Problem Statement 14: Financial Advisory AI

Problem Overview

Summary: Build an AI-powered financial advisory system that provides personalized investment recommendations, portfolio optimization, and financial planning guidance based on individual user profiles and market conditions.

Problem Statement: Traditional financial advisory services are often expensive, limited in availability, and may not provide personalized guidance for individual investors. Your task is to create an AI system that democratizes financial advice by providing intelligent, personalized recommendations for investment strategies, portfolio management, risk assessment, and financial planning. The system should analyze user financial profiles, market conditions, and regulatory requirements to deliver actionable insights.

Key Requirements

Core Functionality

- **Personalized Investment Recommendations:** AI-driven analysis of user profiles, risk tolerance, and financial goals
- **Portfolio Optimization:** Dynamic rebalancing and asset allocation strategies
- **Risk Assessment:** Comprehensive risk profiling and scenario analysis
- **Market Analysis:** Real-time market data integration and trend analysis
- **Regulatory Compliance:** Adherence to financial regulations and disclosure requirements
- **Educational Content:** Financial literacy resources and explanation of recommendations

Implementation Steps

1. Design user profiling system capturing financial goals, risk tolerance, and investment preferences
2. Implement market data integration for real-time analysis and trend identification
3. Create AI-powered recommendation engine using modern portfolio theory and machine learning
4. Build portfolio optimization algorithms with risk-adjusted returns and diversification
5. Develop regulatory compliance framework ensuring fiduciary standards
6. Include educational components and transparent explanation of advice rationale

Data Requirements

Essential Datasets

- **Market Data:** Historical and real-time stock prices, bond yields, commodity prices, economic indicators
- **Financial Instruments:** Comprehensive database of stocks, bonds, ETFs, mutual funds, alternatives
- **Economic Data:** GDP, inflation, interest rates, employment statistics, sector performance
- **User Profiles:** Demographics, income, expenses, assets, liabilities, investment experience
- **Regulatory Data:** Compliance requirements, disclosure standards, fiduciary guidelines

Data Sources

- **Market Data Providers:** Yahoo Finance, Alpha Vantage, Quandl, Bloomberg API
- **Economic Indicators:** Federal Reserve Economic Data (FRED), Bureau of Labor Statistics
- **Financial Instruments:** SEC EDGAR database, fund prospectuses, rating agencies
- **Regulatory Guidelines:** SEC, FINRA, CFTC compliance frameworks

Technical Themes

- **GenAI & Techniques:** Natural language processing for financial advice generation
- **Portfolio Optimization:** Modern Portfolio Theory, Black-Litterman model, risk parity
- **Machine Learning:** Predictive modeling, clustering for user segmentation, reinforcement learning

Expected Business Outcomes

Performance Targets

- **Recommendation Accuracy:** >80% user satisfaction with investment suggestions
- **Portfolio Performance:** Risk-adjusted returns competitive with human advisors
- **User Engagement:** >70% monthly active usage among registered users
- **Compliance Rate:** 100% adherence to regulatory requirements and disclosure standards

User Benefits

- **Democratized Access:** Professional-grade financial advice accessible to all income levels
- **Personalization:** Tailored recommendations based on individual circumstances and goals
- **Cost Efficiency:** Significantly lower fees compared to traditional advisory services
- **Educational Value:** Enhanced financial literacy through transparent explanations

Implementation Strategy

Phase 1: Core Advisory Engine (Months 1-4)

- User profiling and risk assessment framework
- Basic portfolio optimization algorithms
- Market data integration and analysis
- Simple recommendation engine with rule-based logic

Phase 2: AI Enhancement (Months 5-8)

- Machine learning models for market prediction
- Advanced portfolio optimization with AI-driven insights
- Natural language generation for advice explanations
- Personalized educational content delivery

Phase 3: Advanced Features (Months 9-12)

- Multi-goal financial planning (retirement, education, major purchases)
- Tax optimization strategies and loss harvesting
- Alternative investment recommendations
- Integration with financial institutions and brokerages

This AI-powered financial advisory system will democratize access to professional investment guidance while maintaining the highest standards of fiduciary responsibility and regulatory compliance. # Product Requirements Document (PRD) ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - Problem statement analysis completed and documented in README.md - Market research and competitive analysis conducted - Stakeholder requirements gathered and validated - Regulatory compliance requirements identified

TASK: Define comprehensive product requirements including business objectives, user personas, feature specifications, success metrics, and risk assessment for the Financial Advisory AI system.

VERIFICATION & VALIDATION: - Requirements traceability matrix established - Stakeholder sign-off on business objectives and success metrics - Regulatory compliance validation with financial industry standards - Technical feasibility assessment completed

EXIT CRITERIA: - Complete PRD document approved by stakeholders - Clear product vision and roadmap established - Success metrics and KPIs defined and measurable - Foundation established for Functional Requirements Document (FRD)

---

1. Product Vision and Strategic Objectives

1.1 Product Vision

To democratize professional-grade financial advisory services through AI-powered personalization, making sophisticated investment guidance accessible, affordable, and transparent for investors across all wealth levels while maintaining fiduciary standards and regulatory compliance.

1.2 Business Objectives

- **Market Democratization:** Provide professional financial advice to underserved market segments (assets under $100K)
- **Cost Reduction:** Deliver advisory services at 80% lower cost than traditional human advisors
- **Scalability:** Serve 100,000+ users simultaneously with personalized recommendations
- **Compliance Excellence:** Maintain 100% regulatory compliance with SEC, FINRA, and state regulations
- **Performance Leadership:** Achieve risk-adjusted returns competitive with top-quartile human advisors

1.3 Strategic Goals

- **Year 1:** Launch MVP with 10,000 active users and $50M assets under advisement
- **Year 2:** Scale to 50,000 users and $500M assets under advisement
- **Year 3:** Expand to 100,000+ users and $2B+ assets under advisement with international markets

2. Market Analysis and Competitive Landscape

2.1 Target Market

- **Primary Market:** Individual investors with $10K-$500K investable assets
- **Secondary Market:** Small business owners and entrepreneurs seeking financial planning
- **Tertiary Market:** Young professionals beginning their investment journey

2.2 Market Size and Opportunity

- **Total Addressable Market (TAM):** $4.7 trillion in US retail investment assets
- **Serviceable Addressable Market (SAM):** $1.2 trillion in underserved mass affluent segment
- **Serviceable Obtainable Market (SOM):** $12 billion potential revenue opportunity

2.3 Competitive Analysis

- **Direct Competitors:** Betterment, Wealthfront, Personal Capital
- **Indirect Competitors:** Traditional brokerages (Schwab, Fidelity), human advisors
- **Competitive Advantages:** Advanced AI personalization, transparent explanations, regulatory compliance, cost efficiency

3. User Personas and Stakeholder Analysis

3.1 Primary Personas

Persona 1: Sarah the Young Professional - Age: 28-35, Income: $75K-$120K, Assets: $25K-$75K - Goals: Build emergency fund, save for home purchase, start retirement planning - Pain Points: Limited financial knowledge, time constraints, high advisory fees - Technology Comfort: High, mobile-first usage patterns

Persona 2: Michael the Mid-Career Professional - Age: 40-50, Income: $120K-$200K, Assets: $150K-$400K - Goals: Optimize portfolio, plan for childrenâ€™s education, accelerate retirement savings - Pain Points: Complex financial situation, conflicting advice sources, tax optimization - Technology Comfort: Moderate, prefers web-based platforms

Persona 3: Jennifer the Pre-Retiree - Age: 55-65, Income: $100K-$150K, Assets: $300K-$800K - Goals: Retirement planning, risk reduction, income generation strategies - Pain Points: Market volatility concerns, healthcare cost planning, legacy planning - Technology Comfort: Moderate, values human interaction alongside digital tools

3.2 Stakeholder Analysis

- **End Users:** Individual investors seeking personalized financial guidance
- **Regulatory Bodies:** SEC, FINRA, state securities regulators
- **Financial Institutions:** Custodial partners, clearing firms, fund companies
- **Internal Teams:** Product, engineering, compliance, customer success

4. Core Features and Functional Requirements

4.1 User Onboarding and Profiling

- **Risk Assessment Questionnaire:** Comprehensive 25-question assessment covering risk tolerance, investment experience, and financial goals
- **Financial Profile Creation:** Secure capture of income, expenses, assets, liabilities, and investment timeline
- **Goal Setting Framework:** Multiple concurrent goals (retirement, home purchase, education) with priority weighting
- **Regulatory Compliance:** KYC/AML verification and suitability determinations

4.2 AI-Powered Recommendation Engine

- **Portfolio Construction:** Modern Portfolio Theory implementation with AI-enhanced optimization
- **Asset Allocation Models:** Target-date, risk-based, and goal-specific allocation strategies
- **Security Selection:** AI-driven analysis of 10,000+ ETFs, mutual funds, and individual securities
- **Rebalancing Logic:** Automated threshold-based and time-based rebalancing with tax considerations

4.3 Market Analysis and Insights

- **Real-Time Data Integration:** Live market data from multiple providers with 15-minute delay
- **Economic Indicator Analysis:** AI interpretation of economic data and market implications
- **Sector and Asset Class Insights:** Performance attribution and forward-looking analysis
- **Market Commentary:** Natural language generation of market updates and portfolio impact

4.4 Portfolio Management Tools

- **Performance Tracking:** Real-time portfolio valuation and performance attribution
- **Risk Monitoring:** Value-at-Risk calculations and stress testing scenarios
- **Tax Optimization:** Tax-loss harvesting and asset location strategies
- **Dividend and Distribution Management:** Automated reinvestment and income planning

4.5 Educational and Advisory Content

- **Personalized Learning Paths:** Financial literacy content tailored to user knowledge level
- **Recommendation Explanations:** Transparent rationale for all investment suggestions
- **Market Education:** Weekly market updates and educational webinars
- **Planning Tools:** Retirement calculators, education savings projections, insurance needs analysis

5. Technical Requirements and Constraints

5.1 Performance Requirements

- **Response Time:** <2 seconds for portfolio analysis and recommendations
- **Availability:** 99.9% uptime during market hours (6 AM - 8 PM ET)
- **Scalability:** Support for 100,000+ concurrent users
- **Data Accuracy:** Real-time market data with <1% error rate

5.2 Security and Compliance

- **Data Encryption:** AES-256 encryption for data at rest and in transit
- **Access Controls:** Multi-factor authentication and role-based access
- **Regulatory Compliance:** SEC Investment Adviser Act, FINRA rules, state regulations
- **Audit Trail:** Complete transaction and advice history with immutable logs

