Analysis

Background and problem definition

Landau Forte Academy Tamworth Sixth Form is in Tamworth, England and has a large body of A-Level/ BTEC students. The students attending the sixth form are typically 16-18, studying 2-4 courses. Mr. Dave, a computer science teacher at the school, manages and attends a chess games club every Wednesday after school closing hours. Recently, the number of attendees at the club has been declining this is due to students becoming frustrated by the lack of boards and pieces. To stop the decline and to remove the need to buy more and more boards as members increase, he has asked me to develop a digital chess solution that will allow players to log in and play against each other saving their win and loss counts. This will hopefully bring back leaving players and remove any issues in the future.

Dave wants the chess sessions to be local so that a game can be played at one computer. This will remove the need for unnecessary networking that can result in issues since they are all playing in the same room anyway. This will also half the number of computes required to play games. This also retains the community in the chess club that would be lost if playing on a network since players will be able to directly verbally communicated with each other. Players wins and losses will be stored allowing students to compare their statistics and abilities in chess.

He also wants me to include a play against the AI option for students that do not have a partner to play against. This ensures players always have something to do at the club and gives members an option to practice rather that always having the pressure of a real game against a real opponent. Wins against the ai will also count as wins on the stats sheet so players will be able to stay competitive using this feature even if they have no one to play with.

The main aim of my project is to create a chess app that can be played player vs player and player vs ai. It will be able to store wins and losses creating a more competitive environment at the school’s chess club. I feel this will allow new members to not be as intimidated waiting to play on the few boards and instead they can feel comfortable playing locally on a computer or against ai hopefully keeping them attending the club sessions.

This a general overview I will be interviewing a member of the club named Jim to gather further information towards my solution allowing me to set out objectives for my project.

Description of the Current System

The chess club is currently held after school on a Wednesday specifically from 15:05-16:00. When the club members arrive they go to the front of the room to collect a chess board if available. If there are no chess boards available, they must wait for a pair to finish their match. Sometimes players go the entire session without playing a game due to the length of the games. When a game is finished the pair reassemble the pieces into their starting configurations and allow another pair to play, no pair is able to play multiple games in a row. When the chess club is over the players pack up the board sets and return them to the front of the room where they are stored. To maintain a competitive environment the wins and losses of each player are tallied in a notebook. When a match is finished the pair must open the notebook and tally themselves in there; any player can open the book and look at the statistics of other players to compare.

Chess is an abstract strategy board game for two players. Each player controls his 16 of his 6 types of pieces on the chessboard. Each type of piece moves differently. The object of the game is to checkmate (necessarily capture) the opponent's king. A game can end in several ways besides checkmate. A player can resign, and there are several ways the game can end in a draw. The 6 pieces are: king, queen, pawn, rook, knight, bishop.

The king moves exactly one square horizontally, vertically, or diagonally. A special move involving the King known as castling is only allowed once per player per game

A [rook](https://en.wikipedia.org/wiki/Rook_(chess)) moves any number of vacant squares horizontally or vertically. It also is moved when castling.

A [bishop](https://en.wikipedia.org/wiki/Bishop_(chess)) moves any number of vacant squares diagonally.

