HOME SUPPORT:

Your Daily Service Partner

Created By-: Satyam Jain

B.Tech (Information Technology)-1st Year (2025-26)

Project Duration: 17/07/2025 - Ongoing

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to everyone who played a crucial role in the development of this project.

First and Foremost I sincerely thank to Mr. Amit Bhatia, Faculty of Computer Science at Holy Angels' Convent School Muzaffaranagar and founder of ITSKILLS Muzaffarnagar, for his invaluable guidance, inspiration, and support throughout my learning journey.

It was under his mentorship that I was first introduced to the world of programming, and it is because of his efforts that I was able to understand core Java and apply it meaningfully to create this project. His way of teaching, clarity of thought, and constant motivation pushed me to go beyond classroom learning and build something practical.
I also extend my sincere thanks to my **parents** who provided the right environment and encouragement throughout this journey.

Though I am at the very beginning of my B.Tech journey, this project reflects my passion for solving real-life problems through technology and my deep curiosity to explore beyond textbooks.

Satyam Jain

INTRODUCTION

In today’s fast-paced world, managing household services efficiently has become a growing challenge for many individuals and families. Whether it's finding a reliable electrician, plumber, painter, or appliance technician, people often struggle to get timely and trusted support.

“Home Support: Your Daily Service Partner” is a Java-based desktop application designed to bridge this gap by acting as a centralized platform for managing daily service needs in a structured and efficient way. This project simplifies the process of requesting services, assigning qualified service partners, and tracking service status — all through a clean and user-friendly interface built with Java Swing and MySQL.

This project was developed before the start of my formal B.Tech journey, purely out of passion for learning and building real-world solutions through code. While the application follows a basic structure, it lays the foundation for how service-based portals can function in a digitized society.

The project is modular in nature, consisting of four main panels:

1. **User Panel** – For service request management
2. **Company Panel** – For employee-level request tracking and partner assignment
3. **Admin Panel** – For overall control and workforce management
4. **Service Partner Panel** – For viewing assigned tasks and tracking service progress

This documentation outlines the motivation, structure, and functionality of the system in detail and serves as a complete guide to its development and scope.

Problem Statement

In a rapidly evolving digital age, people increasingly rely on quick, reliable, and on-demand services for their day-to-day needs. However, when it comes to home-based support services like electricians, plumbers, appliance repair specialists, painters, and carpenters — finding trustworthy and available professionals remains a challenge, especially in tier-2 and tier-3 cities.

Users often face problems such as:

- Not knowing where to find service partners.
- Uncertainty about technician availability.
- Delays in response and coordination.
- No way to track request status or get confirmation.

On the other hand, service providers and managing companies lack a centralized system to:

- Receive and track user requests.
- Assign service tasks efficiently.
- Monitor service quality, task status, or user feedback.
- Manage service partner data or performance history.

Despite growing demand, there is a noticeable **gap between users who need help and service professionals who can provide it** — primarily due to lack of integration and poor workflow handling.

Objective

The objective of this project, **“Home Support: Your Daily Service Partner,”** is to build a structured and easy-to-use software system using **Core Java (Swing for GUI)** and **MySQL** to provide a solution for these challenges. This project intends to:

- **Bridge the gap** between users and local service partners through a simple request-based workflow.
- Empower users to raise, cancel, and track service requests through a dedicated **User Panel**.
- Allow company employees to view requests, assign service partners, and update task statuses via the **Company Panel**.
- Enable future scalability with additional features such as performance tracking, feedback system, and task analytics using an **Admin Panel** and a **Service Partner Panel**.
- Promote structured and transparent backend operations using database integration, panel-based modules, and clear role-based access.

At its core, this project reflects a **problem-solving approach**, simulating a **real startup model** for home service management and creating a strong foundation for future mobile/web-based deployment.

System Scope

This project, *Home Support: Your Daily Service Partner*, is aimed at streamlining the daily household service management process. It is designed with scalability, user-friendliness, and real-time communication in mind. The scope of the system includes:

- Enabling users to easily raise, cancel, and track service requests.
- Providing a dedicated interface for company employees to view, approve/reject, and assign service tasks to available partners.
- Maintaining accurate and structured data flow through MySQL database integration.
- Ensuring errorless, button-driven functionality for a smooth user experience.
- Displaying all four major panels on the home screen:
→ **User Panel**,
→ **Company Panel**,
→ **Admin Panel**,
→ **Service Partner Panel**.
(Currently, only the User and Company panels are fully functional; Admin and Partner panels are planned for the next version.)

This version serves as a base model, with future updates focusing on enhancing automation, partner-side features, and admin-level control for deeper insights and analytics.

Technologies Used

The development of *Home Support: Your Daily Service Partner* involved the use of the following technologies and tools:

- **Java (Core + Swing)**
  Used for designing the entire graphical user interface (GUI) and implementing the project's logic in a desktop-based environment.
- **MySQL**
  A relational database used for storing, retrieving, and managing all the service requests, user details, status updates, and service partner records.
- **JDBC (Java Database Connectivity)**
  Acts as a bridge between the Java Swing application and the MySQL database to perform real-time data operations.
- **Apache NetBeans IDE**
  An open-source integrated development environment used for writing, designing, and testing the entire application with GUI drag-and-drop features.
- **Manual Error Handling & Input Validations**
  Ensured smooth execution and avoided system crashes through conditional logic and field-level checks.

