



Software Architectures

Assignment 1 : Enterprise Resource Planning System

BITS Pilani

Harmalkar Rahul Rajan Sayali
ID No.: 2024TM93073

Purpose of the System



ERP System:

- The ERP (Enterprise Resource Planning) system is designed to streamline and integrate various business processes across the organization. It provides a unified platform to manage key functions such as finance, human resources, supply chain, procurement, and customer relationship management (CRM).

Key Benefits:

- **Improved Efficiency:** By centralizing data and automating routine tasks, the system minimizes manual work and reduces the likelihood of errors.
- **Real-time Information:** Enables real-time data access, facilitating better decision-making by providing accurate and timely insights.
- **Scalability:** As the organization grows, the ERP system can scale with it, supporting new functions or expanded operations.
- **Cross-departmental Collaboration:** Encourages collaboration across departments (like OIC) by providing a shared view of data and operations.

Key Requirements



Functional Requirements:

User Management:

Authentication and authorization for different user roles (e.g., Admin, Manager, Employee).

Role-based access control to various modules and data.

Finance Management:

General ledger, accounts payable, accounts receivable, and financial reporting.

Integration with banking systems for transaction processing.

Human Resources:

Employee data management, payroll processing, and benefits administration.

Recruitment and onboarding processes.

Supply Chain Management:

Inventory management, procurement, and order processing.

Supplier and vendor management, including contracts and performance tracking.

Customer Relationship Management (CRM):

Customer data management, sales tracking, and support ticketing.

Integration with marketing tools and customer feedback systems.

Reporting and Analytics:

Dashboards and reporting tools for business intelligence and performance tracking.

Customizable reports for different departments and stakeholders.

Non-functional Requirements:

Performance:

System must handle concurrent users efficiently with minimal latency.

Response times for user interactions should meet predefined standards.

Scalability:

Ability to scale horizontally or vertically to accommodate growth in data volume and user load.

Support for additional modules or features as needed.

Security:

Data encryption both at rest and in transit.

Regular security updates and vulnerability assessments.

Reliability:

High availability with minimal downtime.

Regular backups and disaster recovery plans.

Usability:

Intuitive user interface with easy navigation.

Comprehensive documentation and support resources.

Compliance:

Adherence to industry standards and regulations (e.g., GDPR, HIPAA).

Regular audits to ensure compliance.

Utility tree of ASR



Utility Tree of Architecturally Significant Requirements

Performance

- Goal: Ensure the system can handle high loads and provide quick responses.
- ASR: System must support a large number of concurrent users with minimal latency.

Scalability

- Goal: Enable the system to grow with the organization's needs.
- ASR: System should be able to scale both horizontally and vertically.

Security

- Goal: Protect sensitive data and ensure secure access.
- ASR: System must provide strong data protection and user authentication.

Reliability

- Goal: Maintain system availability and data integrity.
- ASR: System should have minimal downtime and robust disaster recovery.

Usability

- Goal: Ensure a user-friendly experience for all users.
- ASR: System should have an intuitive interface and provide adequate support.

Compliance

- Goal: Adhere to industry standards and regulatory requirements.
- ASR: System must comply with relevant laws and regulations.

Tactics to Achieve Top 5 ASRs



1. Performance

- Load Balancing: Distributes traffic across servers to avoid overload.
- Caching: Stores frequent data in memory for faster access.
- Database Optimization: Efficient indexing and query tuning to boost performance.

2. Scalability

- Modular Architecture: Independent modules that scale separately.
- Microservices: Services scaled individually, reducing bottlenecks.
- Distributed Databases: Spreads data across servers for faster access during high load.

3. Security

- Data Encryption: Protects data at rest and in transit.
- Role-Based Access Control (RBAC): Restricts access based on user roles.
- Security Updates: Regular patches to prevent vulnerabilities.

