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:

- This project is going to create a Discord bot that will track trades made by a single person and then make the data easily accessible to all the members inside the Discord channel. The bot will be given specific commands that will trigger particular functionality within its code to either store the presented information or to retrieve already stored information. The bot may be required to perform various calculations (still under discussion) in order to tell how profitable a trade was. The bot should be able to operate by itself without needing access to its database to input/output or remove entries.

1.2 Project Deliverables

- Database for storing the information
- Bot ability to store information
    - Send the data to the database
- Bot ability to retrieve information
    - Retrieve data from the database
- Bot ability to respond to various commands from the users
- Bot ability to calculate and either store or return the results

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

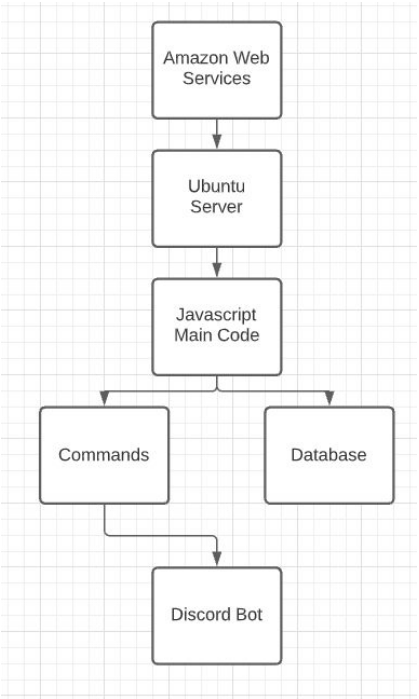
1.5 Definitions and Acronyms:

- 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



2.3 Organizational structure

- 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

- 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.
- The priorities within the requirements would be topped with the functionality of the bot. The communication between the Discord chats, the server, and database are more important than making it look nice or even extremely simple to use. First is functionality.
- Budget is $10 or less

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.
- The project depends on the reliability of Amazon Web Services. If they prove unreliable, then a different server will be sought out.
- 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 will happen weekly as modules of the project are completed. The mechanisms to test the progress are whether the particular functionality of the bot is operational or not.

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.
- 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