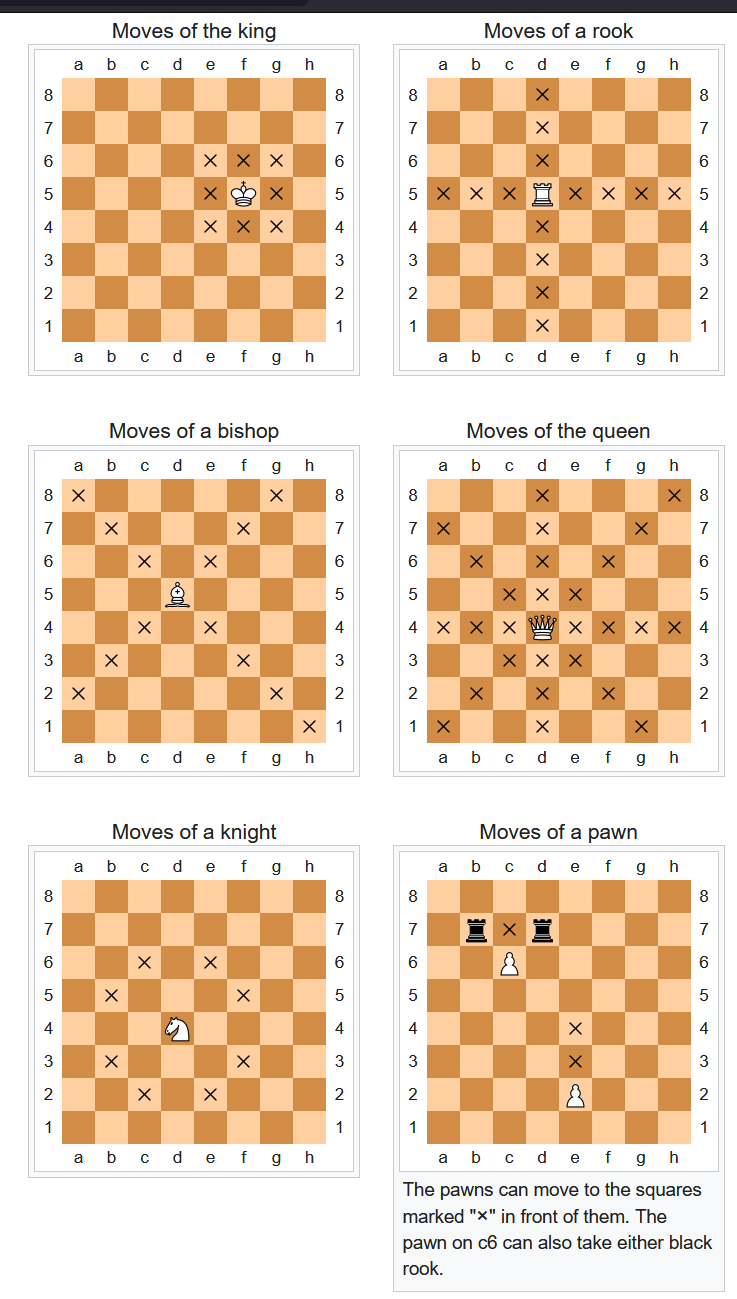
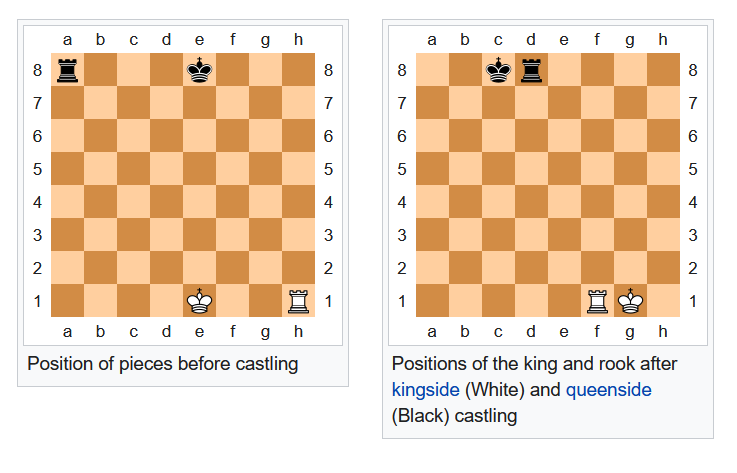
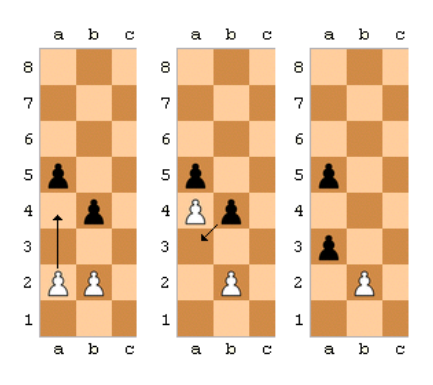
The [queen](https://en.wikipedia.org/wiki/Queen_(chess)) moves any number of vacant squares horizontally, vertically, or diagonally.

A knight moves to one of the nearest squares not on the same rank, file, or diagonal. (This can be thought of as an L pattern.) The knight is not blocked by other pieces; it jumps to the new location.

[Pawns](https://en.wikipedia.org/wiki/Pawn_(chess)) have the most complex rules of movement: A pawn moves straight forward one square, if that square is vacant. If it has not yet moved, a pawn also has the option of moving *two* squares straight forward, provided both squares are vacant. Pawns cannot move backwards. A pawn, unlike other pieces, captures differently from how it moves. A pawn can capture an enemy piece on either of the two squares diagonally in front of the pawn. It cannot move to those squares when vacant except when capturing *en passant*.

The pawn is also involved in the two special moves [*en passant*](https://en.wikipedia.org/wiki/Rules_of_chess#En_passant) and [promotion](https://en.wikipedia.org/wiki/Rules_of_chess#Promotion). En passant is where a pawn jumps, and an enemy pawn takes it as if it has moved only 1 square. Promotion is where a pawn can change into any piece excluding the king and itself upon reaching the end of the board.

On the next page are some examples of the moves.



Enpassant

A king isin check when it is under attack by at least one enemy piece. It must not be in check by the next move.

A king is in checkmate when it is under attack by at least one enemy piece and it cannot be saved in the next turn. In this situation the opponent wins.

A king is in stalemate when no piece can be moved without the king becoming in check. In this situation there is a draw.

