Trade Net Trade & Portfolio Management Software:

Software Architecture Description

Kazizakia Sultana: ks2190
Hunter Lovvorn: hbl20
Chase Burns: cab589
Will Harrison: jwh335

CSE 4233/6233
Fall 2014

1. Architectural Documentation (Document Control Information)

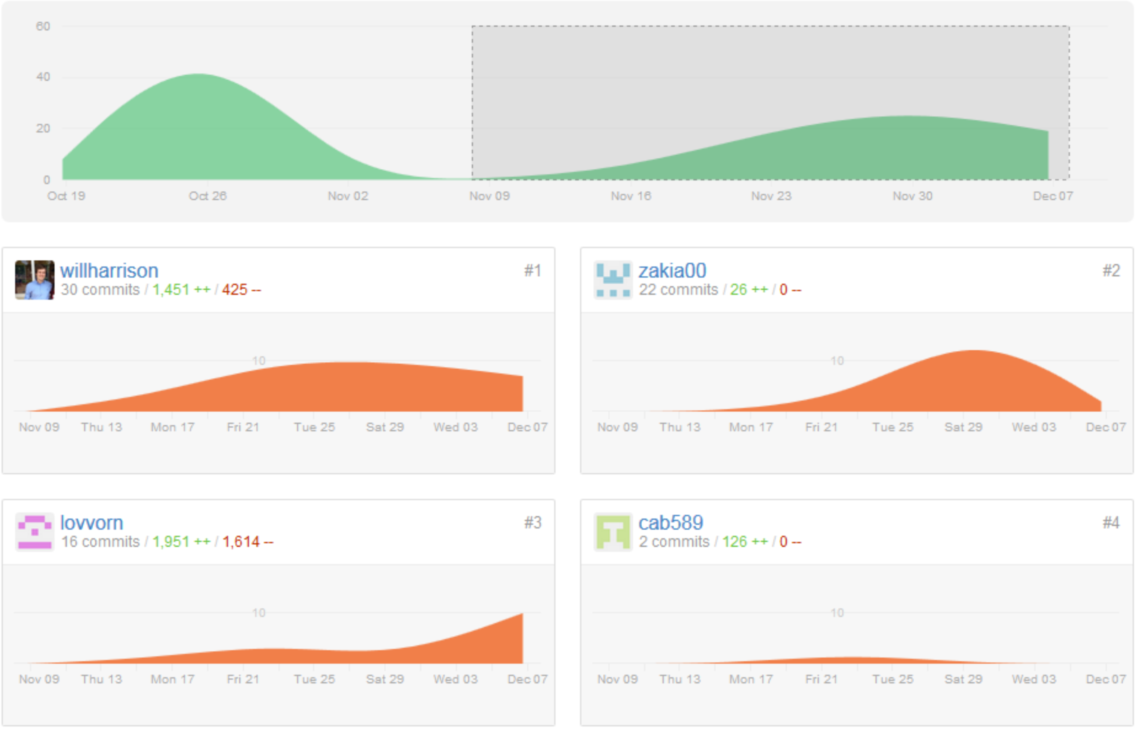
1.1 Date of issue and status

The document is generated for the final submission of the Architecture Description of the Trade Net Trade & Portfolio Management Software. The report is issued on the 10th December, 2014.

1.2 Issuing organization

The document is the final project report for the course Software Architecture and Design (CSE 4233/6233) instructed and supervised by Byron Williams in the department of Computer Science and Engineering of Mississippi State University.

