**Automated Crypto Trading Bot**

**Software Requirements Specification**

**Practicum - I**

**Presented By**

**Syed Waqar Ali 24k-8301**

**Mohammad. Mustansir 24k-8305**

**Asad 24k-xxxx**

Table of Contents

[Introduction 3](#_Toc209871346)

[Purpose 3](#_Toc209871347)

[Document Conventions 3](#_Toc209871348)

[Intended Audience and Reading Suggestions 4](#_Toc209871349)

[Product Scope 4](#_Toc209871350)

[References 5](#_Toc209871351)

[Overall Description 5](#_Toc209871352)

[Product Perspective 5](#_Toc209871353)

[Product Functions 5](#_Toc209871354)

[Operating Environment 5](#_Toc209871355)

[Design and Implementation Constraints 5](#_Toc209871356)

[User Documentation 5](#_Toc209871357)

[Assumptions and Dependencies 5](#_Toc209871358)

[External Interface Requirements 5](#_Toc209871359)

[User Interfaces 5](#_Toc209871360)

[Hardware Interfaces 5](#_Toc209871361)

[Software Interfaces 5](#_Toc209871362)

[Communications Interfaces 5](#_Toc209871363)

[System Features 5](#_Toc209871364)

[User Management 5](#_Toc209871365)

[Bot Control & Configuration 5](#_Toc209871366)

[Real-Time Market Data 5](#_Toc209871367)

[Trade History 5](#_Toc209871368)

[Curated News Feed 5](#_Toc209871369)

[Other Nonfunctional Requirements 5](#_Toc209871370)

[Performance Requirements 5](#_Toc209871371)

[Safety Requirements 5](#_Toc209871372)

[Security Requirements 5](#_Toc209871373)

[Software Quality Attributes 5](#_Toc209871374)

[Business Rules 5](#_Toc209871375)

[Other Requirements 5](#_Toc209871376)

[Appendix A: Glossary 5](#_Toc209871377)

[Appendix B: Analysis Models 5](#_Toc209871378)

[Appendix C: To Be Determined List 5](#_Toc209871379)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This Software Requirements Specification (SRS) defines the functional and non-functional requirements for the **Automated Crypto Trading Bot (Release 1.0)**. The product under specification is a **web-based trading automation system** that integrates with the **MetaTrader 5 (MT5)** platform to facilitate cryptocurrency trading.

The scope of this SRS is limited to the features identified in the practicum’s vision and scope document. Specifically, this release will deliver an MVP (Minimum Viable Product) focusing on secure user management, configuration of a single basic trading strategy (Simple Moving Average crossover), real-time market data visualization, a curated cryptocurrency news feed, and trade history tracking.

This SRS only covers the **software component of the Bot** that interacts with MT5 via its Python API. It does not specify requirements for the MT5 platform itself, broker infrastructure, or external financial systems.

## Document Conventions

This document follows **IEEE SRS standard formatting** and conventions. The following typographical standards are applied:

* **Bold text** is used for key terms, roles, or system names.
* *Italic text* is used for emphasis or to denote external sources.
* Requirements are expressed in **“shall”** form to indicate mandatory features, whereas **“should”** denotes optional or desirable features.
* Numbered lists (e.g., 1.0, 1.1, 1.2) indicate hierarchical requirement sections for easier traceability.
* Priorities for requirements are explicitly stated where needed, and higher-level requirements are not assumed to be inherited unless otherwise specified.

## Intended Audience and Reading Suggestions

This document is intended for multiple stakeholders:

* **Developers** – to understand the technical requirements and constraints for implementation.
* **Project Managers** – to monitor scope, ensure alignment with practicum goals, and track deliverables.
* **Quality Assurance (QA) Team** – to derive test cases, validate compliance with functional and non-functional requirements, and ensure product reliability.
* **End Users (Traders)** – to gain clarity on the intended capabilities of the Bot.
* **Documentation Writers** – to prepare user manuals and training materials.

The document is organized logically:

* Section 1 provides the **introduction, purpose, scope, and references**.
* Section 2 will describe **overall system details** including user needs, assumptions, and constraints.
* Section 3 will cover **functional requirements**.
* Section 4 will outline **non-functional requirements** such as performance, security, and availability.

Readers unfamiliar with the project should start with Sections 1 and 2 to gain an overview before moving to detailed requirements in Section 3. Developers and testers should pay close attention to Sections 3 and 4, while project managers may focus more on the scope, risks, and success metrics.

## Product Scope

The **Automated Crypto Trading Bot** is designed to simplify and automate cryptocurrency trading for users of the MT5 platform. The primary goal is to allow traders to configure and execute automated strategies without needing constant market supervision.

