***Algorithmic Trading***

***System Requirements Specification***

Okanagan College

Algorithmic Trading System

2024-04-09

Version 1.6

Authors: Jacob Rawlings,Jake Fischer**,** Alan Abdollahzadeh, Devon Volberg, William Blake, Benjamin Carrier, Vanessa Dubouzet, Dakota Flath, Parker Green, Yuan Hu, Jacob Labelle, Isaac Lengacher-Bergeron , Dominic Presch, Dylan Soares, Jaeden Soukoroff

**TABLE CONTENTS**

[**1.0 Introduction 5**](#_heading=h.30j0zll)

[**1.1 System Purpose 5**](#_heading=)

[**1.2 System Scope 5**](#_heading=)

[**1.3 Goals and Objectives 5**](#_heading=h.z337ya)

[**1.4 Definitions, Acronyms, and Abbreviations 5**](#_heading=)

[**1.4.1 Key Definitions 5**](#_heading=)

[**1.4.2 Key Acronyms and Abbreviations 6**](#_heading=)

[**1.5 Key References 7**](#_heading=)

[**2.0 General System Description 7**](#_heading=)

[**2.1 System Context 7**](#_heading=)

[**2.2 System Modes and States 9**](#_heading=)

[**2.3 Major System Capabilities 9**](#_heading=)

[**2.4 Major System Conditions 10**](#_heading=)

[**2.5 Major System Constraints 10**](#_heading=)

[**2.6 User Characteristics 10**](#_heading=)

[**2.7 Assumptions and Dependencies 11**](#_heading=)

[**2.8 Operational Scenarios 11**](#_heading=)

[**3.0 System Capabilities, Conditions, and Constraints 12**](#_heading=)

[**3.1 Physical 12**](#_heading=h.39kk8xu)

[3.1.1 Construction 12](#_heading=h.1opuj5n)

[**3.1.2 Dependability 12**](#_heading=)

[**3.1.3 Adaptability 12**](#_heading=)

[**3.2 System Performance Characteristics 13**](#_heading=)

[**3.2.1 Load 13**](#_heading=)

[**3.2.2 Stress 13**](#_heading=)

[**3.2.3 Availability 13**](#_heading=)

[**3.3 Information Technology Management 13**](#_heading=)

[**3.4 System Operations 13**](#_heading=)

[**3.4.1 System Human Factors 13**](#_heading=)

[**3.4.2 System Usability 14**](#_heading=)

[**3.4.3 System Maintainability 14**](#_heading=)

[3.4.4 System Reliability 14](#_heading=h.2y3w247)

[**3.5 Policy and Regulation 14**](#_heading=)

[**3.6 System Life Cycle Sustainment 14**](#_heading=)

[**4.0 System Interfaces 14**](#_heading=)

[**4.1 Configuration files 14**](#_heading=h.1qoc8b1)

[**4.2 Financial Modelling Prep Api 14**](#_heading=h.4anzqyu)

[**4.3 IONOS 15**](#_heading=h.2pta16n)

[**4.4 phpMyAdmin 15**](#_heading=h.14ykbeg)

[**4.5 MySQL 15**](#_heading=h.3oy7u29)

[**4.6 Database (OLTP) 15**](#_heading=h.243i4a2)

[**4.7 Data Warehouse (FUTURE SCOPE) 15**](#_heading=h.j8sehv)

[**4.8 ATS User Interface 15**](#_heading=h.pwisjtbmqxt9)

[**5.0 Specific Requirements 16**](#_heading=)

[**6.0 Requirements Traceability Matrix 16**](#_heading=h.odc9jc)

