International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Paasmitra: A Location-Based Hyperlocal Platform, for Paid and Free Community Assistance

Ms. Lakshmi R
Student, Department of BCA
St. Francis College
Bengaluru
lakshmiacharya2006@gmail.com

Mr. Gajanan Ullas Revankar
Assistant Professor, Department of BCA
St. Francis College
Bengaluru
gajaurevankar26@gmail.com

***Abstract*— Many people require assistance with minor daily tasks that in principle do not involve any cost, yet there is generally no easy method of finding a local helper. This paper presents PaasMitra, a mobile application which links a Seeker (the person who needs help) with a nearby Mitra (the helper) by employing a task model that combines both free community help and paid micro-tasks, together with a community feed allowing users to browse local individuals. We developed the prototype using Flutter, with Firebase Authentication and Cloud Firestore for the backend. Features include logging into the app, creating or discovering tasks, filtering between free and paid tasks, locating nearby users, engaging in the community, and using in-app messaging. We outline the architecture, the data model, and the proximity-based discovery, place our work in the context of spatial crowdsourcing, mutual aid, sharing-economy trust, and task crowdsourcing platforms, and give the results of the functional testing of the prototype [and a usability pilot involving N participants: include the actual results].**

***Keywords*—community assistance, Firebase, Flutter, hyperlocal services, location-based services, mutual aid, sharing economy, spatial crowdsourcing**

I. INTRODUCTION

Everyday life frequently involves situations in which people have jobs that are too small to warrant hiring a professional but at the same time too inconvenient to carry out by themselves, for example, moving boxes up a flight of stairs, picking up prescriptions for an elderly neighbour, waiting at a local shop for a delivery or finding someone to study with. People typically turn to their family, friends, social networks or else pay for a service that provides more assistance than is actually needed for the task in question. When none of these alternatives are possible, it can be difficult to discover someone who is willing, available, and close by to offer help. Existing digital task platforms do somewhat address this issue. Home-service and task marketplaces provide paid services, but this covers only one part of the picture: the gig-economy literature examines the implications of such platforms for workers and the local economy [1]– [5], including in emerging economies such as India [6][7]. Neighbourhood and mutual-aid platforms enable unpaid help, and recent studies have looked at how digital tools function within these volunteer networks [8]– [13]. In the literature, we considered these different approaches, which were generally separate: if a person wished to post a task to walk a dog and then find someone nearby to help with grocery shopping, they would have to use two platforms.

The paper presents PaasMitra (derived from the Hindi words Paas which means 'near' and Mitra which means 'friend'), an app that allows a Seeker to post a task and enables a nearby Mitra to volunteer or agree to assist. PaasMitra includes both unpaid and paid tasks, where users will also be able to look at a community feed in order to discover people and tasks. The following research questions were examined:

- RQ1: What is the way of combining person-to-person tasks which involve paid assistance in terms of task representation and workflows?
- RQ2: In what way can people's locations be utilized to identify local opportunities both for helping others and for receiving help?
- RQ3: What trust, privacy and safety features should be designed when people are helping one

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

What we have provided consists of (1) a task model which includes both paid assistance, (2) a method for identifying possible tasks and helpers based on proximity, (3) a community feed to aid in the discovery of both people and tasks, and (4) a cross-platform implementation using Flutter and Firebase, together with an examination of its limitations. The rest of the paper is structured as follows: Section II provides a review of related work; Section III and Section IV deal with the design and implementation; Section V outlines the features; Section VI presents the evaluation of the platform; Section VII discusses the implications of our approach; and Section VIII concludes the paper.

II. RELATED WORKS

A. Spatial crowdsourcing and task assignment

Research into spatial crowdsourcing looks at the way location-dependent tasks can be distributed to people who are able to carry them out. Earlier studies have examined task allocation among different types of tasks [14], stable assignment between several agents [15], preference-aware task recommendation [16], [17], and preference matching for a group [18]. Similar work has also taken the volunteers' requirements into account together with the objectives of task completion [19]. Most of these studies concentrate on algorithmic task assignment and recommendation. PaasMitra, on the other hand, uses a more simplified method appropriate for a prototype developed by students, allowing tasks to be published and for users nearby to independently decide whether or not to offer help.

B.. Location privacy

Since assignment requires location, concerns about privacy arise frequently. Recent approaches have provided protection for either the worker's or the task's location both when assignment is taking place [20], [21] and when the task is pushed [22]. We are not including such protection in the current prototype and regard location privacy as a limitation and as a focus for future work (Section VII).

C. Gig and task platforms

Research into platform-mediated work has looked at the socio-technical aspects of digital labour [1], the effect of platforms on workers [5], job quality [3], and the connection between online gig platforms and local employment [2]. Investigations of gig work in India have also examined its growth and the challenges linked to it [6], [7], and some studies have looked at the relationship between platform-based work and personal life [4]. Although such platforms are concerned with paid work, PaasMitra expands this approach to include both paid microtasks and free community assistance on the same platform.

