M_p, which is relevant to blood donation, the model is expected to output the parsed information in a structured JSON format. The output can be expressed as:

$$P(M_p) = \text{JSON}\{\text{blood\_group}, \text{bags\_needed}, \text{location}, \text{contact\_number}\}$$

where $P(M_p)$ is the parsed JSON output containing the necessary details for blood donation. For a **negative example** message M_n, which is unrelated to blood donation, the model is expected to flag it as irrelevant. The model's output is: $P(M_n) = \text{FLAG}_{\text{negative}}$ where $P(M_n)$ indicates that the message does not pertain to blood donation.

When a new message M_{new} arrives, the model uses Few-Shot Prompting to determine whether the message is relevant or irrelevant. The decision process can be represented as:

$$P(M_{\text{new}}) = \begin{cases} \text{JSON}\{\text{blood\_group}, \text{bags\_needed}, \text{location}, \text{contact\_number}\}, & \text{if relevant} \\ \text{FLAG}_{\text{negative}}, & \text{if irrelevant} \end{cases}$$

This ensures that relevant messages are correctly parsed. For robustness, the parsed results—whether structured JSON or negative flags—are stored in our database. This stored information is then used to retrain the initial screening model, represented by the training dataset D_t. Each new parsed result $P(M_{\text{new}})$ is added to the training data: $D_t = D_t \cup \{P(M_{\text{new}})\}$. This continuous feedback loop allows the model to incrementally improve its accuracy.

The linguistic parsing, however, does not produce all the necessary information needed for the further analysis. For instance, to find nearby donors, we also need to know the approximate latitude and longitude of the location of expected donation. Therefore, after parsing, we utilize Geocoding API to extract the latitude and longitude of the location string provided by the parser model. Finally, we log this structured information in our database and call for an immediate search for most prospective donors.

3.5 Systematic identification of potential donors

Potential donors are filtered from our database based on blood group, location, last donation date and urgency. We start by calculating the Haversine distance to filter donors within a certain threshold who match the requested blood group. The Haversine formula is given by:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\cdot\cos(\varphi_2)\cdot\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

where d is the distance between two points on the Earth's surface, r is the Earth's radius, φ_1 and φ_2 are the latitudes of the two points in radians, $\Delta\varphi$ is the difference in latitude, and $\Delta\lambda$ is the difference in longitude.

Once we filter out donors within the specified Haversine distance whose blood groups match the requested blood group, we then exclude those whose last donation dates exceed 120 days. Finally, a list of these prospective donors, along with their respective distances, is prepared.

3.6 Optimized donor notification workflow

Efficiently notifying the donors poses some additional challenges. Firstly, notifying too many or too few donors may negatively affect the system in distinct ways. Secondly, the users often themselves edit their messages stating that blood has already been managed. In those cases, previously notified donors must also be kept updated.

To grapple with the first challenge, we decide to notify the donors iteratively in subsequent stages. While notifying we ask the donor to confirm whether they will be able to donate or not. After a confirmation is received, we skip the rest of the stages and inform the blood seeker of the interested donor. The number of notifications for the stages in turn depends upon the urgency of the blood seeking message as estimated by our parsing model. And to tackle the second problem, we keep track of each of the messages and the corresponding notifications that we have sent. Whenever such a message is edited stating that blood has been managed, we again inform those donors of the update.

3.7 User Interface

The BNet interface majorly comprises two components: the chatbot interface and a single point of donor information intake. Since we plan to employ our bots as members of already running chat groups, the chat interface is essentially the same as the interfaces of those corresponding chat platforms. For our initial design, we selected two prominent chat platforms named Telegram and Discord. Both of these platforms feature engaging conversational interfaces which we utilize for our purpose. Unlike several other platforms, Telegram and Discord share a particular feature, namely, the use of slash user commands in the chat interface.

Since the interactions with our chatbot are limited in possible options, we opt to design convenient user commands rather than parsing natural language messages from users on the fly. The currently available user commands in our chat interface are as follows:

- `/start`
- `/help`
- `/show_my_info`

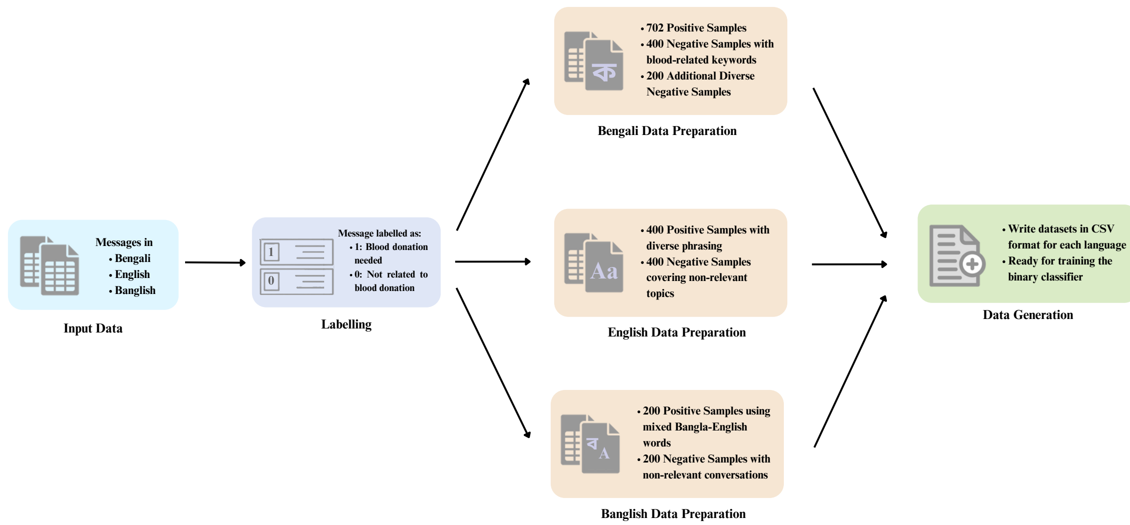


Fig. 3. This figure shows the workflow of the dataset preparation procedure. We collect three types of multilingual messages, both blood donation-related and non-related. We then label and balance them accordingly.

- `/update_my_info`
- `/register_as_donor`
- `/goodbye`

To facilitate the input of donor information, we designed a single-page web application featuring a form that receives the blood group, last donation date, and GPS location from the browser. To eliminate the need for reiterating a donor's chat platform identity, we generate a unique URL for the donor based on their user account on the chat platform. By visiting this unique URL, the donor can update their information directly from the chat interface at any time.

4 EXPERIMENTAL EVALUATION

To conduct experiments on our screening and parsing modules, we curate a diverse dataset of positive and negative samples, where positive samples refer to blood donation-seeking messages and negative samples refer to others. In this section, we demonstrate the process of curating the dataset. Also, we present the implementation details to ensure reproducibility.

4.1 Dataset preparation

The dataset comprises messages in Bengali, English, and Banglish (a mix of Bengali and English). Handling these languages effectively is critical since emergency blood donation requests in Bangladesh are often communicated using all three. It is constructed to address imbalances and linguistic diversity. The data collection includes both positive samples (blood donation needed) and negative samples (non-relevant messages), generated and labeled carefully. Each message in the dataset is labeled as:

- 1: Blood donation needed.
- 0: Not related to blood donation.

