De La Salle University- Manila
Gokongwei College of Engineering

PROLOGI
Programming Logic and Design

Project Proposal

Chess Console: A Simple Chess Game with Python Arrays

Harold Sebastian Blas
Irabelle Pristine Y. Tan
Josh Angelo Ty

I. Project Description

This project aims to allow people to play the game of chess even when a board is not readily available. This program will be made using Python, and will allow users to have an account system where they can tally their wins and losses, which will also be shown on a leaderboard. Moreover, there will be a timer that runs when waiting for an input, and standard chess moves such as castling and en passant will be possible.

The objective of this program is to allow people to play either chess or checkers in convenience as it is not required to have a physical chess set. Moreover, it aims to allow players to have a record of their wins and losses without having to record it themselves since the system would do it for them already.

II. IPO

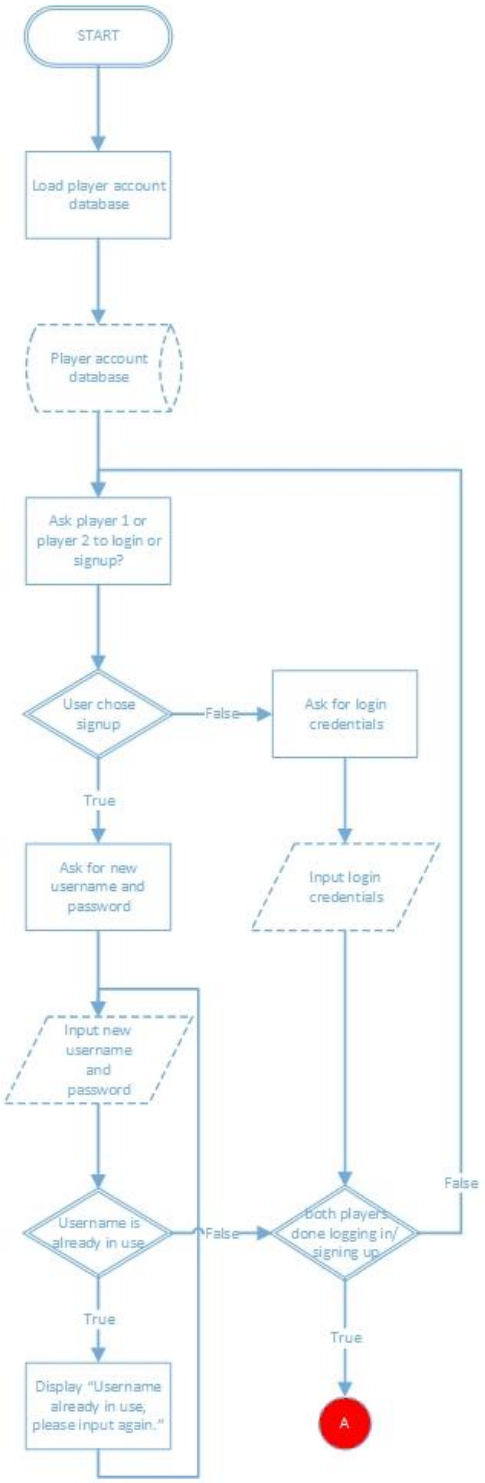
Input -
Player 1 username & password (login or signup);
Player 2 username & password (login or signup);
Player 1 move using chess notation (repeat input while game has not ended or if input is invalid);
Player 2 move using chess notation (repeat input while game has not ended or if input is invalid);
Draw or Resign offers

