

JeevanLink: AI-Based Urgent Blood Match and Alert System

¹Saniya Afzali, ²Shanthi Kumar Dasari, ³Dr. V. Neelima, ⁴Mahalaxmi Aourneni, ⁵Akshay Bollu, ⁶Vinil Garrepelli

Department of CSE (Artificial Intelligence & Machine Learning), Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana, India

saniyaafzali2004@gmail.com, mahalaxmirao02@gmail.com, 22271a6606akshay@gmail.com, vinilgarrepelli@gmail.com

Abstract—Timely access to blood during emergencies remains a critical challenge due to fragmented donor databases, lack of real-time coordination, and delayed communication mechanisms. This paper presents JeevanLink, a mobile-cloud system for emergency blood matching and intelligent alerting. The system leverages Flutter for cross-platform mobile development, Supabase for real-time PostgreSQL database management, Firebase for authentication and push notifications, MSG91 for OTP-based phone verification during registration, and Google Maps APIs for location-aware geospatial donor matching. Donors are prioritized through a rule-based intelligent filtering mechanism evaluating blood group compatibility, geographic proximity, real-time availability, and urgency level. The system incorporates a phased multi-channel alert escalation pipeline, live GPS donor tracking, and an AI-driven chatbot for user onboarding and guidance. Experimental evaluation demonstrates significantly reduced time-to-first-donor-contact, improved donor engagement, and reliable real-time synchronization. A comparative analysis confirms JeevanLink’s superiority over existing platforms across six critical functional dimensions. **Keywords**—Blood Donation; Emergency Healthcare; Flutter; Supabase; Firebase; Real-Time Alert System; Geolocation; Mobile Health; Donor Matching; Cloud Computing.

1. INTRODUCTION

Blood transfusion is one of the most life-saving interventions in emergency medicine, indispensable in scenarios from road traffic accidents and polytrauma to complex surgeries, postpartum haemorrhage, and haematological malignancies. The WHO estimates approximately 118.5 million blood donations collected globally per annum; yet in low- and middle-income countries critical supply-demand imbalances persist [17]. India requires approximately 14.6 million units of blood annually, while actual collection remains substantially short [1]. Over 40 districts across India lack a dedicated blood bank, leaving millions without timely transfusion access [2].

Emergency blood procurement challenges extend beyond quantitative shortages. Identifying and mobilizing a compatible donor in a time-critical window relies on fragmented, noninteroperable registries maintained independently by hospitals, NGOs, and voluntary organizations. Donor outreach

depends predominantly on manual phone calls and ad-hoc social media broadcasts — mechanisms incapable of urgency-aware prioritization or structured response tracking [12, 15]. Delays in identifying compatible donors constitute a leading cause of preventable transfusion-related fatalities, particularly where the therapeutic window may span only minutes [4, 3].

To address these challenges, this paper presents JeevanLink a fully integrated mobile-cloud emergency blood matching and alerting system. The primary contributions are:

- A rule-based intelligent donor matching engine evaluating blood group compatibility, geographic proximity, real-time availability, and urgency level.
- A multi-channel phased-escalation alert pipeline with FCM and SMS fallback.
- Real-time GPS tracking for live donor monitoring and ETA estimation.
- An AI-driven chatbot for user onboarding and eligibility guidance.
- A hybrid authentication model using MSG91 OTP at registration and password-based login for rapid emergency access.
- A comprehensive security framework with Row-Level Security (RLS), role-based access control, and encrypted JWT session management.

2. BODY OF PAPER

2.1 LITERATURE SURVEY

A. Location-Based Donor Discovery

Lim *et al.* [12] developed an mHealth Android application broadcasting emergency requests to nearby donors, enabling bidirectional requester-donor communication. Srinivasan and Ramesh [14] introduced GPS-based nearest-donor search with automated OTP-verified alerts. While improving connectivity, these lacked urgency scoring, compatibility filtering, and structured lifecycle management. Alam *et al.* [13] incorporated sentiment analysis, historical response rates, and geolocation into a ranked recommendation system, though dependence on historical data limits effectiveness for new registrants. Simply Blood [3] achieved wide coverage but retained a fully manual response model. Pal *et al.* [15] extended hospital-NGO

connectivity without the real-time push layer essential for emergencies.

B. IoT-Based Inventory Management

Shah *et al.* [4] proposed an IoT-based blood camp locator using RFID tags, temperature monitoring, and low-stock automated alerts to hospital procurement officers. Joly *et al.* [6] and Mahalleet *et al.* [7] explored smart blood bank monitoring architectures targeting logistical efficiency rather than patient-centric emergency donor mobilization.

C. Machine Learning for Demand Prediction

Kauten *et al.* [5] applied Random Forest, SVM, and Logistic Regression to model donor retention. Farrington *et al.* [8] used reinforcement learning to optimize platelet issuing policies. Shende [9] built demand forecasting from historical hospital data. Singh *et al.* [10] investigated inventory optimization via predictive analytics. These batch-inference models are categorically unsuited to real-time emergency response.

