Software Project Management Plan

Pokémon League Manager Discord Bot

By: Andrew Fleming

Part 1 – Introduction

1.1 Project Overview:

- This project is intended to be a Discord bot that streamlines the process of managing a Discord server for a Pokémon League. The bot will allow the moderators of the server to easily set up weekly random matches, allow players to register and edit their information and statistics, as well as update leaderboards in real time based on players' wins and losses.
- Resources needed for the bot include:
    - Dedicated workstation
    - Discord bot token from their developer tools
    - Hosting service
    - Database service
    - Node modules necessary for the coding process

1.2 Project Deliverables:

- The deliverables to the customer will be the bot in its completed form, documentation on how to use the bot, and project design documentation for future development teams' use.
  The documents include:
    - Requirements
    - SPMP
    - Use Cases
    - Project Test Skeleton
    - Milestone Review
    - Presentation slides and working prototype for demonstration
    - Final product
- Included is a SQLite3 database that will store all of the player data which will be organized and maintained by the admins of the server through the associated commands.

1.3 SPMP Evolution:

- Anticipated Changes:
    - This document will be updated if more features are added to the bot to share more information with possible future developers.
- Unanticipated Changes:
    - The client will be notified of possible delays in deliverable submission dates and the document will be updated to reflect these changes.

1.4 Reference Materials:

- PowerPoints provided by Professor Broadwater.
- Documentation for the Discordjs package utilized by the project.
- Documentation for SQLite3 and the Sequelize package used to modify it.
- Documentation for Vultr hosting.
- Organization of the Discord server utilizing the finished bot.

1.5 Definitions and Acronyms:

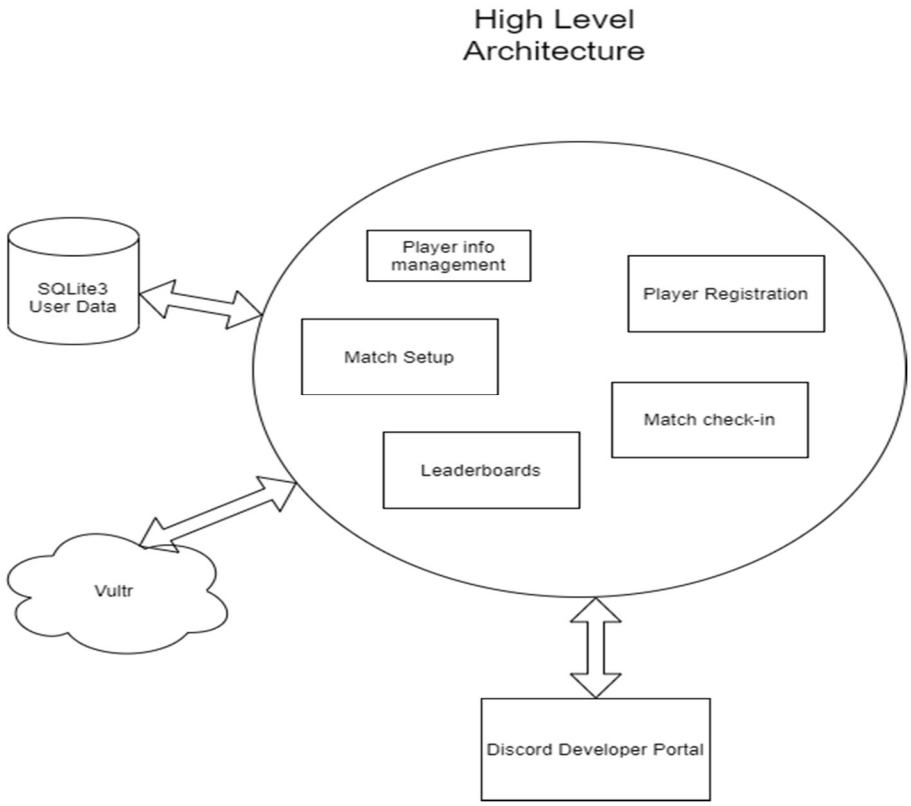
- HLA – High Level Architecture
- SPMP – Software Project Management Plan
- WBS – Work Breakdown Structure
- Discord Bot – An automated process activated by adding the bot as a member of a Discord server that allows users to access extra features not provided by the application's initial install.

Part 2 – Project Organization

2.1 Process Model:

- The project will make use of the agile development model. As I am the sole developer for the project, the only other party affected is the client. I will be prepared to deal with any unexpected circumstances by making sure to keep a constant dialogue with the client.
- The projects major milestones will be:
    - Completing the implementation of player info functionality (March 27th. 2021)
    - Presenting an early tech demo to the client (April 3rd, 2021)
    - Completing the implementation of the match scheduling and check-in functionality (early April 2021)
    - Completing the implementation of the leaderboard functionality (mid to late April 2021)
    - Creating a prototype and a powerpoint presentation to show my project to the class (May 4th, 2021)
    - Delivering the completed product as well as presenting it to the clients (late May 2021)

- The HLA visualization has been provided below:



2.2 Organization Structure:

- Since I am the sole developer of the bot, there isn't an organization structure.
- There will be a constant line of communication between the client and myself to ensure the projects' functionality meets the client's standards. There will also be a constant line of communication with Professor Broadwater.

Part 3 – Managerial Process

3.1 Project Constraints:

- Time: There is a strict due date for the project in order to obtain a grade for it.
- Skill: As I am new to developing bots, there will be extra time needed for me to become accustomed to the language regarding it.
- Schedule: As I have many other responsibilities, it may be difficult to set aside work time on this project with other responsibilities.
- Unforeseen complications with tools used.

