KLE Technological University
Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

CENTER FOR AUTOMATION AND ROBOTICS RESEARCH

INSTITUTE SPONSERED PROJECT

Project report on

Design and Implementation of User Interfaces for Monitoring and control of Smart Manufacturing Lab

Submitted

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Gauri Thambkar	01FE21BCS098
Aishwarya Patil	01FE21BCS112

Under the guidance of

Mr. Shridhar Dodamani and Dr. Sachin Karadgi

CENTER FOR AUTOMATION AND ROBOTICS RESEARCH

KLE Technological University, Hubballi

2024-2025

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

CENTER FOR AUTOMATION AND ROBOTICS RESEARCH

2024-25

CERTIFICATE

This is to certify that project entitled Design and Implementation of User Interfaces for Monitoring and control of Smart Manufacturing Lab is a bonafied work carried out by the student team Gauri Thambkar 01FE21BCS098, Aishwarya Patil 01FE21BCS112, in partial fulfillment of the completion of 7th semester B. E. course during the year 2024 – 2025. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course

Guide Name	Dr. Sachin Karadgi.
Mr. Shridhar Dodamani	Director CARR

External Viva-Voce

Name of the examiners	Signature with date
1 ________________________	________________
2 ________________________	________________

PREAMBLE

The rise of Industry 4.0 has revolutionized manufacturing processes by integrating advanced technologies such as automation, artificial intelligence, and the Internet of Things (IoT). These advancements have paved the way for the development of smart manufacturing labs, where precision, efficiency, and innovation drive operations. A cornerstone of these labs is the Automatic Storage and Retrieval System (ASRS), SCARA, and multiple robots which automate the placement and retrieval of goods with minimal human intervention. The system integrates multiple robotic units to streamline operations within the smart manufacturing lab. These robots, working in tandem, enhance operational precision, reduce human intervention, and optimize overall productivity.

Given the critical role of automation in smart manufacturing, the need for an intuitive and effective user interface for monitoring and control becomes evident. Such an interface should provide real-time tracking and ensure seamless interaction between operators and the system. Additionally, it must offer scalability and adaptability to meet evolving operational demands.

Our project addresses these challenges by developing a centralized system to monitor and control machines in real-time, with a particular focus on ASRS. The system features a user-friendly Graphical User Interface (GUI) tailored for smart manufacturing operations, ensuring operators can efficiently manage processes, respond to system alerts, and access actionable insights. The GUI is complemented by a robust, scalable database for real-time data storage and retrieval, empowering decision-makers with accurate and timely information.

Furthermore, the project incorporates role-based access control to ensure security and operational efficiency. Operators, administrators, and managers have access to functionalities relevant to their roles, fostering a secure and collaborative environment. Advanced data visualization tools and automated alerts enhance the system's capabilities, allowing users to analyze historical and real-time data and quickly address any anomalies.

By streamlining operations, enhancing decision-making, and promoting real-time control, this project significantly improves the functionality and efficiency of smart manufacturing labs. The integration of these features represents a vital step forward in adopting Industry 4.0 principles, contributing to a smarter, more productive, and innovative manufacturing environment.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of a number of individuals whose professional guidance and encouragement helped me in the successful completion of this report work.

We take this opportunity to thank Dr. Ashok Shettar, Pro-Chancellor, Dr. P. G. Tewari, Vice-Chancellor and to Dr. Basavaraj Anami, Registrar, KLE Technological University, Hubballi.

We also take this opportunity to thank Dr. B. B. Kottarshettar, Dean (Applied Materials), and Dr. Sachin Karadgi, Professor and Head, Automation and Robotics for having provided us academic environment which nurtured our practical skills contributing to the success of our project.

We sincerely thank our guide Mr. Shridhar Dodamani, Department of Automation and Robotics for his guidance, inspiration and wholehearted co-operation during the course of completion.

We sincerely thank our project co-ordinators Mr. Amogh Sutar for their support, inspiration and wholehearted co-operation during the course of completion.

Our gratitude will not be complete without thanking the Almighty God, our beloved parents, our seniors and our friends who have been a constant source of blessings and aspirations.

Gauri Thambkar - 01FE21BCS098
Aishwarya Patil - 01FE21BCS112

CONTENTS

PREAMBLE **i**

ACKNOWLEDGEMENT **i**

CONTENTS **iii**

LIST OF FIGURES **iv**

1 INTRODUCTION **1**

1.1 Motivation 2
1.2 Problem Statement 2
1.3 Assembly Line Analysis 3
1.4 Objectives 4
1.5 Scope of the project 4

2 REQUIREMENT ANALYSIS **6**

2.1 Functional Requirements 6
2.2 Non Functional Requirements 7
2.3 Hardware Requirements 8
2.4 Software Requirements 9

3 SYSTEM DESIGN **10**

3.1 Architectural Framework- MVC Architecture 10
3.2 MERN Framework 11
3.3 System Design 11
3.4 System Workflow Representation 13
3.4.1 Use Case Diagram 13
3.4.2 User Sequence Diagram Representation 13
3.4.3 Operator Sequence Diagram Representation 14

4 IMPLEMENTATION **19**

4.1 Database Design 19
4.1.1 Database Design for Customer Schema 19
4.1.2 Database Design for Operator Schema 21
4.2 Interface Designed 23

5 RESULTS AND DISCUSSIONS **31**

6 CONCLUSION AND FUTURE SCOPE OF THE WORK **32**

LIST OF FIGURES

1.1 Assembly Line. 1
1.2 Assembly Line Workflow 3

3.1 Client-Server Architecture 10
3.2 Working of the System 15
3.3 Use Case Diagram 16
3.4 User Sequence Diagram 17
3.5 Operator Sequence Diagram 18

4.1 User Schema Diagram 20
4.2 Operator Schema Diagram 22
4.3 Login Page 24
4.4 Customer Home Page 25
4.5 Add Pattern Page 25
4.6 Customer Order Status Page 26
4.7 Operator Home Page 26
4.8 Operator New Orders Page 27
4.9 Operator Status Page 27
4.10 ASRS Status Page 28
4.11 ASRS Rack Status Page 28
4.12 ASRS Rack Edit Page 29
4.13 Cobot Status Page 29
4.14 Production History Page 30

Chapter 1

INTRODUCTION

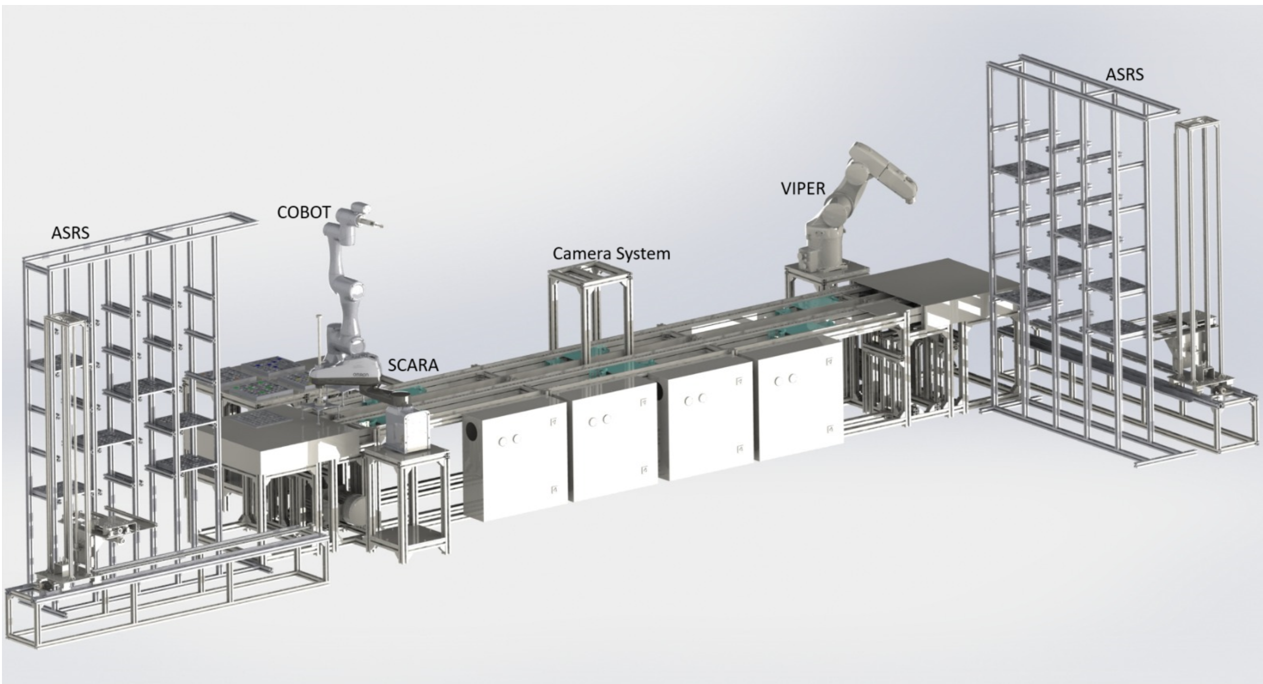

Figure 1.1: Assembly Line.

Industry 4.0 has transformed traditional manufacturing processes by integrating advanced technologies such as automation, artificial intelligence, and the Internet of Things (IoT). Smart manufacturing labs embody these innovations, fostering precision, efficiency, and real-time control. At the heart of these labs is the Automatic Storage and Retrieval System (ASRS), a critical component designed to automate the handling of materials and optimize space utilization with minimal human intervention. However, the effective functioning of such labs requires seamless integration between ASRS, robotic systems, and quality assurance mechanisms to ensure smooth workflows and operational accuracy.

This project aims to develop a centralized system that enables real-time monitoring, control, and data management of these interconnected components. At its core, the system features a robust, scalable database and a user-friendly Graphical User Interface (GUI) to enhance usability and streamline processes. The ASRS works with various robots to handle tasks such as material retrieval, quality inspection, and finished goods storage. Robot 2 (SCARA) plays a crucial role in material handling, retrieving empty pallets from the ASRS and placing them onto the conveyor system to facilitate downstream processes. Robot 3 (Cobot), equipped with a vision system, inspects items for quality and conformity, ensuring they meet required specifications before arranging them on the pallet. A camera system further verifies pallet patterns, sending correctly arranged pallets for storage and redirecting faulty ones for recirculation. Once the quality inspection is complete, Robot 4 (VIPER) transfers the verified pallets back to the ASRS, storing them in designated racks for finished goods. Together, these components form a cohesive framework that improves efficiency, reduces human intervention, and ensures adaptability in a fast-paced industrial environment.

1.1 Motivation

The growing complexity of manufacturing processes, driven by increased demand for precision and efficiency, highlights the need for smarter and more adaptable systems. Traditional methods of material handling and quality control often result in inefficiencies, errors, and delays due to their dependence on manual operations. In the context of smart manufacturing labs, this creates a strong motivation to leverage advanced technologies to address these challenges. The integration of automated systems such as ASRS and robots not only reduces human workload but also enhances operational accuracy and consistency.

