* 1. **INTRODUCTION**
  2. **Background**

Chess has attracted players for centuries, known for its depth of strategy and intellectual challenge. Every piece on the square chessboard has unique tactical possibilities, providing players with an infinite number of strategic paths to explore. Chess has not only survived but flourished in an age where technology and the internet have left their everlasting stamp, finding a new and intense life in the modern world.

It has experienced exceptional growth in recent years due to advances in technology and increased online access. Chess has been transformed by websites like chess.com and lichess.org, which have opened it up to a worldwide fan base. Players of all skill levels can participate in thrilling matches on these platforms, analyze their games using sophisticated software, and interact with other chess players from across the world. Chess has been given new life by the digital revolution, which has changed it from a static, regional pastime into a vibrant, global activity.

* 1. **Rational**

My dissertation thesis is at the meeting place of software development and game development which takes center stage within this quickly expanding online multiplayer gaming ecosystem. The main goal of this project is to create a multiplayer online gaming platform that can accommodate the wide range of modern gamers' preferences. This system has several features designed to improve the gaming experience, including a user-friendly portfolio for tracking progress, simple multiplayer functionality with integrated chat, an effective rating system with leaderboards for competition, several gaming modes to suit different tastes, an advanced analysis tool for strategic improvement, and a variety of entertaining puzzles for entertainment and challenge. In a time when playing video games is no longer just for fun, my project aims to make an important addition to the field of online multiplayer gaming where strategy, social interaction, competition, and entertainment come together to rethink the future of interactive gaming.

We will go into the specifics of this software engineering dissertation as we delve into the field of online multiplayer gaming, looking at its design, development, and creative solutions for the gaming sector. We will explore the inner workings of this project in the upcoming chapters, breaking down its architecture, functionality, and practical applications. We'll see how technology has changed the gaming scene by making connections and challenges possible across borders. With this dissertation, I hope to promote understanding of the rapidly developing field of gaming on the internet and the important impact that software engineering will have on its future development.

* 1. **Aims and Objectives**

The project's goal is to develop an effective online multiplayer gaming platform that improves gameplay and creates a sense of community. The key aims include:

* Making the whole gaming experience better.
* Promoting online gaming interaction with one another.
* Utilizing analysis techniques to enable strategic improvement.
* Offering a range of game options.
* Maintaining fair competition with a strong rating system and leaderboards.

The project will carry out the following technical activities to achieve the aims listed above.

* Create a responsive and visually impressive user interface (UI) for gameplay using HTML5, CSS3, and JavaScript (JS).
* Use MySQL as the database system and Laravel as the framework for effective backend administration to ensure smooth user interactions and data management.
* Use WebSocket technology to provide real-time online matchmaking and chat features that allow for quick connections and communication between participants.
* Create and build an adaptive rating system that gives newly registered players initial ratings and uses statistical methods to change ratings in accordance with game results.
* Pairing players with similar skill levels, a rating system which provides fair matchmaking for proper challenging opponents in online games.
* Give players access to a variety of game modes and categories, including time-based games like blitz, bullet, and traditional variations within the online multiplayer framework, to improve the gaming system.
* Make interactive leaderboards that encourage players to improve their game and compete for the top places.
* Create a tool that allows players to carefully analyze their matches by integrating the Stockfish API.
  1. **Dissertation Methodology**

Agile is the software development methodology selected for this dissertation. It is an approach that emphasizes flexibility and customer participation throughout the development process and is highly adaptive and iterative. To provide continuous feedback, improvements to changing requirements, and active stakeholder involvement, it divides projects into tiny, manageable units known as sprints. Agile is well known for its benefits, which include the capacity to adapt quickly to changing priorities, reduce project risks through incremental project delivery, improve client satisfaction, and enable quick replies to feedback. It is essential to recognize the disadvantages as well, such as the decreasing importance of complete paperwork and its limited applicability for projects with predictable requirements. In-depth study of the concepts, benefits, and drawbacks of the Agile methodology's applicability as well as its impact on the creation of an online multiplayer gaming system will be provided in this dissertation.

Some advantages of agile methodology are listed below:

* Agile is flexible and adaptable, making it a good choice for projects that are dynamic and constantly changing.
* In Agile, incremental releases help detect problems early and address them quickly, lowering the chance of project failure.
* Constant engagement with customers encourages a collaborative environment, ensuring that the finished result closely matches the client's expectations.
* It starts delivering functional components early in the project, giving stakeholders a sense of concrete progress and early access to important features.

Some disadvantages of agile methodology are listed below:

* It can be difficult to keep detailed records of design decisions and requirements since agile promotes functioning software above detailed documentation.
* It is most useful for projects with rapidly changing requirements, for projects with fixed requirements, it may add unnecessary complexity.
* It can be resource intensive as it requires a large amount of time and effort investment from team members, stakeholders, and customers.
  + 1. **Software Requirements Engineering and Solution Specification**

This research equally focuses on three important techniques in the fields of Software Requirements Engineering and Solution Specification, interviews, study of comparable systems, task observation. To completely understand the project's requirements and objectives, it is essential to use all techniques mentioned here.

