

PES UNIVERSITY
Department of Computer Science and Engineering
UE22CS341A: Software Engineering



Deliverable 2

Architecture Document
for
Sales Analytics Dashboard

Prepared by
Priyanka Kumari - PES1UG22CS454
Prayashi Verma - PES1UG22CS446

PES University RR Campus

1. Introduction

1.1 Purpose

The purpose of this architecture document is to describe the structure, components, and design of the Sales Analytics Dashboard. The architecture supports efficient data processing, storage, and visualization.

1.2 Scope

System Design:

- Detailed description of the overall architecture, including the ETL pipeline, OLAP capabilities, data visualization, and reporting modules.
- Explanation of the logical, process, and deployment views of the system.
- Consideration of modularity, ensuring that components can be updated or modified independently.

Deployment Considerations

- Frontend & Backend Setup: Use React.js for frontend, Django/Flask for backend, and deploy databases.
- Authentication : Role based access
- Testing: Perform unit, integration, and load testing to ensure system stability.
- Deployment of the system

Performance Consideration

- Scalability: Designed to handle growing data volumes across multiple regions and product categories without performance degradation.
- User Management: Handles multiple users with role-based access without performance loss.
- Fast Query Execution: Optimized database queries for quick retrieval of large datasets
- Responsive Design: Provides a seamless user experience across devices

1.3 Definitions, Acronyms, and Abbreviations

- **ETL**: Extract, Transform, Load
- **OLAP**: Online Analytical Processing
- **CRM**: Customer Relationship Management
- **RBAC**: Role-Based Access Control
- **DB**: Database

1.4 References

- Software Requirements Specification (SRS) for Sales Analytics Dashboard.
- IEEE Standard 1471-2000 for System Architecture Descriptions.

2. Architectural Representation

The Sales Analytics Dashboard follows the **4+1 View Model**, which divides the architecture into different views to address various stakeholders' concerns. The model includes:

- **Logical View:** Shows the system's structure and key components for business functions.
- **Implementation View:** Organizes the codebase, modules, and development structure.
- **Process View:** Focuses on processes, concurrency, and performance considerations.
- **Physical View:** Describes the hardware and network setup where the system will be deployed.
- **Use-Case View:** Maps user interactions and functional requirements to system components.

3. Architectural Goals and Constraints

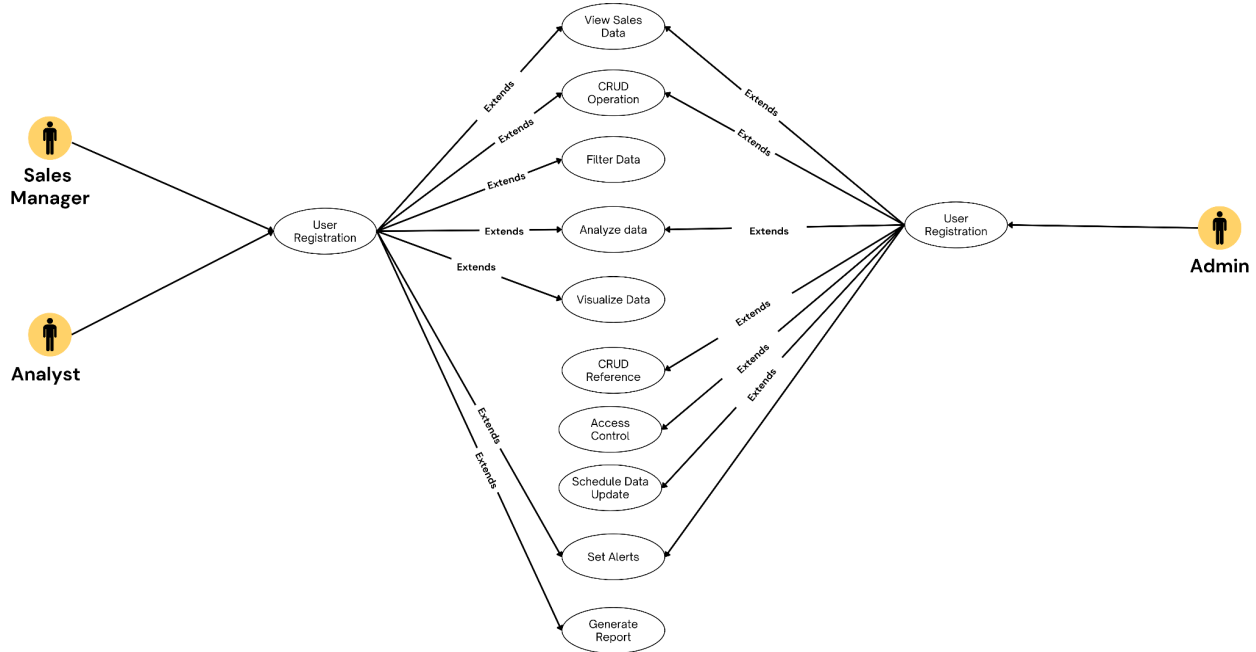
3.1 Goals

- **Scalability:** The architecture must support large volumes of data and a growing number of users.
- **Security:** Ensuring protection against unauthorized access.
- **Availability:** The system should guarantee minimal downtime for maintenance.
- Simple interface offering quick, actionable insights.