System Requirements

This section lists the hardware and software environment required to run the application smoothly.

Hardware Requirements

- Minimum 4 GB RAM (Recommended: 8 GB)
- Dual Core Processor or above
- Minimum 200 MB of disk space

Software Requirements

- Java JDK 17 or above
- Apache NetBeans IDE (v12+ recommended)
- MySQL Server & Workbench
- Windows OS (Tested on Windows 10)
- JDBC Driver (MySQL Connector/J)

Modules Overview

1. User Panel

- Submit new service request (Electrician, Plumber, etc.)
- Cancel submitted request
- Track request status using Request ID

2. Company Panel

- Login for company employees
- View requests by date
- Update request status (Approve/Reject/Assign)
- Assign service partner to request
- Search available service partners

(Admin & Service Partner Panels are in progress and will be added in future versions)

Tables Used in Version 1.0

1. request

Stores customer service requests submitted via the User Panel.

Field Name	Data Type	Description
r_id	varchar(15)	Unique request ID generated for every new service request.
c_name	varchar(25)	Full name of the customer who raised the request.
cno	varchar(12)	Contact number of the customer.
email	varchar(25)	Email ID of the customer (for communication).
address	varchar(50)	Full residential address of the customer.
l_mark	varchar(30)	Landmark near the customer's address to help in locating.
s_type	varchar(20)	Type of service requested (e.g., Electrician, Plumber).
p_desc	varchar(100)	Description of the problem written by the user.
p_date	date	Preferred date selected by the customer for service.
p_time	varchar(15)	Preferred time (Morning/Evening/Afternoon).
r_date	date	Request creation date (auto-added at time of submission).
res_note	varchar(100)	Resolution note added by company after service completion.

2.employee

Stores data of all employee autorised for using company panel and handle requests.

Field Name	Data Type	Description
emp_id	varchar(20)	Unique ID of the employee.
emp_password	varchar(20)	Password for login authentication.
emp_name	varchar(30)	Full name of the employee.
ecno	varchar(13)	Contact number of the employee.
desig	varchar(20)	Designation (e.g., Executive, Manager, etc.).
salary	decimal(10,0)	Monthly salary of the employee.
qualify	varchar(15)	Educational qualification (e.g., B.Tech, MCA).

3. status

Tracks the progress of each request and allows the Company Panel to update status

Field Name	Data Type	Description
r_id	Varchar(15)	Unique request ID
c_name	varchar(25	Name of the customer (copied from request for quick view).
cno	varchar(12)	Customer’s contact number
s_type	varchar(20	Service type requested by the customer
r_date	date	Date when the request was initially submitted. Default is the system date.
status	varchar(20)	Current status of the request — values
sp_id	varchar(15)	Service Partner ID assigned for handling the task
up_date	date	Last updated date — reflects when status was last modified
appl_type	varchar(25)	Appliance type, used only when service is of type Appliance Care
brand	varchar(25)	Appliance brand, used only when service is of type Appliance Care

4. partner

Stores data of all available service partners, searchable via Company Panel

Field Name	Data Type	Description
sp_id	varchar(15)	Unique ID assigned to each service partner.
sp_name	varchar(30)	Full name of the service partner.
sp_cno	varchar(15)	Contact number of the service partner.
sp_type	varchar(20)	Service category (e.g., Electrician, Plumber, etc.).
spap_type	varchar(25)	Type of appliance (only for appliance care service).
spap_brand	varchar(20)	Brand of appliance (only if applicable).
sp_status	varchar(20)	Current status (Available, Busy, etc.).
tp_pending	decimal(10,0)	Total pending tasks assigned to the partner.
vs_charge	decimal(10,0)	Visiting charge of the partner.

Version Control

Versions	Description	Status
v1.0	User Panel + Company Panel (with full backend logic)	✅ Completed
v2.0	Admin Panel (Add/Remove Employee & Partners, View Requests, Salary Management)	🔜 Under Development
v3.0	Service Partner Panel (Task View, Completion Update, Feedback Access)	🔜 Under Development
v4.0	Web/App Version for Production Use	🔜 Planned

🚧 Note

The current documentation includes **only Version 1.0** features and database design. As the project progresses, upcoming versions will introduce more panels and database tables such as:

- admin (for login and privileges)
- feedback (user feedback to partners)
- task_summary (partner's ratings, task completed, task pending, etc.)

All future additions will be documented separately as part of Version 2.0 and beyond.

Working Flow of the Application

The application follows a structured flow to handle daily service requests efficiently. Below is the detailed working flow for the currently implemented modules:

---

1. User Panel Flow

- User launches the application and selects a service (Electrician, Plumber, etc.).
- Fills a **New Request Form** with personal and service details.
- Upon submission, entries are inserted into both request and status tables.
- A unique **Request ID** is generated and shown to the user.
- User can:
    - Cancel the request using **Request ID**, which removes records from both tables.
    - Check the current status by entering the **Request ID** (from status table).