The product will deliver value in three key ways:

1. **Efficiency** – Automating repetitive trading actions, reducing manual intervention.
2. **Accessibility** – Providing a responsive web interface usable by both novice and experienced traders.
3. **Reliability & Security** – Ensuring secure handling of user credentials and consistent execution of trades.

By enabling automated crypto trading, the system supports corporate strategies of promoting **financial technology innovation**, **scalable cloud-based trading solutions**, and **broader accessibility for retail traders**.

Since a separate vision and scope document exists, this SRS builds upon it without duplicating unnecessary details. Instead, it refines the requirements into precise technical and functional specifications for the development team

## References

The following documents and sources are referenced in this SRS:

1. **Vision and Scope Document for Automated Crypto Trading Bot**
   * Author: Practicum Development Team
   * Version: 1.0
   * Date: [Insert Date]
   * Source: Internal project documentation
2. **MetaTrader 5 (MT5) Python API Documentation**
   * Author: MetaQuotes Software Corp.
   * Version: Latest stable release (as of project start date)
   * Source: <https://www.metaquotes.net/en/metatrader5/api/python>
3. **IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)**
   * Author: IEEE Computer Society
   * Standard reference for structuring SRS documents.
4. **Cryptocurrency Market Data Sources (for live feeds & news)**
   * Example: CoinMarketCap, Binance API, CryptoNews APIs
   * Source: Public APIs (to be finalized during integration).

# Overall Description

## Product Perspective

The Automated Crypto Trading Bot is a **new, self-contained product** designed to integrate with the **MetaTrader 5 (MT5)** trading platform via its Python API. It is not part of a product family, nor a replacement for any existing system, but rather an original solution that complements MT5 by adding **web-based automation and monitoring capabilities** for cryptocurrency trading.

The system functions as a middleware between the user and MT5, providing a user-friendly interface for managing Bot strategies, monitoring real-time data, and reviewing trade history. The MT5 platform handles the actual execution of trades through the broker, while the Bot handles strategy logic, user preferences, and data visualization.

A high-level context diagram is as follows (simplified text description):

* **Users (Novice/Experienced Traders)** interact with the **Web Application**.
* The **Web Application** communicates with the **MT5 Python API**.
* The **MT5 API** connects to the **Broker MT5 Server** for trade execution and market data.
* External **News APIs** feed curated crypto market updates into the application.

This architecture ensures that the Bot remains lightweight while relying on MT5 for execution reliability.

## Product Functions

The major functions of the Automated Crypto Trading Bot include:

* **User Management**: Secure registration, login, authentication.
* **Bot Control & Configuration**: Start/stop bot, configure parameters (lot size, stop loss, take profit, SMA crossover rules).
* **Market Data Dashboard**: Display of live prices for predefined crypto assets, plus customizable watchlist.
* **Curated News Feed**: Integration of real-time cryptocurrency-related news.
* **Trade History**: Display of open and closed positions, including profit/loss summaries.
* **Audit Logging**: Tracking of all trade actions and configuration changes for accountability.

These functions together provide a **complete automated trading experience** while remaining within the MVP scope.

## Operating Environment

The software will operate in the following environment:

* **Server Environment**:
  + MT5 terminal installed and running on a dedicated server.
  + Supported OS: Windows Server 2019+ (MT5 requirement).
  + Python 3.9+ runtime environment for MT5 API integration.
* **Client Environment**:
  + Web-based front-end accessible via modern browsers (Chrome, Firefox, Edge, Safari).
  + Responsive design supporting both desktop and mobile devices.
* **Networking**:
  + Stable internet connection to communicate with broker MT5 servers and external news APIs.

## Design and Implementation Constraints

* **MT5 API Limitation**: Strategy development restricted to functions exposed by the MT5 Python API.
* **Crypto-Only Scope**: Forex, stocks, and indices are out of scope.
* **Single MT5 Account**: Each user is limited to one account integration.
* **Server Dependency**: MT5 terminal must remain active on the deployment server.
* **Security Standards**: User credentials and MT5 details must be encrypted.
* **UI Technology**: Must be a web application with responsive design.

## User Documentation

The following documentation will be provided alongside the software:

* **User Manual** (PDF & Web version): Step-by-step guide for registration, bot configuration, and trade monitoring.
* **Quick Start Guide**: A simplified one-page guide for novice traders.
* **Online Help/FAQ**: Integrated into the application for troubleshooting common issues.
* **Video Tutorial (Optional)**: Short demo of how to set up and start the Bot.

## Assumptions and Dependencies

**Assumptions**:

* Users have an MT5 broker account supporting cryptocurrency trading.
* Both demo and real MT5 accounts are supported.
* Users have stable internet connectivity.