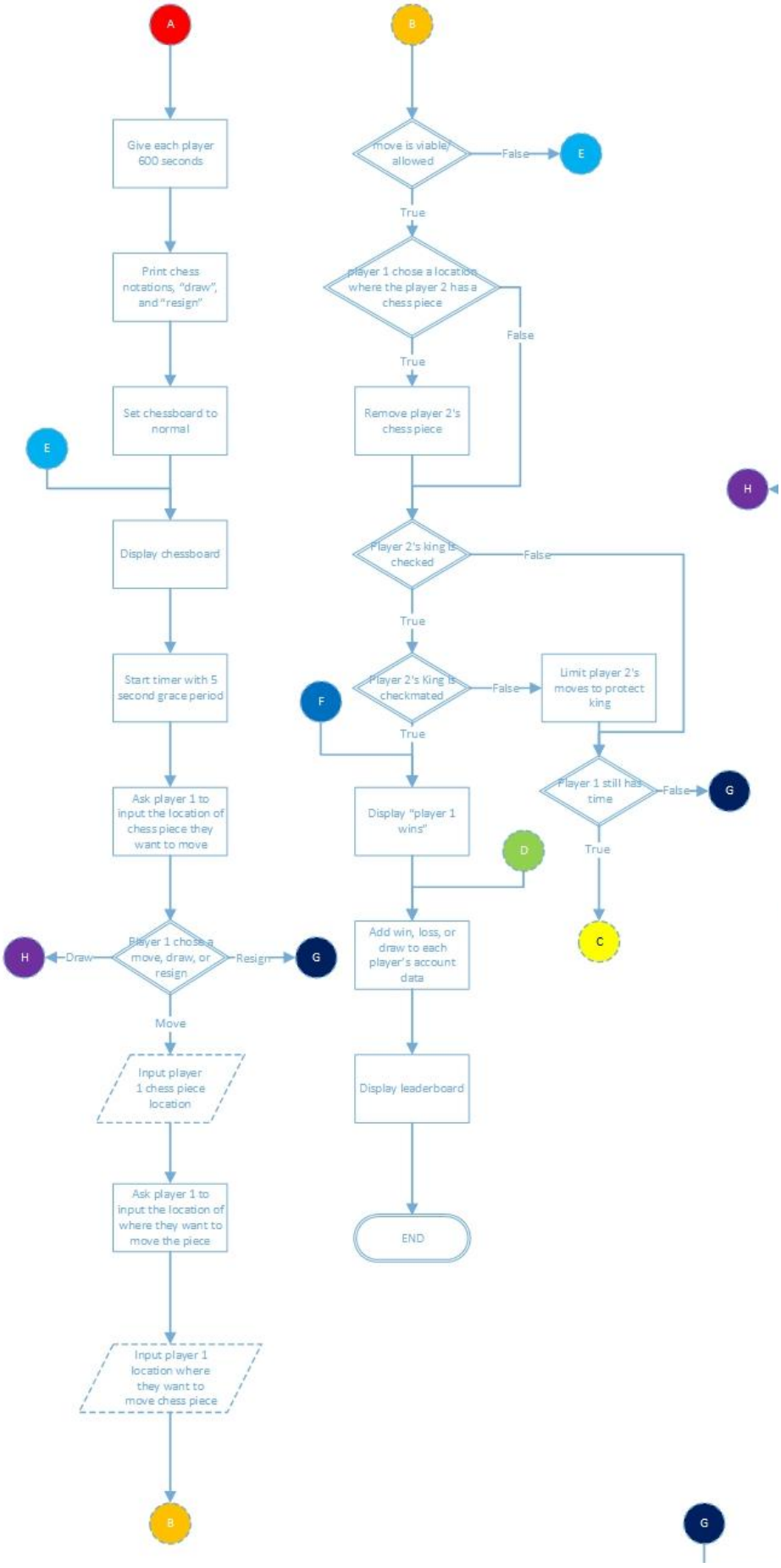
Process -
If Player is logging in and information aligns with an account in the account database, then show stats;
If Player is logging in and information does not align with an account in the account database, then ask for input again;
If Player is signing up and username does not align with an account in the account database, then show stats;
If Player is signing up and username aligns with an account in the account database, then ask for input again;
While game has not ended(A Player's time has not reached 0, both kings on the board):
{
Countdown time (initial time of 10 minutes, counts down per second while there is no input, 5 second grace period before start of countdown);
Ask Player for move input using chess notation (may also draw or resign);
If draw: Ask both Players if they want to draw, if yes then end game and add draw to their stats, if no then continue game;
If resign: add loss to resigning Player's stats and add win to other Player's stats;
If move input is not valid: Ask input until move input is valid;
If move input is valid: do chess move;
Update chessboard with Player move inputs;
Update time;
Check if any game end scenarios have been reached, if yes then end game, if no then continue;
}(Cycled between the 2 Players)

