SOCIAL MEDIA ANALYTICS DASHBOARD

SOFTWARE ENGINEERING AND CONCEPTS

DOCUMENTATION

SUBMITTED BY:

JAGANAATH P	220701095
GOUTHAM AK	220701077
ENIYAN P	220701071
HARINE M	220701083
HARSHINI P	220701089
ISHANT R	220701093

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM CHENNAI-602105

2023 - 24

INDEX

OVERVIEW OF THE PROJECT:

What is the problem it is trying to resolve?

- The problem lies in the overwhelming amount of data generated by social media platforms, making it challenging for businesses to extract meaningful insights efficiently. Without proper analysis, they might miss out on valuable opportunities for engagement, marketing, and understanding their audience.

Explanation with Respect to Data:

What data does it deal with?

The project deals with various types of data from social media platforms, including but not limited to:

- User engagement metrics (likes, shares, comments)
- Demographic information (age, gender, location)
- Content performance (most popular posts, trending topics)
- Sentiment analysis (positive, negative, neutral sentiments)
- Competitor analysis (comparison with competitors' performance)

How is this data collected and processed?

- Data is collected using APIs provided by social media platforms or through web scraping techniques. Once collected, it undergoes preprocessing steps like cleaning, normalization, and structuring. Then, it's analyzed using statistical methods, machine learning algorithms, and natural language processing techniques to derive insights.

User Benefits upon Implementation

How would it help users when we implement the system?

The Social Media Analytics Dashboard provides several benefits:

- **Insightful Decision-Making:** Businesses can make informed decisions about their social media strategies based on data-driven insights. They can identify what type of content resonates with their audience, understand peak engagement times, and refine their marketing campaigns accordingly.
- **Competitive Advantage:** By monitoring competitors' social media performance, users can benchmark their own performance and identify areas for improvement. This competitive analysis can lead to strategic advantages in the market.
- **Resource Optimization:** Understanding audience demographics and preferences helps in allocating resources effectively. Users can focus on channels and content formats that yield the highest ROI, thus maximizing their efforts.

SOFTWARE REQUIREMENTS SPECIFICATION

EX.NO.1 **DATE: 20-02-2024**

1.Introduction

1.1 Purpose

The purpose of this document is to describe the software requirements for the development of a Social Media Analytics Dashboard. This dashboard will aggregate, analyze, and visualize data from various social media platforms to provide insights into user engagement, content performance, and audience demographics.

1.2 Scope

The Social Media Analytics Dashboard will be a web-based application aimed at social media managers, marketers, and analysts. It will integrate data from multiple social media platforms (e.g., Facebook, Twitter, Instagram, LinkedIn) and present it in an intuitive and interactive format. The dashboard will offer real-time analytics, historical data analysis, customizable reports, and data export features.

1.3 Definitions, Acronyms, and Abbreviations

- **API**: Application Programming Interface
- **UI**: User Interface
- **UX**: User Experience

- **KPI**: Key Performance Indicator
- **SMA**: Social Media Analytics

2. Overall Description

2.1 Product Perspective

The Social Media Analytics Dashboard is a standalone web application that will integrate with third-party social media APIs to fetch data. It will be developed using a modern web framework and will be accessible from any device with internet connectivity.

2.2 Product Functions

- **Data Integration**: Connect to multiple social media platforms and retrieve data via APIs.
- **Data Processing**: Aggregate and normalize data from different sources.
- **Data Visualization**: Provide interactive charts and graphs to visualize key metrics.
- **Reporting**: Generate customizable reports and allow data export in various formats (PDF, CSV).
- **User Management**: Allow user registration, authentication, and role-based access control.
- **Notifications**: Send alerts and notifications based on predefined criteria.

2.3 User Classes and Characteristics

- **Social Media Managers**: Require detailed insights into engagement and content performance.
- **Marketing Analysts**: Need comprehensive reports for strategic decision-making.
- **Executives**: Require high-level overviews and KPIs.

2.4 Operating Environment

- **Client Side**: Modern web browsers (Chrome, Firefox, Safari, Edge)
- **Server Side**: Cloud-based server running on Linux, with a database and backend service.

2.5 Design and Implementation Constraints

- **Data Privacy**: Must comply with GDPR and other relevant data protection regulations.
- **API Rate Limits**: Adherence to the rate limits imposed by social media APIs.
- **Scalability**: System must handle large volumes of data efficiently.

2.6 Assumptions and Dependencies

- **Availability of APIs**: Continuous access to social media platform APIs.
- **Internet Connectivity**: Reliable internet connection for data retrieval and user access.

- **User Training**: Users have a basic understanding of social media analytics.

3. Appendices

3.1 Appendix A: Glossary

- **Engagement Rate**: A metric that measures the level of interaction that a piece of content receives.
- **OAuth 2.0**: An authorization framework that enables applications to obtain limited access to user accounts on an HTTP service.

3.2 Appendix B: Analysis Models

- **Data Flow Diagrams**
- **Use Case Diagrams**

3.3 Appendix C: Issues List

- **Issue 1**: API rate limit handling.
- **Issue 2**: Data synchronization across different time zones.

4. References

- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications
- RESTful Web Services documentation
- Social Media Platforms API documentation (e.g., Twitter, Facebook, Instagram)

SCRUM METHODOLOGY

EX.NO.2 **DATE: 01-03-2024**

1. Project Vision

Vision Statement:

- The Social Media Analysis Dashboard is designed to provide users with comprehensive insights into social media performance, trends, and engagement.
- It aims to empower businesses, marketers, and social media managers to make data-driven decisions, optimize their strategies, and enhance their online presence.

Goals:

- Enhance Data-Driven Decision Making
- Improve Engagement and Audience Understanding
- Facilitate Competitive Benchmarking
- Optimize Resource Allocation
- Streamline Reporting and Monitoring

The Product Backlog

The product backlog is a prioritized list of features, enhancements, and bug fixes required for the project.

Product Backlog Items:

1. Data Integration
2. Analytics and Metrics
3. Audience Demographics
4. Sentiment Analysis
5. User Interface and Experience
6. User-Friendly Interface
7. Graphical Representations

3.The Scrum Team

1. **Product Owner:** Define the vision of the product, manage the product backlog, and ensure the team delivers value to the business.
2. **Scrum Master**: Facilitate the Scrum process, ensure the team follows Scrum practices, and remove any impediments that hinder progress.
3. **Development Team**: Design, develop, test, and deliver the product increment. Cross-functional team members with skills in front-end development, back-end development, data analysis, and QA.

4.Planning the Sprints

- Sprint Duration: Typically 2-4 weeks.
- Sprint Planning Meeting: The team selects items from the product backlog to commit to during the sprint.

1.Sprint Planning Meeting - Goal: Define what will be delivered in the sprint and how it will be achieved. - Input: Product backlog, team capacity, past performance. - Output: Sprint backlog (tasks for the sprint), sprint goal.

2.Daily Stand-up Meetings - Duration: 15 minutes - Purpose: Discuss what was done yesterday, what will be done today, and identify any impediments.

3.Sprint Execution - Development: Team works on the tasks in the sprint backlog. - Testing: Continuous integration and testing of features.

4.Sprint Review - Purpose: Demonstrate the working product increment to stakeholders. - Activities: Team shows what was accomplished during the sprint. Stakeholders provide feedback

5. Sprint Retrospective - Purpose: Reflect on the sprint and identify improvements for future sprints. - Activities: Discuss what went well, what didn’t, and how to improve.

6.Release Planning - Release Goal: Determine when and what features will be released to the users. - Activities: Prioritize features, finalize the release date, prepare for deployment.

Sprint Breakdown:

Sprint 1: Project Initialization and Basic Setup

- Set up project infrastructure.

- Establish development environment.
- Define user stories and prioritize backlog.

