

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**

# Hyperlocal Smart Donation Engine: An AI-Assisted, Role-Based Mobile Platform for Donation Coordination and Fulfillment Tracking

Mrs. Rajeswari G<sup>1</sup>, Gowtham B<sup>2</sup>, Karthikeyan P<sup>3</sup>

<sup>1</sup>Assistant Professor, <sup>2,3</sup>Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam, Sivaganga, Tamil Nadu, India

**Abstract:** Organizations face challenges because local donation coordination through phone calls and group messages creates separate information streams which prevent anyone from obtaining complete operational data. The Hyperlocal Smart Donation Engine (HSDE) exists as a mobile app which tracks donations from their initial NGO need announcement until the point of physical product delivery. The system provides different function sets to donors and NGO administrators and volunteers and a super administrator who receives special access controls which the server enforces through Firebase authentication. The evidence module helps reviewers by running YOLOv11n object detection and EasyOCR text extraction on offer photographs before it applies category rules to produce a weighted Evidence Confidence Score. The score shows how many supporting evidence pieces exist but it does not determine authenticity because humans decide to accept or reject offers. The donation session becomes active for an accepted offer which an NGO administrator then transfers to a volunteer who received an invitation. The delivery process must finish before the requested amount becomes available to the volunteer. The developers created the application through React Native and FastAPI and MongoDB Atlas which they tested by running it on an actual Android device. The test photograph received a 0.44 rating which led to its classification as INSUFFICIENT EVIDENCE while the delivery of 7 units against a 10-unit requirement showed a 70% completion rate.

**Keywords:** donation management; hyperlocal computing; role-based access control; YOLOv11n; EasyOCR; decision support; volunteer logistics; React Native; FastAPI

## I. INTRODUCTION

Donated goods have value when they get to places that truly need them. Most places depend on chance meetings and personal relationships to make that happen. When an NGO does not have blankets or food packets it posts a request online. A household that has items may never see that request. Both sides share information about how much's available show pictures and talk about collection plans. They share this information through phone calls. Chat messages but those messages disappear and cannot be used again later. The need is often nearby. Donors and recipients are usually not away. So the practical question is: which local organizations need help today? The system works as a solution to answer that question. The system shows how many items are needed. Nearby donors can respond to those needs until the donated goods get to the hands. There are three weaknesses in this study that need to be looked at. The donation process saves requests, offers, conversations and pictures in software programs. This stops the creation of one donation record. NGO staff have to check each donor picture. This creates a process that becomes hard to manage when there are more donations. The platforms see accepted offers as the step even though the actual items have not started moving yet. The Hyperlocal Smart Donation Engine or HSDE solves the problems that exist in the system. The research paper presents four contributions in its content. One mobile workflow covers posting a need making a pledge, chatting, inviting volunteers assigning tasks and tracking delivery. Access control, for four roles is set up on the server. Firebase helps with user identity. The backend decides who can do what. A transparent evidence module uses object detection, text recognition and rules to create a score. This score helps the NGO without taking control of their decisions. A fulfillment model makes sure an accepted offer is only counted once the donation is delivered. This way the numbers show how many items are actually received.

The remainder is organised as follows: related work (Section II), the proposed system (III), architecture (IV), the evidence module (V), the fulfillment workflow (VI), the implemented application (VII), results (VIII), testing (IX) and conclusions (X).

## II. REVIEW OF LITERATURE

Existing work can be grouped based on what it digitizes. Mobile applications for giving allow users to find causes while they can donate money from locations [1] and donation processing systems now use AI-based features through administrative systems [2]. Food-based research approaches sell merchandise through two methods that combine applications with cloud-based supply chain management for food donation operations [3]. Several applications exist that let food owners find food recipients through their listing and matching services [4]–[6]. The research uses technology and machine learning methods to find the paths for moving extra food between NGOs and compost production facilities [7]. These systems show that digital channels create connections between what suppliers offer and what customers need. The research mainly focuses on building systems that handle identification tasks to distribute types of items based on their names and their specific research areas. The current systems do not have the mix of features that would allow users to track posted needs along, with evidence evaluation and donor-NGO communication and volunteer scheduling and service completion verification. YOLO family detectors do real-time detection using their pass object detection system [8]. The scientific community has done studies that focus on learning detection systems [9]. The detection system finds object types of checking the value of items people give. The HSDE system works as a human reviewer tool that shows detection results and recognized text to make decisions.

Table I. Comparison Of Existing Approaches

| Approach                                 | Typical strength           | Gap addressed by HSDE                        |
|------------------------------------------|----------------------------|----------------------------------------------|
| Manual coordination (calls, chat groups) | Flexible, no setup         | No shared record; hard to audit              |
| Online donation platforms [1], [3]       | Easy discovery and giving  | Little support after the offer is made       |
| NGO management tools                     | Donor and records handling | No evidence review or field logistics        |
| Volunteer coordination tools             | Task scheduling            | Not tied to a specific accepted donation     |
| AI image analysis [8], [9]               | Extracts objects and text  | Needs rules and human oversight to be useful |

Research gap. The current research fails to develop a unified system which connects need publication with evidence-supported review processes and volunteer-verified delivery services under defined role limits. HSDE is designed to fill that gap.

## III. PROPOSED SYSTEM

HSDE provides users with screens and actions that depend on the job they are doing. It also stores all the information related to donations, in the location where every donation is documented.

### A. User Roles and Access Control

I appreciate how the platform outlines four roles as shown in Table II and Fig. 1. The server verifies each role whenever a request is protected. This role verification keeps users from seeing hidden buttons within the application.

Table II. Roles And Permissions

| Role        | What the role can do                                                                                                                    |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Donor       | Browse open needs, submit offers with photographs, chat with the NGO about an offer, track own offers                                   |
| NGO Admin   | Manage the organisation, publish needs, review offers and AI evidence, accept or reject, invite and assign volunteers, monitor sessions |
| Volunteer   | Accept an organisation's invitation, view assigned tasks, update pickup and delivery stages                                             |
| Super Admin | Perform platform-level administration; does not run any NGO's day-to-day donations                                                      |

### B. Needs and Offers

I am an NGO Administrator. I create a need by choosing a category and saying how much money is needed. I write a description for the need. I say how urgent the need is. I also add a location for the need. When the need is first made the status is DRAFT. Later I can change the status to OPEN. When the need is OPEN everyone who is a donor can see the need in the list. If needed I can still cancel the need. When a donor sees the need the donor can respond by making an offer. The offer says how much is donated. The donor can also add a message and up to three private photos. The server keeps all donor identity details. The server keeps request identity facts, timestamps and offer status updates. Clients cannot change these details.

### C. Offer-Specific Communication

For each offer the system creates a chat room. This chat room lets donors talk directly with the NGO that needs help. I can turn off the chat room whenever I want. I control the chat room settings. The system stops any messages from users who are not allowed. The conversation between donors and NGOs stays private.

### D. Controlled Volunteer Onboarding

I invite a volunteer by sending an email. The system connects the invitation to the NGO because it checks the Google account the volunteer will use to sign in. The volunteer must get an invitation before the volunteer can pick an organization. The volunteer cannot join any organization without permission.

### E. Donation Lifecycle and Data Flow

A donation goes through these steps: the need is published the offer is sent evidence is checked the NGO looks at the offer the offer is accepted a session is created the volunteer is assigned pickup is scheduled the donation is on the way. The donation is. Finished (see Fig. 2). The data flow, between the four groups is shown in Fig. 3. The AI step gives a review. The AI never moves the offer forward by itself.