D. Mutual aid and community platforms

Studies of mutual-aid platforms have examined how digital technologies can be used to help communities in organising and coordinating assistance [9]. Other research has looked at the sociotechnical systems employed by community networks during the COVID-19 period [12], at grassroots mutual-aid practices [11], and at the capabilities and limitations of online mutual-aid platforms [13]. Research has also suggested community-based mutual-aid systems for groups such as the elderly [10] and has investigated mutual aid within internet-based communities [8]. PaasMitra is based on these approaches by combining community assistance with paid tasks and proximity-based discovery.

Aspect	Home-service booking	Task marketplace	Neighbourhood network	PaasMitra
Primary model	Booking of professionals	Paid tasks for independent workers	Social network of neighbours	Person-to-person local assistance
Who helps	Vetted professionals	Independent taskers	Neighbours (via posts)	Nearby users (Mitras)
Free help	No	No	Informal	Yes
Paid tasks	Yes	Yes	Not core	Yes
Proximity discovery	Service area	Yes	Neighbourhood	Yes (device location)
Community feed	No	No	Yes	Yes
Payment	In platform	In platform	Not applicable	Not integrated in prototype [confirm]

TABLE I.: COMPARISON OF PLATFORM CATEGORIES

E. Trust, safety and user acceptance

When interacting with strangers over the internet, trust is a major factor. Previous studies have looked at trust in peer-to-peer and sharing platforms [23], [24], the role of reputation and feedback systems [25], digital trust [26], [27], and how trust has changed after the COVID-19 pandemic [28]. Research into the acceptance of mobile applications has employed extended technology-acceptance models [29], and studies concerning skill-sharing

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

platforms have examined the factors that affect continued use [30]. This body of work has shown that trust, privacy/safety, and user acceptance are all important when designing a system for connecting people in the same locality.

F. Mobile development and the backend services

Studies have compared cross-platform frameworks such as Flutter with respect to development effort and performance [31]–[34], highlighting Flutter's contribution to speeding up development [35]. As for the backend, research has involved performance monitoring of Firebase's real-time database [36], an analysis of Firestore optimisation and the use of Firebase Cloud Messaging . These factors have underpinned our technology choices.

G. Positioning of Passmitra

Table I presents a comparison between PaasMitra and three typical categories of platform. This comparison is grounded in the way these categories have positioned themselves to the public and does not imply that PaasMitra is the first system of its kind; rather, its purpose is to indicate which combination of features we aim to include. of properties we target.

III. System Design and Architecture

A. Actors and requirements

PaasMitra has two roles. A Seeker posts a task and a Mitra offers help; one account can play both roles. The functional requirements are: FR1 register and sign in; FR2 create a task with a category, description, paid/free type and reward where applicable; FR3 discover tasks near the user; FR4 filter tasks by paid/free type and category; FR5 discover nearby users; FR6 take part in a community feed; FR7 communicate through chat; and FR8 manage one's own and completed tasks. The non-functional requirements are ease of use, real-time updates, low operating cost and careful handling of location data.

B. Architecture

The system is organised in four layers (Fig. 1). The Flutter client presents the Home, Explore, Tasks, Communities, Nearby Mitras and Profile modules. An application layer holds state and routing. Device services provide location, and the Firebase backend provides authentication and storage. Using a managed backend removed the need to build and host a custom server, which suited a small team.

C. Data model

Think of Cloud Firestore is a database that stores data in documents that are organized into collections. Table II below shows an overview of the logical data areas used in the PaasMitra system. User-related information is stored in the users collection and task information in the tasks collection. The one-to-one communication data is also stored in the chats collection with a subcollection of messages.

Collection	Main fields	Purpose
Users	User ID, name, authentication details, profile information [location fields: confirm]	Accounts, profiles, nearby-user discovery
Tasks	Title, description, category, paid/free type, reward amount (paid tasks), creator, status, creation time, location	Task creation, discovery and management
Chats / messages	Task ID, participant IDs, message text, time [confirm]	Real-time communicatio n between task participants

TABLE II.: Firestore Data Model

D. Task lifecycle

A task is created when a Seeker creates a request. It is then saved in the tasks collection in the Cloud Firestore database, marked as open. An open task can be found and responded to by a nearby Mitra. The Mitra can propose or accept the task, chat with the Seeker, do the task, and mark it as completed.

task, chat with the Seeker, do the task, and mark it as completed. The task status is changed during the process according to the application logic.