Sprint 2: Expanding Data Integration and Basic UI

- Integrate Twitter API and retrieve basic data (tweets, retweets, likes).
- Create initial UI components for displaying Facebook data.
- Ensure data is updated in real-time.

Sprint 3: Enhanced UI and Additional Metrics

- Develop UI components for displaying Twitter data.
- Implement additional metrics (engagement rates, impressions).
- Add customization options for the dashboard layout.

Sprint 4: Integrating Instagram and Sentiment Analysis

- Integrate Instagram API and retrieve basic data (posts, likes, comments, followers).

- Develop sentiment analysis algorithm and integrate it into the dashboard.Display sentiment analysis results on the UI.

Sprint 5: Audience Demographics and LinkedIn Integration

- Develop features to analyze and display audience demographics.
- Integrate LinkedIn API and retrieve basic data (posts, likes, shares, comments, followers).
- Display LinkedIn data on the UI.

- **Sprint 6: Competitor Analysis and Reporting Features**
- Develop competitor data fetching and comparison tools.
- Create templates for automated reports.
- Implement alert system for significant metric changes.
- Refine and optimize existing features.

Sprint 7: Final Testing, Documentation, and Deployment

- Perform extensive testing (manual and automated).
- Fix identified bugs and issues.
- Write user guides and technical documentation.
- Deploy the final product to the production environment.
- Conduct a final review and retrospective.

USER STORIES

EX.NO.3 **DATE:12-03-2024**

1.Login page

User Story

As a user, I want to log in to the social media analytics dashboard using my existing social media account credentials to access the platform seamlessly**.**

Acceptance criteria

Users must be able to log in to the social media analytics dashboard using their existing social media account credentials without needing to create a new account.
Upon successful login, users should be directed to the dashboard homepage, where they can access their analytics data seamlessly.

2.Audience demographics

User story

As a user I want my content to match to my audience's preferences

Acceptance criteria

The content recommendation system must analyze audience preferences and suggest content that aligns with those preferences**.**

3.Sentimental analysis

User story

As a user I want to analyze the mindset in comments to gauge overall audience reactions and adapt our content strategy

Acceptance criteria

The system must analyze comments and provide sentiment analysis results (positive, negative, neutral) for overall audience reactions.

The analysis should include actionable insights or suggestions for adapting the content strategy based on the sentiment trends.

4.Data visualisation

User story

As a user I want to easily visualize engagement metrics to identify trends and decisions for optimizing my content.

Acceptance criteria

The dashboard must provide clear and interactive visualizations of engagement metrics such as likes, shares, comments, and views.

The visualizations should highlight trends over time and allow users to filter data by date range and other relevant parameters for informed decision-making.

5.Comparative analysis

User story

As a user I want to track the performance of my content over time to understand trends and identify areas for improvement.

Acceptance criteria

The system must provide historical data and performance metrics of content over specified time periods, including metrics such as views, likes, shares, and comments.

Users should be able to view trends and generate reports that highlight performance patterns and areas needing improvement.

6.Segmentation

User story

As a user, I want to be able to segment and analyse the demographic attributes of the audience engaging with my content, including gender, location (state), age group, and community, to gain insights into the specific demographics that are interacting**.**

Acceptance criteria

The system must allow users to segment and analyse audience engagement data by gender, location (state), age group, and community, providing insights into specific demographic interactions.

7.Summary metrics

User story

As a user I want to quickly access high-level metrics for my social media performance to understand overall engagement**.**

