

Software Requirements Specification (SRS) for Crypto Trading Bot

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for a **Crypto Trading Bot** that integrates with Binance and is controlled via a Discord bot. The system is designed to offer both manual and automated trading functionalities for cryptocurrency traders, using Binance as the trading platform. The document aims to provide a clear specification for developers to implement a user-friendly discord bot for crypto trading.

1.1.1 Background and Objectives of the Project

The Crypto Trading Bot project is intended for cryptocurrency enthusiasts seeking an intuitive way to execute trades and automate trading using Discord commands. The system enables users to control Binance accounts, set alerts, and track trading activity, thus combining convenience with effective trading tools. The bot's objectives include ensuring stability, monitoring, alerting, automation, and providing a straightforward interface for managing trades. We decided to go with this project due to our passion and hope for the crypto space. The objective of this project is decidedly not to create profitable trading strategies but rather to develop a bot that serves as groundwork upon which, in later iterations, proper strategies for trading can be implemented.

1.2 Product Scope

1.2.1 Must-Criteria

1. The system shall function as a Discord bot.
2. The system shall be using the Binance Spot Market.
3. The system shall accept commands only if issued from a Discord guild channel.
4. Until multi-account support is implemented, the system shall use a single Binance account for all commands issued by users.
5. Until multi-account support is implemented, the system shall provide guild admins with the ability to set Binance API credentials via a Discord command.
6. The system shall provide users with the ability to place market orders via a Discord command.
7. The system shall provide users with the ability to place limit orders via a Discord command.

8. The system shall be able to monitor executed orders and notify users in a Discord channel about them.
9. The system shall include eventual transaction fees in order execution notifications.
10. The system shall support multiple Binance-supported spot market currency pairs.
11. The system shall provide users with the ability to view the balance of kept assets via a Discord command.
12. The system shall support at least one automated trading strategy.
13. The system shall provide users with the ability to deploy and undeploy an automated trading strategy via a Discord command.

1.2.2 Should-Criteria

14. The system should provide users with the ability to configure a specific Discord channel for order monitoring notifications.
15. The system should provide users with the ability to configure alerts for when a given currency pair reaches a given price.
16. The system should provide users with the ability to configure a specific Discord channel for alert notifications.
17. The system should support multiple deployments of the same automated trading strategy.
18. The system should provide users with the ability to view current strategy deployments.
19. The system should provide users with the ability to configure the maximum amount of assets the bot is allowed to use for automated trading per strategy deployment.
20. The system should provide users with the ability to toggle whether realized earnings are reinvested or excluded from the amount of usable assets per strategy deployment.
21. The system should provide users with the ability to view the current balance of a given asset and how much of it is allocated to strategy deployments.
22. The system should provide users with the ability to set strategy-specific configuration parameters per strategy deployment.

1.2.3 Could-Criteria

23. The system could have at least two automated trading strategies.
24. The system could provide users with the ability to generate performance charts showing asset development over time via a Discord command.
25. The system could provide users with the ability to view general statistics regarding trades and their performance.
26. The system could provide users with the ability to pause and resume strategy deployments..
27. The system could provide each user in the Discord guild with the ability to command their own Binance account independently of others (multi-account support).
28. When multi-account support is implemented, the system could provide users with the ability to add Binance API credentials for their account via a Discord command.

1.2.4 Won't-Criteria

- 29. The system will not support integration with platforms other than Binance and Discord.
- 30. The system will not support margin or futures trading in any version.
- 31. The system will not check for over assignment of assets to strategy deployments.

1.3 Definitions, Acronyms, and Abbreviations

- **Discord Guild:** A term used to refer to a server within Discord, which is a community platform for communication. A Discord guild is a virtual space where users can interact through text, voice, and video channels. Each guild can have its own set of channels, roles, and permissions, allowing users to create communities with distinct purposes, such as gaming, education, or interest-based groups.
- **Binance Account:** A cryptocurrency trading account used to execute trades.
- **Deployment/Strategy Deployment:** An instance of an automated trading strategy acting independently of others with its own set of parameters.
- **Strategy/Automated Trading Strategy:** An in code implemented trading strategy with a set of customizable parameters.
- **Asset:** A crypto currency of which an amount can be owned.
- **Currency-Pair:** Two Assets together. One base asset's price is quoted in another asset. Ex.: BTC_FDUSD -> Base asset BTC (BitCoin) is Quoted in FDUSD (USD stable coin)
- **Discord Guild/Guild:** Often also referred to as Discord Server.
- **Market Order:** To order to buy/sell a certain asset for another asset at the current price.
- **Limit Order:** To order to buy/sell a certain asset for another asset at a specified price when available.