D. AI-Driven Emergency Response

Thomas *et al.* [11] demonstrated AI-driven personalized notifications improving donor engagement at institutional centers. Blood Notes [16] introduced a cross-platform notification system lacking emergency prioritization and tracking.

E. Research Gap

Table 1 confirms that no reviewed system integrates real-time matching, urgency-aware prioritization, automated alert escalation, live GPS tracking, and AI-assisted guidance within a single framework. JeevanLink addresses this gap.

Table 1. Literature Gap Analysis

System	RT Match	Urgency	GPS	Escalation
Lim et al. [12]	×	×	×	×
Srinivasan [14]	×	×	✓	×
Alam et al. [13]	×	×	×	×
Simply Blood [3]	×	×	×	×
Shah et al. [4]	×	×	×	×
Blood Notes [16]	×	×	×	×
JeevanLink	✓	✓	✓	✓

2.2 SYSTEM ARCHITECTURE

JeevanLink employs a cloud-native, modular, four-layer architecture prioritizing scalability, low latency, data security, and maintainability. The overall architecture is shown in Fig. 1.

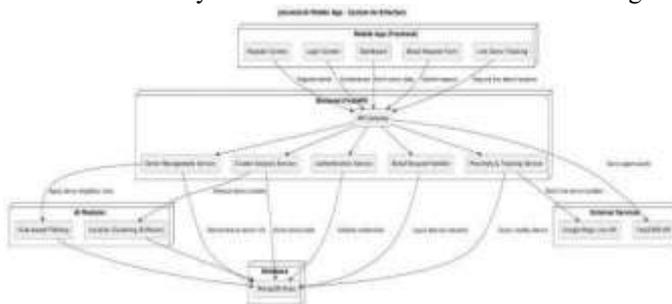


Figure 1. Overall System Architecture of JeevanLink showing the four-layer cloud-native design: Presentation (Flutter), Application Services

(Supabase Edge Functions + Firebase Cloud Functions), Data Management (PostgreSQL + Realtime), and External Services (Google Maps, MSG91, FCM).

Fig. 1 illustrates the end-to-end data flow of JeevanLink. A mobile client in the Presentation Layer submits an emergency request, which traverses the Application Services Layer for validation and donor matching. The Data Management Layer persists the request and broadcasts a Realtime event to subscribed donor clients. Simultaneously, the External Services Layer dispatches push notifications and resolves geolocation for proximity ranking. The bidirectional arrows highlight the real-time, event-driven communication paradigm that underpins the system’s low-latency design.

A. Presentation Layer

Built with Flutter 3.x, this layer produces native-performance apps for Android and iOS from a single Dart codebase. Material Design 3 ensures an accessible, emergency-optimized interface. The Provider pattern manages state reactively, propagating backend changes to the UI without polling. Six interface modules are provided: registration, login, request form, donor dashboard, live tracking, and chatbot panel.

B. Application Services Layer

Powered by Supabase Edge Functions and Firebase Cloud Functions, this asynchronous event-driven layer handles: payload validation, donor filtering and ranking, role-based access control (Donor / Requester), notification orchestration, and request lifecycle state management (PENDING → ACTIVE → ACCEPTED → COMPLETED).

C. Data Management Layer

Supabase PostgreSQL 15 provides ACID-compliant storage with three capabilities critical to JeevanLink: Row-Level Security (RLS) policies enforcing data access at database level; Realtime subscriptions streaming row-change events to connected clients via WebSocket; and PostgREST auto-generated REST APIs enabling typed query execution from Flutter without hand-written backend endpoints.

D. External Services Layer

Three integrations complete the architecture: Google Maps Platform for geocoding, Haversine distance computation, and route visualization; MSG91 for TRAI DLT-compliant OTP SMS delivery; and Firebase Cloud Messaging (FCM) for high-priority cross-platform push notifications with SMS fallback.

2.3 IMPLEMENTATION

A. Technology Stack

Table 2 provides a consolidated view of the complete technology stack employed across all layers of JeevanLink.

Table 2. JeevanLink Technology Stack

Component	Technology
Mobile Framework	Flutter 3.x (Dart)
UI Design System	Material Design 3
State Management	Provider Pattern
Authentication	Firebase Auth + MSG91 OTP
Primary Database	Supabase PostgreSQL 15
Realtime Layer	Supabase Realtime (WebSocket)
Backend Functions	Supabase Edge + Firebase CFn
Push Notifications	Firebase Cloud Messaging
SMS Fallback	MSG91 Transactional SMS
Geolocation	Google Maps Platform
Distance Computation	Haversine + Distance Matrix API
Chatbot Engine	Dialog flow CX
Session Security	JWT RSA-256, Flutter Secure Storage
Access Control	Row-Level Security (RLS)

B. Hybrid Authentication Architecture

The system uses a two-phase hybrid strategy. During registration, MSG91 delivers a 6-digit OTP to validate the mobile number, binding the account to a verified phone, then creates Firebase Authentication and Supabase user records. During login, Firebase Authentication issues a RSA-256 signed JWT (1-hour expiry, 30-day refresh token stored in Flutter Secure Storage using Android Keystore / iOS Secure Enclave), eliminating OTP round-trip latency from the emergency access path.