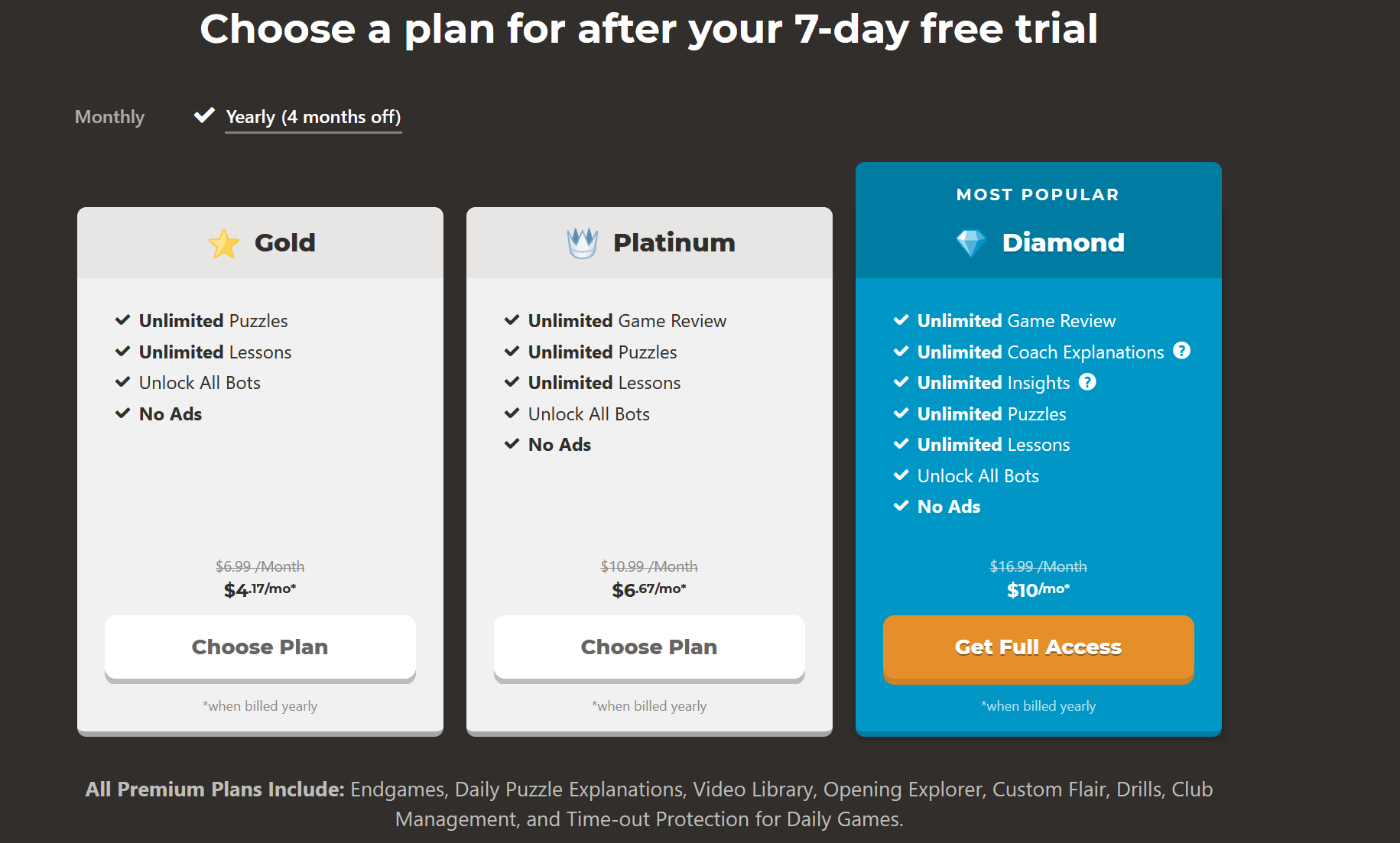
Analysis of Similar Systems

From what Dave has told us there are multiple chess solutions online that could meet the specification. These versions however require accounts with external companies and are online this causes child safety concerns. There is also the issue that we will not have the win ranking system we desire.

System 1 (chess.com):



Simplistic UI which is nice but contains a lot of unnecessary features for our app this can be overwhelming and will not be required.



A major concern is that there is a paid plan to remove ads and get access to the AI. This is unreasonable for a casual player in the club and no ads or payments can be there.

Overall, we can see that it has a simple UI with nice looking pieces. However, there are ads and a lot of complicated chess information that would not be wanted.

|  |  |
| --- | --- |
| Good   * User friendly experience simple to use * Can play against ai * Clean UI | Bad   * No local play options * Global chess ranking not just wins and losses * Overcomplicated options when in a match- to much information |

Proposed solution

* I will need a minimalistic UI that does not overwhelm the user.
* I need a local mode
* I need a play against the ai mode that you can gain score for

System 2 (freechess.com)



This site is simple you just start a new game and play. You can play online against random people or against a bot. This simplicity means you do not get confused and it is easily accessible for a person. There is also no ads. The issue however is the console on the side of the screen it will be very intimidating for a new user to see. The board and pieces also look weird and the light theme is not very attractive and modern. A darker theme would be better.

|  |  |
| --- | --- |
| Good   * User friendly experience simple to use * Can play against ai * No ads * Can get quickly into a game * Not an overwhelming UI | Bad   * No local play options * No ranking * A horrible console that occupies half the screen * Old fashion site with light themes * Random online |

Proposed solution

* I cannot have ads
* A score system based of the ai you have beat
* Can quickly get into a local game
* Simple UI
* Just a board no fluff

Now after this analysis I have more of an idea of what I am going to ask Jim during our questions.

Problems with the current system

Problems with the current system include a lack of boards. This creates an issue where not many games can be played, and people are forced to watch rather than play. The club has no funding, and no one is willing to pay out of their own pocket, so this has stunted new players joining the club due to the intimidating atmosphere of people watching them play. There is also the issue that every player needs a partner and cannot practice on their own. This forces new players into matches where they are out of their depth and will never enjoy the game since they are not even close to winning.

