WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SMART CAMPUS TRAINING SYSTEM Servi: A Location-Based On-Demand Platform for Connecting Households with Nearby Non-Employed Engineers for Home Appliance Services

Ms. Hemlatha M A[1], Girish Balaji V S[2], Monesh S[3], Jeeva Williams A[4], Gokul M[5]

Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India.
E-mail ID: girishbalaji15@gmail.com, moneshs25ad@srishakthi.ac.in, jeevawilliamsa31@gmail.com, gokulm0708@gmail.com.

ABSTRACT :

Household appliances and fittings such as taps, switches, fans and furniture frequently develop small faults that need quick attention. Finding a reliable technician for such minor repairs is often slow and uncertain, while many trained engineers and diploma holders remain without regular employment. To address both problems, this project presents the Smart Campus Training System, implemented as a mobile-friendly application named Servi, which connects households with nearby non-employed engineers who can earn an income by providing short, local repair services.

The proposed system follows the working idea of ride-hailing applications such as Rapido. A customer signs in, chooses a service such as plumber, electrician or carpenter, and views the nearest available professionals with their distance and estimated arrival time. A short description of the problem is sent as a request. The service provider receives the request, ordered nearest first, and may accept or decline it. After acceptance, the customer can track the request status and call the provider, while the provider can view the route on a map and open it in Google Maps. The job is then marked as completed.

Because the nearest provider is selected, travel time is kept short and customers do not have to wait for long periods. The prototype was exercised through a complete request cycle, in which a plumbing request raised from a customer account was accepted by a provider located 0.07 km away, with an estimated arrival time of about one minute. The results demonstrate the working of the main workflow in the prototype; wider testing with real users is still required.

Keywords: Smart Campus Training System, Servi, On-Demand Services, Location-Based Matching, Nearest Provider, Home Services, Service Provider Tracking, OpenStreetMap.

INTRODUCTION

Minor repairs at home, such as a leaking tap, a faulty switch or a broken cupboard hinge, are part of everyday life. When such problems occur, people usually depend on personal contacts or a local search to find a technician. This process can take a long time, the arrival time is uncertain, and the cost is not always clear. For small jobs, the effort of finding a technician can be larger than the repair itself.

At the same time, many engineering graduates and diploma holders do not have regular jobs even though they possess useful technical skills. These people could carry out minor household repairs if they were connected to customers who need such services in their own neighbourhood.

Ride-hailing applications such as Rapido have shown that customers and nearby workers can be matched quickly using location information. The same idea can be applied to home services. In this project, the customer requests a service, the system shows the nearest available provider, and the provider accepts the request and travels to the customer's address.

This project proposes the Smart Campus Training System, implemented as the Servi application. The objectives are to reduce the time needed to find a technician for small household problems, to create an earning opportunity for non-employed engineers, to give them practical field experience, and to avoid long waiting times for customers. The system acts as a connecting platform between customers and providers and does not replace professional contractors for large or specialised work.

1. SYSTEM ARCHITECTURE

The Servi architecture consists of a customer interface, a service provider interface, a location and matching module, a request management module, a navigation module and a shared data store. These modules work together so that a request raised by a customer reaches the nearest provider and its status is reflected on both sides. The overall workflow is shown in Figure 1.