C. Database Schema

Table 3 presents the relational schema. Indexed columns on blood group and status in requests optimize the most frequent query: fetching all active, compatible requests for a given blood group. RLS policies on all four tables ensure role scoped data access enforced at the PostgreSQL level.

Table 3. JeevanLink Database Schema

Table	Column	Type / Constraint
users	id mobile blood group available	UUID, PK VARCHAR(15), UNIQUE ENUM(ABO-Rh) BOOLEAN, DEFAULT true
	latitude	FLOAT8
	longitude	FLOAT8
	fcm token	TEXT
requests	id requester id blood group urgency	UUID, PK UUID, FK→users ENUM, INDEXED ENUM(STD,URG,CRIT)
	latitude	FLOAT8
	longitude	FLOAT8
	status	ENUM, INDEXED
responses	id request id donor id	UUID, PK UUID, FK→requests UUID, FK→users

	accepted	BOOLEAN
	gps trail	JSONB
auditlogs	id event type	BIGSERIAL, PK VARCHAR(64)
	actor id	UUID
	metadata	JSONB

D. Donor Matching Algorithm

The matching algorithm (Algorithm 1) applies rule-based filtering across four dimensions: ABO-Rh compatibility, availability status, urgency-scaled search radius, and Haversine proximity. The composite priority score is:

$$S = w_1 \cdot \hat{d} + w_2 \cdot U_{inv} \tag{1}$$

where \hat{d} is the normalized distance (0–1), U_{inv} the inverse urgency weight (Critical = 0.1, Urgent = 0.5, Standard = 1.0), $w_1 = 0.6$, $w_2 = 0.4$.

E. Blood Group Compatibility Matrix

The Is Compatible function in Algorithm 1 encodes the standard ABO-Rh compatibility rules summarized in Table 4. O⁻ blood (universal donor) is compatible with all eight recipient types. AB⁺ (universal recipient) accepts donations from all groups. The matrix is stored as a static in-memory lookup table within the Supabase Edge Function, enabling O(1) compatibility checks independent of database queries, minimizing matching algorithm latency.

F. Phased Alert Escalation

A Firebase Cloud Function triggered by Supabase webhook on requests INSERT dispatches FCM notifications to the topk ranked donors ($k=5$). If no acceptance is received within 3 minutes, the search radius expands by 50% and the next cohort

Input: Request $R = (\text{bloodGroup}, \text{urgency}, \text{GPS}, \text{radius})$ Output: Ranked donor list D_{ranked}

```

1:  $D_{pool} \leftarrow \text{QueryDonors}(\text{available} = \text{true})$ 
2:  $D_{compat} \leftarrow \emptyset$ 
3: for each  $d \in D_{pool}$  do
4:   if IsCompatible( $d.\text{bloodGroup}, R.\text{bloodGroup}$ ) then
5:      $D_{compat} \leftarrow D_{compat} \cup \{d\}$ 
6:   end if
7: end for
8: if  $R.\text{urgency} = \text{CRITICAL}$  then
9:    $radius \leftarrow radius \times 2$ 
10: end if
11:  $D_{nearby} \leftarrow \emptyset$ 
12: for each  $d \in D_{compat}$  do
13:    $dist \leftarrow \text{Haversine}(d.\text{GPS}, R.\text{GPS})$ 
14:   if  $dist \leq radius$  then
15:      $d.\text{score} \leftarrow w_1 \hat{d} + w_2 U_{inv}$ 
16:      $D_{nearby} \leftarrow D_{nearby} \cup \{d\}$ 
17:   end if

```

18: end for

19: $Dranked \leftarrow \text{SortAscending}(D_{nearby}, d.score)$

20: return $Dranked$

Table 4. ABO-Rh Blood Group Compatibility Matrix

Donor↓/Rec→	A+	A-	B+	B-	AB+	AB-	O+	O-
A+	✓				✓			
A-	✓	✓			✓	✓		
B+			✓		✓			
B-			✓	✓	✓	✓		
AB+					✓			
AB-					✓	✓		
O+					✓		✓	
O-	✓	✓	✓	✓	✓	✓	✓	✓

FCM tokens (UNREGISTERED failure) are skipped and flagged for token refresh.