This project is motivated by the need to create a unified, intelligent system capable of managing and monitoring the various components of a smart manufacturing lab in real-time. By incorporating robotics, vision systems, and centralized control mechanisms, the system provides significant advantages, such as enhanced material flow, improved quality assurance, and seamless decision-making. Moreover, the role-based access and real-time data storage features contribute to better accountability and operational transparency. Ultimately, this project seeks to advance the principles of Industry 4.0 by designing a framework that ensures higher productivity, optimized resource usage, and adaptability to evolving industrial requirements.

1.2 Problem Statement

The smart manufacturing lab requires a centralized system to monitor and control machines in real-time. The lack of an integrated platform makes it challenging for operators to manage processes, respond to system alerts, and analyze data effectively. The project aims to develop a GUI and a robust, scalable database for control, data storage, and real-time monitoring. This system will streamline operations, enhance decision-making, and provide role-based access.

1.3 Assembly Line Analysis

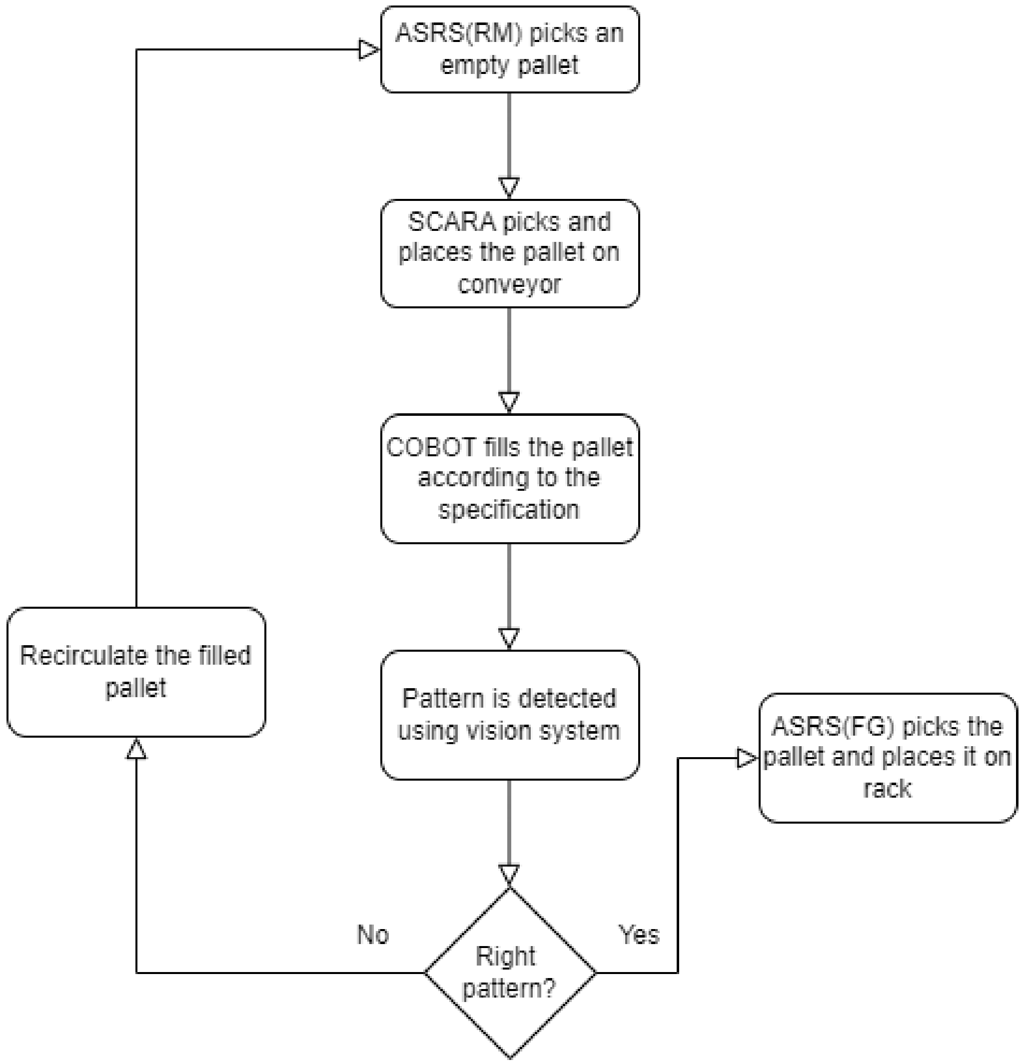

Figure 1.2: Assembly Line Workflow

- ASRS(RM) Picks an Empty Pallet: The Automated Storage and Retrieval System (ASRS) for raw materials initiates the workflow by selecting an empty pallet. This step ensures that the process begins with a ready-to-use pallet, reducing manual intervention.
- SCARA Picks and Places the Pallet on the Conveyor: A SCARA (Selective Compliance Assembly Robot Arm) robot transfers the empty pallet to the conveyor system. This automation step streamlines the material flow, enabling a consistent and efficient transition to the filling phase.

- COBOT Fills the Pallet According to Specifications: A collaborative robot (COBOT) accurately fills the pallet based on predefined specifications. The COBOT's precision ensures the payload meets required standards, reducing errors in material handling.
- Pattern Detection Using Vision System: The vision system inspects the pallet to verify the pattern of the items. This step introduces quality control by ensuring the items are correctly arranged and meet specifications.
- Decision Point - Is It the Right Pattern? Based on the vision system's analysis, a decision is made: Yes: If the pattern is correct, the pallet moves to the next stage. No: If the pattern is incorrect, the pallet is recirculated for reprocessing.
- ASRS(FG) Picks and Places the Pallet on the Rack: The ASRS for finished goods (FG) retrieves the correctly filled pallet and places it in the designated storage rack. This step completes the process by ensuring the finished product is ready for dispatch or further processing.

1.4 Objectives

- Develop Intuitive and User-Friendly GUIs
- Integrate Robust Database Systems
- Enable Seamless Communication with Industrial Equipment
- Implement Security and Access Control
- Ensure Real-Time Monitoring and Control

1.5 Scope of the project

The project focuses on the development of a centralized system for smart manufacturing labs, aimed at achieving real-time monitoring, control, and data management. This system will incorporate advanced features such as user-friendly Graphical User Interfaces (GUIs), a robust and scalable database, and secure communication protocols to manage and streamline various industrial operations.

The scope includes designing and implementing intuitive GUIs that provide operators with real-time insights into machine operations, process status, and quality assurance results. These GUIs will interact seamlessly with a robust database to ensure consistent data storage, retrieval, and analysis, facilitating effective decision-making and operational tracking.

Integration with industrial equipment is a key aspect of the project. The system will interface with Automatic Storage and Retrieval Systems (ASRS), multiple robotic units, vision systems, and quality inspection cameras to enable synchronized control of workflows. The design will also support automated data updates and notifications, minimizing manual intervention and reducing errors.

Additionally, the project includes implementing security measures, such as role-based access control, to ensure data integrity and restrict unauthorized access to critical functions. The system will be modular and scalable, allowing for future expansion or upgrades as new technologies and equipment are integrated into the manufacturing process.

In summary, the project is scoped to deliver a fully operational, secure, and efficient control system tailored to the needs of a smart manufacturing lab, ensuring adaptability and enhanced performance across industrial environments.

Chapter 2

REQUIREMENT ANALYSIS

The following outlines the functional, non-functional, hardware, and software requirements for the project, aimed at developing a comprehensive system for real-time monitoring and management of a smart manufacturing lab. The system is designed to support two key roles—User and Operator—with distinct responsibilities, such as adding patterns, placing orders, and tracking production. The functional requirements include seamless integration between the frontend and backend, real-time updates on order and production status, and a robust database to manage and store data. Non-functional requirements focus on ensuring high performance, security, scalability, and maintainability. Additionally, the system will require specific hardware for hosting the database and supporting frontend interactions, along with software tools including React, Node.js, and MySQL to build a secure, user-friendly, and efficient platform.

2.1 Functional Requirements

- User Role Management:
  - Users can add new patterns to the system.
  - Users can select from their existing patterns.
  - Users can place orders based on selected patterns.
  - Users can track the status of their orders in real-time.
- Operator Role Management:
  - Operators can view new, pending, and completed orders on a dashboard.
  - Operators can analyze business insights using graphical reports.
  - Operators can start and stop production on selected orders.
  - Operators can pause and resume production as needed.
  - Operators can monitor machine status in real-time.
  - Operators can view the production history and machine history.
  - Operators can dispatch completed orders and view dispatch details.

- Real-Time Data Interaction:
  - Real-time tracking of order status and machine data.
  - Real-time updates on order production status and progress.
  - Notification alerts for key events (order completion, production updates).
- Database Management:
  - Store user, pattern, and order data efficiently.
  - Update order status, production data, and machine history dynamically.
  - Ensure that data is synchronized between the frontend and backend.
- Role-Based Access Control:
  - Different user roles (User, Operator) with tailored access to features.
  - Admin-level features for managing users, roles, and permissions.

2.2 Non Functional Requirements

- Performance:
  - The system should be able to handle multiple simultaneous users without significant latency or downtime.
  - Real-time updates on order and machine status should occur with minimal delay.
- Scalability:
  - The system should be scalable to accommodate more users, orders, and machines as the manufacturing operation grows.
  - It should support the addition of new features or modules with minimal rework.
- Security:
  - User data and order information should be stored securely.
  - Implement SSL/TLS encryption for secure data transmission.
  - Role-based access control should be strictly enforced to protect sensitive data.
  - Prevent unauthorized access to sensitive functions, such as starting/stopping production and dispatching orders.
- Usability:

  - The front end (React) should be intuitive and easy to navigate.
  - The system should provide clear, actionable feedback to users when interacting with the system (e.g., order updates, and production statuses).
  - The system should include user-friendly forms for adding patterns, placing orders, and managing production.
- Availability and Reliability:
  - The system should have high availability, ensuring minimal downtime during critical operations.
  - Backup and recovery mechanisms should be in place for the database to prevent data loss.
- Maintainability:
  - The codebase should be modular and well-documented to ensure easy maintenance and updates.
  - The system should be designed with easy-to-integrate components for future improvements or modifications.

2.3 Hardware Requirements

- Server:
  - A server (local or cloud) capable of running MySQL and Node.js for backend operations.
  - Adequate storage for databases and logs.
  - Sufficient processing power and RAM to handle multiple concurrent users and real-time data updates.
- Client Machines:
  - Devices with internet access (PCs, laptops, or tablets) capable of running a web browser to access the front end (React).
  - Devices should have modern web browsers (Chrome, Firefox, Safari, etc.) for optimal performance.
- Networking:

- A stable and secure internet connection to ensure real-time communication between the frontend, backend, and MySQL database.
- A local network for any internal devices like machines or other industrial systems if required.