**Revision Sheet**

| Revision Number | Date | Brief summary of changes |
| --- | --- | --- |
| Version 0.0 | 2023-10-07 | Baseline draft document |
| Version 0.1 | 2023-10-08 | Intro, purpose, and scope written |
| Version 0.2 | 2023-10-09 | Sys context, capabilities, conditions written |
| Version 0.3 | 2023-10-10 | General System Description fully written |
| Version 0.4 | 2023-10-11 | System Capabilities, Conditions, and Constraints |
| Version 1.0 | 2023-10-12 | Finalized version 1.0 |
| Version 1.1 | 2023-10-22 | Added/updated Figures, RTM, and general revision |
| Version 1.2 | 2023-11-05 | Final review and release |
| Version 1.3 | 2023-12-06 | Revised contributors, layout, data warehouse defined as future scope. Multiple section revisions. |
| Version 1.4 | 2024-01-17 | Updated for 471 |
| Version 1.5 | 2024-02-26 | * Updated section 5.0 with system requirements for user interface * Updated section 4 with ATS UI System Interface (4.8) * Updated section 1 - removed warehouse related stuff and added ATS UI * Updated section 2 - removed warehouse added UI * Updated section 3 * Updated architecture diagram * Updated use case diagram |
| Version 1.6 | 2024-04-09 | Final Revisions |

# **Introduction**

## **System Purpose**

The Algorithmic Trading System (ATS) aims to gather financial stock market data from various streams, to facilitate a larger stream of data for use with Machine Learning (ML) model analysis. Data is stored and prepared for ML training.

## **System Scope**

The ATS is a combination of components defined as:

* Automated Data collection scripts
  + Historical data collection for stocks, bonds, index composites and commodities
  + Real time data collection for stocks, bonds, index composites and commodities
  + Daily changes for stock ticker symbols representing traded companies
  + Quarterly company information
* Data consistency verification
* OLTP Database
  + Short term storage of data
* Automated Data insertion scripts
  + From data collection script(s) output to database
* Web Hosted user interface
  + System configuration management
  + Data export
  + System logging administration

## Goals and Objectives

* Automatically Gather financial data from one or more API sources
* Store data in a normalized database (At least 2NF)
* Ensure quality data through preprocessing
* Provide a User Interface for configuration management and system monitoring
* Automatic deletion of obsolete data

## **Definitions, Acronyms, and Abbreviations**

### **Key Definitions**

| **plan** | “A detailed scheme, program, or method worked out beforehand for the accomplishment of an objective.” [Heritage85]  —Defined set of procedures and the required resources to implement a policy. |
| --- | --- |
| **requirement** | “(1) A condition or capability needed by a user to solve a problem or achieve an objective.” [IEEE90] |
|  | “(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.” [IEEE90] |
| **software** | “Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system.” [IEEE90] |
| **specification** | “A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and, often, the procedures for determining whether these provisions have been satisfied.” [IEEE90] |
| **validation** | “The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.” [IEEE90] |
| **verification** | “(1) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. |
|  | (2) Formal proof of program correctness.” [IEEE90] |
| **product** | —Any tangible item that results from a project function, activity, or task. Examples of work products include customer requirements, project plans, design documents, source and object code, user’s manuals. |
| **database** | “A collection of data fundamental to a system.” [IEEE91] |

### **Key Acronyms and Abbreviations**

| IEEE | Institute of Electrical and Electronics Engineers |
| --- | --- |
| RTM | Requirements Traceability Matrix |
| EPG | Engineering Process Group |
| SysRS | System Requirements Specification |
| ATS | Algorithmic Trading System |
| OLTP | Online Transactional Processing |
| SRS | Systems Requirements Specification |
| SSH | Secure Socket Shell |
| DB | Database |
| GDPR | General Data Protection Regulation |
| ETL | Extract Transform Load |
| 2NF | Second Normal Form |

## **Key References**

* IEEE Std 1233-1998, IEEE Guide for Developing System Requirements Specifications [R1]
* [Financial Modelling Prep](https://site.financialmodelingprep.com/developer/docs) [R2]
* ATS Configuration Management Plan [R3]
* ATS Requirements Traceability Matrix[R4]
* ATS Software Design Document [R5]
* ATS User Manual[R6]

# **General System Description**

## **System Context**

The ATS system interacts with a variety of external entities, through the following means:

* Financial Data APIs
  + Enables collection of financial data
* Financial markets
  + Ensures availability of data, and by extension the availability of data collection
* Hosting service
  + IONOS, used to host the database, UI, and system software