1.3 Change history



1.4 Summary (System Overview)

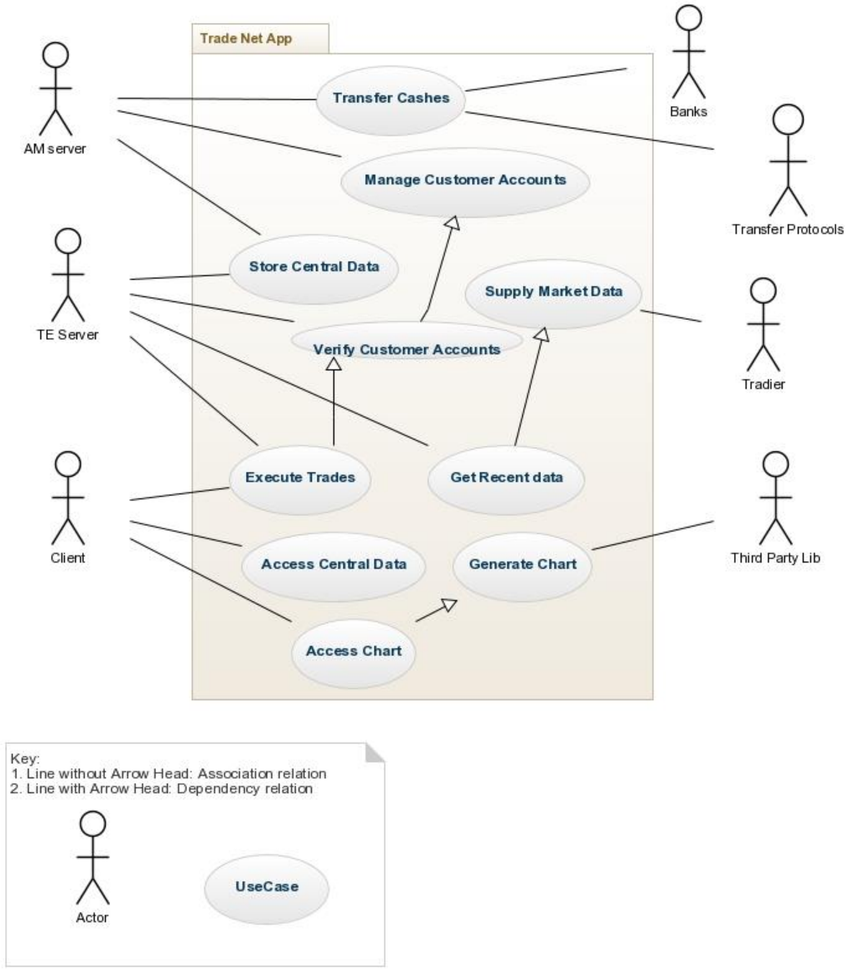


Figure 1: System Level Context Diagram

System level Context Diagram (Figure 1) shows all the actors (internal and external) and usecases of the system. It also presents the interactions of the actors with the usecases. Customers execute trades which depends on verifying customer accounts by the AM server. Customers access charts generated by the third party applications. TE server get recent market data supplied by Tradier.

1.5 Context

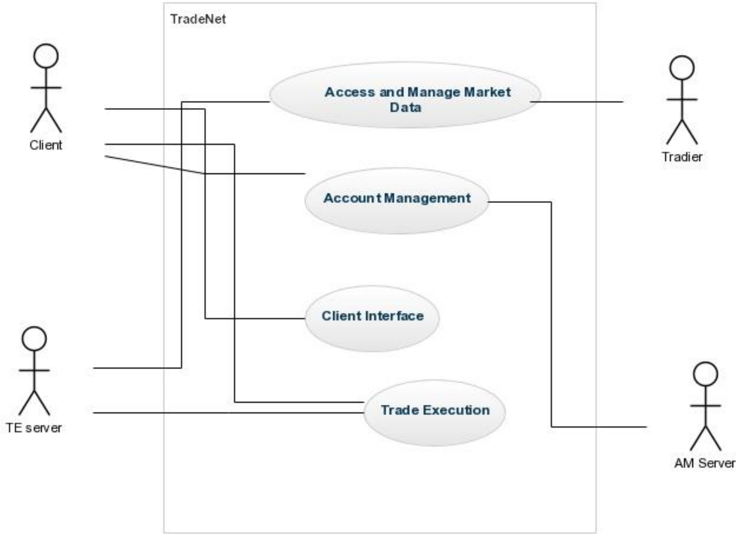


Figure 2: Context Diagram

Context Diagram (Figure 2) presents the abstract view of the system including major entities (Client, AM Server, TE Server and Tradier) as well as four major usecases. Client provides account information which are managed by AM server. TE Server executes trade after the verification of customer information. TE server accesses current market data provided by Tradier.