E. Free and paid tasks

The structure is the same for both types, with the only difference being the type of flag and the reward amount which applies solely to paid tasks. Free tasks include things like studying together, offering cooking help, walking together or picking up medicine, while paid tasks involve short jobs such as grocery pickup and delivery and carrying boxes.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

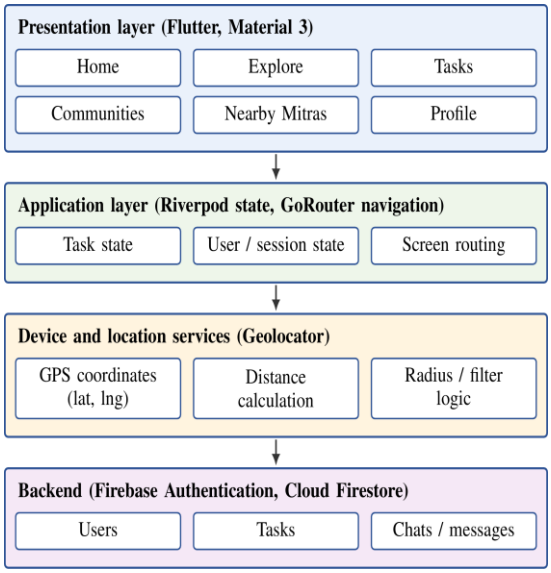


Fig. 1. Layered architecture of PaasMitra.

IV. Implementation

A. Technology stack

Table III contains the stack; Flutter enables a single Dart codebase to be used for multiple platforms, a point that is looked at in the cross-platform comparisons [31]–[35]. Firebase provides authentication and a real-time document database, both of which have been investigated for their performance and optimisation [36], [37].

B. Authentication

Users register and log in using Firebase Authentication, and the user ID thus obtained links the account to the corresponding profile, tasks and messages. Navigation takes place in the following order: from the initial page to the login screen and then to the Home screen. [The methods of sign-in confirmed: email/password, phone, Google.]

C. Proximity-based discovery

The Geolocator is used to obtain the geographical coordinates (latitude and longitude) of the user. Whenever a task is generated, geographical information about the location of that task is saved along with the task itself. Distance between the geographical locations of the user and a particular task can then be calculated to facilitate proximity-based task discovery. Tasks are extracted from the task database, and then the required filters on the tasks are applied.

Algorithm 1

1: Obtain the user's current position (*lat*, *long*) using Geolocator.

2: Read the available tasks from the **Tasks** collection in Cloud Firestore.

3: For each task, obtain the latitude and longitude stored with the task.

4: Calculate the geographical distance between the user's location and the task location.

5: Identify tasks that are geographically close to the user.

6: Apply the selected task filters, including the available task criteria.

7: Display the matching tasks to the user.

Algorithm 1 shows the general process of location-based task discovery. Distance between geographical coordinates is calculated, whereas the value of proximity distance can be adjusted as per the needs of the application.

D. Real-time chat

To start a chat for a given task, a chat screen is created and the other user's ID and name as well as the task title are entered using the task ID. The messages are stored in and retrieved from Firestore, and the listeners associated with Firestore update both screens without it being necessary to manually refresh them.

Component	Role	Rationale
Flutter / Dart	Client application	Single codebase for several platforms [31]–[35]
Material 3	User interface	Consistent, familiar components
Riverpod	State management	Separates application state from screens
GoRouter	Navigation	Declarative routing between modules
Firebase Authentication	Sign-in	Managed identity without a custom credential store
Cloud Firestore	Database and chat	Document model with real-time updates [36], [37]
Geolocator	Device location	Supplies latitude and longitude for proximity logic

TABLE III.: Technology Stack and Rationale

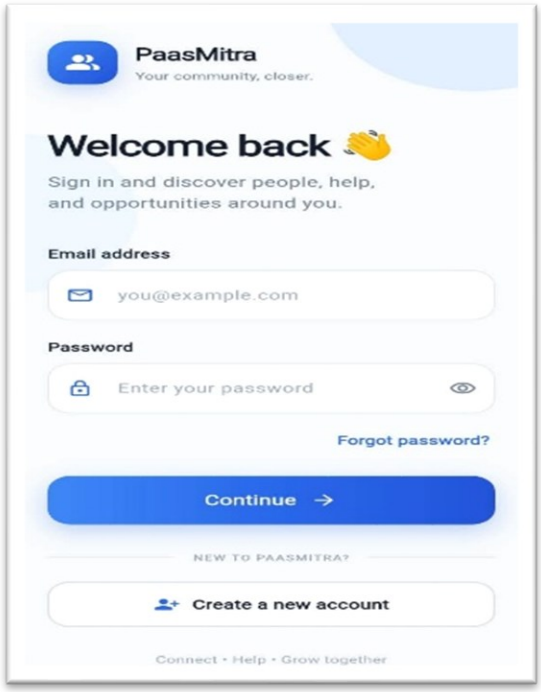
E. Security Configuration

The access to the data of user, task, community, and messages is managed using the Firebase Authentication and Cloud Firestore security rules. The authenticated users will have access to the protected collections. The creation and updating of the user profile are limited to the creator himself whereas the task data reading and task operations are accessible to the authenticated users. The access to communities will be limited to the authenticated

