VAULTEX
Digital Trading and Wallet Platform
……..,
……

Date: December 16, 2025

SUPERVISOR:

Instructor Laraib Afzal

GROUP MEMBERS:

Shahan Ali	2024572
Mian Rabia Ilyas	2024289
Usman Musawar	2024655
Syed Hamza Samee	2024705

CO-SUPERVISOR:
TA Memona Sajid

Revision History:

Revision History	*Date*	*Comments*
1.00		
2.00		

Document Approval:

The following document has been accepted and approved by the following:

Signature	*Date*	*Name*

List of Contents

List of Figures

List of Tables

1. INTRODUCTION

PURPOSE

This document lays out what the Vaultex system is supposed to do, in the simplest terms possible. It's mainly meant to guide everyone involved in the project, so they understand what we're building and don't end up making assumptions later on. Vaultex is being developed for students in Pakistan who often find budgeting confusing or just don't have a reliable way to keep track of their daily expenses and early investment attempts. By collecting all the requirements here, the goal is to make the development process smoother and avoid unnecessary back-and-forth.

A lot of students struggle with money management—sometimes it's loose budgeting, sometimes it's uncertainty about how to even start saving or investing. Vaultex tries to simplify all that by putting several tools together: a wallet, a basic tracker, and a small-scale PKR trading feature. Some users may only want the tracking part, while others might try experimenting with the trading tools. Both are completely fine and are part of what the system aims to support.

PRODUCT SCOPE

Vaultex is built to help students handle their money in PKR while giving them a place to slowly improve their financial habits. Instead of jumping between a bunch of unrelated apps, users can access budgeting, wallet features, and a very simple trading environment in one spot. The intention isn't to overwhelm anyone—it's more about giving students a friendly entry point into managing their finances.

Table 1: Terms used in this document and their description

Name	Description
User Account System	Registration, login, verification, and profile updates with secure user data handling.
Budget Management	Students can log their spending and any income they receive. They can also sort expenses into categories and set personal goals. A few simple charts or summaries will help them see patterns without needing to interpret anything too technical.
Trading & Learning Module	This module keeps trading very small and easy to understand. The idea is to let students try out little PKR-based simulations—or tiny real trades—just to see how markets move. Nothing advanced or overwhelming.
Portfolio Tracking	Users can look back at their activity, see what they currently hold, and get a basic sense of how their choices are working out over time
Real-Time Market Data	Price changes and simple market information will come from external services like the Yahoo Finance API. The

	point is to help students connect their actions with real-world movements.
Educational Features	Short notes, tips, and explanations will appear throughout the app to help guide students who might be new to personal finance or investing.

2. Overview

THE OVERALL DESCRIPTION

Vaultex is a student-focused financial management and trading platform created by Pakistani students for Pakistani students. Many young users struggle with budgeting, saving, and investing because existing tools are either too complicated, dollar-based, or simply not designed for beginners.

In a single spot, Vaultex integrates all by providing safety of a wallet, easy trading features, and useful financial advice. The aim is to provide students with a hands-on method of managing their expenses, developing savings habits, and learning to invest all this in a simple and nurturing environment.

PRODUCT PERSPECTIVE

Vaultex functions as an independent application but connects with external services to enhance its features; these include:

- Market data APIs for real-time price updates
- Optional bank integration for PKR deposits and withdrawal
- Secure user authentication and data protection

Unlike platforms such as Binance or Mint, Vaultex is built specifically for local needs. It supports PKR, keeps the interface beginner-friendly, and combines budgeting with basic trading instead of focusing only on one area. The system emphasizes simplicity, accessibility, and learning, making it suitable for students who are completely new to financial tools.

PRODUCT FUNCTIONS

Vaultex offers six main functionalities, these are:

- **User Account Management**
  All users can securely sign up, log in, and be authenticated.

- **Budgeting Tools**
  Besides monitoring their incomes and expenses, the users will have the possibility to classify the expenditures and decide how much they want to save for personal goals.

- **Trading Module**
  Users will be able to purchase and sell assets in PKR following an easier trading process, which can be simulated or performed on a real small-scale transaction basis.

- **Portfolio and Transaction Tracking**
  Investors will be able to see how their assets are doing, what transactions they have made, and how their overall balance has changed over time.

- **Real-Time Data Integration**
  API connections will provide users with live updates of market prices so that they can make informed decisions.

- **Financial Insights**
  The app will offer various forms of support for users through user-friendliness, provision of basic analytics, and educational tips to assist them in building their financial habits.

USER CHARACTERISTICS

Vaultex is designed primarily for students, but different user groups interact with the system in different ways. This section describes each user class, along with their skills, needs, and expected behaviors.

1. Student User - Primary User

Description:

Students who use Vaultex to manage their money, track expenses, and experiment with small-scale PKR trading and investment.

Skill Level:

- Mostly non-technical
- Beginner level in financial literacy
- Basic smartphone/computer users

Responsibilities / Actions:

- Register and create a personal account
- Enter and categorize expenses
- View budget summaries and insights
- Perform simulated or small-scale trades
- Review portfolio history

Characteristics:

- May not understand advanced financial terms
- Needs simple UI and clear instructions

- Likely to make mistakes → requires confirmations and tooltips
- Uses the app frequently but for short durations

Limitations:
- Low financial knowledge
- Can only access their own data
- No access to system configuration or admin tools

2. System Administrator

Description:
Responsible for system maintenance, managing technical issues, and ensuring data integrity.