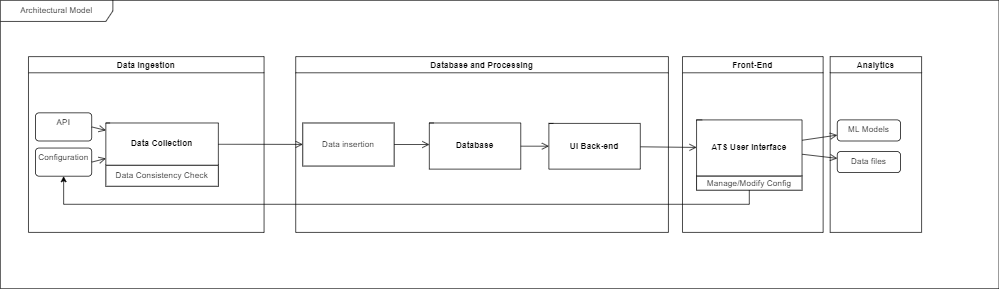


Figure 1: Top-level diagram of the ATS pipeline architecture

## **System Modes and States**

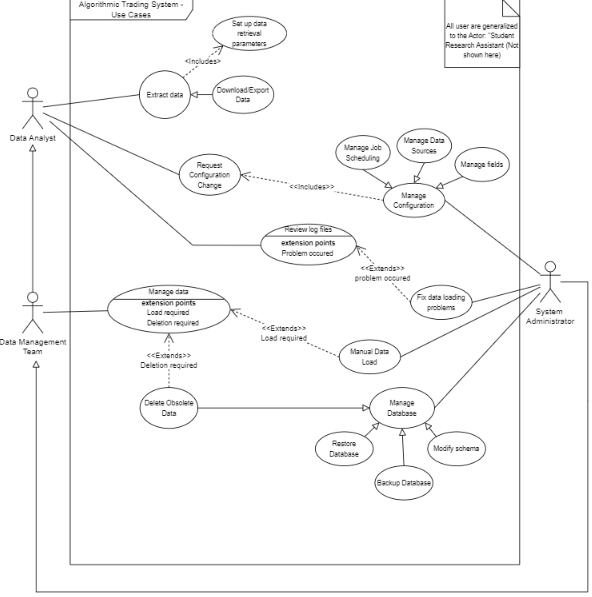


Figure 2: System Use Case Diagram

## **Major System Capabilities**

* The system can collect financial data via external data providers
* The system has the ability to clean and preprocess data
* The system is capable of storing data in an OLTP Database
* The system functions automatically with minimal need for human intervention
* Job scheduling and collection parameters are modifiable via the ATS UI

## **Major System Conditions**

* The system guarantees that collected data is available during and after trading hours
* The system is designed to function automatically to minimize manual operation
* To ensure relevance, the system must retrieve trading data every trading day
* The system must have a rigorous data validation process
* The system must retain financial data for 3 years
* Data older than 3 years should be automatically removed from the database

## **Major System Constraints**

* Data collection capabilities are contingent on the availability of data from external sources
  + Disruptions and inaccuracies in data sources may affect the system's ability to provide accurate data
  + To handle this problem, the system must implement error handling and notify users/administrators when such problems arise
* The database is currently hosted by a third-party service (IONOS)
  + Storage per database is currently limited to 2GB
  + Database integrity is contingent on the state of the hosting service
* Speed of data collection is limited by API rate limits
* The user interface is web hosted (IONOS), meaning an internet connection will be required to access it

## **User Characteristics**

* **Database Administrator**
  + Will handle system maintenance and user permissions
  + Will access and manage the database with full permissions
  + Will perform regular database backups

* **Data Analyst**
  + Be able to configure query parameters for the data collection processes.
    - Choose stock symbols, companies, and dates
  + Be able to access and extract data from the ATS
* **Student Research Assistant**

## **Assumptions and Dependencies**

* All data exported will use USD as the currency
* This system is dependent on IONOS web hosting services and functionality will be put on hold if those services are down or otherwise unavailable
* Data can be collected from alternative REST API’s if necessary
* Data sources are operational
* The database hosting service will have sufficient storage capacity

## **Operational Scenarios**

***Scenario 1:***

A group of financial data analysts are conducting research using a machine learning model to create market forecasts. In order to conduct accurate research, the group of analysts need historical and current data on stocks, bonds, indexes, and commodities:

1. The data analysts log into the ATS system interface with valid credentials.
2. The analysts navigate to the “Download Data” page via the navigation panel.
3. From here, the user selects the type of data they would like, date range, and the export data format. When all options have been selected, they press “Export File”
4. The data analyst is then able to use the data to train machine learning models.