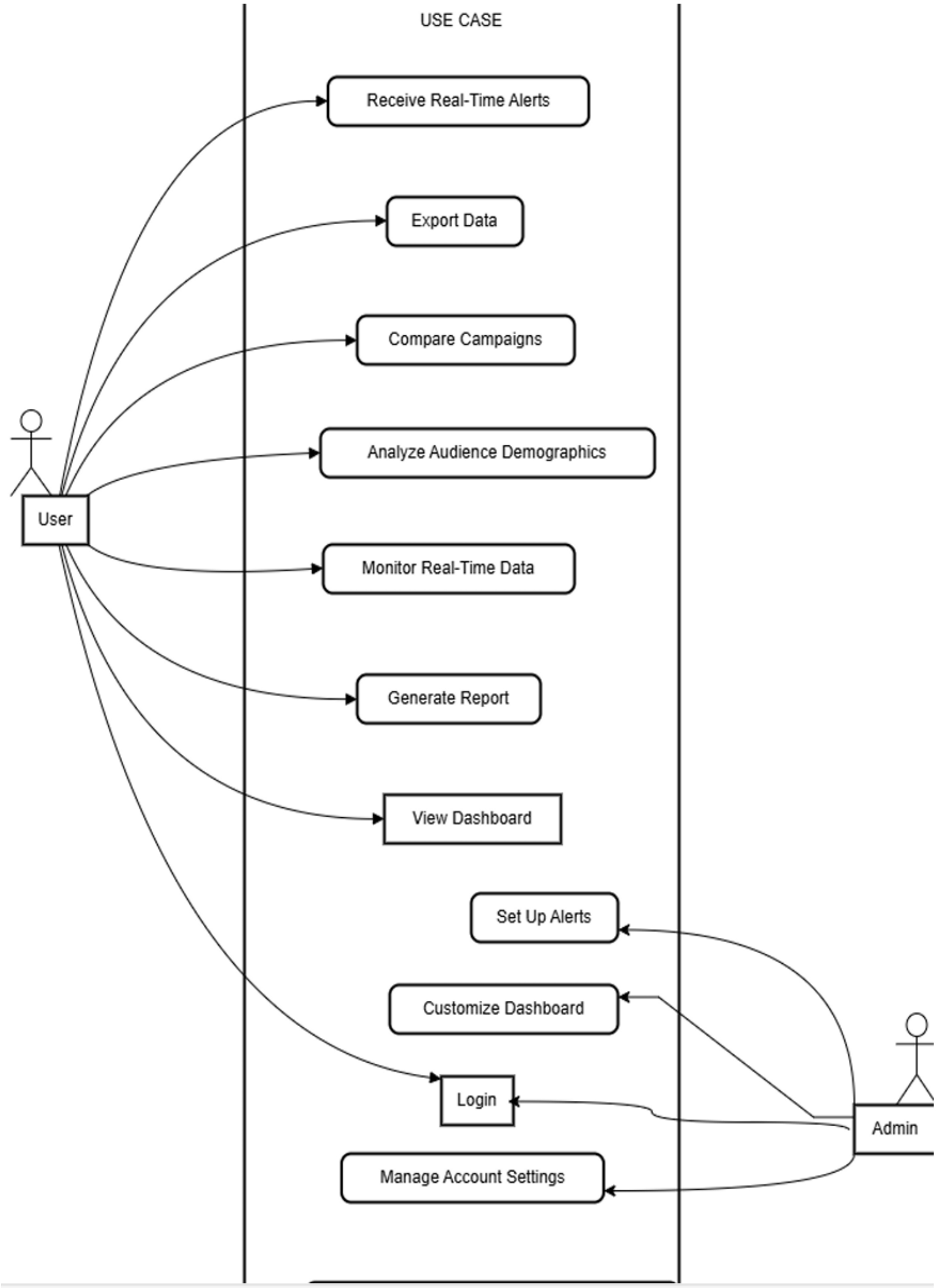
Acceptance criteria

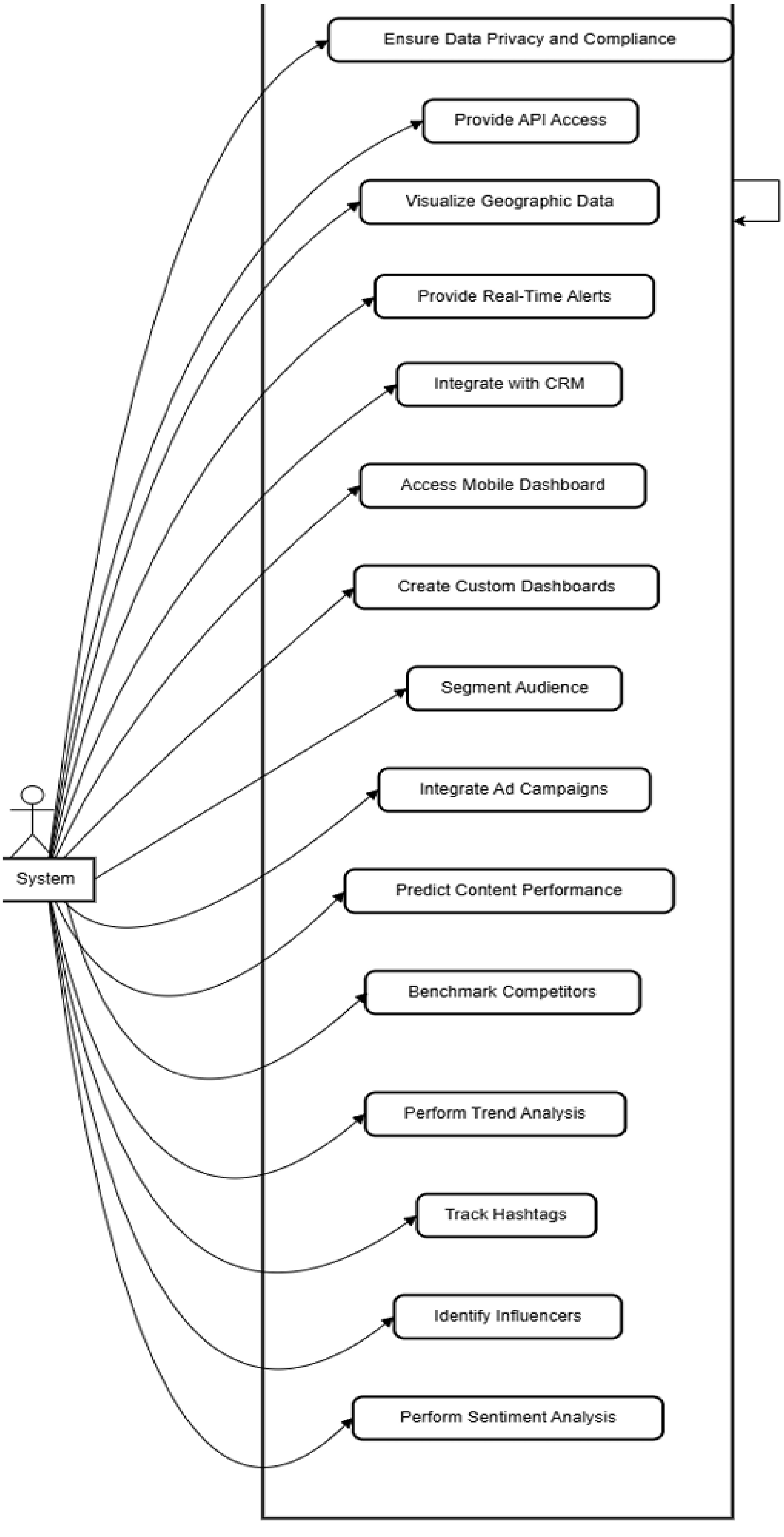
The dashboard must display high-level metrics such as total likes, shares, comments, and views at a glance, providing a quick overview of overall social media performance.

The high-level metrics should be updated in real-time and easily accessible from the main dashboard page.

USE CASE DIAGRAM

EX.NO:4 **DATE:19-03-2024**




Ensure Data Privacy and Compliance
Provide API Access
Visualize Geographic Data
Provide Real-Time Alerts
Integrate with CRM
Access Mobile Dashboard
Create Custom Dashboards
Segment Audience
Integrate Ad Campaigns
Predict Content Performance
Benchmark Competitors
Perform Trend Analysis
Track Hashtags
Identify Influencers
Perform Sentiment Analysis
System

System Features: These are the functionalities provided by the system. They include actions such as logging in, viewing dashboards, generating reports, monitoring real-time data, and many others.

User and Admin: These are actors interacting with the system. The User can perform actions like logging in, viewing dashboards, and generating reports. The Admin can perform similar actions and also manage account settings, customize dashboards, and set up alerts.

Interaction: The arrows indicate interactions between the actors and the system features. For example, the User can log in, view dashboards, and generate reports, while the Admin can log in, manage account settings, customize dashboards, and set up alerts.

System Actor: Positioned on the right side, it represents the system itself, interacting with the various features. This arrangement suggests that the system manages and responds to the actions initiated by the User and Admin.

Non-Functional Requirement(NFR)

EX.NO:5 **DATE:29-03-2024**

1. **Performance:** The dashboard should be responsive and provide quick access to data and insights, even when handling large volumes of social media data. It should be optimized for efficient data retrieval, processing, and visualization, ensuring minimal latency and fast response times.

2. **Scalability:** The dashboard should be scalable to accommodate increasing user loads and growing volumes of social media data over time. It should be capable of handling concurrent user interactions and large datasets without experiencing performance degradation or downtime.

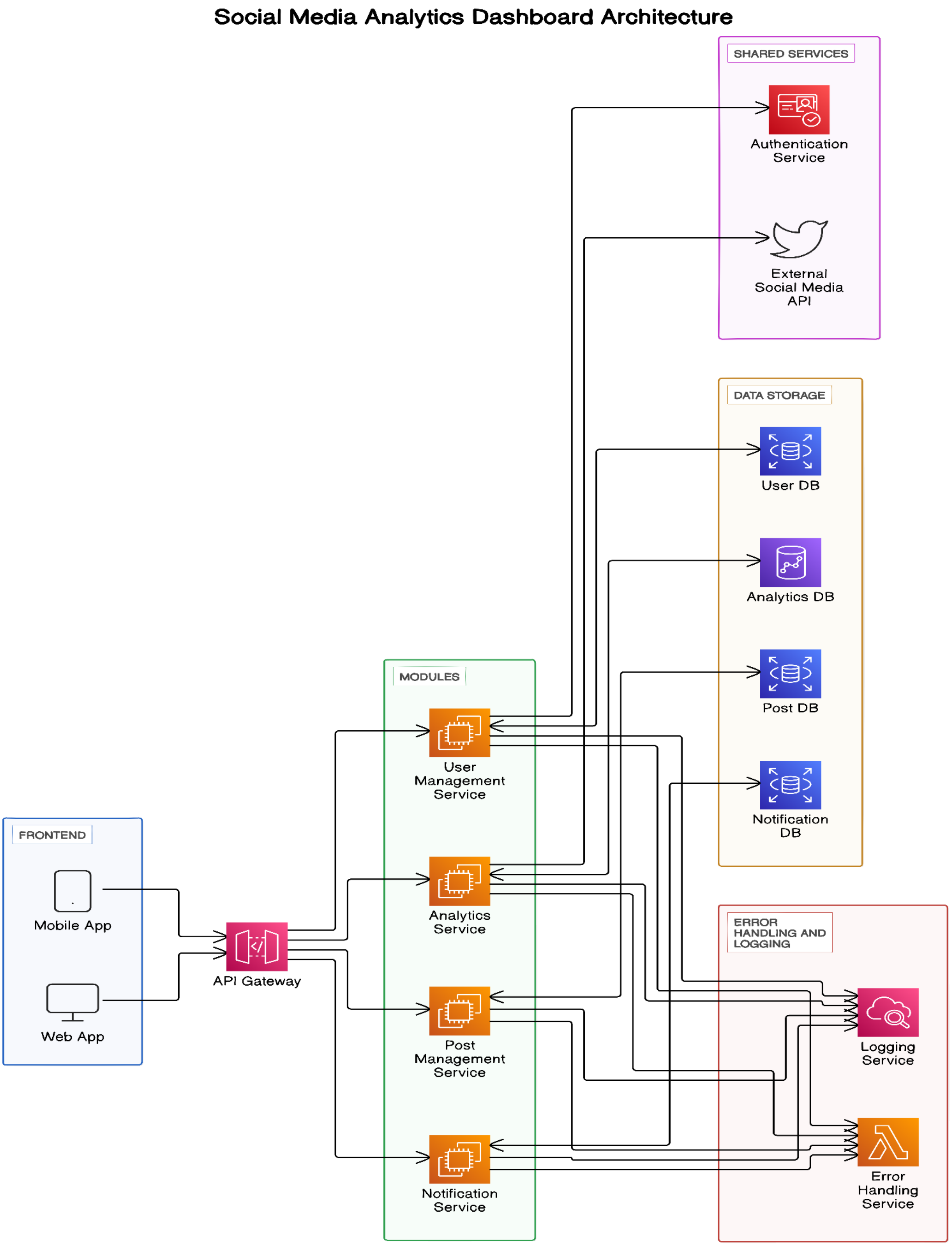
3. **Security:** The dashboard should enforce robust security measures to protect sensitive user data, including social media credentials, analytics results, and user preferences. It should implement authentication, authorization, and encryption mechanisms to ensure data privacy, prevent unauthorized access, and mitigate security risks such as data breaches or cyber attacks.