Secondly, it is also a struggle for the players to maintain competitiveness since they have to tally wins and losses in a notepad and simply look at it to see who the best is. This is bad because there are no other statistics to look at and overall, its hard to understand who is winning from lots of notes.

People also keep losing pieces this annoying because players either must omit the piece and play anyway to the advantage of the other player or they must use makeshift pieces out of paper. This is a temporary solution and takes from the game whilst also making the club look unprofessional.

Finally, the games take a long time to complete and often only a few players get to play per session. As I have said before this is due to the lack of boards and pieces. Even if more boards were purchased it would not be scalable as every time the club has there have been more people joining which results in the same issue again and they end up leaving.

Identification of the Users, Their Needs, and Acceptable Limitations.

Identification of the Prospective Users

Students will be the main users of my program, they can login using their school email or their home email. This is required to play any games although the ability to search through players is available to anyone. Students will not be able to view the code but teachers, who will oversee the upkeep, will have access to the code. This is so students do not rig the statistics. Teachers will need access to make changes if there are any problems in the future.

Identification of User Needs and Acceptable Limitations

So, I can better understand the user requirements, I conducted an interview with a chess club member with a strong attendance, Jim. I asked Jim questions about the current system, I think this will be an accurate portrayal of what the players at the club what since he is very active in the community. The interview transcript is below:

Q1. What is the current system for chess, at the school?

A1. Well, we arrive at 3pm and start setting up the boards on the tables. We only have a few boards, so it doesn’t take long. We then take it in turns to play with a partner. A lot of the club time is spent watching matches at the moment.

Analysis: Jim has told us that there is a lack of boards, and that he spends a lot of time waiting.

Q2. What problems are there with the current system?

A2. We only have an hour a week at the club so I feel there is a lot of time wasted in setting up the game. As I’ve said before there is a lot of time waiting so every second counts. I find myself not even getting to play most of the time. We also lack pieces we have to make fake pieces and it is a bit pathetic really, the other schools chess clubs don’t do that. Sometimes I don’t want to play with someone either I just want to practice while they all wait but there are no options for that.

Analysis: Jim has told me that they have a lack of pieces. He has also indicated he really doesn’t like waiting for people to play and that having to improvise with fake pieces is not a realistic solution. He has also shown he prefers to not play with people sometimes.

Q3: You said you like to practice on your own. How do you think you would be able to practice on your own in my solution?

A3: I’ve seen AI in chess before and I think that would be a great way for me to practice my skills. I think a lot of students would love that implemented.

Analysis: Jim wants an AI opponent to face he believes it will improve peoples time in the club.

Key objective 1: AI opponent

Q4: How does the club want the webpages formatted?

A4: We are all a fan of minimalistic pages that don’t overload us with information. A nice dark modern design is good.

Analysis: Jim and the club want a simplistic style for the page.

Key objective 2: Minimalistic UI

Q5: Is there any user data you want to be stored?

A5: Yes. I want the wins, draws, losses and other chess data to be stored. This would allow us to view every ones information and get an idea of who’s performing the best.

Analysis: Jim has expressed the need for data to be stored and accessed by other players.

Key objective 3: database with user data.

Key objective 4: a page to view this data

Q6: This would require a login system how would the administration work for this?

A6: We have quite a few computer scientists in the chess club I feel I would want these trusted individuals to have access to a GUI api that they could then make requests on and configure the users. This would require the passwords to be salted and hashed though so that we can insure these people don’t know the passwords.

Analysis: Jim wants a graphical interface for the computer scientists to work with the api when configuration is needed.

Key objective 5: login page

Key objective 6: GUI login/user data api.

Q7: What would you want this GUI api to show the users?

A7: I want it to show them the requests, responses and schemas and allow them to make requests directly through the interface.

Analysis: Jim wants the computer scientists to be able to view all of the api details so they can make their own requests rather then relying on using the actual websites ui to test.

Key objective 7: show requests, responses and schemas on the gui.

Q8: What are the minimum requirements the website will need when displaying and playing a chess board?

A8: The board will need to be able to show the moves of a piece that you click. It will have to be configured to only let a player use a piece if its there turn and if it’s a valid move. It will have to display checks and show win/ draw when a checkmate or stalemate is met.

Analysis: Jim wants the board to be able to operate a valid game only.

Key objective 8: Show valid chess game.

Q9: Is there a neeed for a nav menu at all?

A9: Yes, I would like to have a dashboard where I can see all of the features of the app so I can easily navigated to what I want. This will allow me to not get confused when using the solution.

Analysis: A simple navigation menu that shows all of the tools in separate pages will be needed.

Key objective 9: a navigation menu to all of the features of the app in seperated pages.

Q10: Do you think players will like the new solution?

A10: I think players will love the new solution as they finally get to play. I think it will attracct new players also

Analysis: Jim has said he likes the solution.

Q11: Are there any limitiations in this project?

A11: We need the project done ASAP as I’ve said before people are leaving the club and we do not have a good influx of new players due to the fact the immidiately leave. If its not done soon our club might get shut down.

Analysis: Jim has expressed the importance of this project.

