SOFTWARE ARCHITECTURE
AND
SYSTEM DESIGN DOCUMENTATION

for

PREDICTIVE MAINTENANCE AUTOMATION SYSTEM

Prepared By:

- **Pranay Siwach**
- **Nikhil Singh**
- **Sanchit Dangwal**
- **Praful Prakash Goel**

Indian Institute of Information Technology Guwahati

Software Architecture Style

Our PMAS (Predictive Maintenance Automation System) project involves:

- Technicians
- System Administrator
- Maintenance Tickets
- Alerts
- Responses (accept/reject)

This is a structured, domain-driven system - not a distributed SaaS platform with millions of users.

For the PMAS project, a **Layered Monolithic Architecture** is selected as it provides a structured, scalable, and maintainable solution aligned with the project's functional and non-functional requirements.

A. Why PMAS Falls Under Layered Architecture

A layered architecture organises the system into vertical layers where each layer has a clear responsibility and exposes limited interfaces to the layer above.

PMAS can be structured like:

1. Presentation Layer
   Handles interaction with users
   - UI (Web Interface)
   - Alert/Ticket Dashboard
   - Machine Dashboard
   - Technician Response Interface
   - Admin Panel

2. Business Logic Layer
   Contains the core logic of the system
   - Authorization & Role-Based Access Control
   - Ticket Management Service (creation, validation, status transitions)
   - Response Handling Logic (accept/reject processing, reassignment rules)
   - Prediction Service (ML module for failure prediction and priority scoring)
   - Technician Domain Logic
   - System Administrator Domain Logic

3. Data Access Layer

   Handles communication with the database

   - CRUD operations
   - Ticket Repository
     i. Create, update, delete tickets
     ii. Retrieve tickets by status or technician
     iii. Fetch ticket history
   - Technician Repository
     i. Retrieve available technicians
     ii. Update technician availability
     iii. Manage technician profile
   - Alert Repository
     i. Create alerts
     ii. Retrieve alerts by status
     iii. View/Acknowledge alerts
   - Machine Repository
     i. Update machine configuration
     ii. Retrieve machine by id

4. Database Layer

   Stores system data

   - User table
   - Machine table
   - Ticket table
   - Alert table

Granularity Justification:

- Each layer has **well-defined responsibility**
- Classes are grouped by **functionality**, not by service
- All components are deployed as a **single unit (monolith)**
- Communication happens internally (method calls), not over network

B. Why this is the best choice for PMAS

The Layered Monolithic Architecture is the most suitable architectural style for the PMAS (Predictive Maintenance Automation System) project because it aligns with the system's functional requirements, expected scale, and development constraints. The justification is discussed below:

1. Scalability

The PMAS system is expected to serve a limited number of users, including administrators and technicians within an organization. The workload primarily involves ticket creation, technician assignment, response handling, and predictive analysis.
A monolithic layered architecture supports:

- Vertical scaling (increasing server resources)
- Efficient in-memory communication between layers
- Simple deployment as a single unit

Since the system does not require independent scaling of components or distributed service management, adopting microservices would introduce unnecessary complexity. The layered monolithic design provides sufficient scalability for the expected workload.

2. Maintainability

Maintainability is significantly improved due to clear separation of concerns across layers:

- Presentation Layer handles user interaction.
- Business Logic Layer contains system rules and decision-making.
- Data Access Layer manages database interaction.
- Database Layer stores persistent data.

Each layer has a well-defined responsibility. Changes in one layer (e.g., updating the response handling logic or modifying prediction logic) do not directly impact other layers.

For example:

- Updating the ML-based prediction module affects only the Business Logic Layer.
- Changing database technology affects only the Data Access Layer.

This modular separation makes debugging, testing, and feature enhancement easier.

3. Performance

In a layered monolithic architecture:

- Components communicate through direct method calls.
- There is no network overhead between services.
- No serialization/deserialization cost between modules.

This results in lower latency and improved performance compared to distributed architectures such as microservices, which require inter-service communication over a network.
Since PMAS requires quick ticket assignment and response processing, minimizing communication overhead improves overall system responsiveness.

4. Simplicity and Development Efficiency

The project is developed within academic constraints and limited development time. A layered monolithic architecture:

- Is easier to implement and test
- Requires simpler deployment configuration
- Avoids distributed system complexities (API gateways, service discovery, container orchestration)

This allows us to focus on core functionalities such as prediction modeling and assignment logic rather than infrastructure management.

5. Suitability to Project Requirements

PMAS requires:

- Structured business rules
- Controlled access roles (Admin, Technician)
- Predictive analytics integration
- Centralized data management

A layered architecture naturally supports structured domain logic and centralized control, making it ideal for systems with strong internal rule processing.
Microservices architecture is more appropriate for large-scale, highly distributed systems with independent teams and massive traffic, which is not the case for PMAS.

6. Integration of Predictive Analytics

The layered structure allows seamless integration of a Prediction Service within the Business Logic Layer. The ML module operates as a supporting component without requiring independent deployment, simplifying integration and maintenance.

This ensures that predictive intelligence enhances decision-making without increasing architectural complexity.

Components

Our PMAS (Predictive Maintenance Automation System) includes the following components:

1. Authentication & Authorization Component

- Authenticates users by validating login credentials of System Administrators and Technicians
- Supports registration and account creation of Technicians by the System Administrator
- Ensures secure handling of user credentials and access rights.
- Handles access control to restrict system functionalities based on user roles

2. Ticket Management Component

- Supports manual ticket creation by the System Administrator and automated ticket creation by the Maintenance System
- Supports updating the ticket status by the System Administrator and Technicians
- Provides functionality to view tickets by their id or to view assigned tickets
- Handles history of past tickets generated by the system

3. Machine Management Component

- Manages the addition and removal of machines by the System Administrator
- Provides functionality to view machines by their id or to view the current health status of the machine
- Provides functionality for the System Administrator to update machine attributes

4. Alert Management Component

- Supports automated alert generation by the Maintenance System
- Provides functionality to view alerts by their status whether they are active or not
- Handles acknowledgment of alerts by the System Administrator

5. Technician Management Component

- Handles technician profile management and provides the functionality to update technician details
- Provides the functionality to register/remove technicians
- Provides the functionality to view technicians by whether they are available or not

6. Response & Assignment Component

- Provides the functionality for the technicians to accept/reject to work on a ticket
- Provides the functionality for the System Administrator to assign a particular technician to a ticket based on their availability and response
- Handles the logic to update assignment status of technicians and their response to a ticket

7. Prediction & Analytics Component

- Handles preprocessing of the data and feed the data into the prediction model
- Handles failure prediction of different machines based on their health scores and RUL
- Generate alerts and tickets if a machine is predicted to have low health score or RUL