SYNOPSIS

MCA
Semester III

Submitted by

SAJAL VERMA
Batch Year : 2024-2026
Enrolment No: U2449017

Project Guide – Ms. SUNITA TRIPATHI

Center of Computer Education & Training
Institute of Professional Studies
University of Allahabad, Prayagraj
Uttar Pradesh - 211002

Table of Content

1 INTRODUCTION

Project Name: OWMO

Category: Full Stack

OWMO (One World Mobile Operation) is a smart and centralised platform designed to manage mobile repairing and diagnostics in an efficient, transparent and systematic way. It brings together users, technicians and administrators on a single digital system to ensure smooth operations.

Currently, traditional mobile repair services suffer from major problems such as lack of transparency, poor service tracking, manual record-keeping, communication gaps and customer dissatisfaction. Technicians often struggle with unclear job assignments, delayed updates and no proper tool to manage their work efficiently. Customers on the other hand, face uncertainty about repair status, costs and service reliability.

OWMO helps solve these challenges by offering a structured, role-based system. Technicians can easily receive and update jobs, upload diagnostic reports, and track their performance. Customers gain real-time visibility into the repair process, while administrators can monitor operations, manage technicians, and maintain inventory. This makes OWMO a scalable, reliable, and transparent solution for modern mobile repairing services.

1.2 PROBLEM DEFINITION

The mobile repair industry is rapidly growing, but traditional repair and management systems still rely on manual processes that create inefficiency and customer dissatisfaction. Customers often face difficulties such as unclear repair timelines, hidden charges and lack of real-time updates, while technicians struggle with unorganised job handling and poor communication. Administrators, on the other hand, lack proper tools to monitor operations, assign jobs and track inventory effectively

Key Challenges

1. **Lack of Transparency:** In traditional mobile repair services, customers often have no clear visibility of the repair process. They hand over their devices and are left uncertain about the status, progress and expected delivery time. This creates a communication gap and reduces customer trust in the service provider.
2. **Unorganised Technician Workflow:** Technicians usually manage jobs manually, without a proper system for assignment, status updates or repair logs. This unorganised approach leads to confusion, delays in service completion and difficulty in tracking technician performance or accountability.
3. **Poor Communication Between Stakeholders:** The absence of a centralised platform creates barriers between customers, technicians and administrators. Customers cannot directly track updates, technicians struggle to convey repair status and administrators face delays in monitoring jobs and resolving issues.
4. **Inefficient Record and Inventory Management:** Managing invoices, spare parts, accessories and repair history manually often results in data loss, duplication or errors. This not only increases operational workload but also affects service quality and business growth.
5. **Customer Dissatisfaction and Low Trust:** Due to delays, hidden costs and uncertainty in service reliability, customers often feel dissatisfied. The lack of OTP verification or real-time tracking further reduces their confidence in traditional mobile repair services.

1.3 MOTIVATION

The idea of OWMO (Online Workshop for Mobile Operations) came from real-life problems faced in mobile repair services. Customers often feel stressed due to hidden charges, no real-time updates, data security issues and delayed repairs. Technicians, too, struggle with unorganised workflows and lack of proper tools.

Experiencing these challenges ourselves, we felt the need for a transparent, reliable, and efficient system where customers can track repairs in real time, verify jobs through OTP and trust the service. OWMO was created to solve these problems and bring clarity, trust and efficiency to the mobile repairing industry.

2 OBJECTIVE

The objective of the Owmo project is to create a smart, transparent and efficient mobile repairing management system that connects users, technicians and administrators on a single platform. It focuses on simplifying repair requests, ensuring trust through real-time updates and OTP verification, automating workflows for faster service, integrating inventory and accessory management with secure payments and providing AI-powered diagnostics and customer support.

3 REQUIREMENT ANALYSIS AND SPECIFICATION

3.1 Functional Requirements

1. **User Profile Management:**
   i. Allow customers to create and manage their profiles.
   ii. Store repair history, invoices and contact details securely.