2. General Overview

2.1 Current State Description

The current landscape of crypto trading requires manual interaction with trading platforms, which is cumbersome for users who want automation. The Crypto Trading Bot addresses this by providing a convenient method to execute trades using Discord commands, integrating seamlessly with Binance.

2.2 Product Deployment

2.2.1 Application Areas

The bot will be used by individual traders or small groups on Discord to automate their cryptocurrency trading activities.

2.2.2 Target Audience and Skill Level

The target audience includes cryptocurrency enthusiasts and traders with basic to intermediate knowledge of cryptocurrency trading and Discord.

2.2.3 Operating Conditions

The bot will be operated from a dedicated server hosted at Hetzner, ensuring 24/7 availability and responsiveness to user commands. The system is developed using Spring Boot and PostgreSQL for data management. It is a Docker Compose project deployed on the dedicated Hetzner server, which ensures a consistent deployment environment.

2.3 Product Environment

2.3.1 System Interface

The bot will interface with Binance through their public API and SDK.

2.3.2 User Interface

Users will interact with the bot through Discord commands.

2.3.3 Software Interface

The system will use the Discord API for command processing and the Binance API for executing trades.

3. Functional Requirements

Functional requirements have been detailed under product scope using the MoSCoW prioritization. All requirements are formulated to be verifiable to ensure that implementation meets the expected outcomes.