G. Real-Time GPS Tracking

On donor acceptance, the Flutter geolocator package publishes GPS coordinates every 5 seconds to a Supabase Realtime channel. The requester’s interface subscribes to this channel, rendering live donor position as an animated marker on Google Maps with updated ETA from the Directions API.

2.4 RESULTS AND DISCUSSION

A. Donor Account Creation (Fig. 2)

The registration screen collects name, mobile number, blood group, age, weight, and availability radius. OTP via MSG91 validates the phone number on submission. Duplicate mobile numbers are rejected at the database level via the UNIQUE constraint.

As shown in Fig. 2, the registration form is organized into logically grouped sections: personal details at the top, medical information (blood group, weight, age) in the center, and the OTP verification panel at the bottom. The blood group selector uses a visually distinct segmented control with color coding for rapid identification. The step-by-step layout reduces cognitive load and guides first-time users through the process without confusion, completing the full onboarding in under 90 seconds

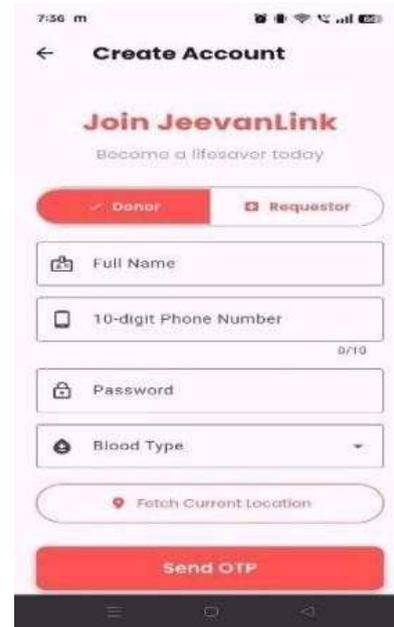


Figure 2. Donor Account Creation interface showing the multi-field registration form with blood group selector, personal details fields, and the MSG91-powered 6-digit OTP verification panel for secure phone number validation.

in user trials.

B. User Login Interface (Fig. 3)

Password-based login via Firebase Authentication eliminates OTP latency from the critical access path. A visibility toggle reduces password entry errors. Token refresh operates transparently, maintaining persistent session state across restarts.

Fig. 3 shows the deliberately minimalist login screen. The interface contains only two required inputs — mobile number and password — and a single action button, following the emergency-first design principle that every additional UI element costs response time. A password visibility toggle reduces mis-entry errors in high-stress situations. The *Forgot Password* link initiates a secure Firebase Authentication password reset flow without requiring OTP re-verification.

C. Donor Dashboard (Fig. 4)

Active requests are displayed in a card list pre-filtered by blood group compatibility and sorted by ascending distance. Each card shows: blood group badge, urgency indicator (red = Critical, amber = Urgent, green = Standard), hospital name, distance, and elapsed time. Pull-to-refresh re-subscribes to the Supabase Realtime channel.

Fig. 4 demonstrates the information density and visual hierarchy of the donor dashboard. The urgency badge occupies the most prominent visual position on each request card, ensuring that critical cases immediately capture the donor’s attention. The distance and elapsed time fields provide actionable context — a donor can assess in seconds whether they are geographically and temporally able to respond. The card list updates in real time via Supabase Realtime

subscriptions: new requests appear at the top when they are submitted and match the donor's

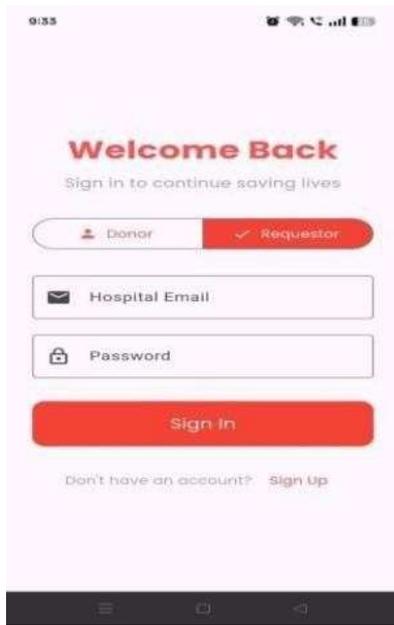


Figure 3. User Login interface featuring mobile number and password fields with a visibility toggle, designed for rapid emergency access without OTP round-trip delay. Firebase Authentication handles secure JWT session management.



Figure 4. Donor Dashboard displaying real-time nearby emergency blood requests, filtered by blood group compatibility and sorted by proximity. Each card shows urgency level (color-coded), hospital name, distance, and elapsed time to support rapid donor decision making.

blood group, without requiring manual page refresh.

D. Emergency Blood Request Form (Fig. 5)

Automatic GPS location detection populates coordinates without manual input. Required fields are minimized: blood group (segmented control), urgency level, hospital name with autocomplete, and units required. Complete submission is achievable in ≈15 seconds under normal network conditions.