4. **Reliability:** The dashboard should be highly reliable and available, ensuring uninterrupted access to social media analytics functionalities for users. It should be resilient to failures and errors, with built-in mechanisms for error handling, fault tolerance, and disaster recovery to minimize service disruptions and data loss.

5. **Usability:** The dashboard should be user-friendly and intuitive, catering to the needs of both novice and experienced users. It should have a well-designed user interface with intuitive navigation, interactive visualizations, and customizable dashboards to enhance user experience and facilitate efficient data exploration, analysis, and decision-making.

OVERALL PROJECT ARCHITECTURE

EX.NO:6 DATE:09-04-2024



Frontend:

Mobile App: This is the interface for users accessing the system via mobile devices.

Web App: This is the interface for users accessing the system via web browsers.

Both the Mobile App and Web App interact with the backend through the API Gateway.

API Gateway:

The API Gateway acts as an entry point for all client requests. It routes the requests to the appropriate backend services and modules.

Modules:

User Management Service: Handles user-related operations such as registration, authentication, and profile management. Interacts with the User DB and external Authentication Service.

Analytics Service: Processes and provides analytical data derived from social media interactions. Interacts with the Analytics DB for storing and retrieving analytical data.

Post Management Service: Manages the creation, updating, and deletion of social media posts. Interacts with the Post DB.

Notification Service: Handles notifications sent to users. Interacts with the Notification DB.

Data Storage:

User DB: Stores user-related information.

Analytics DB: Stores analytical data generated by the Analytics Service.
Post DB: Stores data related to social media posts.
Notification DB: Stores notification data to be sent to users.

Shared Services:

Authentication Service: Authenticates users and provides access tokens or other authentication mechanisms. Used by the User Management Service.

External Social Media API: Represents third-party social media platforms (e.g., Twitter, Facebook) from which data is fetched or to which data is posted.

Error Handling and Logging:

Logging Service: Collects and stores logs generated by different services for monitoring and debugging purposes.

Error Handling Service: Manages errors and exceptions that occur within the system, ensuring they are logged and appropriate responses are sent to the users or services.

Theory and Concepts:

Microservices Architecture: The diagram represents a microservices architecture where different functionalities of the application are divided into independent services. Each service can be developed, deployed, and scaled independently.

API Gateway: This is a common pattern in microservices architecture. The API Gateway provides a single entry point for all client requests, simplifying client-side interaction and allowing for centralized management of request routing, rate limiting, authentication, etc.

Data Storage: The separation of databases for different services (User DB, Analytics DB, Post DB, Notification DB) aligns with the microservices principle of having decentralized data management, allowing each service to manage its own data.

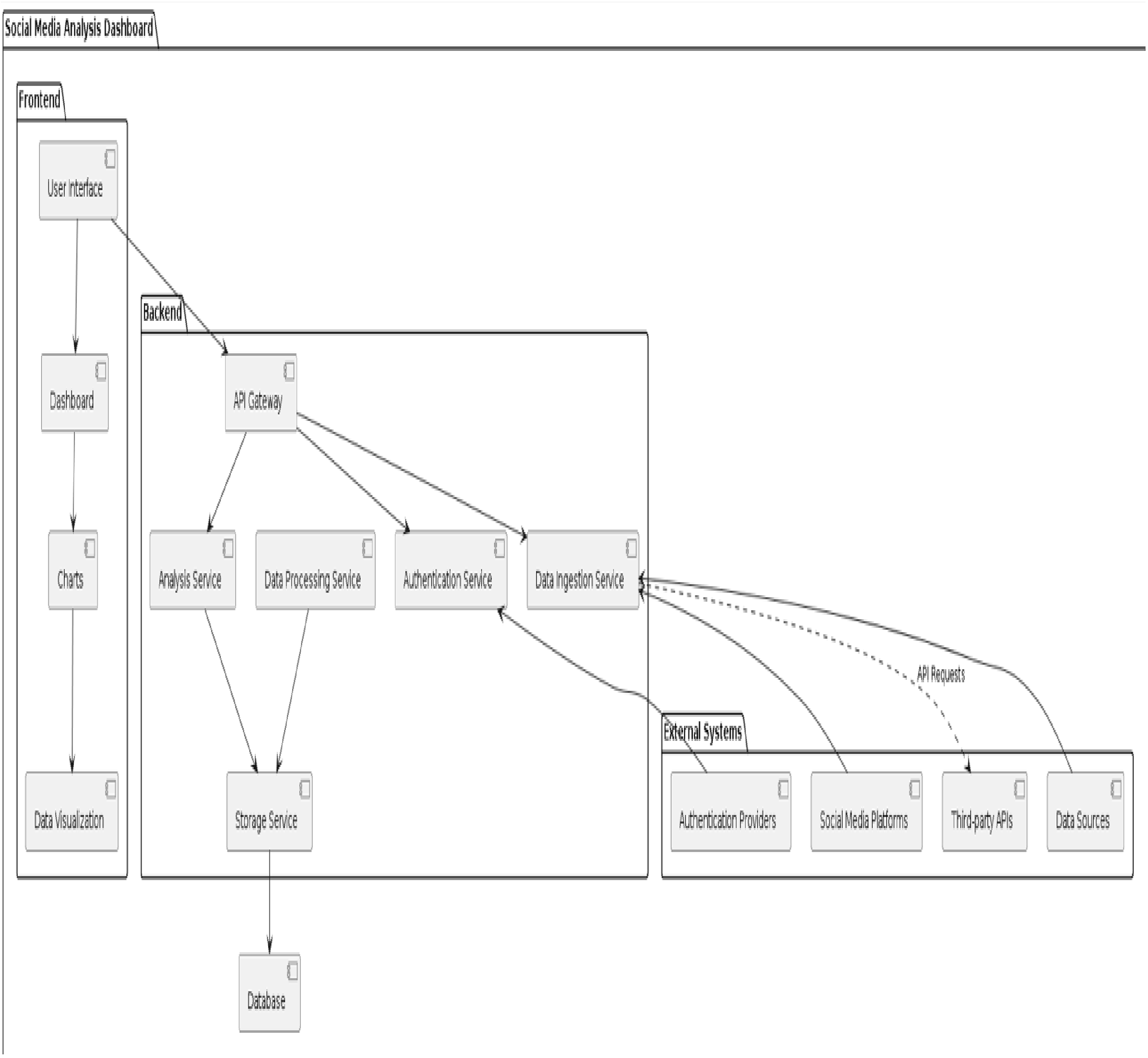
Shared Services: Authentication Service and External Social Media API are shared among multiple modules, promoting reusability and reducing redundancy.

