

System Design Document

Project Milestone 3

AI-Powered Forex Trading Analysis System

CSC-225 – Software Engineering

Department of Computer Science

Namal University, Mianwali

Prepared by:

Saad Nawaz

Shahbaz Ali

Hafsa Gul

Submitted to:

Course Instructor

CSC-225 Software Engineering

January 18, 2026

Contents

1	Introduction	6
1.1	Purpose	6
1.2	Scope	6
1.3	Design Methodology	6
1.4	Reference Documents	7
1.5	Document Organization	7
2	Design Assumptions and Constraints	8
2.1	Design Assumptions	8
2.2	Design Constraints	8
2.2.1	Platform Constraints	8
2.2.2	Performance Constraints	9
2.2.3	Security Constraints	9
2.2.4	Usability Constraints	10
3	Key Design Decisions	10
3.1	Architectural Style	10
3.2	Data Management Strategy	10
3.3	Real-Time Communication	10
3.4	AI Model Integration	10
3.5	Database Design	11
3.6	API Management	11
3.7	Authentication Mechanism	11
4	Behavioral Diagrams	11
4.1	Use Case Diagram	11
4.2	Data Flow Diagrams	12
4.2.1	Level 0 – Context Diagram	12
4.2.2	Level 1 – Major System Processes	13
4.2.3	Level 2 – Data Retrieval Process Detail	14
4.3	Sequence Diagrams	15
4.3.1	Sequence Diagram 1 – User Authentication	15
4.3.2	Sequence Diagram 2 – Data Retrieval and Analysis	16
4.3.3	Sequence Diagram 3 – Alert Management	17
4.4	Activity Diagrams	18
4.4.1	Activity Diagram 1 – Authentication & Account Management	18
4.4.2	Activity Diagram 2 – Data Retrieval & Analysis	19

4.4.3	Activity Diagram 3 – Customization Workflows	20
4.4.4	Activity Diagram 4 – Reports & Alerts	21
4.4.5	Activity Diagram 5 – Administration & Error Handling	22
5	Structural Diagrams	23
5.1	Class Diagram	23
5.2	Component Diagram	25
5.2.1	Component Diagram 1 – Presentation Layer	25
5.2.2	Component Diagram 2 – Business Logic Layer	26
5.2.3	Component Diagram 3 – Complete System Architecture	27
6	Requirements–Design Traceability	28
6.1	Functional Requirements Traceability	28
6.2	Non-Functional Requirements Traceability	49
6.3	Diagram Coverage Matrix	49
7	Prototype Screen Coverage	50
7.1	Prototype–Requirements Mapping	50
7.2	Key Prototype Features	51
8	Project Resources	52
8.1	GitHub Repository	52
8.2	Figma Prototype	52
8.3	Collaboration Tools	52
9	Meeting Summary	53
9.1	Meeting 1 – Initial Design Review	53
9.2	Meeting 2 – Prototype Demonstration	53
9.3	Meeting 3 – Design Validation & Traceability Review	54
10	Design Notes and Considerations	55
10.1	Scalability Considerations	55
10.2	Security Measures	55
10.3	Performance Optimization	55
10.4	Maintenance and Monitoring	56
10.5	Future Enhancements	56
11	Conclusion	56
11.1	Design Completeness	56
11.2	Verification Summary	57
11.3	Next Steps	57

11.4 Project Team Acknowledgment	57
--	----

List of Figures

1	Use Case Diagram – System-Actor Interactions	29
2	DFD Level 0 – Context Diagram	30
3	DFD Level 1 – Major System Processes	31
4	DFD Level 2 – Data Retrieval & API Management Detail	32
5	Sequence Diagram 1 – User Registration and Login	33
6	Sequence Diagram 2 – Data Retrieval, Analysis, and Visualization	34
7	Sequence Diagram 3 – Alert Creation and Monitoring	35
8	Activity Diagram 1 – Authentication & Account Management	36
9	Activity Diagram 2 – Data Retrieval & Analysis Workflow	37
10	Activity Diagram 3 – Indicator Customization & Preferences	38
11	Activity Diagram 4 – Reports & Alert Management	39
12	Activity Diagram 5 – Administration & Error Handling	40
13	Class Diagram – System Class Structure	41
14	Component Diagram 1 – Presentation Layer Components	42
15	Component Diagram 2 – Business Logic Layer	43
16	Component Diagram 3 – Complete System Architecture	44

List of Tables

1	Functional Requirements–Design Traceability Matrix	45
2	Non-Functional Requirements Traceability Matrix	49
3	Design Artifact Coverage Matrix	50
4	Prototype Screen–Requirements Mapping	51

1 Introduction

1.1 Purpose

The purpose of this Design Document is to provide a detailed system design for the **AI-Powered Forex Trading Analysis System**. This document translates the approved Software Requirements Specification (SRS) into comprehensive design models that guide system implementation. It serves as a reference for developers, instructors, and stakeholders to understand how system requirements are realized through architectural, behavioral, and structural designs.

This document ensures full traceability between requirements and design artifacts, providing a blueprint for successful system implementation and validation.

1.2 Scope

This design document covers the complete system design of the AI-Powered Forex Trading Analysis System, including:

- **Behavioral Models:** Use Case Diagrams, Data Flow Diagrams (Levels 0, 1, and 2), Sequence Diagrams, and Activity Diagrams
- **Structural Models:** Class Diagrams and Component Diagrams
- **Requirements Traceability:** Complete mapping between functional/non-functional requirements and design artifacts
- **Prototype Coverage:** UI/UX screen designs mapped to system requirements
- **Design Validation:** Verification of requirement coverage and design completeness

1.3 Design Methodology

The design process follows a systematic approach:

1. **Requirements Analysis:** Review and validation of approved SRS document
2. **Architectural Design:** Definition of system components and their interactions
3. **Behavioral Modeling:** Creation of use case, sequence, activity, and data flow diagrams
4. **Structural Modeling:** Development of class and component diagrams
5. **Traceability Mapping:** Ensuring each requirement maps to design artifacts

6. **Prototype Design:** Creating interactive UI mockups aligned with requirements

7. **Design Review:** Validation with stakeholders and requirement providers

1.4 Reference Documents

- Software Requirements Specification (SRS) Document – AI-Powered Forex Trading Analysis System
- IEEE Std 1016-2009: Software Design Description
- UML 2.5 Specification
- Draw.io UML Diagramming Tool Documentation
- Figma UI/UX Prototyping Guidelines
- W3C Web Standards and Best Practices

1.5 Document Organization

This document is organized into the following sections:

- Section 2: Design assumptions and constraints
- Section 3: Key design decisions and rationale
- Section 4: Behavioral diagrams (use case, DFD, sequence, activity)
- Section 5: Structural diagrams (class and component)
- Section 6: Requirements-design traceability matrices
- Section 7: Prototype screen coverage
- Section 8: Project resources and collaboration details
- Section 9: Meeting summaries and validation records
- Section 10: Conclusion and next steps

2 Design Assumptions and Constraints

2.1 Design Assumptions

The following assumptions guide the system design:

1. Users possess basic knowledge of Forex trading concepts and financial terminology
2. The system is accessed through modern web browsers (Chrome, Firefox, Safari, Edge) with JavaScript enabled
3. Reliable third-party Forex market and economic data APIs are available with acceptable uptime
4. Internet connectivity is stable during system usage with minimum 5 Mbps bandwidth
5. Users authenticate using secure login mechanisms with valid credentials
6. The system is expected to support up to 10,000 concurrent users under normal load
7. AI-based analysis models provide reasonably accurate market insights based on available historical and real-time data
8. OAuth 2.0 or API key-based authentication is used for securing external API calls
9. Backend infrastructure remains operational 24/7 with redundancy mechanisms for critical failures
10. Users have devices with minimum screen resolution of 1024x768 pixels
11. Database transactions maintain ACID properties for data consistency

2.2 Design Constraints

2.2.1 Platform Constraints

- The system is developed as a web-based application accessible through standard web browsers
- Frontend technologies are restricted to modern JavaScript frameworks such as React.js or Angular
- Backend services are implemented using Python-based frameworks (Flask or Django) or Node.js
- MySQL is used as the primary relational database management system

- External Forex and economic data APIs are subject to availability, latency, and rate-limit constraints imposed by third-party providers
- Cloud hosting platforms (AWS, Azure, or Google Cloud) must support containerized deployments

2.2.2 Performance Constraints

- Standard user interactions must respond within 2 seconds under normal operating conditions
- AI-based analysis and prediction processes must complete within 5 seconds
- Historical data queries must return results within 10 seconds
- Real-time market data updates must be reflected on the user interface within 1–3 seconds
- System must handle 10,000 concurrent users with less than 5% performance degradation
- Database queries must be optimized with proper indexing for sub-second response times

2.2.3 Security Constraints

- All communication between client and server must be secured using TLS 1.3 encryption
- User authentication and authorization must follow role-based access control (RBAC) policies
- User passwords must be stored using secure hashing algorithms such as bcrypt with salt
- Sensitive data, including API keys and user information, must be stored in encrypted form
- Session tokens must expire after 30 minutes of inactivity
- GDPR compliance required for user data collection, storage, and processing
- Regular security audits and penetration testing must be conducted

2.2.4 Usability Constraints

- Interface must be responsive and support mobile, tablet, and desktop devices
- System must provide contextual help and tooltips for complex features
- Accessibility standards (WCAG 2.1 Level AA) must be followed
- Maximum three clicks to reach any major feature from the dashboard

3 Key Design Decisions

3.1 Architectural Style

Decision: Adopt a layered architecture with microservices-ready design.

Rationale: This approach provides clear separation of concerns, enables independent scaling of components, and facilitates future migration to microservices architecture as the system grows.

3.2 Data Management Strategy

Decision: Implement caching layer with Redis for frequently accessed data.

Rationale: Reduces API calls to external services, improves response times, and minimizes costs associated with third-party API usage. Cache invalidation policies ensure data freshness.

3.3 Real-Time Communication

Decision: Use WebSocket protocol for real-time market data updates.

Rationale: Provides bidirectional, low-latency communication essential for delivering real-time forex data and price alerts to users without constant polling.

3.4 AI Model Integration

Decision: Integrate AI analysis engine as a separate component with RESTful API interface.

Rationale: Allows independent development, testing, and deployment of AI models. Enables model updates without affecting other system components.

3.5 Database Design

Decision: Use relational database (MySQL) with normalized schema.

Rationale: Forex trading data has structured relationships (users, analyses, alerts, preferences). ACID properties ensure data consistency for financial data.

3.6 API Management

Decision: Implement API Gateway pattern with fallback mechanisms.

Rationale: Centralizes API management, provides load balancing, implements circuit breaker pattern for fault tolerance, and ensures system availability even when primary APIs fail.

3.7 Authentication Mechanism

Decision: JWT-based stateless authentication with refresh tokens.

Rationale: Enables horizontal scaling, reduces server-side session storage, and provides secure, industry-standard authentication mechanism.

4 Behavioral Diagrams

This section presents the behavioral models that illustrate how the system functions, including user interactions, data flows, and process workflows.

4.1 Use Case Diagram

The Use Case Diagram (Figure 1) shows the complete set of interactions between actors (Guest User, Registered User, Administrator) and the system. It identifies 41 use cases organized into functional groups:

- **Authentication & Account Management:** User registration, login, logout, guest access
- **Data Retrieval & Analysis:** Historical data retrieval, real-time updates, technical indicators, AI predictions
- **Customization:** Indicator customization, user preferences, multi-timeframe analysis
- **Alerts & Notifications:** Price alerts, indicator alerts, alert management
- **Reporting:** Report generation, data export
- **Administration:** System monitoring, user management, API analytics

4.2 Data Flow Diagrams

Data Flow Diagrams represent how data moves through the system at different levels of abstraction.

4.2.1 Level 0 – Context Diagram

Figure 2 presents the context-level view of the system as a single process interacting with external entities.