3.2 Risk Management

- Hosting Service Failure: These could include payment failure or their servers going down. In order to prevent this, it is essential to make sure funds are correctly

transferred to the service, and to keep a backup of the software on my machine so I can run it locally if needed.

- Interfering Commitments: Another possible risk is having unforeseen responsibilities getting in the way of development time. In order to prevent this, I need to be able to pad enough time for my Agile sprints to allow for wiggle room.
- Client Abandonment: It is possible that the client for this project could suddenly cancel their request and support for the project, whether it be because of money or interest.
- Discord Issues: If Discord's servers go down, there is no environment for my project to be used on. If this happens while interfacing with the client, it could have an impact on client's interest in the project and faith in me.

Part 4 – Technical Process

4.1 Methods, Tools, and Techniques

- Operating System: Windows 10
- IDE: Microsoft Visual Studio Code
- Programming Language: JavaScript
- Pre-built Library: Discord.js, Sequelize
- Runtime: Node.js
- Data Storage: SQLite3

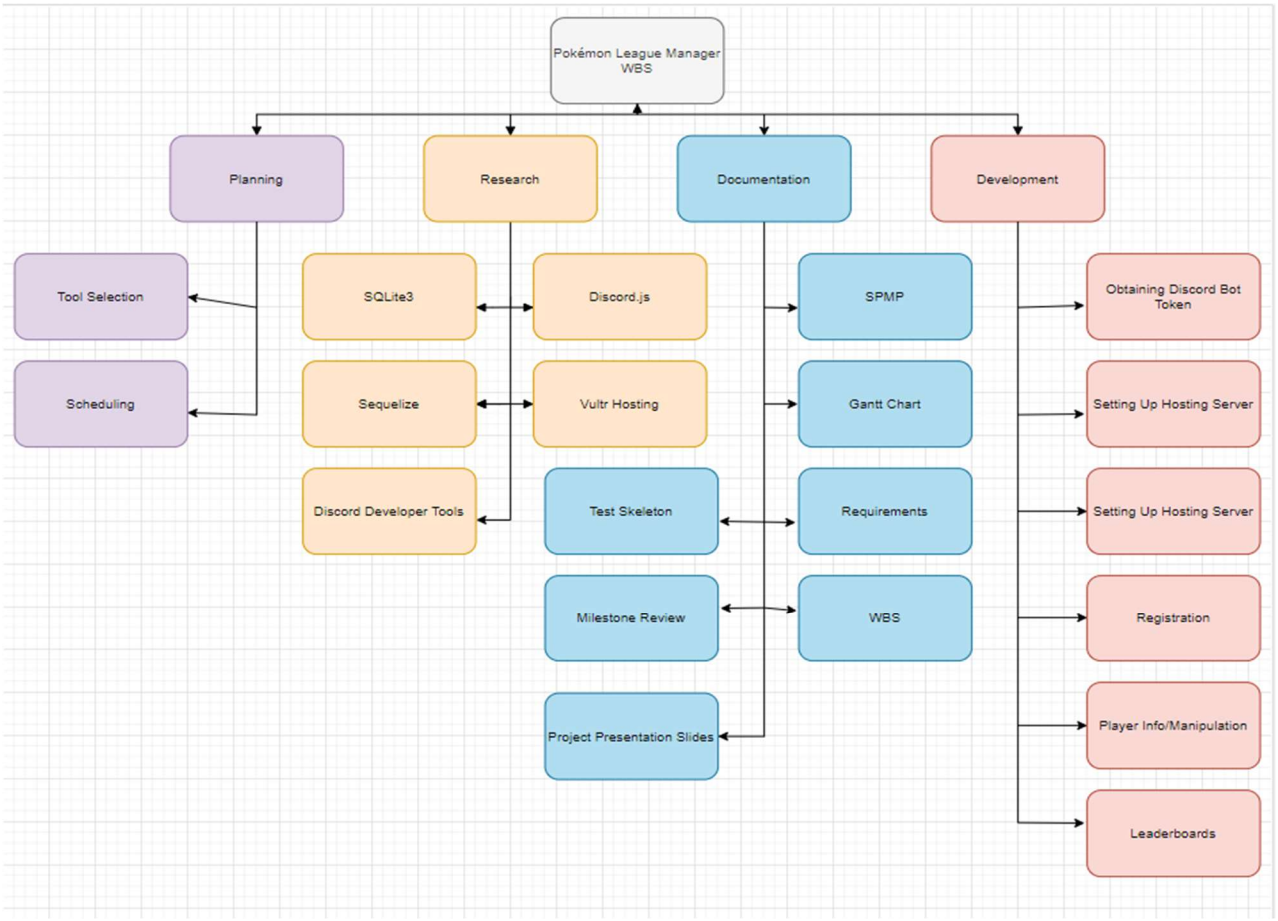
4.2 Software Documentation

- SPMP
- HLA
- Use Cases
- Interface Diagram
- Project Requirements
- Object Identification
- Case Diagrams
- Sequence Diagram
- Project Skeleton
- Test Cases

4.3 Project Support Functions

- Use Discord for communication with clients
- Use GitHub for project documentation, distribution, and tracking progress.

Part 5 – WBS


Pokémon League Manager WBS
Planning
Research
Documentation
Development
Tool Selection
Scheduling
SQLite3
Sequelize
Discord Developer Tools
Discord.js
Vultr Hosting
Test Skeleton
Milestone Review
Project Presentation Slides
SPMP
Gantt Chart
Requirements
WBS
Obtaining Discord Bot Token
Setting Up Hosting Server
Setting Up Hosting Server
Registration
Player Info/Manipulation
Leaderboards