2. **Technician Module:**
   i. View and accept assigned repair requests.
   ii. Update repair progress with images and status.

3. **Service Request Management:**
   i. Let users raise repair requests by selecting brand, model and issue.
   ii. Provide cost estimation and OTP-based confirmation.

4. **Admin Panel:**
   i. Manage users, technicians and service requests.
   ii. Monitor progress and verify service completion.

5. **Authentication & Security:**
   i. Implement JWT-based login with role-based access.
   ii. Secure all sensitive data with encryption and OTP verification.

6. **Inventory & Accessories:**
   i. Allow admins to manage spare parts and accessories.
   ii. Enable users to purchase items with online payment.

7. **Customer Support & Feedback:**
   i. Provide ticket-based support and live chat.
   ii. Collect ratings and reviews after service completion.

3.2 Non Functional Requirements

1. **Performance:** The system should handle multiple service requests simultaneously and deliver real-time updates within seconds to ensure a smooth user experience.
2. **Usability:** The interface must be simple, responsive and easy to navigate for users, technicians and administrators across web and mobile platforms.
3. **Reliability:** Ensure high system uptime with secure backup and recovery mechanisms to avoid data loss and service disruption.
4. **Scalability:** The system should support future growth, handling an increasing number of users, technicians and transactions without performance degradation.
5. **Security:** Provide strong data encryption, secure authentication (JWT + OTP) and role-based access to ensure privacy and data protection.

Technical Requirements

1. **Programming Languages:** JavaScript (Node.js and Express for backend, React.js for frontend).
2. **Frameworks:**
    - **Frontend:** React + Tailwind CSS for a responsive UI.
    - **Backend:** Node.js and Express with REST APIs.
3. **Database:** MongoDB Atlas for secure storage of user profiles, repair requests and inventory.
4. **Authentication & Security:** JWT for login sessions, OTP-based verification, bcrypt for password hashing.
5. **AI & Automation:** An AI-powered diagnostic module for initial issue detection based on user input, implemented using the Naive Bayes classifier (Natural library) in Express.js.
6. **Cloud Deployment:** Vercel for frontend hosting, Render for backend and cloud storage (Cloudinary) for images.
7. **Payment Gateway:** Integration with Razorpay for secure online transactions.
8. **Analytics & Reporting:** Rechart for admin dashboard insights into repairs, earnings and performance.
9. **Third-Party Services:**
    - Twilio for SMS & OTP delivery.
    - Nodemailer for email notifications.
    - Socket.io for live technician interviews.

3.3 Software & Hardware Requirements

1. Hardware Requirements

Name of Components	Specifications
Processor	Intel Core i3 or higher
RAM	4GB or higher
Storage	256GB HDD or SSD
Operating System	Windows 10 / 11