External Entities:

- Guest User, Registered User, Administrator
- Forex Data API
- Economic Events API
- User Database

4.2.2 Level 1 – Major System Processes

Figure 3 decomposes the system into seven major processes:

Major Processes:

1. **P1.0 – User Authentication & Account Management**
2. **P2.0 – Data Retrieval & API Management**
3. **P3.0 – Technical Analysis Processing**
4. **P4.0 – AI-Based Market Analysis**
5. **P5.0 – Visualization & Reporting**
6. **P6.0 – Alert Management & Notifications**
7. **P7.0 – System Administration & Monitoring**

4.2.3 Level 2 – Data Retrieval Process Detail

Figure 4 provides detailed breakdown of the Data Retrieval process (P2.0):

Sub-processes:

- P2.1 – API Request Handler
- P2.2 – Historical Data Retrieval
- P2.3 – API Response Validation
- P2.4 – API Fallback Handler
- P2.5 – Economic Events Data Retrieval
- P2.6 – Cache Management

4.3 Sequence Diagrams

Sequence diagrams illustrate object interactions over time for critical system workflows.

4.3.1 Sequence Diagram 1 – User Authentication

Figure 5 shows the registration and login processes with session management.

Key Interactions:

- User credential validation
- Password hashing and verification
- Session token generation
- Database persistence
- Error handling for login failures

4.3.2 Sequence Diagram 2 – Data Retrieval and Analysis

Figure 6 demonstrates the complete flow from data retrieval to AI analysis and visualization.

Process Flow:

1. Cache check for existing data
2. API call to Forex data provider
3. Technical indicator calculation
4. AI model prediction
5. Data visualization preparation
6. Response delivery to user

4.3.3 Sequence Diagram 3 – Alert Management

Figure 7 illustrates alert creation, monitoring, and notification delivery.

Alert Workflow:

- User defines alert conditions
- Alert stored in database
- Continuous monitoring of market conditions
- Trigger detection and notification dispatch
- Alert status update

4.4 Activity Diagrams

Activity diagrams model the workflow logic and decision points within system processes.

4.4.1 Activity Diagram 1 – Authentication & Account Management

Figure 8 shows the workflows for user registration, login, and account management.

4.4.2 Activity Diagram 2 – Data Retrieval & Analysis

Figure 9 illustrates the complete data processing workflow.

4.4.3 Activity Diagram 3 – Customization Workflows

Figure 10 demonstrates user customization processes.

4.4.4 Activity Diagram 4 – Reports & Alerts

Figure 11 shows report generation and alert management workflows.

4.4.5 Activity Diagram 5 – Administration & Error Handling

Figure 12 illustrates administrative workflows and error handling processes.

5 Structural Diagrams

Structural diagrams define the static organization of the system, including classes, components, and their relationships.

5.1 Class Diagram

The Class Diagram (Figure 13) presents the complete object-oriented design of the system with all entities, their attributes, methods, and relationships.

Key Classes and Packages:**1. User Management:**

- User (abstract base class)
- GuestUser
- RegisteredUser
- Administrator
- UserPreferences

2. Authentication & Security:

- AuthenticationService
- SessionManager
- SecurityManager

3. Data Management:

- ForexDataService
- APIGateway
- CacheManager
- DataRepository

4. Analysis Engine:

- TechnicalAnalyzer
- IndicatorCalculator
- AIAnalysisEngine
- PredictionModel

5. Visualization & Reporting:

- Chart
- Analysis
- ReportGenerator

6. Alert System:

- AlertManager
- Alert

- AlertMonitor
- NotificationService

7. Administration:

- SystemMonitor
- SystemHealth
- APIAnalytics

Design Patterns Used:

- **Singleton:** SessionManager, CacheManager
- **Factory:** User creation (GuestUser, RegisteredUser)
- **Observer:** Alert monitoring and notification
- **Strategy:** Multiple technical indicators
- **Repository:** Data access abstraction

5.2 Component Diagram

The Component Diagrams provide architectural views of the system showing high-level components and their dependencies.

5.2.1 Component Diagram 1 – Presentation Layer

Figure 14 shows the frontend components and their organization.

5.2.2 Component Diagram 2 – Business Logic Layer

Figure 15 illustrates the core business logic components.

5.2.3 Component Diagram 3 – Complete System Architecture

Figure 16 presents the integrated view of all system components.

Component Organization:

- **Presentation Layer:** Web UI, Mobile Interface
- **Application Layer:** API Gateway, WebSocket Manager
- **Business Logic Layer:** Authentication, Analysis Engine, Alert System
- **Data Access Layer:** Data Repository, Cache Store
- **External Services:** Forex API, Economic Events API

6 Requirements–Design Traceability

This section provides comprehensive traceability matrices mapping each requirement to its corresponding design artifacts, ensuring complete requirement coverage.

6.1 Functional Requirements Traceability

Table 1 maps all 30 functional requirements to design artifacts.

