



# **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));

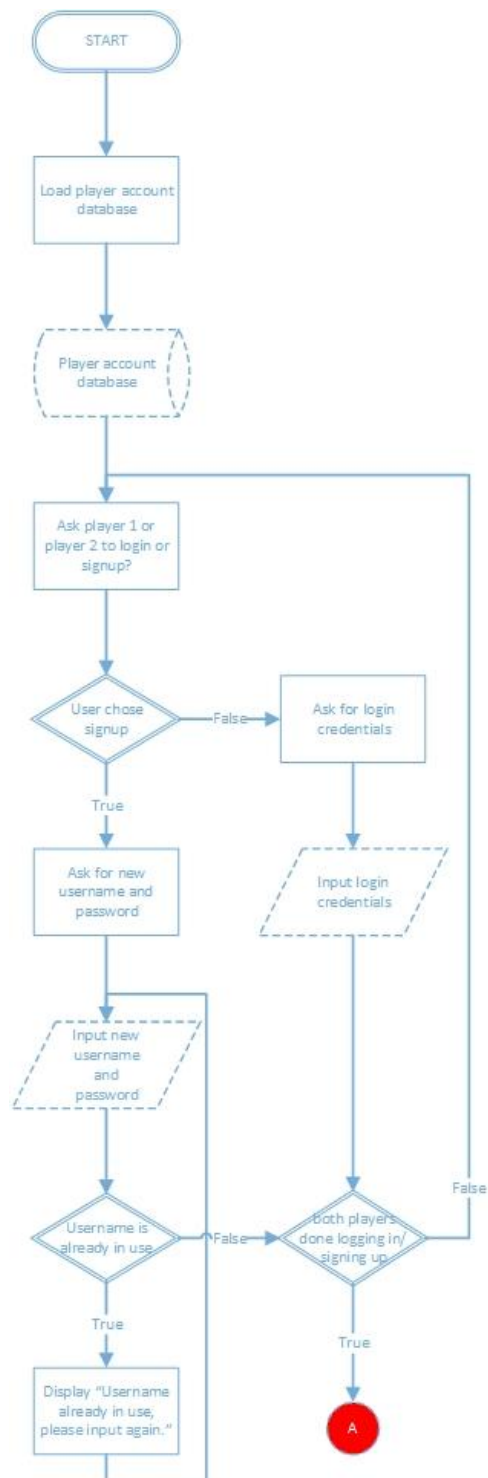
Current chessboard (updates and prints after every move);

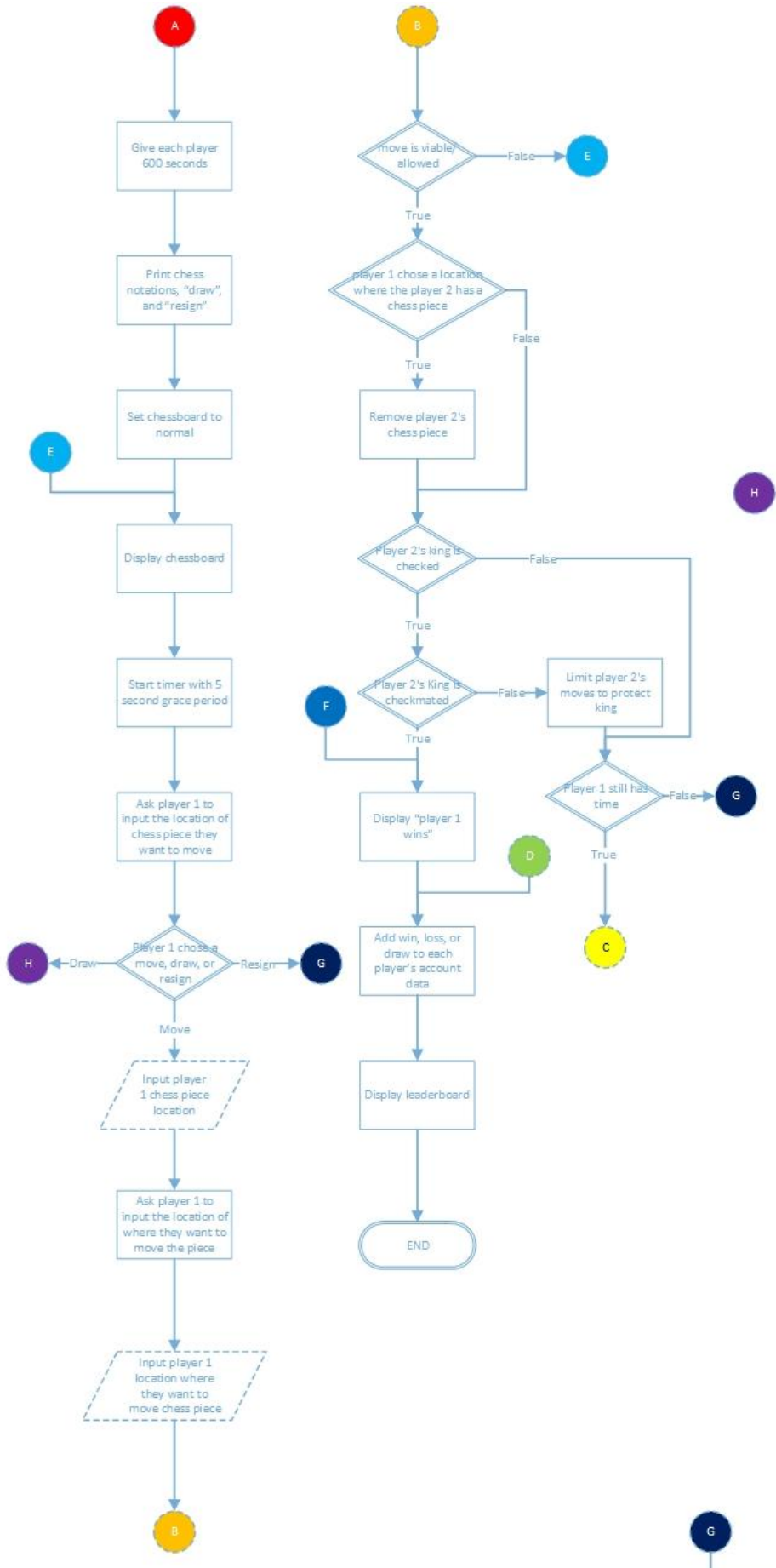
Winner of chess game;