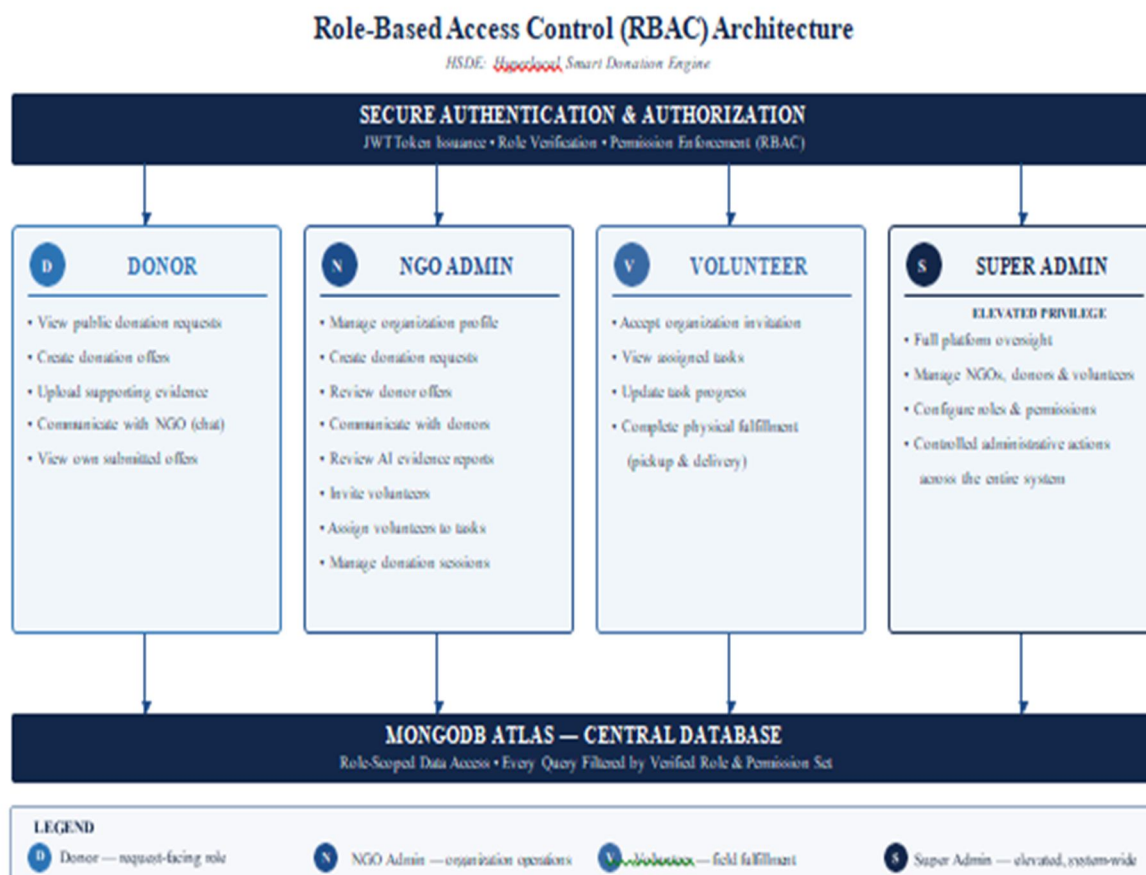


**Fig. 1.** Role-based access control architecture of HSDE showing the four roles and their permitted operations.

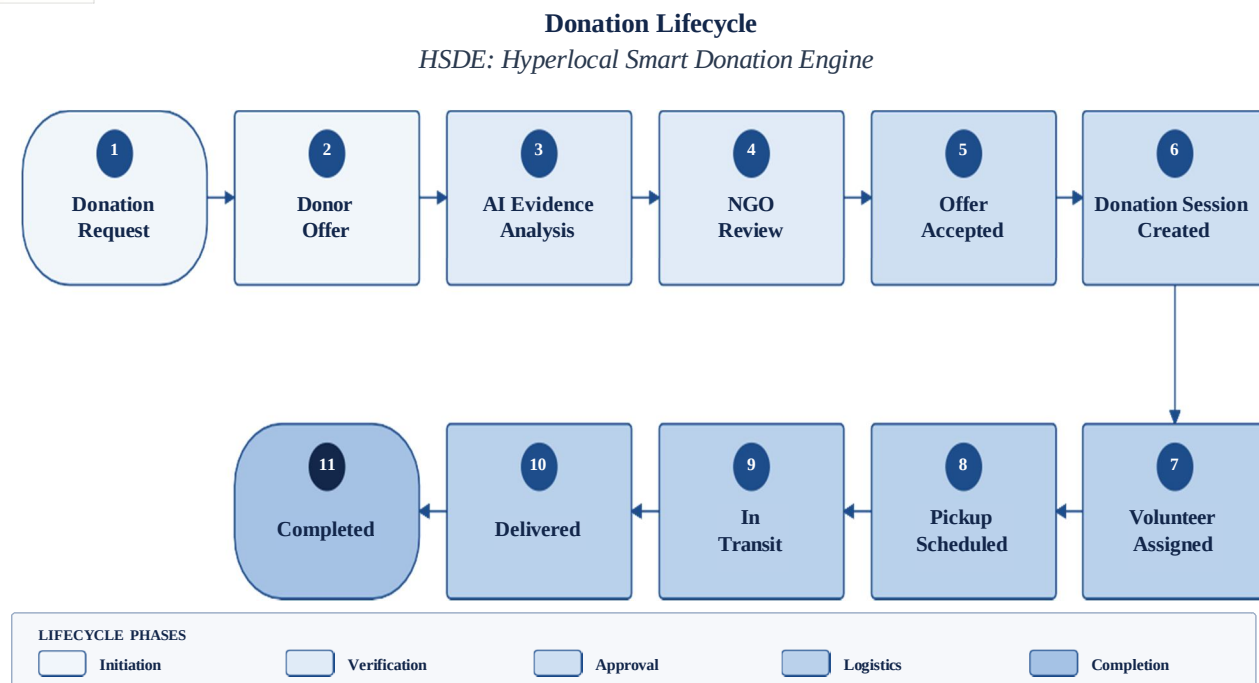


Fig. 2. End-to-end donation lifecycle from published need to completed delivery.

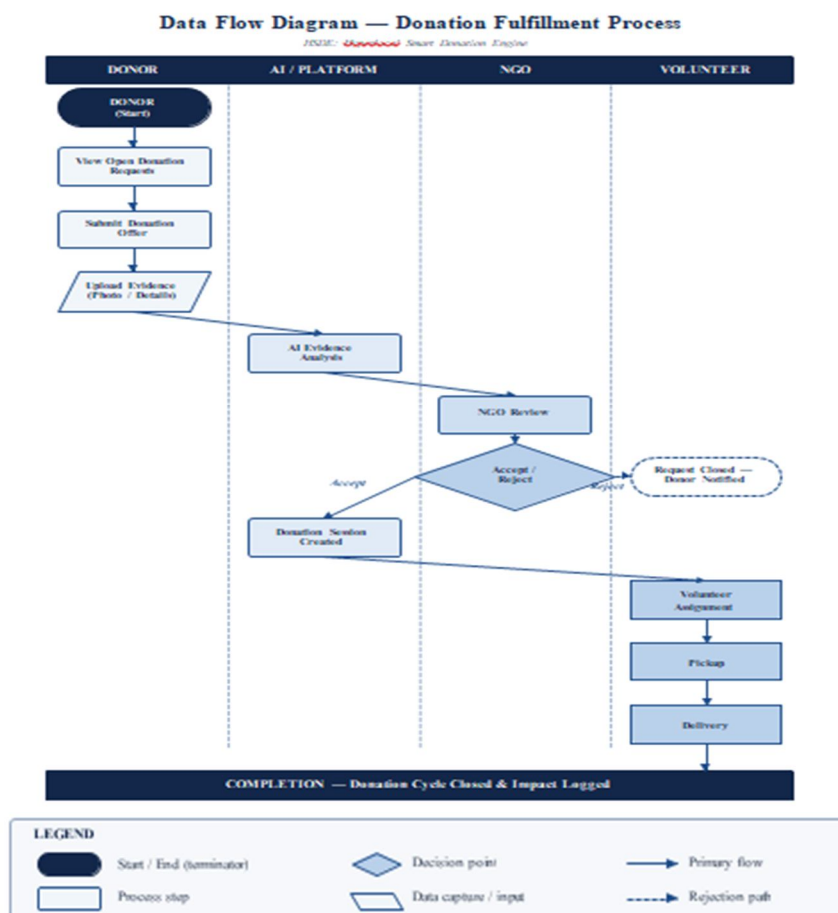


Fig. 3. Data flow diagram of the donation fulfilment process across donor, platform, NGO and volunteer.