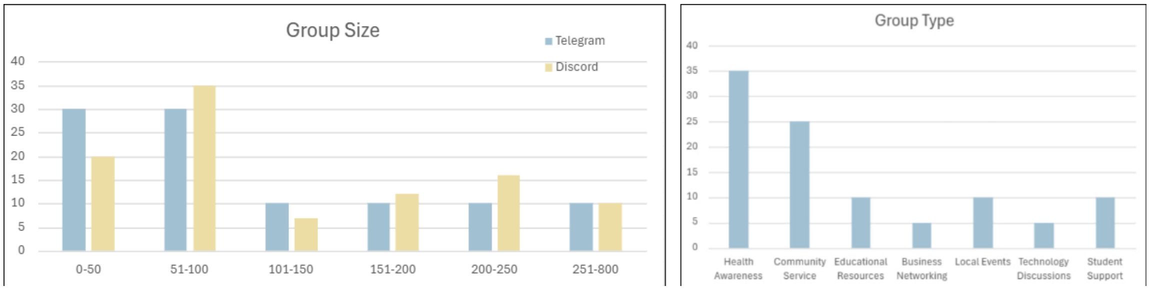


Fig. 4. This figure shows demographic distribution by group size and type

Figure 3 shows the workflow of dataset preparation. Below is a detailed breakdown of the data preparation process.

Bengali data preparation.

Positive samples. We parse 702 positive samples, most of which are written in Bengali. These samples contain blood group information, patient status, location, and contact numbers.

Negative samples. 400 Bengali negative samples are generated with frequent blood-related keywords embedded, such as ”রক্ত,” ”হাসপাতাল,” and ”ব্যাগ,” to ensure the model is exposed to potential confusions. To avoid limiting the vocabulary, 200 additional diverse Bengali negative samples are created to cover a wide range of topics beyond emergency contexts.

English data preparation. To ensure the model can handle English messages effectively, we generate equal numbers of positive and negative samples.

Positive samples. 400 examples include requests for specific blood groups, with details such as hospital names, patient conditions, and contact numbers. These samples vary in structure and phrasing to avoid overfitting to patterns.

Negative samples (400). 400 samples cover a broad range of non-relevant topics, such as travel plans, sports, events, and daily activities, to simulate real-world message variance.

Banglish data preparation. Messages written in Banglish (Bengali transliterated into English) are common in communication channels in Bangladesh. We generate 200 positive and 200 negative Banglish samples to ensure the model can handle this mixed-language input effectively.

Positive samples. 200 samples follow typical blood donation requests but use a mix of English and transliterated Bengali words, such as ”dorkar,” ”contact,” and ”lagbe.”

Negative samples. 200 negative samples cover diverse, non-relevant conversations, such as casual chats, daily activities, and announcements, using Banglish.

4.2 Implementation Details

The chatbots are implemented with standard libraries released and maintained by the corresponding chat platform. For instance, to design the chatbot for Telegram, we use the library python-telegram-bot in python and discord.js library for the Discord bot. These libraries help us take appropriate actions based on slash commands and user messages. In case of slash commands, we perform string matching and execute corresponding methods. On the other hand, for any non-command text,

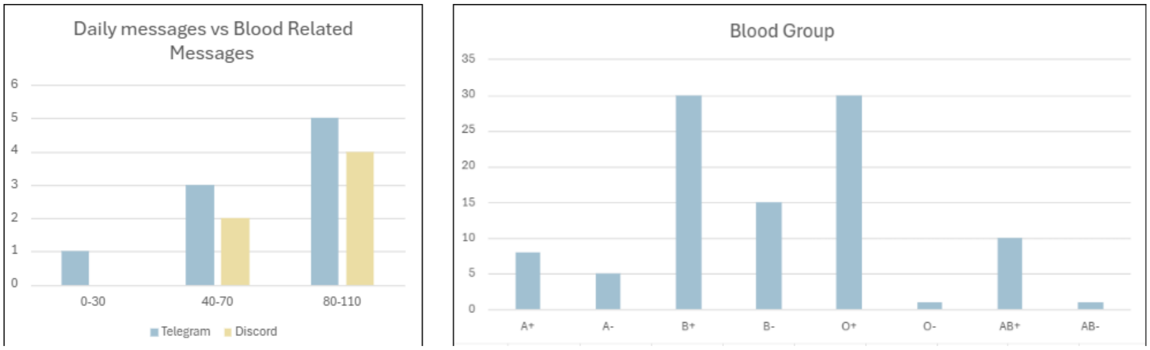


Fig. 5. This figure shows blood donation related messages numbers in different groups and blood group of users

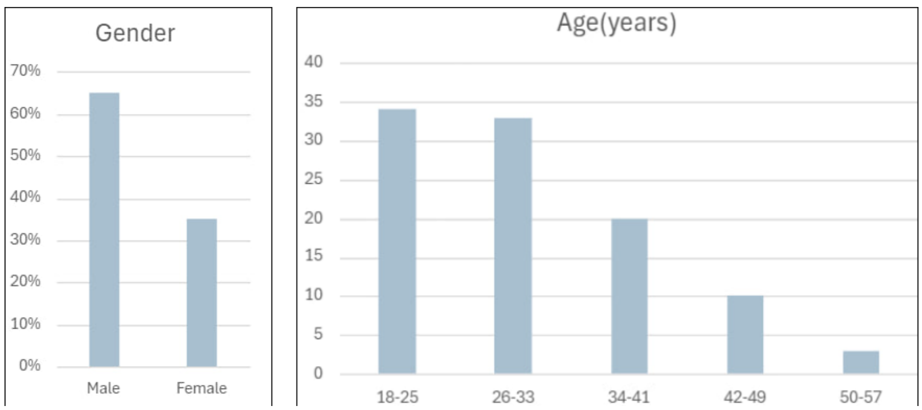


Fig. 6. This figure shows demographic distribution by gender and age

we first call the screening API with the text to determine whether it is actually seeking blood donation or not. If yes, we further call our parsing API to parse the text into a JSON format. The screening API is implemented with a library named fastText from Meta Research. We first perform training on a curated dataset and then carry out inferences. We train the model for 1000 epochs using a learning rate 1.0. We use trigrams (wordNgrams=3) to capture better context from word sequences. Subword length is configured with minn=3 and maxn=6 to handle out-of-vocabulary words. The parsing API is implemented with Langchain. As LLM, we use Gpt-4o-mini with few shot prompting. The unified donor information intake application is built with React and these pieces of information are stored in MongoDB under appropriate models

5 USER STUDY

To assess the effectiveness of multi-platform bots for timely message screening and notifications, we conducted a study on various indicators like response time, user interface, command usability and satisfaction levels. Participants shared their experiences regarding delays in receiving blood donation requests. We organized questions into two parts: one for those who frequently share these messages and another for blood donors. Different demographics were included to ensure a balanced study.