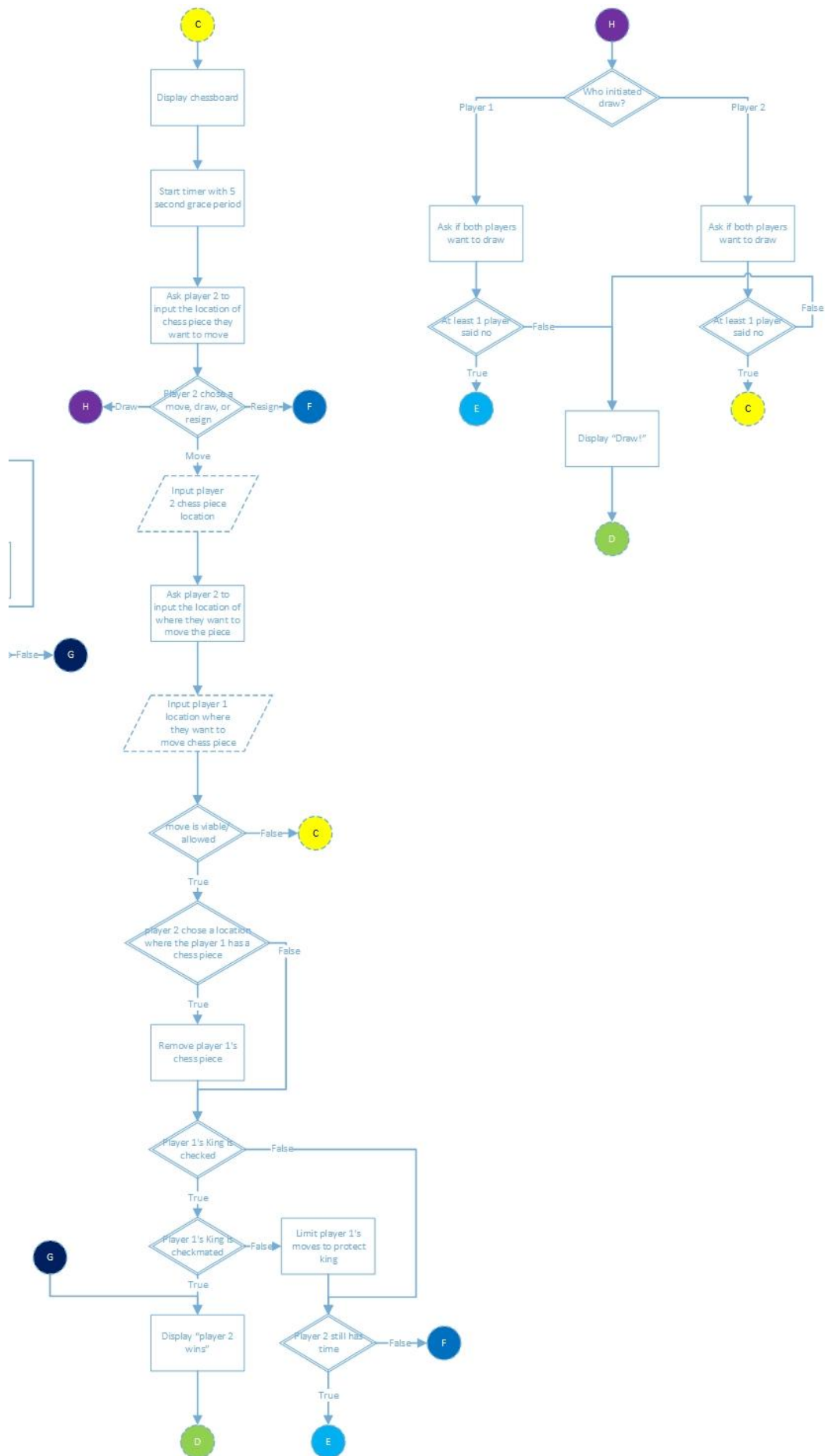
Leaderboard according to account database wins;

### III. Methodology

[https://drive.google.com/file/d/1RkQG4a4fQ6U3a5lg2sQADHq2hD32fHpk/view?usp=share\\_link](https://drive.google.com/file/d/1RkQG4a4fQ6U3a5lg2sQADHq2hD32fHpk/view?usp=share_link)







#### IV. Schedule of Activities

## Team Dionysus

[illegible]

## **V. References**

To be edited