2.4 Software Requirements

- Frontend:
  - React: Used for building the user interface, providing dynamic and responsive views for users and operators.
  - Firebase: Hosting the front end to ensure scalability and easy deployment.
- Backend:
  - Node.js: Used for building the server-side logic, APIs, and connecting the frontend to the database.
  - Express.js: A web application framework for Node.js that helps handle routing and middleware.
- Database:
  - MySQL: The relational database management system (RDBMS) used for storing user data, order details, and production information. The database is hosted locally.
- Development Tools:
  - Visual Studio Code or other code editors for development.
  - Postman for API testing.
  - Git for version control.
- Security and Encryption:
  - SSL/TLS encryption: For secure communication between frontend and backend.
  - OAuth for secure authentication and role-based access control.
- Miscellaneous Tools:
  - Charting Libraries (e.g., Drraw.io) for generating business insights and graphs for the operator dashboard.

Chapter 3

SYSTEM DESIGN

3.1 Architectural Framework- MVC Architecture

The client-server architecture is highly suitable for the Result Analysis web application, providing a robust and scalable framework to meet the specific requirements of managing student grades, generating result summaries, and visualizing performance trends.

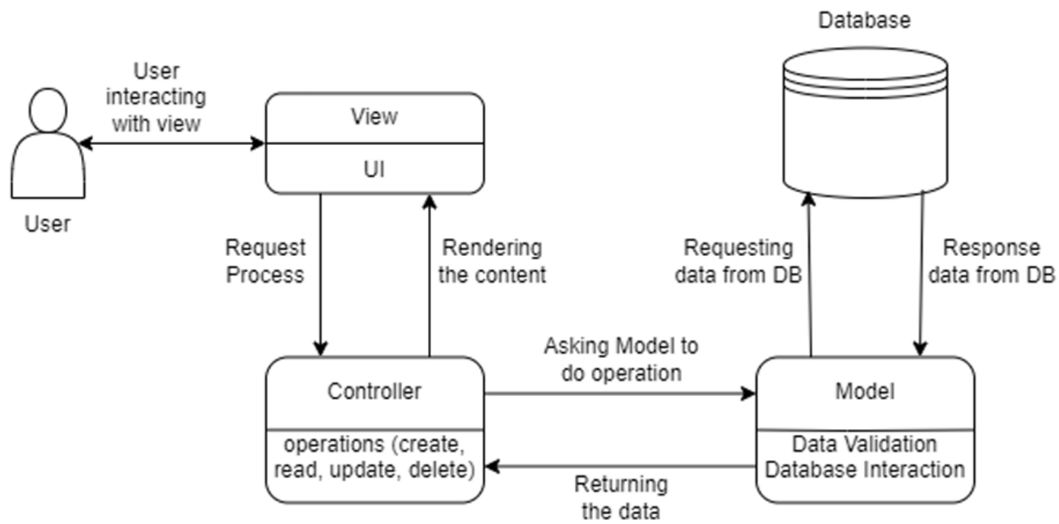

Figure 3.1: Client-Server Architecture

Advantages:

- Separation of concerns: Clear distinction between frontend and backend functionalities.
- Scalability: Components can be scaled independently based on demand.
- Security: Critical operations, such as authentication and database access, are centralized and controlled on the server side.
- Maintainability: Easier to update and maintain different components independently.

3.2 MERN Framework

The MERN stack is a popular JavaScript-based framework for building web applications, consisting of four main components: MySQL, Express.js, React, and Node.js. Each component plays a specific role in the development process, collectively providing a full-stack solution. Let's explore each component in the context of the Result Analysis web application:

- MySQL: MySQL is used for storing structured customer information, orders, and operators' production history with their history of orders. Its relational database model ensures data integrity and consistency, making it ideal for managing complex queries and relationships. MySQL's scalability supports future enhancements.

- Express.js: Express.js serves as the backend framework, handling HTTP requests from the client, routing those requests to the appropriate endpoints, and managing communication with the MySQL database using native SQL queries.

- React: React is used on the client side to create a responsive and interactive user interface. It facilitates the rendering of result summaries, graphs, and other visual components, providing a seamless user experience.

- Node.js: Node.js serves as the server runtime, enabling the execution of server-side logic. It works in conjunction with Express.js to handle incoming HTTP requests, process data, and communicate with the MySQL database.

3.3 System Design

The system design integrates customer and operator workflows into a smart assembly line using advanced robotics, database management, and real-time monitoring. This ensures efficient production and seamless customer interaction. The flowchart outlines the following key components:

- User Login System
  - Login/Sign-Up: Users (either Customers or Operators) log in to the system. If login fails, an error message is displayed.
  - Role-Based Access: The system provides distinct functionalities for customers and operators based on their roles.
- Customer Workflow

- Pattern Input: Customers can input and manage patterns, which are stored in the database.
- Order Management: Customers select patterns and create orders.
- Orders are validated by the system controller for feasibility.
- Once validated, the orders are confirmed, and the status is updated for the customer.
- System Status and Information: Customers can access real-time information about their orders and the system's status.

- Operator Workflow
  - Production Management: Operators select patterns for production and update the controller for execution.
  - Real-time status of production is displayed for operators.
  - Dispatch Management: Operators manage the dispatch of completed orders, updating both the controller and database.
- Automated Storage and Retrieval System (ASRS)
  - Raw Material (RM) Handling: ASRS dynamically picks empty pallets from the RM racks based on availability.
  - Finished Goods (FG) Storage: Once production is completed, filled pallets are placed in FG racks by the ASRS.
  - Dynamic Rack Allocation: Pallets are allocated using a shortest-distance matrix to minimize movement and optimize storage.
- Robotic Integration
  - SCARA Robot: Places empty pallets on the conveyor for processing.
  - COBOT: Performs precise filling of pallets based on customer specifications.
  - Vision System: Detects and verifies patterns to ensure accuracy. If errors are detected, pallets are recirculated for correction.
  - Viper Robot: Handles the movement of pallets between production stages and racks.
- Database Management
  - Real-Time Updates: The database maintains updated information about:

  * Patterns and their statuses.
  * Customer orders.
  * Production stages and completed tasks.
- Dynamic Rack Allocation: Tracks rack usage and assign empty racks based on the shortest-distance matrix.

3.4 System Workflow Representation

The System Workflow Representation section provides a detailed overview of the interactions and processes within the system, capturing how users and operators interact with various components to achieve desired functionalities. This section includes visual diagrams such as use case diagrams and sequence diagrams, which illustrate the dynamic workflows and logical sequences of operations. These representations clarify the system's behavior, highlight user and operator roles, and demonstrate how different system components work cohesively to meet functional requirements. By analyzing these workflows, we can gain insights into the efficiency and coherence of the system design, ensuring seamless operation and user satisfaction.

3.4.1 Use Case Diagram

The Use Case Diagram Representation provides a visual summary of the system's functionality by outlining the primary actors (users and operators) and their interactions with the system. This diagram highlights the major use cases, detailing the specific functionalities available to each actor and their connection to the core components of the system. It serves as a high-level representation of the system's capabilities and user interactions, helping to understand the scope and boundaries of the system in a simplified manner.

3.4.2 User Sequence Diagram Representation

The User Sequence Diagram Representation illustrates the step-by-step interaction between a user and the system for various functionalities. This diagram provides a dynamic view of the processes initiated by the user, such as inputting patterns, placing orders, and viewing the status of the system. It captures the sequential flow of events, emphasizing the communication between the user interface, backend logic, and database, ensuring the smooth execution of user requests.

3.4.3 Operator Sequence Diagram Representation

The Operator Sequence Diagram Representation details an operator's sequential flow of actions to manage the system's backend functionalities. It highlights processes such as selecting patterns, initiating production workflows, updating system controllers, and managing dispatch operations. This diagram emphasizes the operator's role in coordinating system activities and ensuring that the production and storage workflows are carried out efficiently while maintaining seamless communication with the database and automated systems.