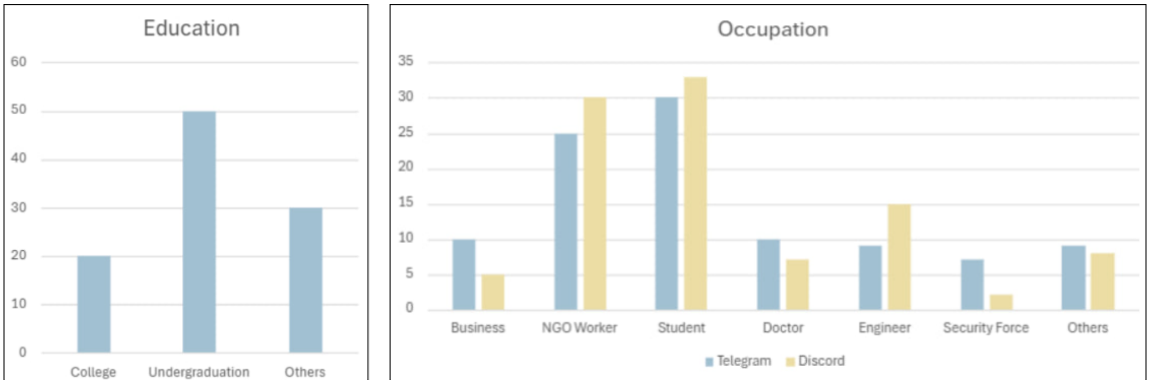


Fig. 7. This figure shows demographic distribution by education and occupation

5.1 Conditions

We conducted a between-subject study with two conditions for a predefined period — the baseline social media group without a bot and one integrated with BNet. Each condition featured a consistent set of questions designed to gather insights on user experience, response time, engagement levels, and challenges. The baseline system relied on manual messaging and user coordination; in contrast, the BNet integration introduced blood donation message screening, automated responses, real-time donor matching, and geo-location-based notifications.

5.2 Participants

We recruited members from 20 active groups on Telegram and 10 active groups on Discord, all based in Bangladesh. Figure 4 and Figure 5 shows the demographic distribution across the groups. The groups varied in size, ranging from 50 to 8,00 members, and were created for different purposes, including health awareness (35%), community service (25%), educational resources (10%), business networking (5%), local events (10%), technology discussions (5%) and student support (10%). Furthermore, the volume of messages within these groups ranged from 30 to 110 with 1-20 as blood donation requests per day. A total of 114 participants, with 38 potential donors, joined the survey on pre- and post-integration of BNet. Figure 6 and Figure 7 shows the demographic breakdown of a diverse group of participants, aged 18 to 57. Most were between 18-25 (34%) and 26-33 (33%). A majority (80%, 91/114) had at least a college or bachelor's degree. The participants came from various professions: 10% in business, 25% as NGO workers, 10% as doctors, 9% as engineers, 7% in security forces, and 30% were students.

5.3 Procedures

At the start of the study, we selected diverse groups based on different targets and ages. Admins signed consent form from the group. We recorded the average daily messages and blood-related messages per group. Each participant signed a consent form and completed a pre-study questionnaire to gather demographic information including gender, age, occupation, education level, prior experience with social media groups, and other BDSs regarding blood donation initiatives. They also shared their initial expectations from BNet.

The study was conducted over three days, from October 23 to 26, 2024. We integrated bots into the groups. A total of 108 individuals registered as donors from different locations. The last donation dates and blood groups are stored during registration on bots. Among the donors, 30% are O+ and 30% are B+ blood types, while 1% are O- and AB-, indicating a lower proportion of negative blood types as shown in Figure 5.

They did not receive training on the bot to assess the user interface's intuitiveness. We then began collecting feedback from them. Participants were asked about their satisfaction levels in areas such as prior problems, satisfaction with slash-command prompts, user interface, overall functionality, comparisons with other apps, challenges in using the bots, and suggested improvements. Responses were gathered using a five-point Likert scale to evaluate their experiences. Our contributor survey had two types: users who made requests and donors who were notified through bots and donated in the last three days. Users answered 11 questions, while donors answered 7 questions given in Appendix A.

6 DATA ANALYSIS

To assess our research questions, we formulated three key hypotheses:

H1 - Auto screening of blood donation messages and geo-location-based notifications of BNet will accelerate the speed of donor response.

H2 - Dual-layered screening architecture of BNet will cost-effectively filter and parse blood donation messages from extensive social media streams.

H3 - BNet, as a multi-platform bot, will serve diverse demographic groups equally and improve convenience across OSNs We first analyzed group messages, blood donation requests per day, and group demographics. For RQ1, we proposed hypothesis H1. To validate H1, we tracked donor response times using timestamps at each stage: message dispatch, bot execution, notification delivery, and response received. For RQ2, we introduced H2, evaluated dual-layered screening accuracy with precision, recall, and F1-score and assessed the cost-efficiency of this approach. To answer RQ3, we proposed H3, using both quantitative and qualitative analyses of usage logs, pre- and post-study surveys and feedback to assess response quality and satisfaction. Post-study Likert-scale feedback on slash-command prompts, UI design, and user satisfaction, alongside insights from open-ended responses helped highlight improvements and challenges. We demonstrated network growth through multi-platform integration. We also applied Pearson's correlation and Spearman's rank correlation and to examine associations between indicators of satisfaction index. A summary of metrics and measures is provided in Table 2. Additionally, we reviewed user demographics, compared current blood donation apps, and identified areas for improvement. To minimize message overflow and enhance satisfaction, we explored optimal request frequency, and emphasized security for future implementation.

6.1 Metrics and Measurements

To evaluate all hypotheses, we define metrics mentioned in Table 2.

Timely Response: We define Timely Response as the measurement of the time taken from when a message arrives to when it is parsed, a donor is notified and a response is received. We assess it in two ways. First, we use timestamps to track the time from the arrival of the message to the response in each stage. The second method involves measuring satisfaction with the timely response of BNet in identifying potential donors between October 23 and October 26, 2024, after its integration into groups. Participants respond on a scale from 1 to 5: 5 indicates ”Very satisfied,” 4 means ”Satisfied,” 3 means ”Neither,” 2 means ”Dissatisfied,” and 1 represents ”Very dissatisfied.”

Screening Accuracy: We define Screening Accuracy as the ability to identify and extract blood donation requests from a vast array of messages. This assessment incorporates key performance indicators: precision, F1-score, and recall to ensure robust evaluation [39].

Table 2. This table shows the metrics for the evaluation of BNet performance.

Hypotheses	Metric	Explanation	Measurement Method
H1	Timely Response	Measurement of the elapsed time between message arrival, request parsing, donor notification and donor response	Timestamping
H2	Screening Accuracy	Identifying and parsing blood donation requests from a large pool of messages	Precision F1-score Recall
	Cost Efficiency	The financial implications of filtering and parsing blood donation requests from a large pool of messages	Pricing Model
H3	Command Usability Intuitiveness Satisfaction Index	Perception of ease of use of the slash command prompt Perception of user-friendly interface Overall user satisfaction in functionality and performance of the bot	Likert scale response

Cost Efficiency: We measure Cost Efficiency as the financial impact of filtering and processing blood donation requests from a large volume of messages. This involves a pricing model that evaluates various message volumes across different groups and analyzes expenses based on existing cost structures (see Open AI cost link). We then compare these costs to the expenses associated solely with parsing blood-related messages.