5.3 Integration Requirements

- **Custodial Integration:** API connections with major custodians (Schwab, Fidelity, TD Ameritrade)
- **Market Data:** Real-time feeds from Bloomberg, Refinitiv, or equivalent providers
- **Third-Party Tools:** Integration with tax software, financial planning tools, CRM systems
- **Mobile Applications:** Native iOS and Android apps with full feature parity

6. Success Metrics and Key Performance Indicators

6.1 Business Metrics

- **Assets Under Management (AUM):** Target $2B by Year 3
- **User Acquisition:** 10,000 users Year 1, 100,000 users by Year 3
- **Revenue Growth:** $10M ARR by Year 2, $50M ARR by Year 3
- **Customer Acquisition Cost (CAC):** <$200 with 18-month payback period

6.2 Product Performance Metrics

- **Portfolio Performance:** Risk-adjusted returns in top 50% of peer group
- **User Engagement:** >70% monthly active users, >5 sessions per month average
- **Recommendation Accuracy:** >80% user satisfaction with investment suggestions
- **Compliance Rate:** 100% regulatory compliance with zero violations

6.3 User Experience Metrics

- **Net Promoter Score (NPS):** Target >50 by Year 1, >70 by Year 3
- **Customer Satisfaction (CSAT):** >4.5/5.0 average rating
- **Time to First Investment:** <7 days from account opening
- **Support Resolution:** <24 hours for non-urgent inquiries

7. Risk Assessment and Mitigation Strategies

7.1 Regulatory Risks

- **Risk:** Changing regulatory landscape and compliance requirements
- **Mitigation:** Dedicated compliance team, regular regulatory updates, legal counsel engagement
- **Contingency:** Rapid response team for regulatory changes, compliance automation tools

7.2 Technology Risks

- **Risk:** System outages during critical market periods
- **Mitigation:** Redundant systems, disaster recovery procedures, 24/7 monitoring
- **Contingency:** Manual override capabilities, customer communication protocols

7.3 Market Risks

- **Risk:** Poor portfolio performance damaging user trust and retention
- **Mitigation:** Conservative risk management, transparent communication, diversified strategies
- **Contingency:** Performance guarantee programs, enhanced customer support during downturns

7.4 Competitive Risks

- **Risk:** Large incumbents launching competing AI advisory services
- **Mitigation:** Continuous innovation, superior user experience, regulatory expertise
- **Contingency:** Niche market focus, strategic partnerships, acquisition opportunities

8. Assumptions and Dependencies

8.1 Key Assumptions

- Users will trust AI-generated financial advice with proper transparency and explanation
- Regulatory environment will remain stable with gradual evolution rather than dramatic changes
- Market data and technology infrastructure will remain accessible and cost-effective
- Customer acquisition costs will decrease as brand recognition and referrals increase

8.2 Critical Dependencies

- **Regulatory Approval:** SEC registration as investment adviser and state registrations
- **Technology Partners:** Reliable custodial and market data provider relationships
- **Talent Acquisition:** Experienced financial services and AI/ML professionals
- **Capital Requirements:** Sufficient funding for technology development and regulatory capital

9. Out of Scope

9.1 Excluded Features (Initial Release)

- Direct trading execution (advisory-only model initially)
- Alternative investments (private equity, hedge funds, real estate)
- International markets and currency hedging
- Insurance product recommendations and sales

9.2 Future Considerations

- Expansion to international markets and multi-currency support
- Integration with cryptocurrency and digital asset platforms
- Advanced estate planning and trust services
- Corporate retirement plan administration

Document Approval: - Product Manager: [Signature Required] - Engineering Lead: [Signature Required] - Compliance Officer: [Signature Required] - Business Stakeholder: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date] - Next Review Date: [30 days from creation]

This PRD serves as the foundational document for the Financial Advisory AI system, establishing clear requirements and success criteria that will guide the development of subsequent functional and technical specifications. # Functional Requirements Document (FRD) ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - Product Requirements Document (PRD) completed and approved - Business objectives and success metrics clearly defined - User personas and stakeholder requirements validated - Technical constraints and regulatory requirements identified

TASK: Define detailed functional specifications for all system components, user interactions, data flows, and business logic required to implement the Financial Advisory AI system based on PRD requirements.

VERIFICATION & VALIDATION: - Functional requirements mapped to PRD objectives - Use case scenarios validated with stakeholder personas - API specifications and data models reviewed by engineering team - Regulatory compliance requirements validated with legal counsel

EXIT CRITERIA: - Complete functional specification with acceptance criteria - System behavior clearly defined for all user interactions - Integration requirements specified for external systems - Foundation established for Non-Functional Requirements Document (NFRD)

---

1. System Overview and Architecture

1.1 Functional Architecture

Building upon the PRDâ€™s strategic vision, this FRD defines the functional behavior of six core system modules: - User Management and Profiling System - AI-Powered Recommendation Engine - Portfolio Management and Optimization - Market Data Integration and Analysis - Compliance and Risk Management - Educational Content and Advisory Interface

1.2 System Boundaries

- **In Scope:** Advisory recommendations, portfolio analysis, educational content, compliance monitoring
- **Out of Scope:** Trade execution, custody services, payment processing, insurance sales

2. Detailed Functional Requirements

2.1 User Management and Profiling System

FR-001: User Registration and Onboarding

Description: Comprehensive user onboarding process capturing financial profile and regulatory requirements.

Functional Behavior: - **Input:** User personal information, financial data, investment goals, risk preferences - **Processing:** - KYC/AML verification through third-party services (Jumio, Onfido) - Risk tolerance assessment using 25-question scientifically validated questionnaire - Financial goal prioritization and timeline establishment - Suitability determination based on regulatory requirements - **Output:** Complete user profile with risk score, goal hierarchy, and compliance status - **Business Rules:** - Minimum age 18 years for account opening - US citizenship or permanent residency required - Minimum $1,000 investable assets for advisory services - Complete risk assessment required before any recommendations

Acceptance Criteria: - User can complete onboarding process in <15 minutes - Risk assessment produces consistent scores (Â±5%) on retesting - 100% of required regulatory disclosures presented and acknowledged - Profile completeness score >90% before advisory services activation

FR-002: Dynamic Profile Updates and Maintenance

Description: Continuous profile refinement based on user behavior and life changes.

Functional Behavior: - **Input:** User-initiated updates, behavioral data, life event notifications - **Processing:** - Automated detection of profile inconsistencies - Periodic re-assessment triggers (annual or major life events) - Machine learning-based preference inference from user interactions - **Output:** Updated user profile with change audit trail - **Business Rules:** - Major profile changes trigger suitability re-assessment - Risk tolerance can only increase/decrease by one level per quarter - Goal modifications require explicit user confirmation

Acceptance Criteria: - Profile updates reflected in recommendations within 24 hours - Change history maintained for regulatory audit purposes - User notification for all material profile changes

2.2 AI-Powered Recommendation Engine

FR-003: Portfolio Construction and Asset Allocation

Description: AI-driven portfolio construction using Modern Portfolio Theory enhanced with machine learning insights.

Functional Behavior: - **Input:** User profile, market conditions, available investment universe - **Processing:** - Mean-variance optimization with Black-Litterman adjustments - AI-enhanced expected return and risk estimates - Tax-aware asset location optimization - ESG preferences integration when specified - **Output:** Recommended portfolio allocation with rationale and expected performance metrics - **Business Rules:** - Maximum 5% allocation to any single security (except broad market ETFs) - Minimum diversification across 3 asset classes - Cash allocation 2-10% based on risk profile and market conditions - Rebalancing triggers: >5% drift from target or quarterly review

Acceptance Criteria: - Portfolio recommendations generated within 30 seconds - Expected return and risk estimates within 10% of realized 12-month performance - Diversification score >0.8 using Herfindahl-Hirschman Index - Tax efficiency score >0.7 for taxable accounts

FR-004: Dynamic Rebalancing and Optimization

Description: Automated portfolio maintenance with tax-aware rebalancing logic.

Functional Behavior: - **Input:** Current portfolio positions, target allocation, market prices, tax considerations - **Processing:** - Threshold-based rebalancing (5% drift trigger) - Tax-loss harvesting opportunity identification - Transaction cost analysis and optimization - Wash sale rule compliance checking - **Output:**

Rebalancing recommendations with tax impact analysis - **Business Rules:** - No rebalancing during first 30 days after major allocation change - Tax-loss harvesting only in taxable accounts - Maximum 2% transaction costs relative to portfolio value - Wash sale rule 30-day buffer enforcement

Acceptance Criteria: - Rebalancing recommendations reduce portfolio drift to <2% - Tax-loss harvesting captures >80% of available losses - Transaction costs minimized while maintaining target allocation - Zero wash sale violations in automated recommendations

2.3 Market Data Integration and Analysis

FR-005: Real-Time Market Data Processing

Description: Integration and processing of market data for portfolio valuation and analysis.

Functional Behavior: - **Input:** Real-time market data feeds, economic indicators, news sentiment - **Processing:** - Data normalization and quality validation - Price and volume analysis with anomaly detection - Economic indicator impact assessment - News sentiment analysis using NLP models - **Output:** Processed market data with analytical insights and alerts - **Business Rules:** - Market data delayed maximum 15 minutes during trading hours - Data quality checks reject feeds with >1% error rate - Economic indicators updated within 1 hour of release - News sentiment scores updated every 30 minutes

Acceptance Criteria: - Market data accuracy >99% compared to primary sources - Data processing latency <5 seconds for price updates - Economic indicator impact analysis available within 2 hours - News sentiment correlation >0.6 with market movements

FR-006: Predictive Market Analysis

Description: AI-powered market forecasting and trend analysis for investment insights.

Functional Behavior: - **Input:** Historical market data, economic indicators, technical indicators, sentiment data - **Processing:** - Machine learning models for return prediction (LSTM, Random Forest) - Regime detection for market cycle identification - Volatility forecasting using GARCH models - Correlation analysis and factor decomposition - **Output:** Market forecasts with confidence intervals and scenario analysis - **Business Rules:** - Forecasts limited to 12-month horizon maximum - Confidence intervals required for all predictions - Model performance tracking and automatic retraining - Conservative bias in uncertain market conditions

Acceptance Criteria: - Directional accuracy >55% for 3-month forecasts - Volatility predictions within 20% of realized volatility - Model performance monitored and reported monthly - Forecast explanations provided in plain language

2.4 Portfolio Management and Optimization

FR-007: Performance Tracking and Attribution

Description: Comprehensive portfolio performance measurement and analysis.