Error Handling and Logging: Dedicated services for logging and error handling ensure that the system is robust and maintainable, facilitating easier troubleshooting and monitoring.

Scalability and Flexibility: The architecture allows for individual components to be scaled according to the load and need. For example, if the Analytics Service is heavily used, it can be scaled independently without affecting other services.

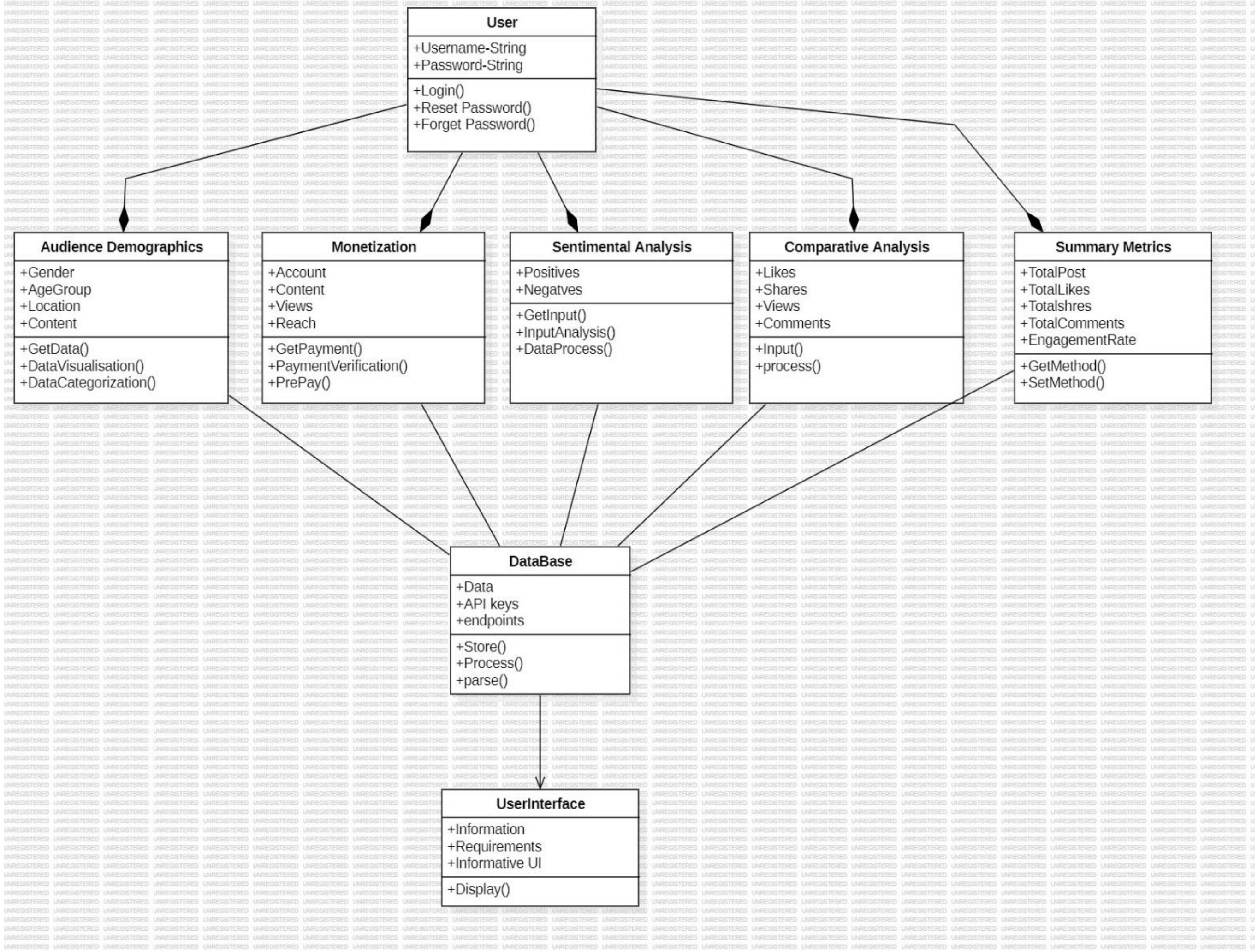
BUSINESS ARCHITECTURE

EX.NO:7 **DATE:19-04-2024**



CLASS DIAGRAM

EX.NO:8 **DATE:30-04-2024**



Class Diagram Overview:

The class diagram depicts the structure of the Social Media Analytics dashboard showcasing the classes involved, their attributes, methods, and relationships.

Explanation of Class Components:

1. **User**

Attributes: +Username-String
+Password-String

Methods: +Login()
+ Reset Password()
+Forget Password()

2. **Audience Demographics**

Attributes: +Gender

-+AgeGroup

-+Location

-+Content

Methods: +GetData0

-+DataVisualisation()

-+DataCategorization()

Purpose: serves the purpose of providing insights into the demographic characteristics of the audience engaging with a particular social media account or content.

3. **Monetization**

Attributes: +Account

-+Content

-+ Views

-+ Reach

Methods:

+GetPayment()

+PaymentVerification()

+PrePay()

Purpose: The purpose of monetization can vary depending on the specific goals and business model of the social media analytics platform.

4. **Sentimental Analysis**

Attributes: +Positives

-+Negatves

-

Methods:+Getinput()

-+InputAnalysis()

-+DataProcess()

Purpose: Analyse the good and bad comments for the user

5. **Comparative Analysis**

Attributes: +Likes

-+Shares

-+ Views

-+Comments

Methods:+Input()

-+process()

Purpose: The purpose of conducting a comparative analysis in the context of a social media analytics dashboard.

6. **Summary Metrics**

Attributes: +TotalPost

-+TotalLikes

-+ Totalshres

-+TotalComments

-+EngagementRate

Methods:+GetMethod()

-+SetMethod()

Purpose: Stores information.

7. **DataBase**

Attributes: +Data

-+API keys|

-+endpoints

Methods:+Store()

-+Process()

-+parse()

8.**UserInterface**

Attributes: + Information

-+ Requirements

-+Informative Ul

Methods: +Display()

Purpose: The purpose of the user interface (UI) in a social media analytics dashboard is multifaceted and crucial for delivering a seamless and effective user experience.