Advantages, Disadvantages, and proper justification for using these techniques are mentioned below:

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Advantages | Disadvantages | Justification |
| Interview | * Wide understanding of opinions from individual users. * Direct and interactive communication. * Allows clarification and immediate feedback. | * Limited to individual viewpoints. * Scheduling and conducting takes a lot of time. | Interviews are chosen as a requirement gathering technique due to its unique ability to provide in-depth and personalized insights directly from users. This approach allows for a proper exploration of individual perspectives, requirements, preferences, and expectations in system solution. It also encourages direct and engaging dialogue, allowing for quick explanation and feedback, which can greatly improve the quality of information gathered. Additionally, interviews provide a chance to establish a connection with customers, establishing an environment favorable to open and honest dialogues. |
| Comparable System Study | * Provides ideas of designs and best practices. * It can prevent possible system failure. | * Possibly won't address specific project needs. * Chances of not considering latest innovations. | Comparable system study utilizes established techniques which improve project reliability by accepting proven processes and best practices. This not only reduces risks but also offers a strong framework for carrying out projects successfully. |
| Task Observation | * Gives a direct understanding of system working mechanism. * Proper findings of inefficiencies and limitations. * Quick detection of problems | * Expensive in terms of time and resources * Regular tasks of users may be disturbed. | Task observation is an important technique for identifying the need to customize software to users' specific requirements, which makes the system more efficient and increases its usability. Investigators get essential insights into customers' regular operations, difficulties, and unique requirements through the careful observation inside their working environments. |

* + 1. **System Analysis and Design**

This section focuses on the systematic method used for the system analysis and designs. Many modeling tools were used to achieve clear alignment with user needs. Activity diagrams, class diagrams, sequence diagrams, and use case diagrams were used in understanding system behavior, designing structure, and showing interaction events. The advantages and disadvantages of these model are listed below:

|  |  |  |
| --- | --- | --- |
| Modeling Technique | Advantages | Disadvantages |
| Activity Diagram | Visualizes workflow and process models efficiently. | Complexity may increase for elaborate procedures. |
| Class Diagram | Clearly represents system structure and class relationships. | Limited in capturing workflow of class. |
| Sequence Diagram | This diagram shows time-related features of component relationships. | In complex environments, this can become complicated. |
| Use Case diagram | Provides an in-depth perspective of system functionality. | Oversimplifies complex interactions within the system. |

* + 1. **System Implementation**

Many important parts must be developed during the system implementation phase to ensure the system's operation. These components include user interface design, real time communication between players, and backend programming. Specific programming languages and frameworks have been chosen to do these objectives effectively:

* User Interface Design:

A variety of web technologies will be used to develop the user interface, which determines how users interact with the system. HTML5 (Hypertext Markup Language) is used for content structuring, CSS3 (Cascading Style Sheets) for style and layout, and JavaScript (JS) for chess programming and user experience enhancement.

* Backend Programming:

We chose PHP, a widely used programming language famous for web backend programming, for the backend, which is important in controlling server-side activities, data storage, and overall system functionality. To improve development productivity and maintain a disciplined approach, we'll use the framework of php known as Laravel which is well-known platform for building strong web apps. We would like to speed up and fortify our backend development efforts by utilizing Laravel's inherent features such as routing, database management, and effective safety methods.

* Realtime Communication:

The system will use WebSocket technology to enable smooth communication between players during matches. The Laravel framework provides features for broadcasting events which will be utilized for developing these capabilities.

* Integration with Stockfish Chess Engine

To provide extensive chess analysis and move suggestions, our system will easily link with the Stockfish chess engine. We'll use Node.js to send commands to the Stockfish executable file and collect replies. This integration improves the entire gaming experience by providing gamers with expert-level insights into their chess positions.

* + 1. **MySQL for Database Development**

The selection of a reliable database management system (DBMS) is an essential decision in the development of our chess platform. We are certain that MySQL fulfills our project's requirements after a careful evaluation of numerous database solutions and consideration of their relative advantages and disadvantages. It is the best solution because of its dependability, scalability, performance optimization, and compliance with our selected technology stack.

* + 1. **System Testing**

System testing is essential for guaranteeing the stability and dependability of our chess platform. We used unit testing, a core part of Laravel framework development, to carefully evaluate its functioning. Unit testing enables us to thoroughly evaluate all our platform's features and capabilities, ensuring their correctness and accuracy. Furthermore, we concentrated on certain components, such as the login, logout, and registration procedures, to conduct thorough unit tests. This methodical testing methodology assists us in discovering and correcting any possible problems, hence improving the overall quality and stability of our system.

* + 1. **System Evaluation**

In this part, we examine the system's performance, functionality, and overall user experience in detail. The assessment process attempts to assess the platform's usefulness and find areas for improvement, mostly through relevant user feedback. We attempt to obtain firsthand information on their experiences with the system by asking for ideas and feedback from our user base, ensuring that it matches their expectations and provides a complete and enjoyable chess-playing experience.