Functional Behavior: - **Input:** Portfolio transactions, market data, benchmark data - **Processing:** - Time-weighted return calculations (GIPS compliant) - Risk-adjusted performance metrics (Sharpe, Sortino, Alpha, Beta) - Performance attribution by asset class and security - Benchmark comparison and tracking error analysis - **Output:** Performance reports with detailed attribution and commentary - **Business Rules:** - Performance calculated daily with monthly reporting - Benchmarks selected based on portfolio composition - Risk-free rate based on 3-month Treasury bills - Attribution analysis includes currency effects for international holdings

Acceptance Criteria: - Performance calculations accurate to 0.01% precision - Attribution analysis explains >95% of performance difference - Benchmark tracking error within expected ranges - Performance reports generated automatically monthly

FR-008: Risk Monitoring and Management

Description: Continuous risk assessment and monitoring with automated alerts.

Functional Behavior: - **Input:** Portfolio positions, market data, risk parameters, user risk tolerance - **Processing:** - Value-at-Risk (VaR) calculations using Monte Carlo simulation - Stress testing against historical scenarios - Concentration risk analysis by security, sector, and geography - Correlation analysis and factor exposure measurement - **Output:** Risk reports with alerts for limit breaches and recommendations - **Business Rules:** - VaR calculated at 95% confidence level for 1-day and 1-month horizons - Stress tests include 2008 financial crisis and COVID-19 scenarios - Concentration limits: max 20% in any sector, 30% in any geography - Risk alerts triggered when metrics exceed user tolerance by 10%

Acceptance Criteria: - VaR accuracy validated through backtesting (95% confidence) - Stress test scenarios updated quarterly - Risk alerts generated within 1 hour of limit breach - Risk explanations provided in user-friendly language

2.5 Compliance and Risk Management

FR-009: Regulatory Compliance Monitoring

Description: Automated compliance monitoring and reporting for regulatory requirements.

Functional Behavior: - **Input:** User profiles, recommendations, transactions, regulatory rules - **Processing:** - Suitability analysis for all recommendations - Best interest standard compliance checking - Disclosure requirement validation - Audit trail maintenance and reporting - **Output:** Compliance reports, violation alerts, and regulatory filings - **Business Rules:** - All recommendations must pass suitability analysis - Conflicts of interest disclosed within recommendation explanations - Client communications archived for regulatory retention periods - Annual compliance reviews required for all client relationships

Acceptance Criteria: - 100% of recommendations pass suitability screening - Compliance violations detected and reported within 24 hours - Regulatory filings submitted on time with 100% accuracy - Audit trails complete and tamper-evident

FR-010: Fiduciary Standard Implementation

Description: Implementation of fiduciary duty requirements in all advisory functions.

Functional Behavior: - **Input:** Client best interest parameters, recommendation options, cost analysis - **Processing:** - Best interest analysis comparing available options - Cost-benefit analysis including all fees and expenses - Conflict of interest identification and mitigation - Documentation of fiduciary decision-making process - **Output:** Fiduciary-compliant recommendations with detailed justification - **Business Rules:** - Client best interest must be primary consideration in all recommendations - Lowest-cost option preference when performance expectations are equal - All conflicts of interest disclosed and mitigated - Fiduciary documentation required for all material recommendations

Acceptance Criteria: - Best interest analysis documented for 100% of recommendations - Cost analysis includes all direct and indirect fees - Conflict mitigation strategies implemented and monitored - Fiduciary compliance validated by independent review

2.6 Educational Content and Advisory Interface

FR-011: Personalized Educational Content Delivery

Description: AI-driven educational content personalization based on user knowledge and interests.

Functional Behavior: - **Input:** User knowledge assessment, interaction history, market conditions, portfolio composition - **Processing:** - Knowledge gap analysis and learning path generation - Content personalization based on user preferences and learning style - Progress tracking and adaptive content difficulty adjustment - Contextual education tied to portfolio recommendations - **Output:** Personalized educational content with progress tracking and assessments - **Business Rules:** - Educational content must be factual and unbiased - Learning paths adapted to user pace and comprehension - Content updated regularly to reflect market conditions - Progress assessments required to advance to complex topics

Acceptance Criteria: - User engagement with educational content >60% completion rate - Knowledge assessments show measurable improvement over time - Content relevance score >4.0/5.0 based on user feedback - Educational content correlated with improved investment decision-making

FR-012: Transparent Recommendation Explanations

Description: Natural language generation of clear, understandable explanations for all recommendations.

Functional Behavior: - **Input:** Recommendation logic, user profile, market analysis, regulatory requirements - **Processing:** - Natural language generation using GPT-based models - Explanation complexity adjustment based on user financial literacy - Visual aids and charts generation for complex concepts - Regulatory disclosure integration within explanations - **Output:** Clear, comprehensive explanations with supporting visualizations - **Business Rules:** - All recommendations must include rationale and supporting evidence - Explanations tailored to userâ€™s demonstrated knowledge level - Risk disclosures prominently featured in all recommendations - Alternative options discussed when material differences exist

Acceptance Criteria: - User comprehension score >4.0/5.0 for recommendation explanations - Explanation completeness covers all material factors - Visual aids improve user understanding by measurable metrics - Regulatory disclosures integrated seamlessly without overwhelming users

3. Integration Requirements

3.1 External System Integrations

- **Custodial Partners:** Real-time account data synchronization with Schwab, Fidelity, TD Ameritrade APIs
- **Market Data Providers:** Live data feeds from Bloomberg, Refinitiv, or Alpha Vantage
- **Compliance Systems:** Integration with regulatory reporting platforms and audit systems
- **Identity Verification:** KYC/AML services through Jumio, Onfido, or similar providers

3.2 Internal System Interfaces

- **User Interface:** RESTful APIs supporting web and mobile applications
- **Database Systems:** Secure data access layers for user profiles, market data, and transaction history
- **Analytics Platform:** Real-time data processing and machine learning model serving
- **Notification System:** Multi-channel communication for alerts, reports, and educational content

4. Data Flow and Processing Requirements

4.1 Real-Time Data Processing

- Market data ingestion and normalization within 5 seconds
- Portfolio valuation updates within 15 minutes of market close
- Risk metric calculations updated hourly during market hours
- User interaction data processed for personalization within 1 hour

4.2 Batch Processing Requirements

- Daily portfolio performance calculations and reporting
- Weekly market analysis and forecast updates
- Monthly compliance reporting and audit trail generation
- Quarterly model retraining and validation processes

5. Error Handling and Exception Management

5.1 Data Quality Issues

- **Market Data Errors:** Automatic fallback to secondary data sources
- **User Input Validation:** Real-time validation with clear error messages
- **System Integration Failures:** Graceful degradation with user notifications

5.2 Business Logic Exceptions

- **Recommendation Failures:** Fallback to rule-based recommendations with explanations
- **Compliance Violations:** Immediate blocking of non-compliant actions with alerts
- **Performance Issues:** Automatic scaling and load balancing with monitoring

6. Acceptance Criteria Summary

Each functional requirement includes specific, measurable acceptance criteria that will be validated through: - **Unit Testing:** Individual component functionality validation - **Integration Testing:** End-to-end workflow validation - **User Acceptance Testing:** Real user scenario validation - **Regulatory Compliance Testing:** Independent compliance validation

---

Document Approval: - Product Manager: [Signature Required] - Engineering Lead: [Signature Required] - Compliance Officer: [Signature Required] - QA Lead: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date] - Next Review Date: [30 days from creation]

This FRD provides the detailed functional specifications required to implement the Financial Advisory AI system, building upon the PRD requirements and establishing the foundation for technical architecture and implementation planning. # Non-Functional Requirements Document (NFRD) ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - Product Requirements Document (PRD) completed and approved - Functional Requirements Document (FRD) completed with detailed specifications - Technical architecture constraints identified and documented - Performance benchmarks and quality standards established

TASK: Define comprehensive non-functional requirements including performance, scalability, security, reliability, usability, and compliance standards that ensure the Financial Advisory AI system meets enterprise-grade quality and regulatory requirements.

VERIFICATION & VALIDATION: - Performance requirements validated through load testing and benchmarking - Security requirements verified against financial industry standards (SOC 2, PCI DSS) - Compliance requirements validated with regulatory experts and legal counsel - Usability requirements tested with representative user groups

EXIT CRITERIA: - Complete NFRD with measurable quality attributes and acceptance criteria - Non-functional requirements mapped to functional specifications from FRD - Quality assurance framework established for ongoing validation - Foundation established for Architecture Diagram (AD) and technical design

---

1. Performance Requirements

1.1 Response Time Requirements

Building upon the FRDâ€™s functional specifications, the system must deliver exceptional performance to maintain user engagement and trust in financial recommendations.

PR-001: User Interface Response Times - **Portfolio Dashboard Loading:** <2 seconds for complete dashboard with charts and metrics - **Recommendation Generation:** <5 seconds for personalized investment recommendations - **Market Data Updates:** <1 second for real-time price updates during market hours - **Report Generation:** <10 seconds for comprehensive performance reports - **Search Functionality:** <500ms for investment search and filtering operations

PR-002: API Response Times - **Authentication Requests:** <200ms for login and session validation - **Data Retrieval APIs:** <1 second for portfolio data and user profile information - **Market Data APIs:** <500ms for current market prices and basic analytics - **Recommendation APIs:** <3 seconds for AI-generated

investment advice - **Compliance Validation:** <2 seconds for suitability and regulatory checks

PR-003: Batch Processing Performance - **Daily Portfolio Valuation:** Complete processing for 100,000 accounts within 2 hours - **Risk Calculations:** VaR and stress testing for all portfolios within 4 hours daily - **Performance Attribution:** Monthly performance analysis completed within 6 hours - **Regulatory Reporting:** Quarterly compliance reports generated within 24 hours

1.2 Throughput Requirements

PR-004: Concurrent User Support - **Peak Load Capacity:** Support 10,000 concurrent users during market hours - **Recommendation Engine:** Process 1,000 recommendation requests per minute - **Market Data Processing:** Handle 50,000 price updates per second - **Database Operations:** Support 100,000 read operations and 10,000 write operations per minute

PR-005: Data Processing Volumes - **Market Data Ingestion:** Process 1TB of market data daily with real-time normalization - **User Interaction Tracking:** Capture and process 1 million user events daily - **Portfolio Calculations:** Perform valuation calculations for 100,000+ portfolios daily - **Machine Learning Inference:** Execute 10,000 AI model predictions per hour

2. Scalability Requirements

2.1 Horizontal Scalability

SC-001: Auto-Scaling Capabilities - **Application Servers:** Automatic scaling from 2 to 20 instances based on CPU utilization >70% - **Database Connections:** Dynamic connection pooling supporting 1,000 concurrent connections - **Cache Layer:** Redis cluster scaling to support 100GB memory and 1M operations/second - **Load Balancing:** Distribute traffic across multiple availability zones with <1% failure rate