#### IV. SYSTEM ARCHITECTURE AND TECHNOLOGY

The system design of HSDE follows a multi-level approach where all requests are routed through a single standardized route from the user input to the access of the database. The React Native client communicates with the FastAPI backend using REST calls, while the chat messages are handled using WebSocket communication.

The system performs identity verification through the backend system. Role-based and organizational access controls are enforced prior to performing business services and AI evidence processing, which saves the data to MongoDB Atlas (Fig. 4).

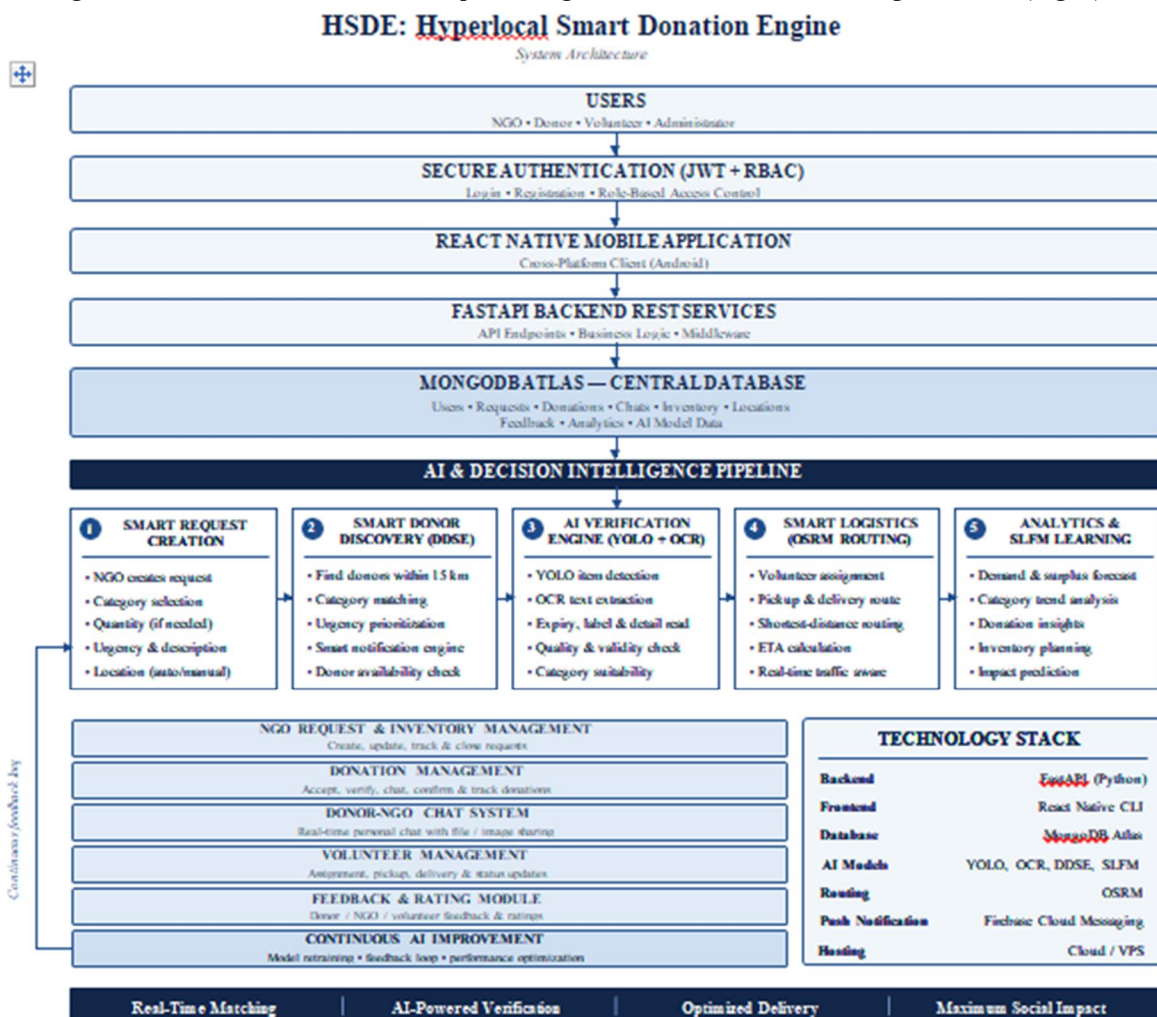


Fig. 4. Layered system architecture of HSDE, from mobile client to backend services, AI pipeline and database.

##### A. Identity and Authorisation

Firebase Authentication is the system that proves who the user is.

An app sends the Firebase ID token with every request after the user completes the Google Sign-In process. The backend uses the Firebase Admin SDK to verify the token and obtain the Firebase UID for the user. Using that UID, the system retrieves the user record in HSDE to determine the user's role. Firebase Authentication identifies the user ("who") while HSDE determines their permissions ("what may this person do"). Organization ownership details are obtained from the server-based databases rather than relying upon any organization identification numbers that the user may provide. The verification process follows the structure below.

```
claims = verify_id_token(token) # Firebase Admin SDK
user = users.find_one(uid=claims.uid)
assert user.role in route.allowed_roles
assert user.org_id == resource.org_id # server-held ownership
```

## B. Technology Stack

TABLE III. TECHNOLOGY STACK

| Layer         | Technology                   | Purpose                                                    |
|---------------|------------------------------|------------------------------------------------------------|
| Mobile client | React Native CLI, TypeScript | Android app with native camera, gallery and sign-in access |

## C. Data Model

MongoDB has a collection for each entity so that every stage of the life cycle has a record (Table IV). The system finds answers, about pledged amounts and accepted values and actual deliveries by using the identification system that links Offers with sessions and tasks.

Table IV. Main Collections

| Collection                         | Holds                                                                           |
|------------------------------------|---------------------------------------------------------------------------------|
| users, organizations               | Identity-linked user records with role; NGO profile and its admin               |
| donation_requests                  | Needs with category, quantity, urgency, location, status and fulfilled quantity |
| donation_offers, offer_media       | Donor pledges and their private evidence photographs                            |
| verification_results               | Stored AI findings, component scores and status                                 |
| chat_rooms, chat_messages          | Offer-scoped conversations                                                      |
| volunteer_invitations              | Pending and accepted organisation invitations                                   |
| donation_sessions, volunteer_tasks | Fulfilment sessions and per-volunteer task progress                             |

## D. Security Design

- 1) Every protected endpoint requires a Firebase ID token which users must confirm to get access to the system. The system gets user roles from HSDE user records after it checks their Firebase ID tokens.
- 2) The server has control to create and update protected fields which include donor information and request data and organisation details and status updates and timestamp records.
- 3) Evidence photographs which are kept private will only be seen by participants who have confirmed their identity and show they own or are interested in the offer.
- 4) The system does checks, on uploaded files to make sure their format is correct and their size is acceptable and their visual elements are okay and the total number of files is right.

## E. Test Environment

The application was built as an Android APK from React CLI without Expo and it was installed on a real Android phone instead of an emulator. This setup let us use Google Sign-In, the camera and the gallery. It gave us real network behavior and normal screen usage, in real conditions. Development was done on an x64 computer that had at least 8 GB of RAM and an SSD.