1. **REQUIREMENT ENGINEERING**

We are going through the detailed documenting of client interviews, task observations, and a comprehensive review of related systems in this section. By combining these efforts, we can ensure alignment with user requirements and industry standards by filling the gap between the features of our product with other comparable systems that were originally developed.

* 1. **Elicitation Activities**

The elicitation activities section of the report starts with interviews which serves as a foundational starting point. These interviews are important for gathering critical information from users, helping us to find useful insights into their needs, preferences, and expectations. In addition to interviews, we will also walk through alternative techniques such as task observation and a detailed evaluation of comparable systems. These activities work together to shape the software's fundamentals that ensure fulfillment of user expectations.

Mr. Nishedh Karki, an intermediate chess player with a rating of 1600 on chess.com, was interviewed. His experience is not just limited to a single chess platform, as he has experience of utilizing multiple other chess platforms like lichess.org and stockfish for skill enhancement. His thoughts and point of view towards variety of chess platforms were gathered which provided us proper understanding of user expectations and preferences in the field of online chess game.

* + 1. **Interview Plans**

Client: Mr. Nished Karki

Interview Date: Friday 22nd June 2023

Time: 01:00 to 03:00 PM

Location: Baneshwor, Kathmandu

|  |  |  |
| --- | --- | --- |
| Objective | Interview Question | Interview Answer |
| Motivation | How long have you been playing chess, and what motivates you to continue playing? | I've been playing chess since childhood, and it has always been a hobby of mine. Playing chess online has become my primary source of entertainment after exposer to the internet and technology. |
| Which platform do you primarily use for playing chess online? Is there a specific reason for choosing this platform? | I primarily use chess.com for online chess gaming. I wouldn't say there's any specific reason for choosing it, but I find chess.com very useful because of its extensive user base. The broad user community offers a wide variety of matches, which aligns well with my playing strategies and preferences. |
| Do you feel that the platform you use currently fulfills all your needs for online chess gaming? | The platform I use is sufficient for playing multiplayer online modes, although other features are very limited. As a free-to-play player, I would appreciate having access to more varieties of games within the same platform to enhance the overall gaming experience. |
| Platform details | Can you explain in detail about the most appreciated features of current platforms? | As a free-to-play user, I primarily engage myself in online multiplayer games, solving puzzles, using game analysis tools, and keep up with community updates of chess.com. These features contribute significantly to my overall experience and skill development. |
| Is there anything about your present platform that you dislike or find problematic? | While I appreciate using chess.com, its profit-driven approach has several negative impacts. Frequent advertisement and restrictions for free players are extremely irritating. Furthermore, the limitation of puzzle and game analysis tool frequently slow down my skill improvement. |
| How important is the user interface and overall visual experience on your current platform? Are there any aspects you particularly like or dislike? | User engagement is dependent on the user interface and visual experience. Chess.com has a user-friendly UI, although there is always room for improvement. The overall gaming experience is boosted because of its simple and user-friendly layout. |
| System limitations | Have you ever encountered any major imperfections or problems while using the current chess platform? | Yes, my biggest dissatisfaction with the existing system is with the game engine and the rating system. Traditional chess engines can fail to deliver accurate evaluation in important positions. Furthermore, players need to engage themself in large number of games to achieve the ratings they deserve. |

* + 1. **Interview Findings**

Mr. Nishedh Karki, a competitive chess player, provided excellent information related to real world of online chess gaming platform. His experiences and perspectives as a player who has been deeply involved in chess for many years have provided an in-depth knowledge of the motivations, preferences, and challenges that players face in the online chess community.

The interview findings clearly show his dissatisfaction with current system and would be very excited if we deliver a chess platform with following features:

* Chess platform with responsive and attractive UI design.
* Providing a wide variety of puzzle games that are unlimited.
* Strong analysis tool that can even outperform traditional chess engine.
* A platform with many supportive chess communities.
* A platform offering a wide variety of multiplayer modes online.
  + 1. **Comparable System Analysis**

Comparable System Analysis research was carried out by closely evaluating important platforms and software such as chess.com, lichess.org, and the stockfish engine. The study investigated the complexities of these systems to obtain useful information and benchmark against standard practices which can contribute to proper development of our chess platform.

* + - 1. **Chess.com**

Chess.com is one of the most well established and important sites for online chess gaming. It was founded in 2005 by two friends, Jay and Erik, who saw a need for a more reliable and feature-rich chess website. Chess.com has evolved rapidly since it was launched, and it is now a global hub for chess lovers. With over a decade of experience, it has evolved into an extensive platform that not only provides a gaming area but also promotes a strong chess community. We are having an in-depth review of Chess.com, evaluating its strengths and weaknesses to guide the development of our own chess platform.