**Dependencies**:

* MT5 Python API (subject to changes by MetaQuotes).
* Broker server availability for trade execution.
* Third-party crypto news APIs for market updates.
* Cloud infrastructure availability for scalability and uptime guarantees.

# External Interface Requirements

## User Interfaces

The system will feature a **web-based UI** with the following design standards:

* Responsive layout for desktop and mobile.
* Standard navigation bar with **Dashboard, Bot Settings, Trade History, News**.
* Clear, minimalistic design with emphasis on charts and data visualization.
* Error messages displayed in modal pop-ups with clear corrective suggestions.
* Consistent use of colors/icons to indicate trade status (profit/loss).
* “Help” button on each page linking to documentation.

## Hardware Interfaces

* The software interacts with the **MT5 client application** installed on the server.
* No direct hardware integration is required other than a stable server with adequate memory (minimum 8 GB RAM) and CPU resources.

## Software Interfaces

* **MetaTrader 5 Python API**: Used for trading execution and market data retrieval.
* **Database (PostgreSQL/MySQL)**: For storing user profiles, bot configurations, trade history, and logs.
* **Web Framework (Django/Flask/Node.js)**: For API and business logic.
* **External News API**: For curated crypto market updates.

## Communications Interfaces

* The application will use **HTTPS** for secure client-server communication.
* MT5 communication with the broker uses proprietary MT5 protocols.
* External APIs (news, market data) accessed via **REST APIs** with JSON.
* All data transfers must be encrypted (TLS 1.2+).

# System Features

## User Management

**Description and Priority**:  
Secure account creation, login, and authentication. Priority: **High**.

**Stimulus/Response Sequences**:

* User registers → System validates input → System creates account → Confirmation message sent.
* User logs in → Credentials verified → Access granted.

**Functional Requirements**:

* REQ-1: The system shall allow new users to register with email and password.
* REQ-2: The system shall encrypt all stored passwords.
* REQ-3: The system shall authenticate users via login before granting access.

## Bot Control & Configuration

**Description and Priority**:  
Ability to start, stop, and configure Bot parameters. Priority: **High**.

**Stimulus/Response Sequences**:

* User sets parameters (lot size, stop loss, SMA crossover).
* User starts bot → Bot connects to MT5 → Trades executed automatically.

**Functional Requirements**:

* REQ-4: The system shall allow users to start and stop the Bot at will.
* REQ-5: The system shall allow users to configure lot size, stop loss, and take profit.
* REQ-6: The system shall execute trades based on SMA crossover logic.

## Real-Time Market Data

**Description and Priority**:  
Display live crypto prices and watchlist. Priority: **High**.

**Functional Requirements**:

* REQ-7: The system shall display price updates with <1s delay.
* REQ-8: The system shall allow users to maintain a custom watchlist.

## Trade History

**Description and Priority**:  
Record and display bot-executed trades. Priority: **Medium**.

**Functional Requirements**:

* REQ-9: The system shall maintain a log of open and closed trades.
* REQ-10: The system shall display profit/loss summaries.

## Curated News Feed

**Description and Priority**:  
Provide crypto-related news. Priority: **Low-Medium**.

**Functional Requirements**:

* REQ-11: The system shall integrate a news feed API.
* REQ-12: The system shall refresh news updates hourly.

# Other Nonfunctional Requirements

## Performance Requirements

* Market data updates must occur with **<1s delay**.
* API response time must be **<500ms under 100 concurrent users**.

## Safety Requirements

* The system shall prevent accidental trade execution without user confirmation of bot start.
* The system shall log all actions to prevent misuse

## Security Requirements

* All credentials stored with encryption.
* Communication over HTTPS/TLS.
* Session management with auto-timeout.

## Software Quality Attributes

* **Availability**: 99.5% uptime.
* **Usability**: Intuitive interface for novices.
* **Scalability**: Cloud-ready to handle growth.
* **Auditability**: Logs maintained for all trade actions.

## Business Rules

* Only authenticated users can run a Bot.
* Each user limited to one MT5 account.
* The Bot trades **only crypto instruments**.

# Other Requirements

* Database must support ACID transactions for trade records.
* System should be deployable on a cloud environment (AWS/Azure).
* Legal compliance with data protection laws (e.g., GDPR if EU users).

# Appendix A: Glossary

* **MT5** – MetaTrader 5 trading platform.
* **SMA** – Simple Moving Average.
* **CFD** – Contract for Difference.
* **MVP** – Minimum Viable Product.

# Appendix B: Analysis Models

(TBD – UML diagrams, class diagrams, data flow diagrams to be added later).

# Appendix C: To Be Determined List

* TBD-1: Final selection of third-party news API.
* TBD-2: Choice between PostgreSQL or MySQL for database.