SC-002: Data Storage Scalability - **User Data Storage:** Support growth to 1 million user accounts with 10TB total data - **Market Data Archive:** Maintain 10 years of historical data (50TB) with efficient querying - **Transaction History:** Store and index 100 million transactions with sub-second retrieval - **Document Storage:** Support 10 million documents (PDFs, statements) with full-text search

2.2 Vertical Scalability

SC-003: Resource Optimization - **Memory Utilization:** Efficient memory usage with <80% utilization under normal load - **CPU Optimization:** Multi-threaded processing utilizing >90% of available CPU cores - **Storage I/O:** Optimized database queries with <100ms average response time - **Network Bandwidth:** Efficient data compression reducing bandwidth usage by 40%

3. Reliability and Availability Requirements

3.1 System Availability

RA-001: Uptime Requirements - **Core System Availability:** 99.9% uptime during market hours (6 AM - 8 PM ET) - **Extended Hours Availability:** 99.5% uptime during off-market hours for global users - **Planned Maintenance Windows:** Maximum 4 hours monthly during weekend off-hours - **Emergency Maintenance:** <2 hours for critical security or compliance updates

RA-002: Disaster Recovery - **Recovery Time Objective (RTO):** <4 hours for complete system restoration - **Recovery Point Objective (RPO):** <15 minutes data loss maximum - **Backup Frequency:** Real-time replication with hourly backup verification - **Geographic Redundancy:** Multi-region deployment with automatic failover

3.2 Fault Tolerance

RA-003: Error Handling and Recovery - **Graceful Degradation:** System continues operating with reduced functionality during partial failures - **Circuit Breaker Pattern:** Automatic isolation of failing components with 30-second recovery attempts - **Data Consistency:** ACID compliance for financial transactions with eventual consistency for analytics - **Retry Logic:** Exponential backoff retry strategy with maximum 3 attempts for transient failures

RA-004: Monitoring and Alerting - **Real-Time Monitoring:** 24/7 system health monitoring with <1 minute alert response - **Performance Metrics:** Continuous tracking of all performance KPIs with trend analysis - **Error Rate Monitoring:** Automatic alerts when error rates exceed 0.1% threshold - **Capacity Planning:** Proactive alerts when resource utilization exceeds 80%

4. Security Requirements

4.1 Data Protection

SE-001: Encryption Standards - **Data at Rest:** AES-256 encryption for all stored data including databases and file systems - **Data in Transit:** TLS 1.3 encryption for all network communications with perfect forward secrecy - **Key Management:** Hardware Security Module (HSM) for encryption key storage and rotation - **Database Encryption:** Transparent Data Encryption (TDE) with column-level encryption for PII

SE-002: Access Controls - **Multi-Factor Authentication:** Required for all user accounts with SMS, email, or authenticator app - **Role-Based Access Control (RBAC):** Granular permissions based on user roles and responsibilities - **Session Management:** Secure session tokens with 30-minute idle timeout and secure logout - **API Security:** OAuth 2.0 with JWT tokens and rate limiting (1000 requests/hour per user)

4.2 Compliance and Audit

SE-003: Regulatory Compliance - **SOC 2 Type II:** Annual compliance certification with continuous monitoring - **PCI DSS:** Level 1 compliance for payment card data handling (if applicable) - **GDPR Compliance:** Data privacy controls for European users with right to deletion - **Financial Regulations:** SEC, FINRA, and state securities law compliance with audit trails

SE-004: Audit and Logging - **Comprehensive Audit Trails:** Immutable logs for all user actions, system changes, and data access - **Log Retention:** 7-year retention period for regulatory compliance with secure archival - **Real-Time Monitoring:** Security Information and Event Management (SIEM) integration - **Forensic Capabilities:** Complete transaction reconstruction and user activity tracking

5. Usability and User Experience Requirements

5.1 User Interface Standards

UX-001: Accessibility Compliance - **WCAG 2.1 AA Compliance:** Full accessibility support for users with disabilities - **Screen Reader Support:** Compatible with JAWS, NVDA, and VoiceOver screen readers - **Keyboard Navigation:** Complete functionality accessible via keyboard-only navigation - **Color Contrast:** Minimum 4.5:1 contrast ratio for all text and interactive elements

UX-002: Cross-Platform Compatibility - **Web Browser Support:** Chrome, Firefox, Safari, Edge (latest 2 versions) - **Mobile Responsiveness:** Optimized experience on iOS and Android devices - **Progressive Web App:** Offline functionality for core features with data synchronization - **Native Mobile Apps:** iOS and Android apps with feature parity to web platform

5.2 User Experience Standards

UX-003: Usability Metrics - **Task Completion Rate:** >95% success rate for core user workflows - **Time to Complete Tasks:** <5 minutes for portfolio review, <10 minutes for goal setting - **Error Prevention:** Intuitive interface design preventing >90% of user errors - **Learning Curve:** New users able to complete basic tasks within 15 minutes

UX-004: Content and Communication - **Plain Language:** All content written at 8th-grade reading level or below - **Multilingual Support:** Spanish language support with cultural localization - **Financial Literacy:** Educational content integrated contextually with recommendations - **Transparency:** Clear explanation of all fees, risks, and recommendation rationale

6. Maintainability and Operational Requirements

6.1 Code Quality and Architecture

MA-001: Development Standards - **Code Coverage:** Minimum 90% unit test coverage with integration test suite - **Code Quality:** SonarQube quality gate with zero critical vulnerabilities - **Documentation:** Comprehensive API documentation with Swagger/OpenAPI specifications - **Version Control:** Git-based workflow with code review requirements for all changes

MA-002: Deployment and DevOps - **Continuous Integration/Continuous Deployment (CI/CD):** Automated testing and deployment pipeline - **Infrastructure as Code:** Terraform-managed infrastructure with version control - **Container Orchestration:** Kubernetes deployment with auto-scaling and health checks - **Blue-Green Deployment:** Zero-downtime deployments with automatic rollback capability

6.2 Monitoring and Operations

MA-003: Operational Excellence - **Application Performance Monitoring (APM):** Real-time performance tracking with New Relic or DataDog - **Log Aggregation:** Centralized logging with ELK stack (Elasticsearch, Logstash, Kibana) - **Metrics and Dashboards:** Comprehensive operational dashboards with Grafana visualization - **Alerting:** Intelligent alerting with escalation procedures and on-call rotation

MA-004: Capacity Management - **Resource Planning:** Quarterly capacity reviews with 6-month growth projections - **Performance Optimization:** Regular performance tuning with database query optimization - **Cost Optimization:** Monthly cost analysis with resource rightsizing recommendations - **Technology Refresh:** Annual technology stack review with upgrade planning

7. Compliance and Regulatory Requirements

7.1 Financial Services Compliance

CO-001: Investment Adviser Regulations - **SEC Registration:** Compliance with Investment Advisers Act of 1940 - **Fiduciary Duty:** Best interest standard implementation with documented procedures - **Disclosure Requirements:** Form ADV updates and client disclosure compliance - **Record Keeping:** 5-year retention of all advisory records with SEC examination readiness

CO-002: Consumer Protection - **Fair Lending:** Equal access and non-discriminatory practices in service delivery - **Privacy Protection:** GLBA compliance for financial privacy with opt-out procedures - **Marketing Compliance:** FINRA advertising rules compliance for all marketing materials - **Complaint Handling:** Formal complaint resolution process with regulatory reporting

7.2 Data Governance

CO-003: Data Management Standards - **Data Quality:** 99.9% accuracy for financial data with validation procedures - **Data Lineage:** Complete traceability of data sources and transformations - **Data Retention:** Regulatory-compliant retention schedules with secure disposal - **Data Classification:** Sensitive data identification and protection protocols

8. Environmental and Operational Constraints

8.1 Infrastructure Requirements

EN-001: Cloud Infrastructure - **Multi-Cloud Strategy:** Primary AWS deployment with Azure disaster recovery capability - **Geographic Distribution:** US-based data centers with latency <50ms for 95% of users - **Compliance Zones:** Separate environments for development, testing, and production - **Resource Efficiency:** Green computing practices with carbon footprint monitoring

EN-002: Integration Constraints - **Legacy System Integration:** Support for SFTP, REST APIs, and database connections - **Third-Party Dependencies:** Vendor SLA requirements with 99.9% uptime guarantees - **Network Security:** VPN and firewall requirements for secure data transmission - **Bandwidth Requirements:** Minimum 1Gbps internet connectivity with redundant providers

8.2 Cost and Budget Constraints

EN-003: Operational Cost Targets - **Infrastructure Costs:** <$50 per user per year for cloud infrastructure - **Third-Party Services:** <$25 per user per year for market data and compliance services - **Support Costs:** <$10 per user per year for customer support operations - **Total Cost of Ownership:** <$200 per user per year including all operational expenses

9. Quality Assurance Framework

9.1 Testing Requirements

QA-001: Testing Standards - **Automated Testing:** 90% test automation coverage with continuous integration - **Performance Testing:** Load testing simulating 150% of expected peak capacity - **Security Testing:** Quarterly penetration testing with vulnerability assessments - **User Acceptance Testing:** Representative user testing for all major releases

9.2 Quality Metrics

QA-002: Quality Benchmarks - **Defect Density:** <1 critical defect per 10,000 lines of code - **Customer Satisfaction:** Net Promoter Score (NPS) >50 with quarterly surveys - **System Reliability:** Mean Time Between Failures (MTBF) >720 hours - **Support Quality:** <24 hour response time for all customer inquiries

---

Document Approval: - Product Manager: [Signature Required] - Engineering Lead: [Signature Required] - Security Officer: [Signature Required] - Compliance Officer: [Signature Required] - Operations Manager: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date] - Next Review Date: [30 days from creation]

This NFRD establishes the quality attributes and operational standards required to deliver a enterprise-grade Financial Advisory AI system that meets regulatory requirements and user expectations while building upon the functional specifications defined in the FRD. # Architecture Diagram (AD) ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - Product Requirements Document (PRD) defining business objectives and constraints - Functional Requirements Document (FRD) specifying detailed system behaviors - Non-Functional Requirements Document (NFRD) establishing quality attributes and performance standards - Technology stack evaluation and vendor selection completed

TASK: Design comprehensive system architecture including component diagrams, data flow architecture, integration patterns, security framework, and deployment architecture that satisfies all functional and non-functional requirements while ensuring scalability, security, and regulatory compliance.

