

# **HOME SUPPORT:**

## **Your Daily Service Partner**



# **HOME SUPPORT**

## **Your Daily Service Partner**

**Created By:-** Satyam Jain

B.Tech (Information Technology)-1<sup>st</sup> 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. Over time, it has been expanded version by version to include more structured and professional features for real-life usability.

The project is modular in nature, and as of **Version 2.0**, the following panels are available:

1. **User Panel** – For service request management (raise, cancel, and track requests).
2. **Company Panel** – For employee-level request tracking, partner assignment, and status updates.
3. **Admin Panel (NEW in Version 2.0)** – For overall control and workforce management, including employees, service partners, requests, and administrative privileges.
4. **Service Partner Panel** – Planned for upcoming versions, where service partners will be able to view assigned tasks, update completion status, and access feedback.

This documentation outlines the motivation, structure, and functionality of the system in detail and serves as a complete guide to its development and current 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**.
- Provide **Admin-level control** through an **Admin Panel** for managing employees, service partners, requests, and administrative privileges.
- Keep the **Service Partner Panel** planned for upcoming versions to enable partners to view tasks, update completion, and access feedback.
- 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.
- Providing **Admin-level functionalities for overall control**, including employee management, service partner management, request handling, and administrative privileges. (*new scope*)
- 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 (fully functional in Version 2.0)**
  - **Service Partner Panel (planned for upcoming versions)**.

With Version 2.0, the system goes beyond request and partner assignment to include centralized administration. Future updates will focus on the Service Partner Panel, salary automation, feedback, and deeper 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 requests (Electrician, Plumber, Painter, Carpenter, Appliance Care).
- Cancel submitted requests.
- Track request status using Request ID.

## 2. Company Panel

- Secure login for company employees.
- View requests by date.
- Approve or reject service requests.
- Assign service partners to approved requests.
- Search and filter available service partners.

## 3. Admin Panel (NEW in Version 2.0)

- **Employee Block** – Add employee, remove employee, search employee, and view all employees.
- **Partner Block** – Add service partner, remove partner, search, filter, and view all service partners.
- **Request Block** – Search requests, filter requests, view request details, and generate count summaries.
- **Admin Block (Head Admin only)** – Add/remove admins, search admins, update admin details, view passwords, and update status (employees/partners/admins).

*(Service Partner Panel is in progress and will be added in future versions)*

# Tables Used in Version 2.0

## 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. 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

### 3. 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.

### 4. admin

Stores data of all admins including Head Admin. Used for admin login, privilege management, and overall control.

Field Name	Data Type	Description
adm_id	varchar(20)	Unique ID for the admin
adm_password	varchar(20)	Admin's login password
adm_name	varchar(30)	Full name of the admin
acno	varchar(13)	Admin's account/contact number
desig	varchar(20)	Designation of the admin

salary	decimal(10,0)	Salary of the admin
qualify	varchar(15)	Qualification of the admin
status	varchar(20)	Current status (Active/Inactive)
j_date	date	Joining date of the admin

## 5. employee

Stores details of company employees who use the Company Panel.

Field Name	Type	Description
emp_id	varchar(20)	Unique ID of the employee
emp_password	varchar(20)	Password for login
emp_name	varchar(30)	Full name
ecno	varchar(13)	Contact number
desig	varchar(20)	Designation (Executive/Manager)
salary	decimal(10,0)	Salary
qualify	varchar(15)	Qualification
j_date	date	Joining date
email_id	varchar(30)	Email of the employee
status	varchar(20)	Current status (Active/Inactive)

## 6.cancel\_request

Maintains record of cancelled service requests for history and auditing.

Field Name	Type	Description
r_id	varchar(15)	Unique request ID.
cname	varchar(30)	Customer name.
s_type	varchar(20)	Service type.
cancel_date	timestamp	Date/time when the request was cancelled.

## 7. remove\_mem

Keeps history of all remove members (Employee/ Partner/ Admin ).

<b>Field Name</b>	<b>Type</b>	<b>Description</b>
mem_id	varchar(15)	ID of removed member.
mem_name	varchar(25)	Name of removed member.
mem_cno	varchar(13)	Contact number.
mem_type	varchar(20)	Type (Employee/Partner/Admin).
mem_desig	varchar(20)	Designation.
admin_id	varchar(20)	ID of admin who removed the member.
verification	varchar(15)	Verification status (Done/Pending).
rem_date	date	Removal date.
auth_confirm	varchar(15)	Authority confirmation flag.
responsibility	varchar(15)	Responsibility handover info.
reason	varchar(100)	Reason for removal.