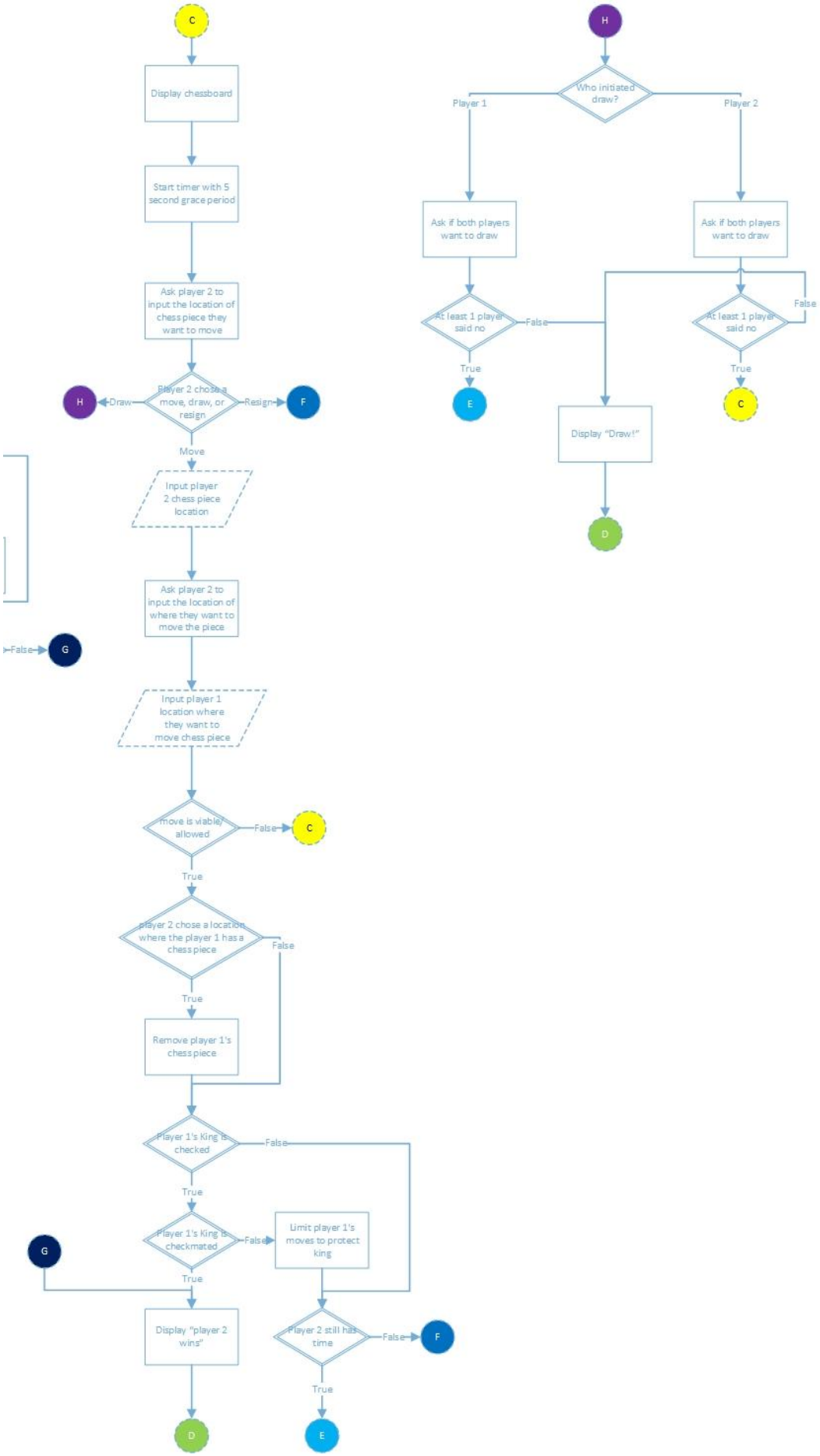
Output -
Player stats;
List of possible moves;
Current time per player (updates and prints every second (replacing));
Current chessboard (updates and prints after every move);
Winner of chess game;
Leaderboard according to account database wins;

III. Methodology

https://drive.google.com/file/d/1RkQG4a4fQ6U3a5lg2sQADHq2hD32fHpk/view?usp=share_link




A
Give each player 600 seconds
Print chess notations, "draw", and "resign"
Set chessboard to normal
E
Display chessboard
Start timer with 5 second grace period
Ask player 1 to input the location of chess piece they want to move
Player 1 chose a move, draw, or resign
H
Draw
Resign
G
Move
Input player 1 chess piece location
Ask player 1 to input the location of where they want to move the piece
Input player 1 location where they want to move chess piece
B
B
move is viable/ allowed
False
E
True
player 1 chose a location where the player 2 has a chess piece
False
True
Remove player 2's chess piece
H
Player 2's king is checked
False
True
Player 2's King is checkmated
False
Limit player 2's moves to protect king
F
True
Player 1 still has time
False
G
True
C
Display "player 1 wins"
D
Add win, loss, or draw to each player's account data
Display leaderboard
END
G