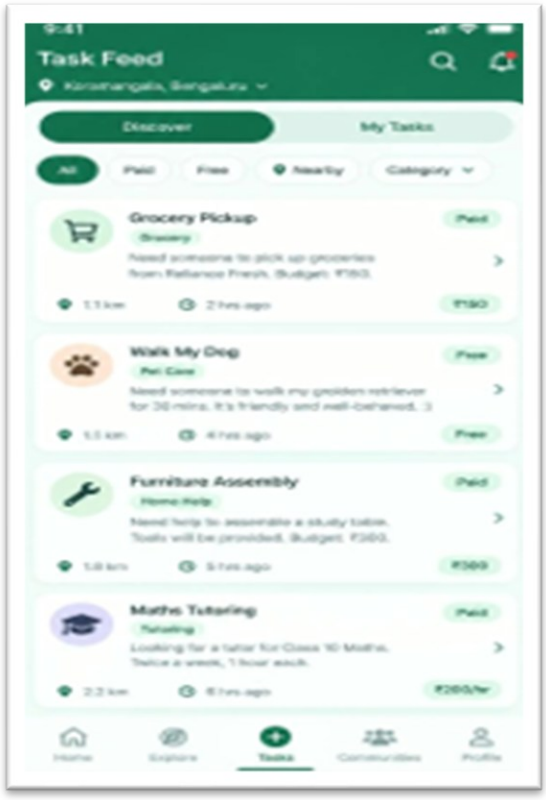
(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

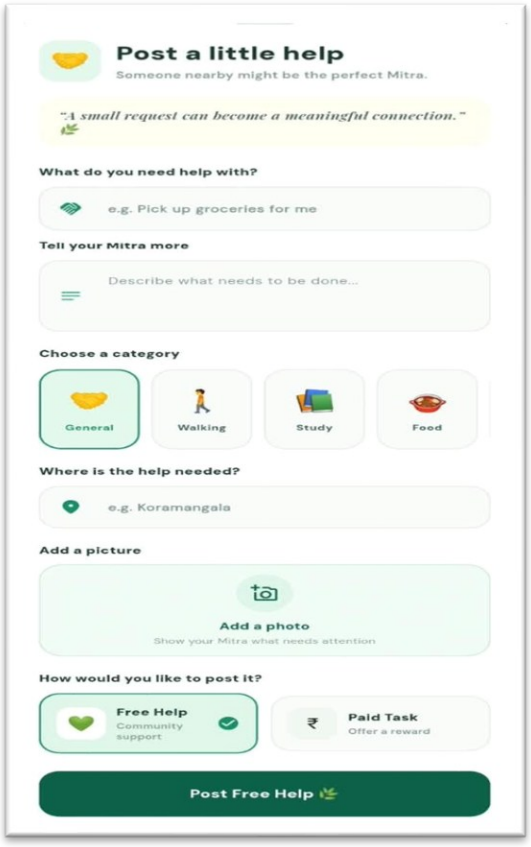
users whereas the members of the community can manage their membership records.



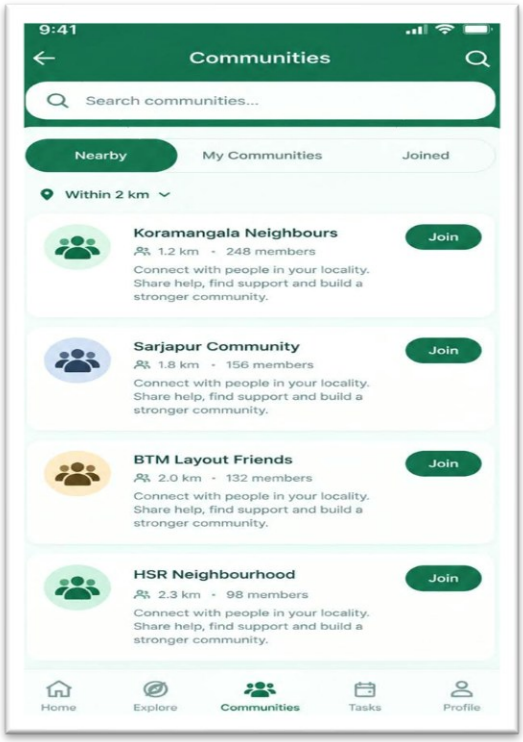
(a) Login



(b)Task feed



(c) Task creation



(d) Communities

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

V. FEATURES AND WORKFLOW

Upon successful login, the user is directed to the

Upon successful login, the user is directed to the Home screen where he/she can access assistance services, location information and task discovery. Explore module allows users to browse available tasks, categories and local assistance opportunities. You can use the Tasks module to create and find tasks, view My Tasks and Completed Tasks, and filter tasks by using the options that are available. Tasks are categorized as free or paid and grouped into categories such as Grocery, Pet Care, Delivery, Repairs, Tutoring, Elder Care, Home Help, Tech Help and Other.