Command Usability: We define Command Usability as the ease with which users can utilize slash commands. To assess this, we posed the question: ”How easy do you find using BNet through slash-command prompts (e.g., /start, /show_my_info, etc.)?” Responses are rated on a scale from 1 to 5, where 5 signifies ”Extremely easy,” 4 indicates ”Very easy,” 3 denotes ”Moderately easy,” 2 represents ”Slightly easy,” and 1 means ”Not at all easy.”

Intuitiveness: We measure Intuitiveness as the perception of the user interface in relation to usability, color scheme, layout, and overall aesthetic appeal. To evaluate this, we ask, ”How intuitive do you find the user interface of BNet?” Respondents rate their experience on a scale from 1 to 5, where 5 represents ”Extremely intuitive,” 4 indicates ”Very intuitive,” 3 signifies ”Moderately intuitive,” 2 denotes ”Slightly intuitive,” and 1 means ”Not at all intuitive.”

Satisfaction Index: We ask both donors and requesters to evaluate the overall functionality of BNet with the question, ”How would you rate the overall functionality of BNet?” Responses are rated on a scale from 1 to 5, where 5 represents ”Excellent,” 4 indicates ”Above Average,” 3 signifies ”Average,” 2 reflects ”Below Average,” and 1 denotes ”Very Poor.”

Additionally, we gather user feedback regarding challenges and potential improvements. To assess performance, we inquire, ”Do you find BNet more effective than the blood donation apps or methods you have previously used?” This comparison provides valuable insights into BNet’s effectiveness in enhancing the blood donation experience.

Table 3. This table shows the result of time tracking in each stage of BNet from arrival to response

Task	Average Time	Standard Deviation
Parsing Time	4s	0.45s
Retrieval Time	5s	3s
Response Time	81min	110min

Table 4. This table presents classification report of fastText framework

Class	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	249
1	0.99	0.98	0.99	276
Macro Avg	0.99	0.99	0.99	525
Weighted Avg	0.99	0.99	0.99	525

7 FINDINGS

In this section, we present our key findings regarding response time, operational cost efficiency, and user convenience in detail. Our analyses reflect the significant reduction in the parsing and retrieval time after deploying BNet. The dual-layered screening architecture helps BNet maintain adequate accuracy while reducing the parsing cost. Our survey results also indicate that BNet can be helpful for both donors and recipients across diverse demographic groups.

7.1 H1 Results: Auto-screening of blood donation messages and geo-location notifications speeds up donor response by reducing parsing and retrieval time

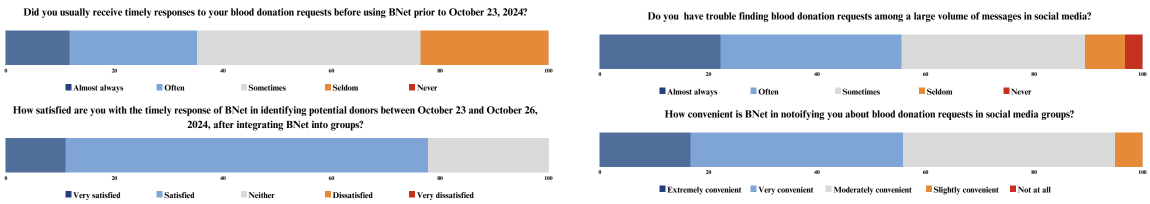
We track four specific timestamps from message arrival to donor response. First, we log the time the message arrives in the group. Next, we record when the parsed blood donation request is stored in our database; the difference between these two timestamps indicates the time taken to parse the message. Third, we log when a notification is sent to the first matching donor; the time between the second and third timestamps represents the retrieval and matching process. Finally, we capture the first affirmative response from a donor, with the time between the third and fourth timestamps indicating donor response time. As shown in Table 3, we observe notable differences at each stage, from message arrival to donor response. Parsing time averaged 4 seconds with a low variability of 0.45 seconds. Retrieval time averaged 5 seconds with a standard deviation of 3 seconds. The most significant finding was Response time, averaging 81 minutes with a high variability of 110 minutes.

7.2 H2 Results: The dual-layered screening architecture of BNet efficiently filters and parses blood donation messages from large social media streams, delivering high accuracy at a lower cost

The fastText model was evaluated on the test set, achieving an overall accuracy of 98.7%. In the classification of messages, non-blood-related messages are denoted as 0 and blood-related messages as 1. As shown in Table 4, for class 0, the model attained a precision of 99%, with a recall at 99%, resulting in an F1-score of 0.99. For class 1, precision remained at 99%, while recall was 98%, yielding an F1-score of 0.99 as well. Overall, the macro and weighted averages for precision, recall, and F1-score are all 0.99.

Table 5. This table shows statistical result of Spearman Correlation and p-values

Metric	**Spearman Correlation**	**p-value**
Timely Response	0.06	0.80
Command Usability	0.52	0.02
Intuitiveness	0.54	0.01
Timely Notification	0.43	0.08



(a) Timely Response

(b) Satisfaction Index (donor)

Fig. 8. This figure shows: a) users' experience receiving timely responses to blood donation messages before and using BNet, and b) the overall satisfaction level of blood donor with BNet.

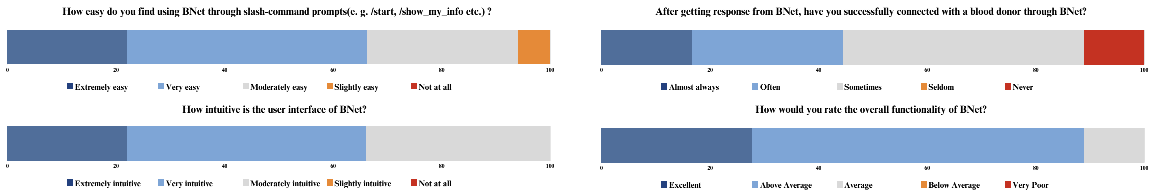
We, furthermore, analyzed the cost efficiency of single-layered screening (using only GPT-4-0-mini) compared to dual-layered screening (using the BNet architecture) as shown in Table 6. We first recorded the daily message volume from our observed groups per day. Next, we logged the number of blood donation requests identified by BNet. Using GPT-4-0-mini at a rate of \$0.0003 per message, direct processing costs would be \$0.0045, \$0.0165, and \$0.0285 per day for average message counts of 15, 55, and 95, respectively. In the initial layer, BNet filters messages with 98.7% accuracy, isolating blood donation requests with average counts of 1, 3, and 5 per day. These filtered messages then proceed to the second layer, where GPT-4o-mini performs validation and parsing at a cost of \$0.003, \$0.009, and \$0.015, respectively. Overall, this dual-layered architecture reduces costs by approximately 33.33% to 47.37%, depending on message volume.