Skill Level:
- High technical knowledge
- Familiar with backend systems, servers, and databases
  Responsibilities / Actions:
- Manage user accounts
- Monitor logs and performance
- Handle API keys for market data
- Resolve data-related issues

Characteristics:
- Rarely interacts with front-end UI
- Works mostly through admin panel or backend tools

Limitations:
- Cannot perform trades or budgeting operations as a normal user
- Bound by security policies and audit logs

3. Financial/Content Editor (Optional Future Role)

Description:
A team member who updates the educational tips, financial learning modules, or stock recommendations.

Skill Level:
- Basic knowledge of finance
- Moderate computer skills

Responsibilities:
- Add/update educational content
- Improve onboarding tips
- Maintain quality of market explanation material

Limitations:

- No access to user data
- Cannot view or modify internal system settings

CONSTRAINTS

These are the limitations and restrictions under which Vaultex must function. They affect design, development, and operational behavior.

1. Technical Constraints

- The system must retrieve real-time price data from external API like Yahoo Finance.
- The accuracy of real-time market data depends on external API responsiveness and availability.
- Vaultex should function smoothly on devices with mid-range processing power without requiring many cores or heavy GPUs.
- The system must use secure encryption methods for passwords.
- The platform is web-based; device-specific mobile apps are not included in the current version.

2. Regulatory & Security Constraints

- User data must comply with basic data privacy laws applicable in Pakistan.
- No real currency trading beyond allowed small-scale learning transactions.
- The system must not store or process sensitive banking credentials directly; bank integration, if enabled, must use secure OAuth or third-party compliant methods.

3. Operational Constraints

- Internet is required for all trading operations, API calls, and dashboard updates.
- Real-time functions will not be operational offline.
- System updates may temporarily interrupt usage.

4. Design Constraints

- The UI must remain beginner-friendly and non-technical.
- Visual charts must not overload the user with complex data.
- All features must support PKR as the primary currency.
- External libraries used must be open-source or licensed properly.

5. Performance Constraints

- Dashboard loading must not exceed 5 seconds.
- API calls must complete within a reasonable timeout (e.g., 3–5 seconds).

- The system must handle up to 100 concurrent users without performance degradation.

ASSUMPTIONS AND DEPENDENCIES

ASSUMPTIONS:

These are conditions assumed to be true for the system to function properly. If any assumption fails, parts of the system may not work as expected.

1. User-Related Assumptions

- Students have access to a stable internet connection.
- Users can enter correct personal details during registration.
- Users understand the basic difference between "expense", "income", and "investment".
- Users are expected to have basic English literacy skills.
- Users will remember their login credentials or reset them when needed.
- Users will understand simple instructions like "Add Expense" or "Buy Stock".
- Users will use the system within normal hours and not overload it intentionally.
- Users have basic knowledge of how to navigate a desktop interface.

2. Technical Assumptions

- External APIs used for real-time market data (e.g. Yahoo Finance) will remain available and functional.
- Server hosting environment provides stable uptime.
- Database remains consistent and accessible during operations.

3. Operational Assumptions

- The system will be accessed primarily from Pakistan (PKR-based operations).
- Users will use supported browsers (Chrome, Edge, Firefox).
- The financial data provided by APIs is reasonably accurate.

DEPENDENCIES

These are external elements the system **depends on**, and failure or change in these components will affect functionality.

1. External API Dependencies

Vaultex depends heavily on:

- Yahoo Finance API
- Other PKR market data providers

If these APIs slow down or fail, real-time trading features will not function correctly.

2. Hosting and Cloud Dependencies

- The platform depends on the hosting service (AWS, Azure, or local cloud provider).
- Database uptime is critical—if the database goes down, no user can log in or trade.

3. Third-Party Libraries and Tools

- Charting libraries for graphs
- Authentication libraries
- Database engines (MySQL, PostgreSQL, etc.)

Any updates or failures in these libraries can break functionalities.

4. Optional Bank Integration Dependencies

If future updates include deposits/withdrawals:

- Requires secure connection with banking APIs
- Requires compliance with bank authorization standards

5. Development Environment Dependencies

- Programming languages, frameworks, and versions used
- Continuous integration tools
- Version control system (Git)

3. STATE OF THE ART

3.1 Literature Review:

3.1.1. Financial Literacy Gap in Pakistan:

Recent national data substantiate the need for localized financial management tools. The State Bank of Pakistan (2023) identifies, in its National Financial Inclusion Strategy, youth-focused initiatives as a matter of critical importance because a large portion of the young population remains unbanked or under-banked due to restricted access points to the formal financial system.

Despite these national imperatives, however, a gap seems to persist between theoretical knowledge of finance and its practical application. Khan and Rehman published a study on this in 2022, titled The Impact of Financial Literacy on Saving Behavior of University Level Students, the study found that students in Pakistan have a strong saving intention but lack accessible online facilities that minimize impediments to small savings investments.
The authors thus concluded that students with access to interactive financial tools have a much greater chance of fostering long-term saving behavior than students with purely manual, traditional approaches.

3.1.2 The effectiveness of Experiential Learning & Gamification

To alleviate these shortcomings, international studies recommend experiential learning over passive teaching. As Harter and Harter (2010) prove, students who participate in stock market simulations enjoy significantly higher financial literacy score improvements than their peers who obtain conventional classroom teaching only.