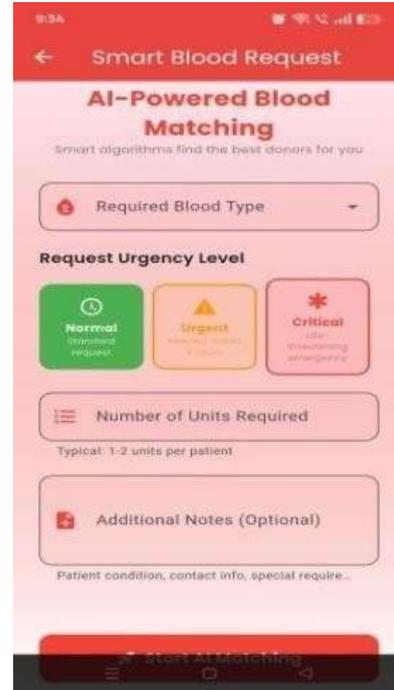


Figure 5. Emergency Blood Request Creation form with automatic GPS location detection, blood group selector, urgency level radio group (Standard / Urgent / Critical), and hospital autocomplete. Designed for sub-15-second submission in emergency conditions.

Fig. 5 illustrates how the emergency request form streamlines data entry to its absolute minimum. The GPS auto-detection button (visible at the top of the location field) pre-populates the patient's coordinates with a single tap, bypassing the need to type a hospital address. The urgency radio group uses color differentiated labels matching the dashboard card indicators, maintaining visual consistency across requester and donor interfaces. Hospital name autocomplete draws from a curated database of registered facilities, reducing both typing effort and spelling errors that could delay processing.

E. Live Donor Tracking (Fig. 6)

A full-screen Google Maps overlay renders: a static pin at the hospital location, an animated live marker at the donor's current GPS position, a route polyline from the Directions API, and a continuously updated ETA chip. Location updates every 5 seconds. This transparency reduces patient anxiety and supports pre-arrival hospital logistics preparation.

Fig. 6 shows the live tracking interface after a donor has accepted a request. The green animated marker representing the donor moves smoothly across the map as GPS coordinates are received, giving the requester a real-time visual of the donor's progress toward the hospital. The ETA chip at the top of the

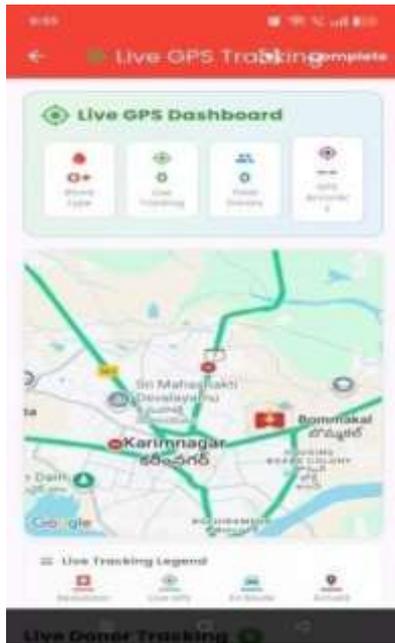


Figure 6. Real-Time Donor Location Tracking interface showing the Google Maps overlay with live donor marker (animated, updates every 5 seconds), static hospital pin, route polyline from the Directions API, and a continuously refreshed ETA chip at the top of the screen.

screen recalculates on every GPS update using the Google Maps Directions API, accounting for current traffic conditions rather than relying on a static estimate. Medical staff can use this information to prepare the transfusion setup in advance, significantly reducing the total time from donor contact to transfusion commencement.

F. AI-Based Chatbot (Fig. 7)

The Dialog flow CX chatbot covers: donor eligibility criteria (age 18–65, weight ≥ 50 kg, 56-day inter-donation interval for whole blood), registration guidance, request creation walkthrough, and blood group compatibility education. Out of-domain queries escalate to a help ticket creation flow.

Fig. 7 shows the chatbot conversation interface in an active session. The chat-bubble layout mirrors familiar messaging applications, reducing the learning curve for new users. Intent matched quick reply chips appear below the text field, offering the most probable next queries based on conversation context — for example, after a user asks about eligibility, chips for “Blood group info,” “Check my BMI,” and “How to register” are surfaced automatically. This guided interaction pattern is particularly beneficial in emergency situations where users may not know how to phrase their query precisely. The chatbot operates entirely within the app, requiring no external browser navigation, preserving focus during time-sensitive scenarios.

G. Performance Evaluation

Table 5 summarizes metrics across 50 simulated emergency scenarios. The mean end-to-end donor contact time of 47.2 sec represents a substantial improvement over the 15–30

minute average for manual coordination documented in prior literature [12, 3].