Strengths of chess.com are listed below:

* Chess.com has a massive and interactive community which promotes a diverse and active group of chess players. This environment allows players to interact with opponents of a variety of skill levels, leading to a dynamic gaming experience.
* One of the strongest features of Chess.com is its vast teaching resources. From instructional videos and tutorials to articles and puzzles, the website provides players with the resources they need to further enhance their chess knowledge and abilities. This dedication to player development guarantees that users have access to useful educational materials.
* It conducts a variety of competitions and events on a regular basis, focusing to both casual and competitive players. These events not only give an environment for friendly competition, but they also serve as a tool for evaluating one's abilities on a stage, which gives a sense of belonging within the chess community.
* It provides a diversity of game variation along with time settings to meet the different needs of its user. Along with that, the platform enables a variety of playing styles, whether players choose a rapid blitz game or a more silent and long-lasting classical match.

Weakness of chess.com are listed below:

* Chess.com has some of the interesting and helpful features and many important functionalities, such as limitless game analysis and infinite puzzles which are only available for paid subscribers. This creates dissatisfaction among free-to-play users looking for a complete chess experience.
* It has been a long time since this platform has been established resulting in huge number of functionalities in current time making the interface very complicated for new players.

In conclusion, Chess.com is a major online chess gaming platform which has its specific strengths and weaknesses. The success of this platform is represented by its active community, educational offerings, and numerous gameplay possibilities. However, limitations in advanced features, interface complexity, and player portfolio management highlight significant areas for improvement.

* + - 1. **Lichess.org**

Lichess.org is an open-source chess server developed by donations and volunteers which is a popular chess platform known world widely. Thibault Duplessis founded lichess.org in 2010 with the goal of delivering a free-to-play chess experience to chess players all over the world. He started as a hobby project making this site an open source for anyone to read the source code or make any contributions. Today, more than five million games are played every day in Lichess remaining 100% free. This section conducts an in-depth review of lichess.org, finding its strengths and weaknesses.

Strengths of lichess.org are listed below:

* The main difference that lichess.org offers lies in being completely free and open source. This makes sure that all its features are available to users without any payments, restrictions, or advertisement. This strategy not only attracts chess players but also developers who want to contribute to such a platform.
* Even though it is completely free, this chess platform has an extraordinary set of features. A strong game analysis tool, a wide variety of chess gaming modes, and proper training tools are some of the features. It also provides an open API, which encourages developers to create chess-related projects.
* It has created an active chess community. Its lively forums, discussion boards, and chat tools encourage interaction among chess players giving them feeling of linked to a larger community of chess lovers.

Weaknesses of lichess.org are listed below:

* The status of lichess.org as a nonprofit organization can lead to many problems. Such problems include delays in the development of new features as it lacks strong financial support.
* The user interface of this platform is less visually appealing as compared to its commercial rivals. It requires more improvement in its appearance and usability for improving overall user experience.

In conclusion, lichess.org is a competitive online chess platform which has a good reputation for its dedication to open-source ideals and free access. Its wide range of features, active community, and commitment to fair play makes this platform a strong competitor. However, financial resources and user interface design are still under the area of improvement.

* + 1. **Task Observation**

This section investigates the findings from our observations of Chess.com, lichess.org, and the Stockfish chess engine. These observations will be helpful in finding the specifications for our chess project. We will gather important knowledge on user behaviors, platform functionalities, and the chess engine's decision-making processes by closely looking into the features and procedures of these platforms.

Some significant observation in online chess platform is mentioned below:

* **Attractive User Interface:**

Both chess.com and lichess.org have impressive user-friendly and visually appealing interfaces that significantly contribute to the online chess gaming experience. Chess.com has a simple design with clearly displayed chessboards, game controllers, dashboard and navigation making it very simple to use. It also provides a full range of tools and materials for improving chess knowledge. lichess.org, on the other hand, focuses on responsiveness and minimum design with the idea of offering a clean design that favors chess players. Users can customize their experience by choosing a variety of boards and pieces representing their taste. Both platforms' interface designs prioritize comforting user experience by giving quick access to important functionality which delivers a satisfying user experience.

* **Well implementation of ratings:**

Both platforms have a strong dedication for well implementation of rating systems. The dynamic matchmaking algorithm between similar rating opponents is very impressive. Along with that the statistical algorithms for increasing or decreasing rating after each match completion ensures update in players rating.

* **Working mechanism of stockfish:**

Observing how Stockfish chooses the best move and evaluates chess positions is an important part of knowing how it works. This task observation is carried out by running Stockfish in the command-line for information related to analyzing positions and moves with commands.  
  
Command for finding best move with specific depth and fen position in stockfish.

A black background with white text

Description automatically generated

Output generated by stockfish.

A screenshot of a computer screen

Description automatically generated

* + 1. **Literature Review**
       1. **Algorithm for Updating FEN Chessboard Character Strings (2020, Azlan Iqbal)**
          1. **Introduction**

The Forsyth-Edwards Notation (FEN) is a standardized notation system widely used in chess to represent chessboard positions. Following each move in a game, one of the main tasks in chess programming and related applications is updating the FEN representation. The FEN string must be modified to appropriately reflect the current state of the chessboard. An innovative algorithm has been developed to address this basic challenge which can very efficiently handle the process of FEN string manipulation.