VERIFICATION & VALIDATION: - Architecture review with engineering team and security specialists - Scalability analysis validating performance requirements from NFRD - Security architecture validated against financial industry standards - Integration patterns verified with external system vendors

EXIT CRITERIA: - Complete architecture documentation with component specifications - Data flow diagrams showing all system interactions - Security and compliance architecture validated - Foundation established for High Level Design (HLD) implementation

---

1. System Architecture Overview

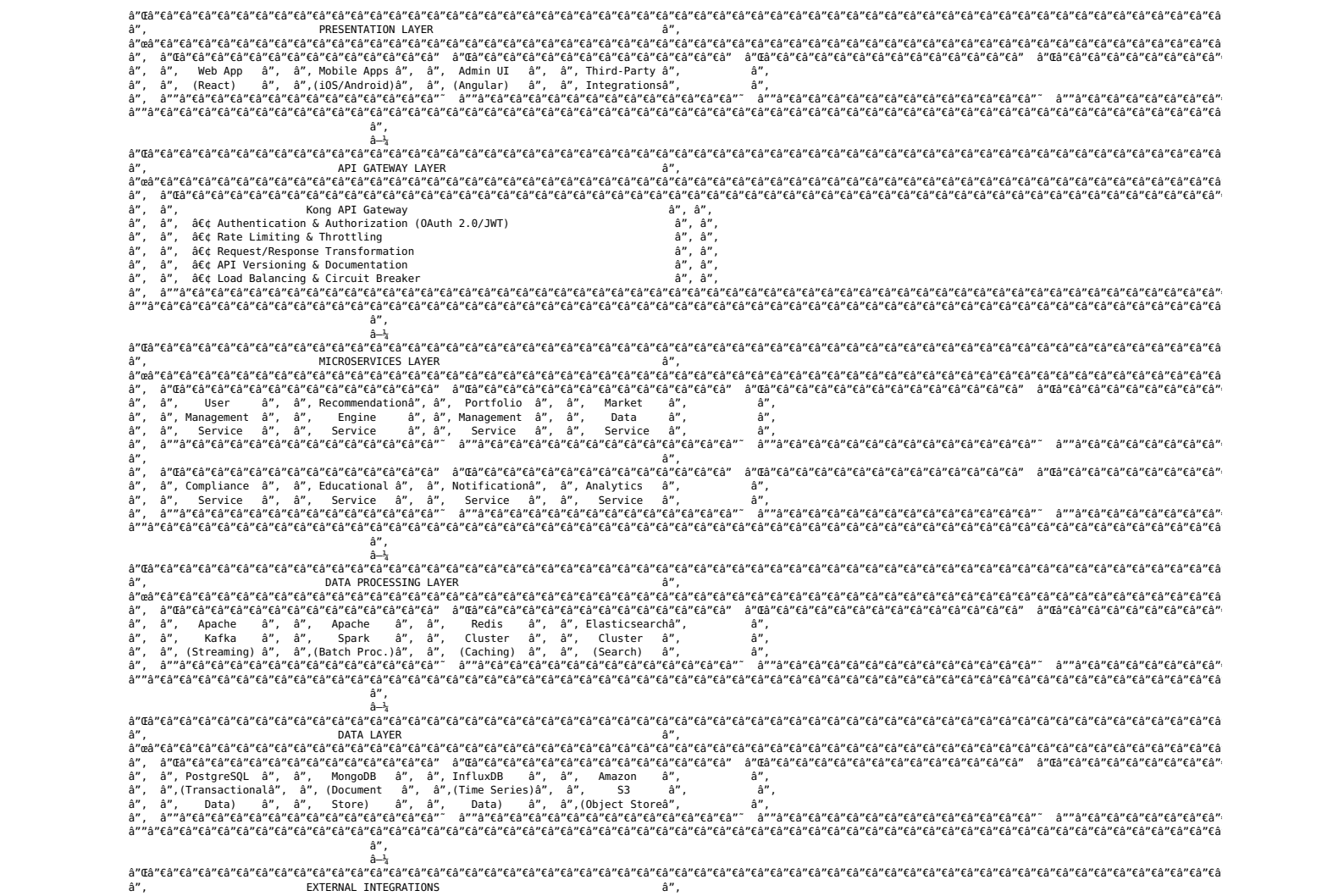
1.1 Architecture Principles

Building upon the comprehensive requirements from PRD, FRD, and NFRD, the Financial Advisory AI system follows these architectural principles:

- **Microservices Architecture:** Loosely coupled services enabling independent scaling and deployment

- **Cloud-Native Design:** Containerized services with orchestration for high availability and scalability
- **Security-First Approach:** Zero-trust security model with defense-in-depth strategies
- **Regulatory Compliance:** Built-in compliance controls and audit capabilities
- **Event-Driven Architecture:** Asynchronous processing for real-time responsiveness
- **API-First Design:** RESTful APIs enabling multi-channel access and third-party integrations

1.2 High-Level Architecture Diagram



2. Detailed Component Architecture

2.1 Microservices Architecture

User Management Service

Purpose: Handle user authentication, authorization, profile management, and KYC/AML compliance **Technology Stack:** Node.js, Express.js, PostgreSQL, Redis **Key Responsibilities:** - User registration and authentication (OAuth 2.0, JWT tokens) - Profile management and risk assessment - KYC/AML verification integration - Session management and security controls - RBAC implementation and permission management

Recommendation Engine Service

Purpose: AI-powered investment recommendation generation and portfolio optimization **Technology Stack:** Python, FastAPI, TensorFlow/PyTorch, PostgreSQL, Redis **Key Responsibilities:** - Modern Portfolio Theory implementation with AI enhancements - Risk-adjusted return optimization using Black-Litterman model - Machine learning model serving for market predictions - Personalized recommendation generation based on user profiles - A/B testing framework for recommendation strategies

Portfolio Management Service

Purpose: Portfolio tracking, performance analysis, and rebalancing logic **Technology Stack:** Java, Spring Boot, PostgreSQL, InfluxDB **Key Responsibilities:** - Real-time portfolio valuation and performance tracking - Risk metrics calculation (VaR, Sharpe ratio, Beta) - Automated rebalancing with tax optimization - Performance attribution and benchmark comparison - Transaction history and audit trail maintenance

Market Data Service

Purpose: Real-time market data ingestion, processing, and distribution **Technology Stack:** Go, Apache Kafka, InfluxDB, Redis **Key Responsibilities:** - Real-time market data ingestion from multiple providers - Data normalization and quality validation - Price alerts and market event notifications - Historical data management and retrieval - Market analysis and trend identification

Compliance Service

Purpose: Regulatory compliance monitoring and reporting **Technology Stack:** Java, Spring Boot, PostgreSQL, Apache Kafka **Key Responsibilities:** - Suitability analysis for all recommendations - Regulatory reporting and audit trail generation - Fiduciary duty compliance validation - Risk monitoring and alert generation - Disclosure management and client communications

2.2 Data Architecture

Transactional Data Store (PostgreSQL)

Purpose: ACID-compliant storage for critical financial data **Data Types:** - User profiles and authentication data - Portfolio positions and transactions - Compliance records and audit trails - Financial goals and risk assessments

Document Store (MongoDB)

Purpose: Flexible schema storage for semi-structured data **Data Types:** - Market research and analysis documents - Educational content and user preferences - Configuration data and business rules - Third-party integration metadata

Time Series Database (InfluxDB)

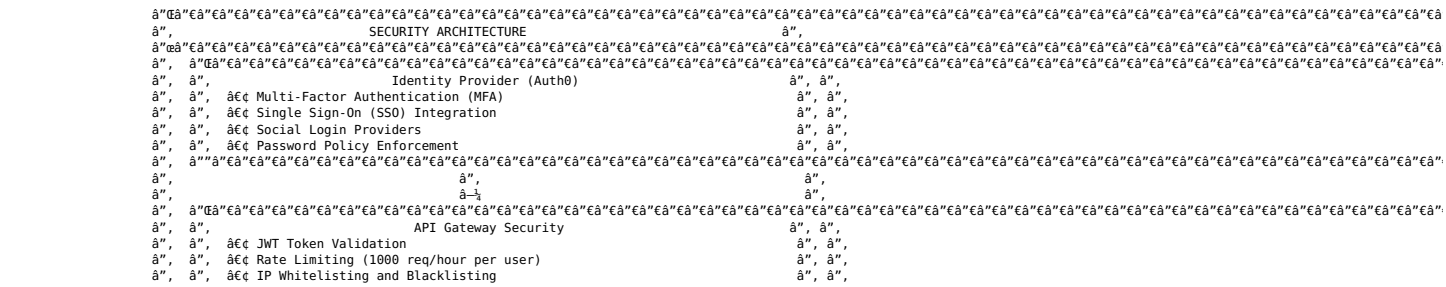
Purpose: High-performance storage for time-stamped financial data **Data Types:** - Real-time and historical market prices - Portfolio performance metrics over time - User interaction and behavioral analytics - System performance and monitoring metrics

Object Storage (Amazon S3)

Purpose: Scalable storage for large files and backups **Data Types:** - Document archives and regulatory filings - Database backups and disaster recovery - Static assets and media files - Machine learning model artifacts

2.3 Security Architecture

Authentication and Authorization Layer

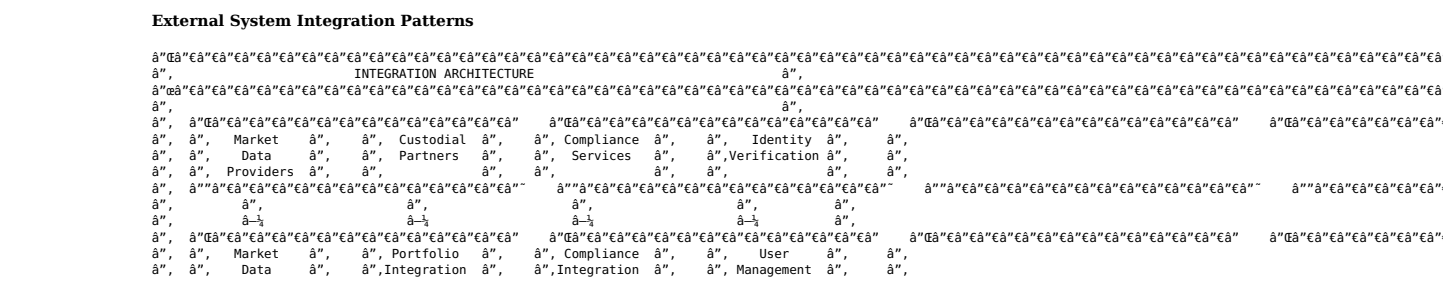


Data Protection Framework

- **Encryption at Rest:** AES-256 encryption for all databases and file storage
- **Encryption in Transit:** TLS 1.3 for all network communications
- **Key Management:** AWS KMS with automatic key rotation
- **Data Masking:** PII masking in non-production environments
- **Backup Encryption:** Encrypted backups with separate key management