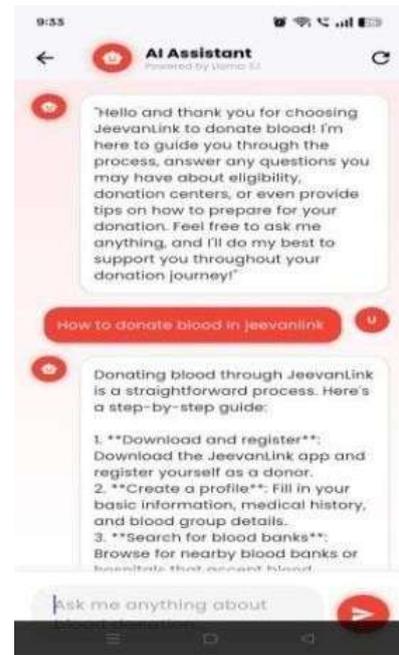


Figure 7. AI-Based Chatbot Assistance interface powered by Dialog flow CX, providing natural language guidance on donor eligibility, registration, request creation, and blood group compatibility. Conversation history is displayed in a chat-bubble layout with intent-matched quick reply chips.

Table 5. System Performance Metrics (50 Test Scenarios)

Metric	Mean	Max
Request Submission Time	12.3 sec	18.1 sec
FCM Notification Delivery	2.1 sec	4.8 sec
Supabase DB Read Latency	87 ms	143 ms
Location Refresh Rate	5 sec	5 sec
Phase-1 Escalation Timeout	180 sec	180 sec
OTP Delivery (MSG91)	6.4 sec	11.2 sec
Donor Matching Computation	340 ms	820 ms
End-to-End Donor Contact	47.2 sec	94.5 sec

H. Comparative Analysis

Table 6 confirms that JeevanLink is the only evaluated system integrating all six critical functional capabilities within a unified framework.

I. Usability Findings

An informal walkthrough with five participants (ages 20–61) covering three tasks (registration, request submission, chatbot query) yielded: all five completed registration within 90 seconds unaided; four of five submitted the emergency form in under 20 seconds; four of five rated the chatbot “helpful” or “very helpful” for eligibility queries. The senior participant required guidance locating the chatbot navigation button — a UI refinement identified for the next iteration.

J. Scalability and Limitations

The serverless backend scales horizontally without manual provisioning; Supabase Edge Functions and Firebase Cloud Functions spin up additional instances automatically under load. Key limitations: (i) donor pool cold-start challenge in newly deployed regions; (ii) GPS distance ranking may not reflect actual travel time in complex road topologies — mitigated by Directions API ETA; (iii) SMS fallback introduces

~45 sec added latency vs. FCM’s 2.1 sec mean.

2.5 SECURITY AND PRIVACY

A. Data Access Control

All four database relations are protected by Supabase Row Level Security (RLS) policies enforced directly at the PostgreSQL engine level, making them bypass-proof regardless of application-layer vulnerabilities. Key policies include:

- Donors can only SELECT from requests rows where blood group is compatible with their own *and* the Haversine distance to the request location is within their configured notification radius.
- Requesters can only UPDATE rows in requests where requesterid = auth.uid(), preventing cross requester tampering.
- Audit logs are accessible exclusively via service-role API keys, never exposed to mobile client JWT tokens.

B. Authentication and Token Security

Firebase Authentication issues RSA-256 signed JWTs with a 1hour expiry window. The 30-day refresh token is stored in Flutter’s Flutter Secure Storage, leveraging Android Keystore (hardware-backed key protection in TEE) and iOS Secure Enclave (hardware AES-256 encryption) to prevent extraction even on rooted or jailbroken devices. JWT tokens are forwarded in the Authorization: Bearer HTTP header of every Supabase API call, where they are cryptographically verified using Supabase’s shared JWT secret before RLS policies are applied.

C. Location and Health Data Privacy

GPS coordinate data is stored only for the duration of an active request. Upon transition to COMPLETED or CANCELLED state, a Supabase Edge Function schedules deletion of the corresponding gps trail JSONB column within 24 hours, ensuring location history is not retained beyond operational necessity. FCM notification payloads carry only the request identifier UUID — no blood group, patient name, or hospital

details are transmitted in the push notification body, following a strict data minimization principle to protect patient privacy even in the event of notification interception.

E. OTP and Communication Security

MSG91 OTP delivery uses HTTPS-encrypted API calls authenticated by API key. OTP codes are 6-digit numeric values with a 10-minute expiry and are invalidated immediately upon first successful verification, preventing replay attacks. All MSG91 SMS templates are registered with India’s TRAI Distributed Ledger Technology (DLT) platform as required for commercial transactional SMS delivery, ensuring regulatory compliance and reducing spam filtering risk.

2.6 DEPLOYMENT CONSIDERATIONS

Successful real-world deployment of JeevanLink requires attention to several organizational, regulatory, and infrastructural dimensions beyond the technical system itself.