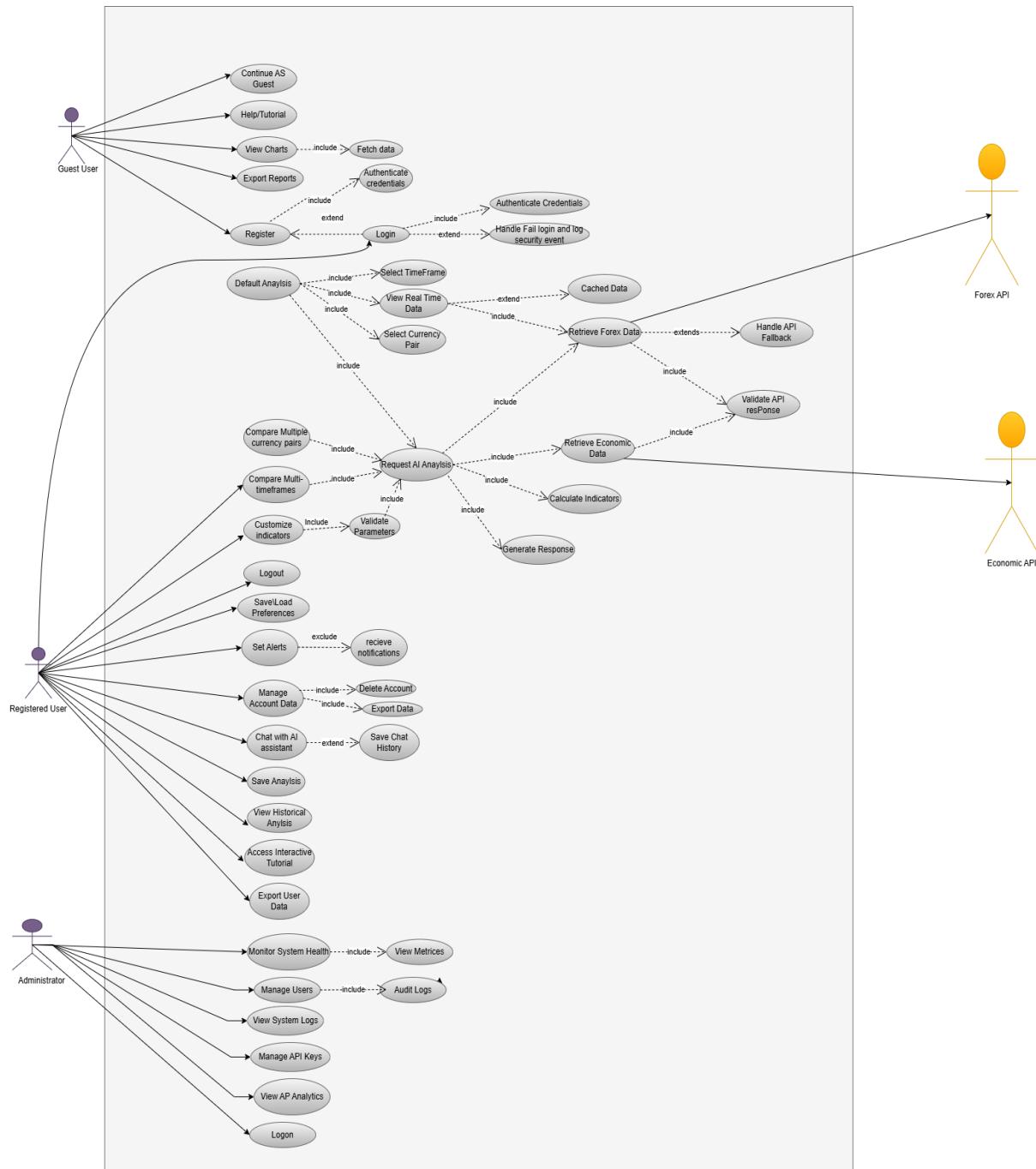


Figure 1: Use Case Diagram – System-Actor Interactions

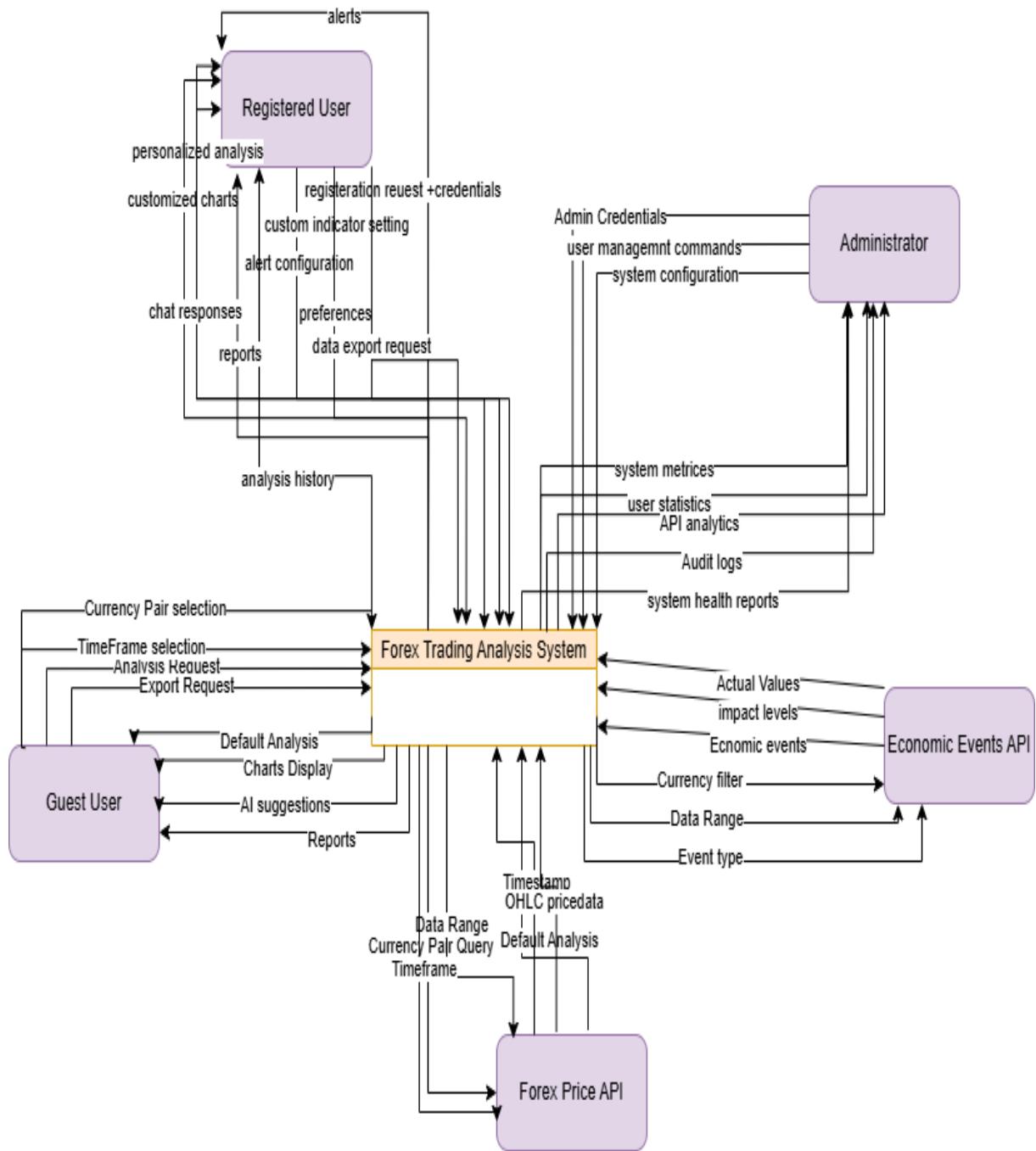


Figure 2: DFD Level 0 – Context Diagram

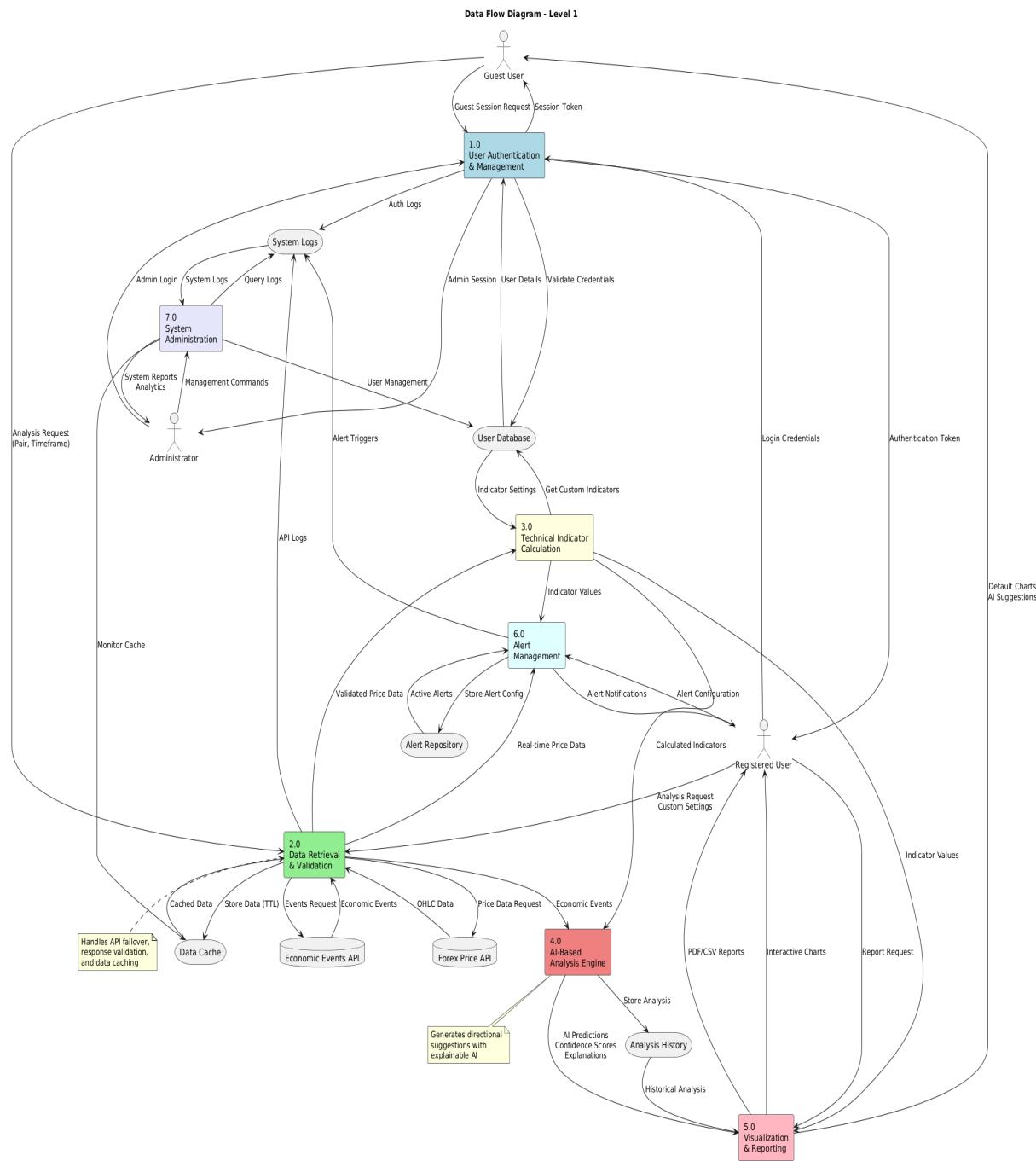


Figure 3: DFD Level 1 – Major System Processes

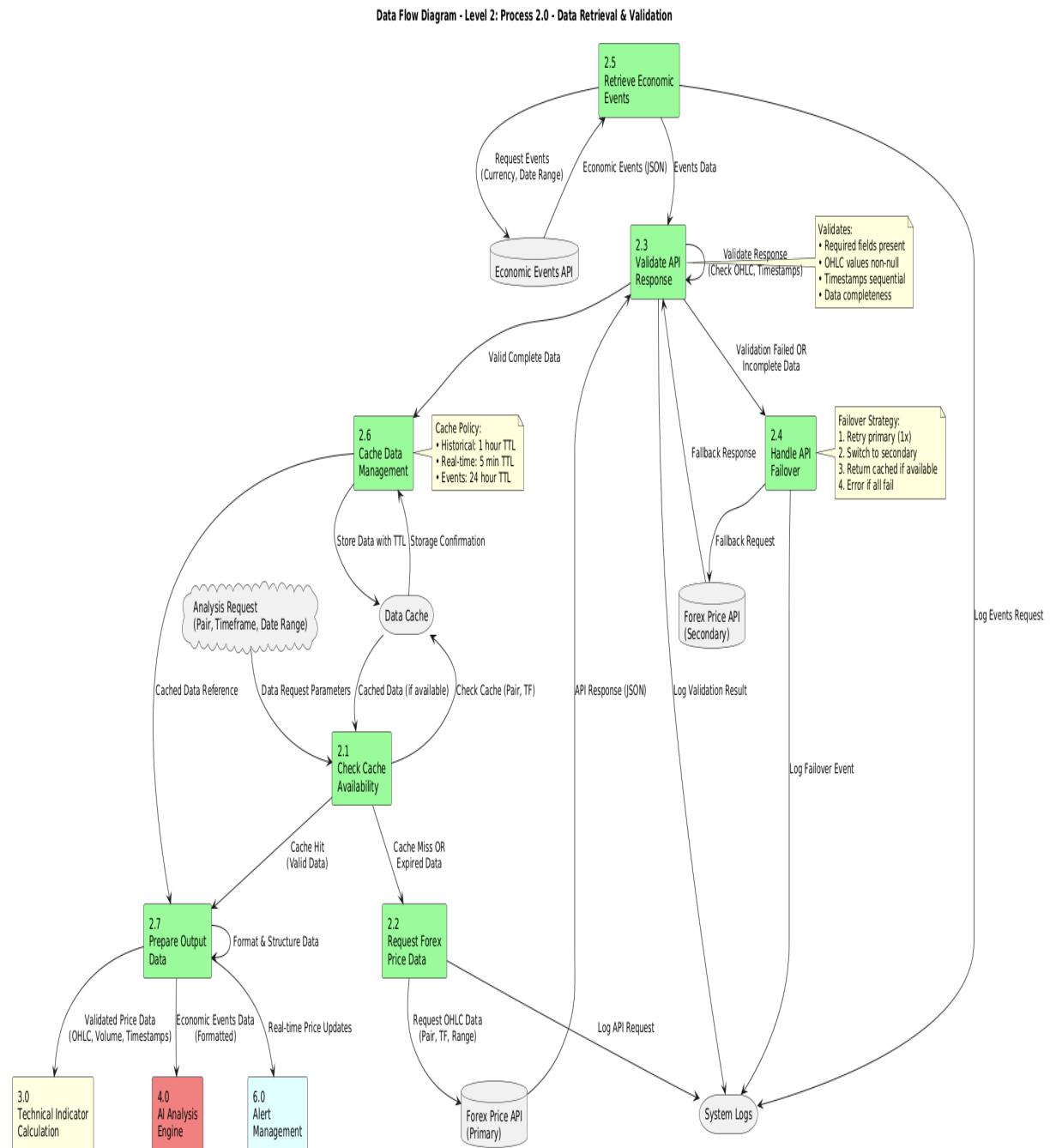


Figure 4: DFD Level 2 – Data Retrieval & API Management Detail

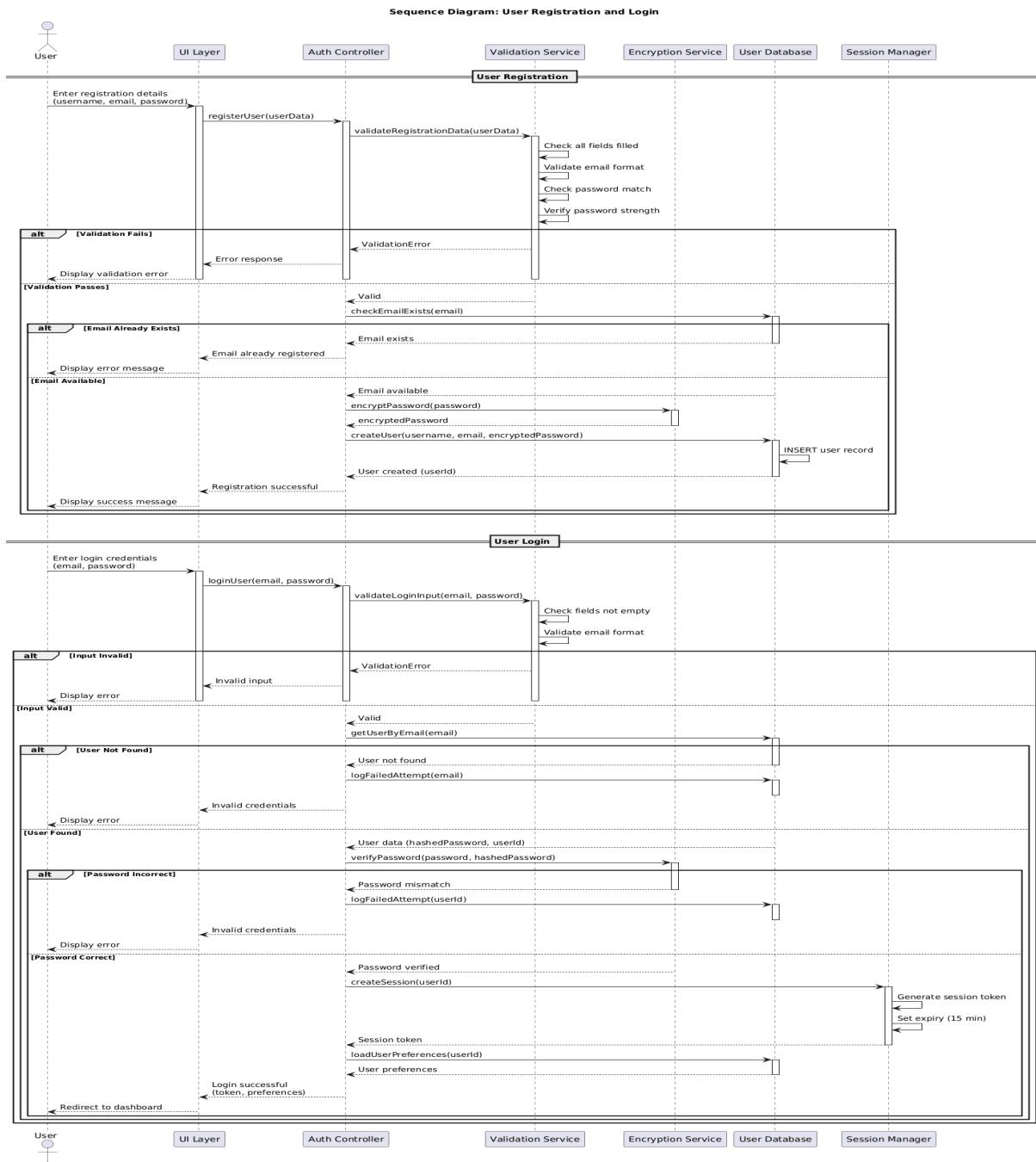


Figure 5: Sequence Diagram 1 – User Registration and Login

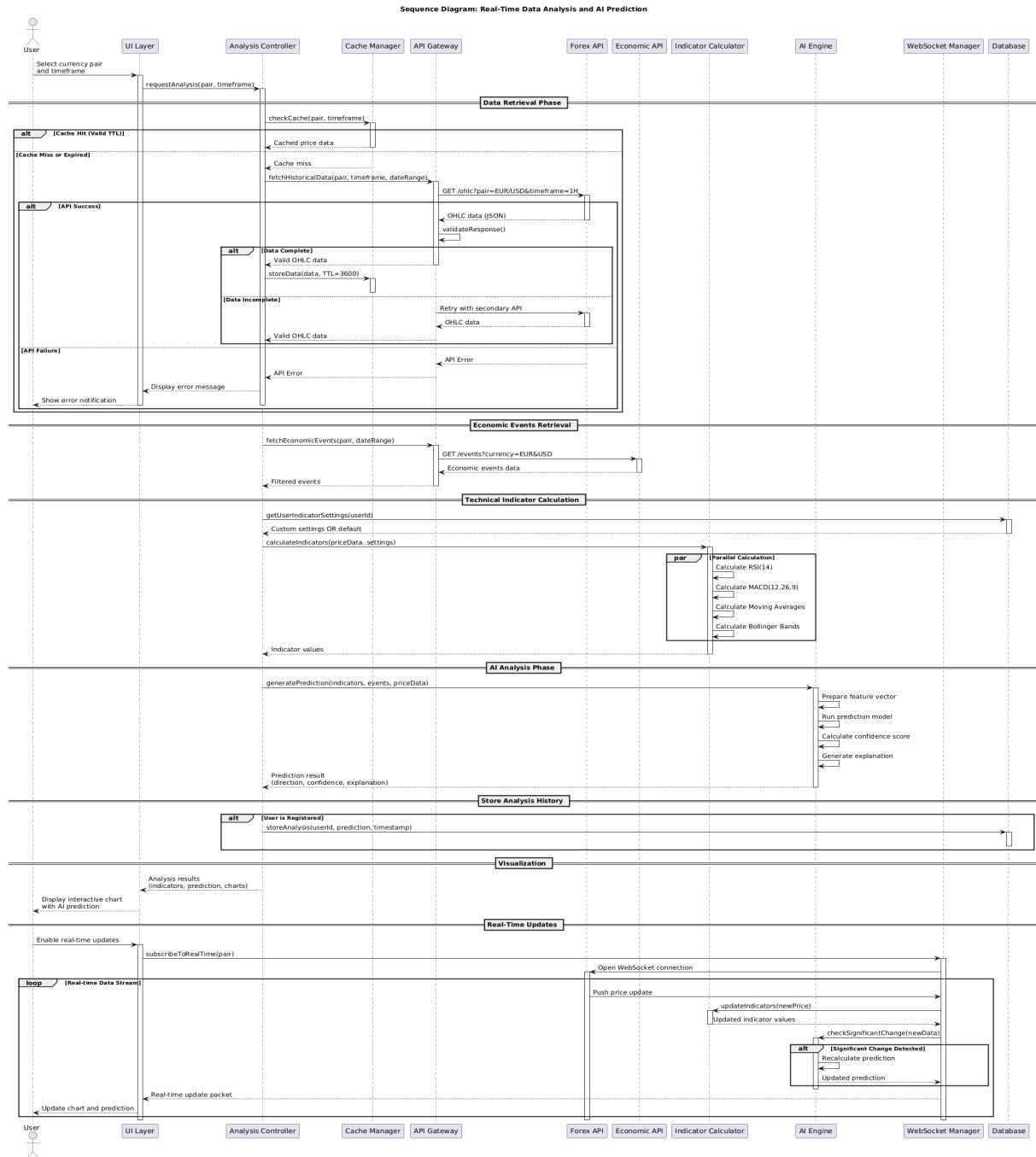


Figure 6: Sequence Diagram 2 – Data Retrieval, Analysis, and Visualization

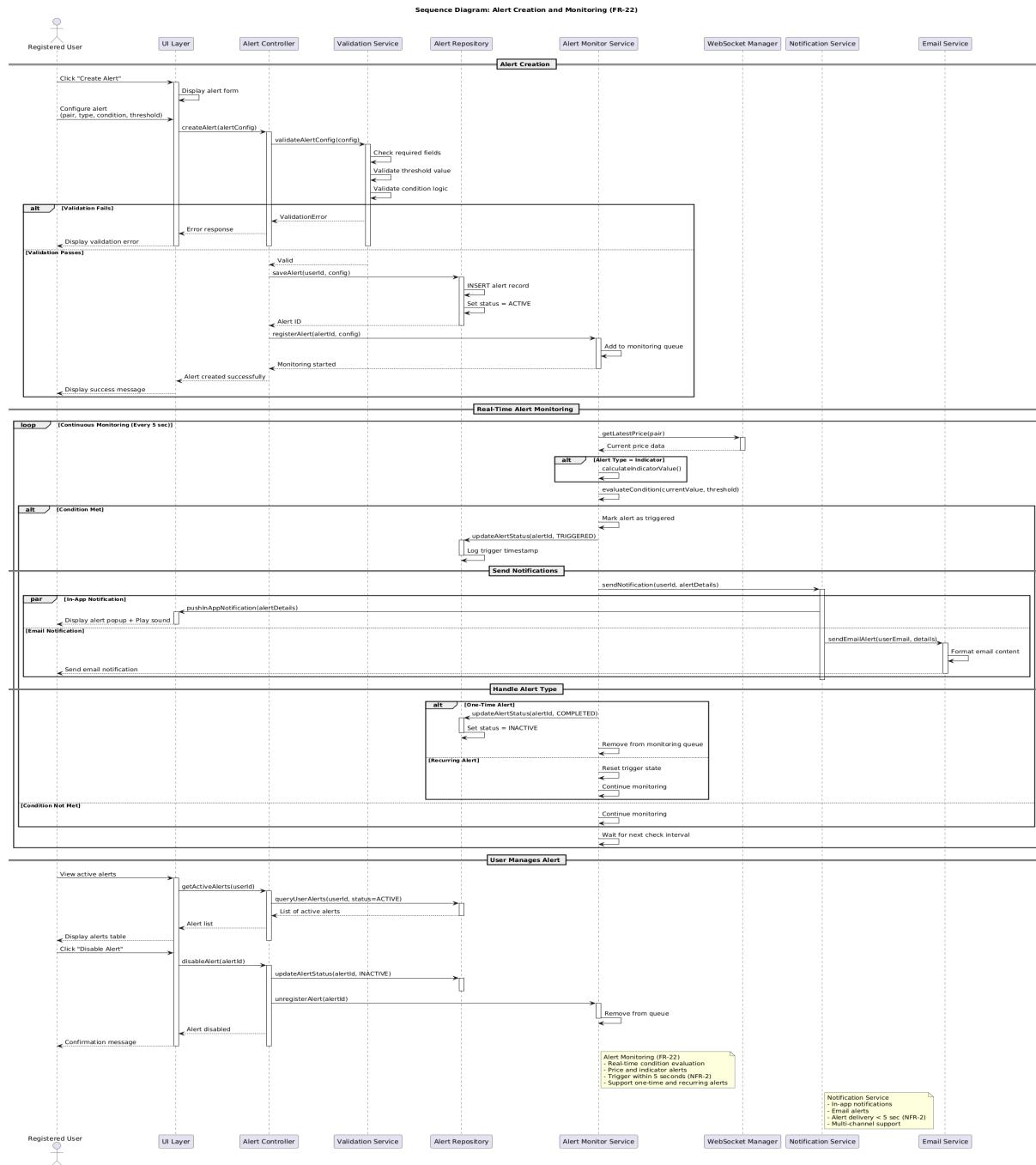


Figure 7: Sequence Diagram 3 – Alert Creation and Monitoring

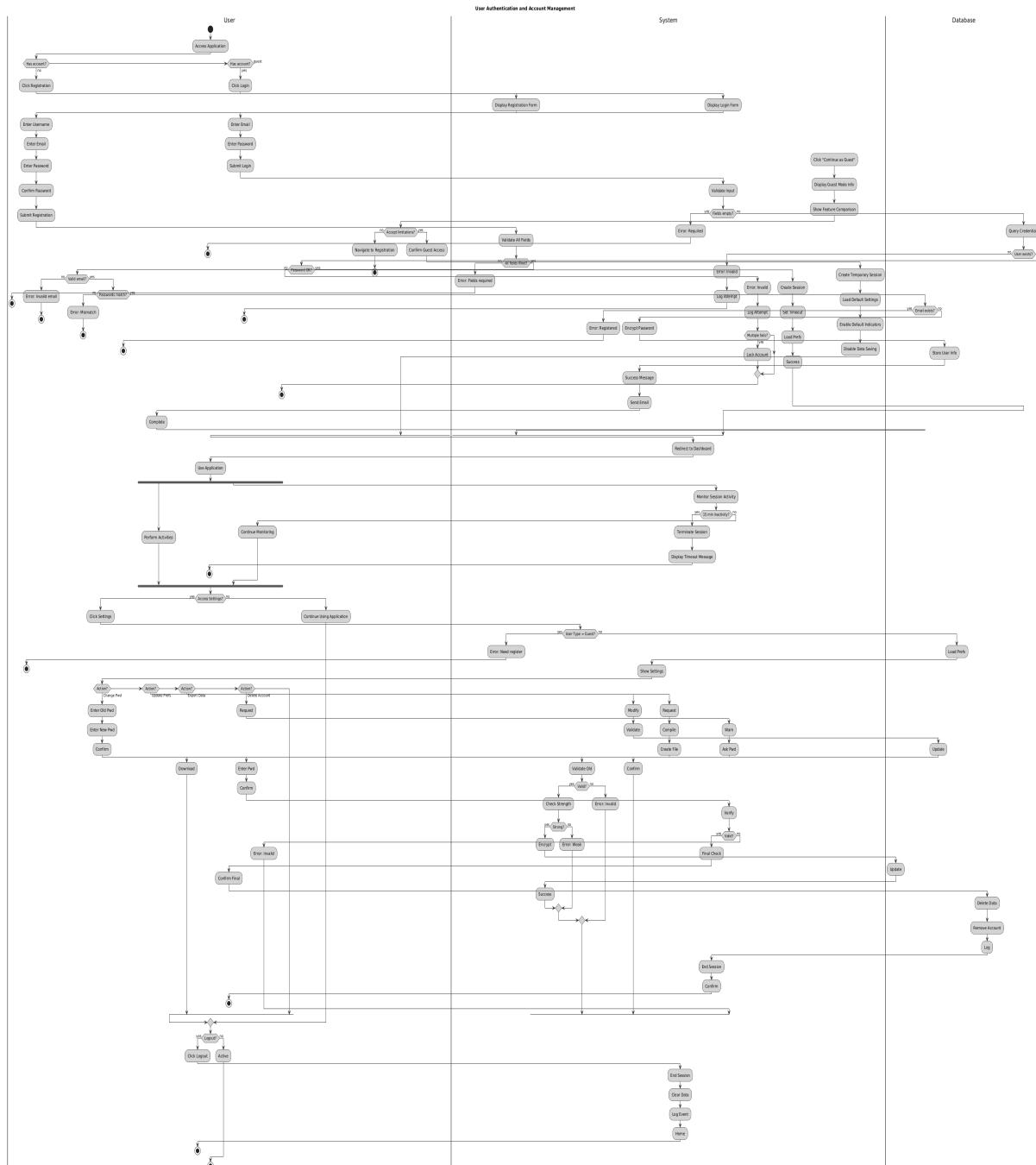


Figure 8: Activity Diagram 1 – Authentication & Account Management

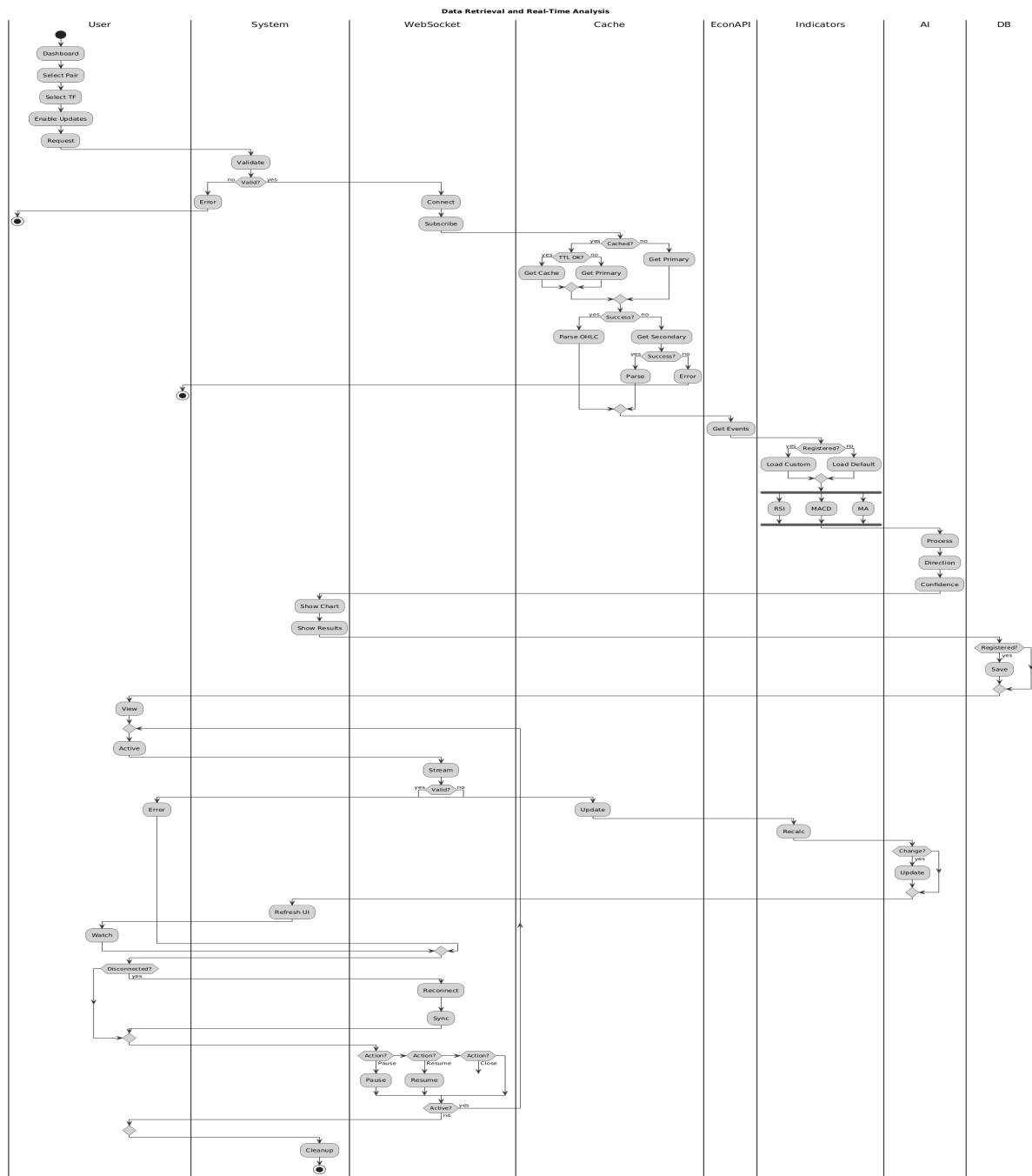


Figure 9: Activity Diagram 2 – Data Retrieval & Analysis Workflow

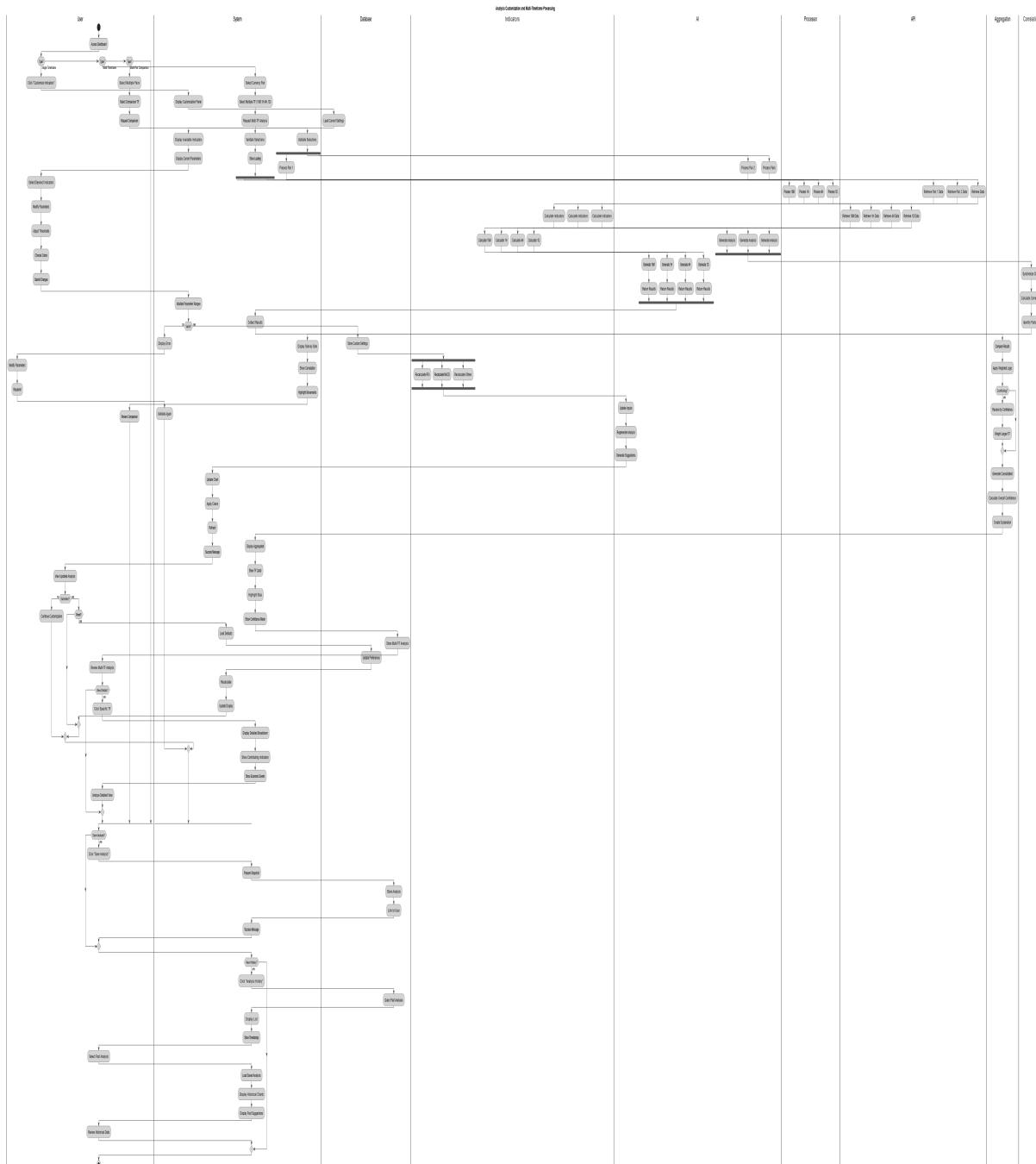


