# Sample Project Specification

PMKing Trading LLC, 10b Seymour, Middlebury, VT 05753

public@pmkingtrading.com

802 349 9501

Version 0.1 10/11/5

PMKING TRADING

# Table of Contents

# Table of Figures

# 1 Notation Used

The notation used in this specification will be based on the Unified Modeling Language (UML)

## 1.1 Actor

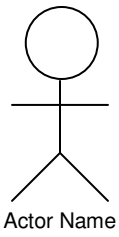


**Figure 1: Actor Notation**

An Actor is a person or external system that participates in, or controls, the business domain being described.

## 1.2 Use Case

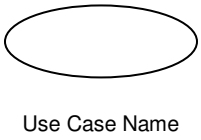


**Figure 2: Use Case Notation**

A Use Case is a logical collection of related functionality that describes a specific event in the business domain being described.

## 1.3 Actor/Use Case Interaction

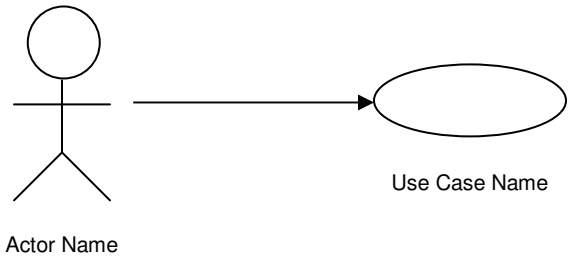


**Figure 3: Actor/Use Case Interaction Notation**

Actors can initiate, or take part in Use Cases. The arrow indicates the main direction of information flow in the interaction.