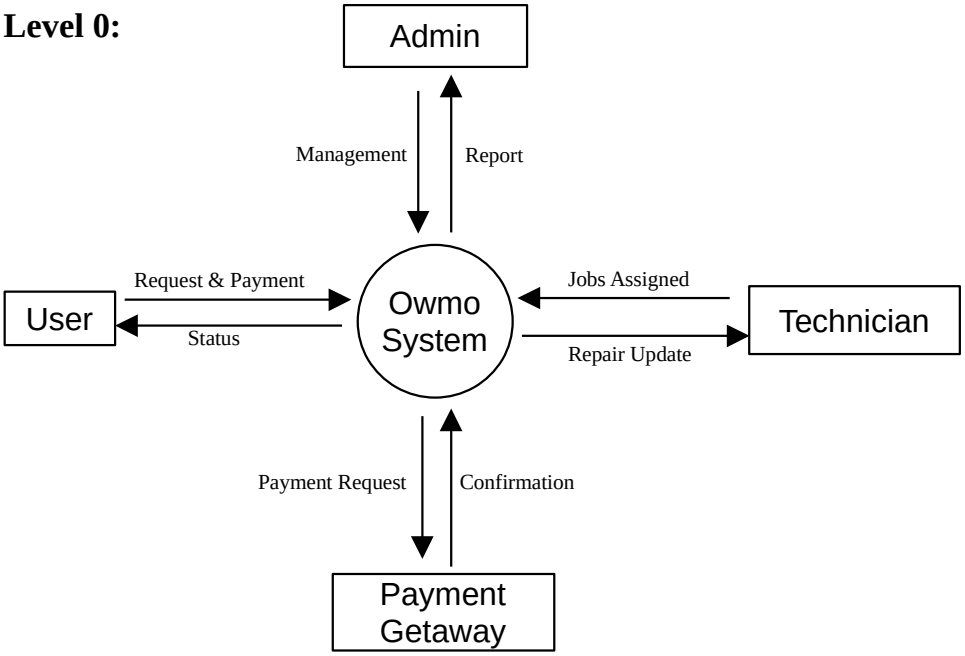
2. Software Requirements

Name of Components	Specifications
Frontend	React.js + Tailwind CSS
Backend	Node.js with Express.js
Database	MongoDB Atlas
Cloud Hosting	Vercel (Frontend), Render (Backend)
File Storage	Cloudinary and Multer
Payment Gateway	Razorpay
Communication APIs	Twilio for OTP & SMS, Nodemailer