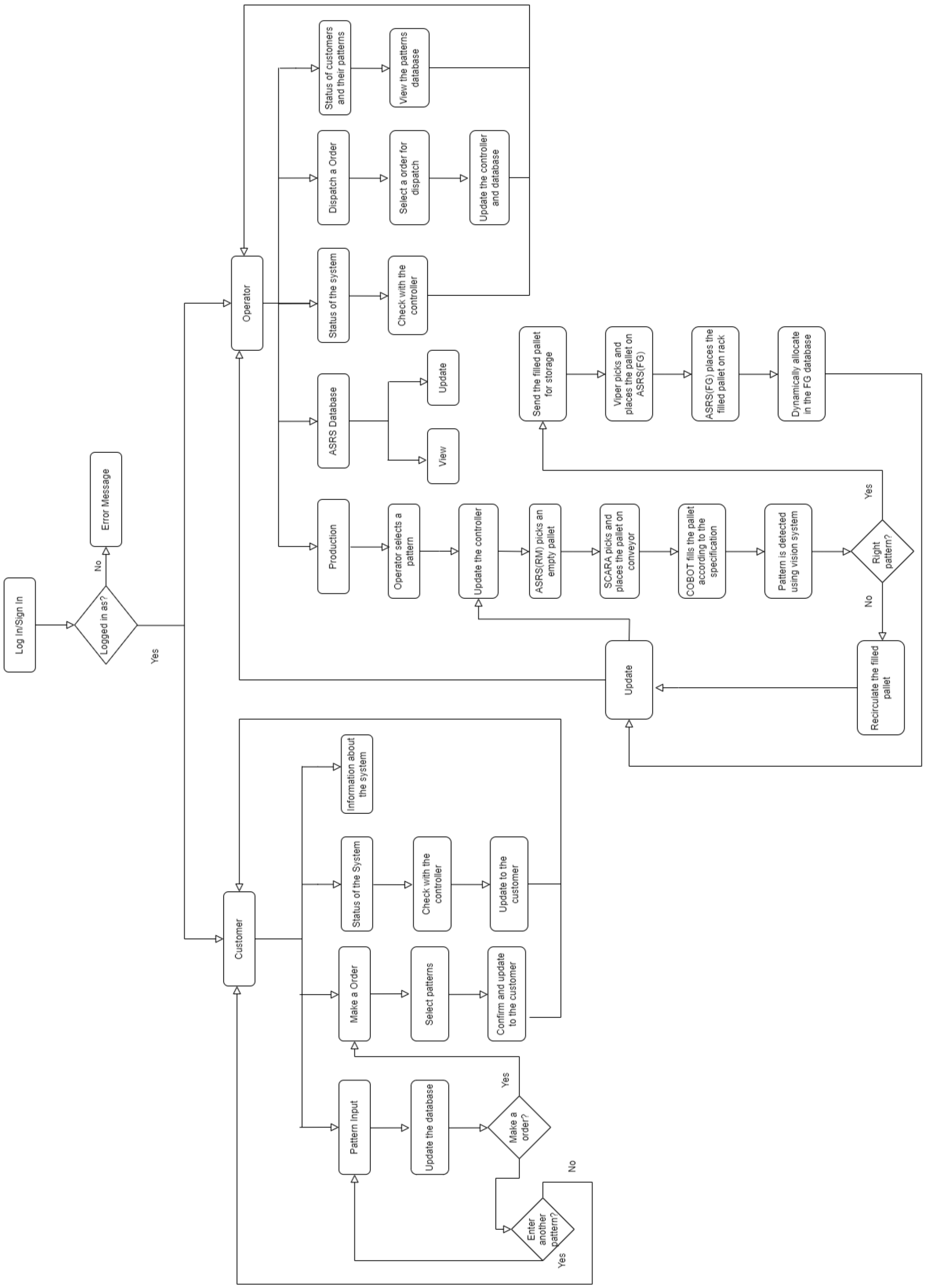

Figure 3.2: Working of the System

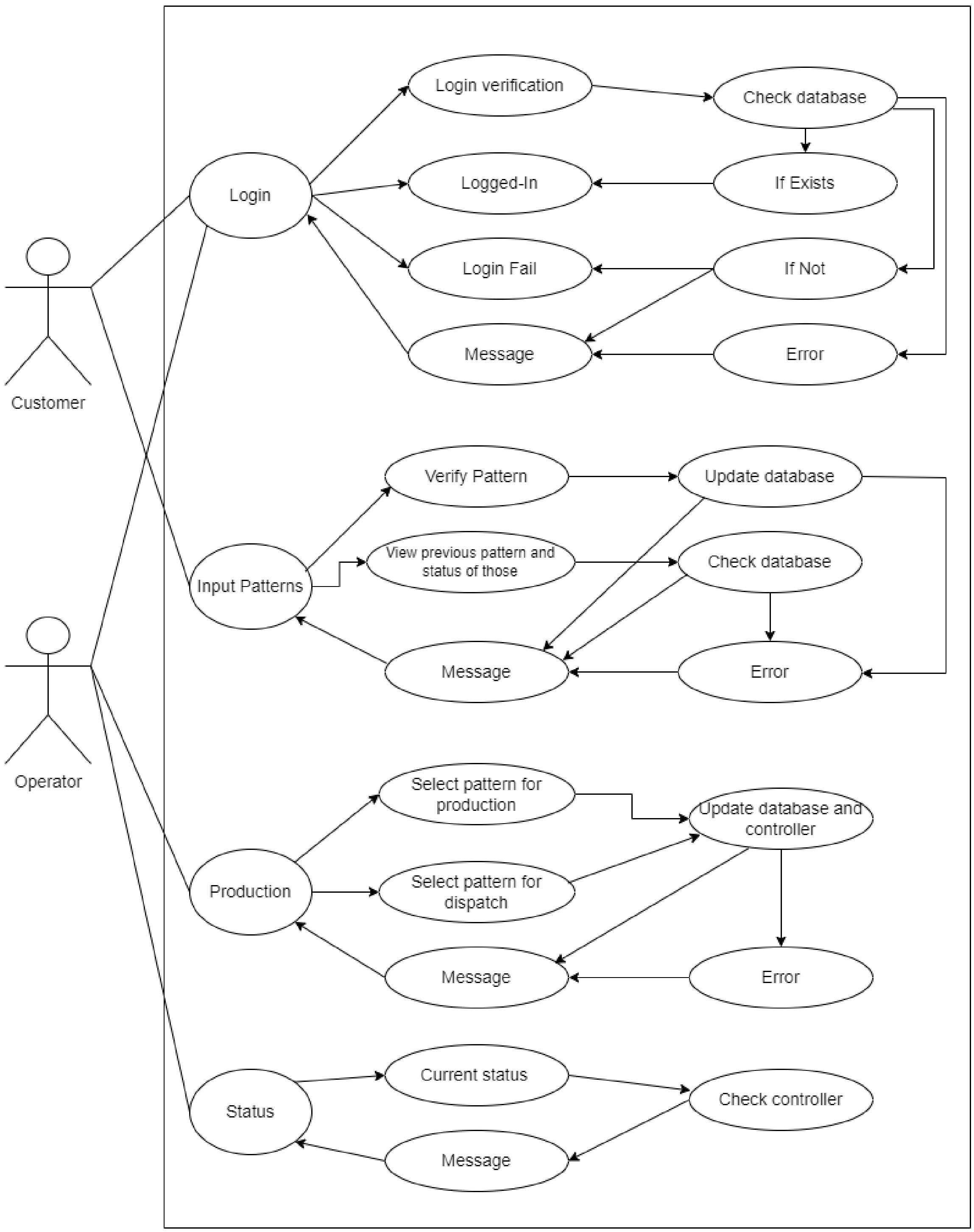

Figure 3.3: Use Case Diagram

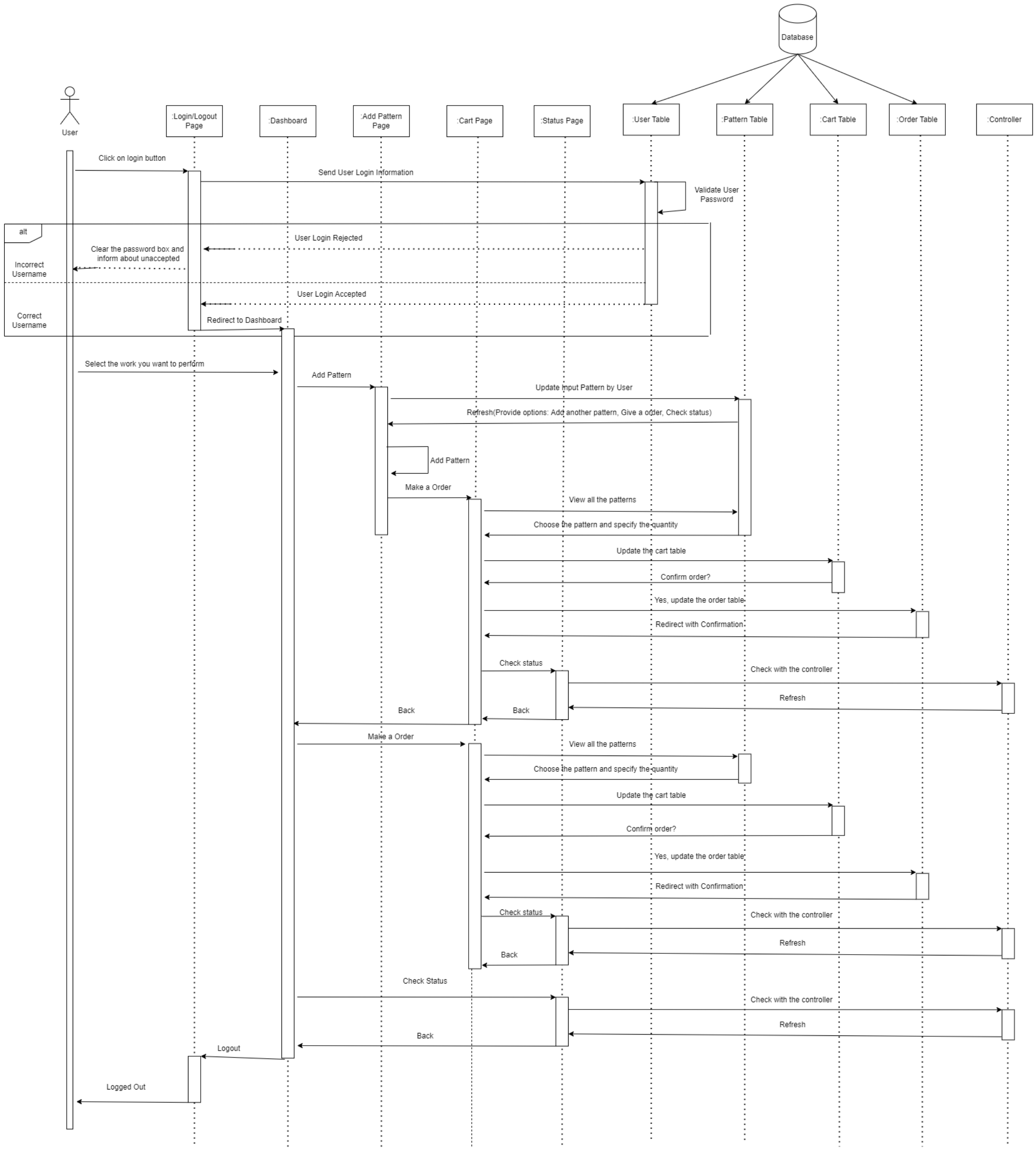

Figure 3.4: User Sequence Diagram

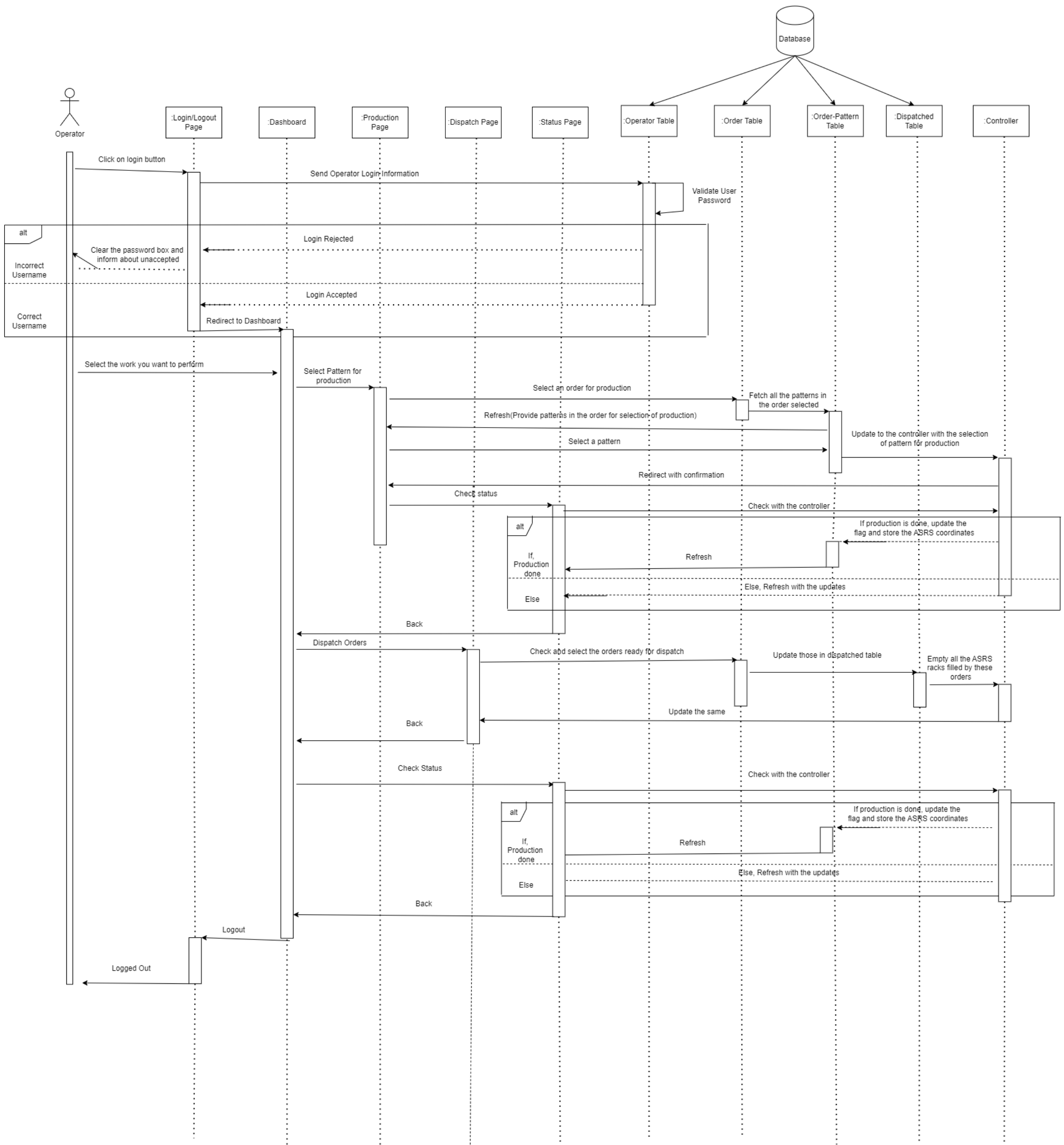

Figure 3.5: Operator Sequence Diagram

Chapter 4

IMPLEMENTATION

This chapter gives a brief description about implementation details of the system by describing each component with its code skeleton in terms of algorithm.