In addition, Smith and Gibbs discovered in 2020 that simulations represent an essential active learning model that provides students with practical experience and builds confidence in a no-risk environment before investing actual capital. This view is corroborated by Nugraheni et al. just 2 years later (2022), whose research into virtual trading platforms suggests that interactive elements within such platforms constitute gamification, which improves student engagement and promotes a better understanding of the most complicated issues in finance

3.1.3 The Vaultex Approach:

Vaultex operationalizes these insights by developing a hybrid educational ecosystem. By marrying the risk-free benefits of simulated trading-as Smith & Gibbs advocate for-with the pragmatic accessibility of real micro-trading and integrated instructional materials, the platform bridges the identified gap by Khan & Rehman. This ensures that students are not merely theoretically

enlightened but practically practice financial discipline within a controlled environment that is sensitive to the Pakistani economic context.

3.2. Analysis Of Existing Systems:
The current market for financial applications is dominated by offshore giants. While powerful and reliable, these systems often fail to address specific constraints of the Pakistani student demographic.

- **Complexity Barrier:** Platforms like **Binance** and **Kraken** offer advanced trading features that are designed for professional or dedicated traders. Their interfaces are cluttered with technical charts and jargon (e.g., "margin trading," "futures"), which overwhelms beginners and potentially alienates the next generation from these tools for grasping financial security and understanding.

- **Currency and Banking Constraints:** Budgeting apps like **Mint** are industry standards in the West but are nearly unusable in Pakistan because they only allow for integration with US or European banks to function, unfortunately offering no service here.

- **Fragmentation:** Currently, to trivialize financial stability via digital methods, a student must use one app for budgeting (e.g., a simple expense tracker) and a completely different app for trading, there is no unified ecosystem that connects "saving money" directly with "investing money". That is where the Vaultex Platform comes in.

Table: table of systems

Application Name	Key Features	Weaknesses	Relevance to Vaultex
Mint	• Budget planning • Bill reminders • Credit score tracking	• Requires integration with US banks • Overly complex for new users	The Vaultex Platform provides a simpler, localized interface that works independently of foreign bank integrations[4].
Binance	• Advanced crypto trading • Real-time charts • Portfolio tracking	• Too complex for beginners • Focused mainly on cryptocurrency	Vaultex introduces beginner-friendly trading in local currency (PKR) while combining it with budget tracking[5].

		• No budgeting or spending control	
Bitget	• Expense and income tracking • Saving goal setting	• No investment features • Lacks personalized financial insights	Vaultex merges budgeting and investment education, encouraging financial growth alongside savings[6].
Kraken	• Secure trading platform • Supports multiple cryptocurrencies	• Not designed for students • Complicated interface • No financial literacy tools	Vaultex focuses on student investors, offering simple interfaces and educational investment features in a single platform[7].

4. USER/SYSTEM REQUIREMENTS

4.1. External Interface Requirements

This section outlines how the Vaultex platform interacts with users, external hardware, and software systems.
The primary goal is to ensure the system is secure, easy to use, and compatible with standard student devices.

4.1.1. User Interfaces

- General Layout: The user interface shall be designed with a focus on simplicity and intuitiveness to cater to students with limited or no prior trading experience.
- Dashboard: The system shall feature a main dashboard that provides an immediate overview of the user's financial health, including Total Portfolio Value, Investment Returns, and a visual comparison of Budget vs. Spending.
- Visualizations: Financial data shall be represented using clear charts, graphs, and visual tools to facilitate easy understanding of asset performance and spending habits.
- Responsiveness: The interface shall be fully responsive, ensuring seamless usability across various screen sizes, including mobile devices and desktop browsers.

- Feedback: The system shall provide clear, immediate feedback and error messages to guide users during interactions, such as failed logins or invalid transaction attempts.

4.1.2. Hardware Interfaces

- Client Devices: The system shall be accessible on standard consumer hardware, including smartphones (Android/iOS) and personal computers (Laptops/Desktops) capable of running a modern web browser.
- Processing Power: The system shall function smoothly on devices with mid-range processing power without requiring specialized GPUs or heavy computational resources.
- Connectivity: The application relies on the device's standard network hardware (Wi-Fi or Mobile Data) to communicate with the server.

4.1.3. Software Interfaces

- Client Browser: The system shall be platform-independent on the client side, functioning within any modern web browser (e.g., Chrome, Safari, Edge).
- External Data Source: The system shall exchange with third-party data providers, specifically Yahoo Finance API, to fetch real-time sales price updates.
- Database System: The system shall interface with a secure relational database (e.g., MySQL or PostgreSQL) to store user profiles, budget logs, and transaction history.

4.1.4. Communication Interfaces

- Network Protocols: The system shall use only the standard HTTP/HTTPS protocols in communications between the client application and the server to guarantee information security.
- Data Exchange Format: Communication with external APIs (e.g., Yahoo Finance) shall be conducted via standard RESTful API calls, utilizing JSON format for data interchange.
- Security: Transmission of sensitive data, such as login credentials and financial transaction information, shall be encrypted using SSL/TLS protocols to prevent interception by unauthorized entities.

5. Functional Requirements

1. User Registration and Authentication

- The system shall permit new student users to register using a username, email, and password.
- The system shall allow authenticated users to access the platform using valid login credentials.
- The system shall verify the accuracy of all login information before granting access.
- The system shall provide a password reset option for users who forget their login details.

2. User Profile and Budget Setup

- The system shall allow users to create and update their profile information.
- The system shall allow users to enter their total budget or starting balance.
- The system shall securely store each user's budget information and make it accessible when required.

3. Stock Market Simulation

- The system shall display a list of available stocks along with updated simulated prices.
- The system shall allow users to buy and sell stocks based on their virtual wallet balance.
- The system shall immediately update the user's portfolio and wallet after every transaction.
- The system shall maintain a complete transaction history for each user.

