



Software Requirements Specification

For

Discord Chess Bot (Chessinator)

Version 2.2 approved

Prepared by Dylan Crockett and Zinedin Bautista

University of Advancing Technology

November 15, 2019

1o Table of Contents

1. Table of Contents.....	1
2. Revision History.....	2
3. Introduction.....	3
3.1 Purpose.....	3
3.2 Tools used to Develop Project.....	3
3.3 Intended Audience and Reading Suggestions.....	3
3.4 Product Scope.....	3
3.5 References.....	4
4. Overall Description.....	5
4.1 Product Perspective.....	5
4.1.1 Sequence Diagram.....	6
4.1.2 Use Case Flow Diagram #1.....	7
4.1.3 Use Case Flow Diagram #2.....	8
4.1.4 UX Diagram.....	9
4.2 Main Product Functions.....	9
4.3 Extra Product Features.....	10
4.4 User Classes and Characteristics Diagram	10
4.5 Operating Environment.....	10
4.6 Design and Implementing Constraints.....	11
4.7 User Documentation.....	11
4.8 Assumptions and Dependencies.....	11
5. External Interface Requirements.....	12
5.1 User Interfaces.....	12
5.2 Hardware Interfaces.....	12
5.3 Software Interfaces.....	12
5.4 Communications Interfaces.....	12
6. System Features.....	12

6.1 System Feature 1.....	12
6.2 System Feature 2.....	13
6.3 System Feature 3.....	14
6.4 System Feature 4.....	15
6.4 Sytem Feature 5.....	15
7. Other Nonfunctional Requirements	16
7.1 Performance Requirements.....	16
7.2 Safety/Security Requirements.....	16
7.3 Software Quality Attributes.....	16

2o Revision History

Revision History:

Name:	Date:	Reason For Changes:	Version:
Users allowed to edit events.	10/18/2019	Issue with Google authentication. Which is causing a data issue and time concern.	1.0
Change Discord bot from Calendar Discord bot to Chess Discord bot.	11/8/2019	Took longer than expected for Google to authorize our webpage service.	2.1

3. Introduction

3.1 Purpose

The purpose of this project is to create a Discord bot that will play chess against the user or have the users play chess against each other.

3.2 Tools used to Develop Project


- Python 3.7
- Flask
- PyCharm
- Github
- Discord API

3.3 Intended Audience and Reading Suggestions

The intended audience can be any person who has a Discord account, but this is more focused on active Discord users who would like a Discord bot that will play chess with them or if they want to play chess with another person.

3.4 Product Scope

Discord is a freeware VoIP and messaging service that many people use to organize teams, events, and other groups as well as for one on one conversation. Discord allows for the use of third party agents called bots, which users can use for a variety of tasks or applications.



The goal for this project is to create a chess bot, that allows the users to challenge each other in the logical game of chess.

Not only should the bot agent allow the users to set up a game of chess, but the bot should also have a help feature to help users navigate and manage the chess bot. Ideally, the actual game of chess should appear on a different website where the users can interact with chess pieces by clicking and dragging the chess pieces.

This Discord bot will be named the Chessinator or just Chess Bot. The Discord bot project will be programmed with Python using PyCharm as the integrated development environment. Also, throughout the project, we will be using Github as our source of tracking and distribution version-control system. We will also be using Flask, which is a micro web framework written in Python. The reason for using Flask is to develop a micro web framework where the actual game of chess will be played, which will mean that a different website will open, which will contain the chessboard and chess pieces. This will also allow the players to click and drag each chess piece through the game board.