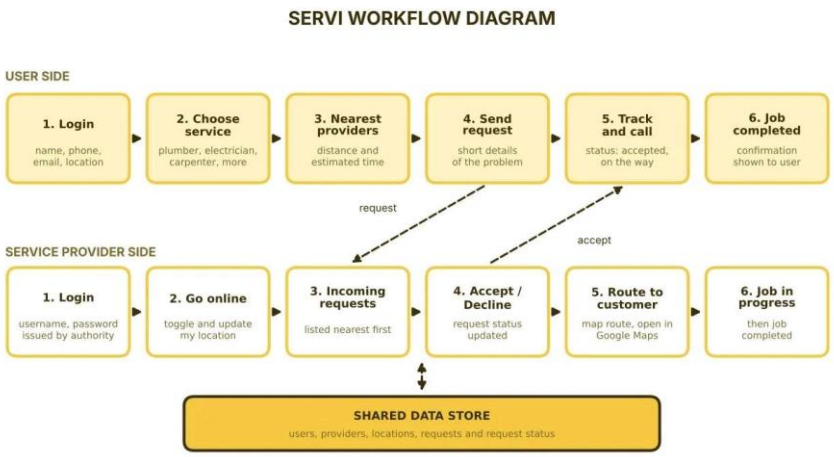


Figure 1: Servi workflow diagram

1.1 Customer Module

The customer module allows a user to access services. A new user enters a full name, phone number, email address and location, and an account is created automatically on the first sign-in. After signing in, the home screen greets the user, displays the current location, provides a search box and lists the available services: Plumber, Electrician, Carpenter and More.

Once a service is selected, the customer sees the nearest available professionals, enters a short description of the problem and sends the request. After the request is accepted, the customer can see the provider's name, rating, distance and estimated arrival time, call the provider, or cancel the request.

1.2 Service Provider Module

Service providers sign in using a username and password issued by an authority, which helps to keep the list of providers controlled. The provider home screen shows an online/offline toggle, an option to update the current location, and summary values such as services completed, rating and pending requests.

When the provider is online, new nearby requests appear under Incoming Requests, sorted nearest first. Each request shows the service type, the customer's name and area, the distance, the estimated time and the problem description, with Accept and Decline options.

1.3 Location and Nearest-Provider Matching Module

This module stores the location of the customer and of every online provider and calculates the distance between them. Providers are arranged in order of distance, and the estimated arrival time is derived from it. The nearest available provider is therefore shown first to the customer, and incoming requests are shown nearest first to the provider. Providers who are offline are not included in the list.

The accuracy of matching depends on the location values reported by the devices. For this reason, the provider can refresh the location with the Update my location option, and the customer can change the location before sending a request.

1.4 Request Management Module

This module handles the life cycle of a request. A request is created when the customer submits the problem details. Its status then changes through the stages accepted, on the way, in progress and completed, or it may be declined or cancelled. The status is stored in the shared data store so that the customer and the provider always see the same information.

1.5 Navigation and Route Module

After a request is accepted, the provider's Current Job card shows a map with the customer's location and the provider's position, together with the fastest road route, its distance and its approximate time. The map is displayed using Leaflet with OpenStreetMap data. A Start route in Google Maps button opens turn-by-turn navigation to the customer's location in Google Maps.

1.6 Shared Data Store

A shared data store keeps the users, providers, locations, requests and request status. The customer side and the provider side both read from and write to this store, which allows a request sent by a customer to appear on the provider's screen and the acceptance made by the provider to appear on the customer's screen.

2. LITERATURE REVIEW

On-demand platforms use the position of the user and of the service worker to match them quickly. Ride-hailing applications are the most familiar example: a rider requests a trip, nearby drivers are notified, and the nearest willing driver accepts the request. The same pattern can be applied to other local services, in which the worker travels to the customer.

Location-based matching relies on calculating the distance between two geographic coordinates. The great-circle (Haversine) formula is a commonly used method for finding the distance between two points on the Earth's surface from their latitude and longitude, and it is suitable for ranking nearby candidates. Road routes, however, may be longer than the straight-line distance, so a routing service is used when an actual travel route is required.

Open mapping data has also made it easier to build location-based applications. OpenStreetMap provides freely usable, community-maintained street maps, and lightweight map libraries such as Leaflet can display them in web and mobile applications.

Existing home-service platforms usually require customers to book a slot in advance and are mainly operated by large service companies. Servi differs in that it focuses on immediate requests for small repairs, selects the nearest provider, and aims to give non-employed engineers a way to earn from local work. Its effectiveness, in terms of cost, customer satisfaction and provider income, has not yet been measured and requires further study.

3. METHODOLOGY

The development of Servi followed a structured process involving requirement analysis, workflow design, interface design, integration of location and map services, and testing of the complete request cycle.