Table 6. This table shows cost analysis of dual-layered screening architecture

Range	Avg. Messages	Blood Related Messages	Avg. Cost	Avg. Cost of Blood Related Messages
0-30	15	1	\$0.0045	\$0.0003
40-70	55	3	\$0.0165	\$0.0009
80-110	95	5	\$0.0285	\$0.0015

7.3 H3 Results: BNet will serve diverse demographic groups equally and improve convenience across OSNs through timely notifications, timely responses, command usability, intuitiveness

Our survey shows diverse demographics concerning gender, age, education, and occupation as shown in Figure 6 and Figure 7. Among the participants, 65% identified as male and 35% as female. All age groups were represented, with individuals aged 18-33 showing the highest interest in blood donation. Males are more inclined to donate blood than females. Among different professionals, students constituted 30% of respondents while NGO workers made up 25% ranking second. Notably, 80% of the participants had college or undergraduate education. Among these groups, 11% "almost always" make donation requests, 39% "seldom" request and 22% "sometimes" request. However, only 11% reported receiving timely responses "always". After integrating BNet, 67% of users were "satisfied" with the timely responses from BNet. We inquired about the slash-command prompts, user interface, and overall functionality of BNet. Notably, 44% of respondents found command usability to be "very easy" while another 44% described the user interface as "very intuitive". Additionally, 61% rated the overall functionality as "above average" with 28% considering it "excellent". In social media, 21% of donors report "always" having trouble finding blood donation requests amid a high volume of messages, while 32% experience this "often," and



(a) Command Usability and Intuitiveness

(b) Satisfaction Index (Requester)

Fig. 9. This figure shows: a) perception levels of command usability and user interface, and b) the overall satisfaction level of blood requester with BNet.

another 32% encounter it ”sometimes.” However, after receiving notifications through BNet, 39% of donors find the process ”very convenient,” and an additional 39% rate it as ”moderately convenient.” Figure 8 and Figure 9 shows full result of different metrics from survey.

We first examined the correlations among four metrics—Timely Response, Command Usability, Intuitiveness, and Satisfaction Index—using Pearson’s correlation analysis to explore their interrelationships as shown in 10. Notably, Command Usability and Intuitiveness show a high correlation coefficient of 0.60. Additionally, there is a moderate positive correlation of 0.54 between Intuitiveness and the Satisfaction Index. Command Usability and the Satisfaction Index exhibit a positive correlation of 0.50. However, Timely Response did not significantly correlate with the other metrics, particularly with the Satisfaction Index (-0.01) and Command Usability (-0.05).

We also conducted a Spearman’s rank correlation to evaluate the monotonic relationship between overall satisfaction and the functionality metrics of BNet as shown in Table 5. We observed that Command Usability exhibited a strong positive association of 0.52, with a p-value of 0.02. This suggested that enhancements in command usability likely resulted in increased user satisfaction. We also noted that Intuitiveness displayed an even stronger correlation of 0.54, with a p-value of 0.01. This reinforced the concept that a more intuitive interface significantly contributed to user satisfaction. This finding was statistically significant, emphasizing the importance of interface design. Furthermore, Timely Notification revealed a correlation of 0.43, with a p-value of 0.08. This indicated a moderate connection with user satisfaction, although it did not reach conventional significance thresholds. Conversely, Timely Response showed a correlation of 0.06, with a p-value of 0.80. This demonstrated a lack of a statistically significant association with user satisfaction.

However, only 17% of users reported being ”almost always” receiving blood, while 44% experienced ”sometimes”, and 11% never connected even after receiving responses from donors through BNet. When asked about the challenges they faced while connecting, one donor P33 highlighted, “I encountered communication and transport issues even if I responded to donate”. P54 expressed, “My family did not allow to donate blood to individuals I did not know”. Additionally, P60 remarked, “The location of donation requests was unclear”. We inquired about existing blood donation apps or methods that participants had used previously. Notably, 50% stated that BNet is ”much better”, while 39% described it as ”somewhat better”. When asked for suggestions for improvement, P51 remarked, “It would be beneficial to incorporate BNet into other social media platforms for wider accessibility”. Another participant, P42, suggested, “A dedicated dashboard displaying donation requests would be helpful for users”.

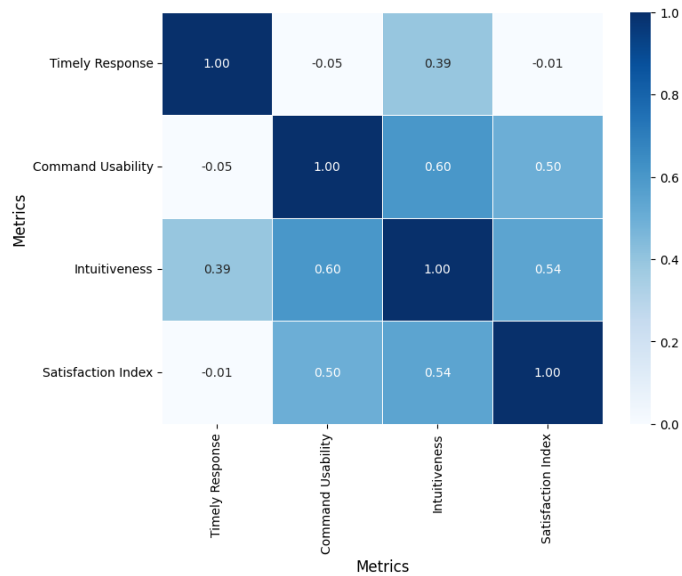


Fig. 10. This figure shows Pearson Correlation Heatmap of User Feedback Metrics

8 DISCUSSION

In this study, we designed and developed a multi-platform bot to engage users and efficiently screen large volumes of messages. To keep screening cost-effective, we implemented a dual-layered screening architecture. Our evaluation with 114 users showed improved response times and engagement and provided more effective support to existing BDSs. Automated screening and notifications enabled faster responses, while multi-platform integration created a versatile donor network. Convenient slash commands and an intuitive interface made it easy for participants to use. This section discusses the implications of our findings and provides design recommendations.

8.1 Creating a fast response system architecture for a multi-platform bot

There exists a delicate balance between success and failure in urgency management [19]. Timely and precise execution during emergencies significantly enhances the likelihood of successful outcomes and mitigates potential risks [19]. Auto-notification system is necessary for system administrators in this regard [7]. It provides real-time updates on system status and enables prompt responses to issues by maintaining optimal operational efficiency [7].

Our findings demonstrate that auto-screening of blood donation messages and geo-location notifications significantly accelerates donor responses. The average parsing time is just 4 seconds with minimal variability. This rapid message screening enables us to sift through large pools of social media messages and quickly identify relevant donation requests. The retrieval time averages 5 seconds, streamlining the matching process. This ensures potential donors are connected to appropriate requests without delay. Auto-notifications use the Haversine distance algorithm to alert nearby donors based on their geographical location. The importance of these features cannot be overstated. This targeted approach minimizes the time and effort needed to locate suitable donations. Response time is a crucial factor we observe. While it averages 81 minutes and shows high variability, it can be significantly influenced by the efficiency of previous steps. Quick parsing and retrieval times help minimize overall delays and facilitates faster connections between donors and recipients.