1.6 Table of Contents

1.7 References

- PHP Manual: http://php.net/manual/en/index.php
- jQuery API: http://api.jquery.com/
- Tradier Documentation: https://developer.tradier.com/documentation
- Bootstrap: http://getbootstrap.com/
- Requirements: SW Architecture Blog (myCourses)

1.8 Acknowledgements

We are really grateful to Dr. Byron J. Williams for providing educational background on Software Architecture and also for his support and continuous supervision in the development of the project.

2. Identification of Stakeholders and Concerns

2.1 Stakeholders

2.1.1 Project Manager

Project Manager cares about Schedule, resources, contingency plans, overall purpose and constraints, interaction with other systems, hardware environment. He will take care of the top level design and work assignment by visualizing the description.

2.1.2 Development Team

They will develop the system using the code resources and infrastructure provided to them for fulfilling the system requirements. As they need to know the requirements of the users and also schedule and budget of the system confined for the development along with their responsibilities, they need to care about architecture description.

2.1.3 Maintainers

Maintainers manage the system by getting information from this description. They will get ideas about any changes or impact on the system for managing them properly. They get information about the dependency across the modules and components.

2.1.4 Designer

Designers are concerned about the architectural structure and design of the whole system. They mostly care about the top level view and conceptual data model from the architecture description.

2.1.5 Customers

Customers will use the system for executing trade, viewing charts, and account management. As they are paying the application for their trade execution, they will care about the overall design of the application and concentrate if it can meet their requirements and interface with other systems properly.

2.2 Concerns

2.2.1 (Concern 1) What are the purpose(s) of the Trade Net?

Trade Net provides brokerage services (i.e., the ability to buy and sale shares of a company) to its customers. Trade Net customers pay for executing their trades and also for gathering updated information provided by the application. So the main purpose of the system is to enable proper and correct execution of trade with the supply of latest market data.

2.2.2 (Concern 2) What are the performance issues of the system?

As the trade execution depends on the current market price, the system should provide the requirement properly. The execution should be time and cost efficient as expected by the **analysts** and **customers**.

2.2.3 (Concern 3) What are the Security issues of the system?

The system maintain cash transfer using wireless protocols and banking interfaces. All kinds of communications and data transfers should be encrypted and secured such that attackers cannot break the system. Moreover, customer account information as managed by AM server should be kept private by maintaining proper access control. This is the primary concern of **customers**, **project manager** and **development team**.

2.2.4 (Concern 4) What are the other quality attributes of the system?

The software should also maintain Flexibility for added features and additional types of client interactions for future addition as needed by the **maintainers**. Availability is also important for avoiding denial of service attacks from any malicious users.

3. Architectural views

3.1 Logical views

3.1.1 Uses View

a. Primary Presentation

Uses view presents “depends on” relationship among the modules. Here we see that TE server depends on AM server for authenticating customer information. AM and TE server access data in the central database as well. Client App depends on TE server for executing trades and also on third party for financial analysis as well as stock charting.

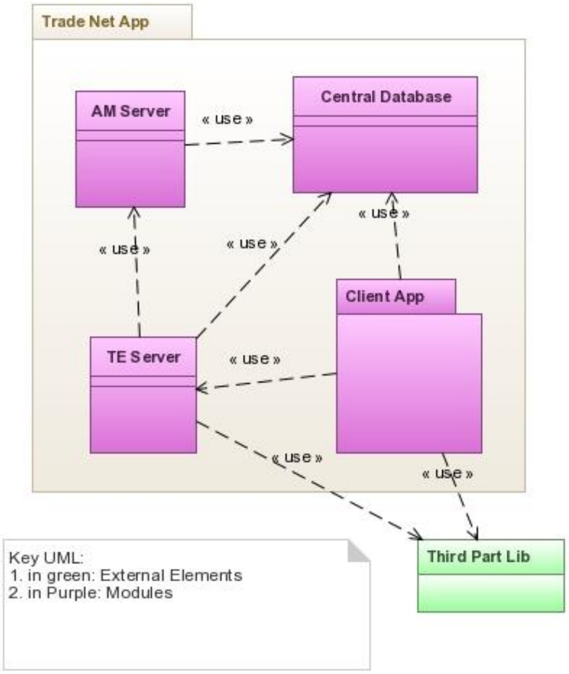


Figure 3: Uses Style (Model View)

b. Element Catalog

Elements	**Modules:** AM Server: - Access Customer account database - Process and store customer portfolio information - Customer reporting data (monthly report P&L, etc.) TE Server: - Access Customer information and legacy data from Account Management Server - Communicate with Tradier Network for Market Data - Access market data for any listed security (equities only) - Execute trades using prices for the listed securities Client Application, Central Database Client App:

	- Stock charting (including technical analysis modules) - Company fundamental data access - Trade Execution - Account Management - Portfolio Information - List Transaction History
Properties of Elements	- AM and Central database are DB servers - TE is the server-side trade execution server - Client Application is the client side application for running trades
Relations	Uses or Depends on: - TE server uses the information from AM server - Both AM and TE servers use Central database - Client Application uses Central database and third party libraries
Properties of Relations	- TE server depends on AM server for customer data verification and authentication - Both AM and TE servers store data on Central database - TE server depends on Tradier Network for market data
Interface	- AM Server with service interface (e.g., API, SOAP, REST, SOCKETS etc.) - web, Mac, PC, iOS, and Android clients interfaces for Client Application
Behavior	- No broad fan out - No loop - No long dependency

1. Architectural Background
   - **Rationale:**
     - Uses view is used for documenting the uses or depend relationships among the modules. As there is a number of dependencies among the AM, TE and central servers as well as servers with client

    application, uses view is helpful to document the relationships with clarity.
  - It clearly describes the relationships among the modules to facilitate the development team, analysts and customers.

- **Analysis Results:**
  - As TE server depends on AM server for customer authentication, AM server should work correctly for better performance of TE server. If AM server does not maintain data integrity and security it will hamper overall system security and customer privacy.
  - Moreover, the whole system depends on a number of third parties (tradier, Banking, Wireless protocol and so on) for executing all the applications. So the interfaces or APIs for communicating with the third parties should be protected and secured such that no data is leaked to the unauthorized users.
- **Assumptions:**
  - Differing development platforms will not conflict with the intended functionality of the application

3.1.2 Data Model View

a. Primary Presentation

Data Model view describes the system in terms of objects and their relationships through interface. In the view, we see various associations of the classes with one another. Each customer maintains many accounts whereas can execute zero to many trades or transactions. Moreover, each customer can communicate with zero or one server. Each server can be related with zero to many third parties for serving the external requirements to the customers.

Keys:

1. Boxes: Entities
2. 1___________ * One to many relation

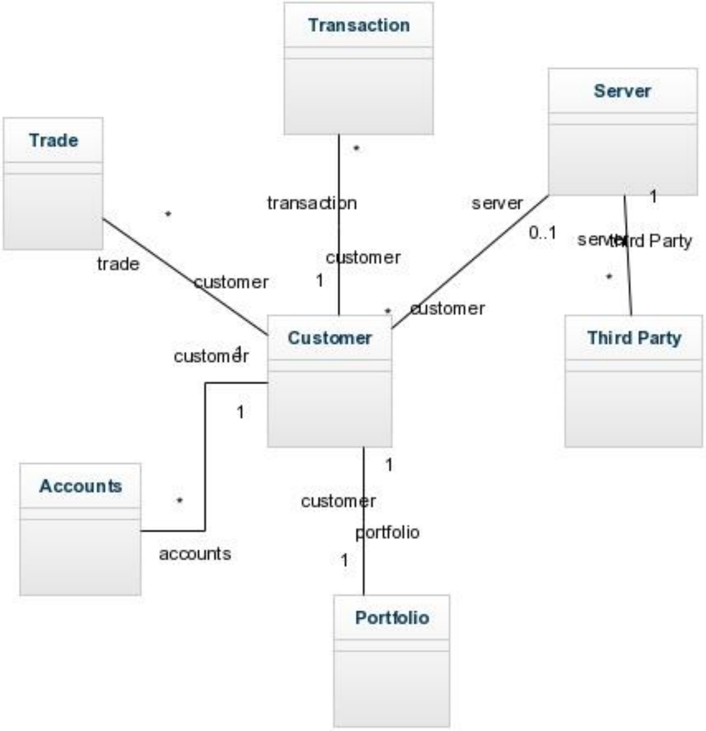


Figure 4: Data Model Style (Module View)

b. Element Catalog

Elements	**Entities:** Trade: - Trade Execution Transaction: - Cash transfer - Buy/Sell Server: - Run Application - Store data - Manage data - Integrate all the modules Portfolio: - Stock information Customer: - Execute trade - Data access - Manage accounts Accounts: - Store customer data - Verify data Third Party: - Supply current market data - Banking activities