3.2 Constraints

- **Storage Capacity:** Large volumes of sales data over time can require significant storage resources, increasing costs.
 - **Scalability Limitations:** Rapid user or data growth may require additional infrastructure upgrades.
 - **Query Performance:** Complex queries on large datasets could slow down response times without proper indexing or optimization.
-

4. Use-Case View



4.1 Architecturally-Significant Use Cases

4.1.1 User Registration and Authentication

- Users must be able to create accounts, log in, and access the dashboard based on their role.
- Upon successful registration, users gain access to the sales dashboard and personalized features.

4.1.2 Sales Data Visualization

- Users can filter and view sales data across various domains.
- Requires integration between the data visualization layer and the data warehouse. The OLAP engine supports data aggregation and presentation
- Alert Notifications: The system generates real-time alerts for significant sales trends or anomalies

4.2 Roles:

- **Admin:** Manages user accounts, roles, and system configurations.
- **Sales Managers:** Upload sales data, view summaries, and generate performance reports.
- **Business Analysts:** Perform detailed analysis using filters, generate customized reports, and deep-dive into sales trends.

5. Logical View

5.1 System Architecture

The system is structured into three primary layers:

1. User Interface (UI) Layer

- **Purpose:** This layer is responsible for interacting with the end users.
- **Subsystems:**
 - **Dashboard UI:** Displays interactive charts, graphs, tables, and filters for sales data.
 - **User Management Interface:** Handles user authentication, login, and role-based access.
 - **Filters and Search:** Allows users to filter and search sales data by categories like region, product, and time period.
- **Interactions:**
 - Receives input from the users (e.g., filters, search queries).
 - Displays data visualization and reports generated by the Business Logic layer.

2. Business Logic Layer

- **Purpose:** This layer handles the core functionality and processes, including business rules, data transformations, and calculations.
- **Subsystems:**
 - **ETL Pipeline:** Extracts data from multiple sources and transforms it into a consistent format.
 - **OLAP Engine:** Enables multidimensional data analysis and querying.
 - **Sales Analytics Engine:** Performs sales trend analysis, customer segmentation, and generates reports for business decision-making.
 - **Access Control Logic:** Implements role-based access control (RBAC) to ensure different user roles can access the appropriate data and functions.
- **Interactions:**
 - Receives requests from the UI layer (e.g., data visualizations, reports).
 - Retrieves or sends data to the Data Access layer.

3. Data Access Layer

- **Purpose:** Manages data storage, retrieval, and indexing to ensure quick access to large datasets.
- **Subsystems:**
 - **Database:** A SQL-based relational database that stores sales data, user information, product data, and other relevant information.
 - **Technologies:** MySQL, optimized for analytical queries.

- **Interactions:**

- Provides raw and aggregated data to the Business Logic layer.
- Executes SQL queries and updates the database with new data.

5.2 Process View

5.2.1 Processes

1. **User Management:** Handles user registration, login, and role-based access.
2. **Data Transformation:** ETL processes that clean and transform sales data into a usable format.
3. **Data Analysis:** OLAP-driven multidimensional analysis for customizable reports.
4. **Data Visualization:** Real-time generation of charts and graphs based on user-selected filters.

5.2.2 Process to Design Elements

User Management:

- **Login Screens:** Allows users to enter credentials (username and password) for authentication.
- **Navigation:** Displays dashboard sections based on user roles after login.
- **Alerts and Notifications:** Notify users if access is denied or login credentials are incorrect.

Authentication Module:

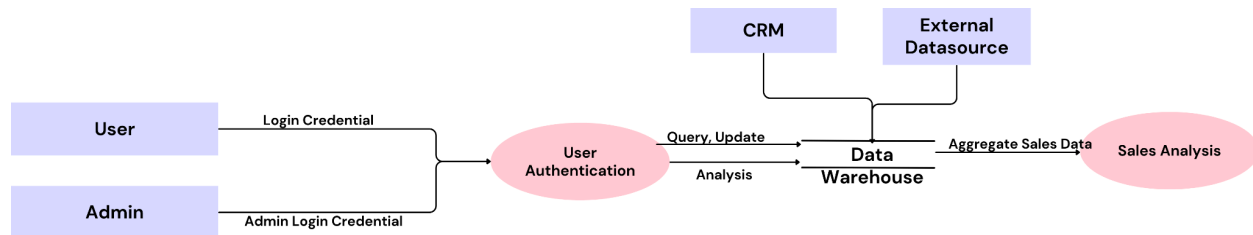
- **User Login:** Verifies credentials through database queries, supporting password.
- **Role-Based Access Control (RBAC):** Grants access to relevant dashboard components based on user roles .
- **Session Management:** Manages active user sessions, ensures security