4. Stock Recommendation System

- The system shall analyze stock performance data including price history and volatility.
- The system shall identify suitable buy or sell moments based on predicted trends.
- The system shall suggest stock options that align with the user's available budget.
- The system shall notify users when a stock reaches a favorable price point.

5. Budget Management Module

- The system shall categorize user spending into groups such as investments, savings, and personal expenses.
- The system shall determine how much of the user's budget can be safely allocated for trading.
- The system shall provide recommendations to help users manage and control their spending habits.

6. Data Management

- The system shall securely store all user information, stock data, and transaction records.
- The system shall allow users to access historical data related to their trading and budgeting activities.

7. Dashboard and Visualization

- The system shall provide a real-time dashboard displaying:
  - Total portfolio value
  - Investment amounts and returns
  - A visual comparison of budget vs. spending
- The system shall represent financial data using charts, graphs, or similar visual tools.

5.1. Functional Requirements with Traceability information

FR-1: User login

Requirement ID	FR-1		**Requirement Type**		Functional		**Use Case #**		1
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	N/A								
Description	The system shall allow users to register, login, and authenticate securely using their credentials.								
Rationale	To ensure only authorized users can access personal trading and budget data.								
Source	Project Team				**Source Document**		SRS		
Acceptance/Fit Criteria	Users can successfully register and login; invalid credentials must show error.								
Dependencies	Database module for storing credentials								
Priority	***Essential***	✓	***Conditional***	-	***Optional***	-			
Change History	Initial version created								

FR-2: User Profile

Requirement ID	FR-2		**Requirement Type**		Functional		**Use Case #**		UC-2
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	UC-1								
Description	The system shall allow users to setup their profile								
Rationale	To ensure only authorized users can access personal trading and budget data.								
Source	Project Team				**Source Document**		SRS		
Acceptance/Fit Criteria	To personalize the user experience.								
Dependencies	FR-1 (User registration)								
Priority	***Essential***	✓	***Conditional***	-	***Optional***	-			
Change History	Initial version created								

FR-3: Stock Market Simulation

Requirement ID	FR-3		**Requirement Type**		Functional		**Use Case #**		UC-3
Status	***New***	✓	***Agreed-***	-	***Baselined***	-	***Rejected***	-	

			to						
Parent Requirement #	N/A								
Description	The system shall display available stocks with simulated prices and allow users to buy or sell them using virtual money.								
Rationale	To help students practice trading without real money.								
Source	Project Team			**Source Document**		SRS			
Acceptance/Fit Criteria	Stock prices update periodically; users can buy/sell successfully.								
Dependencies	FR-1, FR-2								
Priority	***Essential***		✓	***Conditional***	-	***Optional***	-		
Change History	Initial version created								

FR-4: Transaction History Management

Requirement ID	FR-4		**Requirement Type**		Functional		**Use Case #**		UC-4
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	FR-3								
Description	The system shall store and display each user's trading history, including stock name, price, quantity and transaction date.								
Rationale	To help students practice trading without real money.								
Source	Project Team			**Source Document**		SRS			

Acceptance/Fit Criteria	To allow users to track their trading performance.						
Dependencies	FR-3						
Priority	***Essential***	↓	***Conditional***	-	***Optional***	-	
Change History	Initial version created						

FR-5: Stock Recommendation System

Requirement ID	FR-5		**Requirement Type**	Functional		**Use Case #**		UC-5
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	FR-3							
Description	The system shall analyze simulated market trends and provide buy/sell recommendations to users based on the user's budget.							
Rationale	To assist users in making better trading decisions and prevent losses.							
Source	Project Team			**Source Document**		SRS		
Acceptance/Fit Criteria	Recommendations must be generated based on defined stock trends and user budget.							
Dependencies	FR-3							
Priority	***Essential***	↓	***Conditional***	-	***Optional***	-		
Change History	Initial version created							

FR-6: Budget Management Module

Requirement ID	FR-6		**Requirement Type**	Functional		**Use Case #**		UC-6
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	N/A								
Description	The system shall calculate available funds, track expenses and suggest safe investment ranges.								
Rationale	To promote financial awareness and ensure responsible virtual trading.								
Source	Project Team				**Source Document**		SRS		
Acceptance/Fit Criteria	The system must update available budgets each time a transaction is made.								
Dependencies	N/A								
Priority	***Essential***	✓	***Conditional***	-	***Optional***	-			
Change History	Initial version created								

FR-7: Dashboard and Visualization

Requirement ID	FR-7		**Requirement Type**	Functional		**Use Case #**		UC-7
Status	***New***	✓	***Ag***	-	***Baselined***	-	***Rejected***	-	

			reed-to						
Parent Requirement #	N/A								
Description	The system shall display a dashboard showing portfolio value, budget summary and graphic visualization of data..								
Rationale	To help students understand their financial and trading performance.								
Source	Project Team			**Source Document**		SRS			
Acceptance/Fit Criteria	Dashboard loads correctly and data visualizations update in real time.								
Dependencies	FR-3, FR-4, FR-6								
Priority	***Essential***		↓ ***Conditional***	-	***Optional***	-			
Change History	Initial version created								

FR-8: Notifications and Alerts

Requirement ID	FR-8		**Requirement Type**		Functional		**Use Case #**		UC-8
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	FR-5								