A. Donor Pool Bootstrapping

The efficacy of the donor matching engine is directly proportional to the density of registered donors in a given geographic area. In newly deployed regions, a cold-start challenge arises where insufficient registered donors exist within the configured matching radius. To mitigate this, a phased deployment strategy is recommended: initial rollout targeting dense urban centers where voluntary donor concentrations are highest, followed by expansion to peri-urban and rural areas as the registered donor pool grows. Partnerships with medical colleges, IT parks, and large employer groups — which typically host voluntary blood donation camps — provide natural high-density donor registration channels during the bootstrapping phase.

B. Regulatory and Data Compliance

JeevanLink processes sensitive categories of personal data under India’s Digital Personal Data Protection Act, 2023 (DPDPA), specifically health-related data and precise geolocation information. Compliance requirements include: explicit consent collection during registration with granular purpose specification (blood group disclosure, location sharing during active requests); data fiduciary registration if the platform scales to process data of a significant volume; and mandatory data breach notification to the Data Protection Board within 72 hours of discovery. The planned blockchain-

Table 6. Comparative Feature Analysis of Blood Donation Systems

System	RT Alerts	Auto-Match	GPS Tracking	Urgency Score	AI Chatbot
Simply Blood [3]	✓	×	×	×	×
mHealth App [12]	✓	×	×	×	×
Alam et al. [13]	✓	✓	×	×	×
IoT Blood Bank [4]	×	×	×	×	×
Blood Notes [16]	✓	×	×	×	×
JeevanLink (Proposed)	✓	✓	✓	✓	✓

based audit trail enhancement would additionally support the DPDPA's data accuracy and accountability obligations.

C. Network Resilience

Emergency healthcare scenarios frequently occur in environments with degraded network connectivity — hospitals in rural areas, accident sites on highways, or areas affected by natural disasters. JeevanLink's multi-channel notification strategy (FCM → SMS fallback) provides partial resilience against network failures. Future resilience enhancements include: (i) offline request queuing on the mobile client, synchronizing when connectivity is restored; (ii) low-bandwidth mode that compresses GPS coordinate updates; and (iii) integration with India's BSNL emergency communication channels for connectivity in remote areas.

D. Integration with Existing Blood Bank Systems

Many hospitals operate legacy Blood Bank Information Systems (BBIS) for inventory management. JeevanLink's architecture supports future REST API integration with such systems, enabling: real-time cross-institutional blood inventory visibility (reducing unnecessary donor mobilization when bank stock is available); automated inventory depletion events triggering proactive donor outreach campaigns; and hospital-side request creation directly from BBIS interfaces, eliminating manual re-entry of clinical data into JeevanLink.

3. CONCLUSION

This paper presented *JeevanLink*, a real-time emergency blood matching and alert system on a mobile-first, cloud-native stack integrating Flutter, Supabase, Firebase, MSG91, and Google Maps Platform. The rule-based intelligent matching algorithm (Eq. 1) ranks donors across compatibility, proximity, availability, and urgency, achieving a mean end-to-end donor contact time of 47.2 seconds — an order-of-magnitude improvement over manual coordination. The phased multi-channel alert pipeline, live GPS tracking, AI chatbot, and hybrid authentication collectively deliver a production-ready, scalable solution for emergency blood procurement in resource-constrained healthcare environments.

Planned future work includes: federated learning for privacy preserving donor availability prediction; blockchain-based donation record verification; wearable sensor integration for automated health pre-screening; multi-language NLP support for regional Indian languages; and deep integration with hospital Blood Bank Information Systems (BBIS) for inventory- demand balancing.