## V. AI-ASSISTED EVIDENCE MODULE

The module is designed to help an NGO Admin make their photo review process faster. However, the module itself does not allow the Admin to choose their decisions. Instead, it provides results for each photo along with a performance score. Human reviewers can then use the Accept and Decline buttons to process these results. The way the system works is represented in a diagram that looks like Fig. 5(A).

### A. Processing Pipeline

- 1) Preprocessing. The first step in the system is Preprocessing. This step takes all the data that enters the system. It fixes the EXIF orientation of the photos. It normalizes the photos. Furthermore, it adds contrast to the photos to provide data for the models.
- 2) Object detection. The system uses YOLOv11n to perform object detection on the photos. The system identifies the objects within the photos. Furthermore, for each object, the system provides information regarding the class of the object, the confidence of the detection, and the bounding box of the object. Additionally, the system uses the COCO set of classifications to identify specific objects within the photos, such as bottles, cups, and others. It does not, however, identify the price of the objects.

- 3) Text recognition. The system uses EasyOCR to perform text recognition on the photos. This allows the system to identify all the text within the photos. This includes text such as brand names, amounts, dates, and labels. The system provides a confidence level for each text it identifies. The text is used as evidence in the system due to the challenges that can occur in reading text from photos due to blur, excessive brightness, and packaging of the products.
- 4) Evidence extraction. The system uses the data generated from object detection and text recognition to create a list of findings.
- 5) Category rules. The system then uses the list of findings to check against the category of each finding. Using the annotations that were created for these categories, the system can provide a result of MATCH, UNCERTAIN, or MISMATCH for each finding.
- 6) Scoring. After these results are generated, the system uses them to create a score for each photo. The score is the sum of the scores for each of the four parts of the system. Each part has a weight in the final score.
- 7) Result. The system outputs its score for each photo. The NGO Admin can view these scores prior to making their own decisions.

### B. Evidence Score

Each part of the system has a value between 0 and 1. A result of MATCH for a part is given a value of 1.0. UNCERTAIN is 0.5. MISMATCH is 0.0. The weighted sum is

$$S = 0.30 O + 0.30 C + 0.20 T + 0.20 Q \quad (1)$$

where O is object evidence, C is category evidence, T is OCR evidence and Q is quantity evidence. S is reported as the Evidence Confidence Score (0–100%). The combined weight of object evidence and category evidence reaches 60% while text evidence and quantity evidence complete the remaining 40%.

Interpretation. S stands for the evidence strength level that the automated system creates using this formula. The system shows that administrators from -profit organizations have the power to make final decisions, about donations even though it does not show probabilities of donation authenticity.

### C. Failure Handling

The system gives a managed status (VERIFIED, NEEDS\_REVIEW INSUFFICIENT\_EVIDENCE or FAILED) when object detection or OCR or the whole process fails of creating an error that would stop everything. The reviewer can still access the offer content while they look at pictures and send messages to the donor before making their decision.

## VI. FULFILLMENT WORKFLOW

The main design principle requires that an accepted offer should act as a promise, not as a delivery. HSDE keeps three records: the offer details, the donation session information and the volunteer task records. Figure 5b shows the procedure that starts once the approval process is complete. Fig. 5. shows the evidence module processing workflows together with the post-approval fulfilment stage workflows. Session creation. The system creates a donation session that links to the accepted offer and the organisation when an NGO Admin approves the offer. The operational file for pickup operation is present, as the session.

Assignment. The NGO Admin chooses a volunteer who has registered with the organisation via the dispatch dialog. The system then creates a volunteer task that has status ASSIGNED. Only the NGO Admin is allowed to perform this assignment.

Task progress. The volunteer moves the task forward as the physical work happens:

ASSIGNED -> PICKUP\_STARTED -> PICKED\_UP -> IN\_TRANSIT -> DELIVERED -> COMPLETED

The status follows a fixed path that begins at the stage and moves forward to each next stage while the system records the precise time of each transition (see Fig. 6). The session-level states follow the pattern and include these stages: CREATED, VOLUNTEER\_ASSIGNED PICKUP\_SCHEDULED, IN\_TRANSIT, DELIVERED COMPLETED.

Quantity accounting. The delivered amount of a requirement becomes official when the goods arrive at their destination or the work reaches completion instead of during the acceptance phase. The need screen displays the actual received quantity through a progress bar and percentage which shows 7 out of 10 units. OSRM provides route data which covers both the pickup and delivery segments of transportation operations [12].

## VII. IMPLEMENTED APPLICATION

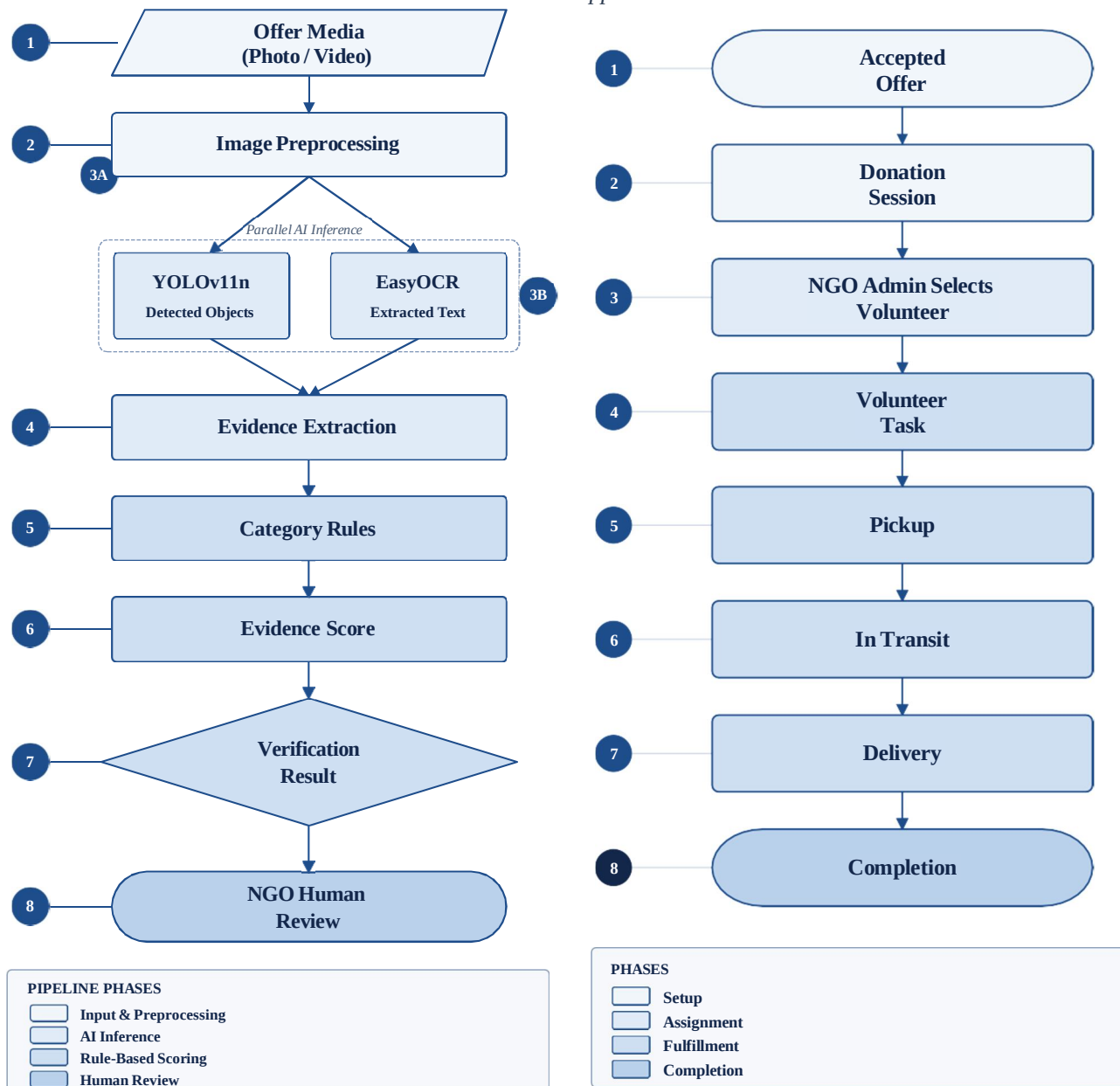
The physical test device shows application screens that show the donation process from beginning to end.