Description	The system shall notify users of significant stock price changes and budget threshold alerts.						
Rationale	To keep users informed about important events without manual checking.						
Source	Project Team		**Source Document**		SRS		
Acceptance/Fit Criteria	Notifications are triggered correctly when set conditions are met.						
Dependencies	FR-5, FR-6						
Priority	***Essential***	-	***Conditional***	✓	***Optional***	-	
Change History	Initial version created						

FR-9: Data Management and Storage

Requirement ID	FR-9			**Requirement Type**	Functional		**Use Case #**		UC-9
Status	***New***	✓	***Agreed-to***	-	***Baselined***	-	***Rejected***	-	
Parent Requirement #	N/A								
Description	The system shall store all user, transaction and stock data, ensuring retrievability and consistency.								
Rationale	To maintain integrity and reliably of data								
Source	Project Team			**Source Document**		SRS			

Acceptance/Fit Criteria	Data remains consistent after restart; no data loss occurs.						
Dependencies	FR-1, FR-3						
Priority	***Essential***	✓	***Conditional***	-	***Optional***	-	
Change History	Initial version created						

6. Nonfunctional Requirements & Software System Attributes

1. Performance Requirements

- Under normal conditions, the system will load the main dashboard within five seconds.
- The system will be able to handle up to 100 chained users with no performance degradation at all.
- The system will finish the buying or selling transactions in five seconds.

2. Security Requirements

- In the first place, all user passwords will be encrypted in a secure manner before being stored.
- Only those users who are authenticated will be allowed to have access to their personal data.
- The system will not allow anyone to make unauthorized changes in stock or pricing data.

3. Usability Requirements

- The interface of the system will be simple, intuitive, and easy for the students who have no experience in trading to use.
- Clear feedback and error messages will be provided by the system during the user's interaction.
- The system will be compatible with both desktop and mobile browsers.

4. Reliability Requirements

- The system will continue to maintain data integrity during unexpected shutdowns.
- The system will guarantee consistency and accuracy in all the budget calculations and stock values.

5. Scalability Requirements

- The system will be designed in such a way that any new users and extra stocks can be added without extensive architectural changes.
- The system will also be capable of supporting the integration of real-world APIs in the future whenever needed.

6. Data Privacy Requirements

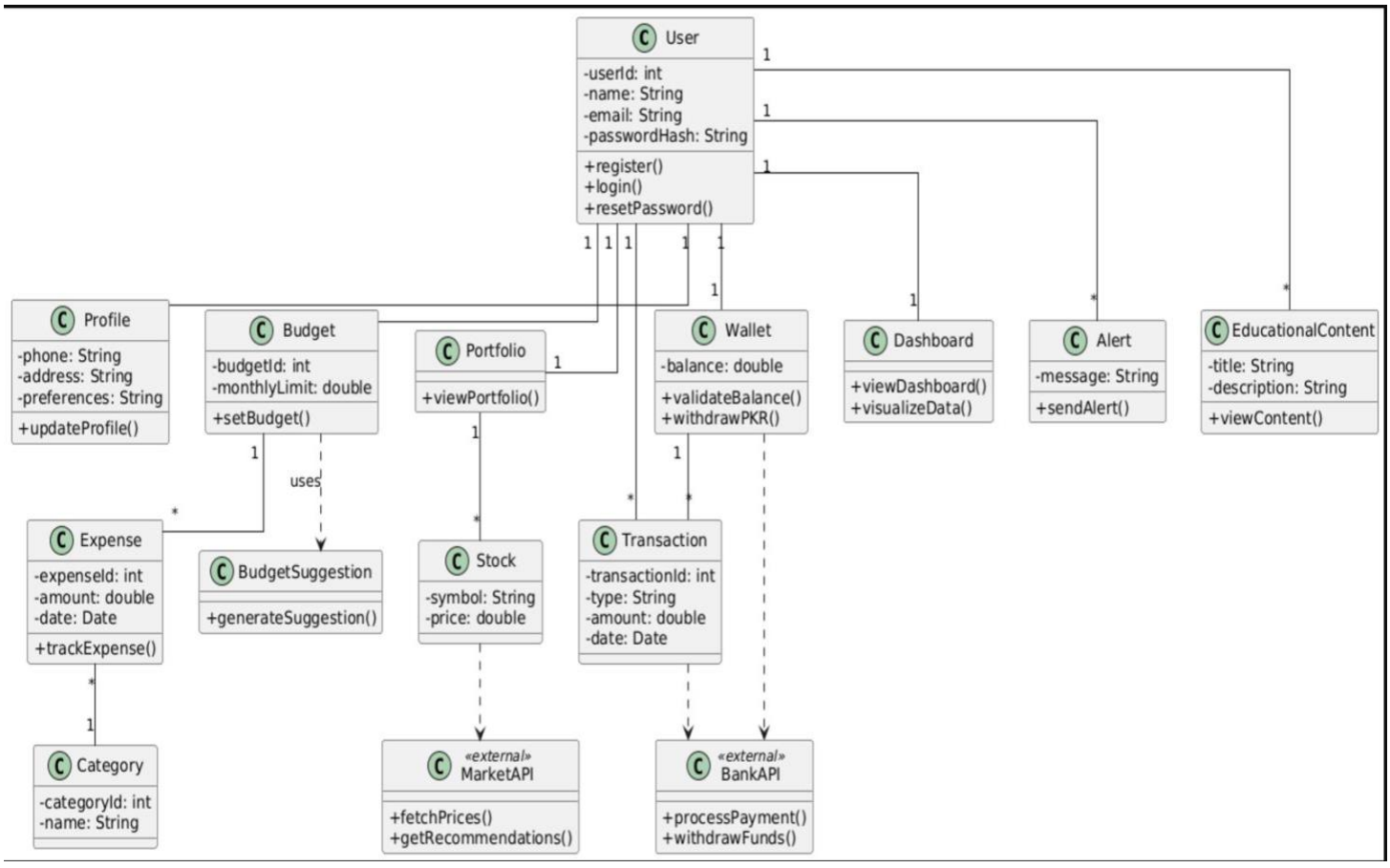
- The system will comply with the basic data protection principles for keeping user information private.
- There will be no sharing of personal or financial information with third-party companies without the user's consent.