Properties of Elements	- Customer (ID, first name, address, account balance, current profit and loss) - Transactions (ID, stock, #shares bought / sold [positive / negative], date and time) - Portfolio (CustomerID, stocks, shares, purchase price)
Relations	- One to one - Many to one
Properties of Relations	- Each customer has one or more accounts - Each customer has one portfolio - One customer can execute many trades - One customer can do zero or more transactions - Each server can be associated with many third parties.
Interface	- Portfolio has the customer ID as the foreign key - Entities communicate with each other through keys
Behavior	- Functional Dependency - Normalization

c. Architectural Background

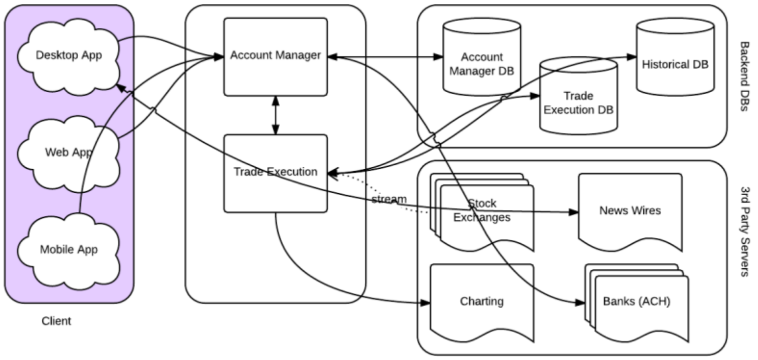
- **Rationale:**
  - Data model view facilitates stakeholder communication during domain analysis and requirements elicitation.
  - It is the Blueprint for implementation of data entities.
- **Analysis Results:**
  - Query Optimization and indexing increase the performance of the system.
  - Modifications must be reflected in the associated entities and dependencies need to be checked.
- **Assumptions:**
  - Stakeholders can interpret the document
  - Engineers create the system in the way planned

3.2 Run-time Structure and Dynamic Views

3.2.1 Client-Server Style

a. Primary Presentation

Client Server Style presents client tier with three platforms such as Desktop App, Web App and Mobile App. They communicate with the AM and TE server for authentication of their identities as well as for executing trades. TE server fulfills the requirements through different third party interfaces.



Client-Server

b. Element Catalog

Elements	Components: AM Server: - Customer account management TE Server: - Trade Execution Client Application (Desktop Application, Web Application, Mobile Application): - Send Request to the server for all the functions listed in module view
Properties of Elements	- AM Server is the DB server that manages customer information - TE is the server-side trade execution server - Client Application is the client side application for running trades. It has applications for

	different platforms such as Desktop or Mobile or Web
Relations	Attachment: - TE server is attached with AM server - Client Applications are attached with the TE server - TE server is attached with third party servers
Properties of Relations	- Client requests for trade execution to the TE server - Client requests are verified by AM server - TE server and Client Applications send requests to the third party servers
Interface	- AM Server with service interface (e.g., API, SOAP, REST, SOCKETS etc.) - web, Mac, PC, iOS, and Android clients interfaces for Client Application
Behavior	- TE server acts as client to the AM server as it sends customer verification requests to AM server - Servers send reply to the clients through connectors

c. Architectural Background

- **Rationale:**
    - Client Server view is used for grouping the common set of functionalities to better understand the system and decouple client applications from the server side.
    - Client and Servers are assigned to different tiers resulting in better performance, scalability, modifiability, reusability and reliability.

- **Analysis Results:**
    - The response from TE server after the client requests should be prompt and reliable.
    - If AM server does not maintain data integrity and security it will hamper overall system security and customer privacy.