### A. Sign-in and NGO Workspace

People can get into the system using Google login. By typing their email address along with their password. The system lets people who get an invitation enter their code (Fig. 7(A)).

The dashboard gives an overview, for NGO Admins that shows needs and active drives and ongoing logistics work (Fig. 7(B)). The request interface shows all the requirements based on their step while showing how many items are needed compared to how many are filled (Fig. 7(C)).

**AI Verification Pipeline**  
*HSDE: Evidence Verification Workflow*  
**Volunteer Assignment & Delivery Fulfillment Workflow**  
*HSDE Post-Approval Process*



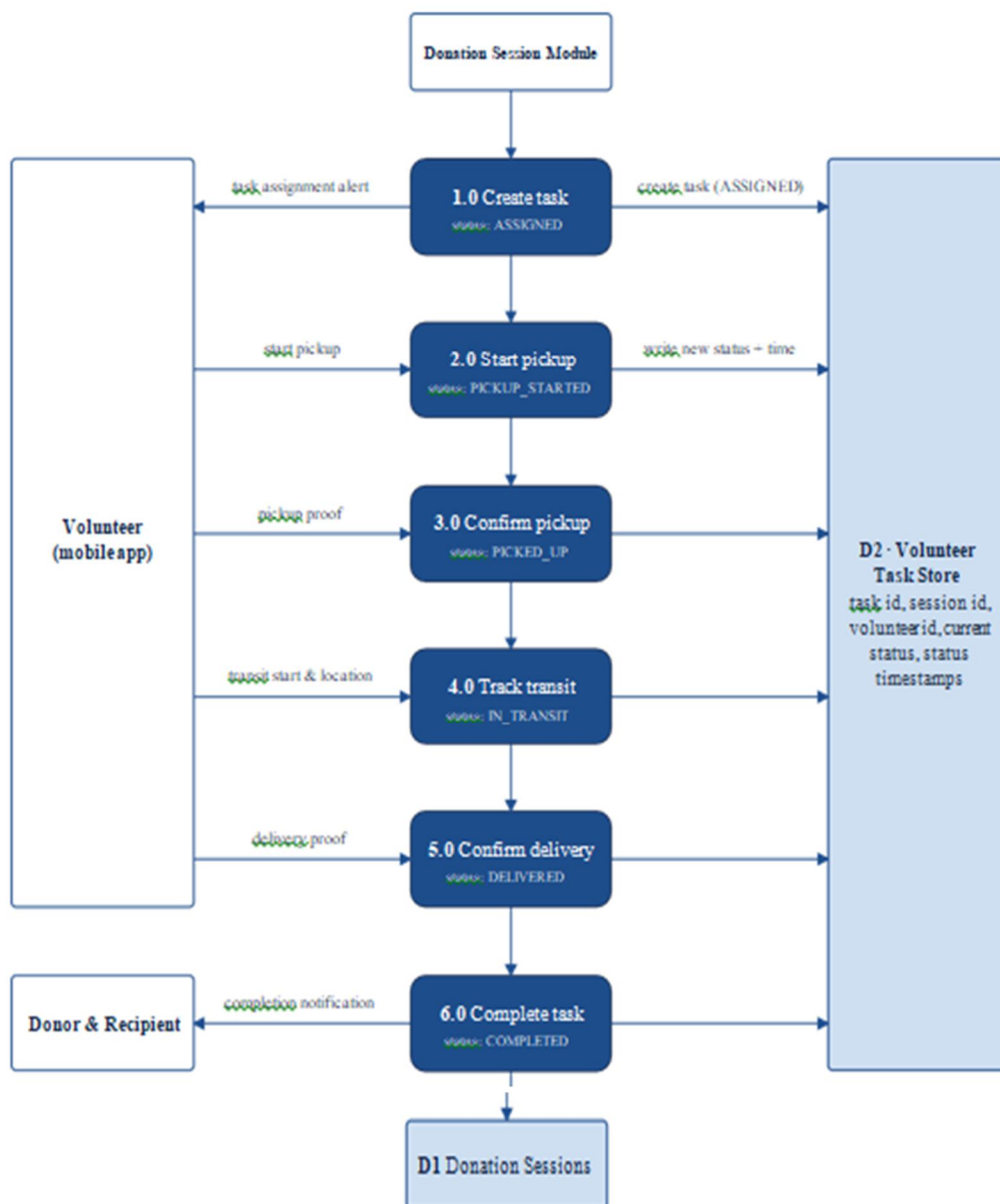
(a) AI evidence verification pipeline

(b) Volunteer assignment and delivery workflow

## B. Donor Journey

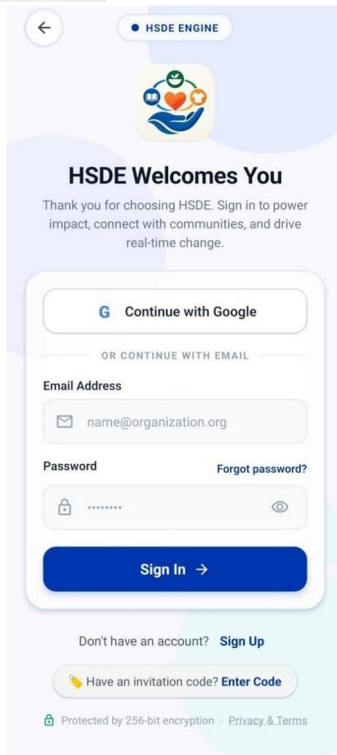
Donors discover needs by using search and category filters (Fig. 8(A)) select a pledge quantity and add optional item notes (Fig. 8(B)) and then follow the offer, on a status timeline that also provides chat and evidence photos (Fig. 8(C)).

### Level 1 DFD: volunteer task lifecycle

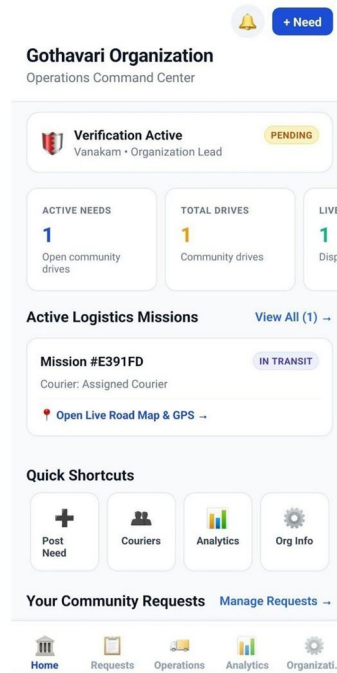


The assigned donation session (donor, volunteer, pickup and drop-off details) flows from the Donor Session Module into 1.0. Arrows between steps hand the task to the next status in the lifecycle. Every step also reads the task in D2 first, so a status can only move forward in the order shown. Step 6.0 marks the donation session completed in D1.

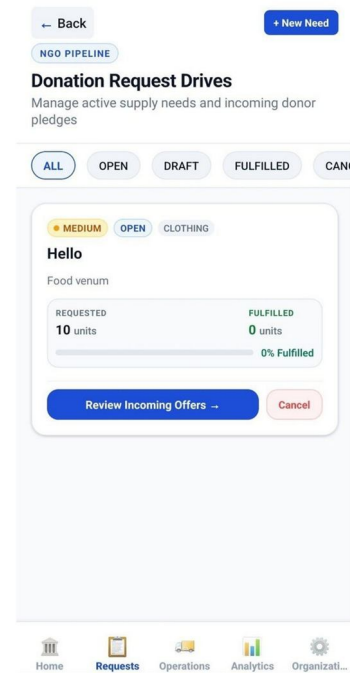
Fig. 6. Level-1 data flow of the volunteer task lifecycle and its interaction with the donation session store.