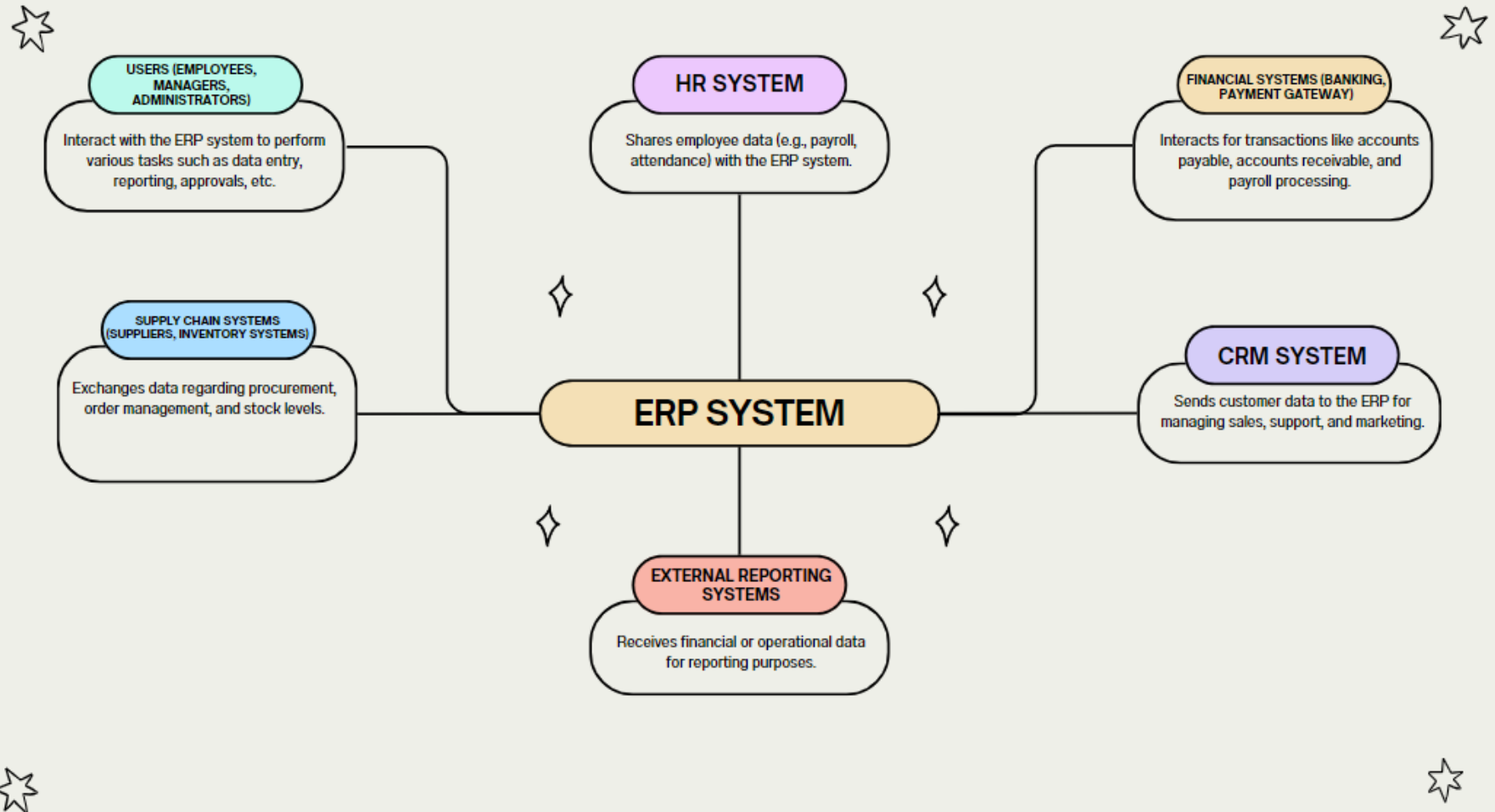
4. Reliability

- Redundant Systems: Backup components to reduce downtime.
- Automated Backups: Regular backups for data recovery.
- Failover Mechanisms: Switch to backup systems during failure.

5. Usability

- User-Centered Design: Intuitive interfaces for better user experience.
- Documentation: Guides to help users navigate the system.
- Training: Programs to boost user proficiency.

Context diagram



Module decomposition



ERP System

User Management Module

- Manages authentication, user roles, and access control.
- Interfaces with all other modules to enforce security and access permissions.

Finance Module

- Handles general ledger, accounts payable/receivable, budgeting, and financial reporting.
- Interacts with the User Management module for access control and HR Module for payroll data.

HR Module

- Manages employee data, payroll, attendance, and benefits.
- Interfaces with the Finance Module for payroll, and the User Management Module for employee access and records.

Supply Chain Module

- Manages procurement, inventory, order processing, and supplier information.
- Interacts with the Finance Module for invoice processing and payments and the CRM Module for order fulfillment.