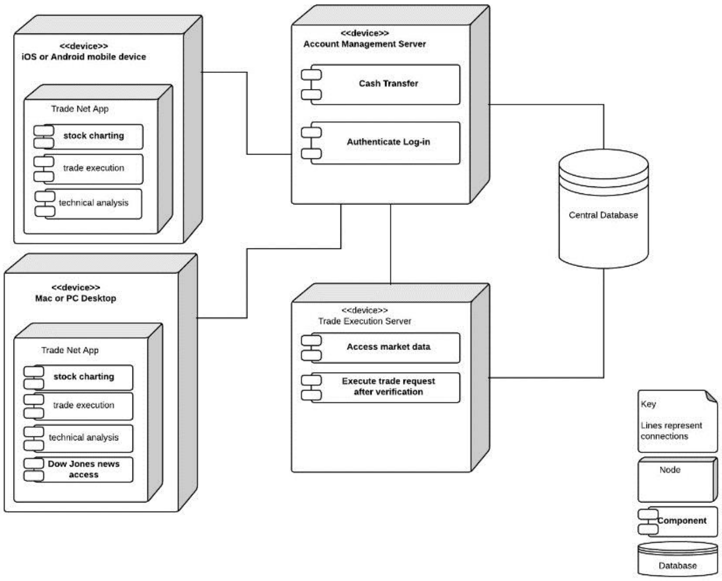
    - The volume and rate of client requests should be handled without delay and any interruption.
    - The interfaces or APIs for communicating with the third parties should be protected and secured such that no data is leaked to the unauthorized users.
- **Assumptions:**
    - 3rd Party services continue functioning
    - Server load does not exceed API restrictions

3.3 Allocation Views

3.3.1 Deployment Style

a. Primary Presentation

Development Style presents the view how the services are allocated to different devices of the system. As for example, desktop or mobile devices are allocated to run the client application that exhibits stock charting, financial analysis as well as trade execution interface. On the other hand, AM server is assigned for cash transfer and login authentication services. Trade Execution server executes server application for accessing market data as well as performing trades.



b. Element Catalog

Elements	**Components:** Mac or PC Desktop, IOS or Android Mobile device: - Run client applications AM server: - Manage customer data TE server: - Execute trade Central DB: - Archive log - Manage other servers
Properties of Elements	- Stock charting, Trade execution, Technical analysis, News access are performed by client devices (desktop/mobile). - AM server and TE server consumes spaces in the Central DB server. - Resource requirements and fault tolerance of the devices are maintained.
Relations	Attachment: - TE server is attached with AM server

	- Client Applications are attached with the TE server - TE server and AM servers are attached with central DB
Properties of Relations	- Client devices communicate with TE server for trade execution and charting and technical analysis. - Client requests are verified by AM server. - AM and TE server are attached with central DB for managing and storing data. - TE server and Client Applications send requests to the third party servers.
Interface	- web, Mac, PC, iOS, and Android clients interfaces for device to server communication
Behavior	- Different client application versions for different device platforms