Q12: Will this project bring back old players?

A12: Yes when they find out they can actually play the game rather then just sitting around they will come back. They will love the new ai and analysis features also. This projet will definitely intice them.

Analysis: The solution will bring back old players so should be a success.

To conclude here are the key objectives I’ve come up with:

1. AI opponent
2. Minimalistic UI
3. Database with user data for analysis
4. A page to view data for analysis
5. Login page
6. GUI api for db maintance/ configuration
7. Show requests, response and schema on api
8. Display chess game that follows valid rules
9. Navigation menu to access pages

Acceptable limitations

There are limitations that I will face during the development and execution of the solution. These are:

Time Limitations

Jim told me that it must be done as fast as possible so that players don’t leave. My goal is to have a full technical solution done by the end of February. This is a good time as I will have lots of personal time to focus on my examinations this year as well as the club getting good use out of my project. I will have failed if it takes longer then May to complete as this is not enough time to focus on my exams nor is it enough time for the club to regain players.

Website will not be deployed forever

Whilst it would be relatively straight forward to deploy my site to the web so that anyone in the school can access it I would have to pay a few to keep it up. I am not willing to do this out of my pocket so eventually after I have left the school it will come down and someone else will have to have the skills to deploy and upkeep the site.

My Programming skills

Whilst I have work experience shadowing software engineers creating a C# API, I have not done it on my own so it will be a challenge to keep it enterprise level. I have also never done AI and engine development so the logic behind that will require a lot of research. I will also have to work on my creative skills as I have little experience on frontend web development. This may result in a loss in time.

Proposed Solution (With Chosen Language and Justification of Language)

After my analysis I have concluded that a website is the best option available. I have decided to create a bespoke solution using ASP.NET’s Web Api and Blazor frameworks. All of my code will be in C# with the exception of the Razor pages that use C# scripting but also HTML and CSS.

Use of .NET

For my chess application I am manly writing in C# and other .NET languages; therefore I have chosen to write in the Jetbrains Rider IDE as I have a lot of experience in their IDEs. I have been using JetBrains IDEs since high school where I started with Pycharm. I like there IDEs because they have many of the popular languages supported in similar IDEs. For example, I can write C++ or Rust on CLion and then move to WebStorm for Javascript/Typescript. They also have a lot of community plugins that will speed up my development like SQL database viewers. They have debuggers and give you hints on how to refactor your code with resharper. This allows me to be more productive and reduce errors/code smells.

C# is an object-oriented language this allows me to have classes to represent the elements of the chess board. I can also use inheritance to share features for example having a piece base class for the 6 pieces. ASP.NET provides me a web framework I can use C# this framework has events that I can use to react to the user inputs.

Programming in Rider for ASP.NET is made simple since they recognise the dotnet watch console command so when I run the page it watches for UI changes I make and automatically refreshes so I can see the changes as I make them. Blazor also has a lot of community support and advice on how to make good pages so making the UI is as simple as possible. It also uses the popular component style of web frameworks where I can make my own custom tags that then display to a page much like how functions and classes work in an object-oriented language.

A big problem with JetBrains is the high cost of there IDEs. A lot of developers are ok with this because they believe the productivity boost is worth it however, I am unpaid sadly. Rider for example costs £120 at the time of writing and all of their ides in a bundle costs £230. Luckily, JetBrains was nice enough to have an educational license for their ultimate package so I signed my school email up for that and got it for free but I will have to consider the pricing when I leave school.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Free IDE available. | May have to pay £120 in the future. |
| Lots of community advice available | .NET and ASP.NET sdks required for people to run |
| Great debugging. | Latest version I’m working with not available on school computers. |
| Great refactoring. | Required to learn razor, html and css on top of my schools chosen language C#. |
| Web events for interacting with the user |  |
| OOP/Procedural programming so I can represent chess well. |  |

Objectives

1. Loads simple landing page that says chess or Welcome “user” if logged in
2. Animated nav menu that has a list of features inside
3. Logo on every page that when clicked returns user to landing page
4. Login/signup page
5. On Signup if username already exists display message
6. On login if username or password is invalid display message
7. Login api has GUI for future development / testing
8. Login api shows all of the different requests
9. Shows all of the different schemas
10. Shows example bodies and responses including bad ones.
11. Shows which requests require authentication and which do not
12. Local game page
13. Displays board with pieces in standard configuration
14. Tiles are indexed
15. When piece is selected displays all valid moves for that turn
16. When valid move square is clicked moves piece respectively
17. When in check king square highlights red
18. When in checkmate win message is displayed to the winner and an option to return to the main menu is shown
19. When in stalemate a draw message is displayed and an option to return to the main menu is shown
20. Ai game button brings user to a game difficulty selector
21. When difficulty selected user is brought to a chess board against an opponent
22. When move is made ai makes a responding move
23. All objectives of local game move apply
24. Add to the api to include request for users favourite move
25. Add to the api to include request of users win rate.

