

Title:
Trade Tracker Discord Bot

Preface:
This document is used to document
the development plan of
the trade tracker Discord bot

Michael Mhonda

1.1 Project Overview

Executive Summary:

Updated executive summary

- This project is going to create a Discord bot that will track and store trade data generated by the owner. Discord is a messaging platform with many robust features including chat servers populated with human users and artificial programmable “bots.” The bot will run as an assistant to the main trader so people do not need to ask him what his latest trades are, they will be able to simply ask the bot and the bot will list out the most recent trades bought or sold. The bot will allow the owner full control in terms of what entries are placed inside it and also have the ability to delete any unwanted entries. All of these features will be seamlessly interwoven into the Discord platform removing the need for the users to need to interact with an outside application. All commands and functionality will be available through the chat server.

1.2 Project Deliverables

- Requirements
- SPMP
- Use Cases
- Possible Tools Needed
- Presentation with the final product

1.3 Evolution of the SPMP

- All changes should be weighed in terms of necessity before being implemented
- All changes must address pre-existing architecture to update all affected functions

1.4 Reference Materials

- Powerpoints on Blackboard from the professor

1.5 Definitions and Acronyms:

- HLA - High Level Architecture
- SPMP - Software Project Management Plan
- WBS - Work Breakdown Structure

2.1 Process model

- Discord → Amazon Web Services → Ubuntu Server → javascript script → SQL database

2.2 Organizational Structure

- The project will be built, managed, and planned from start to finish by myself. There is not a need for an organizational structure as the only person working on the project is myself.

2.3 Organizational Interfaces

- There will be relations with Amazon Web Services in order to host the bot for free with their Ubuntu server services
- Discord will house the bot

2.4 Project Responsibilities

- I will be responsible for every portion of the project from start to finish.
-

3.1 Management Objectives and Priorities

- Priorities:
 - 1. Implementing and learning proper database management
 - 2. Having sustainable connections between the host server and its Discord server
 - 3. Having input validation regarding the user commands to prevent unwanted errors and crashing
 - 4. Initially using the most effective code instead of the most efficient
 - 5. Aesthetics
- The management of this project will be to complete all the portions in a sequential order. Not to find myself attempting to do all of it at the same time, but breaking it down because there are quite a few moving parts to it. Given that there is a steep learning curve for me regarding the SQL database, I must invest in the learning portion early on.
- Budget maximum is \$10

3.2 Assumptions, Dependencies, and Constraints

- The project is assumed to be possible because of the existence of Discord and how reliable Discord is. It will be up throughout the duration of the semester.
- Dependencies
 - The project depends on the reliability of Amazon Web Services. If they prove unreliable, then a different server will be sought out.
 - When the dependencies are updated to the point that they affect the functionality or effectiveness of the bot, then there will be an audit of all current code and implementation then studying to find out how to accommodate for the differences. The bot will be checked weekly for updates, changes, and corrections regarding dependencies. This may require weekend downtime for the proper adjustments to be made, while the trading markets are closed and bot is virtually unnecessary.
- Constraints are truly just time. I will need to do some reading on SQL but that won't be much of a problem, just a step needed before the database can be fully implemented.

3.3 Risk Management

- The only risk here is that the client decides that he no longer wants his business to move in the direction that would require Discord. In which case I would still have a full project to complete, just with no client.
- The other risk is if Discord or Amazon go out of business, which is not likely to happen any time soon.
- Another risk is the complexity. I need to be able to understand SQL strongly enough to be able to manipulate and use it. This means if I am unable to learn all the functionality then my database may be lacking in some areas.

3.4 Monitoring and Controlling Mechanisms

- Monitoring and controlling involves a mixture of self-paced activities and professor meetings. The self paced activities will include deliverable deadlines and the timeline of events planned for production. The professor's monitoring involves the weekly class meetings and any deadlines or check-ins scheduled.

3.5 Staffing Plan

- I am the only staff member working on developing the program.

- There is also one client who will be telling me any fine details of how the product should run.

4.1 Methods, Tools, and Techniques

- The technique is a simple one which I will pick a specific function that can be tested in correlation with what is already built then move piece by piece in this manner.
- Technique:
 - Agile
 - This project will implement the Agile project management technique. Agile is most appropriate because it allows for changes to be made to the plan as the project progresses. There are many different parts to this project and some of the portions will not be clear as to how they will be implemented until they are used.
 - The agile framework basically uses the following steps:
 - Plan → do → improve → feedback → repeat
 - (Source: <https://www.actitime.com/project-management/project-management-tools-and-techniques/#the-waterfall-model>)
- Tools needed
 - IntelliJ community version
 - Amazon AWS
 - VMWare
 - Node.js
 - Discord.js
 - MySQL
 - Sqlite3
 - Javascript
 - SSH
 - Discord
 - WinCP
 - DB Browser
 - Visual Studio Code
 - WinCP
 - LucidChart
 - Google Drive (Docs)

4.2 Software Documentation

- The documentation will happen through Github and
- Any charts for how the software will work will be found on LucidChart

4.3 Project Support Functions

- Quality assurance will be based on how simply the commands can be executed
- The program should also be able to accept all input and properly handle it without crashing
 - This includes errors being absorbed and a dialogue of the improper use being sent back to the user so that they can change how they were interacting with the bot.

5.0 Description of Work Packages