(a) Sign-in

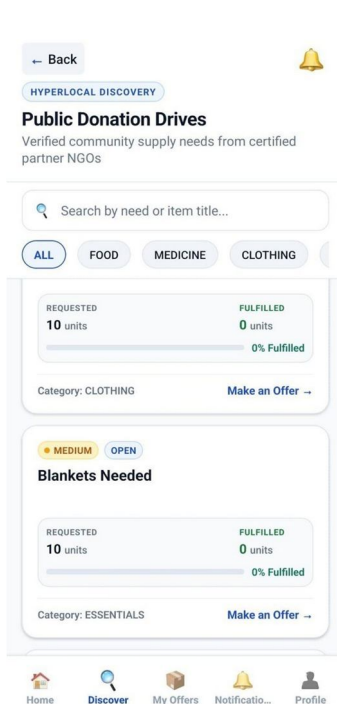


(b) NGO dashboard

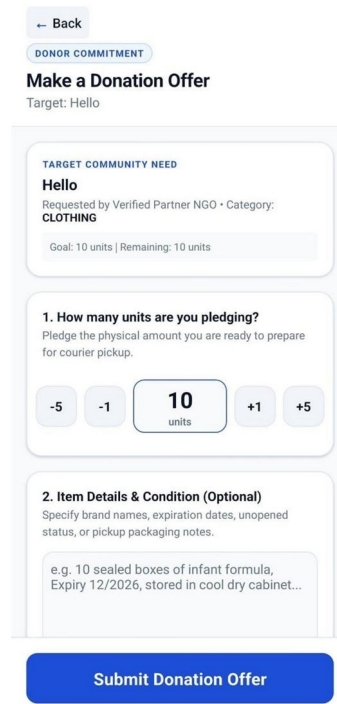


(c) Request drives

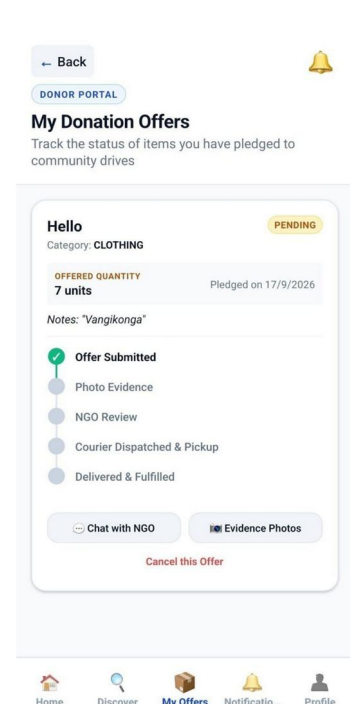
Fig. 7. Authentication and NGO-side screens.



(a) Discover needs



(b) Make an offer

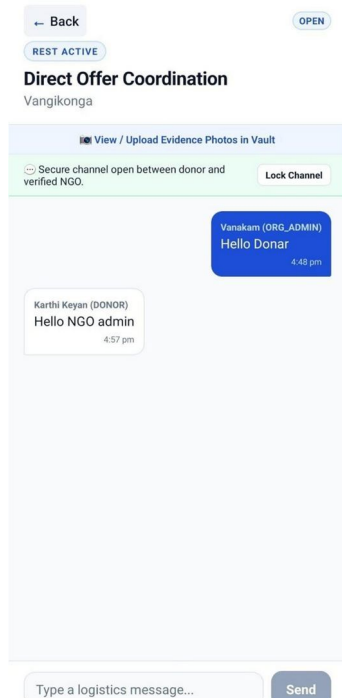


(c) My offers

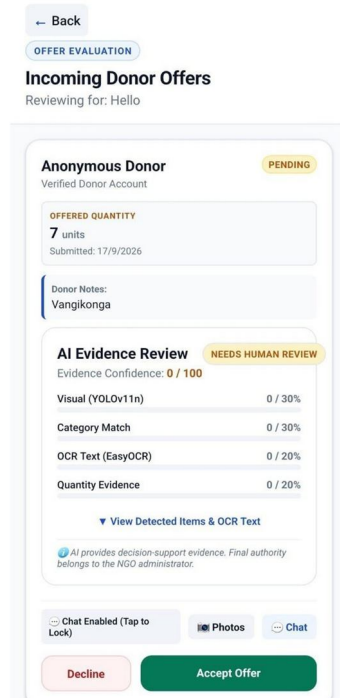
Fig. 8. Donor-side screens for discovery, pledging and tracking.

### C. Chat and Evidence Review

The offer chat system lets NGO Admin users talk to donors while keeping the chat open as shown in Figure 9(a). The offer chat system displays an incoming-offer card that shows donor notes, quantity details, photographs and chat messages. The offer chat system also includes an AI evidence panel that contains four components as shown in Figures 9(b) and 9(c). The Accept and Decline buttons sit below the panel. Are the only way to decide for NGO Admin users.



(a) Offer chat



(b) Evidence panel, initial state

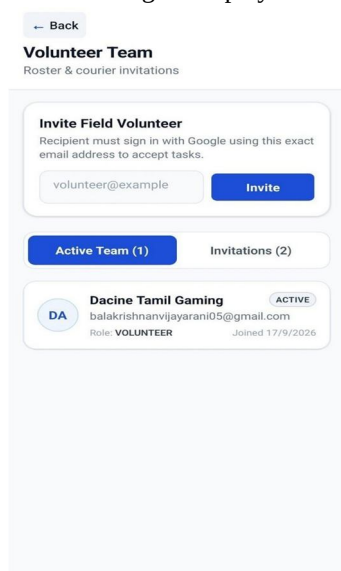


(c) Evidence panel, 68/100

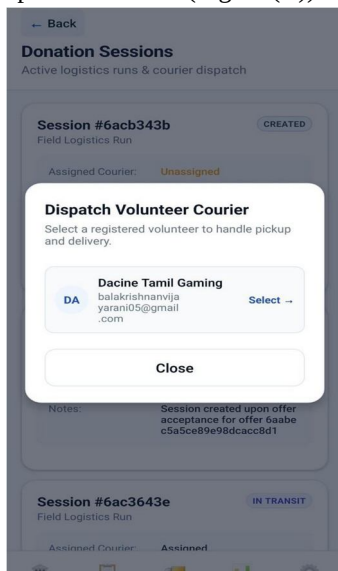
Fig. 9. Offer coordination and AI-assisted evidence review.

### D. Volunteers and Dispatch

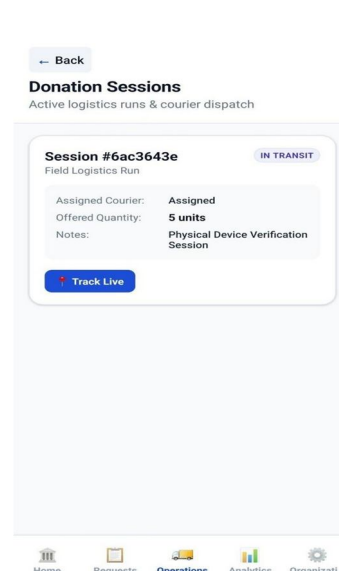
The Volunteer Team page shows two options that let users send invitations and also show current team members (Fig. 10(A)). The session list shows a dispatch dialog that displays all volunteers after an organization accepts an offer (Fig. 10(B)). The system shows sessions through a display that shows their operational state (Fig. 10(C)).



(a) Volunteer team



(b) Dispatch dialog



(c) Donation sessions

Fig. 10. Volunteer invitation, assignment and session tracking.

### E. Volunteer Field Screens

The home screen of volunteers shows the readiness of volunteers, for dispatch and the individual records of volunteers that include all assignments and all current work assignments (Fig. 11(A)). The profile screen shows the role of volunteers and the credentials of volunteers (Fig. 11(B)) and a finished task is marked as delivered and physically verified (Fig. 11(C)).