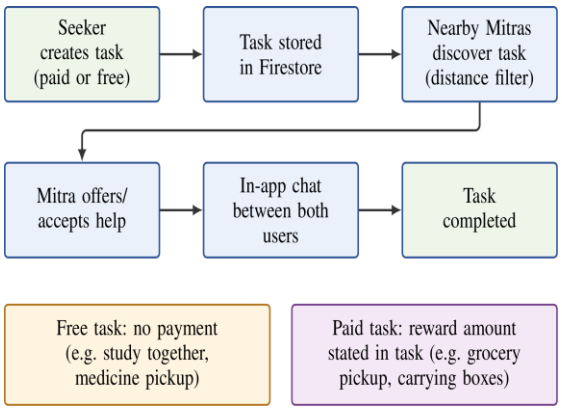


Fig. 2. General task workflow and the free/paid distinction.

PaasMitra uses device location services to obtain geographical coordinates such as latitude and longitude. You can add location information to new tasks you create, to aid discovery based on proximity. Geographical distance calculation is useful to find tasks which are close to the user. With the Communities module, users can discover and interact with people in their vicinity, taking the platform beyond simple task management. Nearby helps find nearby people that can help. Mitras The Dashboard shows activities related to the task and the Profile has account information for the user.Two scenarios are used to demonstrate the workflow.

A student posts a Study Together task in a free-assistance situation. Another student sees the task, gets in touch with the creator via chat and agrees on a suitable time. In the paid-task scenario, a resident creates a task called Carry Boxes and specifies a reward.

A Mitra from the locality checks and approves the task. Both users chat to coordinate the activity. The status can be updated after the task is completed. The two situations share an identical process which starts with task generation and proceeds through task identification and message exchange and support provision and final task tracking. PaasMitra offers a single platform which enables users to access free community assistance paid microtask opportunities for their neighbourhood help needs.

ID	Module	Test case	Expected	Result
T1	Authentication	Register with valid details	Account created	[Pass/Fail]
T2	Authentication	Sign in with correct and incorrect credentials	Access only when correct	[Pass/Fail]
T3	Tasks	Create a free task	Saved and listed	[Pass/Fail]
T4	Tasks	Create a paid task with a reward	Saved with paid type and amount	[Pass/Fail]
T5	Discovery	Open nearby tasks with location enabled	Tasks within radius shown	[Pass/Fail]
T6	Discovery	Open nearby tasks with location denied	Handled without failure	[Pass/Fail]
T7	Filtering	Apply paid/free and category filters	List matches filter	[Pass/Fail]
T8	Nearby Mitras	Open the nearby-users view	Nearby users listed	[Pass/Fail]
T9	Communities	Open and interact with the community feed	Feed loads; interaction works	[Pass/Fail]
T10	Chat	Exchange messages between two accounts	Messages appear on both sides	[Pass/Fail]
T11	Task management	Check My Tasks and Completed Tasks	Lists reflect task status	[Pass/Fail]
T12	Navigation	Switch between all modules	No errors	[Pass/Fail]

TABLE IV.: FUNCTIONAL TEST CASES

Table IV presents twelve functional test cases which assess the system, compares its expected results against the actual outcomes to verify the application remains reliable and functional.

PaasMitra uses Firebase Authentication together with Cloud Firestore security rules to protect their application data from unauthorized access. The system allows only verified users to access both user profile information and task record data. Users have the ability to establish and modify their personal user profiles but only verified users can create new tasks and modify existing tasks and delete them. Users who verify their identity can access community data through read and create and update and delete operations on community records. Each member of the community holds responsibility for maintaining

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

their personal membership information. Users who verify their identity can access community messages through reading and message creation functions but only the original sender of a message can perform message updates and deletions.

Users who verify their identity can access chat records through reading and writing functions when they use one-to-one chat features. Users can access messages in chats through reading and creation functions but only the original sender of a message has the authority to modify or delete it. The application protects its primary data storage through authentication-based access control which operates under these established rules.

VI. Evaluation and Results

A. Functional Testing

Functional testing is used to test the components of the application, which include authentication, task creation, location-based discovery, task filtering, community engagement, chat, task management and navigation. Table IV contains the test cases. The expected results. The test case is verified through the comparison of the application behaviour with the expected behaviour. The result of each test is specified as Pass or Fail after testing the functionality on the selected device or emulator.

B. Usability Evaluation

The usability evaluation in the work can be carried out via asking users to perform various common tasks, e.g., sign up, create a task, search for help, send messages, and view the task history. Participants' feedback can reveal issues related to navigation, task creation and user experience. The standard usability questionnaire can be used to measure user satisfaction. There are no scores and measurements from the participants yet, since this evaluation hasn't been performed yet.