Figure 10: Activity Diagram 3 – Indicator Customization & Preferences

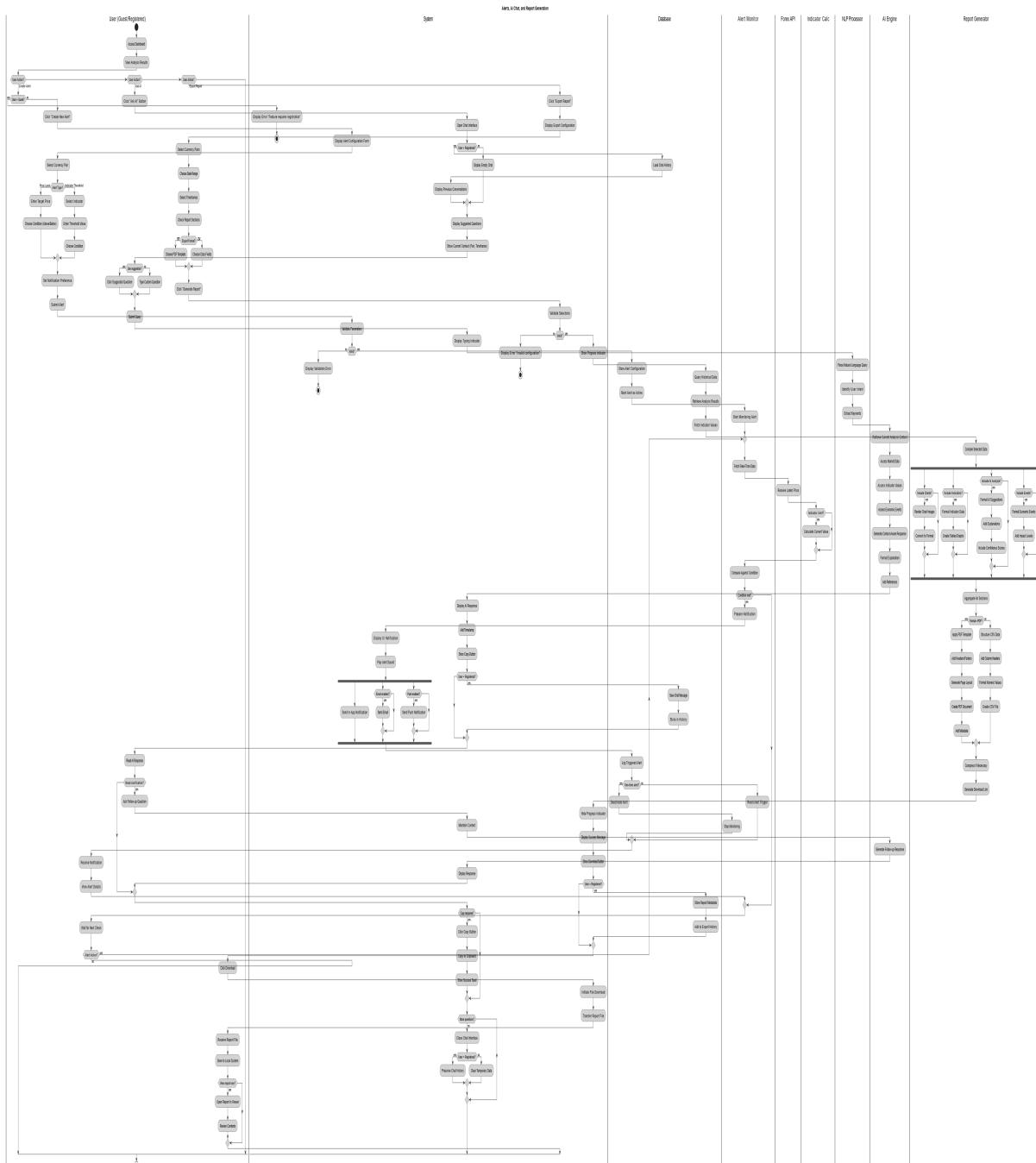


Figure 11: Activity Diagram 4 – Reports & Alert Management

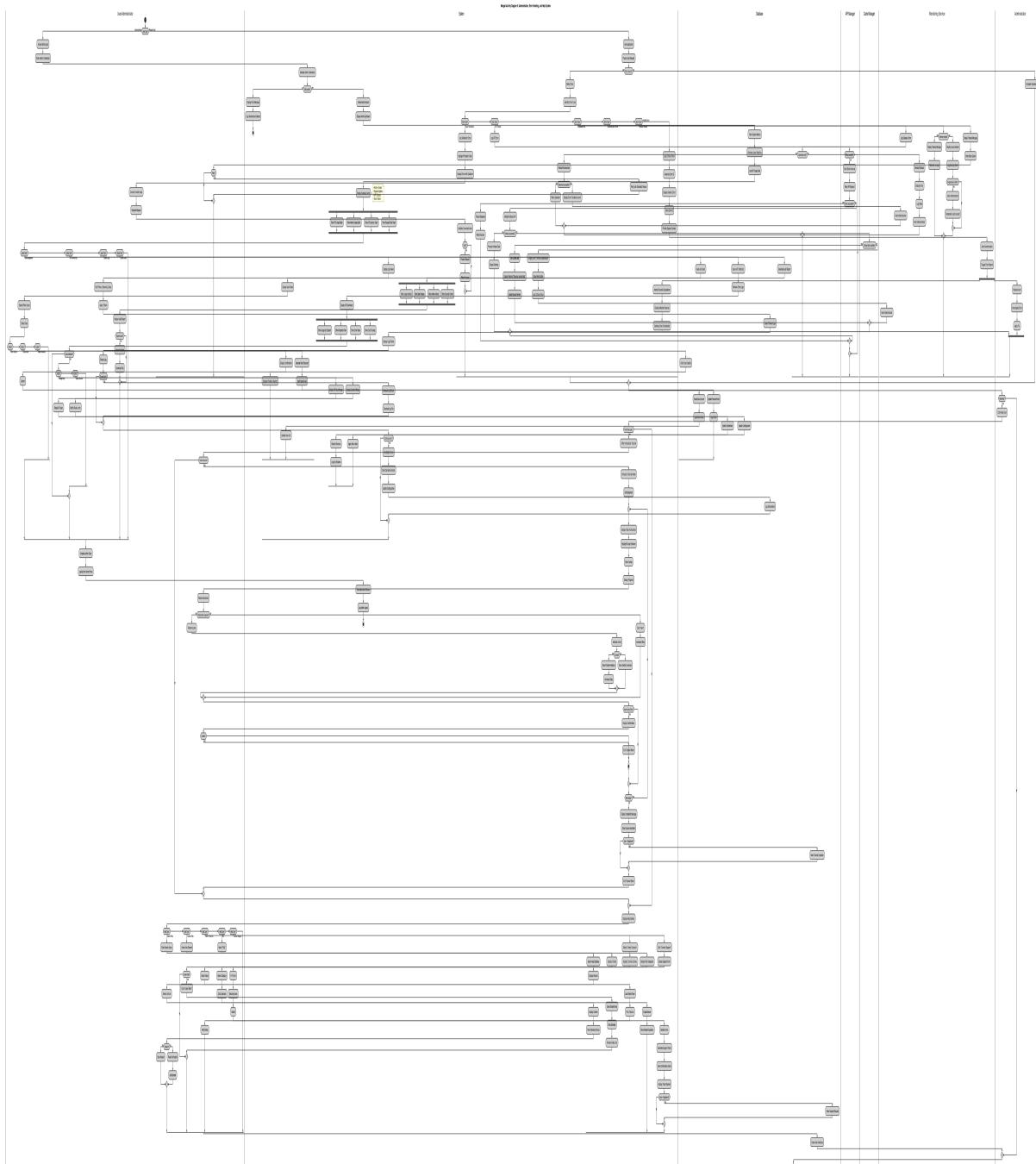


Figure 12: Activity Diagram 5 – Administration & Error Handling

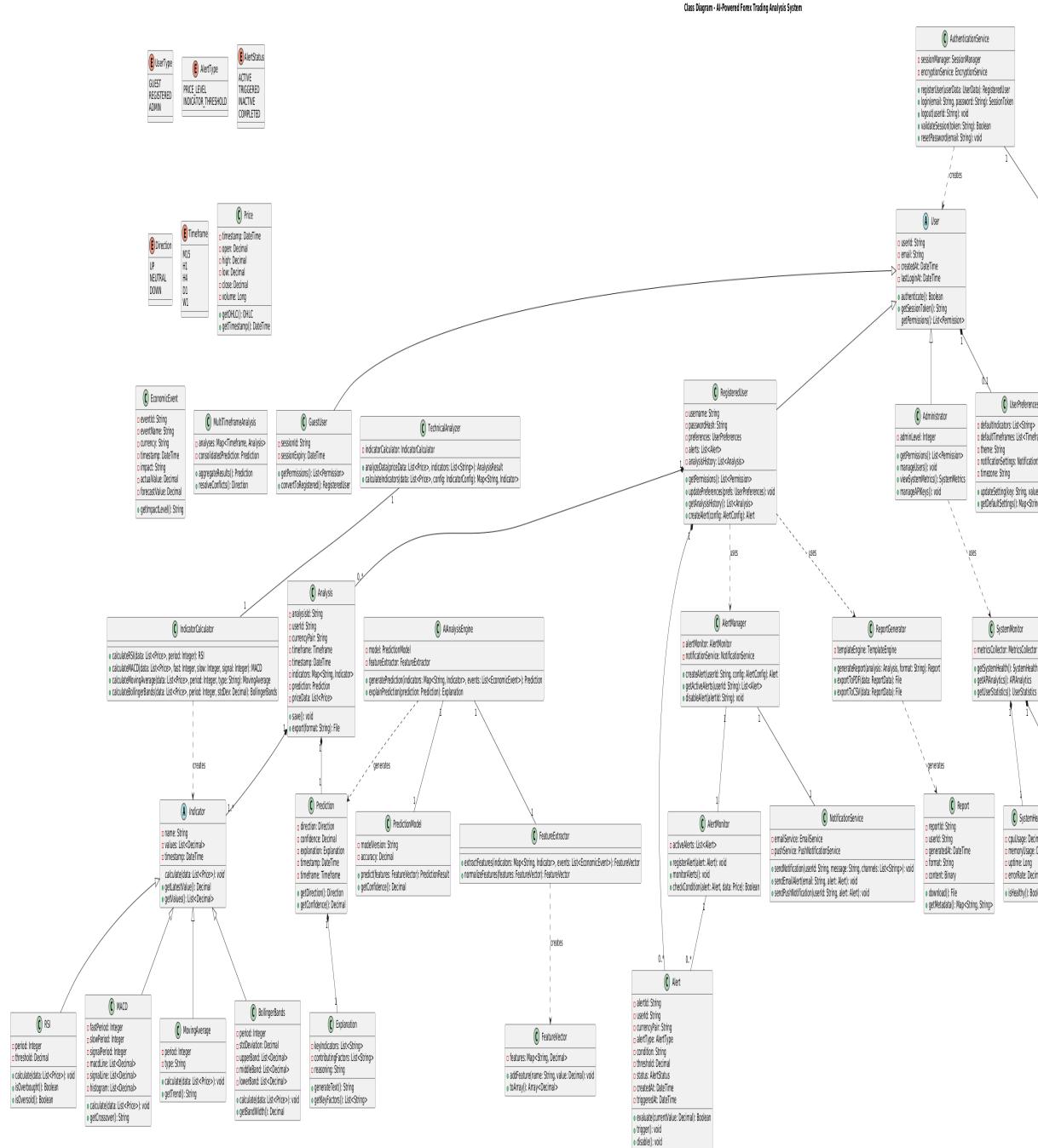


Figure 13: Class Diagram – System Class Structure

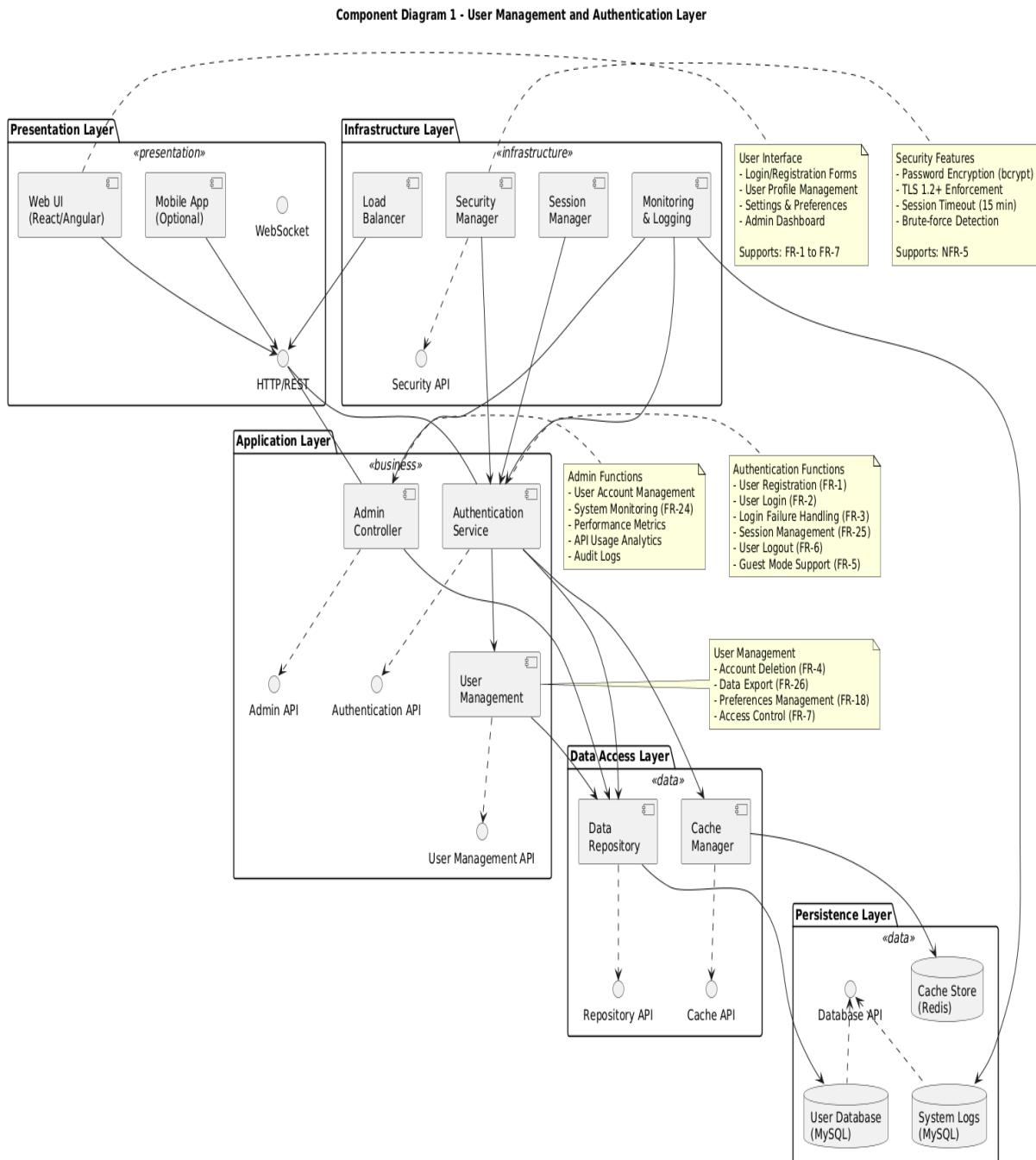


Figure 14: Component Diagram 1 – Presentation Layer Components

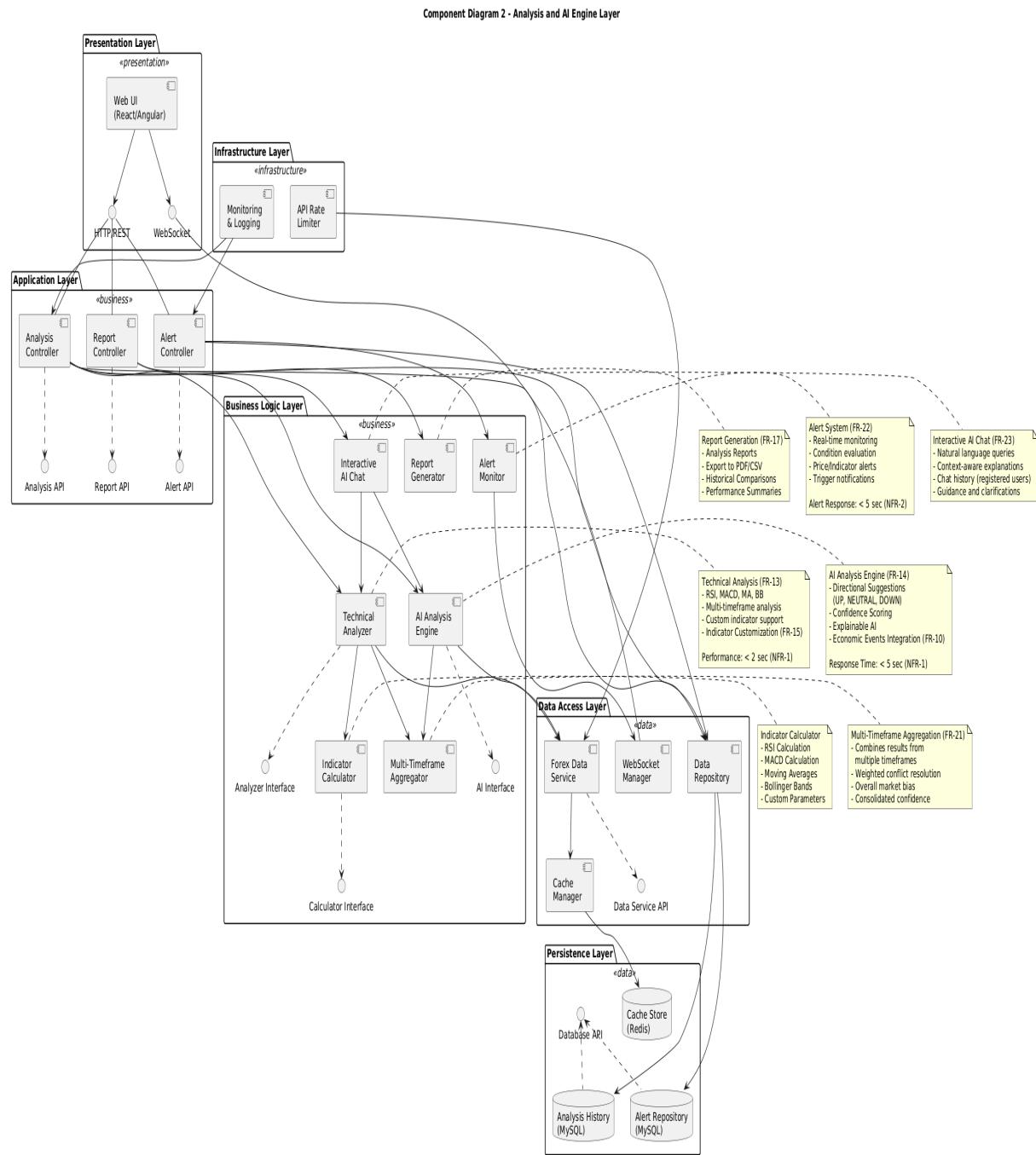


Figure 15: Component Diagram 2 – Business Logic Layer

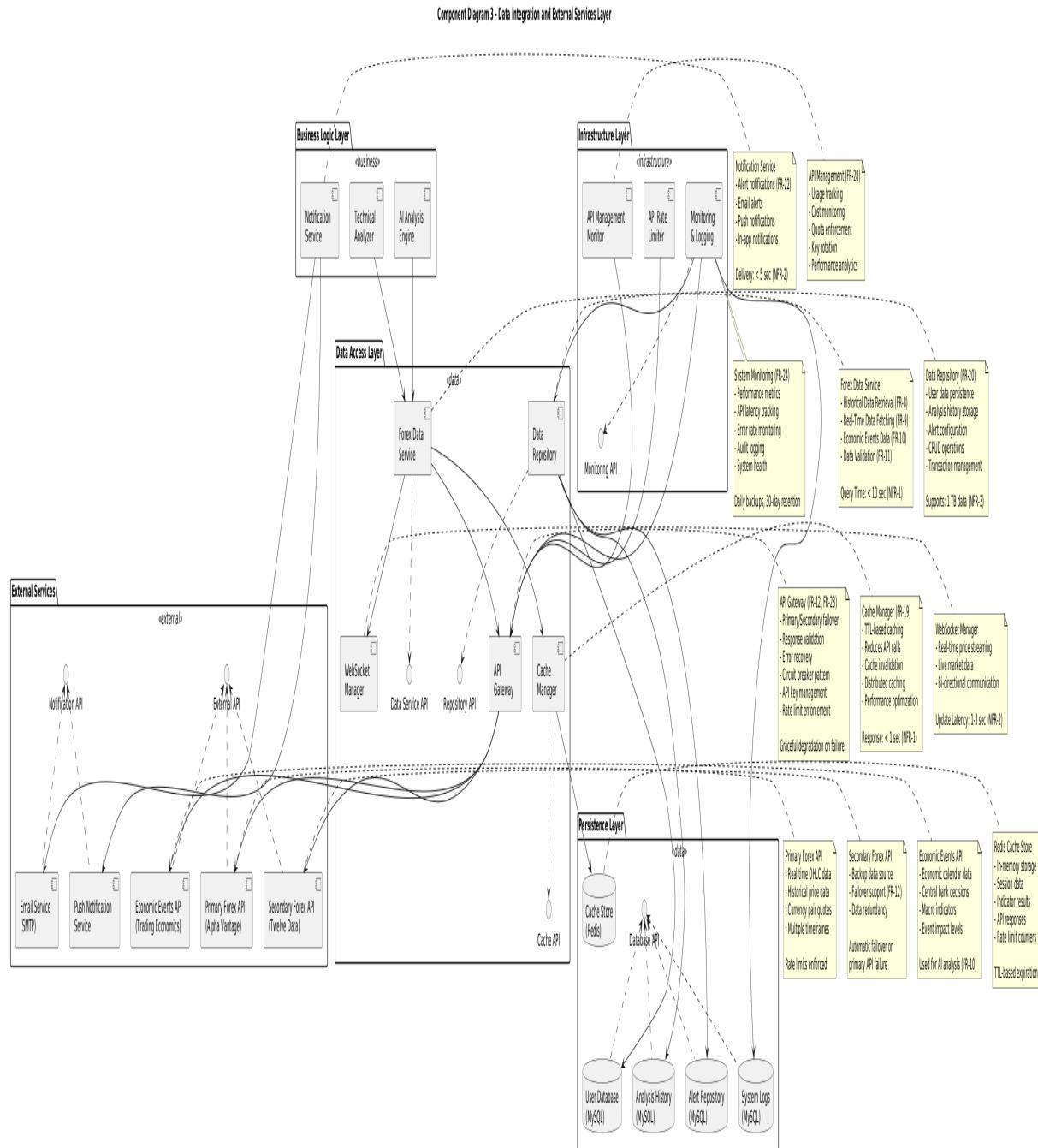


Figure 16: Component Diagram 3 – Complete System Architecture

Table 1: Functional Requirements–Design Traceability Matrix

Req ID	Description	Use Case	DFD	Sequence	Class(es)	Component(s)	Activity
FR-1	User Registration	UC6	P1.0	SD1	User, RegisteredUser, AuthService	Authentication Service	AD1
FR-2	User Login	UC7	P1.0	SD1	User, AuthService, SessionManager	Authentication, Session Mgr	AD1