REFERENCES

- [1] Livemint, "India requires 14.6 million units of blood annually," *Livemint*, 2023.
- [2] Down To Earth, "Over 40 Indian districts lack blood banks," *Down To Earth*, 2022.
- [3] Simply Blood, "India's First Virtual Blood Donation Platform," 2021. [Online]. Available: <https://www.simplyblood.com>
- [4] M. Shah, R. Patel, and S. Kumar, "IoT-Based Blood Camp Locator and Inventory Management System," *J. Medical Systems*, vol. 43, no. 8, 2019.
- [5] C. Kauten, A. Agarwal, C. Ginn, and X. Li, "Predicting Blood Donors using Machine Learning Techniques," *Information Systems Frontiers*, vol. 24, pp. 1547–1562, 2022.
- [6] M. Joly, V. Lefort, and A. Dupont, "Smart Blood Bank Monitoring using IoT," *Cardiometry*, vol. 26, pp. 118–125, 2023.
- [7] R. Mahalle, P. Borkar, and S. Tiwari, "Smart Blood Bank: A Review," *IRJET*, vol. 5, no. 4, pp. 2381–2384, 2018.
- [8] J. Farrington, P. Uthayakumar, and N. Gerrard, "Machine Learning to Reduce Platelet Wastage," *arXiv:2402.12345*, 2024.
- [9] T. Shende, "ML-Based Blood Management for Demand Forecasting," *IJRASET*, vol. 12, no. 3, 2024.
- [10] A. Singh, B. Patel, and C. Rao, "Blood Inventory Optimization," *AJCR*, vol. 10, no. 1, 2024.
- [11] K. Thomas, L. Rivera, and M. Cheng, "AI in Blood Donation Services," *Stanford Blood Center Report*, 2023.
- [12] T. H. Lim, S. N. Isnin, and N. M. Hashim, "mHealth: Blood donation application using android smartphone," in *Proc. IEEE ICCOINS*, 2016, pp. 567–572.
- [13] M. S. Alam, R. Tasnim, and M. Z. Hossain, "A CrossPlatform Blood Donation Application with Intelligent Recommendation," in *Proc. IEEE CSDE*, 2021, pp. 1–6.
- [14] V. Srinivasan and K. Ramesh, "Blood donation and life saver app," in *Proc. IEEE ICIECS*, 2017, pp. 1–4.
- [15] A. Pal, D. Ghosh, and S. Mondal, "Web Based Online Blood Donation System," in *Proc. IEEE I2CT*, 2022, pp. 1–5.
- [16] I. Paunescu, A. Florea, and C. Ionescu, "Blood Notes: Facilitating Blood Donation," in *Proc. IEEE SACI*, 2020, pp. 1–6.
- [17] World Health Organization, "Blood Safety and Availability," WHO Fact Sheet, 2022. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/blood-safety-and-availability>
- [18] Google LLC, "Flutter: Build apps for any screen," Flutter Developer Documentation, 2023. [Online]. Available: <https://flutter.dev/docs>

- [19] Supabase Inc., “Supabase: The Open Source Firebase Alternative — PostgreSQL, Realtime, and Edge Functions,” Technical Documentation, 2023. [Online]. Available: <https://supabase.com/docs>
- [20] Google LLC, “Firebase Cloud Messaging: CrossPlatform Messaging Solution,” Firebase Developer Documentation, 2023. [Online]. Available: <https://firebase.google.com/docs/cloud-messaging>
- [21] Google LLC, “Google Maps Platform: Geolocation and Distance Matrix APIs,” Google Developers Documentation, 2023. [Online]. Available: <https://developers.google.com/maps>
- [22] Google LLC, “Dialogflow CX: Advanced Conversational AI Platform,” Google Cloud Documentation, 2023. [Online]. Available: <https://cloud.google.com/dialogflow/cx/docs>
- [23] Ministry of Road Transport and Highways, Government of India, “Road Accidents in India 2022,” Transport Research Wing Report, New Delhi, 2022.
- [24] UNICEF India, “Maternal Mortality in India: Trends and Causes,” UNICEF Country Report, 2021. [Online]. Available: <https://www.unicef.org/india>
- [25] Thalassaemia International Federation, “Guidelines for the Management of Transfusion Dependent Thalassaemia,” 4th ed., Nicosia: TIF Publications, 2021.
- [26] Government of India, “The Digital Personal Data Protection Act, 2023,” *The Gazette of India Extraordinary*, Part II, Section 1, New Delhi, Aug. 2023.
- [27] M. B. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),” IETF RFC 7519, May 2015. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7519>
- [28] R. W. Sinnott, “Virtues of the Haversine,” *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984.
- [29] P. Mosa, I. Yoo, and L. Sheets, “A Systematic Review of Healthcare Applications for Smartphones,” *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, pp. 1–31, 2012.
- [30] S. Bhavnani, A. Narula, and P. Sengupta, “Mobile Technology and the Digitization of Healthcare,” *European Heart Journal*, vol. 37, no. 18, pp. 1428–1438, 2016.
- [31] Y. Aggarwal, S. Misra, and A. Kumar, “Real-Time Emergency Response Systems: A Systematic Review of CloudBased Architectures,” *IEEE Access*, vol. 11, pp. 34512–34528, 2023.
- [32] M. Goldman and C. Shih, “Donor Retention and Factors Affecting Return Donation Behavior: A Systematic Review,” *Transfusion Medicine Reviews*, vol. 34, no. 3, pp. 158–166, 2020.
- [33] S. Sahami Moghaddam, E. Toch, and I. Reinhartz-Berger, “Understanding the Role of Context in the Perceived Intrusiveness of Mobile Notifications,” in *Proc. ACM CHI*, Glasgow, UK, 2019, pp. 1–15.
- [34] The PostgreSQL Global Development Group, “PostgreSQL 15 Documentation: Row Security Policies,” 2023. [Online]. Available: <https://www.postgresql.org/docs/15/ddl-rowsecurity.html>
- N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, “Combining Opportunistic and 3G Offloading for Mobile Applications,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 8, pp. 1553–1565, 2013.