C. Performance Evaluation

Performance evaluation of the application can be performed in work by checking the speed of the tasks searching with large amount of data and various network conditions. Further testing can include location-based discovery, responsiveness, and tasks handling. Such tests would help to determine the possible application bottlenecks and areas for improvement.

VIII. DISCUSSION

On RQ1 PaasMitra uses task workflow to support both free and paid assistance. A task can be classified by its type and for paid tasks reward details are included. This design enables users to post and discover kinds of neighbourhood help within one app.

On RQ2 device location coordinates help find tasks which are nearby to a user. How well this works depends on the way of distance calculation, how much location data is available. How many tasks are stored. More testing is needed to evaluate how fast tasks are found and how accurate the matches are when volumes of data vary. On RQ3, trust and safety remain important when people

who do not know each other arrange help. PaasMitra uses logged-in accounts and user profiles. Features like identity verification, user ratings, reporting and blocking need further development. Until these safety features exist users must be careful talk before accepting tasks and consider meeting in public places when possible.

Privacy: Location data must be handled carefully as it can expose where users move and what they do. Future versions of PaasMitra should limit location exposure and show approximate distances instead of exact coordinates whenever possible.

Payments: PaasMitra allows users to label tasks as free or paid but PaasMitra does not have an inside payment system. As such payment deals for paid tasks happen outside the app, limiting PaasMitra's ability to handle payment disputes. Addition of in-app payment system could be considered for future.

Limitations: PaasMitra is still a prototype, and PaasMitra performance and ease of use need testing in real life. More tasks and users must be tried to see if PaasMitra can grow.

The proximity discovery method also needs to be evaluated for how it finds tasks and how relevant the results are. Also how people interact and how well the community accepts PaasMitra must be studied with neighbourhood participants.

Societal Value: PaasMitra aims to boost cooperation by putting free help and paid microtasks in one place. PaasMitra could help users gain support, for things while giving them opportunities to earn money from small tasks.

Realisation of these benefits depend on how easy Paas Mitra's to use, how PaasMitra handles location data responsibly, how safe users are and how fair the system is.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

VIII. CONCLUSION AND FUTURE WORK

The present paper introduced PaasMitra, an innovative Flutter and Firebase-based application that aims to combine the idea of proximity-based task discovery and community engagement while also incorporating both volunteering and paid person-to-person assistance. The article discussed the proposed system's architecture, data model, implementation, and task paradigm. PaasMitra's primary strength is offering an integrated platform that enables the users to request both free and paid assistance in their everyday neighbourhood tasks.

The described prototype can be used as a conceptual basis for future solutions that will bring together community-oriented assistance and monetary reward-based microtask platforms. However, additional functional and performance testing is required to evaluate the developed concept's practical applicability and potential impact. The findings from Figure 2, which represent the quantitative results of task-seeking experiments, should also be supported by additional practical trials that will measure the real-world efficacy of the application.

Next, the application's future development should include the addition of user rating, reviews, and identity verification features, as well as notification, billing, and data security systems. The addition of more complex task recommendation algorithms can also be beneficial for thePaasMitra's future development by enabling more relevant matches between task-sellers and task-buyers.

REFERENCES

[1] M. Dedema and H. Rosenbaum, "Socio-technical issues in the platform-mediated gig economy: A systematic literature review: An Annual Review of Information Science and Technology (ARIST) paper," *J. Assoc. Inf. Sci. Technol.*, vol. 75, no. 3, pp. 344–374, Mar. 2024, doi: 10.1002/asi.24868.

[2] X. Guo, Z. Cheng, and P. A. Pavlou, "Skill-biased technical change, again? Online gig platforms and local employment," *Inf. Syst. Res.*, vol. 36, no. 3, pp. 1354–1374, Sep. 2025, doi: 10.1287/isre.2022.0307.

[3] A. J. Wood, N. Martindale, and B. J. Burchell, "Beyond the 'gig economy': Towards variable experiences of job quality in platform work," *Work Employ. Soc.*, vol. 39, no. 5, pp. 1154–1178, Oct. 2025, doi: 10.1177/09500170251336947.

[4] A. James, "Platform work-lives in the gig economy: Recentering work–family research," *Gender Work Organ.*, vol. 31, no. 2, pp. 513–534, Mar. 2024, doi: 10.1111/gwao.13087.

[5] F. D. A. Alauddin, A. Aman, M. F. Ghazali, and S. Daud, "The influence of digital platforms on gig workers: A systematic literature review," *Heliyon*, vol. 11, no. 1, Jan. 2025, Art. no. e41491, doi: 10.1016/j.heliyon.2024.e41491.