## 8. viewpassword

Used to log admin activities when viewing/updating member passwords.

<b>Field Name</b>	<b>Type</b>	<b>Description</b>
mem_type	varchar(18)	Member type (Employee/Partner/Admin).
mem_id	varchar(15)	Member ID.
mem_name	varchar(30)	Member name.
hadmin_id	varchar(15)	Head Admin ID.
hadmin_name	varchar(30)	Head Admin name.
v_date	datetime	Date/time of password view/update.
undertake	varchar(12)	Undertaking/confirmation field.

### ◆ Note

Tables **request**, **status**, **employee**, and **partner** were part of **Version 1.0** and continue to be used in **Version 2.0**. New tables

such as `admin`, `cancel_request`, `remove_mem`, and `viewpassword` have been introduced in Version 2.0 to support Admin Panel functionalities.

# 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)	Completed
v3.0	Service Partner Panel (Task View, Completion Update, Feedback Access)	Under <small>SOON</small> Development
v4.0	Web/App Version for Production Use	Planned <small>SOON</small>

## Note (Updated)

The current documentation includes **Version 1.0 & Version 2.0 features and database design**.

As the project progresses, upcoming versions will introduce additional panels and database tables such as:

- **feedback** – To store user feedback for service partners.
- **task\_summary** – To track partner ratings, tasks completed, tasks pending, etc.
- **salary** – Automated salary calculations for employees and partners.

All upcoming features and panel additions will be documented in subsequent versions, starting from Version 3.0 and onwards.

# Working Flow of the Application

## 1. User Panel Flow

- User launches the application and selects a service (Electrician, Plumber, Painter, Carpenter, Appliance Care).
  - 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 Request:** Removes records from both request and status tables using the Request ID.
    - **Check Status:** View current status by entering the Request ID (from status table).
- 

## 2. Company Panel Flow

- Company employee logs in using credentials from the employee table.
- Access to 4 main functionalities:
  1. **View Requests by Date:**
    - Enter a date to view all requests made on that day from the request table.
  2. **Update Request Status:**
    - Enter Request ID to see full details and Approve or Reject requests.
    - Status is updated in the status table along with the last updated date.
  3. **Search Service Partner:**

- Select criteria like Service Type, Appliance Type, Brand, etc.
- Matching service partners are fetched from the partner table.

#### **4. Assign Service Partner:**

- Enter Request ID, Service Partner ID, and assignment date.
  - Status table updated with assigned sp\_id and status becomes "Assigned".
  - Logout returns the employee to the login screen.
- 

### **3. Admin Panel Flow (NEW in Version 2.0)**

- **Admin Panel Login:** Any admin whose record exists in the admin table can log in.

#### **Blocks:**

##### **1. Employee Block:**

- **Add Employee:** Adds a new employee; details stored in employee table.
- **Remove Employee:** Deletes employee from employee table and logs removal in remove\_mem table.
- **Search Employee:** Search employees by ID, name, or other details.
- **View All Employees:** Displays all employee records.

##### **2. Partner Block:**

- **Add Service Partner:** Adds a new partner; records stored in partner table.
- **Remove Partner:** Deletes a partner from partner table and logs removal in remove\_mem.
- **Search Partner:** Search partners by ID, name, service type, or other filters.
- **Filter Partner:** Filter partners based on service type, availability, or other criteria.

- **View All Partners:** Displays all partner records.

### 3. Request Block:

- **Search Request:** Find specific requests using Request ID or customer details.
- **Filter Request:** Filter requests by service type, date, or status.
- **View Request Details:** Displays full request info including customer details and assigned partner.
- **Generate Count Summaries:** Reports showing total, completed, pending, or cancelled requests.

### 4. Admin Block (Head Admin only):

**Head Admin Login:** Only Head Admin can access the Admin Block

- **Add Admin:** Add a new admin account with credentials and privileges.
  - **Remove Admin:** Delete an admin account and log removal in remove\_mem.
  - **View Password:** View stored passwords; actions logged in viewpassword.
  - **Update Details:** Update admin details like name, contact, or designation.
  - **Search Admin:** Search admins by ID or name.
  - **Update Status:** Change Active/Inactive status of employees, partners, or admins.
- Logout returns admin to the login screen.