FEN notations have significant importance in chess programming. It provides a text which is in a human-readable format along with technique for recording the complexities of a chess positions, including features such as piece placement, castling rights, en passant squares, and move counters. The ability to swiftly and properly update the FEN string following each move is essential in the context of chess engines, databases, and instructional platforms.

* + - * 1. **Existing Approaches:**

Historically, updating FEN strings relied on intermediate array-based checkerboard representations. While these techniques were effective, they generated computational overhead and memory use problems. Recognizing these difficulties, a collaborative effort was made to develop an algorithm that directly manipulates the FEN string. This project aimed to simplify the process while reducing resource use.

* + - * 1. **The Algorithm in Detail:**

The method begins by utilizing the existing FEN string to extract essential data about the current chessboard situation, including the positions of all pieces, castling rights, en passant squares, and move counters. When the first parsing is finished, the algorithm shifts its attention to the move. Given a move, such as "e2 to e4," it generates the new FEN string by properly altering the piece placement data.

The algorithm's ability to handle multiple move events is one of its key features. It handles pawn promotions, castling actions, and en passant captures very efficiently by altering the FEN string in a efficiently way and dos not miss any events. This flexibility makes it ideal for the handling of FEN positions, especially for chess programming.

* + - * 1. **Performance and Practical Implications:**

Because of its simplicity and computing efficiency, the method was well recognized by a variety of chess-related applications. Its direct implementation on the FEN string eliminates the need for expensive intermediate representations, reducing memory consumption and computing overhead. This efficiency is especially important in resource-constrained situations, such as embedded devices or web-based chess platforms with heavy concurrent user demands.

* + - * 1. **Conclusion**

Finally, the technique for updating Forsyth-Edwards Notation (FEN) chessboard character strings defines a significant advancement in the field of chess programming. Its simple way of directly manipulating FEN strings provides a resource-efficient alternative to typical array-based solutions. While it succeeds in many situations, developers should be aware of potential issues and work to improve error-handling techniques. As it gains popularity, this method has great potential for improving the efficiency and accessibility of chess software, benefiting both chess fans and the larger community of software developers. Its influence on the chess programming ecosystem is expected to be very significant.

* + - 1. **Chess AI: Machine Learning and Minimax based Chess Engine (2023, Jyoti Madake, Chinmay Deotale, Geetai Charde, Shripad Bhatlawande)**
         1. **Introduction**

Chess, a game with a roughly 1500-year history, has been a continual source of curiosity and challenge for players all over the world. The development of strong chess engines has resulted in considerable advances in artificial intelligence and machine learning throughout time. In this literature we will investigate the ever-changing world of chess engine development, with a particular focus on traditional techniques that became the foundation for present methods.

* + - * 1. **Race for Development of chess engines:**

The creation of powerful chess engines capable of competing with human players and other engines has emerged as a key aim in the domains of artificial intelligence and machine learning. These engines try to duplicate the thinking and decision-making abilities of experienced human players while outperforming them.

Some of the techniques used for developing chess engines are explained below:

* + - * 1. **Deep Chess Model**

The Deep Chess model uses deep neural networks to forecast game outcomes based on current chess board positions. To create predictions, this technology uses both deep unsupervised and supervised neural networks. It is an early attempt to use deep learning techniques into the creation of chess engines.

* + - * 1. **Convolution Neural Network (CNN)**

Similarly, the use of a Convolutional Neural Network (CNN) model trained using Board Set Datasets has been examined. When verified against the Stockfish chess engine, this CNN model selects preferred positions from two input positions with an accuracy of 39.16%. The use of CNNs represents a shift in the method for evaluating chess situations.

* + - * 1. **Unsupervised and Supervised Modules**

Another strategy involves the use of unsupervised modules that use similarity distance functions to group comparable chess games together to decrease the number of sample games for learning or study. Associative classifiers are used in collaboration with supervised modules to predict and find typical chess board configurations and winning moves. This strategy combines unsupervised and supervised learning techniques to improve chess engine performance.

* + - * 1. **Deep Learning and Evaluation Functions**

Deep learning techniques were used to build Deep Neural Networks (DNN) that reflect evaluation functions. These evaluation functions are essential for determining the strength of different movements. To assess the trained functions, measures such as 1-1 accuracy and Top-1 accuracy, as well as self-plays, were used, showing the use of advanced machine learning approaches.

* + - * 1. **Minimax Algorithm**

The Minimax algorithm, when combined with an evaluation function, is a powerful tool in chess engine construction. This method includes scanning a tree of possible movements up to a specific depth and assessing depth depending on the evaluation function's expected outcome. It reflects a traditional decision-making strategy in chess engines.

* + - * 1. **Conclusion**

The search of powerful chess engines has contributed to the study of numerous techniques, ranging from deep learning and neural networks to classical algorithms such as Minimax. These strategies have established the foundation for new methods that transform machine learning and artificial intelligence, indicating that developments in the field of chess engine development will continue. This literature review and introduction give insights into the growing environment of chess engines, providing foundations for future research into fresh approaches and their application in this interesting topic.