***Scenario 2:***

John is a data analyst, who is working with a machine learning model to create market forecasts. John notices that he will need more sample data from different stocks to conduct their research:

1. John requests a change to the configuration files for the data collection processes to the system administrator.
2. The system administrator reviews the request to ensure it will not cause any disruption to the system.
3. The system administrator logs into the ATS UI with admin credentials
4. Admin navigates to “Change Configurations” page via the navigation panel
5. Admin selects “Add stock” and selects which new companies are to be gathered. Save changes.
6. Once the next scheduled query completes (or a forced query is made), John can now export the database information and receive a larger data set for his research.

# **System Capabilities, Conditions, and Constraints**

## **Physical**

### **Construction**

The system will be a web-hosted service with data stored on cloud servers.

The system should be hosted on servers or data centers. These facilities should maintain stable temperature and humidity levels to prevent overheating and potential hardware damage.

### **Dependability**

The system will be available to operate all year round and extract data from the stock market on a daily basis.

The performance may be reduced due to the high level of transactions in the databases when new data is being inserted, when data backups occur, or when the database is unavailable for other reasons.

Regular backups will be performed on the database.

The ATS User Interface is a web-hosted application that can be run on all common browsers. An internet connection as well as authorized access to the database is required.

### **Adaptability**

Storage sizes available from IONOS are large enough to maintain data entry for the foreseeable future. In the current version of the ATS, data is removed from storage when it is 3 years or older. Should an increase in data input cause issues, then the current hosting plan can be upgraded to allow for increased storage.

## **System Performance Characteristics**

### **Load**

* Data analysts: 9:00 a.m. to 8 p.m. EST workdays

### **Stress**

In the event of the server receiving a large number of requests, the server will be less responsive and may cause time-out errors. Data collection may slow down if the data sources are experiencing high traffic. Average system processing time under normal conditions currently executes each job in <15 minutes.

### **Availability**

The availability heavily relies on the cloud service provider.

## **Information Technology Management**

The first stage is data collection using Python and requests to the API, followed by sanitization, field mapping, and conversion into the format defined by the system. A JSON file is created and saved as a result of this stage.

The second stage is data insertion using Python and SQL into the OLTP database. The JSON files generated are read by these scripts, extracted, and inserted into the database.

A pipeline architecture will be used to allow for modularity and reusability. [IONOS](https://www.ionos.com/) - a third-party hosting service, hosts the database and provides cloud storage. The data collection scripts, as well as the database insertion scripts, will be written in Python.

Finally, the data will be extracted and used by data analysts. This can be done in two ways: either directly through the IONOS phpMyAdmin interface or via the ATS user interface. For specific details on this, please refer to the ATS User Manual [R6].

## **System Operations**

* Data Analyst:
  + Collection and storage of financial data by the system
  + Analysts will incorporate the stored data into their machine learning models, helping them make more accurate risk/reward assessments for their clients or portfolios.

### **System Human Factors**

Users require IONOS web host credentials or ATS credentials to access the system. These may include: secure shell access, database access.

### **System Usability**

Users require a personal computer and a stable network connection to access the system.

### **System Maintainability**

Maintenance of the system refers to software updates and database maintenance, preferably conducted once a month or less. Database clean-up should be conducted during maintenance by a database administrator or authorized developers. The software updates are handled by the maintenance team which will maintain the data integrity and error handling. Hardware inspection and maintenance should be conducted by the cloud service provider on a timely basis.

### **System Reliability**

The system will be available for a minimum of 23 hours a day.