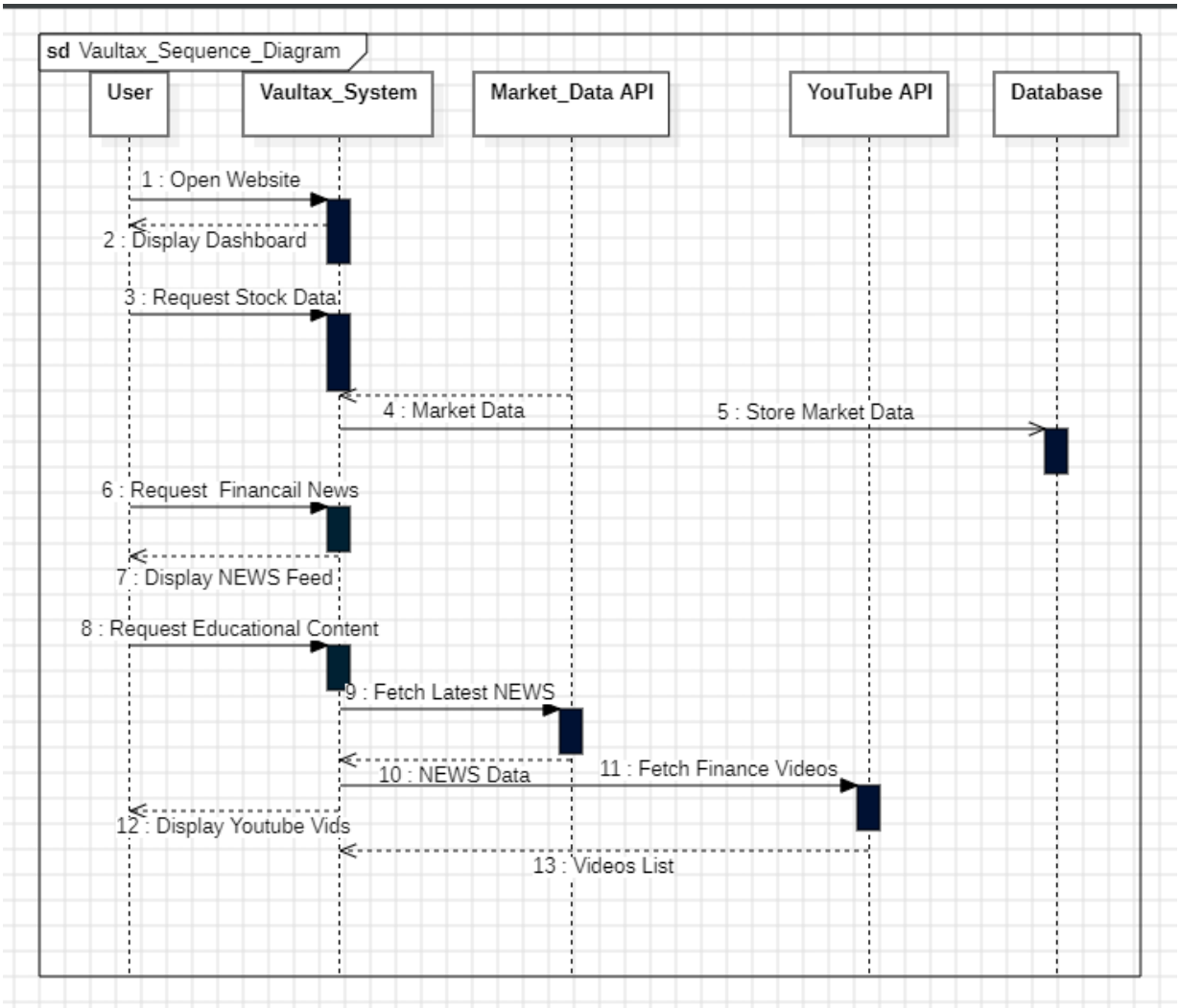
7. Project Design/Architecture

- 4+1 ARCHITECTURE VIEW MODEL

- ## Logical View

Class Diagram