4. Non-Functional Requirements

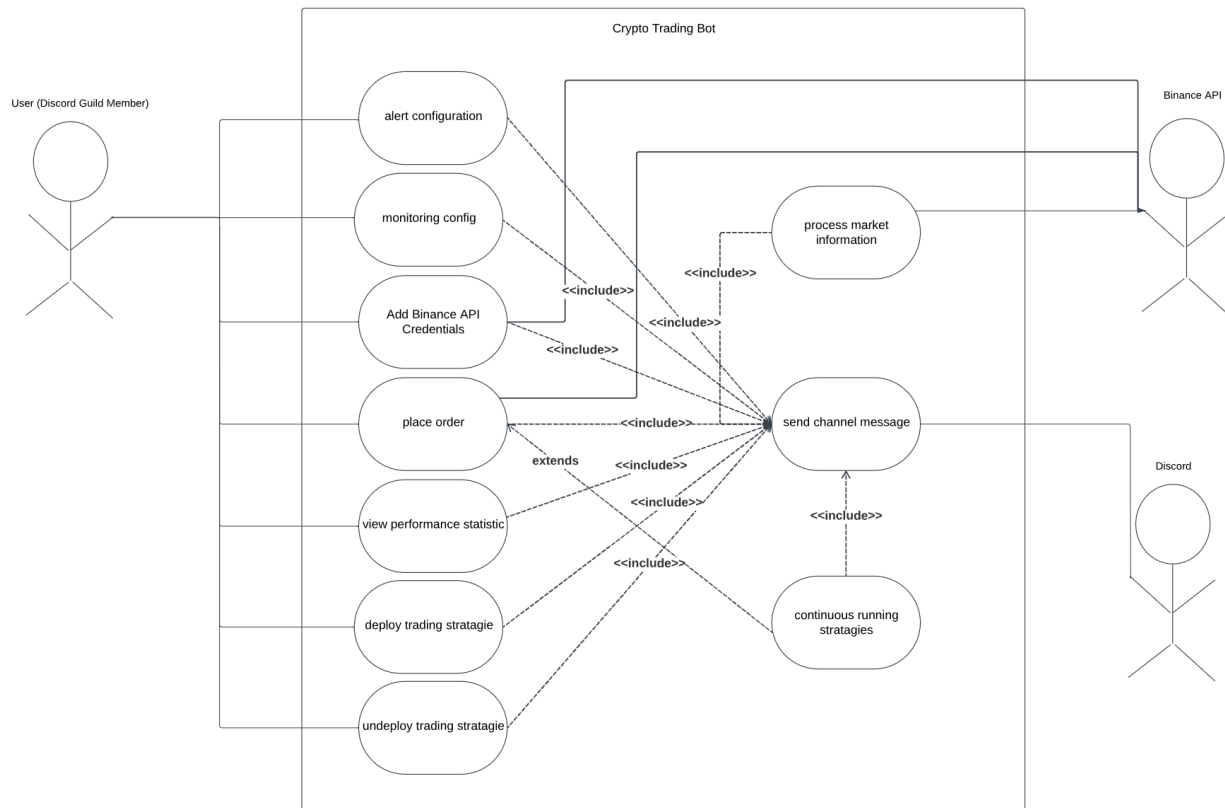
4.1 Data Management Requirements

32. The system should store Binance API credentials securely using AES-256 encryption to ensure the protection of sensitive user data.

4.2 User Interface Requirements

33. The system should respond to commands within 3 seconds to ensure a smooth user experience.

5. Use Case Diagram



6. Use Case Descriptions

Use Case: Add Binance API Credentials

- **Actor:** User (Discord Guild Member)
- **Description:** User adds their Binance API credentials through a Discord command.
- **Stimulus:** Command
- **Response:** The bot securely stores the credentials and responds to the user.

Use Case: Place Manual Orders

- **Actor:** User
- **Description:** User places a manual market or limit order for a currency pair.
- **Stimulus:** Command
- **Response:** The bot executes the order via Binance and confirms the action in Discord.

Use Case: Deploy Trading Strategy

- **Actor:** User

- **Description:** User deploys an instance of an automated trading strategy.
- **Stimulus:** Command
- **Response:** The bot begins executing trades based on the chosen strategy and configuration.

Use Case: Undeploy Trading Strategy

- **Actor:** User
- **Description:** User undeploys an instance of an automated trading strategy.
- **Stimulus:** Command
- **Response:** The bot no longer performs trades with this strategy deployment as it is removed and its configuration lost. Sends a confirmation message to the user.

Use Case: Configure Alerts

- **Actor:** User
- **Description:** User sets price alerts for a specific asset.
- **Stimulus:** Command
- **Response:** The bot sends alerts when the price threshold is reached.

Use Case: Configure Monitoring

- **Actor:** User
- **Description:** User sets which guild channel should be used for monitoring.
- **Stimulus:** Command
- **Response:** The bot confirms the action and new alerts are sent in the new channel

Use Case: View Performance Statistics

- **Actor:** User
- **Description:** User views trading performance, including profit/loss and total trades.
- **Stimulus:** Command
- **Response:** The bot provides a summary of the statistics in Discord.

8. Appendix

8.1 Glossary

- **API:** Application Programming Interface, used for communication between the bot and external services.
- **SDK (Software Development Kit):** A collection of software development tools, libraries, documentation, and code samples provided by vendors to facilitate the creation of applications for specific platforms or environments. An SDK typically includes APIs, debugging tools, and other utilities that simplify development by providing a standardized set of tools and resources.

- **AES-256:** Advanced Encryption Standard with 256-bit keys, used for secure data storage.
- **Discord Guild:** A term used to refer to a server within Discord, which is a community platform for communication. A Discord guild is a virtual space where users can interact through text, voice, and video channels. Each guild can have its own set of channels, roles, and permissions, allowing users to create communities with distinct purposes, such as gaming, education, or interest-based groups.
- **Binance Account:** A cryptocurrency trading account used to execute trades.
- **Deployment/Strategy Deployment:** An instance of an automated trading strategy acting independently of others with its own set of parameters.
- **Strategy/Automated Trading Strategy:** An in code implemented trading strategy with a set of customizable parameters.
- **Asset:** A crypto currency of which an amount can be owned.
- **Currency-Pair:** Two Assets together. One base asset's price is quoted in another asset. Ex.: BTC_FDUSD -> Base asset BTC (BitCoin) is Quoted in FDUSD (USD stable coin)
- **Discord Guild/Guild:** Often also referred to as Discord Server.
- **Market Order:** To order to buy/sell a certain asset for another asset at the current price.
- **Limit Order:** To order to buy/sell a certain asset for another asset at a specified price when available.

8.2 References

- Binance API Documentation: <https://binance-docs.github.io/apidocs/>
- Discord Developer Documentation: <https://discord.com/developers/docs>