3.1 Requirement Analysis

The system was designed to support the following functions:

- Simple sign-in for customers with name, phone number, email address and location.
- Controlled sign-in for service providers using credentials issued by an authority.
- Selection of a service category such as plumber, electrician or carpenter.
- Display of the nearest available providers with distance and estimated time.
- Sending of a request with a short description of the problem.
- Accepting or declining of requests by providers, with the status updated for the customer.
- Display of the route to the customer and navigation through Google Maps.
- Confirmation to the customer when the job is completed.

3.2 Workflow Design

The workflow has a user side and a service provider side, as shown in Figure 1. On the user side, the steps are login, choosing a service, viewing the nearest providers, sending a request, tracking and calling the provider, and job completion. On the provider side, the steps are login, going online, receiving incoming requests, accepting or declining, following the route to the customer, and completing the job. The two sides communicate through the shared data store: the request flows from the customer to the provider, and the acceptance flows back.

3.3 Technology Stack

The prototype uses the following components:

- **Mobile-friendly web interface:** Customer and provider screens designed for use on a phone.
- **Shared data store:** Storage of users, providers, locations, requests and request status.
- **Device location:** Capturing the position of the customer and the provider.
- **Leaflet and OpenStreetMap:** Display of the map and the route on the provider's job screen.
- **Google Maps:** Turn-by-turn navigation from the provider to the customer's address.

3.4 Integration and Testing

The customer and provider screens were connected through the shared data store, and a complete request cycle was tested using a customer account (Jeeva) and a plumber account (Ravi Kumar). The test covered sign-in, service selection, nearest-provider display, request submission, acceptance, route display, navigation in Google Maps and job completion. The screenshots of this test are presented in the next section.

4. RESULT ANALYSIS

This section presents the output of the Servi prototype during one complete request cycle, from the customer's sign-in to the completion of the job.

4.1 Customer Side Output

Figure 2 shows the sign-in screen, where a user chooses between I need a service and Service provider and enters the required details. After signing in, the home screen lists the available services and the user's location.

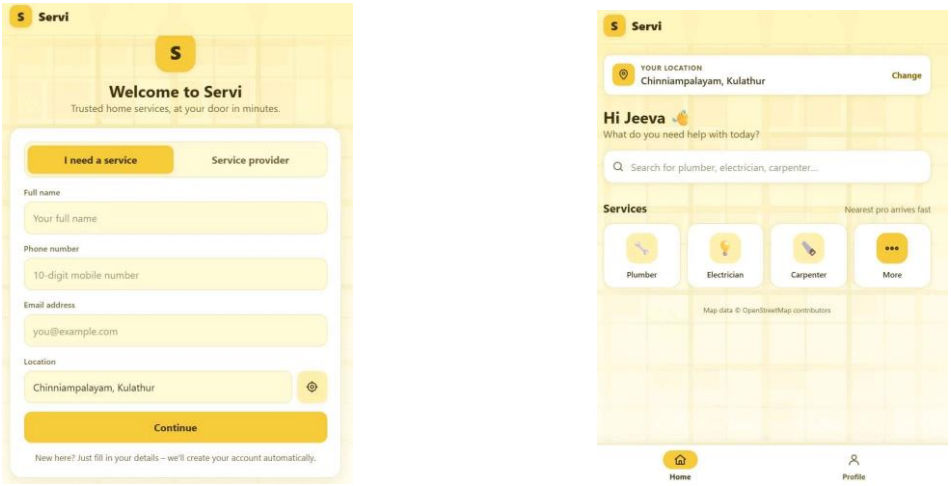


Figure 2: Sign-in screen (left) and customer home screen (right)

On selecting Plumber, the nearest available professional is displayed along with distance and estimated time. The customer entered the problem as "water tap leaking" and sent the request. Once accepted, the screen changed to Plumber is on the way, showing the provider's details, the distance, the arrival time and the options to call or cancel (Figure 3).