3.5 References

- <https://synchronic.uat.edu/courses/2603/files?preview=345479>
- <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>

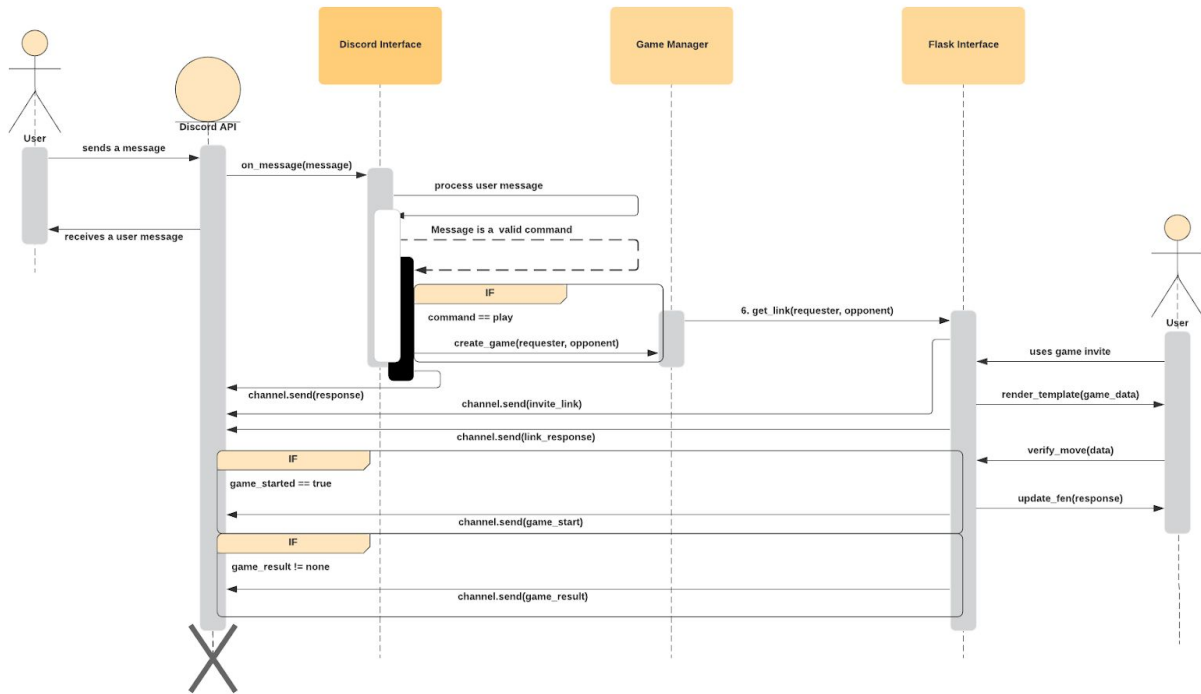
4 Overall Description

4.1 Product Perspectives

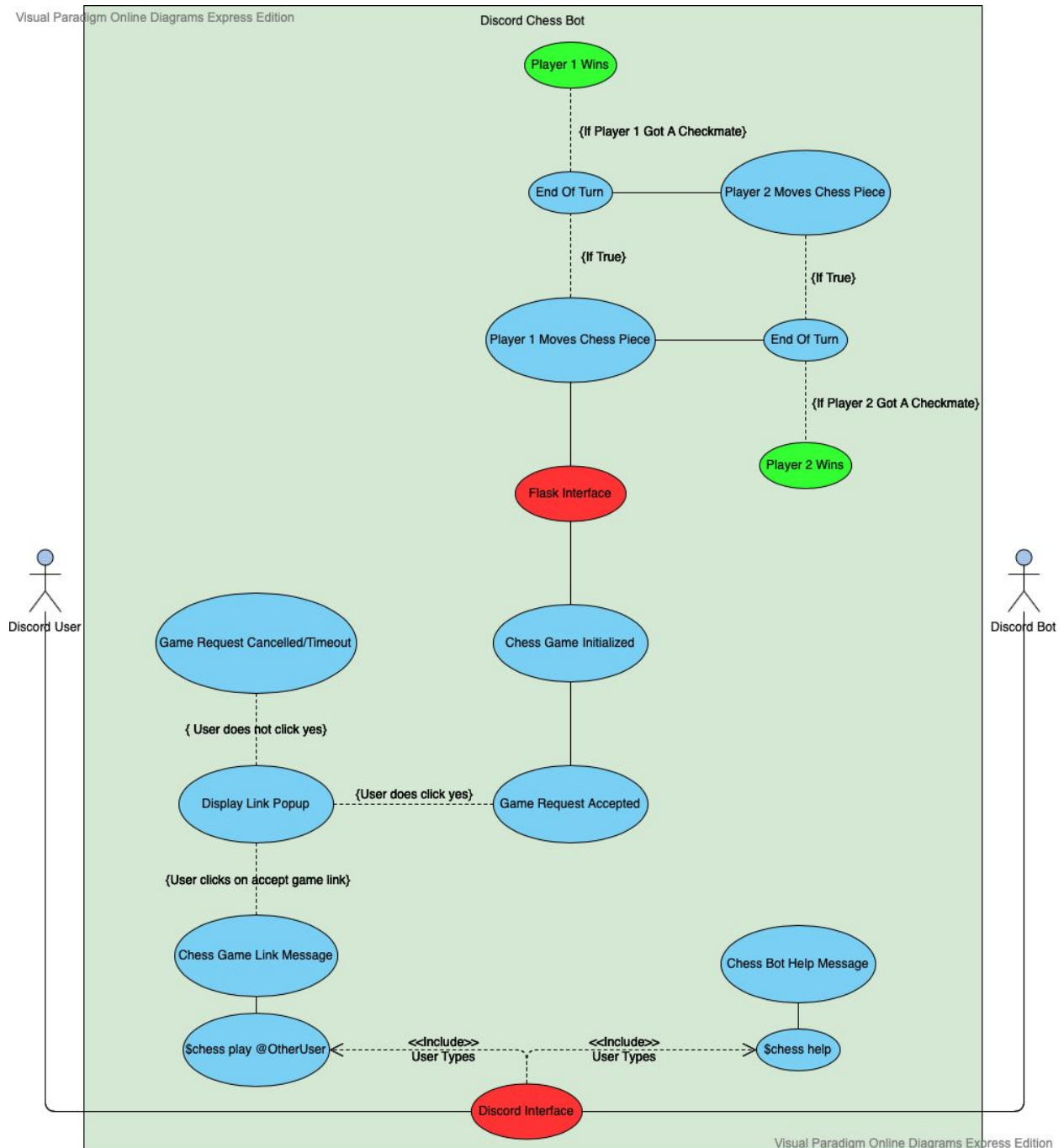
The Discord bot will be able to...

- Help users navigate through the bot by having a help command, which will inform the user on how to run the Discord bot.
- Allow the user to challenge another Discord user to play chess by sending the challenging player an invite through a Discord message and a link to the chess game.
- Display the chess game within a website.
- Announce the winner of the chess game or announce if the chess game ended in a draw.
- Create a spectator link where other people can watch the chess match.
- Create a replay link for the chess game.