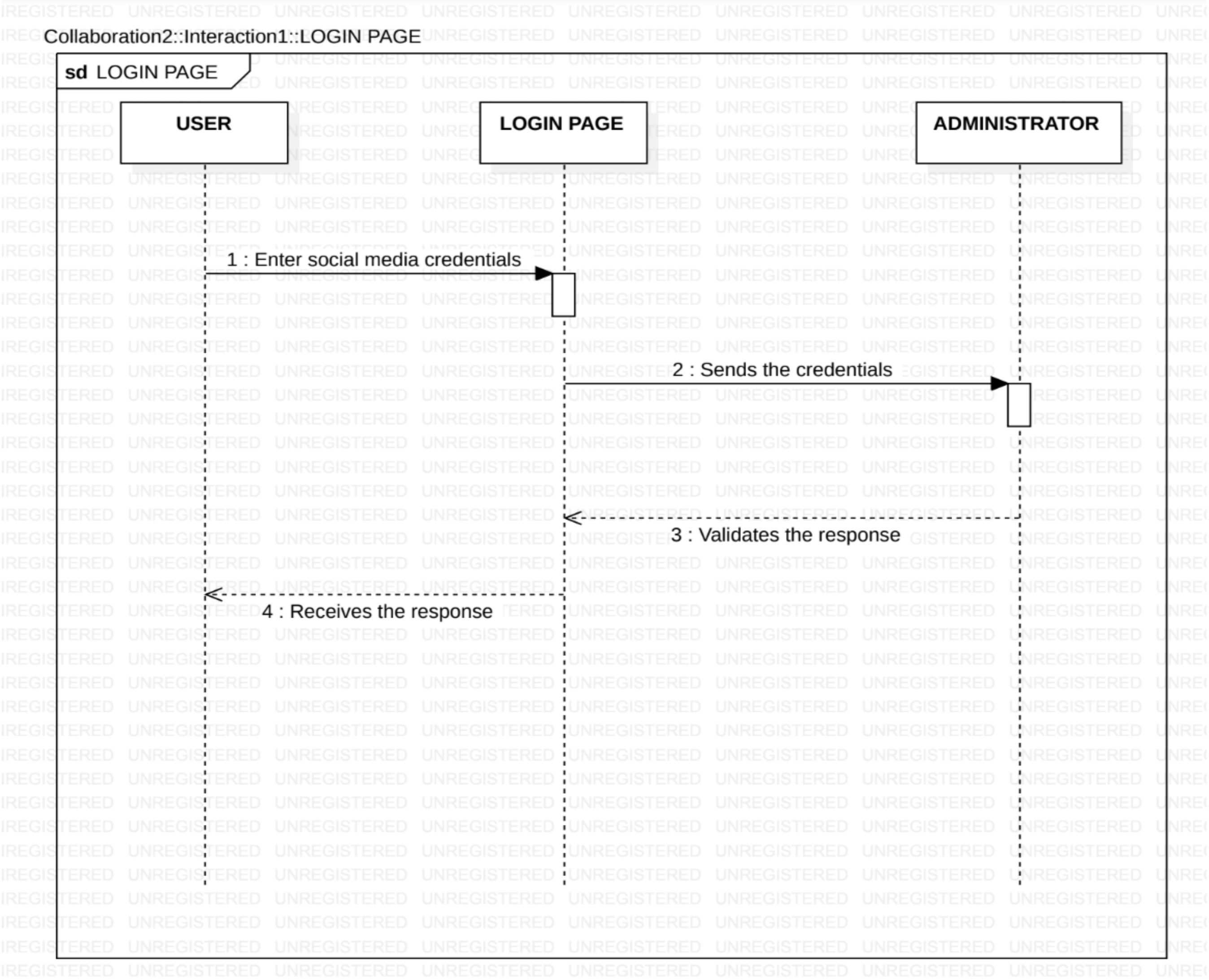
SEQUENCE DIAGRAM

EX.NO:9 **DATE:10-05-2024**

1.User Login Page:

Sequence Diagram Overview:

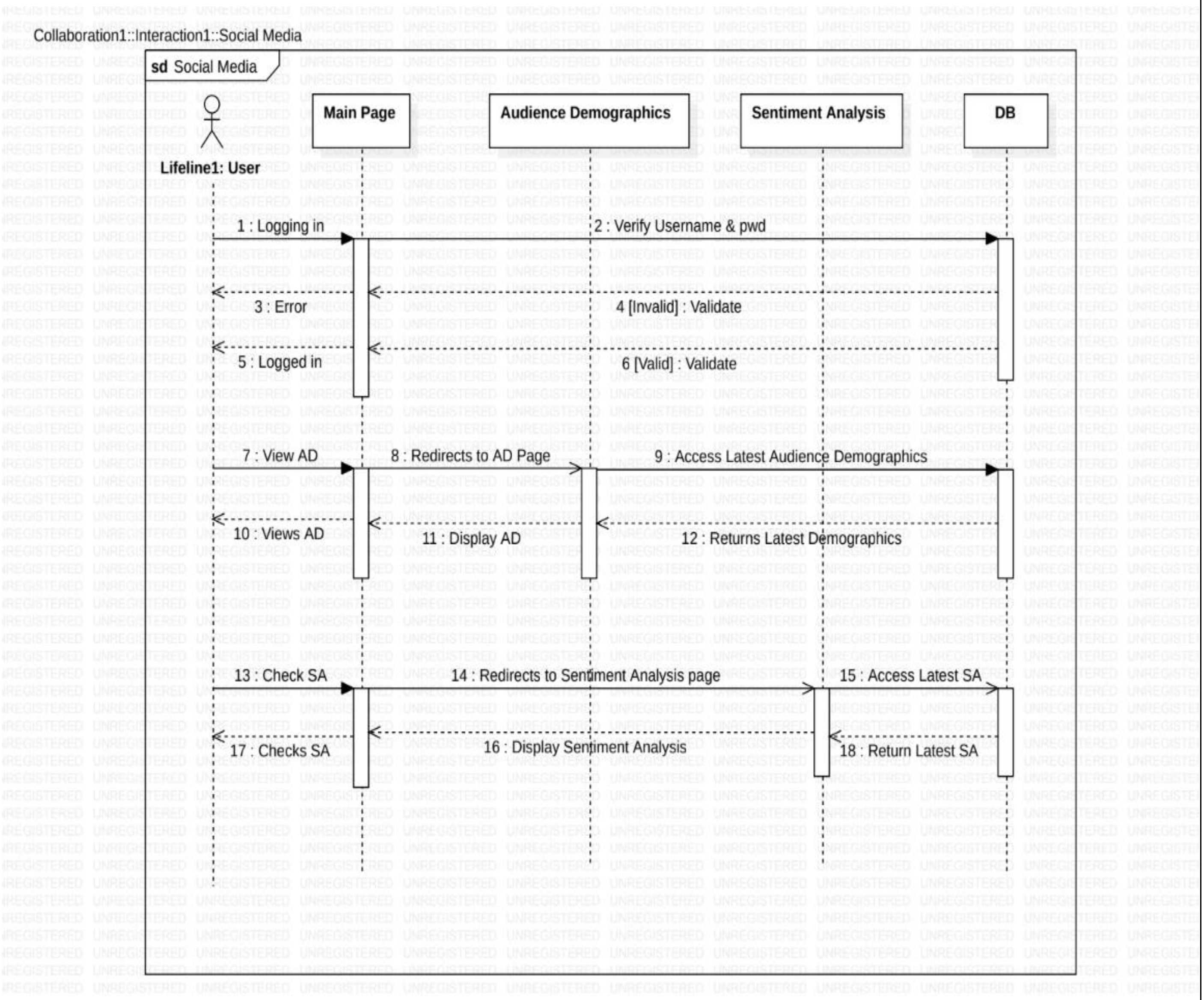
A sequence diagram for a login page illustrates the interactions between the user, the login interface, the authentication server, and the database. It shows the user entering credentials, the interface sending these credentials to the server for verification, and the server querying the database to authenticate the user and return the result to the interface.


Collaboration2::Interaction1::LOGIN PAGE
sd LOGIN PAGE
USER
LOGIN PAGE
ADMINISTRATOR
1 : Enter social media credentials
2 : Sends the credentials
3 : Validates the response
4 : Receives the response

2.AudienceDemoGraphics and Sentimental Analysis:

Sequence Diagram Overview:

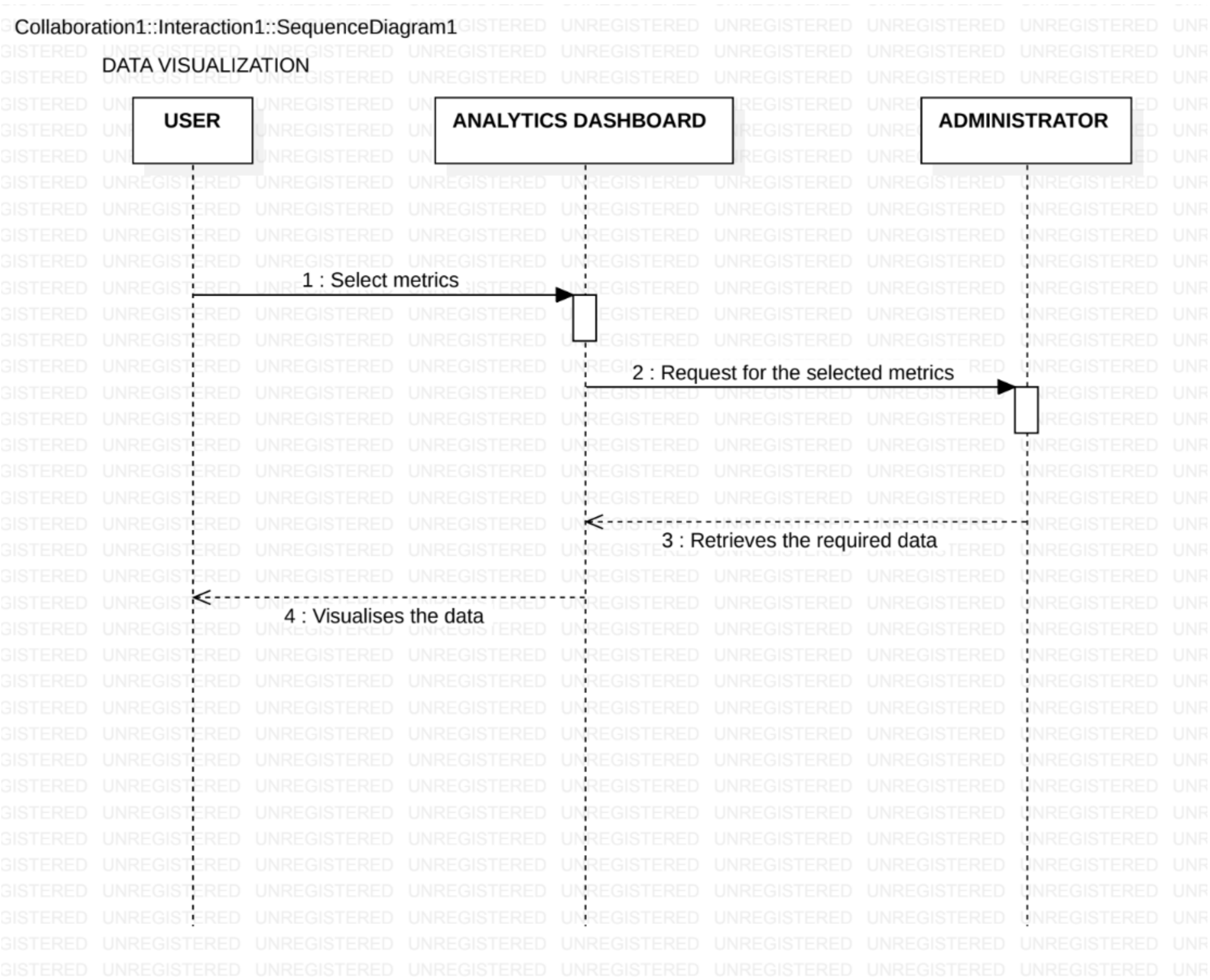
This sequence diagram illustrates the process flow for a Audience Demographics and sentimental analysis that stores in database and analyse to give upon the results to the user



3.Data Visualisation:

Sequence Diagram Overview:

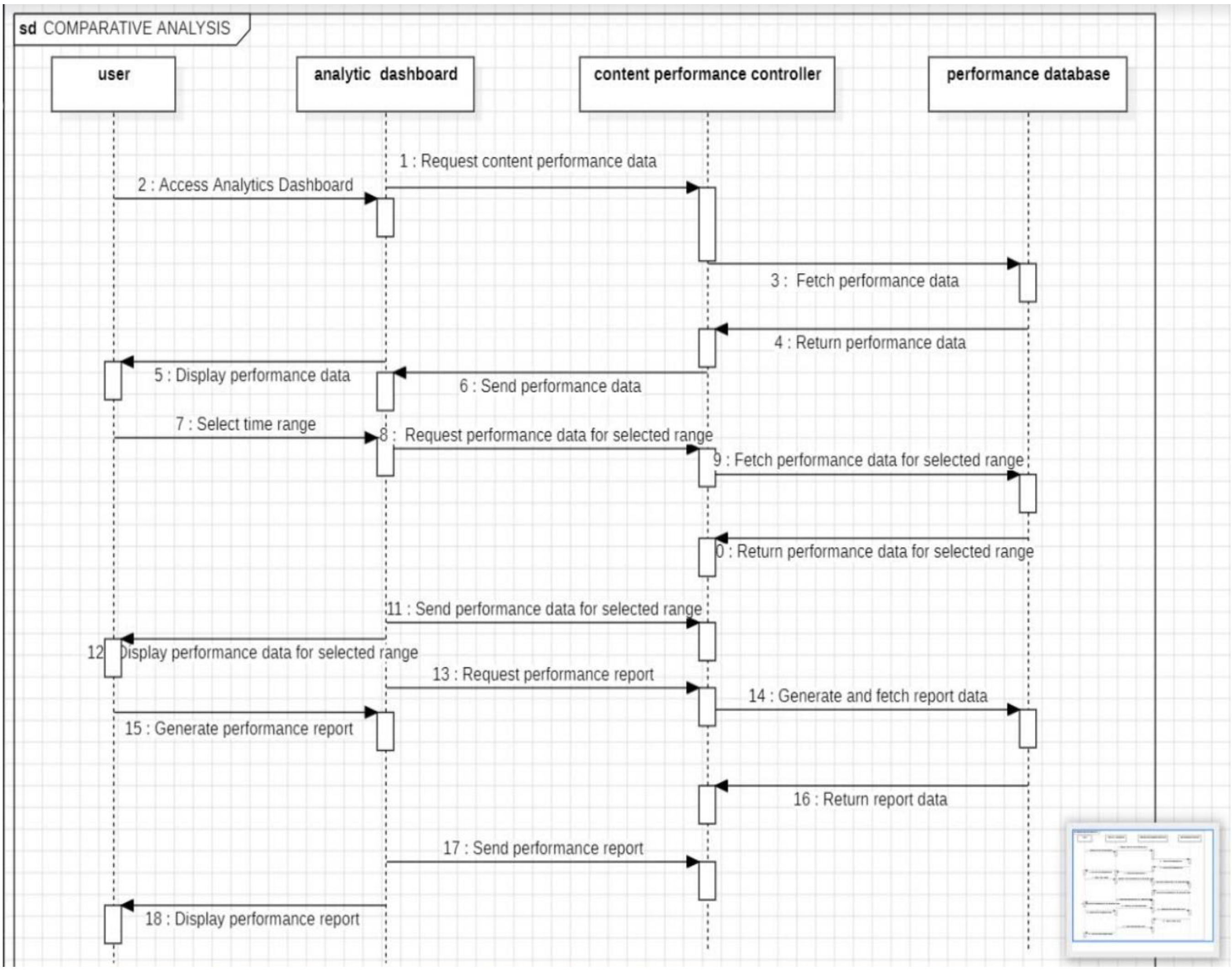
A sequence diagram for data visualization illustrates the interactions between various system components as they work together to process and display data. It typically includes the steps of data retrieval, processing, and rendering on the user interface, showing the flow of data and control between objects.