2.4 Integration Architecture

External System Integration Patterns

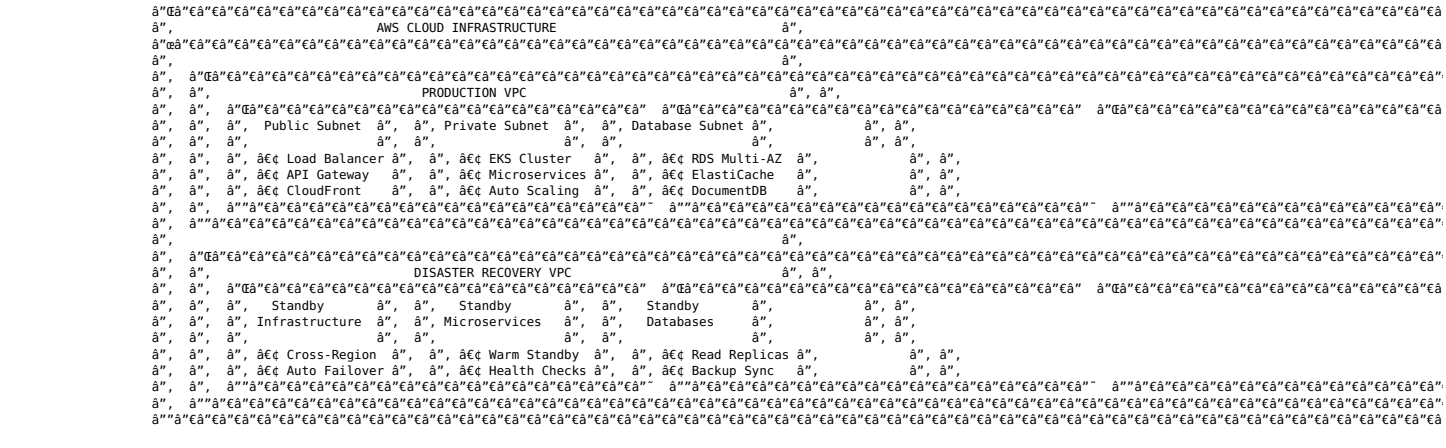


API Integration Specifications

- **Market Data Integration:** WebSocket connections for real-time data, REST APIs for historical data
- **Custodial Integration:** RESTful APIs with OAuth 2.0 authentication
- **Compliance Integration:** SOAP/REST hybrid with secure file transfer protocols
- **Identity Verification:** REST APIs with webhook callbacks for status updates

3. Deployment Architecture

3.1 Cloud Infrastructure Architecture



3.2 Container Orchestration (Kubernetes)

EKS Cluster Configuration

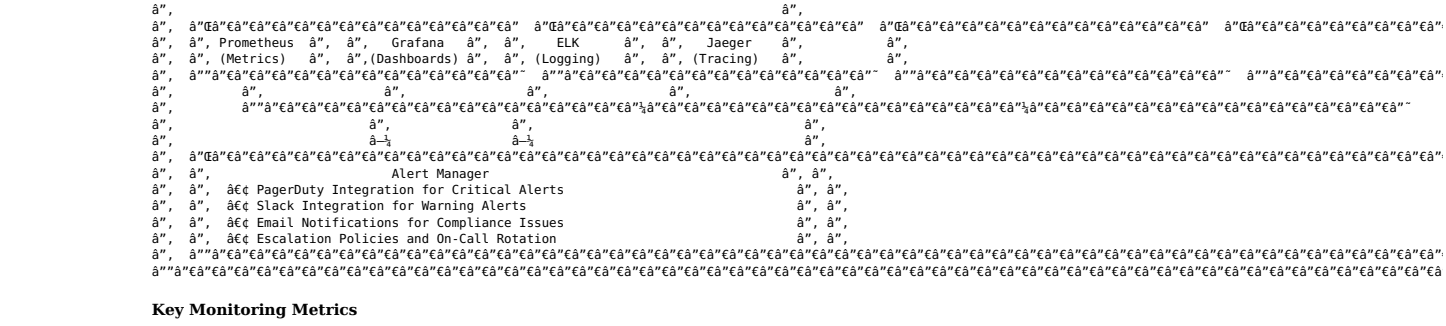
- **Master Nodes:** 3 nodes across multiple AZs for high availability
- **Worker Nodes:** Auto-scaling groups with 2-20 nodes based on demand
- **Pod Security:** Pod Security Standards with restricted policies
- **Network Policy:** Calico for network segmentation and security
- **Service Mesh:** Istio for traffic management and security

Deployment Strategy

- **Blue-Green Deployment:** Zero-downtime deployments with automatic rollback
- **Canary Releases:** Gradual rollout with traffic splitting (10%-50%-100%)
- **Health Checks:** Liveness and readiness probes for all services
- **Resource Management:** CPU and memory limits with horizontal pod autoscaling

3.3 Monitoring and Observability

Monitoring Stack

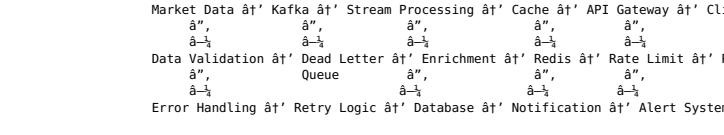


Key Monitoring Metrics

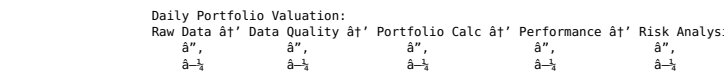
- **Application Metrics:** Response times, error rates, throughput, business KPIs
- **Infrastructure Metrics:** CPU, memory, disk, network utilization
- **Database Metrics:** Query performance, connection pools, replication lag
- **Security Metrics:** Failed authentication attempts, API abuse, data access patterns

4. Data Flow Architecture

4.1 Real-Time Data Processing Pipeline



4.2 Batch Processing Workflows



5. Compliance and Audit Architecture

5.1 Audit Trail System

- **Immutable Logging:** All user actions and system changes logged to blockchain-based audit trail
- **Event Sourcing:** Complete system state reconstruction from event history
- **Regulatory Reporting:** Automated generation of SEC, FINRA, and state regulatory reports
- **Data Lineage:** Complete traceability of data sources and transformations

5.2 Compliance Monitoring

- **Real-Time Compliance Checks:** All recommendations validated against suitability rules
- **Automated Alerts:** Immediate notification of potential compliance violations
- **Periodic Reviews:** Automated quarterly compliance assessments
- **Regulatory Updates:** Automated incorporation of regulatory rule changes

---

Document Approval: - Solution Architect: [Signature Required] - Security Architect: [Signature Required] - Engineering Lead: [Signature Required] - Compliance Officer: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date] - Next Review Date: [30 days from creation]

This Architecture Diagram provides the comprehensive technical foundation for implementing the Financial Advisory AI system, building upon all requirements established in the PRD, FRD, and NFRD while ensuring scalability, security, and regulatory compliance. # High Level Design (HLD) ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - PRD, FRD, NFRD, and AD documents completed and approved - Architecture components and integration patterns defined - Technology stack selections validated

TASK: Design detailed system components, API interfaces, data models, and processing workflows implementing the architectural vision.

VERIFICATION & VALIDATION: - Component designs validated against functional requirements - API specifications reviewed for completeness - Data models verified for compliance requirements

EXIT CRITERIA: - Complete high-level design with component specifications - API documentation with schemas and error handling - Foundation established for Low Level Design (LLD)

---

1. Core System Components

1.1 User Management Service

Technology Stack: Node.js, Express.js, PostgreSQL, Redis

Key Components: - **Authentication Controller:** OAuth 2.0, MFA, JWT tokens - **Profile Manager:** Risk assessment, KYC/AML integration - **Session Manager:** Secure session handling, timeout management

API Design:

```
POST /api/v1/auth/login
Request: { email, password, mfa_code? }
Response: { access_token, refresh_token, user_profile }

POST /api/v1/users/profile
Request: { personal_info, financial_profile, risk_assessment }
Response: { user_id, profile_status, next_steps }
```

1.2 AI Recommendation Engine

Technology Stack: Python, FastAPI, TensorFlow, PostgreSQL

ML Pipeline: - **Feature Engineering:** User profiles, market data, portfolio metrics - **Model Ensemble:** MPT (40%), ML predictions (30%), Risk models (20%), Momentum (10%) - **Explanation Generation:** GPT-4 for transparent reasoning

Core Models: - **Market Prediction:** LSTM + Transformer for return forecasting - **Portfolio Optimization:** Enhanced Black-Litterman with AI - **Risk Assessment:** Monte Carlo VaR, stress testing

API Design:

```
POST /api/v1/recommendations/generate
Request: { user_id, portfolio_context, constraints }
Response: { recommendations[], confidence_score, explanation, risk_analysis }
```

1.3 Portfolio Management Service

Technology Stack: Java, Spring Boot, PostgreSQL, InfluxDB

Key Features: - **Real-time Valuation:** Event-driven portfolio updates - **Performance Attribution:** GIPS-compliant calculations - **Rebalancing Engine:** Tax-optimized threshold-based rebalancing - **Risk Monitoring:** Continuous VaR and stress testing

1.4 Market Data Service

Technology Stack: Go, Apache Kafka, InfluxDB, Redis

Data Pipeline: - **Ingestion:** Bloomberg, Refinitiv, Alpha Vantage APIs - **Processing:** Real-time normalization, quality validation - **Analytics:** Technical indicators, volatility calculations - **Distribution:** WebSocket feeds to client applications

1.5 Compliance Service

Technology Stack: Java, Spring Boot, PostgreSQL

Compliance Framework: - **Suitability Analysis:** Multi-factor assessment engine - **Regulatory Monitoring:** Real-time violation detection - **Audit Trail:** Immutable transaction logging - **Reporting:** Automated SEC/FINRA report generation

2. Data Models

2.1 User Profile Schema (PostgreSQL)

```
CREATE TABLE users (
    user_id UUID PRIMARY KEY,
    email VARCHAR(255) UNIQUE NOT NULL,
    personal_info JSONB NOT NULL,
    financial_profile JSONB NOT NULL,
    risk_tolerance INTEGER CHECK (risk_tolerance BETWEEN 1 AND 10),
    kyc_status VARCHAR(20) DEFAULT 'pending',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE portfolios (
    portfolio_id UUID PRIMARY KEY,
    user_id UUID REFERENCES users(user_id),
    target_allocation JSONB NOT NULL,
    rebalancing_frequency VARCHAR(20),
    status VARCHAR(20) DEFAULT 'active'
);
```