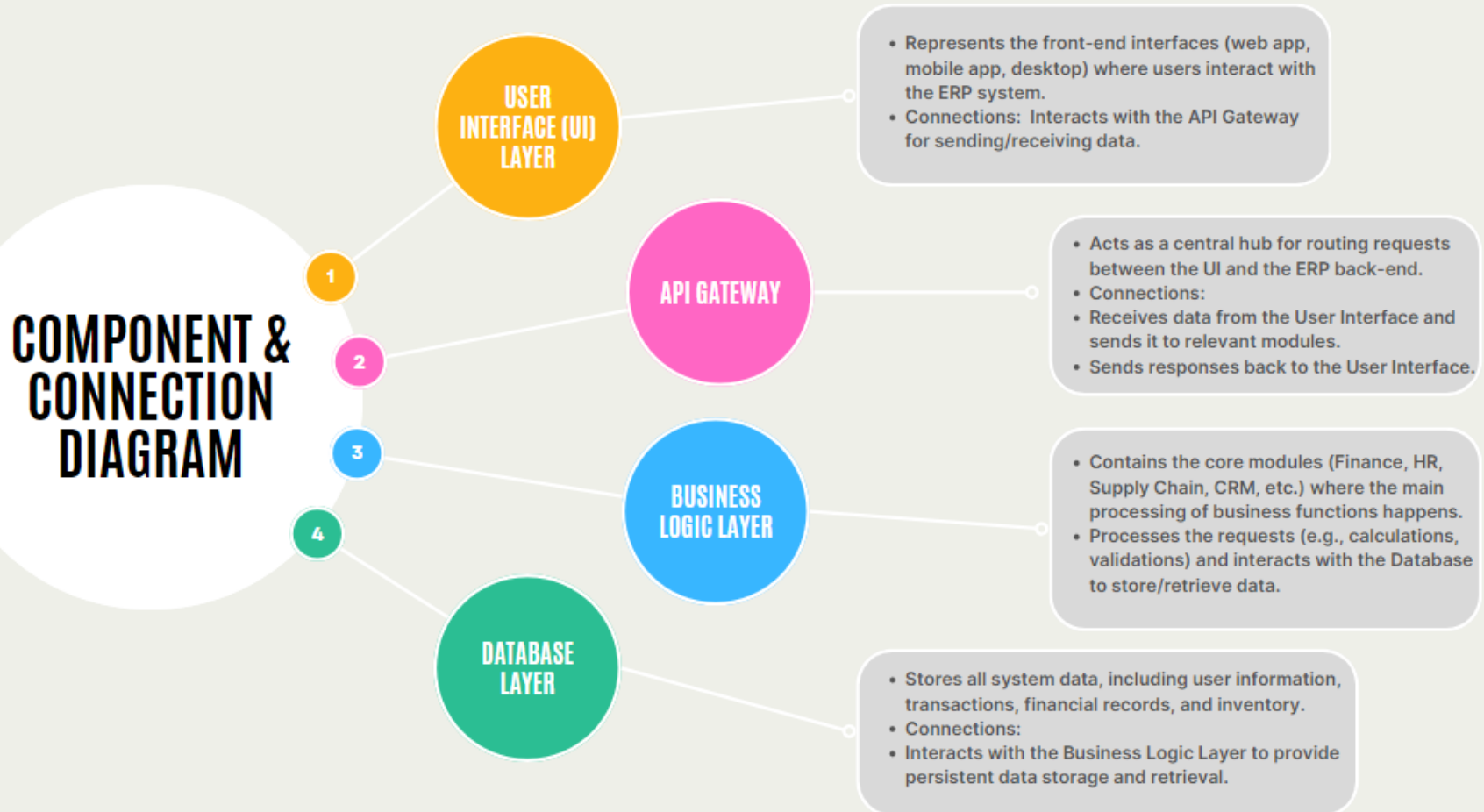
CRM Module

- Manages customer data, sales orders, and support tickets.
- Connects to the Supply Chain Module for orders and the Finance Module for invoicing and payments.

Integration Module

- Manages external system integrations (e.g., banking systems, external HR systems, CRM systems).
- Facilitates data exchange between the ERP and third-party services.