[6] M. Ramachandran and A. Kulandai, "Gig workers: A paradigm shift in the Indian workforce—A bibliometric analysis and future directions," *Metamorphosis: J. Manage. Res.*, vol. 24, no. 1, pp. 7–20, Jun. 2025, doi: 10.1177/09726225241257731.

[7] B. Parthasarathy *et al.*, "The developmental possibilities of gig work on digital platforms: The limits to the hope and hype in India," *Indian J. Labour Econ.*, vol. 68, no. 2, pp. 569–587, Jun. 2025, doi: 10.1007/s41027-024-00522-4.

[8] S. Sun, H. Li, and X. Wei, "Creative mutual aid community in the context of 'Internet +'," in *Proc. 8th Int. Conf. e-Soc., e-Learn. e-Technol. (ICSLT)*, Rome, Italy, Jun. 2022, pp. 1–6, doi: 10.1145/3545922.3545923.

[9] K. R. Wilson, O. M. Roskill, and J. Mahr, "Mutual aid using digital technology: A case study of virtual community organizing during the COVID-19 pandemic," *J. Community Pract.*, vol. 30, no. 3, pp. 255–278, Jul. 2022, doi: 10.1080/10705422.2022.2102101.

[10] Z. Zhou, L. Wang, and Y. Dong, "Research on innovative design of community mutual aid elderly care service platform based on Kano model," *Heliyon*, vol. 9, no. 5, May 2023, Art. no. e15546, doi: 10.1016/j.heliyon.2023.e15546.

[11] N. Kenworthy, E. Hops, and A. Hagopian, "Mutual aid praxis aligns principles and practice in grassroots COVID-19 responses across the US," *Kennedy Inst. Ethics J.*, vol. 33, no. 2, pp. 115–144, Jun. 2023, doi: 10.1353/ken.2023.a904080.

[12] C. Hultquist and R. M. Tubbeh, "Digital sociotechnical systems of mutual aid: How communities connected, adapted, and innovated during the COVID-19 pandemic in New York City," *Citizen Sci.: Theory Pract.*, vol. 7, no. 1, May 2022, Art. no. 20, doi: 10.5334/cstp.454.

[13] D. Sackitey, H. A. Sachdev, and A. G. Parker, "Virtual solidarity, concrete care: A review of mutual aid online," in *Proc. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, Yokohama, Japan, Apr. 2025, pp. 1–23, doi: 10.1145/3706598.3713653.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

[14] X. Lin, K. Wei, Z. Li, J. Chen, and T. Pei, "Aggregation-based dual heterogeneous task allocation in spatial crowdsourcing," *Front. Comput. Sci.*, vol. 18, no. 6, Dec. 2024, Art. no. 186605, doi: 10.1007/s11704-023-3133-6.

[15] W. Huang, P. Li, B. Li, Q. Liu, L. Nie, and H. Bao, "Three-sided online stable task assignment in spatial crowdsourcing," *Inf. Sci.*, vol. 654, Jan. 2024, Art. no. 119878, doi: 10.1016/j.ins.2023.119878.

[16] X. Li, L. Zhang, M. Zhou, and K. Bian, "Task recommendation based on user preferences and user-task matching in mobile crowdsensing," *Appl. Intell.*, vol. 54, no. 1, pp. 131–146, Jan. 2024, doi: 10.1007/s10489-023-05208-w.

[17] Y. Ma, X. Gao, and G. Chen, "Crowdsourcing task assignment with category and mobile combined preference learning," in *Proc. 22nd Int. Conf. Service-Oriented Comput. (ICSOC)*, Tunis, Tunisia, Dec. 2024, pp. 69–84, doi: 10.1007/978-981-96-0805-8_5.

[18] M. Zhang, S. Chen, Z. Wei, and Y. Wu, "Preference-matched multitask assignment for group socialization under mobile crowdsensing," *Sensors*, vol. 23, no. 4, Feb. 2023, Art. no. 2275, doi: 10.3390/s23042275.

[19] M. Puerta-Beldarrain, O. Gómez-Carmona, L. Chen, D. López-de-Ipiña, D. Casado-Mansilla, and F. Vergara-Borge, "A spatial crowdsourcing engine for harmonizing volunteers' needs and tasks' completion goals," *Sensors*, vol. 24, no. 24, Dec. 2024, Art. no. 8117, doi: 10.3390/s24248117.

[20] P. He, Y. Xin, Z. Li, and Y. Yang, "pMATE: A privacy-preserving map retrieval task assignment scheme in spatial crowdsourcing," *Secur. Commun. Netw.*, vol. 2023, Jan. 2023, Art. no. 9477320, doi: 10.1155/2023/9477320.

[21] Y.-F. Ge, H. Wang, E. Bertino, J. Cao, and Y. Zhang, "Multiobjective privacy-preserving task assignment in spatial crowdsourcing," *IEEE Trans. Cybern.*, vol. 55, no. 8, pp. 3584–3597, Aug. 2025, doi: 10.1109/TCYB.2025.3573292.