FR-3	Login Failure Handling	UC7	P1.0	SD1	AuthService, SessionManager	Authentication, Monitoring	AD1
FR-4	Account Deletion & Data Mgmt	UC11, UC12	P1.0, P7.0	–	RegisteredUser, DataRepository	User Mgmt, Data Repository	AD1
FR-5	Guest Mode	UC1	P1.0	–	GuestUser	Authentication Service	AD1
FR-6	User Logout	UC8	P1.0	–	SessionManager	Session Manager	AD1
FR-7	Access Control	UC1, UC7	P1.0	–	User, GuestUser, RegisteredUser	Security Manager	AD1
FR-8	Forex Historical Data Retrieval	UC35	P2.0, P2.2	SD2	ForexDataService, APIGateway	Forex Data, API Gateway	AD2

Req ID	Description	Use Case	DFD	Sequence	Class(es)	Component(s)	Activity
FR-9	Real-Time Data Fetching	UC15	P2.0, P6.0	SD2	ForexDataService, WebSocket, WebSocket-Manager	WebSocket, Forex Data	AD2
FR-10	Economic Events Data	UC36	P2.0, P2.5	SD2	EconomicEvents APIGateway	API Gateway	AD2
FR-11	API Response Validation	UC39	P2.0, P2.3	SD2	APIGateway	API Gateway	AD5
FR-12	API Fallback Handling	UC40	P2.0, P2.4	SD2	APIGateway, ForexAPI	API Gateway	AD5