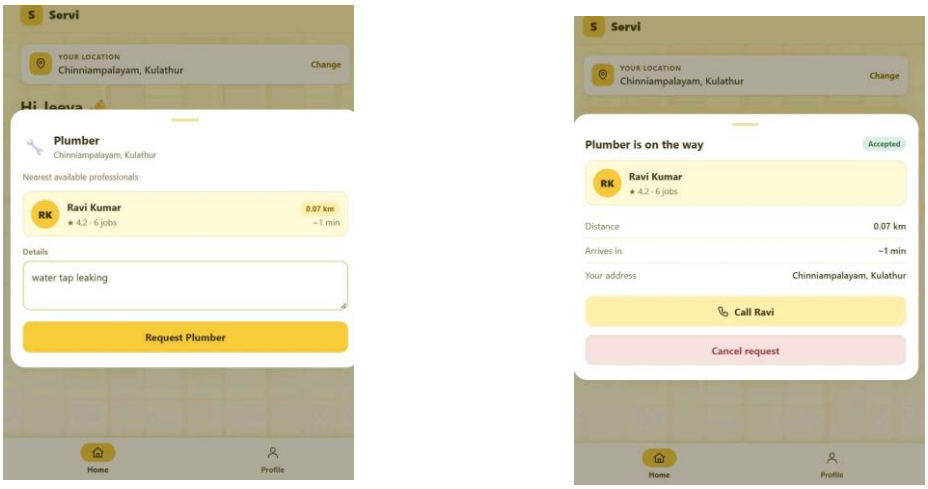


Figure 3: Request screen (left) and request accepted screen (right)

4.2 Service Provider Side Output

On the provider side, the plumber was online and the incoming request appeared with the customer's name, area, description and distance of 0.07 km (about 1 minute). After the provider accepted the request, the job appeared under Current Job with the status In progress and a route of 0.1 km, about 1 minute, was displayed (Figure 4).

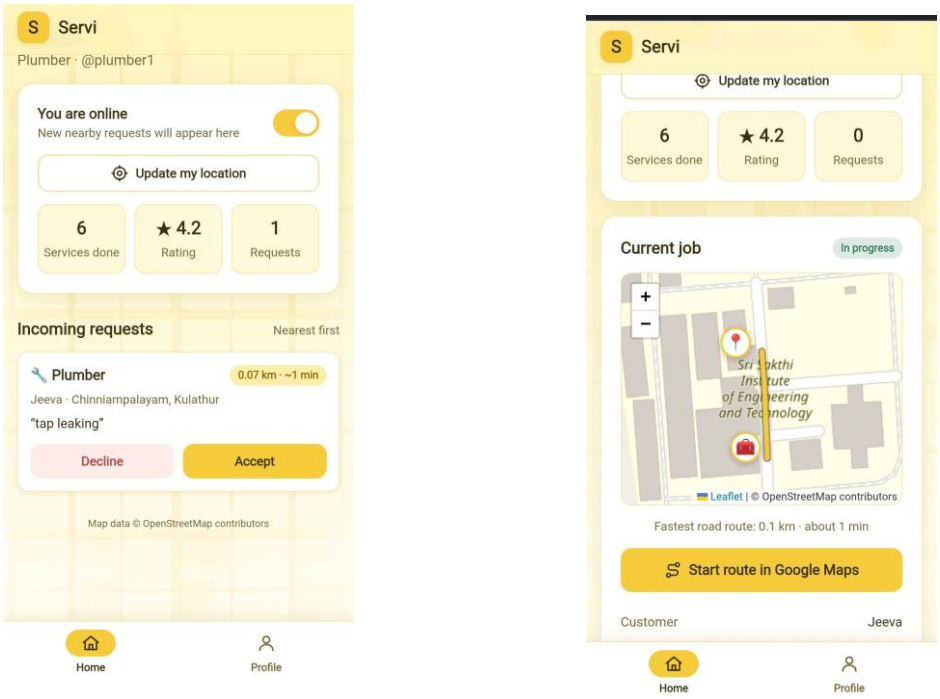


Figure 4: Provider home with incoming request (left) and current job with route map (right)

Pressing Start route in Google Maps opened turn-by-turn navigation to the customer's location, as shown in Figure 5. When the job was finished, the customer received a Job completed message.