Before implementing BNet, we asked users if they received timely responses to their blood donation requests. Many reported they did not get timely replies. However, after using BNet, users expressed satisfaction with the promptness of the responses. When we inquired about difficulties in locating blood donation requests among a large pool of messages, results from our study showed that most struggled to find them from a broad range of messages. In contrast, a significant majority found BNet effective in notifying them about donation requests within social media groups. Our analysis also revealed that even though donors responded, many were unable to complete the donation. When we asked open-ended questions on this issue, we noted that most donors cited unclear donation location addresses as a primary challenge. We also observed that some donors expressed concerns about donating to unfamiliar recipients. It underscores a potential need for member authentication. We plan to broaden our research to tackle these concerns. We invite researchers from HCI to collaborate on finding alternative solutions for these challenges.

8.2 Designing cost-optimized dual-layered screening for free multi-platform Use

80% of people utilize social media for interactions with friends, family, spouses, co-workers, old acquaintances, and new friends [38]. Additionally, 76% turn to these platforms to pass the time, often during idle moments or when seeking entertainment [38]. Importantly, social media has assumed an increasingly vital role in emergencies [24], ranking as the fourth most popular source for accessing emergency information [24]. Recent advancements in state-of-the-art Large Language Models (LLMs), such as the GPT series, have showcased exceptional reasoning capabilities across various tasks, including message filtering and parsing [17]. However, the continuous deployment of LLMs on large message pools can lead to significant operational costs. As illustrated in Figure 5, group message volume varies with group size, emphasizing the necessity for primary screening to effectively manage this substantial influx of messages. Ensuring high accuracy during the screening process is paramount. To strike a balance between cost-effectiveness and accuracy, we implemented a dual-layered structure, resulting in high precision, F1 scores, and recall rates. This approach not only enhances the efficiency of message processing but also minimizes resource expenditure. Our dataset was meticulously curated to provide a balanced representation of positive and negative messages across various languages. It addresses both class imbalance and linguistic diversity over a wide array of topics beyond emergency contexts. The versatility of this dataset is crucial for training models that can generalize effectively in real-world scenarios. Notably, our calculations indicate that primary screening can reduce bot operation costs by up to 47%, as only blood donation requests advance to GPT-4-mini for further processing. This significant cost savings underscores the importance of efficient screening mechanisms in optimizing the functionality of LLMs in social media applications.

Integrating a dual-layered screening approach can greatly enhance the management of large volumes of messages on social media platforms, particularly in emergency contexts. This framework not only ensures timely and accurate responses but also demonstrates the potential for significant cost reduction, paving the way for more efficient use of advanced language processing technologies.

8.3 Developing a multi-platform solution for diverse demographics

Association for the Advancement of Blood & Biotherapies (AABB) reports that the average blood donor is typically college-educated and aged 30–50 years [43]. Younger adults, particularly those aged 18-25, are increasingly likely to donate blood [4]. While males have historically been more frequent donors than females [43], this gap is narrowing as more females become regular donors. Additionally, white individuals tend to donate at higher rates compared to Black, Hispanic, and Asian populations [4]. Blood donors from higher socioeconomic backgrounds are more likely to donate, often due to better access to healthcare facilities and donation centers [43]. Table 7 illustrates that the age group most likely to donate blood is also among the most active on social media platforms. This demographic overlap is significant, as males are slightly more active on social media (53.4%) compared to females (46.6%). The similarities in demographics between blood donors and social media users indicate that social media can be a powerful tool for identifying potential donors across various demographic groups. Integrating bots on social media platforms can effectively trace and engage potential donors from all demographic categories.

Our result shows that using a multi-platform approach greatly broadens the donor network by engaging diverse demographics across popular platforms. Each platform offers unique strengths. Telegram is ideal for exchanging messages, sharing media and files, and supporting private or group calls [41]. Facebook focuses on connecting communities [29]. It is effective for creating and maintaining support groups that foster awareness and keep people updated on ongoing donation needs [29]. Discord, initially popular for gaming, allows for real-time text, voice, and video communication in community-centered ”servers” [23]. This feature helps reach younger, tech-savvy users [23]. Our survey highlighted that not all blood types are equally available. Rare types like O- and AB- are often harder to find. Limiting the donor search to a single platform would risk missing donors who frequently use other social spaces. By adopting a multi-platform strategy, we increase the probability of reaching donors with diverse blood types and availability. Our survey results also confirmed that a multi-platform approach increases the donor pool.

When asked for feedback on areas for improvement, participants suggested extending BNet to other social media channels such as WhatsApp and Facebook. This aligns with our future research plans to integrate more platforms and ensure wider coverage.

Table 7. This table shows social media users by different age groups [21]

Age Group	Age Range	Social Media Users (millions)
Gen Z	11-26	56.4
Gen X	43-58	51.8
Baby Boomers	59-77	36.9

8.4 Exploring slash command prompt and user Interface design for multi-platform bot

The user interface of bots are often referred to as the ”universal UI” due to their flexibility and ease of use across multiple platforms [31]. Integrating command prompt mechanisms into these systems has tremendously enhanced their utility [31]. This enhancement facilitates quicker task completion and reduces the need for extensive documentation [31]. In our findings, we explored how these design choices influenced overall user satisfaction within BNet. We asked users about their perception of command usability and the user interface of the bot. Most reported satisfaction with the performance of BNet. Our analysis demonstrated that command usability and interface intuitiveness play a pivotal role in fostering a positive user experience. Pearson’s correlation analysis revealed strong relationships between Command Usability, Intuitiveness, and Satisfaction Index which indicates that an intuitive, accessible interface is crucial for user engagement. Spearman’s rank correlation further confirmed these insights by showing a consistently positive relationship between command usability and user satisfaction. As command usability improved, satisfaction levels also increased and intuitive design elements in the interface had a significant statistical impact. This analysis shows that a quality user interface and easy command access are key for a smooth user experience on BNet.

In our design phase, we selected Telegram and Discord due to their engaging conversational interfaces and shared features of slash commands. We explored these platforms to enhance interaction efficiency. By opting for a structured command interface rather than real-time natural language parsing, we aimed to reduce miscommunication and increase response speed. This design choice made the bot more accessible and user-friendly, ultimately resulting in higher satisfaction levels. To further streamline the experience, we developed a single-page web application that simplifies donor data entry. This application captures essential information such as blood group, last donation date, and GPS location directly from the user's browser. We observed that this addition, combined with unique URLs linked to users' chat platform identities, allowed donors to update their details effortlessly without needing to re-identify themselves. These design decisions facilitated seamless information management and had a positive impact on satisfaction, as users could easily update their information from the chat interface. Each feature we implemented, such as simplified data entry and unique URLs, contributed significantly to user satisfaction by enhancing usability and reducing friction. We explored how a multi-platform command prompt and user interface enhance user interactions by ensuring consistent access and intuitive navigation across various platforms. When we solicited open-ended feedback regarding potential improvements for BNet, users highlighted the need for a dashboard displaying donation requests. This feedback reflects a strong desire for more organized and accessible information. We plan to delve deeper into this feedback to refine and elevate the user experience.