The system should be operational at all times outside of scheduled updates and maintenance. When possible, this maintenance should occur on weekends when financial markets are not active.

## **Policy and Regulation**

The system must have policies in place to ensure the privacy and security of sensitive financial data. Compliance with industry-specific standards, for example GDPR, should be required. Depending on the jurisdiction in which the organization operates, there may be regulatory requirements related to financial forecasting and stock trading.

## **System Life Cycle Sustainment**

If a change to the system is deemed to be required, then a change request must be submitted by the client to initiate the change process. See *Change Request Policy* in the ATS Configuration Management Plan for details [R3]

# **System Interfaces**

## Configuration files

This system interface is used for the API querying scripts. They provide the configuration settings for the data extraction process. The config files specify the parameters used by automated Python scripts for data collection. The Python querying scripts depend on these configuration files to access the correct API query.

## Financial Modelling Prep Api

The Financial Modelling Prep API [R2] interface is a REST API that allows the system to make requests to financial data sources, including asset and company data. It involves the use of the HTTPS protocol and authentication mechanisms, such as API keys, to access real-time market information. The call to the REST API is made via URI with specific headers, and the response is given in JSON format.The API interface relies on valid API keys and query options supplied by the configuration files.

## IONOS

IONOS serves as a visual interface used by data analysts and system administrators to extract data stored in the database, including data related to stocks and trading activities. It allows users to interact with the system for data retrieval and management, including historical stock data. The IONOS interface is dependent on the hosting providers' servers, and their support offerings.

## phpMyAdmin

PhpMyAdmin is a visual interface that is used on IONOS to create and manage the OLTP databases. It is accessible through a web browser, which means system administration can manage the databases from virtually any location.

## MySQL

MySQL is the programming interface used for extracting, modifying, and inserting data into tables. It is accessed through SSH on the IONOS server utilizing MySQL CLI, and by phpMyAdmin’s interface. This will process and run the database triggers, inserts, and modifications. MySQL must be accessed using SSH, or phpMyAdmin using their respective appropriate credentials.

## Database (OLTP)

The OLTP databases are used to store and retrieve data. This involves database connections, SQL queries, and data manipulation through PhpMyAdmin using the MySQL interface. The querying scripts interact with the OLTP database to store and retrieve stocks and trading data, while the ETL process transfers data from OLTP to the Data Warehouse, including relevant stock and trading data.

## Data Warehouse (FUTURE SCOPE)

The Data Warehouse interface is used to store data long-term in an easy to understand table schema. It will be accessed initially through the same way as the OLTP DB, through phpMyAdmin. MySQL ETL scripts will manage the data transfer between the OLTP database and the Data Warehouse.

## ATS User Interface

The ATS User Interface is used to change configuration files, job scheduling and data exportation. This is a web application that is accessible to admins and regular users such as students (Regular users can only access the data exportation page).

# **Specific Requirements**

***Functional Requirements***

* The system automatically collects data from financial markets using API calls
* System allows for modification of configuration files, including managing data sources or field names
* The system cleans and ensures the consistency of collected data
* Data is automatically inserted into the database
* Data is stored in an OLTP database normalized to at least 2NF
* Data undergoes ETL process into the data warehouse (FUTURE SCOPE)
* All kinds of system documentation
* System collects data via web scraping for data not in the API (FUTURE SCOPE)
* The system UI provides full functionality to admins. Including account creation, configuration changing, job scheduling, and data exportation.
* The system UI provides only data exportation capabilities to regular users such as students.

***Non-Functional Requirements***

* Data is available during and after trading hours
* Data will be processed within a maximum of 24 hours into the OLTP database
* Stock, index, bond, and commodity data will be collected daily at market close
* The system will gather data from all the companies listed on the S&P 500
* Company data will be collected every month
* Historical financial data will be updated daily
* Sensitive data will not be stored anywhere publicly accessible
* Database is backed up monthly
* Data warehouse is populated monthly
* System should be able to run for several years every day to collect enough data for analysis

# 6.0 **Requirements Traceability Matri**x

*Please refer to the separate ATS Requirements Traceability Matrix (RTM) [R4] document*