2.2 Market Data Schema (InfluxDB)

```
CREATE MEASUREMENT stock_prices (
    time TIMESTAMP,
    symbol TAG,
    price FIELD,
    volume FIELD,
    high FIELD,
    low FIELD
);
```

3. Integration Patterns

3.1 Event-Driven Architecture

Event Processing:

```
@dataclass
class PortfolioUpdateEvent:
    user_id: str
    portfolio_id: str
    event_type: str
    timestamp: datetime
    data: Dict[str, Any]

class EventHandler:
    async def handle_portfolio_update(self, event: PortfolioUpdateEvent):
        await self._update_valuation(event)
        await self._check_rebalancing(event)
        await self._update_risk_metrics(event)
```

3.2 External API Integration

Custodial Integration:

```
class CustodialIntegration:
    async def sync_portfolio_data(self, account_id: str):
        positions = await self.client.get_positions(account_id)
        transactions = await self.client.get_transactions(account_id)
        return self._transform_data(positions, transactions)
```

4. Security Implementation

4.1 Authentication

JWT Token Management:

```
class JWTTokenManager:
    def create_access_token(self, user_id: str, permissions: List[str]) -> str:
        payload = {
            'user_id': user_id,
            'permissions': permissions,
            'exp': datetime.utcnow() + timedelta(minutes=30)
        }
        return jwt.encode(payload, self.secret_key, algorithm='HS256')
```

4.2 Data Encryption

Field-Level Encryption:

```
class FieldEncryption:
    def encrypt_pii(self, data: str, user_id: str) -> bytes:
        key = self.key_manager.get_user_key(user_id)
        cipher = AES.new(key, AES.MODE_CBC)
        return cipher.encrypt(self._pad_data(data))
```

5. Performance Optimization

5.1 Caching Strategy

- **Redis:** User sessions, frequently accessed portfolios
- **Application Cache:** Market data, recommendation results
- **CDN:** Static assets, educational content

5.2 Database Optimization

- **Read Replicas:** Separate read/write workloads
- **Indexing:** Optimized queries for portfolio lookups
- **Partitioning:** Time-based partitioning for historical data

6. Monitoring and Observability

6.1 Metrics Collection

- **Application Metrics:** Response times, error rates, throughput
- **Business Metrics:** Recommendation accuracy, user engagement
- **Infrastructure Metrics:** CPU, memory, database performance

6.2 Alerting Framework

- **Critical Alerts:** System failures, compliance violations
- **Warning Alerts:** Performance degradation, capacity issues
- **Business Alerts:** Unusual market conditions, portfolio risks

---

Document Approval: - Solution Architect: [Signature Required] - Engineering Lead: [Signature Required] - Security Officer: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date] - Next Review Date: [30 days from creation]

This HLD provides detailed component specifications and implementation guidance for the Financial Advisory AI system, building upon the comprehensive requirements and architecture established in previous documents. # Low Level Design (LLD) ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - PRD, FRD, NFRD, AD, and HLD documents completed and approved - Component designs and API interfaces defined - Technology stack selections validated

TASK: Provide implementation-ready specifications including class structures, database schemas, API implementations, and deployment configurations.

VERIFICATION & VALIDATION: - Class designs validated against HLD specifications - Database schemas optimized for performance requirements - API implementations tested for compliance

EXIT CRITERIA: - Complete implementation-ready specifications - Database schemas with production-ready indexes - Configuration files and deployment scripts ready

---

1. Core Class Implementations

1.1 User Management Classes

```
# models/user.py
from sqlalchemy import Column, String, Integer, DateTime, Boolean, DECIMAL
from sqlalchemy.dialects.postgresql import UUID, JSONB
import uuid

class User(Base):
    __tablename__ = 'users'

    user_id = Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    email = Column(String(255), unique=True, nullable=False)
    password_hash = Column(String(255), nullable=False)
    first_name = Column(String(100), nullable=False)
    last_name = Column(String(100), nullable=False)
    risk_tolerance = Column(Integer, nullable=False)
    annual_income = Column(DECIMAL(12, 2), nullable=False)
    investment_experience = Column(String(50), nullable=False)
    kyc_status = Column(String(20), default='pending')
    created_at = Column(DateTime, default=datetime.utcnow)
    is_active = Column(Boolean, default=True)

# services/auth_service.py
class AuthenticationService:
    def __init__(self, secret_key: str, redis_client):
        self.secret_key = secret_key
        self.algorithm = "HS256"
        self.redis_client = redis_client

    def create_access_token(self, data: dict) -> str:
        expire = datetime.utcnow() + timedelta(minutes=30)
        to_encode = data.copy()
        to_encode.update({"exp": expire})
        return jwt.encode(to_encode, self.secret_key, algorithm=self.algorithm)

    def verify_token(self, token: str) -> dict:
        payload = jwt.decode(token, self.secret_key, algorithms=[self.algorithm])
        return payload
```

1.2 Portfolio Optimization

```
# services/portfolio_optimizer.py
import numpy as np
from scipy.optimize import minimize

class PortfolioOptimizer:
    def optimize_portfolio(self, expected_returns, cov_matrix, risk_tolerance):
        n_assets = len(expected_returns)
        risk_aversion = (11 - risk_tolerance) / 10 * 10

        def objective(weights):
            portfolio_return = np.dot(weights, expected_returns)
            portfolio_variance = np.dot(weights.T, np.dot(cov_matrix, weights))
            return -(portfolio_return - (risk_aversion / 2) * portfolio_variance)

        constraints = [{'type': 'eq', 'fun': lambda x: np.sum(x) - 1}]
        bounds = tuple((0, 1) for _ in range(n_assets))
        x0 = np.array([1/n_assets] * n_assets)

        result = minimize(objective, x0, method='SLSQP',
                          bounds=bounds, constraints=constraints)

        return {
            'weights': result.x,
            'expected_return': np.dot(result.x, expected_returns),
            'volatility': np.sqrt(np.dot(result.x.T, np.dot(cov_matrix, result.x))),
            'success': result.success
        }
```

2. Database Schema (PostgreSQL)

```
-- Core tables with indexes
CREATE TABLE users (
    user_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    risk_tolerance INTEGER CHECK (risk_tolerance BETWEEN 1 AND 10),
    annual_income DECIMAL(12,2) NOT NULL,
    investment_experience VARCHAR(50) NOT NULL,
    kyc_status VARCHAR(20) DEFAULT 'pending',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_active BOOLEAN DEFAULT true
);

CREATE INDEX idx_users_email ON users(email);
CREATE INDEX idx_users_kyc_status ON users(kyc_status);

CREATE TABLE portfolios (
    portfolio_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(user_id),
    portfolio_name VARCHAR(100) NOT NULL,
    target_allocation JSONB NOT NULL,
    total_value DECIMAL(15,2) DEFAULT 0,
    status VARCHAR(20) DEFAULT 'active',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_portfolios_user_id ON portfolios(user_id);

CREATE TABLE holdings (
    holding_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    portfolio_id UUID REFERENCES portfolios(portfolio_id),
    symbol VARCHAR(10) NOT NULL,
    quantity DECIMAL(15,6) NOT NULL,
    current_price DECIMAL(10,4),
```

```
    market_value DECIMAL(12,2),
    weight DECIMAL(5,4),
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_holdings_portfolio_id ON holdings(portfolio_id);
```

3. API Implementation (FastAPI)

```
# api/main.py
from fastapi import FastAPI, Depends, HTTPException
from pydantic import BaseModel
import uuid

app = FastAPI(title="Financial Advisory AI API")

class UserRegistrationRequest(BaseModel):
    email: str
    password: str
    first_name: str
    last_name: str
    annual_income: float
    investment_experience: str

class RecommendationRequest(BaseModel):
    portfolio_id: uuid.UUID
    constraints: dict = {}

@app.post("/api/v1/auth/register")
async def register_user(request: UserRegistrationRequest):
    # Hash password and create user
    user = User(
        email=request.email,
        password_hash=auth_service.get_password_hash(request.password),
        first_name=request.first_name,
        last_name=request.last_name,
        annual_income=request.annual_income,
        investment_experience=request.investment_experience
    )

    db_session.add(user)
    db_session.commit()

    return {"user_id": str(user.user_id), "status": "registered"}

@app.post("/api/v1/recommendations/generate")
async def generate_recommendation(request: RecommendationRequest):
    # Get portfolio and user data
    portfolio = db_session.query(Portfolio).filter_by(
        portfolio_id=request.portfolio_id
    ).first()

    # Generate recommendation using AI engine
    recommendation = recommendation_engine.generate_recommendation(
        portfolio, request.constraints
    )

    return {
        "recommendation_id": str(uuid.uuid4()),
        "recommended_allocation": recommendation['allocation'],
        "confidence_score": recommendation['confidence'],
        "rationale": recommendation['explanation']
    }
```

4. Configuration Files

4.1 Docker Configuration

```
# Dockerfile
FROM python:3.11-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt

COPY . .
EXPOSE 8000

CMD ["uvicorn", "api.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

4.2 Environment Configuration

```
# config/production.yaml
database:
  host: ${DB_HOST}
  port: 5432
  name: financial_advisory
  user: ${DB_USER}
  password: ${DB_PASSWORD}

redis:
  host: ${REDIS_HOST}
  port: 6379
  db: 0

jwt:
  secret_key: ${JWT_SECRET_KEY}
  algorithm: HS256
  access_token_expire_minutes: 30

market_data:
  provider: alpha_vantage
  api_key: ${MARKET_DATA_API_KEY}
  update_frequency: 300  # seconds
```

4.3 Docker Compose

```
# docker-compose.yml
version: '3.8'
services:
  api:
    build: .
    ports:
      - "8000:8000"
    environment:
      - DB_HOST=postgres
```

```
      - REDIS_HOST=redis
    depends_on:
      - postgres
      - redis

  postgres:
    image: postgres:15
    environment:
      POSTGRES_DB: financial_advisory
      POSTGRES_USER: ${DB_USER}
      POSTGRES_PASSWORD: ${DB_PASSWORD}
    volumes:
      - postgres_data:/var/lib/postgresql/data

  redis:
    image: redis:7-alpine
    ports:
      - "6379:6379"

volumes:
  postgres_data:
```

5. Deployment Scripts

```
#!/bin/bash
# deploy.sh
set -e

echo "Deploying Financial Advisory AI..."