* + - 1. **New Age Chess Engine (2023, Preethi, Mohammed Mujeer Ulla, Sapna R, Mohan K.G)**
         1. **Introduction**

The world of chess engines has changed dramatically in recent years because of rapid development of artificial intelligence and machine learning. This literature study investigates the original methods and strategies used in the construction of chess engine which evolves artificial intelligence to improve amateur chess performance, and its development considers not just software but also hardware and graphical user interface elements.

* + - * 1. **Chess Engine Overview**

A chess engine is computer software that analyzes chess situations and generates optimum moves or movement sequences. These engines frequently employ a command-line interface and are generally used in combination with GUIs that allow users to interact with the engine. The major purpose of the New Age Chess Engine is to provide an AI-powered opponent for amateur players, hence increasing accessibility and engagement.

* + - * 1. **Hardware and GUI Development**

In addition to software issues, the creation of a chess engine includes hardware and GUI issues. Efficient hardware configurations are essential for properly operating chess engines, especially when using sophisticated search algorithms like MinMax. These engines are intended to run on conventional PCs with no extra processing units required. GUI development efforts to give users an enjoyable and engaging experience by including 2D and 3D visuals, background music, and other media components.

* + - * 1. **Variations in Chess**

Several chess variations have been developed to increase the attractiveness of the game and serve to a wide range of tastes. Surakarta chess and hexagonal chess are two examples. These variations bring variety to the chess-playing experience by introducing varied rules for piece movement and unique board layouts.

* + - * 1. **Conclusion**

The New Age Chess Engine is a modern technique to improve amateur chess players' chess playing experience. Its development incorporates software, hardware, and graphical user interface (GUI) aspects, mirroring the larger progress of chess engines. Advanced approaches like Minmax, alpha-beta pruning, and neural networks help the engine analyze locations and determine optimum movements. Furthermore, the engine investigates multi-player variations and unique chess variants, providing players with a variety of difficulties. This survey of the literature gives insight into the creative methods and technology that have motivated the development of current chess engines.

* 1. **Requirements Specification**
     1. **Problem Domain Description**

After conducting interview and a study of comparable systems following problem domains are identified:

* + - 1. **Interface design:**

The challenge of interface design is the difficulty of designing an attractive and user-friendly user interface (UI) that attracts players and improves the overall gaming experience. A user interface that fails to engage users visually and functionally can lead to UI dissatisfaction and poor attentions span. To make their contact with the chess platform entertaining, players frequently want an understandable layout, simple navigation, and responsive design.

To handle the interface design difficulty, it is important to draw inspiration from successful platforms such as Chess.com and Lichess.org, as well as information gathered through user interviews and related ideas. By researching the aspects and functionalities people find engaging and useful in these platforms, developers can identify proper UI/UX design concepts. This involves carefully developing a visually appealing interface using color schemes, layout, and interactive features. User testing should be done on a regular basis to gather input and iteratively modify the design to ensure that it matches user expectations and preferences. Prioritizing simplicity of navigation, based on successful platforms' logical menu layouts and clear signposting, is essential for improving the overall user experience. In addition, responsive design approaches should be applied, capturing ideas from the flexibility seen in leading platforms, to ensure that the interface works smoothly across several devices and screen sizes. This dedication to focusing on user’s design concepts and insights from existing platforms will guarantee that players are provided with a visually appealing and user-friendly experience while accessing the platform.

* + - 1. **Multiplayer Experience:**

It is a major challenge in making sure that everyone can enjoy playing chess together online, even if their connection is poor or not very good. When the internet is slow or not operating properly, the game slows down, and things don't happen as soon as they should. This may be very irritating for players. It's important to find a technique that guarantees that all players, regardless of how excellent or awful their internet is, can enjoy playing chess online.

To overcome this problem and provide a smooth online chess experience, we can use a web technological method known as WebSocket. WebSocket assists the computer in determining the optimum path for each player's game data, taking their location and internet quality into consideration. It's like finding the shortest route on a map. Furthermore, we can use a technique known as caching to make the computer perform quicker and avoid lag.

* + - 1. **Analysis Tool:**

Many chess players have difficulties while using analytical tools on online chess platforms. By giving observations and ideas, these tools are intended to assist players in reviewing and improving their previous matches. However, gamers frequently struggle to make good use of these features. They might not understand how to access and use analytical tools, resulting in irritation and a missed chance to improve their skills.

To solve the issues that players have while using analytical tools is developing an appealing user interface (UI) specially designed for analysis chess board which can aid players in analyzing their match. Players should be able to see all the records of matches like moves and best move continuation.

* + 1. **Functional Requirements**

This part of report has list of all features and functionalities of chess platform, ensuring that it satisfies user expectations and delivers an interesting and smooth chess-playing experience:

* **Dashboard for Player Statistics:**

The dashboard will clearly show key information reflecting a player's overall performance and preferences on the chess platform. It will provide a detailed report of the total number of games played in all modes, including Blitz, Bullet, and Classic. In addition, the dashboard will provide the player's ratings in each of these game types, allowing them to analyze their skill levels and track their development over time. This tool provides players with useful insights about their game history, assisting them in identifying areas for development and adapting their chess-playing experience to their abilities and preferences.