4.1 Database Design

The system's database architecture is divided into two main schemas: Customer Schema and Operator Schema.

- Customer Schema: Managing customer interactions, order processing, and pattern tracking. It includes tables for customers, patterns, orders, and items, ensuring seamless handling of user data, uploaded patterns, and order details.
- Operator Schema: Designed to manage production workflows, resource allocation, and task execution. It includes tables for operators, production, machines, racks, locations, processes, and dispatches, providing comprehensive control over production activities, machine operations, and resource tracking.

These schemas form a robust, interlinked database structure that ensures efficient order management, production tracking, and system scalability. This design supports both customer-facing functionalities and backend production processes, enabling smooth operation across all phases of the system.

4.1.1 Database Design for Customer Schema

The Customer Schema focuses on managing customer-related data, patterns, and orders within the system. It includes four main tables that facilitate user management, order processing, and pattern tracking.

- **1. Customers Table**
  - *Purpose*: Stores customer-specific information such as login credentials and personal details.
  - *Key Attributes*:

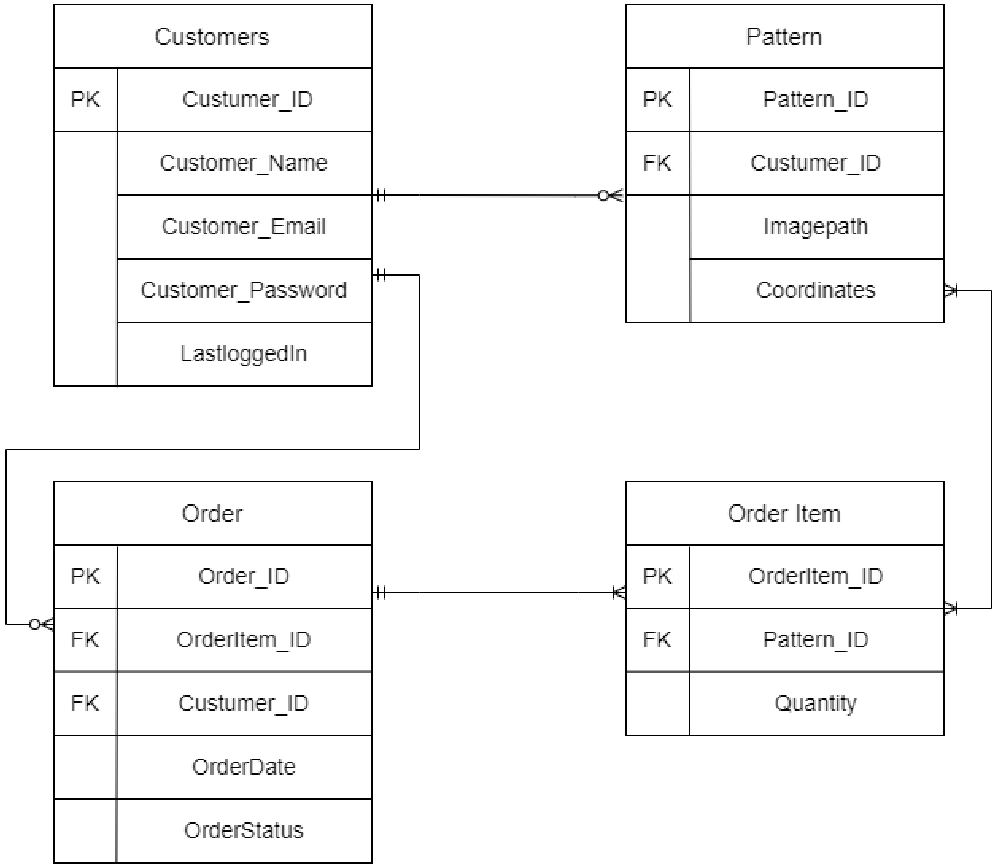

Figure 4.1: User Schema Diagram

   * **Customer_ID (PK)**: Unique identifier for each customer.
   * **Customer_Name**: Name of the customer.
   * **Customer_Email**: Email ID for customer communication.
   * **Customer_Password**: Password for authentication purposes.
   * **LastLoggedIn**: Timestamp for tracking the most recent login activity.

- **2. Pattern Table**
  - *Purpose*: Tracks patterns uploaded by customers, including image-related information and coordinates.
  - *Key Attributes*:
    * **Pattern_ID (PK)**: Unique identifier for each pattern.
    * **Customer_ID (FK)**: Links patterns to a specific customer.
    * **ImagePath**: Path or location of the pattern image stored in the system.
    * **Coordinates**: Position or coordinate data associated with the pattern.

- **3. Order Table**

- *Purpose*: Records all orders placed by customers. Each order links to the uploaded patterns.
- *Key Attributes*:
  - **Order_ID (PK)**: Unique identifier for each order.
  - **OrderItem_ID (FK)**: Links the order to specific items.
  - **Customer_ID (FK)**: Identifies the customer placing the order.
  - **OrderDate**: Timestamp indicating when the order was placed.
  - **OrderStatus**: Status of the order (e.g., new, production, dispatched).

- **4. Order Item Table**
  - *Purpose*: Represents individual items included in an order.
  - *Key Attributes*:
    - **OrderItem_ID (PK)**: Unique identifier for each order item.
    - **Pattern_ID (FK)**: Links the order item to the corresponding pattern.
    - **Quantity**: Specifies the quantity of the pattern ordered.

4.1.2 Database Design for Operator Schema

The Operator Schema focuses on the production process, resource allocation, and task management. It includes various interconnected tables that define workflows, machine operations, and dispatch details.

- **1. Operator Table**
  - *Purpose*: Maintains information about system operators responsible for overseeing production processes.
  - *Key Attributes*:
    - **Operator_ID (PK)**: Unique identifier for each operator.
    - **Order_ID (FK)**: Associates the operator with specific orders in the production line.

- **2. Production Table**
  - *Purpose*: Monitors the production activities corresponding to customer orders.
  - *Key Attributes*:
    - **Production_ID (PK)**: Unique identifier for each production entry.
    - **OrderItem_ID (FK)**: Links production to specific order items.

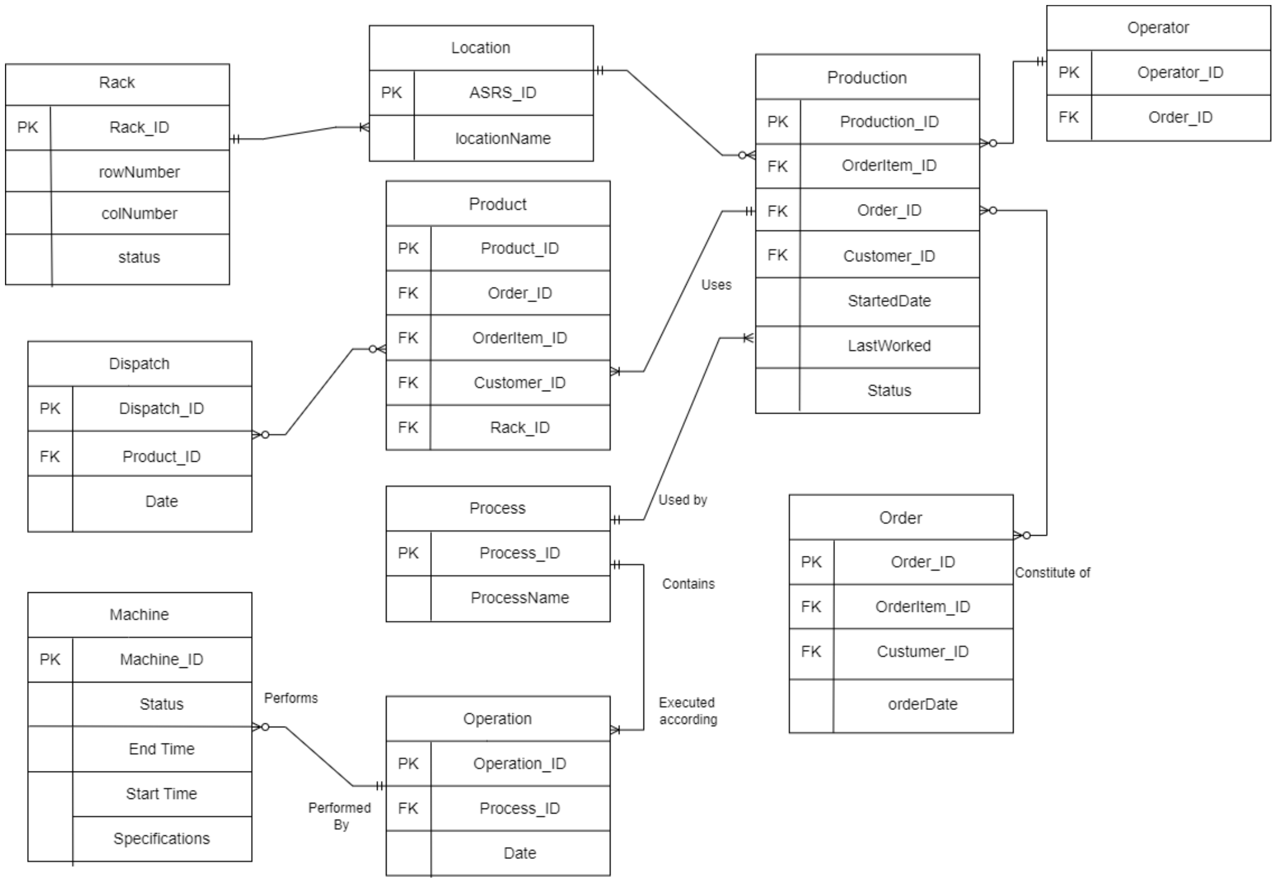

Figure 4.2: Operator Schema Diagram

  - **Order_ID (FK)**: Associates production with a specific order.
  - **Customer_ID (FK)**: Links production data back to the customer.
  - **StartedDate**: Records when the production process began.
  - **LastWorked**: Timestamp indicating the last update in production.
  - **Status**: Indicates the current production status (e.g., stopped, production).

