# Software Requirements Specification

**for**

# Bybit Cryptocurrency Derivatives Trading Bot

**Version 1.1 approved**

**Prepared by Steven Wilkins**

**Eastern Oregon University – CS401 Capstone**

**August 4, 2022**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| Steven Wilkins | 8-4-2022 | Initial document creation | 1.0 |
| Steven Wilkins | 1-5-2023 | Fix Fixed “Error: Bookmark not defined” in table of contents.<br>Changed requirements numbering to be consistent with rest of document. | 1.1 |

# 1. Introduction

The project Bybit Cryptocurrency Derivatives Trading Bot is an automated system for cryptocurrency price data analysis and trade entries. It automates many the tasks which consume a trader's time.

The main Objective of this project is to perform technical analysis on live price data retrieved from the Bybit exchange as well as open and close positions on the trader's behalf. This software allows the trader to choose the currency pair to trade, set the timeframe, the position amount, when to take profit, where to set stop loss, and how much leverage to apply.

This software can be started from the command prompt of any operating system possessing Python 3.8.13 or greater. It also provides a clean and user-friendly interface to the users.

## 1.1 Purpose

The purpose of this project is to provide the average retail trader with an automated system which will trade cryptocurrency derivatives on the Bybit exchange using a trading strategy.

## 1.2 Document Conventions

This document uses the following conventions:

| | |
|---|---|
| Trader | User |
| Exchange | https://www.bybit.com |
| Asset | Cryptocurrency pair being traded |
| Candles | Open, High, Low, Close prices for given timeframe |
| Leverage | Exchange provided loan to trade with using account as collateral |
| TF | Timeframe |
| TA | Technical Analysis |
| This Software | The Bybit Cryptocurrency Derivatives Trading Bot |
| GUI | Graphical User Interface |

## 1.3 Need/Motivation

Cryptocurrency markets are active 24x7, unlike the stock market and forex, which deprives the retail trader of any substantial opportunities to step away from the charts to get some rest or spend time with loved ones. This software aims to remedy this inconvenient situation by automating the TA of an Asset and while entering and exiting positions according to the values provided by the Trader.

## 1.4 Product Scope

The Bybit Cryptocurrency Derivatives Trading Bot is software that allows a cryptocurrency derivatives trader to automate their trades using the Bybit ( https://www.bybit.com/ ) platform. The user configures the software with their Bybit API key and secret, the currency pair that they want to trade, how much leverage (up to 100x) they want to apply to their trades, the dollar amount of their trades, where they would like to set a stop loss if desired, and when to take profit on a trade. Once started, the software continuously pulls candlestick price data directly from the Bybit platform and performs technical analysis to determine whether any buy or sell signals are detected. If a buy signal is discovered, a long position is opened according to user provided settings. Alternatively, if a sell signal is discovered, then a short position is opened with the same settings.

What this software does do:

- Retrieves OHLC (Open, High, Low, Close) price data directly from the Bybit exchange.
- Calculates SuperTrend and 200-Period MA (moving average) from OHLC data.
- Produces entry signals for long and short positions.
- Allows the user to set leverage for trades.
- Allows the user to set position size.
- Allows the user to set stop loss for trades.
- Allows the user to set when to take profit for trades.
- Uses a config.py file to set the user's Bybit API and trade specifications in this software.

What this software does not do:

- Create Bybit account.
- Create Bybit API key.
- Transfer cryptocurrency into or out of Bybit account.
- Close open positions if software disconnects or fails. Setting stop loss and take profit is strongly recommended.

# 2. User Requirements

## 2.1 Product Perspective

The Bybit Cryptocurrency Derivatives Trading Bot allows the trader to step away from their monitor and allows them to participate in their life a little more. This is done by automating their trade entries and exits. This software doesn't buy cryptocurrency but allows you to trade the derivatives of the cryptocurrency that you already own, using leverage.

## 2.2 User Classes and Characteristics

There are only two user classes who would use this software. One is the non-technical user, who will run the software as it is distributed. The other is the technical user, who will modify the software to suit their needs more closely.

### 2.2.1 Non-Technical User

The non-technical user has little to no knowledge of how to modify source code. For this user class it is recommended that they only modify the config.py file in the manner which is specified in Section 2.2 of this document.

### 2.2.2 Software Developers

A user who possesses some software development experience, and knows the Python 3 programming language, may wish to modify the software's source code in the bot.py file. This is done at the user's own risk and should only be attempted if they are confident that they understand the existing algorithms and implementations.

## 2.3 Product Functions

- Provide Trader with a method to configure the software.
- Connect to Trader's Bybit account.
- Fetch price data from Bybit.
- Perform technical analysis on the fetched price data.
- Enter trade positions.
- Exit trade positions.