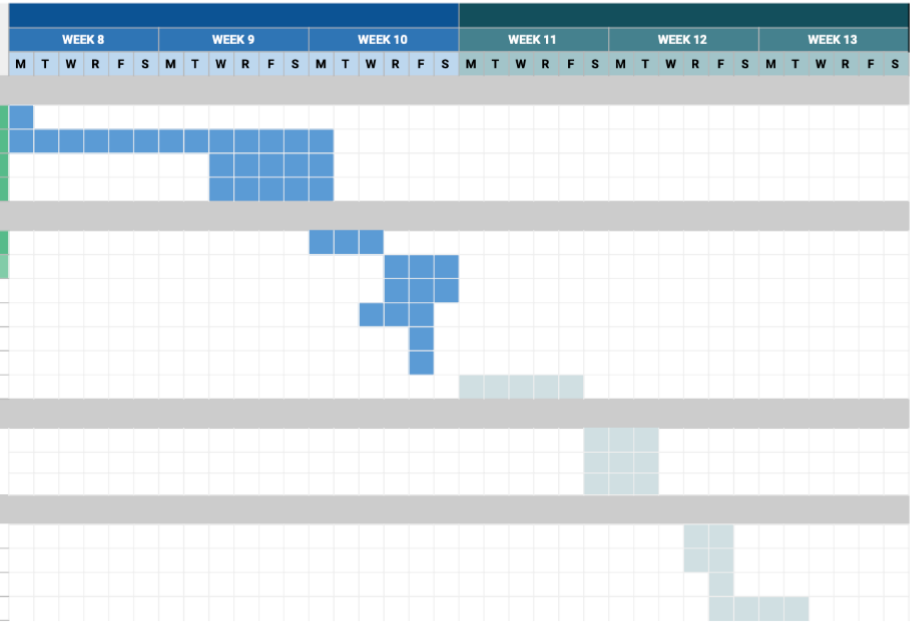

C
Display chessboard
Start timer with 5 second grace period
Ask player 2 to input the location of chess piece they want to move
Player 2 chose a move, draw, or resign
H
Draw
Resign
F
Move
Input player 2 chess piece location
Ask player 2 to input the location of where they want to move the piece
False
G
Input player 1 location where they want to move chess piece
move is viable/ allowed
False
C
True
player 2 chose a location where the player 1 has a chess piece
False
True
Remove player 1's chess piece
Player 1's King is checked
False
True
Player 1's King is checkmated
False
Limit player 1's moves to protect king
G
True
Display "player 2 wins"
Player 2 still has time
False
F
True
D
E
H
Who initiated draw?
Player 1
Player 2
Ask if both players want to draw
Ask if both players want to draw
At least 1 player said no
False
At least 1 player said no
False
True
True
E
C
Display "Draw!"
D

IV. Schedule of Activities

Team Dionysus

PROJECT TITLE	Chess Console: A Simple Chess Game with Python Arrays
MEMBERS	Blas, Harold, Tan, Irabelle, Ty, Josh
DATE	3/20/23

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Conception and Proposal**					
1.1	Initial Project Proposal	Everyone	3/6/23	3/6/23	0	100%
1.1.1	Project Description	Irabelle T.	3/6/23	3/20/23	14	100%
1.2	IPO Model	Josh T.	3/15/23	3/20/23	6	100%
1.3	Methodology	Harold B.	3/15/23	3/20/23	6	100%
2	**Project Initiation**					
2.1	Base Design	Josh T.	3/20/23	3/23/23	4	100%
2.2	Designing Chessboard	Josh T.	3/23/23	3/25/23	3	75%
2.3	Designing Account System	Irabelle T.	3/23/23	3/25/23	3	0%
2.4	Designing Time and Input Thread	Harold B.	3/23/23	3/25/23	3	0%
2.5	Designing of Chess Moves	Josh T.	3/25/23	3/26/23	2	0%
2.6	Desiging of Leaderboard	Irabelle T.	3/25/23	3/26/23	2	0%
2.7	Designing of Checkers	Harold B.	3/26/23	3/31/23	6	0%
3	**Project Documentation**					
3.1	Methodology	Ty, Tan	4/1/23	4/3/23	3	20%
3.2	Conclusion	Harold B.	4/1/23	4/3/23	3	0%
3.3	Finalization	All	4/1/23	4/3/23	3	0%
4	**Project Demonstration**					
4.1	Script/Content Creation	Josh T.	4/6/23	4/7/23	2	0%
4.2	Presentation Creation	Irabelle T.	4/6/23	4/7/23	2	0%
4.3	Recording of Video	All	4/7/23	4/7/23	1	0%
4.4	Editing of Video	Harold B.	4/7/23	4/11/23	5	0%



V. References

Possible References:

Python. (2023a). *queue — A synchronized queue class*. docs.python.org. Retrieved from https://docs.python.org/3/library/queue.html.

Python. (2023b). *threading — Thread-based parallelism*. docs.python.org. Retrieved from https://docs.python.org/3/library/threading.html.

Python. (2023c). *time — Time access and conversions*. docs.python.org. Retrieved from https://docs.python.org/3/library/time.html.

Vallance, L. (2018, April 19). *Exploring the Python Chess Module.* liamvallance.com. Retrieved from http://liamvallance.com/img/Exploring%20python%20chess.pdf.