FR-13	Technical Indicator Calculation	UC37	P3.0	SD2	TechnicalAnalyzer, IndicatorCalculator	Technical Analyzer	AD2
FR-14	AI-Based Market Analysis	UC17, UC38	P4.0	SD2	AIAnalysisEngine, Prediction-Model	AI Analysis Engine	AD2
FR-15	Indicator Customization	UC16	P3.0	–	RegisteredUser, IndicatorCalculator	Indicator Calculator	AD3
FR-16	Charting & Visualization	UC3	P5.0	–	Analysis, Chart	Web UI	AD2
FR-17	Report Generation & Export	UC5	P5.0	–	ReportGenerator, ReportReport	Report Generator	AD4

Req ID	Description	Use Case	DFD	Sequence	Class(es)	Component(s)	Activity
FR-18	User Preferences Management	UC10	P1.0	–	RegisteredUser, UserPreferences	User Management	AD1
FR-19	Data Caching	UC41	P2.0, P2.6	SD2	CacheManager, CachedData	Cache Manager	AD2
FR-20	Historical Analysis Storage	UC24, UC25	P5.0	–	Analysis, RegisteredUser	Data Repository	AD3
FR-21	Multi-Timeframe Analysis	UC18	P4.0	–	MultiTimeframe	AnalysisEngine, Tech Analyzer	AD3
FR-22	Alert System	UC20-23	P6.0	SD3	AlertManager, Alert, Alert-Monitor	Alert Controller	AD4