## 2.4 User Documentation

User instructions and documentation can be found within the project's README.md file.

## 2.5 Design and Implementation Constraints

- python 3.8.13
- pip 3
- ccxt==1.62.2
- pandas==1.3.4
- numpy
- schedule==1.1.0

## 2.6 Assumptions and Dependencies

- That Bybit remains a functional exchange.
- That Bybit maintains a public and private API.
- That the Python CCXT library is maintained and remains available.

- That a reliable internet connection is available.

# 3. System Requirements

The Trader will interface with this software by modifying config.py in a text editor. There is no graphical user interface for this software currently. The following sub-sections introduce and explain the functionality which must be provided by this software.

## 3.1 Connect to Trading Account

### 3.1.1 Description

Connects trader's account to this software.

### 3.1.2 Interface

Trader must set API_KEY and SECRET_KEY in config.py.

### 3.1.3 Functional Requirements

**3.1.3.1:** Use config.py to setup this software with the generated API_KEY and API_SECRET.
**3.1.3.2:** This software will connect to the exchange with the provided API_KEY and API_SECRET.
**3.1.3.3:** The connection will provide this software with access to private account specific data.

## 3.2 Fetch Price Data

### 3.2.1 Description

Fetches price data from the exchange.

### 3.2.2 Interface

The Trader must set SYMBOL and TIMEFRAME in config.py.

### 3.2.3 Functional Requirements

**3.2.3.1:** Use config.py to set the SYMBOL and TIMEFRAME.
**3.2.3.2:** This software will fetch the ohlcv data associated with the specified SYMBOL and TIMEFRAME from the exchange.
**3.2.3.3:** This software will store the fetched price data locally.

## 3.3 Perform Technical Analysis

### 3.3.1 Description

Performs technical analysis on the price data fetched from the exchange.

### 3.3.2 Interface

There is no interface associated with the technical analysis performed by this software for the standard trader with no programming experience. Currently the trading strategy is hard coded and has no customization options.

### 3.3.3 Functional Requirements

**3.3.3.1:** Apply calculations associated with various financial trading indicators on previously fetched price data to determine buy and sell signals.

**3.3.3.2:** Enter and exit positions based off the calculations performed on the price data.

## 3.4 Enter Trade Positions

### 3.4.1 Description

Opens leveraged long position when a buy signal is received or a leveraged short position when a sell signal is received.

### 3.4.2 Interface

The Trader must set AMOUNT, LEVERAGE, STOP_LOSS, and TAKE_PROFIT in config.py.

### 3.4.3 Functional Requirements

**3.4.3.1:** Use config.py to set the AMOUNT, LEVERAGE, STOP_LOSS and TAKE_PROFIT.

**3.4.3.2:** When technical analysis triggers a buy or sell signal a market order is placed to enter a position in the appropriate direction.

## 3.5 Exit Trade Positions

### 3.5.1 Description

Exits open trades from within the software.

### 3.5.2 Interface

The Trader must set SIGNAL_EXIT to True in config.py.

#### 3.5.3 Functional Requirements

**3.5.3.1:** Use config.py to set the SIGNAL_EXIT to True.
**3.5.3.2:** When technical analysis triggers a buy or sell signal in the opposite direction of an open position a market order is placed that exits that position.

# 4. Nonfunctional Requirements

## 4.1 What This Software Should Not Do

- Connect to bank account.
- Connect to Bybit account to account on another exchange.
- Transfer cryptocurrency into or out of exchange.
- Exchange one cryptocurrency for another.

## 4.2 Ease of Use / Usability

Absent a GUI, this software will require that the Trader is comfortable enough with the Linux terminal or Windows command prompt to run a python script from it. The trader must also be comfortable with using a text editor to modify the config.py file with the required settings for this software to function properly. Using this software won't necessarily be difficult, but it will require some users to step outside of their comfort zone. The provided User Documentation will guide even the most GUI dependent trader to effortlessly use this software.

## 4.3 Security Restrictions

#### 4.3.1 Bybit API

The Trader's Bybit account is protected by whatever security features they have enabled in their settings within the exchange dashboard. This software will access the exchange with an API key created by the Trader from their account. It is the Trader's responsibility to keep this API key and secret pair private while also securing the environment which will store and run this software to prevent unauthorized access. It is also suggested not to allow withdrawals using the API key.

#### 4.3.2 CCXT

This software relies heavily on the CCXT Python library for its interaction with Bybit. CCXT is open source, so the code is available to review for security vulnerabilities. None have been found as of the time of this writing, but it is a good thing to be aware of.

## 4.4 Organizational Requirements

- Programming Languages – Python 3.8.13
- Development Environment – Spyder (Anaconda3) on Windows 10
- APIs Used – Bybit API implemented through the CCXT Python library

# Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>