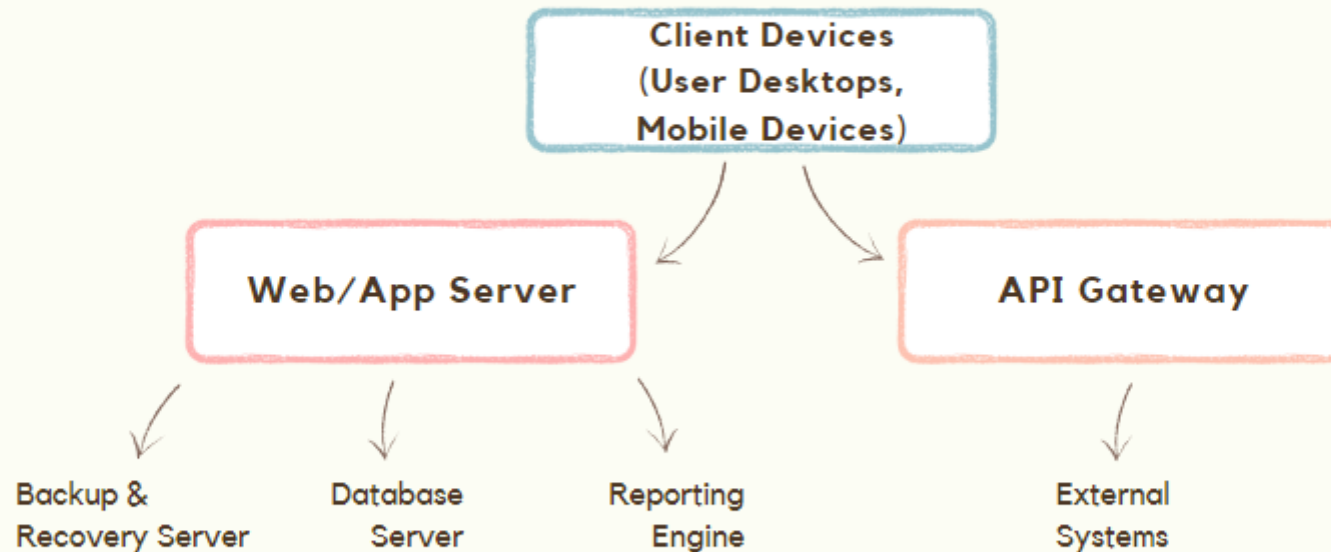
Component & Connection diagram



Deployment diagram



DEPLOYMENT DIAGRAM



System Description



The ERP system at **Trinamix INC** is a comprehensive solution designed to manage and integrate the company's core business processes. It consists of several key modules including **Finance**, **Human Resources**, **Supply Chain**, and **Customer Relationship Management (CRM)**, all of which work together to provide a centralized platform for managing data and operations across the organization.

Key Components and Functionality:

- **User Interaction:** Users (employees, managers, and administrators) access the system through web or mobile interfaces to perform tasks such as data entry, report generation, and approval workflows.
- **Business Logic Layer:** The system processes user requests through various modules:
 - **Finance Module:** Manages accounting, payroll, budgeting, and financial reporting.
 - **HR Module:** Handles employee data, payroll, attendance, and benefits.
 - **Supply Chain Module:** Oversees procurement, inventory, and supplier management.
 - **CRM Module:** Manages customer data, sales orders, and support requests.
- **Database:** A central database stores all ERP data, ensuring real-time access for all departments, facilitating coordination between finance, HR, and supply chain.
- **Integration:** The system communicates with external services (banking, payroll, CRM systems) through an **API Gateway** for seamless operations.
- **Reporting Engine:** Provides customizable reports and dashboards, allowing users to generate insights and track performance.
- **Backup & Recovery:** Data is regularly backed up to ensure minimal downtime and business continuity in case of system failures.

Key Learnings



Understanding Software Architecture:

I learned how different components in an ERP system (such as the web server, business logic, and database) work together to provide seamless business operations, focusing on how the architecture ensures performance, scalability, and reliability.

Importance of Architecturally Significant Requirements (ASRs):

By exploring performance, scalability, security, and usability as key ASRs, I gained insights into how these non-functional requirements shape the design of software systems to meet business needs.

Role of Tactics in Achieving System Goals:

The use of tactics like caching for performance, role-based access control for security, and modular architecture for scalability illustrated how specific design decisions address system challenges.

System Deployment and Data Flow:

The deployment diagram and component interactions highlighted the importance of how physical infrastructure (like servers, databases, and client devices) supports the software system's functionality.

Centralized Data Management:

I gained an appreciation for how centralized data storage in an ERP system allows for real-time access across departments, improving collaboration and decision-making within an organization.

External System Integration:

The study of APIs and external system integration (such as banking and CRM systems) demonstrated how ERP systems extend their functionality and communicate with third-party services, enhancing business flexibility.

Thank You