## 2 Requirements

### 2.1 Need

### 2.2 Cost/Benefits Analysis

## 3 Functional Specification

### 3.1 Overview

### 3.2 Actors

#### 3.2.1 Human

3.2.1.1 Trader

3.2.1.2 Client

### 3.3 External Data Sources

3.3.1.1 Tick Quote Source

3.3.1.2 Historical Quote Source

3.3.1.3 Intra-Day Bar Quote Source

3.3.1.4 Tradable Equities Source

3.3.1.5 Shortable Equities

3.3.1.6 VWAP Equities

#### 3.3.2 Corporate Actions Source

3.3.2.1 News

3.3.2.2 Splits

3.3.2.3 Buy Backs

#### 3.3.3 Order Processing

3.3.3.1 Primary Order Destination

3.3.3.2 Secondary Order Destination

3.3.3.3 Portfolio Datastore

#### 3.3.4 Trading System Management

3.3.4.1 Symbology Datastore

3.3.4.2 Open Trades Datastore

3.3.4.3 Closed Trades Datastore

### 3.3.5 Other Actors

3.3.5.1 Broker

# 4 Use Cases

### 4.1.1 Get List of Tradable Equities

### 4.1.2 Test if Equity is Tradable

### 4.1.3 Validate Symbol

### 4.1.4 Validate All Symbols

### 4.1.5 Refresh Tradable Equities

### 4.1.6 Check for Opening Gap Trades

### 4.1.7 Check for High/Low Trades

### 4.1.8 Enter New Position

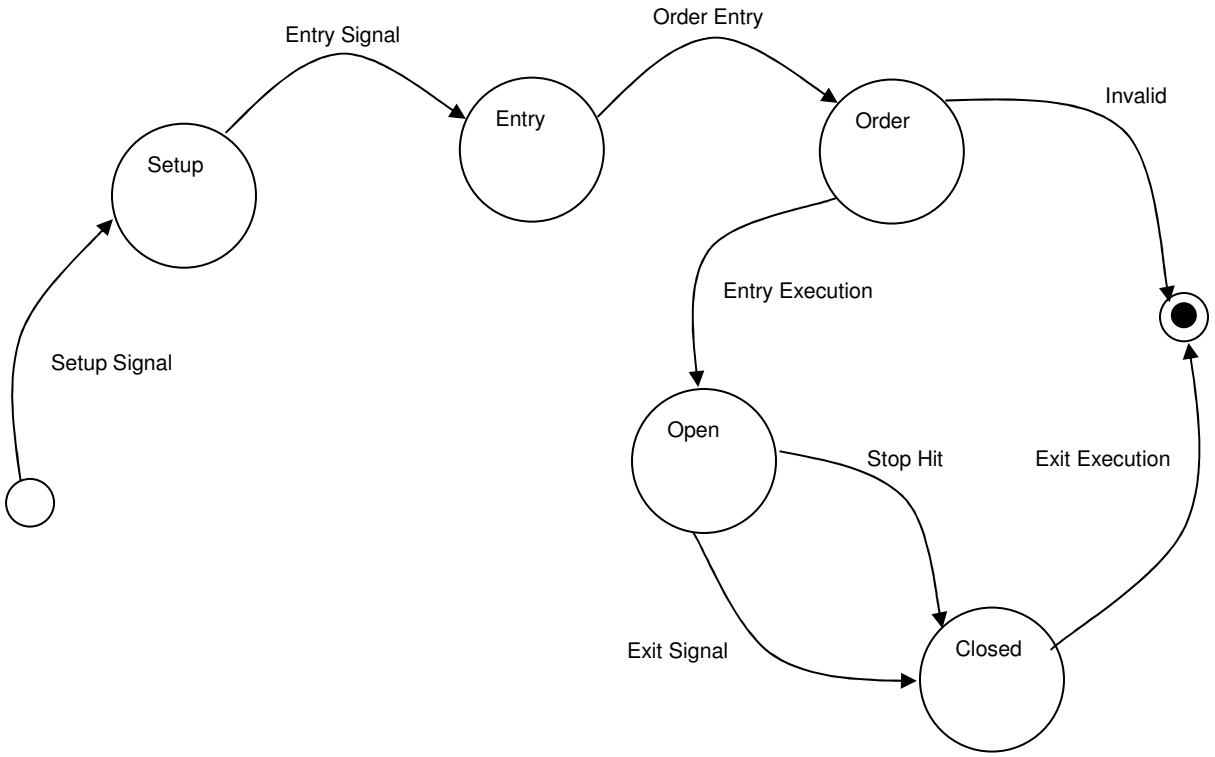


**Figure 4: State Transitions of a Position**

#### 4.1.8.1 Enter New Gaps Position

#### 4.1.8.2 Enter New Splits Position

4.1.8.3 Enter New BuyBacks Position

4.1.8.4 Enter New VWAP Position

4.1.8.5 Enter New ModeX Position

**4.1.9 Update Stops**

4.1.9.1 Update Gaps Stops

4.1.9.2 Update Splits Stops

4.1.9.3 Update BuyBacks Stops

4.1.9.4 Update ModeX Stops

**4.1.10 Exit Positions**

4.1.10.1 Stop Was Hit

4.1.10.2 VWAP Position Priced

4.1.10.3 Exit ModeX Position at Close

4.1.10.4 Exit All Positions

**4.1.11 Calculate Net Liquidation Value of Portfolio**

**4.1.12 Enter Pre-Open ModeX Orders**

**4.1.13 Display Portfolio**

**4.1.14 Update Last Tick Quote**

**4.1.15 Backfill Intra-Day Quotes**

**4.1.16 Backfill Historical Quotes**

**4.1.17 Update Event Schedule**

4.1.17.1 Add Scheduled Event

4.1.17.2 Modify Scheduled Event

4.1.17.3 Remove Scheduled Event

**4.1.18 Process Scheduled Events**

**4.1.19 Report Position Entry or Exit**

**4.1.20 Set Alert**

**4.1.21 Trigger Alert**

## 5 Full Actor/Use Case Diagram

<shows how all the actors and use-cases interact>

# 6 Logical Data Model

## 6.1 Notation Used

The notation used for the persist entities in the system will be based on the Unified Modeling Language (UML)

### 6.1.1 Entity

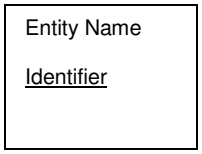


**Figure 5: Entity Notation**

An Entity is a related group of information that persists over time in the business domain being specified. It has an identifier that uniquely describes one instance of the entity.