Collaboration1::Interaction1::SequenceDiagram1
DATA VISUALIZATION
USER
ANALYTICS DASHBOARD
ADMINISTRATOR
1 : Select metrics
2 : Request for the selected metrics
3 : Retrieves the required data
4 : Visualises the data

4.Comparative analysis:

Sequence Diagram Overview:

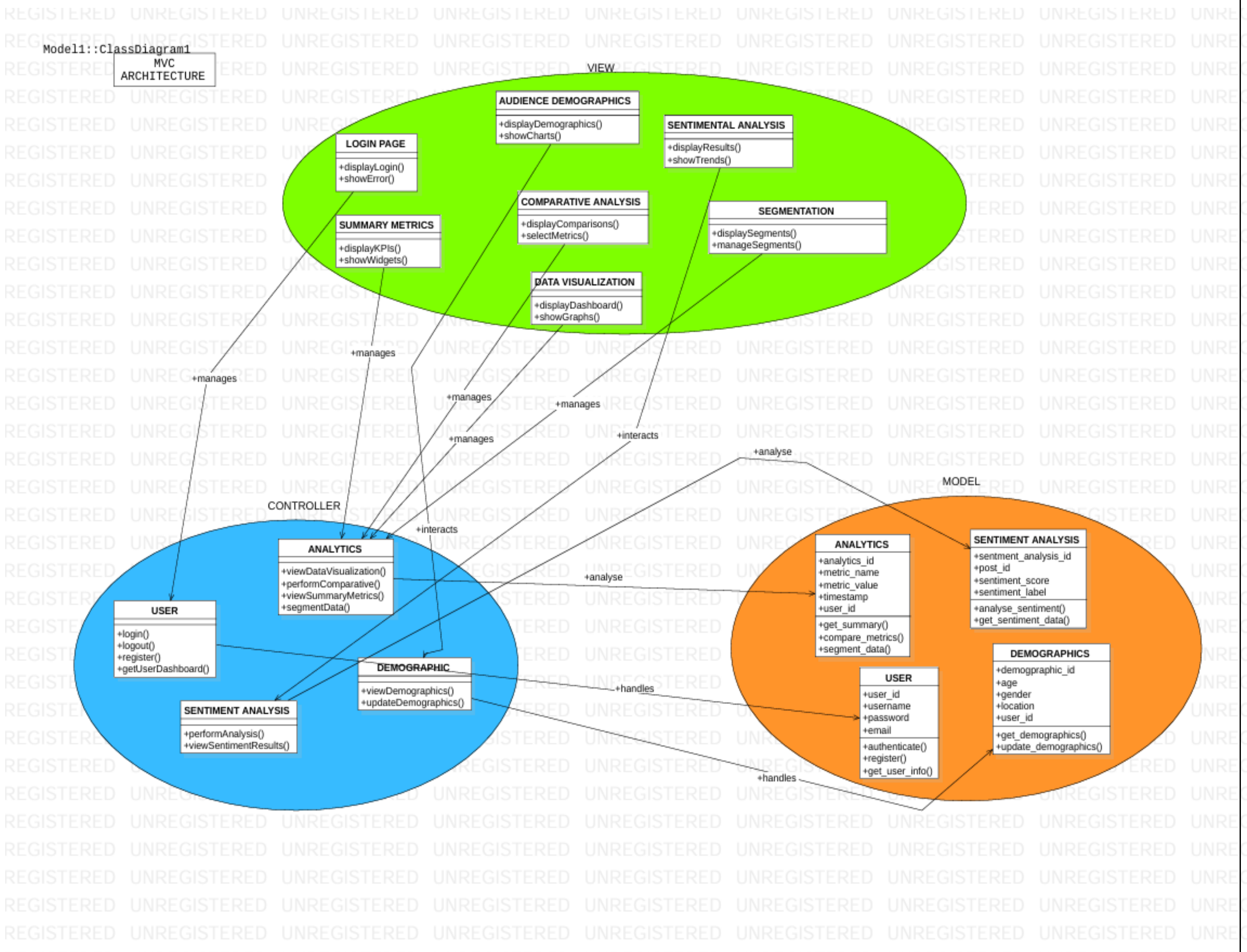
A sequence diagram provides a visual representation of interactions between objects or components in a system, facilitating comparative analysis of system behavior and performance.


sd COMPARATIVE ANALYSIS
user
analytic dashboard
content performance controller
performance database
1 : Request content performance data
2 : Access Analytics Dashboard
3 : Fetch performance data
4 : Return performance data
5 : Display performance data
6 : Send performance data
7 : Select time range
8 : Request performance data for selected range
9 : Fetch performance data for selected range
0 : Return performance data for selected range
11 : Send performance data for selected range
12 Display performance data for selected range
13 : Request performance report
14 : Generate and fetch report data
15 : Generate performance report
16 : Return report data
17 : Send performance report
18 : Display performance report

ARCHITECTURAL PATTERNS

EX.NO.10 **DATE:17-05-2024**

MODEL VIEW CONTROLLER ARCHITECTURE:



MODEL

- **USER**
  - Handles data-related logic such as user authentication, registration, and retrieving user information.Represents a user with attributes: user_id, username, password, email.
  - Methods for CRUD operations and authentication: authenticate(), register(), get_user_info()

- **DEMOGRAPHICS**
    - Handles retrieval and update of demographic data.Represents audience demographics with attributes: demographic_id, age, gender, location ,user_id
    - Methods for CRUD operations on demographic data: get_demographics(), update_demographics()
- **SENTIMENT ANALYSIS**
    - Analyzes sentiment from social media posts and retrieves sentiment data.Represents sentiment analysis data with attributes: sentiment_id, post_id sentiment_score , sentiment_label
    - Methods for CRUD operations and sentiment analysis: analyze_sentiment(), get_sentiment_data()
- **ANALYTICS**
    - Retrieves summary metrics, compares metrics, and segments data..Represents analytics data with attributes: analytics_id, metric_name,metric_value, timestamp ,user_id
    - Methods for CRUD operations and data analytics: get_summary(),compare_metrics(),segment_data()

<u>VIEW</u>

- **LOGIN PAGE**
    - Presents the login form and handles user input. Displays errors if authentication fails.Responsible for displaying login details with methods: displayLogin(),showError()
    -

- **AUDIENCE DEMOGRAPHICS**
  - Displays demographic data in a user-friendly manner, including charts and tables.Responsible for displaying audience demographics with methods:displayDemographics(),showCharts()
- **SENTIMENTAL ANALYSIS**
  - Presents sentiment analysis results, including trends and charts.Responsible for displaying sentiment analysis results with methods: displayResults(),showTrends()
- **DATA VISUALIZATION**
  - Displays data visualizations such as graphs and charts.Responsible for displaying data visualizations with methods: displayDashboard(), showGraphs()
- **COMPARATIVE ANALYSIS**
  - Responsible for displaying comparative analysis with methods: displayComparisons(),selectMetrics()
- **SUMMARY METRICS**
  - Displays data visualizations such as graphs and charts.Responsible for displaying summary metrics with methods: displayKPIs(),showWidgets()
- **SEGMENTATION**
  - Responsible for displaying segmented data with methods: displaySegments(),manageSegments()

<u>CONTROLLER:</u>

USER

  - Handles user interactions related to users with methods: login(),logout(),register(),getUserDashboard()

- **DEMOGRAPHIC**
  - Handles user interactions related to audience demographics with methods: viewDemographics(),updateDemographics()
- **SENTIMENTAL ANALYSIS**
  - Handles user interactions related to sentiment analysis with methods: performAnalysis(),viewSentimentResults()
- **ANALYTICS**
  - Handles user interactions related to data visualization and analytics with methods: viewDataVisualization(),performComparative(),viewSummaryMetrics( ),segmentData()

INTERACTION FLOW

1. **User Authentication**:
   - User accesses LoginPageView.
   - Submits login credentials.
   - UserController handles login request.
   - UserModel authenticates the user.
   - UserController updates LoginPageView with the result.
2. **Viewing/Updating Demographics**:
   - User accesses AudienceDemographicsView.
   - DemographicsController retrieves data from DemographicsModel.
   - Data displayed on AudienceDemographicsView.

- User updates demographics.
- DemographicsController processes the update.
- DemographicsModel updates data.
- AudienceDemographicsView is refreshed.

3. **Performing Sentiment Analysis**:
   - User selects a post for analysis.
   - SentimentAnalysisController initiates analysis.
   - SentimentAnalysisModel performs analysis.
   - Results displayed on SentimentAnalysisView.
4. **Data Visualization**:
   - User accesses DataVisualizationView.
   - AnalyticsController retrieves metrics from AnalyticsModel.
   - Data displayed on DataVisualizationView.
   - User requests comparative analysis.
   - AnalyticsController handles comparison.