[22] Y. Xu, M. Xiao, J. Wu, and H. Sun, "Privacy preserving task push in spatial crowdsourcing with unknown popularity," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 11, pp. 2039–2053, Nov. 2024, doi: 10.1109/TPDS.2024.3434978.

[23] A. Calabrò, T. M. Nisar, M. Torchia, and H.-T. Tseng, "Establishing trust in the sharing economy: An analysis of organizational, systems and interpersonal level trust," *Inf. Technol. People*, vol. 36, no. 7, pp. 3023–3047, Nov. 2023, doi: 10.1108/ITP-04-2022-0313.

[24] A. Pumputis, "Complexities of trust building through sociomaterial arrangements of peer-to-peer platforms," *Curr. Issues Tourism*, vol. 27, no. 11, pp. 1800–1813, Jun. 2024, doi: 10.1080/13683500.2023.2214848.

[25] S. Tadelis, "Reputation, feedback, and trust in online platforms," in *Reengineering the Sharing Economy: Design, Policy, and Regulation*, B. Heydari, O. Ergun, R. Dyal-Chand, and Y. Bart, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2023, pp. 62–82, doi: 10.1017/9781108865630.006.

[26] I. Hidayat-ur-Rehman, H. K. Alay, M. T. Deveciyan, F. Akhter, and S. Alzahrani, "Empirical analysis to investigate the relationship between digital sharing economy and digital trust," *Future Bus. J.*, vol. 12, no. 1, Jan. 2026, Art. no. 12, doi: 10.1186/s43093-025-00720-6.

[27] T. D. Oesterreich, E. Anton, F. M. Hettler, and F. Teuteberg, "What drives individuals' trusting intention in digital platforms? An exploratory meta-analysis," *Manage. Rev. Quart.*, vol. 75, no. 4, pp. 3615–3667, Dec. 2025, doi: 10.1007/s11301-024-00477-2.

[28] D.-C. Dabija, L. M. Csorba, F.-L. Isac, and S. Rusu, "Building trust toward sharing economy platforms beyond the COVID-19 pandemic," *Electronics*, vol. 11, no. 18, Sep. 2022, Art. no. 2916, doi: 10.3390/electronics11182916.

[29] S. An, T. Eck, and H. Yim, "Understanding consumers' acceptance intention to use mobile food delivery applications through an extended technology acceptance model," *Sustainability*, vol. 15, no. 1, Jan. 2023, Art. no. 832, doi: 10.3390/su15010832.

[30] Y. Chen and M. H. Ryu, "Chinese consumers' attitudes toward and intentions to continue using skill-sharing service platforms," *Behav. Sci.*, vol. 14, no. 9, Sep. 2024, Art. no. 765, doi: 10.3390/bs14090765.

[31] B. Suri, S. Taneja, I. Bhanot, H. Sharma, and A. Raj, "Cross-platform empirical analysis of mobile *Inst.*, vol. 28, pp. 217–222, Sep. 2023, doi: 10.35784/jcsi.3677.application development frameworks: Kotlin, React Native and Flutter," in *Proc. 4th Int. Conf. Inf. Manage. Mach. Intell. (ICIMMI)*, Jaipur, India, Dec. 2022, pp. 1–6, doi: 10.1145/3590837.3590897.

[32] J. Piskor and M. Badurowicz, "Performance comparison of Flutter platform GUI in web and native environments," *J. Comput. Sci.*

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

[33] D. Zou and M. Y. Darus, "A comparative analysis of cross-platform mobile development frameworks," in *Proc. IEEE 6th Symp. Comput. Informat. (ISCI)*, Kuala Lumpur, Malaysia, Aug. 2024, pp. 84–90, doi: 10.1109/ISCI62787.2024.10667693.

[34] A. I. Abd ElGhany *et al.*, "Performance evaluation of cross-platform AI-powered content moderation systems: Flutter vs native Android," *Int. J. Mobile Appl. Technol.*, vol. 2, no. 2, Jun. 2026, doi: 10.51137/wrp.ijmat.774.

[35] A. M. Sattar *et al.*, "Accelerating cross-platform development with Flutter framework," *J. Open Source Develop.*, vol. 10, no. 2, pp. 1–11, 2023, doi: 10.37591/joosd.v10i2.580.

[36] M. F. Younis and Z. S. Alwan, "Monitoring the performance of cloud real-time databases: A Firebase case study," in *Proc. Al-Sadiq Int. Conf. Commun. Inf. Technol. (AICCIT)*, Al-Muthana, Iraq, Jul. 2023, pp. 240–245, doi: 10.1109/AICCIT57614.2023.10217953

(This work is licensed under a Creative Commons Attribution 4.0 International License.)