

SMART SERVICE MANAGEMENT APPLICATION

Elango S¹, Sakthivel N²

¹ Student, Master Of Computer Application, Adhiyamaan College Of Engineering, Hosur, Tamil Nadu, India.

² Professor, Master Of Computer Application, Adhiyamaan College Of Engineering, Tamil Nadu, India.

ABSTRACT

The Smart Service Management App is a mobile and web-based application designed to streamline service requests, technician assignments, and job tracking. It provides an efficient and user-friendly platform for customers to book services, track job progress, and receive real-time notifications. Technicians can manage assigned tasks, update job statuses, and receive payments, while administrators oversee the process, optimize resource allocation, and generate performance reports. By integrating modern technology, the app ensures a seamless and organized service management experience for all users.

Keywords: Mobile Application, Service Management, Technician Assignment, Job Tracking, Push Notifications, Performance Analytics.

1. INTRODUCTION

Service management has evolved with the use of technology, increasing efficiency and customer convenience. The Smart Service Management App automates service requests, technician assignments, and job tracking, reducing manual effort and optimizing workflow. It features an interactive user interface, real-time notifications, and a data-driven approach to performance tracking. Built with React Native for the frontend, ASP.NET for the backend, and MS SQL for the database, the app ensures reliability, scalability, and seamless user experience.

2. METHODOLOGY

This study focuses on developing an automated service management system with structured modules for service booking, technician assignments, real-time tracking, notifications, and analytics.

2.1 Lead Registration- The Lead Registration process allows customers to submit service requests by specifying the urgency and service type. The system records these details to prioritize jobs efficiently and assign the appropriate technician. This ensures a structured workflow where service requests are categorized based on importance and availability.

2.2 Technician Assignment- The Technician Assignment module automatically assigns technicians based on their expertise, location, and availability. Admins have the flexibility to manually reassign jobs if required to balance workloads. This method enhances efficiency by ensuring that the most suitable technician is allocated to each job.

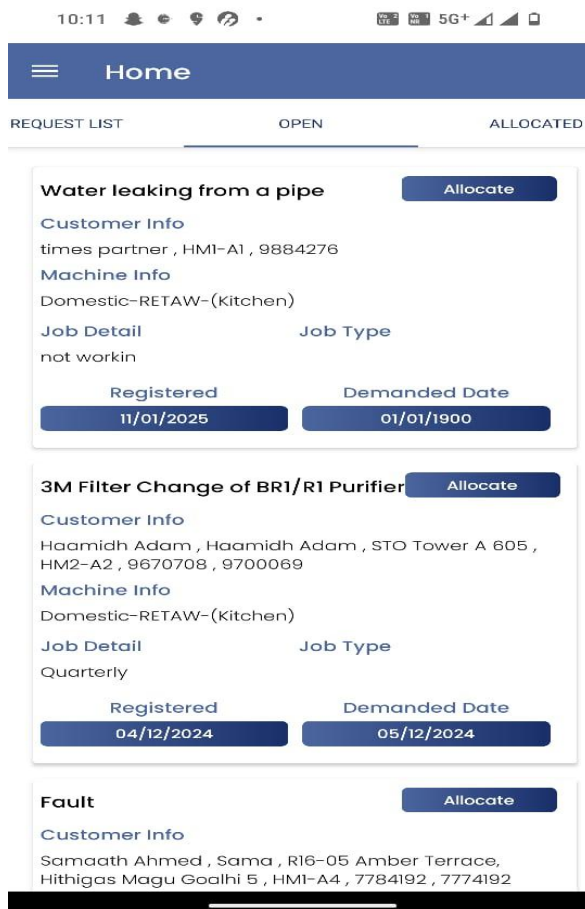
2.3 Service Scheduling- The Service Scheduling feature enables customers and technicians to coordinate appointments through an integrated calendar. Automated reminders notify both parties of scheduled services, reducing missed appointments. This ensures timely service delivery and a seamless experience for all users.

2.4 Job Tracking- The Job Tracking system allows technicians to update job statuses in real-time, such as Pending, In Progress, or Completed. Customers receive live updates and can track the technician's progress through the app. This transparency improves customer trust and satisfaction by keeping them informed throughout the service process.



Chart -1: Flow chart

3. RESULT



The screenshot displays the 'Home' screen of the Smart Service Management App. At the top, there is a navigation bar with a hamburger menu icon and the word 'Home'. Below the navigation bar, there are three tabs: 'REQUEST LIST', 'OPEN', and 'ALLOCATED'. The 'REQUEST LIST' tab is currently selected. The main content area shows a list of service requests. Each request card includes a title, customer information, machine information, job details, and a date. The first request is titled 'Water leaking from a pipe' and has a status of 'not workin'. The second request is titled '3M Filter Change of BR1/R1 Purifier' and has a status of 'Quarterly'. The third request is titled 'Fault' and has a status of 'Quarterly'. Each card has an 'Allocate' button in the top right corner.

4. CONCLUSION

The Smart Service Management App enhances service efficiency by automating the process of lead registration, technician assignment, service scheduling, and job tracking. It provides a seamless experience for customers, technicians, and administrators through real-time updates and notifications. The integration of React Native, ASP.NET, and MS SQL ensures a scalable, reliable, and user-friendly system.

With features like automated scheduling, live tracking, and performance analytics, the app optimizes resource allocation and improves service response times. The inclusion of push notifications and reports ensures timely updates and better decision-making. Future enhancements such as AI-based technician assignment, IoT integration, and chat support will further refine the system and improve user experience.

ACKNOWLEDGEMENTS

This journal paper was truly prepared by me, and I agree to the terms and conditions.

5. REFERENCES

- [1] Microsoft Documentation – ASP.NET Web API Guide. Retrieved from <https://learn.microsoft.com/en-us/aspnet/core/web-api>
- [2] React Native Documentation – Building Mobile Apps with React Native. Retrieved from <https://reactnative.dev/docs/getting-started>
- [3] MS SQL Server Documentation – Database Management and Optimization. Retrieved from <https://learn.microsoft.com/en-us/sql/sql-server>
- [4] Quartz.NET Documentation – Job Scheduling in .NET Applications. Retrieved from <https://www.quartz-scheduler.net/>
- [5] Firebase Documentation – Push Notifications and Real-Time Database Implementation. Retrieved from <https://firebase.google.com/docs>
- [6] Fielding, R. T. (2000) – Architectural Styles and the Design of Network-based Software Architectures (REST APIs). University of California, Irvine.