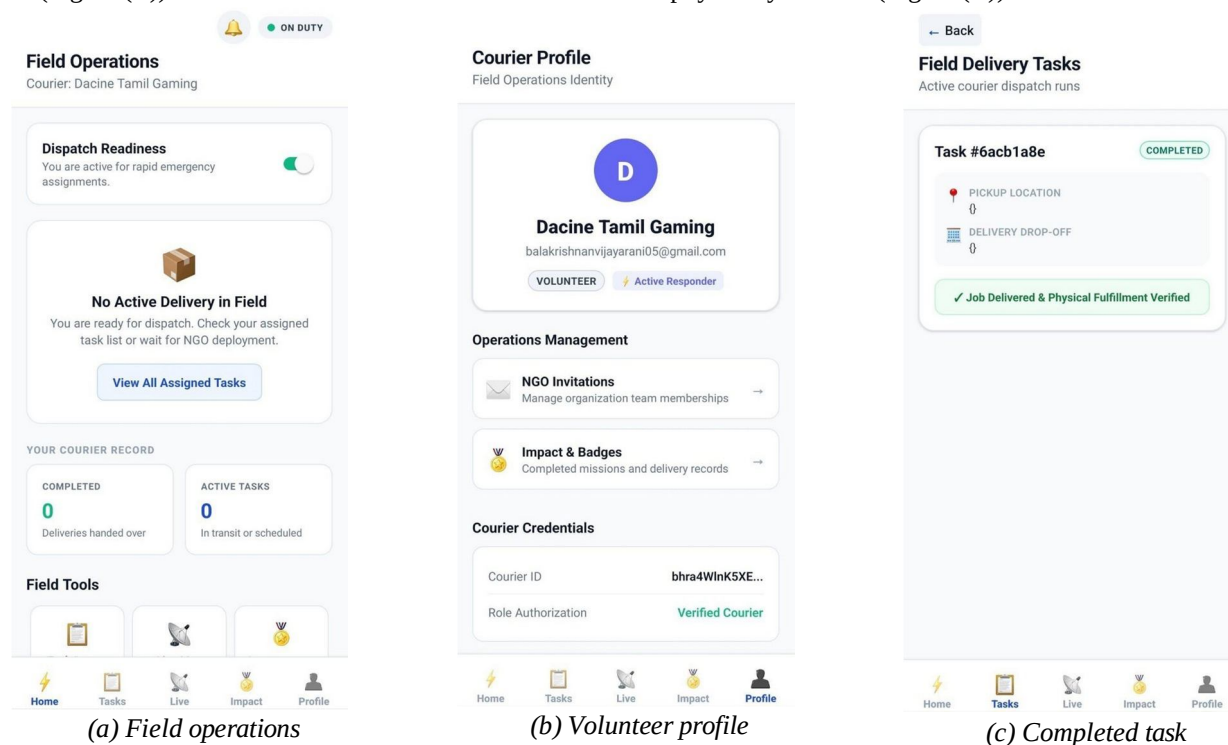


Fig. 11. Volunteer-side screens.

## VIII. RESULTS AND DISCUSSION

The complete process worked from start to finish on an Android phone beginning with signing in and ending with a donation that was sent. The outcomes are not based on numbers or measurements. They are based on how the system worked during tests. These results come from tests that checked if the system functions properly. They do not come from research where real people were asked about their experiences.

### A. Workflow Results

Table V. Observed Results By Module

| Module         | Observed result                                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication | Google Sign-In created a Firebase ID token; the backend checked it and connected the UID to the HSDE user and the role that user had                           |
| Needs          | NGO Admin set up needs; only the needs that were marked as OPEN showed up in the donor's discovery list and those needs could be filtered by category          |
| Offers         | Donors made offers (like 7 units for a request of 10); the NGO could see those offers along with notes from the donor and any proof they provided              |
| Chat           | An offer-specific channel could be locked or opened by the NGO Admin                                                                                           |
| Volunteers     | A volunteer who was invited showed up in the team after that person signed in using the account they were invited with.                                        |
| Fulfilment     | When an offer was accepted, a session was created; when an assignment was made a task was created; when the task was done the need was marked as 70% complete. |

### B. AI Evidence Results

Two test offers reveal how the module behaves. In the test YOLOv11n was able to identify only basic items, such as a bottle, a cup, a dining table, chairs and an umbrella. EasyOCR on the hand found the word "PRINGLES" and also an incorrect version of that word. The system marked the result as UNCERTAIN using the category rule giving a 0.44 score. That score led to an INSUFFICIENT EVIDENCE classification. In the test a photo of clothing displayed a clothing bundle, with 88% confidence and also showed a readable lot label, which I saw in Table VI and Fig. 9(C).

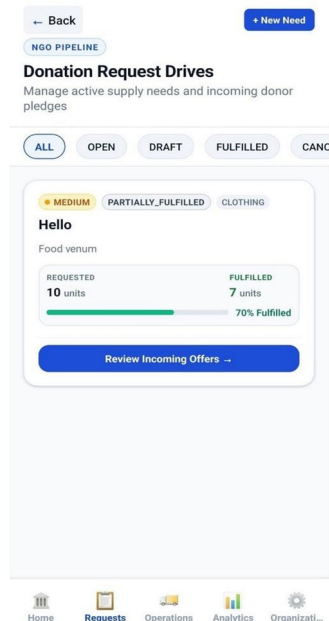
Table VI. Evidence Score Breakdown For Test 2

| Component (weight)      | Points (test 2) | Maximum |
|-------------------------|-----------------|---------|
| Visual, YOLOv11n (30%)  | 24              | 30      |
| Category match (30%)    | 22              | 30      |
| OCR text, EasyOCR (20%) | 12              | 20      |
| Quantity evidence (20%) | 10              | 20      |
| Evidence Confidence     | 68              | 100     |

Both results were marked as needing review. The first case shows a limitation: a COCO-trained detector sees generic items so a snack package photographed on a table yields weak evidence. The misread "PRINGTES" also shows that OCR can be wrong which is why the text is a clue.

### C. Fulfillment Result

Both results were marked as needing review. The initial case demonstrates a restriction because the COCO-trained detector identifies common objects which results in insufficient proof when it captures a snack package on a table. The text contains an OCR error because the system read "PRINGTES" of the correct "PRINGLES" which proves OCR technology makes mistakes during text recognition.



#### What the result shows

- The progress bar shows how far delivery has gone not how many offers have been accepted.
- Three units are still available for donation because the need has not yet been met.
- The NGO can still open offers, from this screen to look at more pledges.

Fig. 12. Request after delivery: 7 of 10 units, 70% fulfilled.

### D. Discussion

- Human control The decision point maintains control because the system generates scores which reviewers can see but these scores do not have the power to make acceptance or rejection decisions.
- Server-side checks The server-side validation system prevents donors from modifying fields which NGOs control. It blocks volunteers from taking control of accounts which do not belong to them.

- Separate records The NGO keeps records for offer, session and task which help them track their promised and accepted and delivered services.
- Limits. The detector system operates as a tool which produces different OCR results depending on the illumination conditions and product wrapping. The evaluation process focused on testing the systems performance of conducting numerical analysis.

## IX. TESTING

Testing covered the backend, the mobile build and the combined workflow on a physical Android device (Table VII).