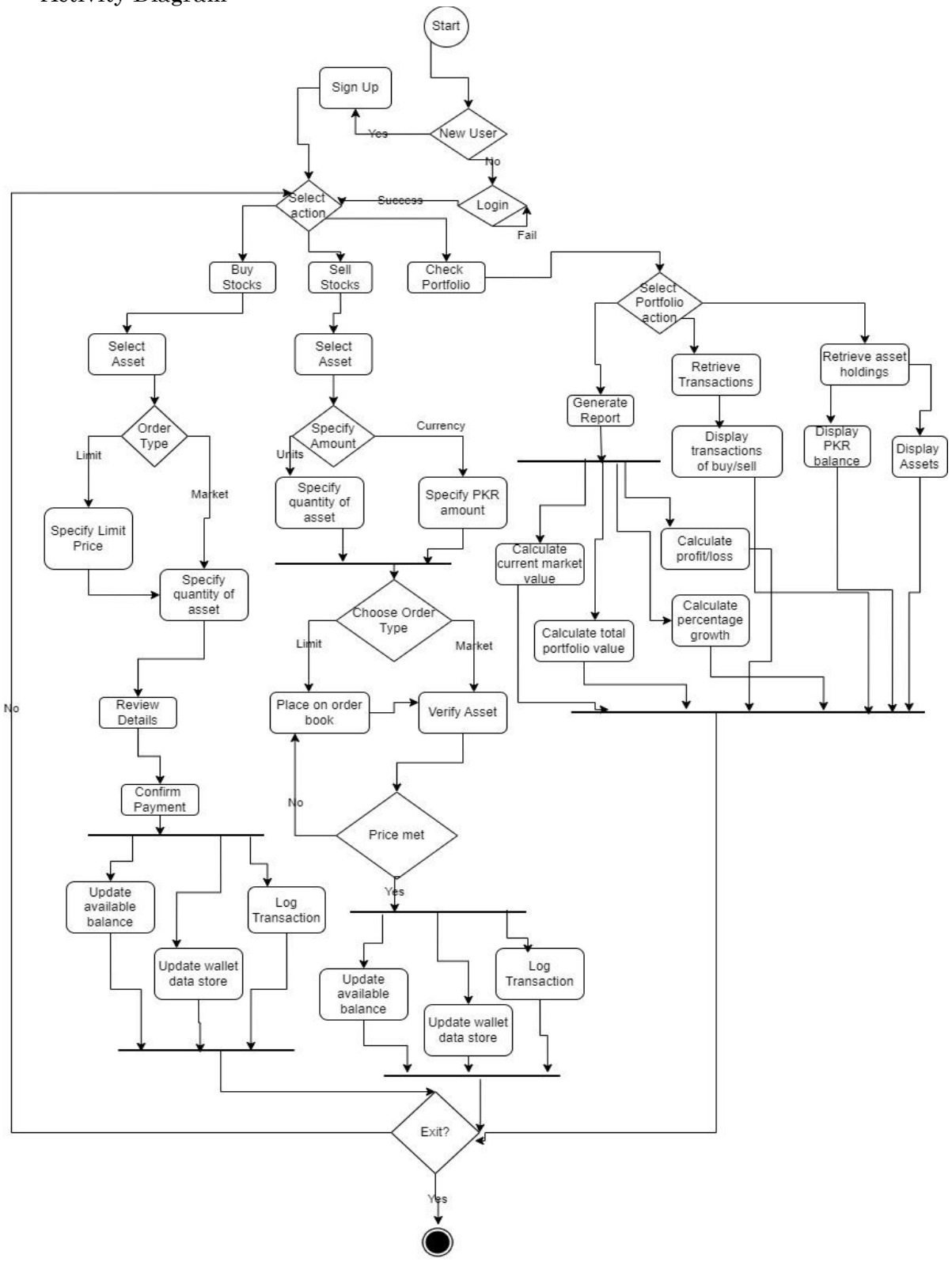




Sequence Diagram

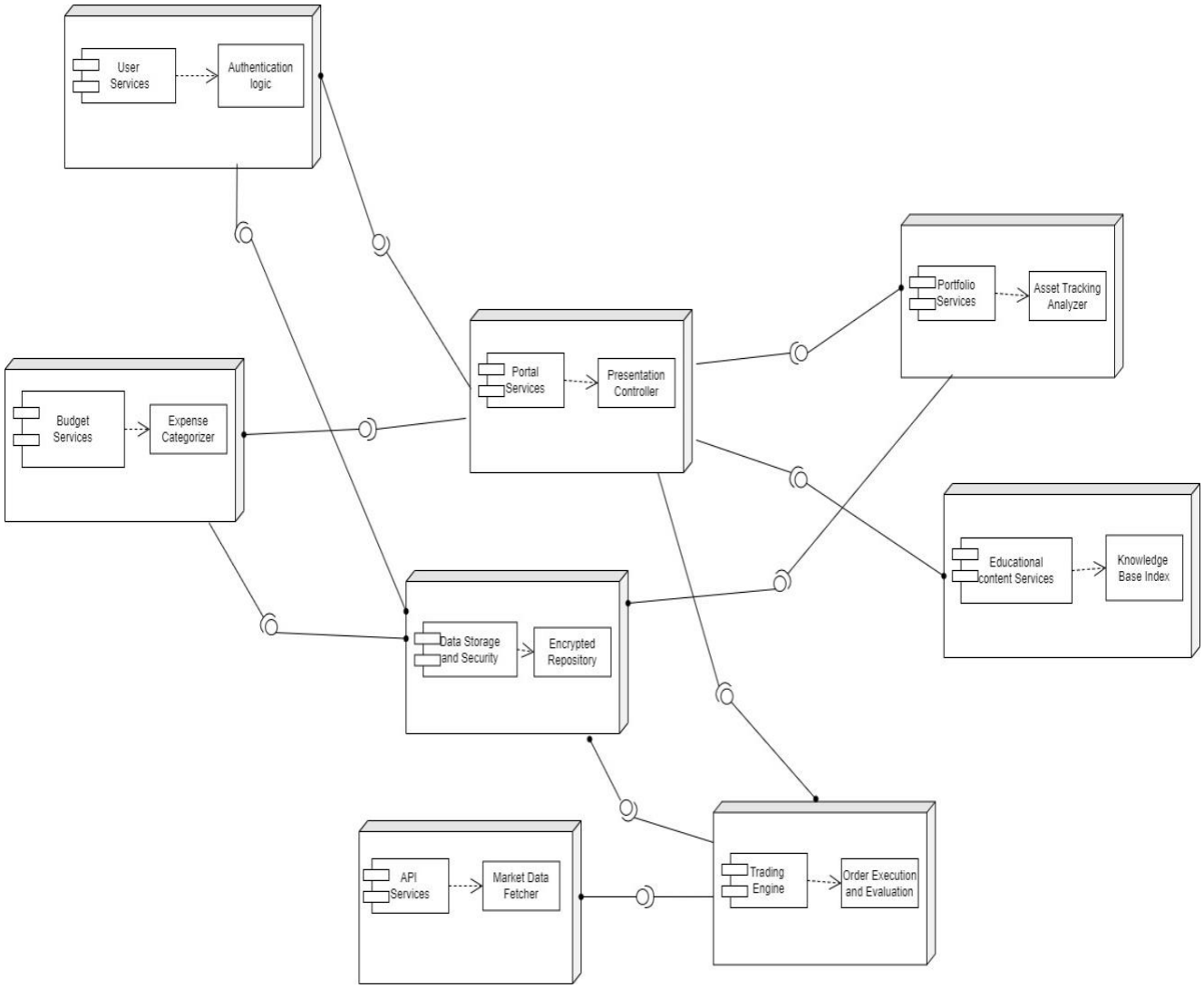
- **Process View:**

Activity Diagram