FR-23	Interactive AI Chat	UC4	P4.0	–	AIAnalysisEngine, RegisteredUser	AI Engine, Web UI	AD4
FR-24	Admin Dashboard	UC26-31	P7.0	–	Administrator, SystemMonitor	Admin Controller	AD5
FR-25	Security & Session Mgmt	–	P1.0	SD1	SessionManager, SecurityManager	Session Mgr, Security	AD1
FR-26	Data Privacy Management	UC11, UC12	P1.0, P7.0	–	RegisteredUser, DataRepository	User Mgmt, Data Repo	AD1

Req ID	Description	Use Case	DFD	Sequence	Class(es)	Component(s)	Activity
FR-27	Multi-Currency Pair Comparison	UC19	P4.0	–	MultiTimeframe	TradeAnalyzer, AI Engine	AD3
FR-28	API Management & Monitoring	UC28, UC29	P7.0	–	APIAnalytics, SystemMonitor	Monitoring, API Gateway	AD5
FR-29	Tutorial & Help System	UC32-34	–	–	–	Web UI	AD5
FR-30	Error Handling & Notifications	–	All	All	NotificationServ	Notification, Monitoring	AD5

6.2 Non-Functional Requirements Traceability

Table 2 maps non-functional requirements to design components and architectural patterns.