9 LIMITATIONS AND FUTURE WORK

Spamming poses a significant risk in BNet. If users repeatedly send unnecessary blood donation-related messages, it could overwhelm donors' inboxes. This would make it difficult for them to identify genuine requests. We asked, "At most how many blood donation-seeking messages do you feel comfortable receiving from BNet per month?" with options ranging from 1-5 to 21+. Results show that most donors prefer to receive request messages 1-5 times per month. This will help us gauge user satisfaction and do further studies. We will explore the message storage aspect of our system further. Currently, we are testing on a smaller scale, but we need to research how the system will perform when scaled up. With a database of 1,000 matching donors, determining which donors to notify effectively will be crucial. This process resembles load balancing in distributed systems which requires sophisticated algorithms for efficient resource allocation.

Additionally, our bot currently operates on only two platforms. We plan to integrate it with more social media channels in the future to maximize its effectiveness. Expanding the bot's reach will enhance usability and diversify the donor pool and make it more accessible to a broader audience. Regarding limitations, the current system may struggle to handle high volumes of requests or differentiate between urgent and non-urgent messages. Our preliminary tests may not fully capture user behaviors in real-world scenarios. Looking ahead, we will focus on refining algorithms for donor selection and enhancing the bot's capabilities across multiple platforms.

10 CONCLUSION

In this research, we have introduced BNet, a multi-platform bot framework designed to enhance blood donation initiatives through effective integration with OSNs. Our findings reveal that the existing challenges in BDSs, primarily reliant on standalone applications, can be significantly mitigated by leveraging the expansive reach of OSNs. By automating the extraction and filtering of urgent blood donation requests, BNet ensures timely notifications to potential donors, improves response time and broadens the donor network. The dual-layered screening architecture developed in this study not only filters blood donation messages efficiently but also reduces operational costs, making it a viable solution for resource-constrained environments. Our empirical evidence supports the effectiveness of BNet in reducing donor response time and enhancing user engagement, as reflected in the quantitative analysis conducted. Our research paves the way for future explorations into the deployment of multi-platform bots in healthcare and other domains. We hope to contribute to the ongoing efforts to improve blood donation outreach and facilitate life-saving connections in our communities.

References

[1] 2014. A Framework for a Smart Social Blood Donation System Based on Mobile Cloud Computing. *Health Informatics - An International Journal* 3 (2014), 1412–7276. Issue 4.

[2] Advait Agnihotri, Harshit Gautam, and Hari Singh. 2022. Discord Bot. (2022).

[3] Turki Alanzi and Batool Alsaeed. 2019. Use of Social Media in the Blood Donation Process in Saudi Arabia. *Journal of Blood Medicine* 10 (12 2019), 417–423. https://doi.org/10.2147/JBM.S217950

[4] America's Blood Centers. 2024. Blood Donation Statistics and Information Guide. https://americasblood.org/statistics_guide/. Accessed: 2024-08-27.

[5] Kaikai An, Shuzheng Si, Helan Hu, Haozhe Zhao, Yuchi Wang, Qingyan Guo, and Baobao Chang. 2024. Rethinking Semantic Parsing for Large Language Models: Enhancing LLM Performance with Semantic Hints. *arXiv preprint arXiv:2409.14469* (2024).

[6] Hiteshwar Kumar Azad and Akshay Deepak. 2017. Query Expansion Techniques for Information Retrieval: a Survey. *arXiv preprint arXiv:1708.00247* (2017). https://doi.org/10.48550/arXiv.1708.00247

[7] Siti Rahayu Abdul Aziz, Adlan Al-Farooq Razalan, Noorhayati Mohamad Noor, and Mohd Suhaimi Sauti. 2010. Proactive notification system using instant messaging bot (IM bot). In *2010 International Conference on Science and Social Research (CSSR 2010)*. IEEE, 695–698.

[8] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. 2006. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Philadelphia, PA, USA) *(KDD '06)*. Association for Computing Machinery, New York, NY, USA, 44–54. https://doi.org/10.1145/1150402.1150412

[9] Sasha Butenko. 2024. Chatbots vs Apps Explained in 7 Points. https://www.jasoren.com/bots-vs-apps-explained-in-7-points/#:~:text=Chatbots%20are%20simpler%20and%20faster,of%20the%20precious%20smartphone%20memory Accessed: 2024-07-25.

[10] Laura Ceci. 2023. *Time Spent on Facebook Messenger App in Selected Countries*. https://www.statista.com/statistics/1295008/time-spent-facebook-messenger-app-selected-countries/ Accessed: [Insert Access Date Here].

[11] Hao-Yung Chan and Meng-Han Tsai. 2023. Alert notifications for governmental disaster response via instant messaging applications. *International Journal of Disaster Risk Reduction* 96 (2023), 103984. https://doi.org/10.1016/j.ijdrr.2023.103984

[12] Victor Chang. 2018. A proposed social network analysis platform for big data analytics. *Technological Forecasting and Social Change* 130 (2018), 57–68. https://doi.org/10.1016/j.techfore.2017.11.002

[13] Brian Dean. 2023. WhatsApp Users: The Ultimate Guide. *Backlinko* (2023). https://backlinko.com/whatsapp-users Last updated: December 12, 2023.

[14] Brian Dean. 2024. *Facebook Users: The Ultimate Guide*. https://backlinko.com/facebook-users Last updated: February 23, 2024.

[15] Liang Guo, Xirong Que, Yidong Cui, Wendong Wang, and Shiduan Cheng. 2012. A Hybrid Social Search Model Based on the User's Online Social Networks. In *Proceedings of the IEEE CCIS 2012*. IEEE, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China.

[16] Hunor Hegedűs, Kata Szász, Károly Simon, Tibor Fazakas, Andor Mihály, and Katalin Nagy. 2019. Blood Notes: Software System for Promoting and Facilitating Blood Donation. In *2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY)*. 77–82. https://doi.org/10.1109/SISY47553.2019.9111536

[17] Chen Huang and Guoxiu He. 2024. Text Clustering as Classification with LLMs. *arXiv preprint arXiv:2410.00927* (2024).

[18] Zhiqiu Jiang, Mashrur Rashik, Kunjal Panchal, Mahmood Jasim, Ali Sarvghad, Pari Riahi, Erica Dewitt, Fey Thurber, and Narges Mahyar. 2024. CommunityBots: Creating and Evaluating a Multi-Agent Chatbot Platform for Public Input Elicitation. (2024).

[19] Abhishek Joshi, CS Nagarjun, and Ravi Srinivas. 2017. The DRASB—disaster response and surveillance bot. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 1–8.

[20] Christopher S.G. Khoo. 2014. Issues in information behaviour on social media. *Library and Information Science Research E-Journal* 24, 2 (2014), 75–96. https://doi.org/10.32655/LIBRES.2014.2.2

[21] Khoros. 2024. Social Media Demographics Guide. https://khoros.com/resources/social-media-demographics-guide#:~:text=Social%20media%20usage%20by%20age&text=The%20next%20closest%20age%20group,with%20only%2036.9%20million%20users. Accessed: 2024-08-27.

[22] Johnson Kolluri, Vinay Kumar Kotte, MSB Phridviraj, and Shaik Razia. 2020. Reducing overfitting problem in machine learning using novel L1/4 regularization method. In *2020 4th international conference on trends in electronics and informatics (ICOEI)(48184)*. IEEE, 934–938.