#### ◆ Notes:

- All transitions are controlled through **Swing GUI buttons**.
- All database operations are handled using **JDBC**.
- Exception handling and input validations are implemented to avoid crashes or incorrect 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 correct data insertion/deletion in request and status tables.
- Verified handling of invalid Request IDs gracefully.

### Company Panel:

- Verified employee login validation using correct and incorrect credentials.
- Checked View Request functionality with valid and invalid dates.
- Ensured request statuses update correctly in status table.
- Confirmed Search Service Partner filters work as expected.
- Verified Assign Service Partner updates status and assigned sp\_id correctly.

### Admin Panel (Version 2.0):

- **Employee Block:** Tested Add, Remove, Search, and View All Employees functionalities; verified proper insertion, deletion, and logging in remove\_mem.
- **Partner Block:** Tested Add, Remove, Search, Filter, and View All Partners; verified proper insertion, deletion, and logging in remove\_mem.

- **Request Block:** Tested Search, Filter, View Details, and Generate Count Summaries; verified data accuracy and summaries.
- **Admin Block (Head Admin only):** Tested Add/Remove Admin, View Password, Update Details, Search Admin, and Update Status functionalities; verified actions are logged correctly in viewpassword and remove\_mem tables.

## 2. Exception Handling

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

## 3. Manual Validation

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

## Conclusion

Version 2.0 of the application has passed all module-wise, exception handling, and manual validation tests. The system works efficiently for real-time usage under defined input conditions, with Admin Panel functionalities fully validated for employee, partner, request, and admin management.

# Future Enhancements (Planned)

The project “**Home Support: Your Daily Service Partner**” is designed to evolve incrementally, with future versions focusing on enhancing functionality, security, and accessibility. The roadmap for upcoming developments is as follows:

## 1. Service Partner Panel (Version 3.0)

- Introduction of a dedicated panel for service partners.
- Service partners will be able to:
  - View assigned tasks in detail.
  - Update task status (e.g., In Progress, Completed).
  - Access user feedback and performance summaries.

## 2. OTP-Based Verification

- Implementation of OTP (One-Time Password) authentication for secure login and verification.
- Ensures enhanced security for users, employees, and service partners.

## 3. Payment Integration

- Integration of digital payment options for service requests.
- Allows users to pay online conveniently and securely.
- Supports reliable payment processing and transaction tracking.

## 4. Android Application Version

- Development of a mobile application for Android devices.
- Provides real-time updates, notifications, and seamless access to all panels.
- Enhances user experience and broadens accessibility beyond desktop usage.

This structured roadmap ensures the project grows in a phased manner, with Version 3 focusing primarily on service partner functionalities, security, and transactions, followed by mobile accessibility and improved user experience.

# Conclusion

The project “**Home Support: Your Daily Service Partner**” provides a practical solution for managing daily household service requests using Core Java and MySQL. Through this project, I explored how different modules—User Panel, Company Panel, and Admin Panel—interact to manage requests efficiently, while laying the groundwork for the upcoming Service Partner Panel.

Key highlights of Version 2 include:

- Full implementation of User, Company, and Admin panels with role-based access.
- Structured workflows for request handling, partner assignment, and administrative management.
- Understanding GUI creation with Java Swing and backend logic supported by MySQL databases.
- Strong foundation for scalability, security, and future enhancements such as OTP verification, payment integration, and mobile application development.

Although still a beginner-level project, Version 2 establishes a solid framework for expanding into more complex, real-world service management systems in future iterations.

# About the Developer

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

This project, "**Home Support: Your Daily Service Partner**", reflects that curiosity and dedication. It was designed and developed independently before starting my B.Tech, driven purely by interest and inspiration. Though I have no formal industry experience yet, I am eager to learn and grow as a developer.

My programming journey began under the guidance of **Mr. Amit Bhatia**, Faculty of Computer Science at Holy Angels' Convent School, Muzaffarnagar, and founder of IT SKILLS. His mentorship nurtured my passion for Java, provided a solid understanding of core concepts, and encouraged me to explore beyond textbooks.

Currently, I am focused on strengthening my core fundamentals in Java and backend development, while planning future enhancements to this project such as:

- **Service Partner Panel** for task management and feedback
- **OTP verification and payment integration**
- **Android and mobile platform accessibility**

My broader goals include learning Data Structures, Web Development (HTML, CSS, JavaScript), App Development, and building full-stack, real-world applications. I aim to contribute meaningfully through projects, internships, and collaborations during my B.Tech journey.

This project may be simple, but it represents sincere effort, perseverance, and vision, marking the first step toward developing scalable and impactful applications.