- **3. Machine Table**
  - *Purpose*: Manages data about machines used in the production process.
  - *Key Attributes*:
    - **Machine_ID (PK)**: Unique identifier for each machine.
    - **Status**: Current operational status of the machine (e.g., active, idle).
    - **Start Time/End Time**: Records machine activity periods.
    - **Specifications**: Detailed information about the machine's capabilities.

- **4. Rack and Location Tables**

- *Rack Table*:
  - **Rack_ID (PK)**: Unique identifier for racks.
  - **rowNumber/colNumber**: Identifies rack positions.
  - **Status**: Indicates rack availability.
- *Location Table*:
  - **ASRS_ID (PK)**: Identifier for locations.
  - **locationName**: Describes the physical location of racks/resources.

- **5. Dispatch Table**
  - *Purpose*: Tracks the final dispatch of products from the production system.
  - *Key Attributes*:
    - **Dispatch_ID (PK)**: Unique identifier for each dispatch event.
    - **Product_ID (FK)**: Links dispatch to the corresponding product.
    - **Date**: Timestamp of the product dispatch.

- **6. Process and Operation Tables**
  - *Process Table*:
    - **Process_ID (PK)**: Unique identifier for each process.
    - **ProcessName**: Name of the process executed.
  - *Operation Table*:
    - **Operation_ID (PK)**: Unique identifier for operations.
    - **Process_ID (FK)**: Links operations to processes.
    - **Date**: Timestamp indicating when the operation was performed.

4.2 Interface Designed

The Interface Designed section focuses on the intuitive and user-centric graphical user interfaces (GUIs) developed for the system, catering to the distinct roles of users and operators.

For the User Interface, the design allows users to seamlessly add custom patterns, select from predefined ones, and place orders. The interface provides a clear and visually appealing way to track the status of their orders, ensuring transparency and ease of navigation. The layout emphasizes simplicity, enabling users to interact with the system efficiently without prior technical expertise.

For the Operator Interface, the dashboard is designed to provide comprehensive oversight of operations. It displays new, pending, and completed orders, alongside visual analytics such as business performance graphs. Operators can easily manage orders, start and stop production processes, and monitor machine statuses in real-time. The interface also provides access to detailed production and machine history, streamlining decision-making and operational control. Additionally, dispatch details are readily available, ensuring smooth order tracking and fulfillment.

Both interfaces are built using React for responsiveness and hosted on Firebase for seamless deployment. The backend, powered by Node.js and MySQL, ensures real-time data synchronization between interfaces and the database, making the system robust and reliable. The user-friendly design, combined with real-time functionality, bridges the gap between operators, users, and industrial automation systems, fostering efficiency and productivity.