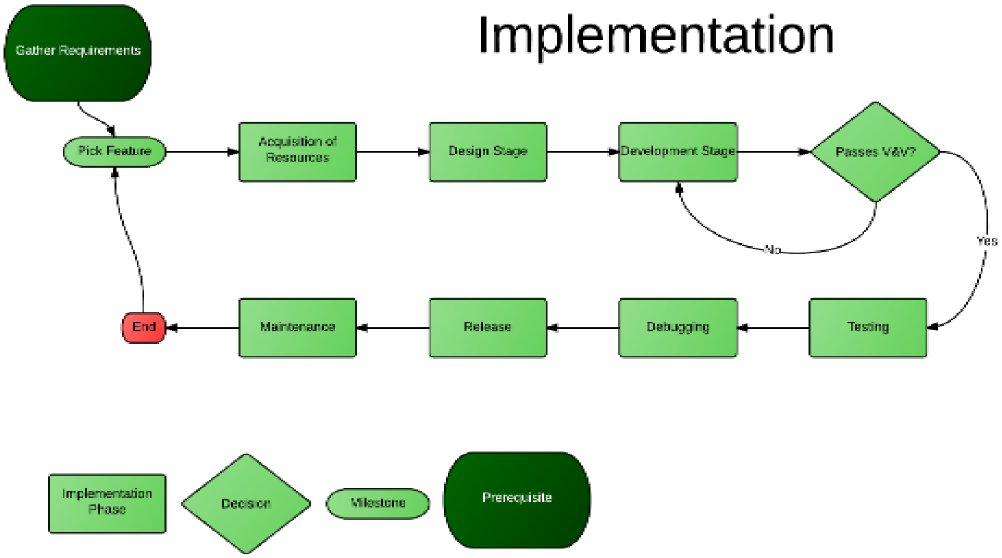
c. Architectural Background

- **Rationale:**
  - Deployment view is used for cost estimation of the infrastructure where the system will be deployed. It shows the requirements for devices and platforms for implementing the system.
  - Project Manager, Development team, Analysts, Customers and Maintainers will require the view for doing their job.
- **Analysis Results:**
  - Security for different devices and platforms need to be analyzed properly.
  - The reliability of the servers and client devices can be increased by maintaining proper backup of data and increasing fault tolerance of the devices and communication channels.
  - Performance issue need to be taken care by increasing the bandwidth of the system.
- **Assumptions:**
  - Technologies used do not require high integration costs

3.3.2 Implementation Style

a. Primary Presentation

Implementation Style presents the architecture of how the modules or components are assigned to the configuration items for implementation purpose.



b. Element Catalog

Elements	- Modules and Components of the Module and component views - Configuration items
Properties of Elements	- Mobile or Desktop File systems - Module dependency - Version of compilers and interpreters
Relations	Attachment: - Allocated to: modules or components to the source codes - Containment: Directory structure and version control
Properties of Relations	- Components should be allocated to proper configuration code depending on the device file system.

Interface	- Components should be interfaced to each other by proper message passing.
Behavior	- Version differences should be specified. - Manage and maintain files by following proper structure during development and build of the system.

c. Architectural Background

- **Rationale:**
  - Implementation view is used by the developers during development of the system.
  - It helps them to get a view on resource allocation and assignments to them for building the system.
  - Software testing and updating can also be made easier by analyzing this view.
- **Analysis Results:**
  - Source codes should be properly organized and structured.
  - SQL queries need to be efficient and optimized.
- **Assumptions:**
  - Requirements can be broken into parts that can be worked on in regular intervals
  - Major obstacles, such as major security flaws, do not interrupt the schedule

4. Consistency among Architectural Views (Mapping Between Views)

To provide for consistency, the same terms, ideas, services, and structure was used throughout our documentation as well as our diagrams. We strived to make the different views show as much of the architecture while keeping it simple and regular. There is a high level of consistency throughout our documentation.

5. Architectural Rationale

Description	Associated Entities	Specifications
The ability to view live market data	3rd Party Stock Integration	1. Must be live data 2. Stream cannot die

		3. Tradier developer account must be maintained
The ability to add funds to your account	Bank Integration (ACH Payment)	1. Must be in good standing with banks. 2. Provide high application security 3. Redundant logging 4. Screen for fraudulent activity
The ability to purchase stock	3rd Party Stock, Tradier, TE Server, AM Server	1. AM Server must be accessible 2. TE Server must be accessible 3. Tradier & 3rd Party Stock link must be synchronous
The ability to view portfolio	Portfolio, Tradier, TE Server, Historical Data	1. User must be authenticated 2. User must have trading history 3. Historical data is not necessary, but can provide gain/loss information
The ability to sell stock	Tradier, Portfolio, AM Server, TE Server	1. Tradier & 3rd Party stock link must be present 2. User must have the proper stock, class, and amount to sell
The ability to view news	News Wire, Portfolio	1. Non-critical requirement—must have proper API links to news outlets 2. Use portfolio trading data to create news recommendations based on trade patterns.
The ability to view charting data	Portfolio, Tradier, Charting, TE Server, Historical Data	1. Data must be securely and anonymously funneled to charting API 2. TE Server & Historical data can be used to show patterns with stock owned by user 3. Tradier link can be used to show patterns in the market
The ability to view historical data	Historical Data	1. A database must exist to track historical market data

		2. Data should not be loaded from API. Should by stored by our application
The ability to manage an account	Account Manager, AM Server	1. User must be authenticated 2. The AM Server must be accessible
The ability to authenticate (and log out)	AM Server, Client App	1. Client App must have internet connection 2. Client App must pass security requirements 3. Client must use proper credentials 4. 2-Factor authentication is required on all accounts