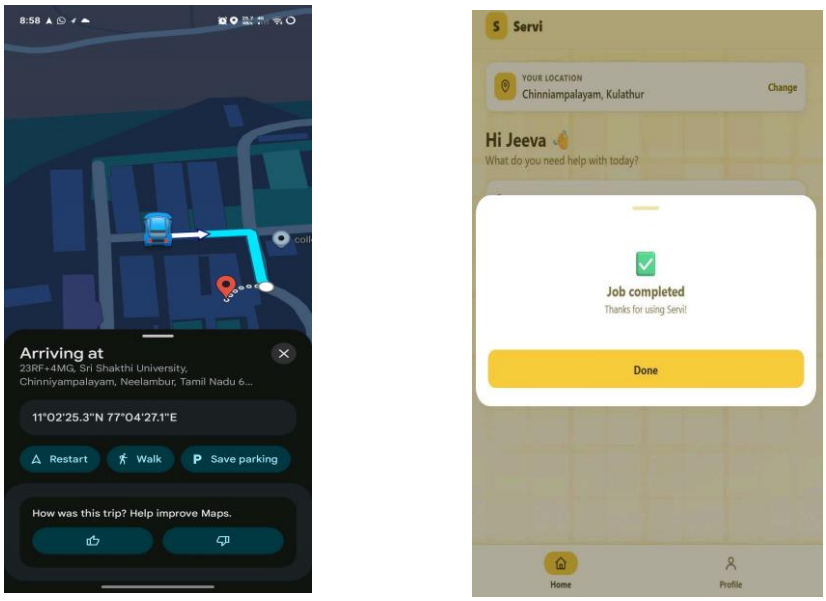


Figure 5: Google Maps navigation (left) and job completion message (right)

4.3 Summary of Observed Results

S.No	Function tested	Observed result
1	Customer sign-in	Account created and home screen displayed with the customer's name and location.
2	Service selection	Plumber, Electrician, Carpenter and More options displayed.
3	Nearest provider	Ravi Kumar displayed at 0.07 km, about 1 min, rating 4.2 with 6 jobs.
4	Request submission	Request "water tap leaking" received by the provider as an incoming request.
5	Accept request	Customer screen changed to Plumber is on the way with Accepted status.

S.No	Function tested	Observed result
6	Route display	Fastest road route of 0.1 km, about 1 min, shown on the map.
7	Navigation	Route opened successfully in Google Maps.
8	Job completion	Job completed message displayed to the customer.

The results show that the main workflow of the prototype operated from request to completion, and that the nearest provider could be reached within a very short time in the tested case. The test was carried out inside a small area with one customer and one provider, so the distance and time values are not representative of a real city. The prototype should be tested with more providers, different locations and real users before conclusions are drawn about its speed, reliability and scalability.

5. CONCLUSION AND FUTURE WORK

The Smart Campus Training System, implemented as Servi, provides a simple way to connect households with nearby non-employed engineers for small repair work. Customers can select a service, view the nearest provider, send a request and track its progress, while providers can go online, receive requests sorted by distance, accept them and follow the route to the customer. The prototype was demonstrated through a complete request cycle in which a plumbing request was accepted and completed.

The platform has the potential to reduce the time customers spend searching for technicians and to give skilled but unemployed people an income and practical experience. Further testing is necessary to establish its performance with real users, and the platform should be limited to minor, low-risk repairs unless providers are properly verified.

Future enhancements include:

- Online payment and transparent price estimates before the job begins.
- Rating and review submission by the customer after every job.
- Identity and skill verification of providers, and basic safety training.
- Live tracking of the provider's position on the customer's screen.
- Push notifications for new requests and status changes.
- Support for more service categories such as AC repair, appliance repair and painting.
- Support for regional languages, including Tamil.
- Evaluation with real customers and providers to measure waiting time, satisfaction and provider income.

REFERENCES

1. Haklay, M., and Weber, P. "OpenStreetMap: User-Generated Street Maps." IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, 2008.
2. Sinnott, R. W. "Virtues of the Haversine." Sky and Telescope, vol. 68, no. 2, p. 159, 1984.
3. OpenStreetMap contributors. "OpenStreetMap." https://www.openstreetmap.org.
4. Leaflet. "An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps." https://leafletjs.com.