Table VII. Test Coverage

| test type            | what was checked                                                                                                                                                                                      |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unit                 | I perform authentication validation handle the request process the offer, validate role and quantity and manage AI outcomes.                                                                          |
| Functional           | Google Sign-In creates or retrieves the HSDE user; the NGO creates a need; the donor submits an offer; the NGO accepts it. A session appears; the NGO assigns a volunteer and a task appears.         |
| Integration          | App ↔ FastAPI, FastAPI ↔ MongoDB, Firebase ↔ backend token verification offer media ↔ AI service ↔ stored result.                                                                                     |
| AI pipeline          | I see multiple, unreadable and unrelated images; I read visible text; I note category and quantity cues; I experience OCR failure; I experience detector failure; I experience full pipeline failure. |
| Performance          | I check API response, database operations, image processing, AI processing time (measured from runs) and screen response.                                                                             |
| Security             | Protected endpoints reject roles. Donors cannot edit NGO fields. Volunteers cannot self-. Join organisations. Organisations' media is blocked.                                                        |
| Usability and device | Navigation is clear. Readability is good. Loading is smooth. Error messages are helpful. Camera and gallery work well. Network behaviour, on the phone is reliable.                                   |

The design needs a check to find out if an AI component failure will make the system stay organized instead of causing a system failure. If the system stays organized reviewers can keep working when the analysis does not work avoiding an AI component failure that would otherwise create a system failure.

## X. CONCLUSION AND FUTURE SCOPE

The HSDE system allows a mobile platform to track donations from published needs until donors confirm that the items have been delivered. The system makes sure that important decisions stay with humans. The system limits users to roles and only lets chat interactions focus on one specific offer. Only approved volunteers can join. Each group works within rules that control what they can do. The evidence module helps speed up photo analysis because it shows the process that leads to the rating it gives. The offer–session–task split system helps organizations keep records of the services they promise and the services they actually provide. The system has three restrictions. These include an object detection system that stays inconsistent OCR results and an assessment process that uses functional methods instead of statistical ones.

Future work includes:

- Showing how each part of the evidence contributes, with its limits to NGOs and donors.
- More notifications using Firebase Cloud Messaging for offers, invitations, assignments and deliveries.
- Planning routes for pickups and deliveries using OSRM.
- An analytics tool to track totals how much is. How it is spread across categories and areas.
- A detector trained on images of donations and OCR that works better in light for rotated labels and for text in many languages.
- Scaling to organizations with strong data separation, between them.

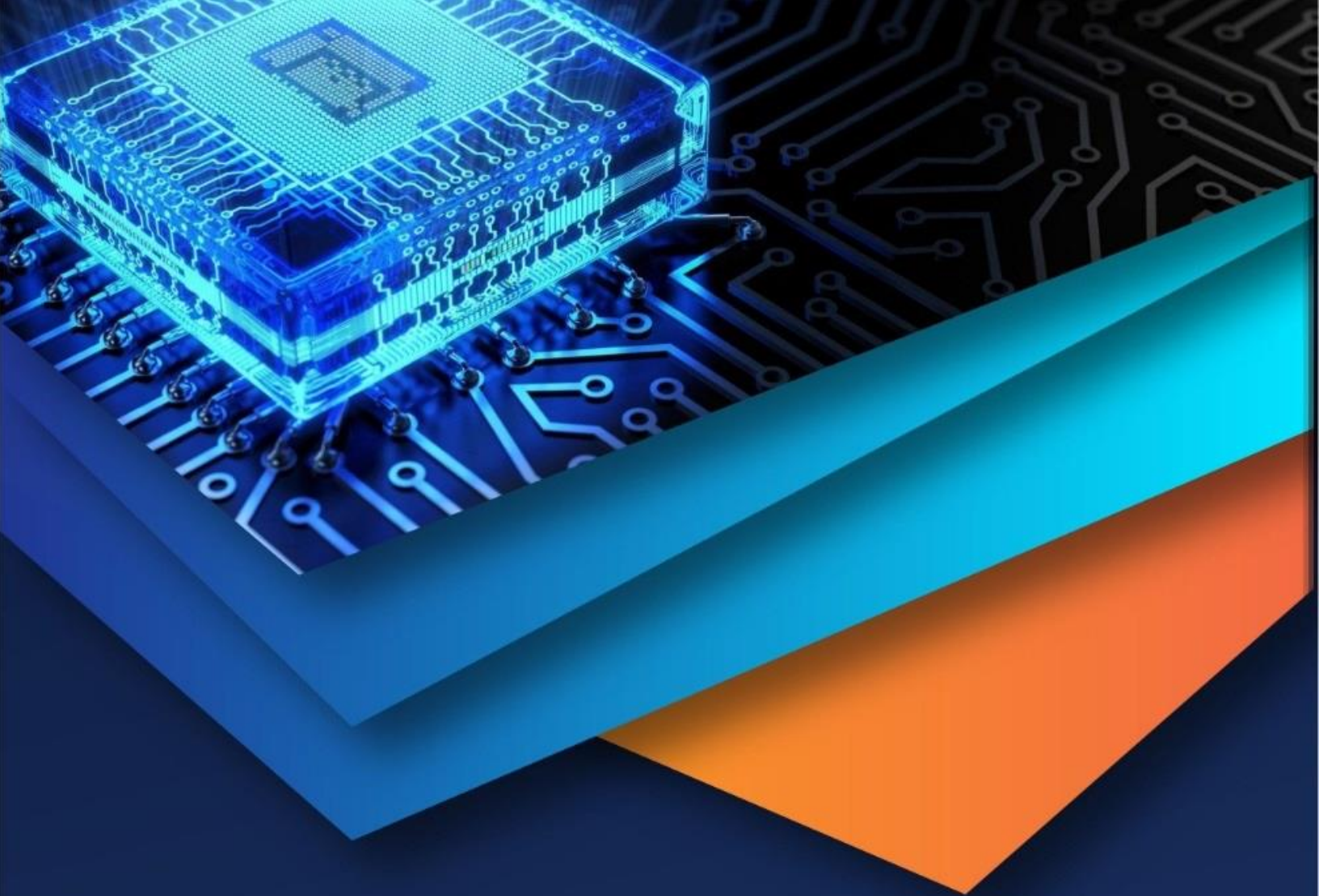
## XI. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering at K.L.N. College of Engineering because they gave support for this work.



## REFERENCES

- [1] Empowering Remote Donations: A Comprehensive Mobile Application Suite for Charitable Contributions, IEEE, 2024.
- [2] AI-Enhanced Donation Management System, International Journal for Research in Applied Science and Engineering Technology (IJRASET), 2025.
- [3] D. Chhibber, A. Tripathi, and S. Ray, "Do VIR: Virtualizing Food Donation Distribution through Mobile Application and Cloud-Based Supply Chain Management," in Proc. IEEE Int. Conf. Consumer Electronics (ICCE), 2021, doi: 10.1109/ICCE50685.2021.9427641.
- [4] "KindNest – An Essentials Connect," in Proc. Int. Conf. Cognitive, Green and Ubiquitous Computing (IC-CGU), 2025, doi: 10.1109/IC-CGU67042.2025.11338008.
- [5] K. Poongodi, M. Sathish Kumar, S. Srinithi, D. Devadharshini, et al., "Bridging the Gap Between Donors and Recipients: A Digital Application to Food Donation," in Proc. 5th Int. Conf. Communication, Computing and Electronics Systems (ICCCES), 2026, doi: 10.1109/ICCCES62661.2026.11436954.
- [6] "FoodBridge: Connecting Surplus to Need," in Proc. Int. Conf. Emerging Technologies in Engineering Applications (ICETEA), 2025, doi: 10.1109/ICETEA64585.2025.11099979.
- [7] "FoodSavior: Distributing Surplus Food to NGOs or Manure Producers Using Internet of Things and Machine Learning," in Proc. Int. Conf. Sustainable Communication Networks and Application (ICSCNA), 2024, doi: 10.1109/ICSCNA63714.2024.10863911.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [9] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A Survey of Deep Learning-Based Object Detection," IEEE Access, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.
- [10] FastAPI Documentation, "FastAPI – Modern Web Framework for Building APIs with Python."
- [11] MongoDB Documentation, "Geospatial Queries and 2dsphere Indexes."
- [12] OSRM Project, "Open Source Routing Machine – Route Planning and Navigation Services," OSRM Documentation.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)