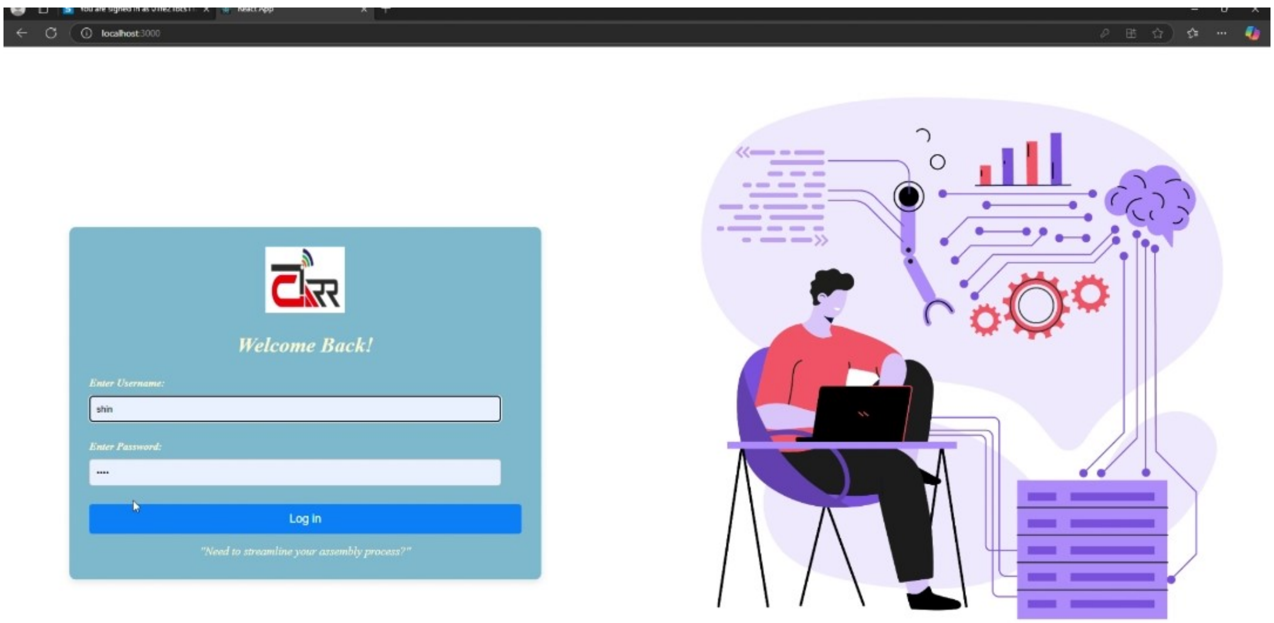

Figure 4.3: Login Page

Figure 4.4: Customer Home Page

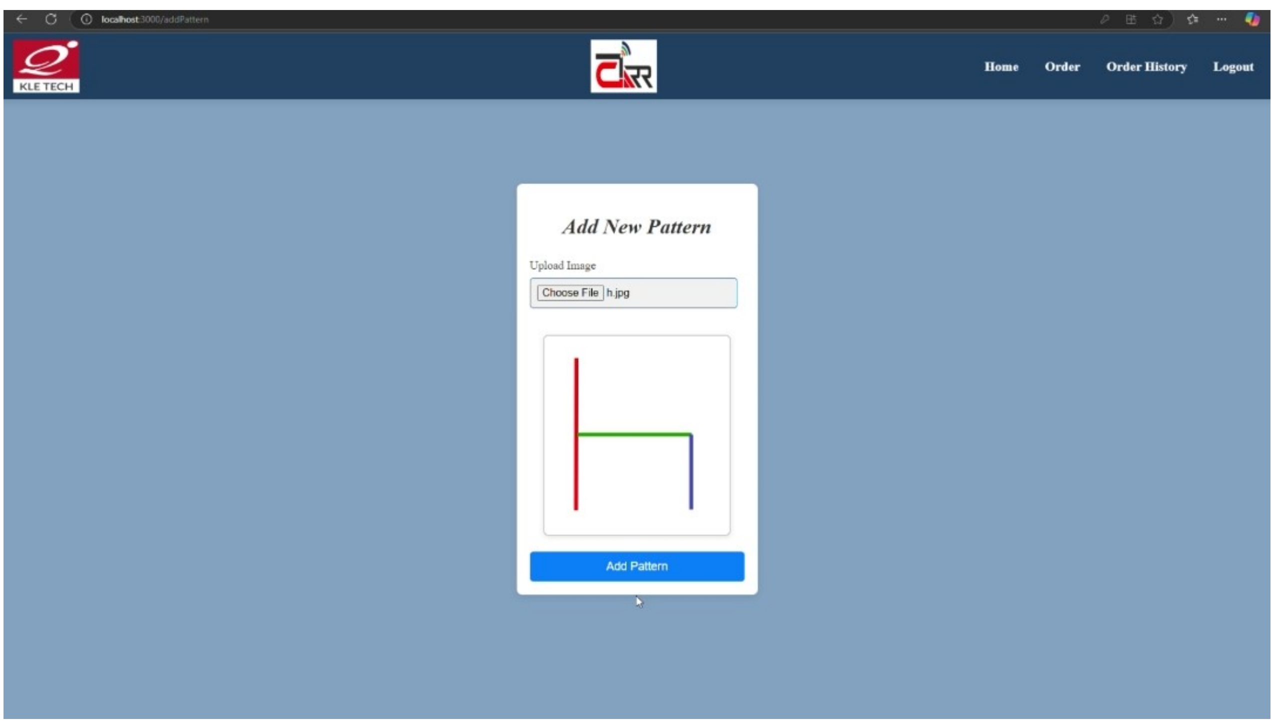

Figure 4.5: Add Pattern Page

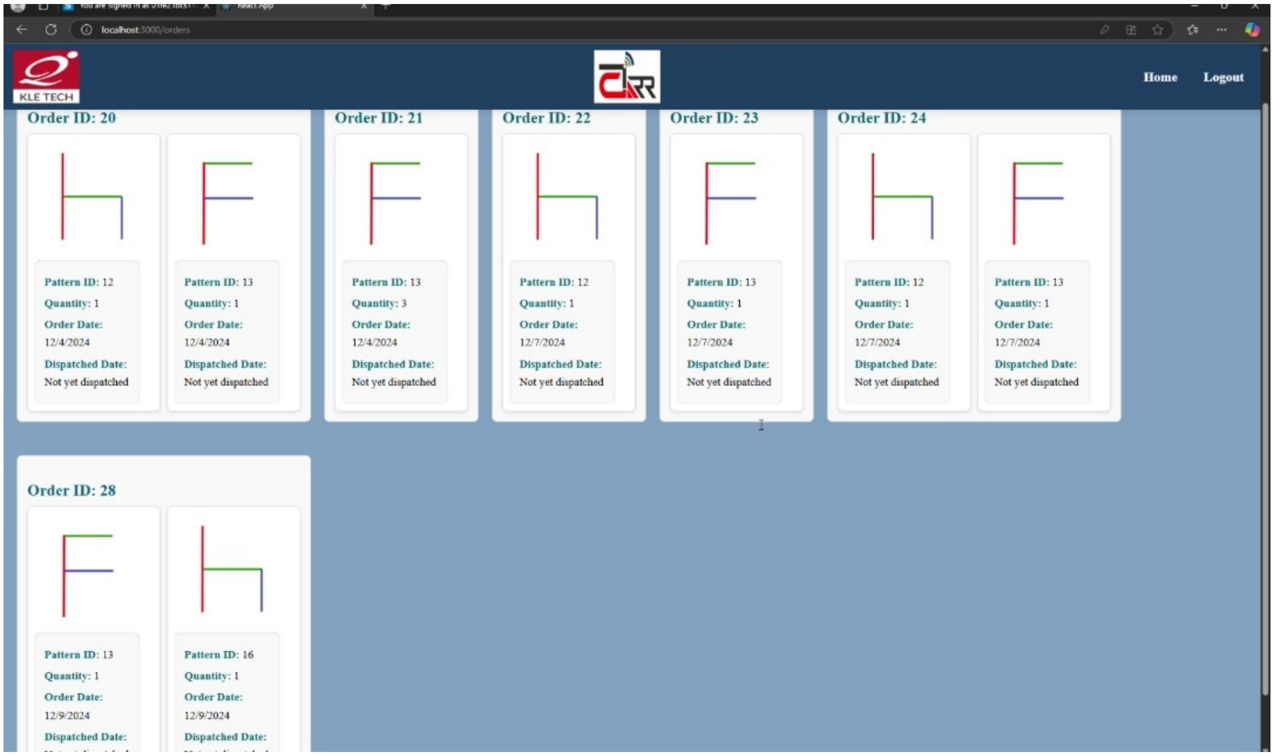

Figure 4.6: Customer Order Status Page

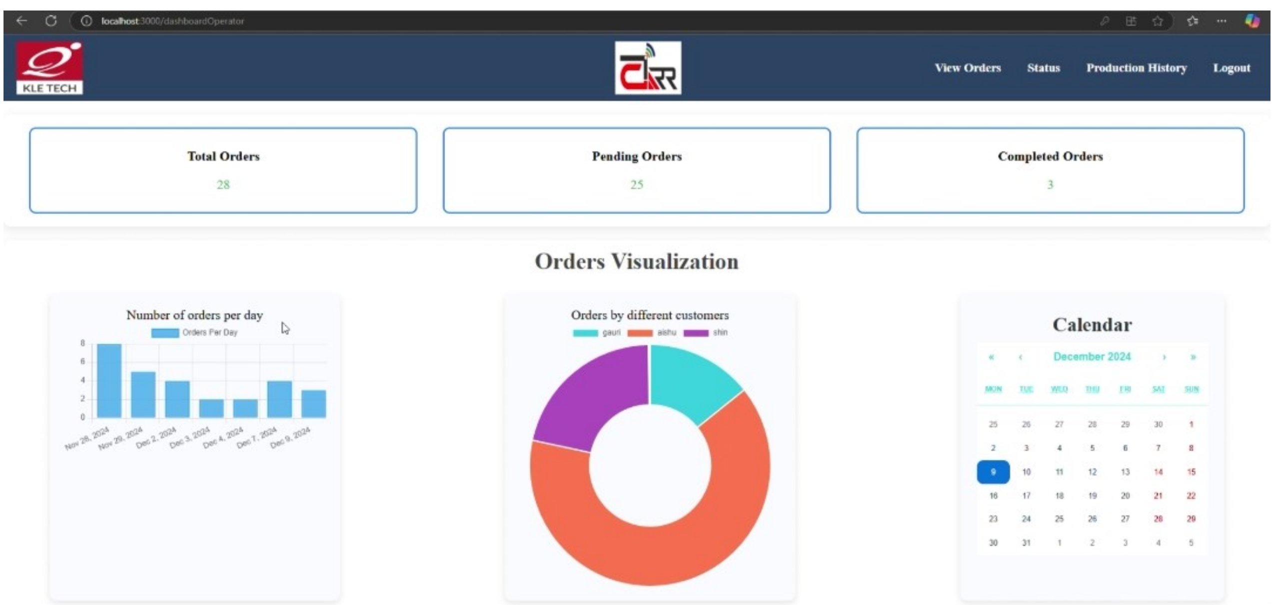

Figure 4.7: Operator Home Page

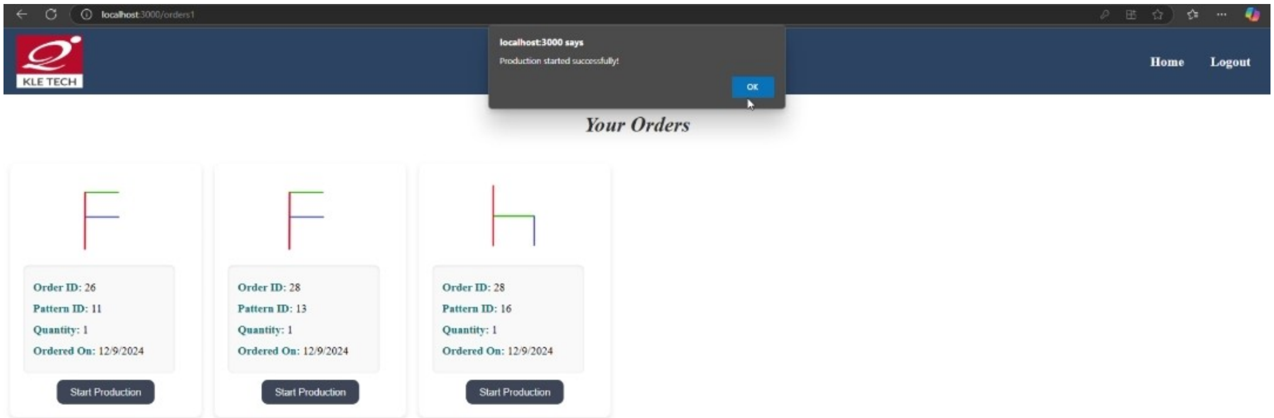

Figure 4.8: Operator New Orders Page

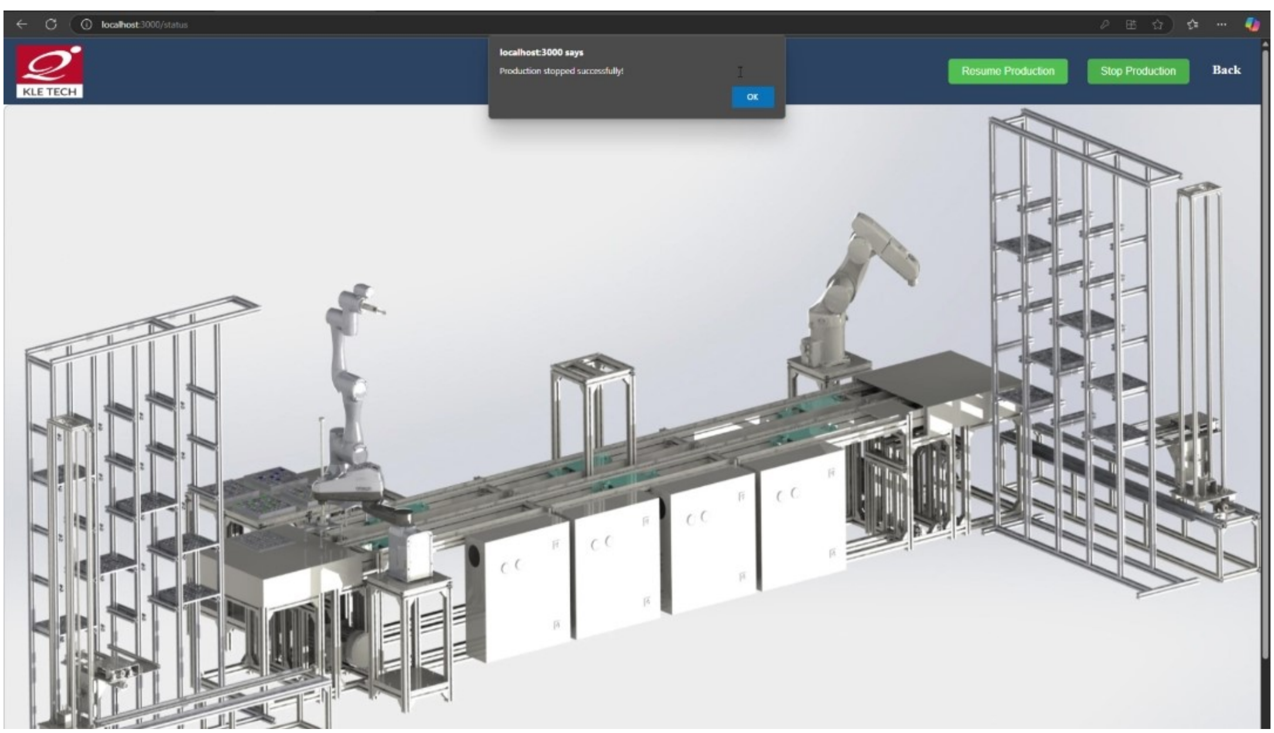

Figure 4.9: Operator Status Page

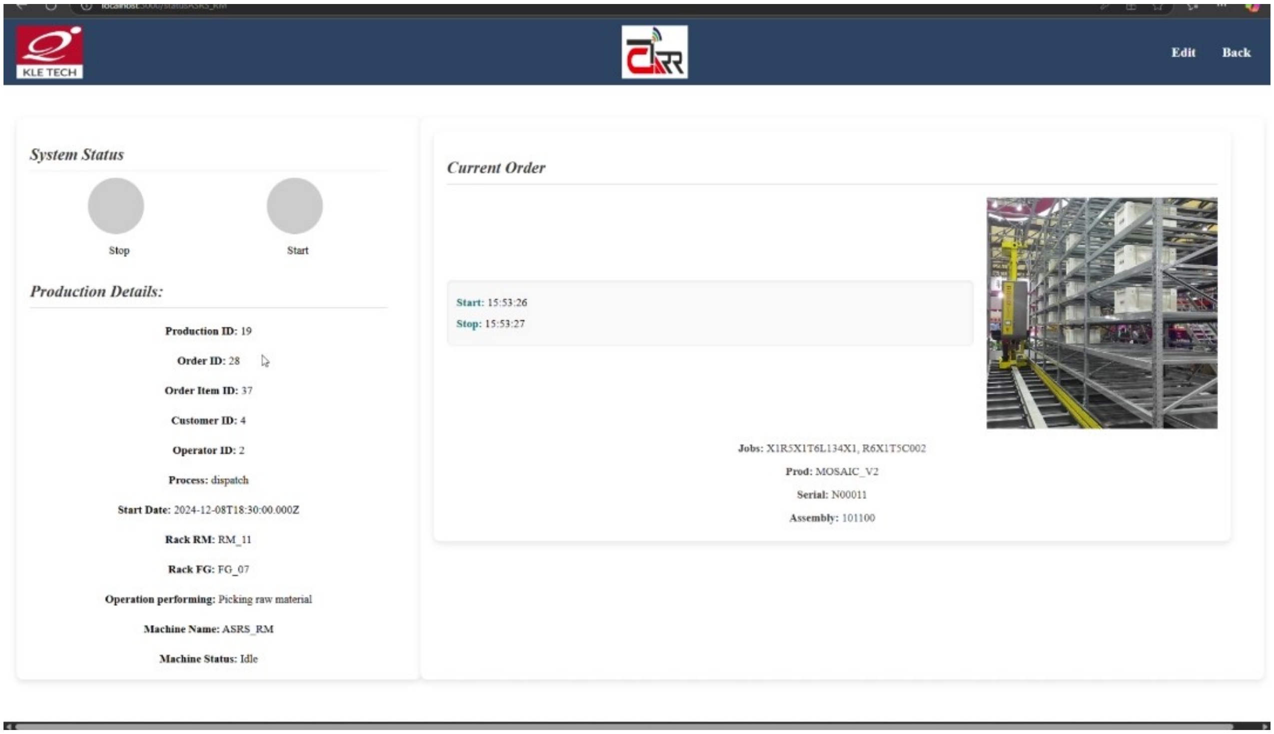

Figure 4.10: ASRS Status Page

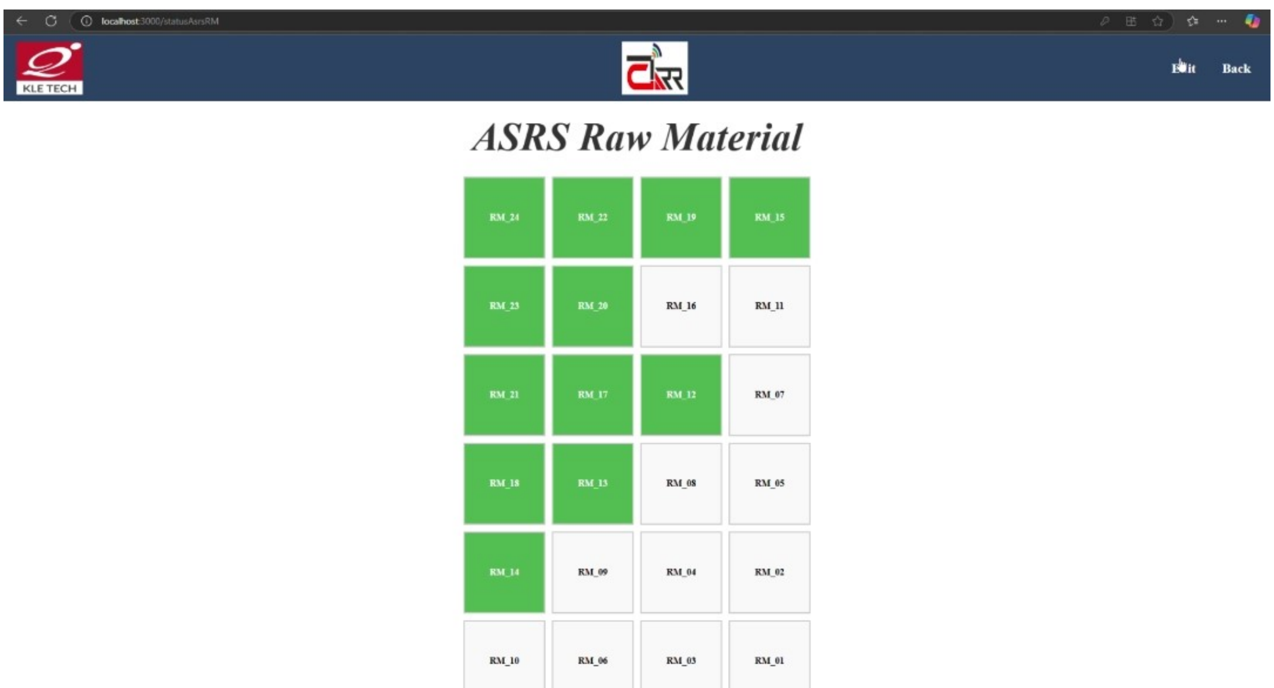

Figure 4.11: ASRS Rack Status Page

Figure 4.12: ASRS Rack Edit Page

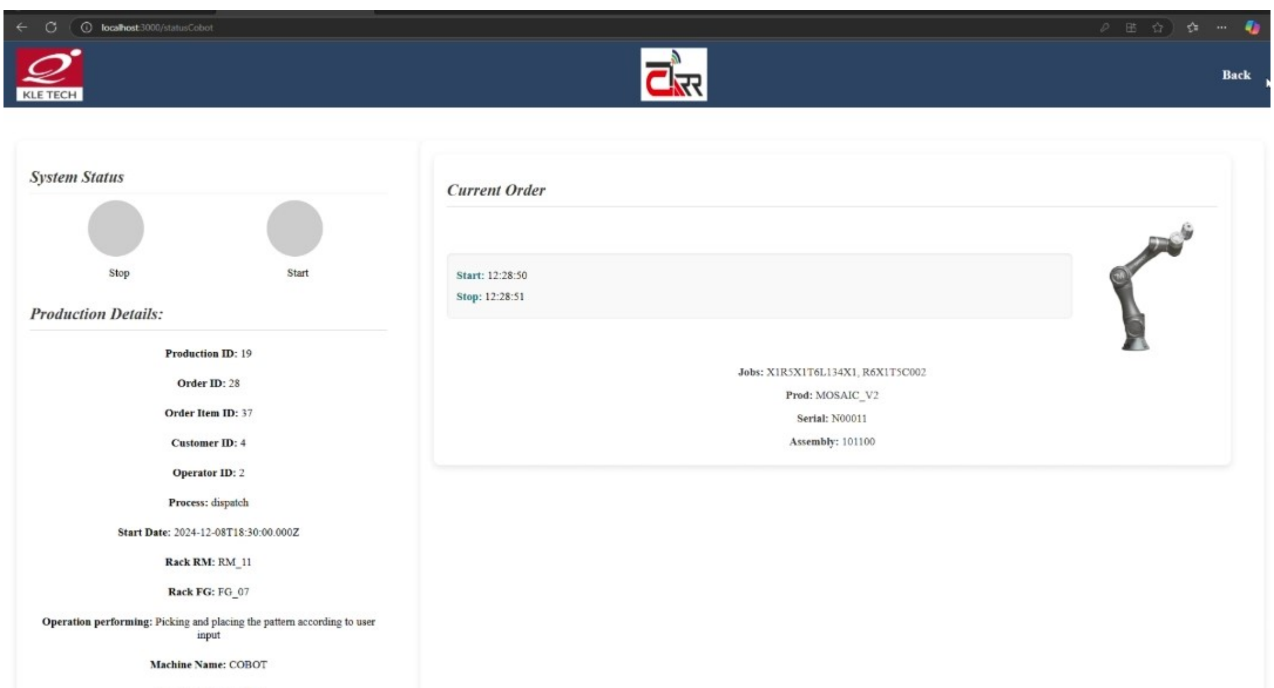

Figure 4.13: Cobot Status Page

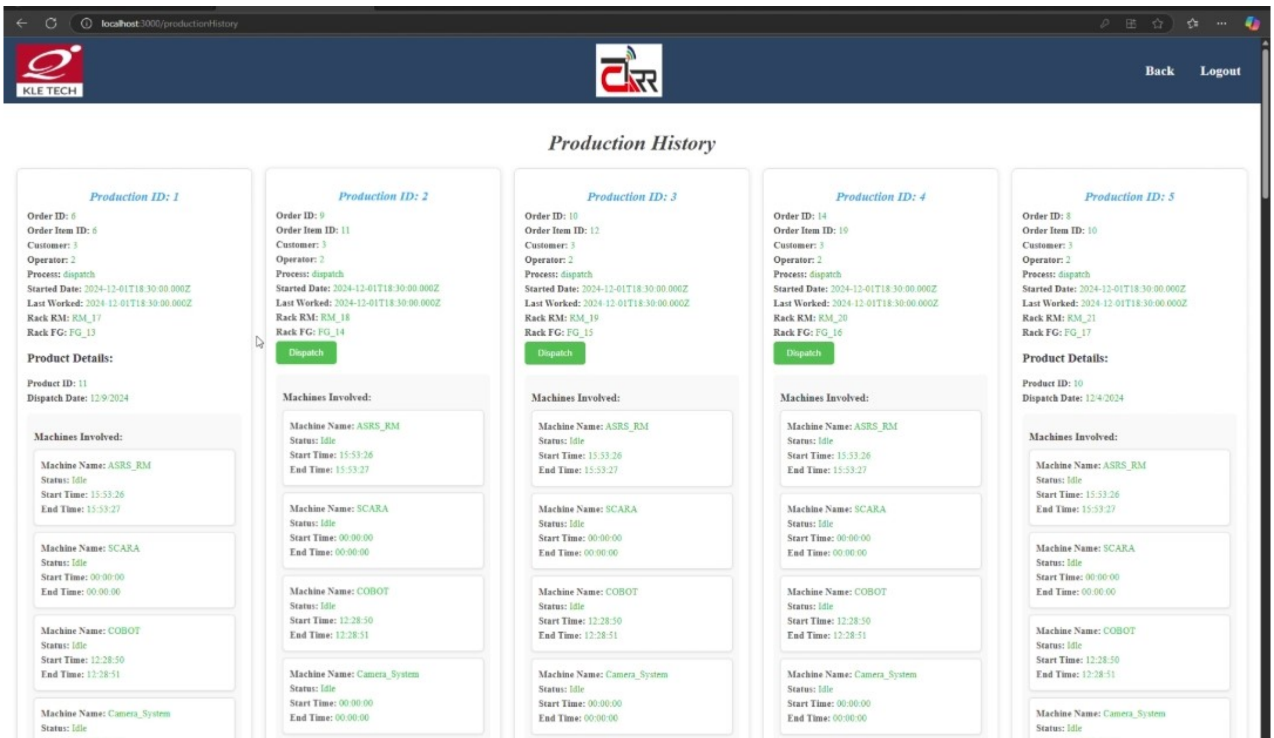

Figure 4.14: Production History Page

Chapter 5

RESULTS AND DISCUSSIONS

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. The development of the system has successfully addressed the need for real-time monitoring and management in a smart manufacturing lab. By integrating various components such as robots, the ASRS, and the database, the system ensures seamless communication and process automation. The user interface, built with React and hosted on Firebase, provides a responsive and intuitive experience for both users and operators. Users can efficiently add and manage patterns, place orders, and track their statuses, while operators benefit from a robust dashboard that displays new, pending, and completed orders along with real-time business insights through graphs.

The system establishes a smooth connection with robots, ensuring precise execution of tasks like pallet handling, quality inspection, and order dispatch. Robot 2 (SCARA) efficiently transfers pallets between the ASRS and conveyor, while Robot 3 (Cobot), equipped with a vision system, ensures that items meet quality specifications before further processing. Robot 4 (VIPER) handles the placement of verified pallets back into storage, ensuring accuracy and speed. The integration of a camera-based quality inspection system further enhances production reliability by automatically identifying and recirculating faulty pallets. Real-time updates in the MySQL database ensure that every operation—from order initiation to dispatch—is recorded and accessible for monitoring and analysis.