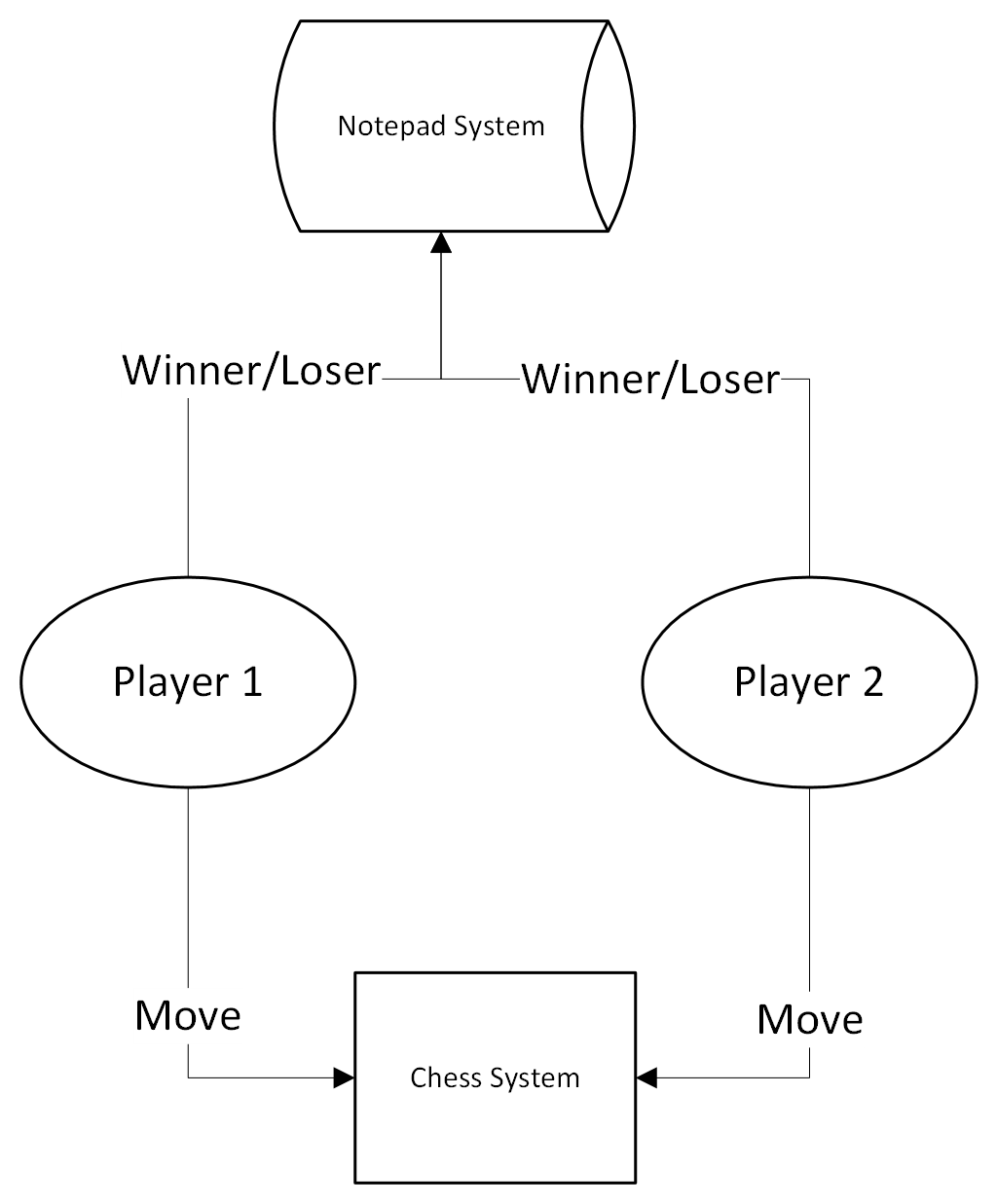
Data Dictionary

In the current system there are small amounts of data stored.