* **Match with Engine opponent:**

The system will provide users with the option of competing against computer-generated opponents in "Stockfish Matches." To meet users with varied skill levels and preferences, the system will offer seven distinct difficulty levels ranging from 1 to 7. These stages are intentionally constructed to mirror the engine's increasing levels of complexity and depth. Players may choose the difficulty level that corresponds to their skill level and desired challenge, offering a personalized and interesting experience regardless of their chess ability.

* **Multiplayer Mode:**

The system's multiplayer feature will promote player-versus-player (PvP) matches by matching players with opponents who have comparable skill levels and same game mode selection. To do this, a matchmaking algorithm will take into consideration factors such as player ratings and the match category selected. This algorithm's objective is to provide balanced and fair matchups, which will enhance the overall playing experience. These connectivity and matchmaking will be completed by real time communication technology known as WebSocket. It provides responsive gaming, even when participants' internet connection rates fluctuate. The multiplayer mode aims to give players entertaining and competitive chess matches that fit their abilities and taste of game mode.

* **Game Variation:**

The system provides a wide range of game modes that are suited to the tastes and skill levels of a broad player numbers. Blitz, Bullet, Classic, and Puzzles are among the modes available. Players engage in fast-paced gaming in Blitz Mode, with a 1-minute limit, demanding rapid and decisive thinking. Bullet Mode increases the speed even further by giving players only 3 minutes, increasing the necessity for quick decision-making. Classic Mode follows conventional chess rules, allowing for more serious and strategic play with a generous 10-minute time limit. Puzzles supplement these modes by offering players with pre-determined board locations to solve, therefore improving their tactical and analytical abilities.

* **Match Recording and Analysis:**

The system will provide complete Match Recording and Analysis capabilities, which records the beginning and ending FEN positions along with player and opponent movements, these details will be recorded in database. Players will be able to analyze their recorded games in the Analysis area, where they will be able to access a full list of moves, board positions, and ideal move continuations based on stockfish reply, helping with skill improvement and strategic understanding. The user interface needs to be simple and easy to use.

* **Match Continuation with Engine:**

In the analysis section, players will have the option to continue a match if it has not reached its ending position due to a stalemate or checkmate.

* + 1. **Performance Requirements**

Performance Requirements are the nonfunctional requirements of a system which mainly focuses on resources required for smooth performance. They are listed below:

* + - 1. **Speed:**

The system must prioritize speed and responsiveness to provide a proper user experience. The reaction time should be practically immediate when players make motions or interact with the platform. This means that move inputs and game state changes should happen in milliseconds or less which makes sure smooth real-time action. Furthermore, when players check their profiles or examine their match history, data must be retrieved and presented quickly, ensuring players have instant access to their statistics and previous matches.

* + - 1. **Capacity:**

The system should be able to handle a high number of online players continuously while offering a smooth and responsive game experience. It must also have a powerful and scalable database system to efficiently store and handle player profiles, match histories, and chess puzzle data, guaranteeing that it can easily scale to meet the demands of a rising user base while maintaining optimal performance.

* + - 1. **Reliability:**

To provide an undisturbed gaming experience, reliability is essential. The system must continually maintain a high degree of uptime while reducing interruptions and server interruptions. Players should be able to rely on the platform for accessing their accounts and games on a constant basis. Furthermore, multiplayer functionality and game analysis systems should be developed for reliability, for data privacy and match disruption. This reliability gives players confidence, advising them that their chess sessions will go effectively.

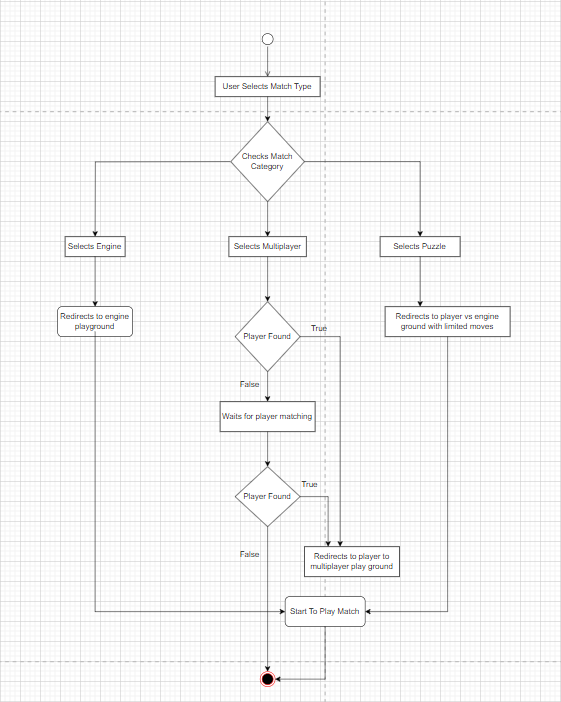
* + - 1. **Usability:**

To guarantee that players can easily explore and use the platform's features, the user interface (UI) should prioritize usability. This involves developing a simple and user-friendly UI with clear and logical menu layouts. Players should be able to easily switch between game modes, access puzzles, and assess their matches. Furthermore, the system should give succinct and easy-to-understand instructions for each function, improving user understanding and engagement. Because of its focus on usability, the chess platform serves both novice and experienced players, providing an easy and enjoyable experience for all.

