Table of Contents

4.1 SDP Plan Introduction

This Software Development Plan outlines the process for the development of the San Jose Earthquakes Football Prediction Discord Bot which provides applications for users in the San Jose server to allow users to predict results and get points based on those predictions.

It will provide a platform for users to predict scores of football games. The bot will allow users to submit their predictions, earn points based on the accuracy of their forecasts, and engage with other users.

The application will notify registered users about upcoming games, provide game notifications, display leaderboards, and offer user engagement features such as score viewing, squad details, and upcoming match information. The rationale behind this development is to foster community engagement and excitement among fans of the San Jose Earthquakes.

4.1.1 Project Deliverables

- Get basic commands working, i.e, users are able to type in the commands and the bot will repeat what the user typed or will provide some sort of information
    - !predict “score1-score2” “first-scorer” : users will try and predict the score of the match and the first goals scorer of the SJ team
        - Bot will just spit back what the user will say in text
        - Will later add decorations for the bot response and saving that info in a database
        - End Date: 02/25
    - !predict top : users can see the top predictors within the server (ideally a top 10) which will show username, how many times they have predicated, and total points
        - For this command, will first give a dummy table and then add the correct functionality by showing correct users, appearances, and points
        - End date: 02/25
- Connect API with Discord.js
    - Be able to get specific information from the API and let the bot spit out the information

        - For example, a command can be called !match-information which gives information on the next match, what time it will start, the two teams playing, and maybe the stadium name (indicating home or away)
            - Another command can be a !team-information giving information on players and seeing which players are healthy or currently injured or maybe suspended if applicable
            - End date: 03/02
- Start implementation of database
    - Decide between Sequelize and Postgres, will depend which one is easier to implement with Discord.js
    - End date: 03/12
- Connect API to Database (save results)
    - Get matches before they are played (so before the score and any goal information)
    - Get results of matches and put it into database, if the match was between San Jose vs Seattle and the score was 3-2, give the 3 to San Jose and Seattle to 2 and specify that San Jose won the game (also first scorer information)
    - End date: 03/19
- Connect Users Prediction score to Database
    - If a user predicated 2-1 for the San Jose and Seattle score using the command, save that score, but if they use the command again and update the score to 3-1, override previous score with the most recent one (in this case being 3-1) and also applies to goal scorer
    - End date: 03/26
- Implement scoring system
    - Based on predictions from users and actual game results, tally up the points and give it to the user
    - End date: 04/02
- **If there is time to implement**
    - Create a command for the bot to also predict scores and/or goal scorer, so it can play along as well
        - Will add some sort of Machine Learning and users can also make a decision to trust it or not
    - Can create a command that gives fun facts about the history of the team and the history of the current opponent
    - End date: TBD

4.2 Project Resources

Resources are vital elements that contribute to the successful completion of the project. In the context of project management, resources encompass not only physical assets like hardware and software but also human resources—the individuals or teams responsible for various aspects of the project. The effective management of these resources is crucial for meeting project objectives and milestones. Below are detailed descriptions of the different resources that

will be utilized during the development of the San Jose Earthquakes Football Prediction Discord Bot:

4.2.1 Hardware Resources

- Development Machine (Jose):
    - Processor: 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz 3.00 GHz
    - RAM: 16.0 GB
    - Storage: 256 GB
- Development Machine (Ed):
    - Processor: Apple M1 Pro chip
    - RAM: 16.0 GB
    - Storage: 512 GB
- Development Machine (Henry):
    - Processor: Apple M1 chip
    - RAM: 16.0 GB
    - Storage: 512 GB
- Hosting Server (if applicable):
    - We will definitely use a hosting service for the final model, however at this moment in time we have no specific platform we are using.