### 6.1.2 Entity Relationship

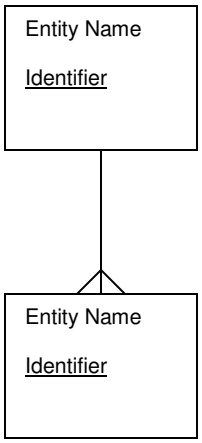


**Figure 6: Entity Relationship Notation**

An entity may have a relationship with other entities. The multiplicity of the relationship (one to many, many to one, or many to many) is represented by the 'crow's foot' connecting the 2 entities.

## 7 Full Entity Relationship Model

<shows all the logical data model components and relationships>

# 8 User Interface

## 8.1 Requests

## 8.2 Responses

# 9 Technical Specification

## 9.1 Technical Architecture Overview

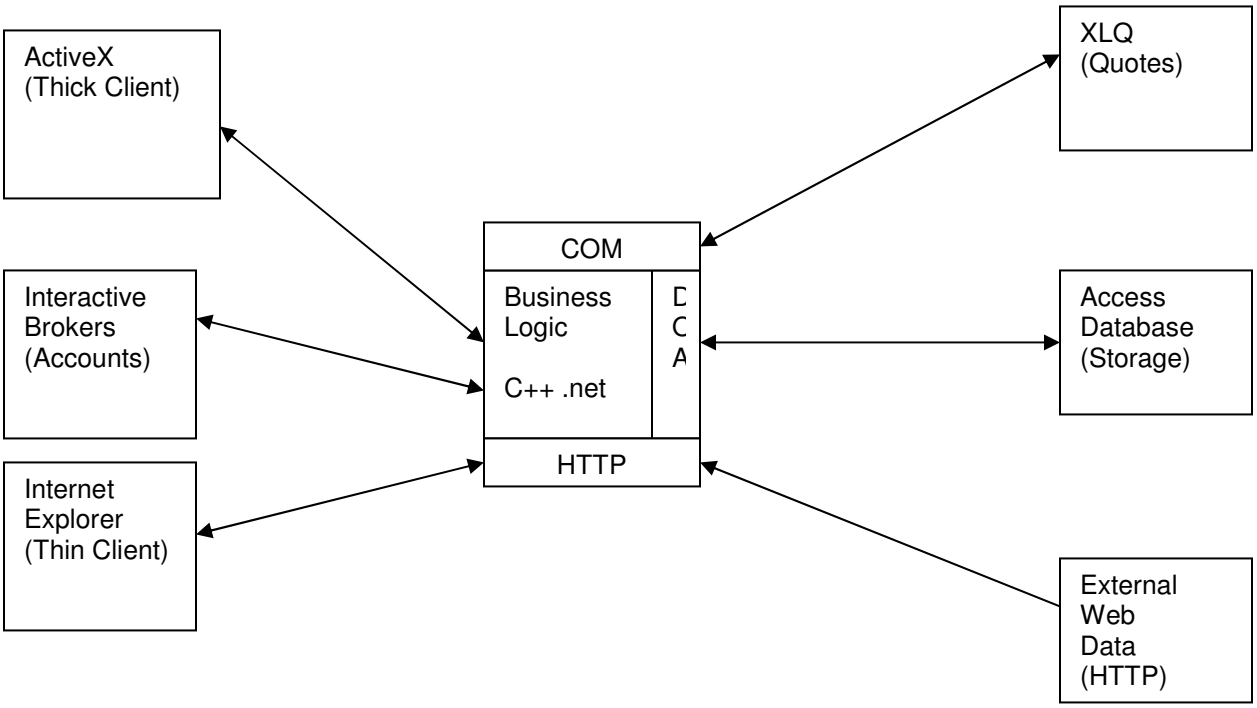


**Figure 7: Technical Architecture Overview**

### 9.2 User Interface

### 9.2.1 Business Logic

### 9.2.2 Persistent Storage

### 9.2.3 External Data

### 9.3 Other Requirements

### 9.3.1 Speed

### 9.3.2 Quote and Symbol Capacity

### 9.3.3 Remote Access

### 9.3.4 Redundancy and Reliability

## 10 Glossary