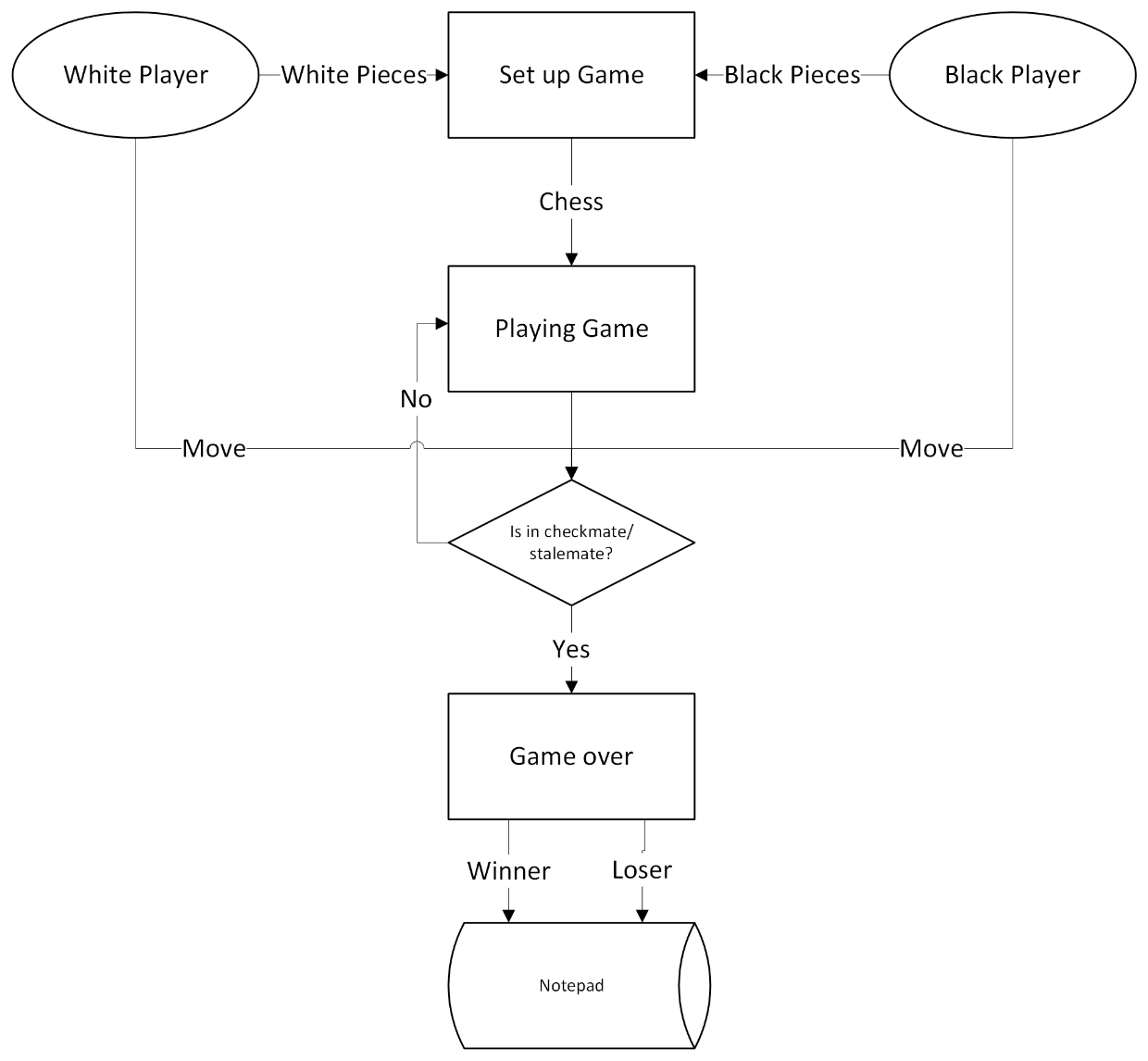
* Player 1’s name (Text)
* Player 2’s name (Text)
* Player 1’s pieces (List of Objects)
* Player 2’s pieces (List of Objects)
* Winner’s name (Text)
* Loser’s name (Text)

Data Follow Diagrams for the Current System

Level 0 Data Flow Diagram:

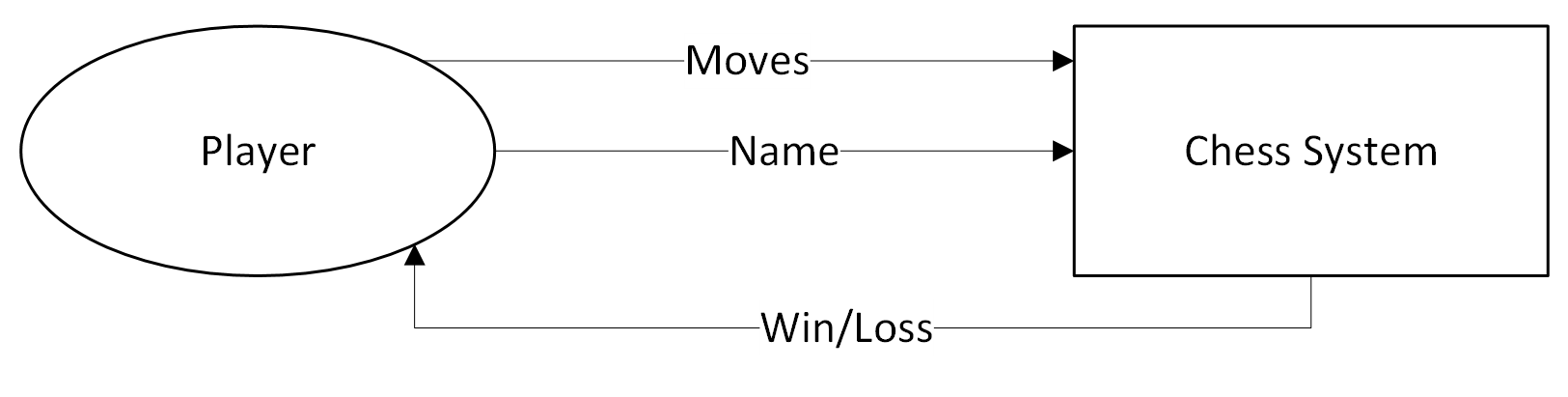


Level 1 Data Flow Diagram:

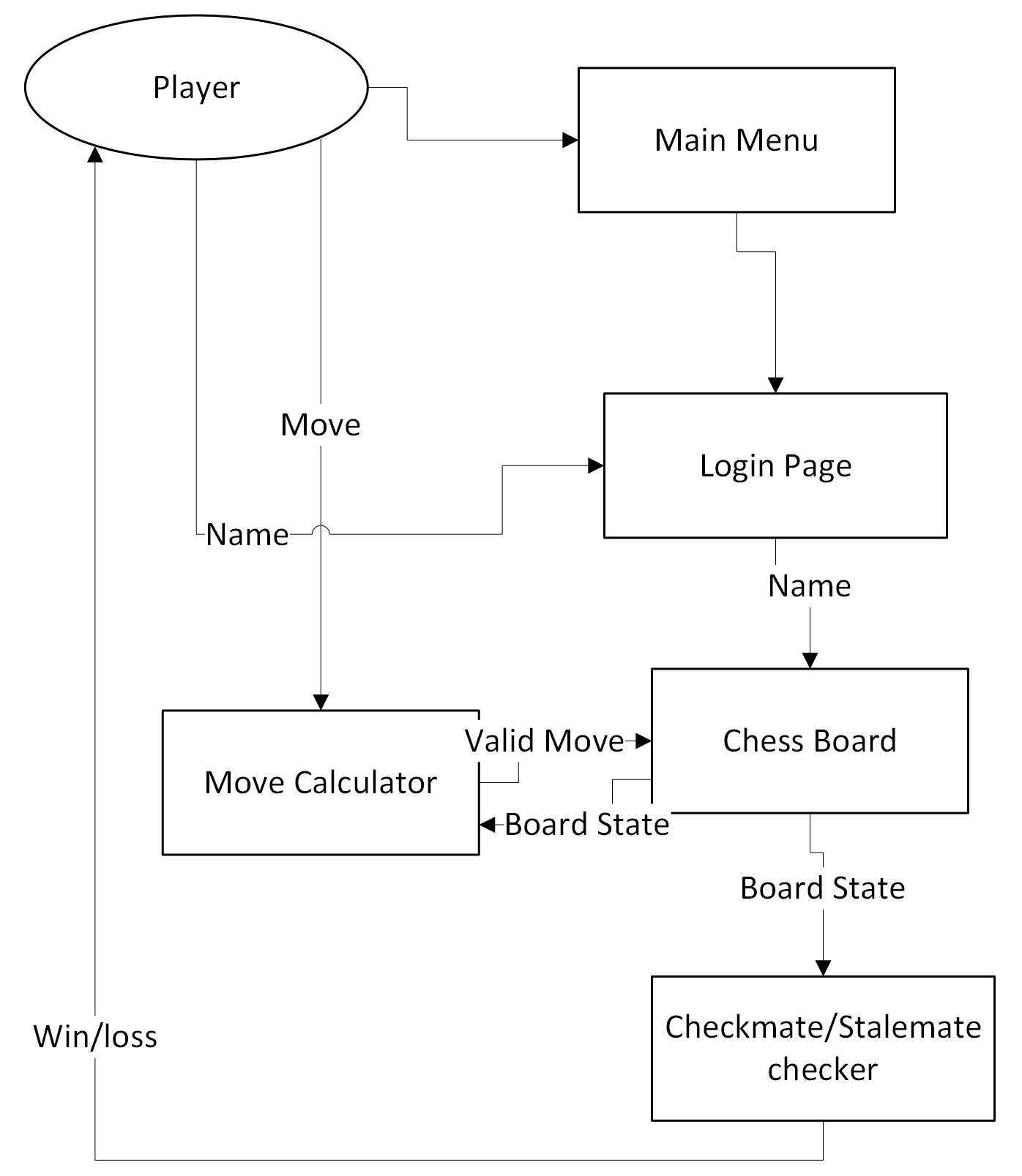


Data Flow Diagrams for the New System

Level 0 Data Flow Diagram:



Level 1 Data Flow Diagram:



Flow of Data for New System

**To do: Create massive flow chart of everything**

Data Sources and Destinations

For the current system to function, certain data is required.

* Player pieces, all of the pieces a player has
* Board state, the positions of everything on the board
* Valid moves all of the moves that follow the rules of chess
* Current turn the alliance that is currently playing

The system also outputs values

* The win or loss back to the user

Data Volumes

Space Complexity

My system uses a large amount of data and will involve large complex class structures. I will try to keep my class sizes to a minimum as the constant allocations and refreshes of the board state will leave a lot for the garbage collector slowing down the speed of the application.

The bytes required is not calculatable at this stage as there will be a large volume of associations resulting in varying allocation sizes.

The data for the other pages is static though so we can estimate the volumes for these.

|  |  |  |
| --- | --- | --- |
| Username | String | 2Bytes \* 55chars + 4Bytes for object allocation = 114Bytes |
| Wins | Integer | 4Bytes |
| Losses | Integer | 4Bytes |
| Move | String | 2Bytes \* 10chars + 4Bytes for object allocation = 24Bytes |
| AiDifficulty | Integer | 4Bytes |

Time Complexity

Move calculation requires every piece to be individually calculated since we need to ensure they all aren’t attacking a king. This results in a large time complexity that is unavoidable in chess. We can avoid this by ensuring to only do one team at a time. The ai will also use a recursive Minimax which will have a large time complexity. We will have to optimise it as much as possible to avoid large processing. The optimisations may take a long time to create slowing the production of the app.