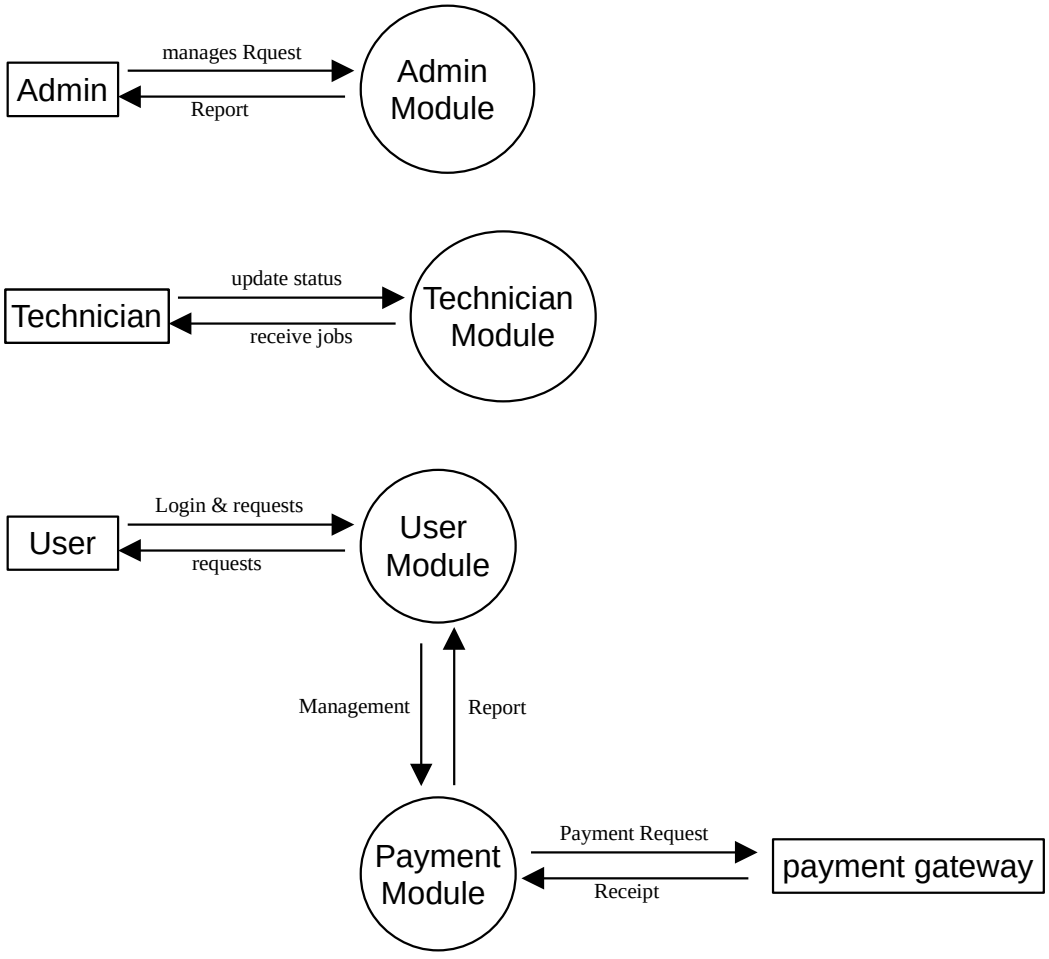
4 SYSTEM DESIGN

System Design of Owmo Using the Data Flow Diagram **(DFD)**

Level 0:



Level 1:



5 ALGORITHMS USED

For the Owmo project, several algorithms and techniques are applied to ensure secure authentication, efficient service management, intelligent diagnostics and safe transaction processing. Some of the key approaches are:

1. **CRUD Algorithms:** Manage Create, Read, Update and Delete operations for handling user profiles, repair requests and inventory records, ensuring reliable and consistent data management.

2. **User Authentication Algorithms:** Verify user identity using OTP verification, JWT token-based sessions and password hashing with algorithms like bcrypt, ensuring secure login and role-based access control.

3. **AI-Based Diagnostic Algorithms:** AI/ML techniques to analyze user-described problems and suggest possible mobile issues (e.g., screen crack → LCD replacement), improving diagnostic accuracy.

4. **Search and Matching Algorithms:** Match users with available technicians based on skills, workload or location, ensuring efficient allocation of repair requests.

5. **Data Encryption and Security Algorithms:** Secure sensitive information such as login credentials, invoices and transaction details using encryption methods like AES and secure transmission via SSL/TLS, preventing unauthorized access.

6. **Payment Processing Algorithms:** Validate online transactions using checksum verification and encryption during integration with payment gateways like Razorpay, ensuring safe and reliable financial transactions

6 MILESTONES

Sr. No.	Project Activity	Estimated Start Date	Estimated End Date
1.	Project Allotment	10/08/2025	20/08/2025
2.	Synopsis Creation	21/08/2025	29/08/2025
3.	Project Synopsis Presentation	30/08/2025	05/09/2025
4.	Frontend Development	06/09/2025	29/09/2025
5.	Backend Development	30/09/2025	28/10/2025
6.	Integration and Testing	29/10/2025	05/11/2025
7.	Documentation	06/11/2025	30/11/2025
8.	Final Presentation	01/12/2025	10/12/2025

7 MEETING WITH SUPERVISOR

Date of the Meeting	Mode	Comments by the Supervisor	Signature of the Supervisor
10/08/2025	Offline	Discussed about the topic of the project	
11/08/2025	Offline	Finalised project	
29/08/2025	Offline	Discussed review paper	
08/09/2025	Offline	Synopsis review	

8 REFERENCES

1. W3Schools. *ReactJS Tutorial*. Available at: https://www.w3schools.com/react

2. YouTube. *Express.js Tutorials*. Available at: https://www.youtube.com

3. YouTube. *MongoDB Tutorials*. Available at: https://www.youtube.com

4. Tailwind Labs. *Tailwind CSS Documentation*. Available at: https://tailwindcss.com/docs

5. Natural Node. *Natural – NLP for Node.js Documentation*. Available at: https://github.com/NaturalNode/natural

6. npm. *Twilio Node.js SDK*. Available at: https://www.npmjs.com/package/twilio

7. npm. *Nodemailer – Node.js Email Library*. Available at: https://www.npmjs.com/package/nodemailer

8. YouTube. *Socket.io Tutorials*. Available at: https://www.youtube.com

9. npm. *Cloudinary Node.js SDK*. Available at: https://www.npmjs.com/package/cloudinary

10. Recharts Community. *Recharts Documentation*. Available at: https://recharts.org/en-US