[23] Vladyslav Kruglyk, Dmitriy Bukreiev, Pavlo Chornyi, Evgeniy Kupchak, and Andrey Sender. 2020. Discord platform as an online learning environment for emergencies. *Ukrainian Journal of Educational Studies and Information Technology* 8, 2 (2020), 13–28.

[24] Bruce R Lindsay. 2011. Social media and disasters: Current uses, future options, and policy considerations.

[25] Md Rafat Jamader Maraz, Rashik Rahman, Md. Mehedi Ul Hasnain, and Hasan Murad. 2021. A Cross-Platform Blood Donation Application with a Real-Time, Intelligent, and Rational Recommendation System. In *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*. 1–4. https://doi.org/10.1109/ICECIT54077.2021.9641395

[26] Lynnette Hui Xian Ng and Kathleen M. Carley. 2022. BotBuster: Multi-platform Bot Detection Using A Mixture of Experts. *Computer Science > Social and Information Networks* (July 2022). Submitted on 27 Jul 2022.

[27] S. Ouhbi, J. L. Fernandez-Alemán, A. Toval, A. Idri, and J. R. Pozo. 2015. Free blood donation mobile applications. *Journal of Medical Systems* 39, 5 (2015), 1–20.

[28] Lakshmi Prasanna, C. Kula Deekshith, C. Yamini, C. Harish, CH. Girish, A. Sravani, and C. Tejaswini. 2024. Online Blood Donation Management System. *International Journal of Research Publication and Reviews* (2024).

[29] Shannon M Rauch and Kimberley Schanz. 2013. Advancing racism with Facebook: Frequency and purpose of Facebook use and the acceptance of prejudiced and egalitarian messages. *Computers in Human Behavior* 29, 3 (2013), 610–615.

[30] Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*. 1–7.

[31] Sivasurya Santhanam, Tobias Hecking, Andreas Schreiber, and Stefan Wagner. 2022. Bots in software engineering: a systematic mapping study. *PeerJ Computer Science* 8 (2022), e866.

[32] Kanish Shah, Henil Patel, Devanshi Sanghvi, and Manan Shah. 2020. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research* 5, 1 (2020), 12.

[33] Rakesh Sharma, Debadri Banerjee, Anupama Singh, and Vikas Anand Saharan. 2022. Smart approaches for encouraging the blood donation. *Asian Journal of Transfusion Science* (2022).

[34] Khris Steven. 2024. Facebook Messenger Statistics: What You Need to Know. *Persuasion Nation* (2024). https://persuasion-nation.com/facebook-messenger-statistics/ Last updated: June 26, 2024.

[35] Backlinko Team. 2024. *Discord Users: The Ultimate Guide*. https://backlinko.com/discord-users Last updated: June 13, 2024.

[36] Dijana R. Vukovic and Igor M. Dujlovic. 2016. Facebook messenger bots and their application for business. In *2016 24th Telecommunications Forum (TELFOR)*. 1–4. https://doi.org/10.1109/TELFOR.2016.7818926

[37] Dijana R. Vukovic and Igor M. Dujlovic. 2016. Facebook messenger bots and their application for business. In *2016 24th Telecommunications Forum (TELFOR)*. 1–4. https://doi.org/10.1109/TELFOR.2016.7818926

[38] Anita Whiting and David Williams. 2013. Why people use social media: a uses and gratifications approach. *Qualitative market research: an international journal* 16, 4 (2013), 362–369.

[39] Reda Yacouby and Dustin Axman. 2020. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems*. 79–91.

[40] Tengjun Yao, Zhengang Zhai, and Bingtao Gao. 2020. Text Classification Model Based on fastText. In *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*. 154–157. https://doi.org/10.1109/ICAIIS49377.2020.9194939

[41] Adesope Rebecca Yinka and Nwaizugbu Nkeiruka Queendarline. 2018. Telegram as a social media tool for teaching and learning in tertiary institutions. *International Journal of Multidisciplinary Research and Development* 5, 7 (2018), 95–98.

[42] Iffah Nur Diyana Zainalabidin, Azrul Amri Jamal, Mokhairi Makhtar, and Mohamad Afendee Mohamed. 2017. Blood Donation Notification System Based on Donor's Location Using Haversine Formula. *World Applied Sciences Journal* (December 2017).

[43] Veronica Zambon. 2020. Understanding and managing digital burnout. *Medical News Today* (June 11 2020). https://www.medicalnewstoday.com/articles/digital-burnout Medically reviewed by Alana Biggers, M.D., MPH.

A Survey Questionnaire

We surveyed 114 participants, including 38 potential donors and gathered valuable insights on their satisfaction levels and open feedback regarding challenges and suggestions for improvement. This provided valuable contributions to our work. The survey questions are given below:

For Users:

(1) Do you request blood donations on social media (e.g., Telegram, Discord, etc.)?
(Almost always, Often, Sometimes, Seldom, Never)

(2) Did you usually receive timely responses to your blood donation requests before using BNet prior to October 23, 2024?
(Almost always, Often, Sometimes, Seldom, Never)

(3) How satisfied are you with the timely response of BNet in identifying potential donors between October 23 and October 26, 2024, after integrating BNet into groups?
(Very satisfied, Satisfied, Neither, Dissatisfied, Very dissatisfied)

(4) After getting a response from BNet, have you successfully connected with a blood donor through BNet?
(Almost always, Often, Sometimes, Seldom, Never)

(5) How easy do you find using BNet through slash command prompts?
(Extremely easy, Very easy, Moderately easy, Slightly easy, Not at all)
(6) How intuitive is the user interface of BNet?
(Extremely intuitive, Very intuitive, Moderately intuitive, Slightly intuitive, Not at all)
(7) How would you rate the overall functionality of BNet?
(Excellent, Above Average, Average, Below Average, Very Poor)
(8) At most how many blood donation seeking messages do you feel comfortable to receive from BNet per month?
(1-5, 6-10, 11-15, 16-20, 21+)
(9) Do you find BNet more effective than existing blood donation apps or methods you have used before?
(Much better, Somewhat better, Stayed the same, Somewhat worse, Much worse, Not applicable- I have never used any app before)
(10) What challenges do you face in connecting with blood donors? How can these be overcome?
(Open-ended response)
(11) What improvements would you suggest to make BNet better for requesters?
(Open-ended response)

For Donors:

(1) How many times have you donated blood in the past year?
(Never, 1 time, 2 times, 3 times, 4 or more)
(2) Do you have trouble finding blood donation requests among a large volumne of messages in social media groups?
(Almost always, Often, Sometimes, Seldom, Never)
(3) How convenient is BNet in notifying you about blood donation requests in social media groups?
(Extremely convenient, Very convenient, Moderately convenient, Slightly convenient, Not at all)
(4) How would you rate the overall functionality of BNet?
(Excellent, Above Average, Average, Below Average, Very Poor)
(5) Do you find BNet more effective than existing blood donation apps or methods you've used before?
(Much better, Somewhat better, Stayed the same, Somewhat worse, Much worse, Not applicable)
(6) What challenges do you face in connecting with blood requesters? How can these be overcome?
(Open-ended response)
(7) What improvements would you suggest to make BNet better for donors?
(Open-ended response)