---

2. Company Panel Flow

- Company employee logs in using credentials (from employee table).
- Access to 4 main functionalities:
    1. **View Requests by Date**:
        - Employee enters a date and views all requests from request table made on that date.

2. **Update Request Status**:
    - Employee enters the Request ID, sees full details, and can **Approve** or **Reject**.
    - Status is updated in the status table along with the updation date.
3. **Search Service Partner** *(based on current demand)*:
    - Employee selects criteria like Service Type, Appliance Type, Brand, etc.
    - Matching service partners are fetched (from upcoming service_partner table).
4. **Assign Service Partner**:
    - Employee enters Request ID, Service Partner ID, and date.
    - Status table is updated with assigned_spid and status becomes **"Assigned"**.

- **Logout** option brings the employee back to the login screen.

📌 ***Notes:***

- All transitions are controlled through Swing GUI buttons.
- All database operations are handled using JDBC.
- Exception handling and validations are implemented to avoid crashes or wrong entries.

Testing and Validation

The application was tested thoroughly at each stage of development to ensure smooth functionality and error-free performance. The following approaches were followed during testing:

---

1. Module-wise Testing

- **User Panel**
  - Tested all 3 features (New Request, Cancel Request, Check Status).
  - Verified that correct data gets inserted/deleted from the request and status tables.
  - Verified that invalid Request IDs are handled gracefully.
- **Company Panel**
  - Verified employee login validation using correct/incorrect credentials.
  - Checked View Request functionality with valid and invalid dates.
  - Ensured that request statuses update correctly in status table.
  - Confirmed Search Service Partner filters work as expected.
  - Assigning partner updates status and partner ID correctly.

---

2. Exception Handling

- Handled SQLException, NullPointerException, and empty field checks in forms.
- Displayed appropriate messages like:
    - “Oops! Try Again”
    - “Request Submitted”
    - “Request ID not found”

---

3. Manual Validation

- Data was manually checked in **MySQL Workbench** after each operation.
- Random test cases (like wrong IDs, empty fields, duplicate requests) were entered to check app robustness.
- All frames and buttons tested multiple times to ensure errorless operation.

Conclusion:

The current version of the application has passed all basic validation and functionality checks. The system works efficiently for real-time testing and usage under defined input conditions.

Future Enhancements (Planned)

This is the first version of *Home Support: Your Daily Service Partner*. It has been built using Java Swing and MySQL for desktop use. Being a beginner-level project, the following improvements are planned in the future as part of learning and upskilling:

---

1. Admin Panel *(In Progress)*

The admin panel will manage:

- Adding/removing employees and service partners
- Viewing request summaries
- Tracking service performance and tasks

---

2. Service Partner Panel *(Planned)*

Each service partner will have a panel to:

- View assigned tasks
- Update task status (e.g., Completed)
- Track number of tasks and feedback

---

3. Improved UI and Validation

- Better form validation (e.g., phone, email check)
- More user-friendly layout using Swing components

---

4. Optional Web Version (Later Learning Goal)

After learning web development (HTML, CSS, JS), this system may be rebuilt as a web-based platform.

This project will continue to evolve as I learn more in my B.Tech journey. Every feature added is a step in my practical learning

Conclusion

The project “Home Support: Your Daily Service Partner” is a simple yet practical approach to understanding real-life service workflows using core Java and MySQL. Through this project, I explored how different modules—User, Company, Admin, and Service Partner—interact to manage daily household service requests.

Though this version focuses mainly on two working panels (User and Company), the structure and logic for Admin and Service Partner panels are already designed and planned for future development. This project has helped me understand not just GUI creation using Java Swing, but also how databases support backend logic.

It is still a beginner-level project, but it lays the foundation for building larger systems. I believe this version is just the beginning of what can be scaled further using better design, security, and frontend technologies in the future.

.

About the Developer

I am **Satyam Jain**, a passionate learner from Muzaffarnagar, Uttar Pradesh. I have recently completed my Class 12th and am about to begin my B.Tech journey. Even before stepping into college, I have always been deeply curious about how real-world problems can be solved using technology.

This project, *"Home Support: Your Daily Service Partner"*, is a reflection of that passion. It was designed and developed before the start of my B.Tech, purely driven by interest and inspiration. I have no formal industry experience yet, but I have a strong desire to learn and grow as a developer.

My journey into the world of programming began under the guidance of **Mr. Amit Bhatia**, Faculty of Computer Science at Holy Angels' Convent School, Muzaffarnagar and founder of ITSKILLS. His mentorship and way of teaching helped me fall in love with Java, and gave me the confidence to explore beyond textbooks.

Currently, I am focused on strengthening my core fundamentals in Java and backend development. My future goals include learning Data Structures, Web Development (HTML, CSS, JavaScript), App Development, and building stronger, full-stack real-world applications. I am also preparing myself to contribute meaningfully through internships, projects, and collaborations during my college years.

This project may be simple, but it is built with sincerity, sleepless effort, and a vision. I believe that it marks the first step towards something bigger.