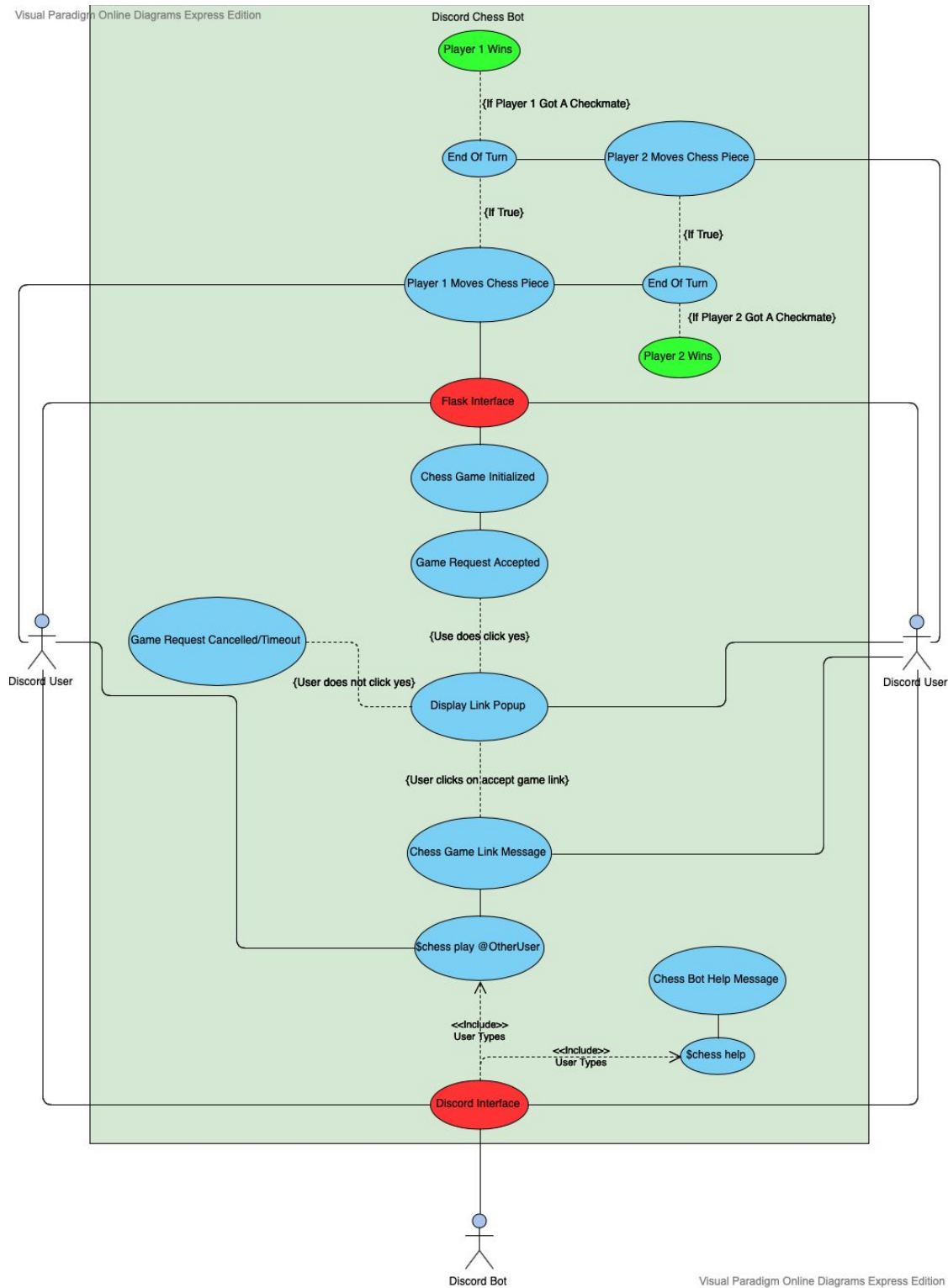
4.1.1 Sequence Diagram:



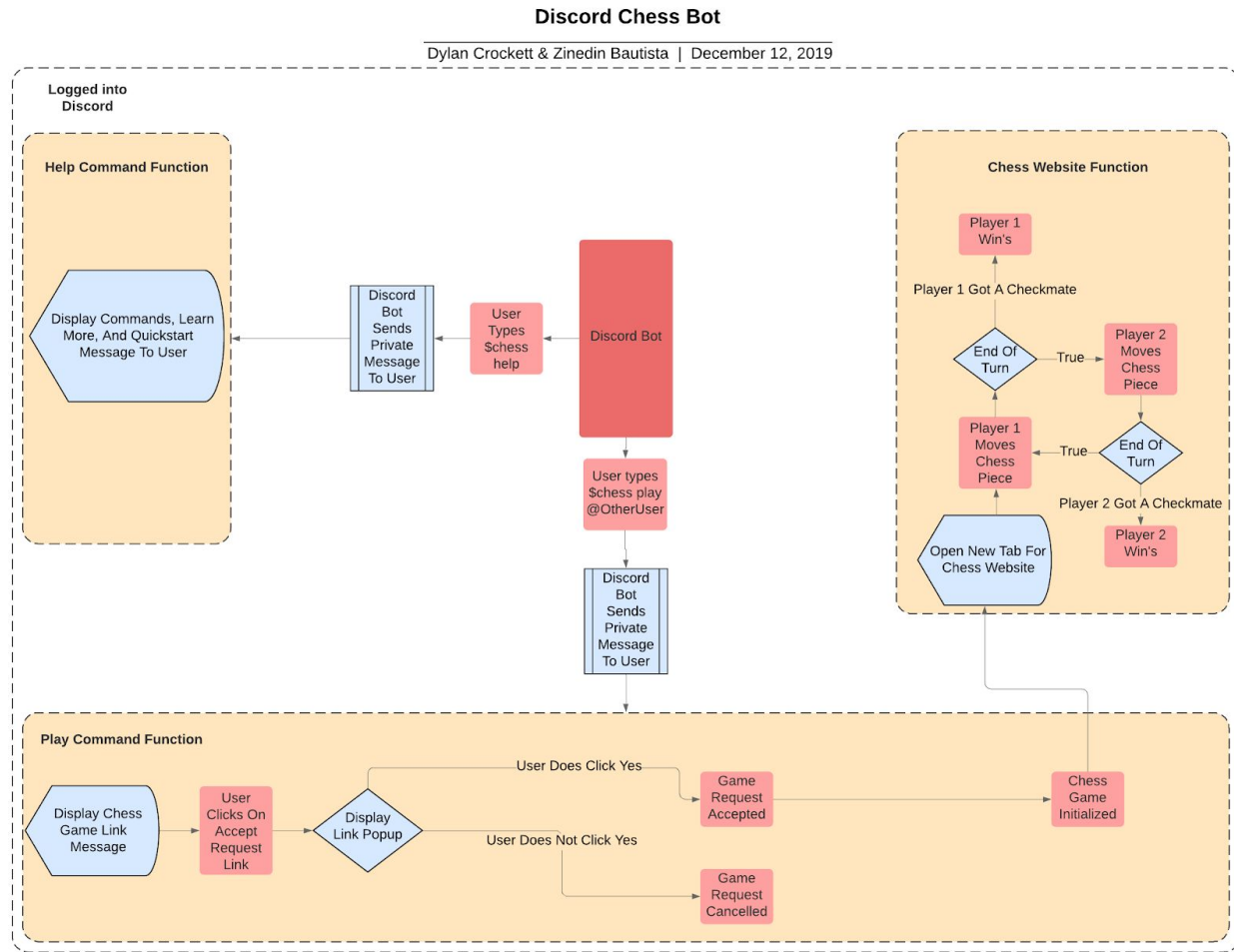
4.1.2 Use Case Flow Diagram #1:



4.1.3 Use Case Flow Diagram #2:



4.1.3 UX Diagram:



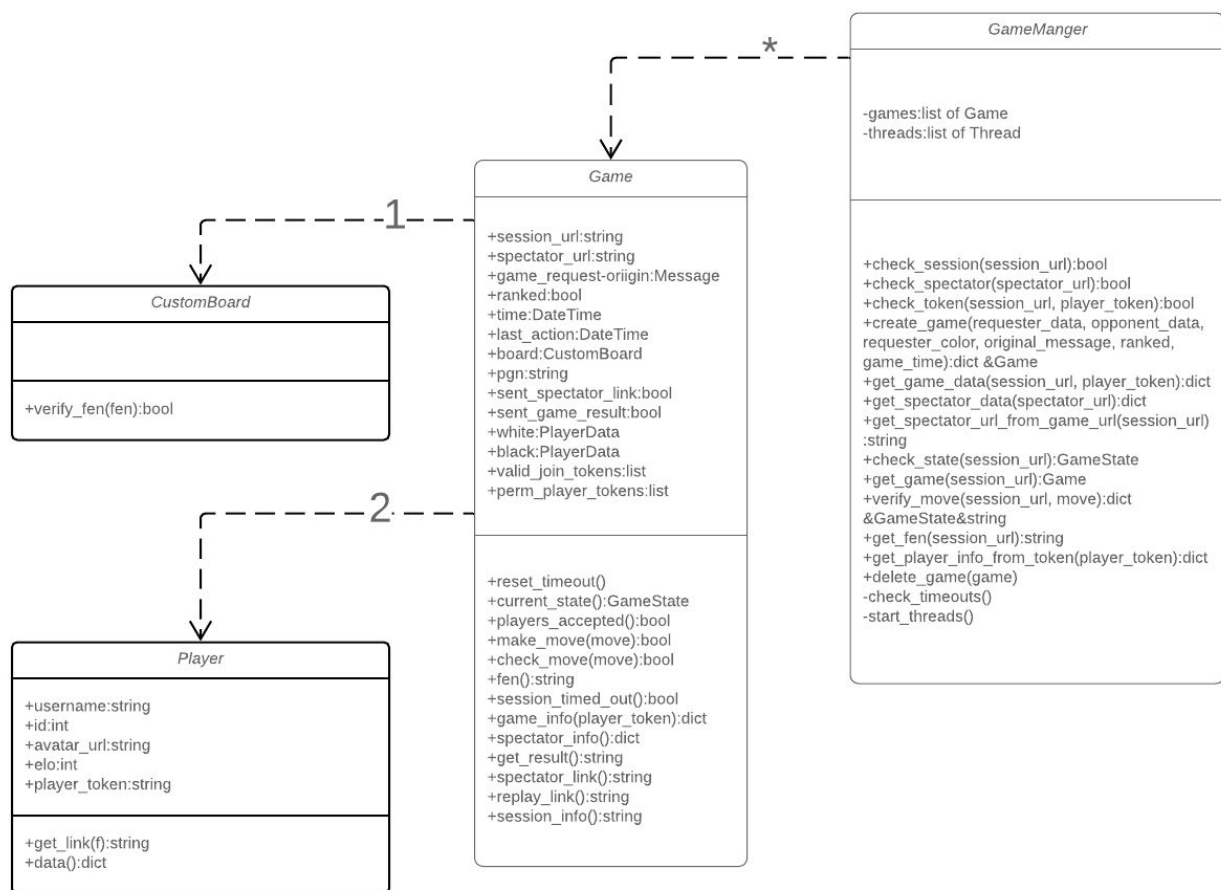
4.2 Main Product Features

1. Have a help command which shows you how to use the chess bot.
2. The user can challenge another user in a game of chess by having the chess bot send an invite message with a link towards the other challenger and a link to play the game.
3. Display the chess game within a website.

4.3 Extra Product Features

1. Create a spectator link so that other people can spectate the chess match.
2. Create a replay link of the chess game.

4.4 User Classes and Characteristics Diagram



4.5 Operating Environment

The Discord bot will be operating on Discord version 0.0.9 and can work on any hardware platform that runs Windows, macOS, Linux, iOS, Android, and any web browsers.

4.6 Design and Implementation Constraints

Timing Constraint: Since we decided to change the Discord bot from a calendar Discord bot to a chess Discord bot, around the end of the second sprint, we only have the third and final sprint to remodify the development of the bot, and we also have to remodify almost all of the documentation as they are deemed outdated.

Server Constraint: Since the chess bot will open a game of chess by opening another window, we have to make sure that the host server is up and running.

4.7 User Documentation

The Discord bot will have a help protocol that will display how it functions with such details as to how the users should type out the commands, a learn more section, and a typed part on how to quickly startup the chess bot.

4.8 Assumptions and Dependencies

One assumption that we have in the development cycle of this project is that since the chess Discord bot will display the chess game by opening another website displaying the chess game. We are assuming that by the end of the project, the web-hosted server will be open and ready for usage.

5 External Interface Requirements

5.1 User Interfaces

- Discord Interface
- Flask Interface

5.2 Hardware Interfaces

- Any device that supports Windows, macOS, Linux, iOS, Android, and web browsers.