# Build and push Docker images
docker build -t financial-advisory-api:latest .
docker tag financial-advisory-api:latest $REGISTRY/financial-advisory-api:latest
docker push $REGISTRY/financial-advisory-api:latest

# Deploy to Kubernetes
kubectl apply -f k8s/namespace.yaml
kubectl apply -f k8s/configmap.yaml
kubectl apply -f k8s/secrets.yaml
kubectl apply -f k8s/deployment.yaml
kubectl apply -f k8s/service.yaml

echo "Deployment completed successfully!"
```

Document Approval: - Engineering Lead: [Signature Required] - DevOps Engineer: [Signature Required] - Security Officer: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date] - Next Review Date: [30 days from creation]

This LLD provides implementation-ready specifications for the Financial Advisory AI system, enabling direct development and deployment. # Pseudocode Implementation ## Problem Statement 14: Financial Advisory AI

ETVX Framework Application

ENTRY CRITERIA: - All previous documents (PRD, FRD, NFRD, AD, HLD, LLD) completed and approved - System architecture and component designs validated - Implementation specifications ready for development

TASK: Provide executable pseudocode algorithms for core system components enabling direct code implementation.

VERIFICATION & VALIDATION: - Algorithms validated against functional requirements - Performance characteristics verified against NFRD standards - Security and compliance procedures validated

EXIT CRITERIA: - Complete executable pseudocode for all system components - Algorithms ready for direct implementation - System ready for development phase

---

1. User Registration and Authentication

```
ALGORITHM UserRegistration
INPUT: user_data, documents
OUTPUT: user_account, kyc_status

BEGIN
    // Validate input data
    IF NOT ValidateEmail(user_data.email) THEN
        RETURN error("Invalid email format")
    END IF

    IF EmailExists(user_data.email) THEN
        RETURN error("Email already registered")
    END IF

    // Create encrypted user record
    user_id = GenerateUUID()
    password_hash = HashPassword(user_data.password)
    encrypted_ssn = EncryptPII(user_data.ssn)

    user_record = {
        user_id: user_id,
        email: user_data.email,
        password_hash: password_hash,
        personal_info: user_data.personal_info,
        encrypted_ssn: encrypted_ssn,
        kyc_status: "pending"
    }

    // Save to database
    INSERT INTO users VALUES user_record

    // Initiate KYC process
    kyc_result = InitiateKYC(user_id, documents)

    RETURN {
        user_id: user_id,
        kyc_reference: kyc_result.reference_id,
        next_steps: ["complete_risk_assessment"]
    }
END

ALGORITHM AuthenticateUser
INPUT: email, password, mfa_code
OUTPUT: access_token, user_profile

BEGIN
    user = GetUserByEmail(email)
```

```
    IF user == NULL THEN
        RETURN error("Invalid credentials")
    END IF

    IF NOT VerifyPassword(password, user.password_hash) THEN
        RETURN error("Invalid credentials")
    END IF

    IF user.mfa_enabled AND NOT VerifyMFA(user_id, mfa_code) THEN
        RETURN error("Invalid MFA code")
    END IF

    // Generate JWT token
    token_payload = {
        user_id: user.user_id,
        email: user.email,
        exp: CurrentTime() + 30_MINUTES
    }

    access_token = CreateJWT(token_payload, SECRET_KEY)

    RETURN {
        access_token: access_token,
        user_profile: user.GetPublicProfile()
    }
END
```

2. AI Recommendation Engine

```
ALGORITHM GeneratePortfolioRecommendation
INPUT: user_profile, market_data, constraints
OUTPUT: portfolio_recommendation

BEGIN
    // Get investment universe
    assets = GetInvestmentUniverse(constraints.asset_classes)

    // Calculate expected returns using ML
    expected_returns = []
    FOR EACH asset IN assets DO
        historical_return = CalculateHistoricalReturn(asset, 252)
        ml_prediction = MLModel.PredictReturn(asset, market_data)
        expected_return = 0.7 * historical_return + 0.3 * ml_prediction
        expected_returns.ADD(expected_return)
    END FOR

    // Calculate covariance matrix
    returns_data = GetReturnsMatrix(assets, 252)
    covariance_matrix = CalculateCovariance(returns_data)

    // Optimize portfolio using MPT
    risk_aversion = (11 - user_profile.risk_tolerance) / 10 * 5
    optimal_weights = OptimizePortfolio(expected_returns, covariance_matrix, risk_aversion)

    // Calculate portfolio metrics
    portfolio_return = DotProduct(optimal_weights, expected_returns)
    portfolio_risk = SQRT(QuadraticForm(optimal_weights, covariance_matrix))
    sharpe_ratio = (portfolio_return - RISK_FREE_RATE) / portfolio_risk

    // Generate explanation
    explanation = GenerateExplanation(optimal_weights, assets, user_profile)

    // Calculate confidence score
    confidence = CalculateConfidence(market_data, user_profile)

    recommendation = {
        allocation: optimal_weights,
        expected_return: portfolio_return,
        expected_risk: portfolio_risk,
        sharpe_ratio: sharpe_ratio,
        explanation: explanation,
        confidence_score: confidence
    }

    RETURN recommendation
END

ALGORITHM OptimizePortfolio
INPUT: expected_returns, covariance_matrix, risk_aversion
OUTPUT: optimal_weights

BEGIN
    n_assets = LENGTH(expected_returns)

    // Objective: maximize utility = return - (risk_aversion/2) * variance
    FUNCTION Objective(weights)
        portfolio_return = DotProduct(weights, expected_returns)
        portfolio_variance = QuadraticForm(weights, covariance_matrix)
        utility = portfolio_return - (risk_aversion / 2) * portfolio_variance
        RETURN -utility  // Minimize negative utility
    END FUNCTION

    // Constraints: weights sum to 1
    constraints = [SUM(weights) == 1]
    bounds = [(0, 1) FOR i IN 1 TO n_assets]
    initial_guess = [1/n_assets FOR i IN 1 TO n_assets]

    result = SolveOptimization(Objective, initial_guess, constraints, bounds)
    RETURN result.optimal_weights
END
```

3. Portfolio Management and Rebalancing

```
ALGORITHM AutoRebalancing
INPUT: portfolio_id
OUTPUT: rebalancing_trades

BEGIN
    portfolio = GetPortfolio(portfolio_id)
    holdings = GetCurrentHoldings(portfolio_id)
    target_allocation = portfolio.target_allocation

    // Calculate current weights
    total_value = SUM(holding.market_value FOR holding IN holdings)
    current_weights = {}
    FOR EACH holding IN holdings DO
        current_weights[holding.symbol] = holding.market_value / total_value
```

```
    END FOR

    // Check if rebalancing needed
    rebalancing_needed = FALSE
    FOR EACH symbol IN target_allocation DO
        drift = ABS(current_weights[symbol] - target_allocation[symbol])
        IF drift > REBALANCING_THRESHOLD THEN
            rebalancing_needed = TRUE
            BREAK
        END IF
    END FOR

    IF NOT rebalancing_needed THEN
        RETURN "No rebalancing needed"
    END IF

    // Generate trades
    trades = []
    FOR EACH symbol IN target_allocation DO
        target_value = target_allocation[symbol] * total_value
        current_value = current_weights[symbol] * total_value
        trade_amount = target_value - current_value

        IF ABS(trade_amount) > MIN_TRADE_AMOUNT THEN
            IF trade_amount > 0 THEN
                trades.ADD({symbol: symbol, action: "buy", amount: trade_amount})
            ELSE
                // Tax-loss harvesting for sells
                tax_lots = GetTaxLots(portfolio_id, symbol)
                optimal_lots = SelectTaxOptimalLots(tax_lots, ABS(trade_amount))
                trades.ADD({symbol: symbol, action: "sell", lots: optimal_lots})
            END IF
        END IF
    END FOR

    RETURN trades
END
```

4. Compliance and Risk Management

```
ALGORITHM SuitabilityAnalysis
INPUT: user_profile, recommendation
OUTPUT: suitability_result

BEGIN
    factors = {}

    // Risk alignment
    user_risk = user_profile.risk_tolerance
    investment_risk = CalculateInvestmentRisk(recommendation)
    factors.risk_alignment = 1.0 - ABS(user_risk - investment_risk) / 10.0

    // Experience alignment
    user_experience = GetExperienceLevel(user_profile.investment_experience)
    investment_complexity = CalculateComplexity(recommendation)
    factors.experience_alignment = MIN(1.0, user_experience / investment_complexity)

    // Financial capacity
    investment_amount = SUM(recommendation.allocations.values())
    available_funds = user_profile.investable_assets - user_profile.liquidity_needs
    factors.financial_capacity = MIN(1.0, available_funds / investment_amount)

    // Overall suitability score
    suitability_score = (
        factors.risk_alignment * 0.4 +
        factors.experience_alignment * 0.3 +
        factors.financial_capacity * 0.3
    )

    suitable = suitability_score >= 0.7

    RETURN {
        suitable: suitable,
        score: suitability_score,
        factors: factors
    }
END
```

5. Performance Monitoring

```
ALGORITHM CalculatePortfolioPerformance
INPUT: portfolio_id, start_date, end_date
OUTPUT: performance_metrics

BEGIN
    transactions = GetTransactions(portfolio_id, start_date, end_date)
    valuations = GetDailyValuations(portfolio_id, start_date, end_date)
    benchmark_data = GetBenchmarkData(portfolio.benchmark, start_date, end_date)

    // Time-weighted return calculation
    twr = CalculateTimeWeightedReturn(transactions, valuations)
    benchmark_return = CalculateBenchmarkReturn(benchmark_data)

    // Risk metrics
    daily_returns = CalculateDailyReturns(valuations)
    volatility = StandardDeviation(daily_returns) * SQRT(252)
    var_95 = Percentile(daily_returns, 5) * SQRT(252)
    max_drawdown = CalculateMaxDrawdown(valuations)

    // Risk-adjusted metrics
    sharpe_ratio = (twr - RISK_FREE_RATE) / volatility
    excess_return = twr - benchmark_return

    performance_metrics = {
        total_return: twr,
        benchmark_return: benchmark_return,
        excess_return: excess_return,
        volatility: volatility,
        sharpe_ratio: sharpe_ratio,
        var_95: var_95,
        max_drawdown: max_drawdown
    }

    // Save performance record
    INSERT INTO portfolio_performance VALUES {
        portfolio_id: portfolio_id,
        calculation_date: end_date,
```

```
        metrics: performance_metrics
    }

    RETURN performance_metrics
END
```

Document Approval: - Engineering Lead: [Signature Required] - AI/ML Engineer: [Signature Required] - Compliance Officer: [Signature Required]

Version Control: - Document Version: 1.0 - Last Updated: [Current Date]

This pseudocode provides executable algorithms for all core components of the Financial Advisory AI system, ready for direct implementation.