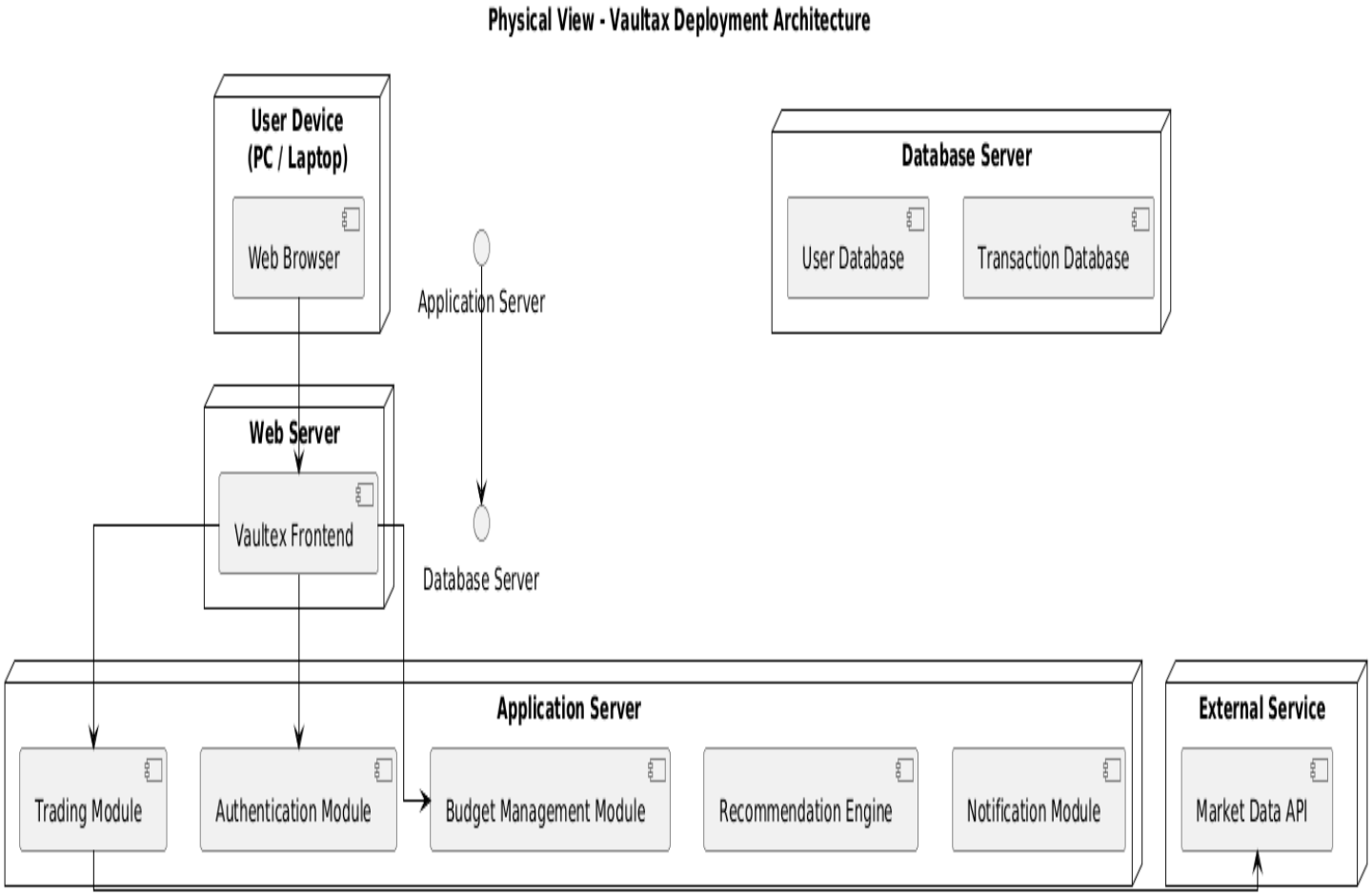
- **Development View:**

Component Diagram



- **Physical View:**

Development Diagram



- Use Case Diagram