5.2.3 Process Model to Design

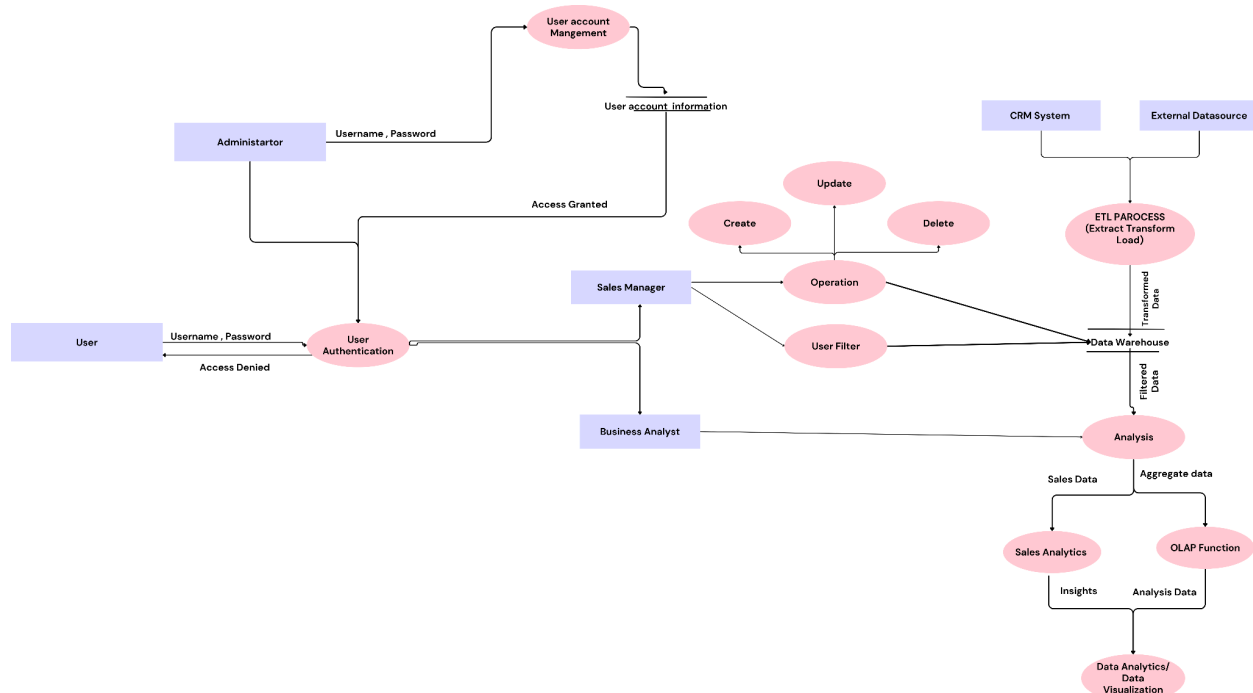
DFD Level 0



DFD Level 1



DFD Level 2



5.2.4 Dependencies

- The UI Layer depends on the Business Logic Layer to fetch processed data.
- The Business Logic Layer depends on the Data Access Layer for fetching and storing sales data.

5.2.4 Process to implementation

- **User Management:** Implemented with Flask for backend authentication
- **Data Transformation:** ETL processes created using Python scripts
- **Data Analysis:** Set up OLAP capabilities using tools like MySQL
- **Data Visualization:** Dashboard components built with React.js, HTML5, CSS interactive visuals.

6. Deployment View

6.1 Physical Deployment Architecture

The system is deployed across several physical components:

- **External Desktop/Client PC:** Hosts the client-side of the web-based UI using modern web browsers (e.g., Chrome, Firefox).
- **Cloud Servers:** Hosts the web application, ETL pipeline, OLAP engine, and database using platforms.
 - **Components:**
 - **Application Server:** Executes business logic and processes user requests.
 - **Database Server:** Stores sales data in the warehouse.
- **Data Warehouse:** Central repository for transformed sales data
- **API Gateway:** Manages integration with CRM, e-commerce platforms

7. Performance

- **Concurrent User Handling:** The system aims to support up to 10,000 concurrent users, ensuring responsive experience even during peak usage times.
- **Data Loading Time:** The dashboard is planned to load data within 2-3 seconds under normal load conditions, enhancing user experience.
- **ETL Performance:** The ETL pipeline is targeted to process new sales data within 15 minutes of receipt.

8. Quality Attributes

- **Security:**
 - Implement robust secure authentication protocols to protect sensitive sales and user data.
- **Maintainability:**
 - Employ modular design principles and clean coding practices to ensure the system is easy to maintain. Comprehensive documentation to coding standards will facilitate future updates and modifications.
- **Usability:**
 - Focus on creating an intuitive and user-friendly interface, ensuring that users can easily navigate and utilize the dashboard features. User feedback will be solicited throughout development to continuously enhance the user experience.