The results demonstrate improved workflow efficiency, reduced manual intervention, and enhanced data accuracy. Operators can monitor machine status and production history, make informed decisions, and manage the entire production cycle effectively. The system's modularity and scalability provide a solid foundation for accommodating future requirements, ensuring its adaptability to evolving industrial needs.

Chapter 6

CONCLUSION AND FUTURE SCOPE OF THE WORK

The project has successfully delivered a centralized system for managing and monitoring operations in a smart manufacturing lab. The seamless integration of React for the frontend, Node.js for the backend, and MySQL for the database has resulted in a robust platform that addresses the needs of users and operators. By connecting with robots and automation systems, the solution ensures efficient production workflows, real-time data synchronization, and a significant reduction in manual effort. The role-based design of the system allows for tailored user experiences, enhancing usability and operational control.

Looking forward, the system can be expanded to include advanced analytics and predictive maintenance features by leveraging machine learning models. Incorporating IoT sensors for enhanced machine monitoring and real-time alerts could further improve the system's efficiency and reliability. The integration of cloud-based database solutions and robotics APIs could enable better scalability and support remote access. Additionally, exploring augmented reality (AR) interfaces for operators could enhance the way data and machine operations are visualized and controlled. This project lays the groundwork for a highly adaptive and efficient smart manufacturing ecosystem, aligning with the principles of Industry 4.0 and paving the way for future advancements.