4.2.2 Software Resources

- Operating Systems
    - Mac OS Ventura Version 13.0
    - Windows 10
    - macOS Sonoma Version 14.3.1
        - Note - operating system won't matter for our project
- Discord API
    - At least version 9
- Discord JS Library - version 14
- Football Game Data API - api that grabs data from all across the globe
    - Will only hone in on one specific league and team (MLS and San Jose Earthquakes)
    - At least version 3.9
- Node.js - at least version 18
- Sequelize or PostgreSQL - again, dependent on which one is easier to integrate with Discord.js (or maybe even use both)
- Text Editor - Visual Studio Code
- Git and Github (Version Control)

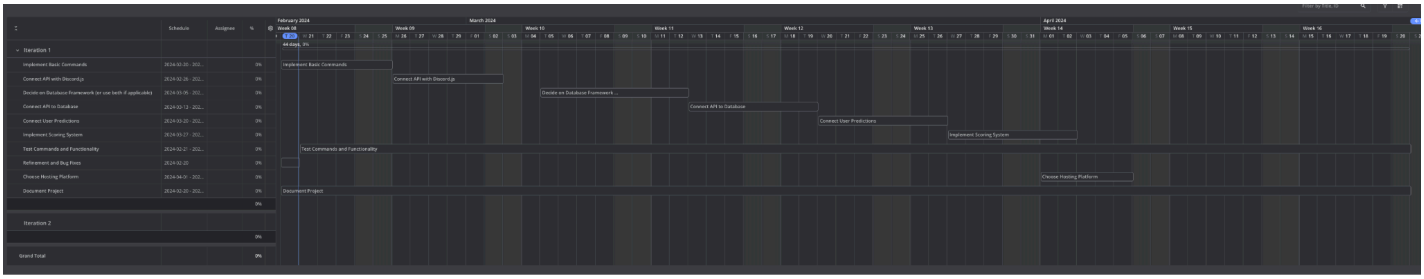
4.3 Project Organization/Human Resources

- The project will be organized into the following major functions:
    - Development: Responsible for coding, testing, and debugging the Discord bot.
        - Will be team responsibility, more details for specific roles later in the section

    - Testing: will mostly just be testing out the commands on our own, thinking of any edge cases, making sure commands are properly used and set, not being mixed in with anything else
        - Will also gather feedback from other users outside of the dev group
    - Deployment: Responsible for deploying the bot onto a hosting platform and providing documentation for deployment.
        - This will mostly be a team responsibility as we research and find out with hosting platform best serves our needs but will designate a leader for this particular category to spearhead planning and research
- Specific Roles for each member
    - Jose - Implement Basic Commands, Connect API to Database, Implement Scoring System,
    - Ed - Connect User Predictions,
    - Henry - Connect API with Discord.js, Refinement and Bug Fixes
    - Everyone - Decide on Database Framework (or use both if applicable), Test Commands and Functionality, Document Project, Choose Hosting Platform

4.4 Schedule

This section provides schedule information for the San Jose Earthquakes Football Prediction Discord Bot project.



4.4.1 Task / Resource Table

Task	**Who Working on it**	**Resources being used (both software and hardware)**
Implement Basic Commands	Jose	Discord API, Discord.js Library, Node.js, VS Code
Connect API with Discord.js	Henry	Discord API, Discord.js Library, VS Code

Decide on Database Framework (or use both if applicable)	Everyone	Discord.js Library, Sequelize or PostgreSQL, VS Code
Connect API to Database	Jose	Discord.js Library, Sequelize or PostgreSQL, Football API, VS Code
Connect User Predictions	Edward	Discord.js Library, Sequelize or PostgreSQL, VS Code
Implement Scoring System	Jose	Discord.js Library, Sequelize or PostgreSQL, VS Code
Test Commands and Functionality	Everyone, test on there own time and getting feedback from	Development Machines, Discord Server for Testing, VS Code
Refinement and Bug Fixes	Henry	Visual Studio Code, git for version control, VS Code
Choose Hosting Platform	Everyone	Hosting Server, Deployment Documentation, VS Code
Document Project	Everyone	Markdown, GitHub, VS Code