Table 2: Non-Functional Requirements Traceability Matrix

NFR ID	Description	Design Components	Architecture Pattern
NFR-1	Performance (2-10 sec response)	Cache Manager, Load Balancer, Database Indexing	Caching Strategy, Async Processing
NFR-2	Real-Time Updates (1-5 sec)	WebSocket Manager, Alert Monitor	Event-Driven Architecture
NFR-3	Scalability (10,000 users)	Load Balancer, Horizontal Scaling	Microservices-Ready Architecture
NFR-4	Availability (99% up-time)	API Fallback, Cache Fallback, Monitoring	Redundancy & Failover
NFR-5	Security (TLS, Encryption)	Security Manager, Session Manager, Encryption Service	Layered Security
NFR-6	Usability	Web UI, Tutorial System, Help System	User-Centered Design
NFR-7	Reliability	Error Handling, Logging, Monitoring	Resilience Patterns
NFR-8	Maintainability	Modular Components, Clean Architecture	Separation of Concerns
NFR-9	Compliance (GDPR, CSV/PDF)	Data Privacy Management, Report Generator	Standards Adherence

6.3 Diagram Coverage Matrix

Table 3 ensures all design artifacts provide complete requirement coverage.

Table 3: Design Artifact Coverage Matrix

Design Artifact	Requirements Covered	Status
Use Case Diagram	All FR (FR-1 to FR-30)	✓ Complete
DFD Level 0	System Context (All FR)	✓ Complete
DFD Level 1	FR-1 to FR-30 decomposed into 7 processes	✓ Complete
DFD Level 2	FR-8, FR-9, FR-10, FR-11, FR-12 (Data Retrieval)	✓ Complete
Sequence Diagram 1	FR-1, FR-2, FR-3, FR-6, FR-25	✓ Complete
Sequence Diagram 2	FR-8, FR-9, FR-10, FR-13, FR-14, FR-16, FR-19	✓ Complete
Sequence Diagram 3	FR-22, FR-23	✓ Complete
Activity Diagram 1	FR-1 to FR-7, FR-18, FR-26	✓ Complete
Activity Diagram 2	FR-8 to FR-14, FR-16, FR-19	✓ Complete
Activity Diagram 3	FR-15, FR-18, FR-20, FR-21, FR-27	✓ Complete
Activity Diagram 4	FR-17, FR-22, FR-23	✓ Complete
Activity Diagram 5	FR-24, FR-28, FR-29, FR-30	✓ Complete
Class Diagram	All FR (comprehensive system structure)	✓ Complete
Component Diagrams	All FR (system architecture)	✓ Complete

7 Prototype Screen Coverage

This section maps prototype screens to functional requirements and user stories, ensuring UI design alignment with system requirements.

7.1 Prototype–Requirements Mapping

Table 4 shows how each prototype screen implements specific requirements.

Table 4: Prototype Screen–Requirements Mapping

Prototype Screen	Requirements	User Story
Landing Page	FR-5, FR-6	As a visitor, I can see the system overview and choose to register or continue as guest
Registration Page	FR-1	As a new user, I can create an account with validation
Login Page	FR-2, FR-3	As a registered user, I can log in securely
Guest Dashboard	FR-5, FR-13, FR-14, FR-16	As a guest, I can view default analysis
Main Dashboard	FR-9, FR-13, FR-14, FR-15, FR-16, FR-18	As a registered user, I can view personalized real-time analysis
Indicator Customization	Cus- FR-15, FR-18	As a registered user, I can customize technical indicators
Multi-Timeframe View	FR-21	As a registered user, I can analyze multiple timeframes simultaneously
Alerts Page	FR-22	As a registered user, I can create and manage price/indicator alerts
AI Chat Panel	FR-23	As a user, I can ask AI questions about the analysis
Reports/Export	FR-17	As a user, I can export analysis reports in PDF/CSV
Settings Page	FR-4, FR-18, FR-26	As a registered user, I can manage preferences and account
History Page	FR-20	As a registered user, I can view past analyses
Admin Dashboard	FR-24, FR-28	As an admin, I can monitor system health and manage users
Help/Tutorial	FR-29	As a user, I can access tutorials and contextual help

7.2 Key Prototype Features

- 1. Responsive Design:** All screens adapt to desktop, tablet, and mobile devices
- 2. Consistent UI Elements:** Unified color scheme, typography, and component library
- 3. Accessibility:** WCAG 2.1 Level AA compliance with proper contrast and keyboard navigation
- 4. Interactive Components:** Real-time charts, drag-and-drop indicators, context-sensitive help

5. **Progressive Disclosure:** Complex features revealed gradually to avoid overwhelming users

8 Project Resources

8.1 GitHub Repository

Project documentation, diagrams, and meeting records are maintained at:

Repository URL: https://github.com/ShahbazAli-8005/FOREX_AI_AGENT.git

Repository Structure:

- `/docs` – SRS, Design Documents
- `/diagrams` – All UML and design diagrams
- `/prototypes` – Figma exports and interactive mockups
- `/meetings` – Meeting minutes and decisions
- `/traceability` – Requirement-design mapping documents

8.2 Figma Prototype

Interactive UI/UX prototypes are available at:

Figma URL: <https://www.figma.com/proto/ZVxV2hXEGmzqHOPOFVZxaz/Forex-Trading-Analyt=6eMNrvZSLCAxKjbk-0&scaling=min-zoom&content-scaling=fixed&page-id=0%3A1&node-id=1-206>

Prototype Features:

- Clickable interactive flows
- Responsive design previews
- Component library
- Design system documentation

8.3 Collaboration Tools

- **Communication:** Microsoft Teams
- **Project Management:** Trello Board
- **Documentation:** Google Docs (collaborative editing)

- **Version Control:** GitHub
- **Diagramming:** Draw.io, Lucidchart

9 Meeting Summary

9.1 Meeting 1 – Initial Design Review

Date: January 10, 2026

Participants: Saad Nawaz, Shahbaz Ali, Hafsa Gul, Course Instructor

Duration: 2 hours

Agenda:

- Review behavioral diagrams (use case, DFD, sequence, activity)
- Discuss structural design (class and component diagrams)
- Identify missing requirements or design gaps

Key Decisions:

- Added Level 2 DFD for data retrieval process detail
- Refined class diagram to include all relationships and multiplicities
- Agreed on microservices-ready architecture approach
- Decided to implement WebSocket for real-time updates

Action Items:

- Complete activity diagrams for all workflows – Assigned to Shahbaz Ali
- Finalize component diagrams with deployment view – Assigned to Saad Nawaz
- Create initial prototype screens – Assigned to Hafsa Gul

9.2 Meeting 2 – Prototype Demonstration

Date: January 14, 2026

Participants: Project Team, Requirement Provider (Industry Expert)

Duration: 1.5 hours

Agenda:

- Demonstrate interactive prototypes
- Validate UI design against requirements

- Gather feedback on user experience

Feedback Received:

- Dashboard layout approved with minor adjustments to chart placement
- Request for more prominent alert notifications
- Suggestion to add quick-access toolbar for frequent actions
- Positive feedback on AI chat interface design

Revisions Made:

- Repositioned notification bell with badge counter
- Added customizable quick-access toolbar to dashboard
- Enhanced visual hierarchy for critical information

9.3 Meeting 3 – Design Validation & Traceability Review

Date: January 17, 2026

Participants: Full Project Team, Course Instructor

Duration: 2 hours

Agenda:

- Verify requirements-design traceability
- Review completeness of all design artifacts
- Final approval for design document submission

Validation Results:

- All 30 functional requirements traced to design artifacts – ✓ Verified
- All 9 non-functional requirements addressed in architecture – ✓ Verified
- Complete diagram coverage confirmed – ✓ Verified
- Prototype screens aligned with user stories – ✓ Verified

Final Approval: Design document approved for submission with no major revisions required.

10 Design Notes and Considerations

10.1 Scalability Considerations

- Stateless component design enables horizontal scaling
- Database connection pooling for efficient resource utilization
- Load balancer distributes traffic across multiple server instances
- CDN integration for static assets reduces server load
- Asynchronous processing for computationally intensive AI analysis

10.2 Security Measures

- All API endpoints protected with JWT authentication
- Rate limiting implemented to prevent abuse (100 requests/minute per user)
- SQL injection prevention through parameterized queries
- XSS protection via input sanitization and CSP headers
- CSRF tokens for state-changing operations
- Regular security audits and dependency vulnerability scanning

10.3 Performance Optimization

- Redis caching for frequently accessed forex data (5-minute TTL)
- Database query optimization with proper indexing on foreign keys
- Lazy loading of chart data to reduce initial page load time
- Image optimization and compression for UI assets
- Minification of JavaScript and CSS resources
- Gzip compression for API responses

10.4 Maintenance and Monitoring

- Comprehensive logging with ELK stack (Elasticsearch, Logstash, Kibana)
- Application Performance Monitoring (APM) with New Relic or Datadog
- Health check endpoints for each microservice
- Automated backup strategy for database (daily full, hourly incremental)
- CI/CD pipeline for automated testing and deployment
- Error tracking and alerting with Sentry

10.5 Future Enhancements

- Mobile native applications (iOS and Android)
- Integration with trading platforms for automated trading
- Advanced machine learning models for market prediction
- Social trading features (follow expert traders, copy strategies)
- Multi-language support for international users
- Voice commands for hands-free analysis

11 Conclusion

11.1 Design Completeness

This design document provides a comprehensive, validated blueprint for implementing the AI-Powered Forex Trading Analysis System. All design artifacts have been:

- **Thoroughly Documented:** Complete behavioral and structural models
- **Fully Traceable:** Every requirement mapped to design artifacts
- **Validated:** Reviewed and approved by stakeholders and requirement providers
- **Implementation-Ready:** Sufficient detail for development teams to begin coding

11.2 Verification Summary

Verification Criteria	Status
All 30 functional requirements traced	✓ Complete
All 9 non-functional requirements addressed	✓ Complete
Behavioral diagrams cover all workflows	✓ Complete
Structural diagrams define complete architecture	✓ Complete
Prototype screens implement requirements	✓ Complete
Design reviewed and approved	✓ Complete

11.3 Next Steps

Following the approval of this design document, the project will proceed to:

- Sprint Planning:** Break down implementation into 2-week sprints
- Environment Setup:** Configure development, testing, and production environments
- Database Implementation:** Create schema based on class diagram
- Backend Development:** Implement business logic and API endpoints
- Frontend Development:** Build UI components based on prototypes
- Integration Testing:** Verify component interactions
- User Acceptance Testing:** Validate system meets requirements
- Deployment:** Launch beta version for pilot users

11.4 Project Team Acknowledgment

This design document represents the collaborative effort of:

- **Saad Nawaz:** Lead System Architect, Structural Design
- **Shahbaz Ali:** Behavioral Modeling, Activity Diagrams
- **Hafsa Gul:** UI/UX Design, Prototype Development

Special thanks to our Course Instructor and Industry Requirement Provider for their valuable guidance and feedback throughout the design process.

Document Status: Final – Approved for Implementation

Last Updated: January 18, 2026

Version: 1.0