5.3 Software Interfaces

- Discord application

5.4 Communications Interfaces

- Discord account
- Discord server
- Flask interface

6 System Features

6.1 System Feature 1

- Description: By typing \$chess help, the chess bot will send you a message which will display a message informing the user on the commands, where you can learn more, and how to activate the bot.
- Stimulus/Response Sequences:

1. The chess bot must be in the desired server. Once that's done, the user will type \$chess help and send that message.
 2. The chess bot will send the user a message which will be notified by Discord.
 3. Once the user clicks on the chess bot icon, the Discord application will direct the user to the conversation history between the chess bot and the user.
 4. The chess bot will display a helpful message.
- Functional Requirements:
 - Users must have the Discord app with their accounts.
 - Users must put type \$chess help precisely as it's displayed here in this document.

6.2 System Feature 2

- Description: By typing \$chess play @OtherUser, the Discord chess bot will send the other challenger a message. That message will be an accept game request link. The user must click on the accept request link then click on the play game link.
- Stimulus/Response Sequences:
 1. User types \$chess play @OtherUser.
 2. The other challenger will receive a Discord message which will be an accept game request link.
 3. The other challenger must click the accept game request link.
 4. The chess game will initialize and begin.
- Functional Requirements:
 - Users must have the Discord app with their accounts.

- The user must type \$chess play @OtherUser precisely as it's displayed here in the document.
- The other challenger must click the accept the requested link to start the game of chess.
- Then they must click on the play game link that the chess bot provided.

6.3 System Feature 3

- Description: Once the game of chess is being initialized, a different website will appear, which will contain the actual game of chess. Also when a user gets a checkmate against another player a pop-up will display announcing the winner. Or if the game of chess ends with a draw then a pop-up will display announcing that the game of chess ended in a draw.
- Stimulus/Response Sequences:
 1. Once both players are ready to play chess, another website will appear to begin the game of chess.
 2. For the pop-up announcing the winner to show up one of the users must achieve a checkmate.
 3. For the pop-up announcing a tie both players must put the chess match in a scenario where both players cannot win nor lose.
- Functional Requirements:
 - Users must have the Discord app with their accounts.
 - Both players must be ready, which will be indicated on whether the proper links were clicked and accepted.

- Either one player must win the game of chess or it can also tie for a popup to appear.

6.4 System Feature 4

- Description: Once a game of chess between two users begins. A spectator link will be available so that anyone within the Discord server can join and view the ongoing chess match.
- Stimulus/Response Sequences:
 1. A game of chess must be initialized and ongoing by two users for the spectating link to be sent out to the server.
- Functional Requirements:
 - Everyone must be logged onto their Discord account.
 - A game of chess has to be initialized between two other users.

6.5 System Feature 5

- Description: Once a game of chess between two users has ended no matter the circumstance than a replay link of the chess game will be sent out.
- Stimulus/Response Sequences:
 1. A game of chess has to be declared as finished.
- Functional Requirements:
 - Users must be logged into their Discord account.
 - A game of chess has is declared as finished.
 - A replay link is sent.

7% Other Nonfunctional Requirements

7.1 Performance Requirements


The requirement for the performance of the Discord bot is the user requires a computer that can process the Discord application, which should be a majority if not almost all computers connected to the internet. The user must have the discord application downloaded and have an account. Once the user and or server has the Discord bot uploaded to their desired location, the bot will have a help command, which will display how the users should type out the commands. It will also have a learn more section and a section dedicated to how to run the chess bot. Also, since the chess game will be played through a new pop-up window, that means that the user(s) computer will have to be able to run Discord and another webpage, which, as mentioned before, should be a majority if not almost all computers. Since the chess bot is only available through the Discord application, the user must have a valid Discord account.

7.2 Safety/Security Requirements

Because the only way someone can use the Discord chess bot is, they have a Discord account, which means that those who are using the Discord chess bot must keep their Discord account credentials protected.

7.3 Software Quality Attributes

Availability: The Discord chess bot can be used on any device that has the Discord application downloaded.



Reliability: Since the chess game will be opened through another window, which is run by a hosted server, virtually, the chess bot will be reliable only if the hosted server gives you access and is open for usage.

Usability: The usability of the Discord bot will be focused on being easy to manage and navigate for the user. The Discord bot will have a help function to help guide users on how to use the chess Discord bot. The Discord bot will also allow users to play chess with either the bot or against another person.