1. **System Analysis and Design**

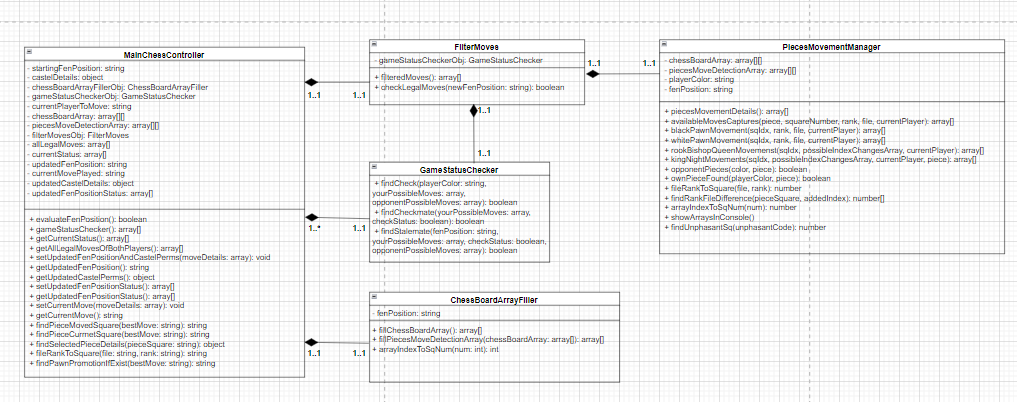
This part of the report covers the system study and design of chess platform using various Unified Modeling Language (UML). UML diagrams provide a consistent and visual way of viewing many aspects of the system which can be very beneficial for the development of system. Different types of UML diagram are explained below:

* 1. **Activity Diagram:**

****

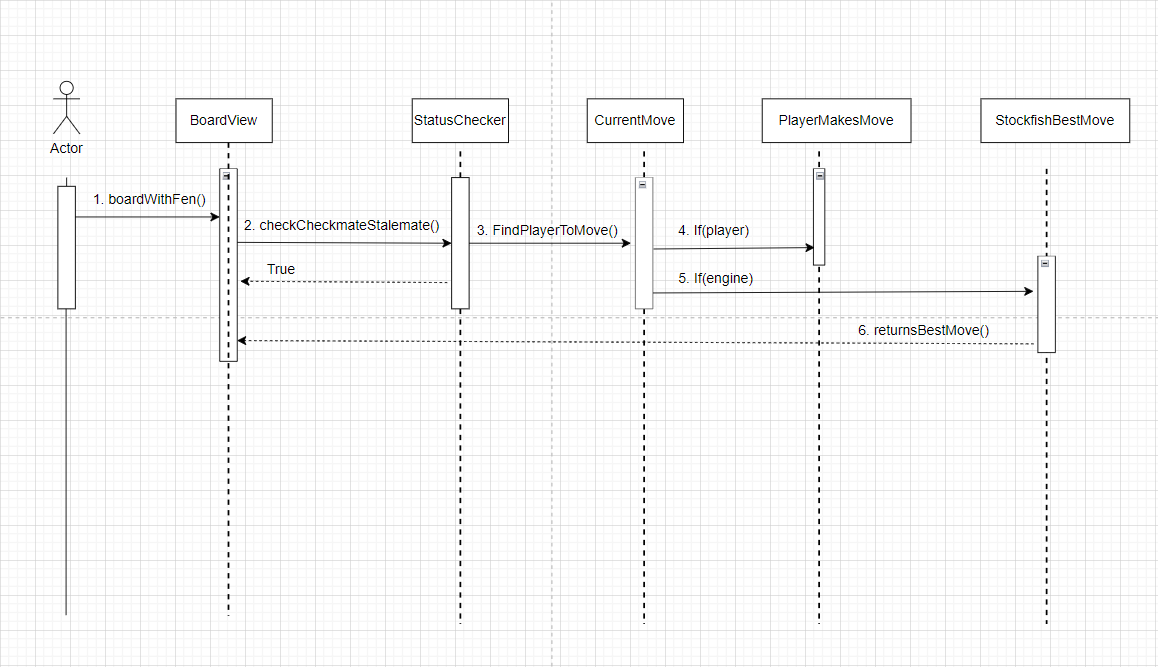
This activity diagram displays a player choosing whether to play a match with engine or online player. When a player selects multiplayer mode, they are placed in a waiting state while the system looks for a suitable opponent. When a match is identified, both players are told, and they can then join in multiplayer games, bringing the process to a close. If a player selects an engine match, the player will be playing against stockfish with various levels they choose.

* 1. **Class Diagram:**

****

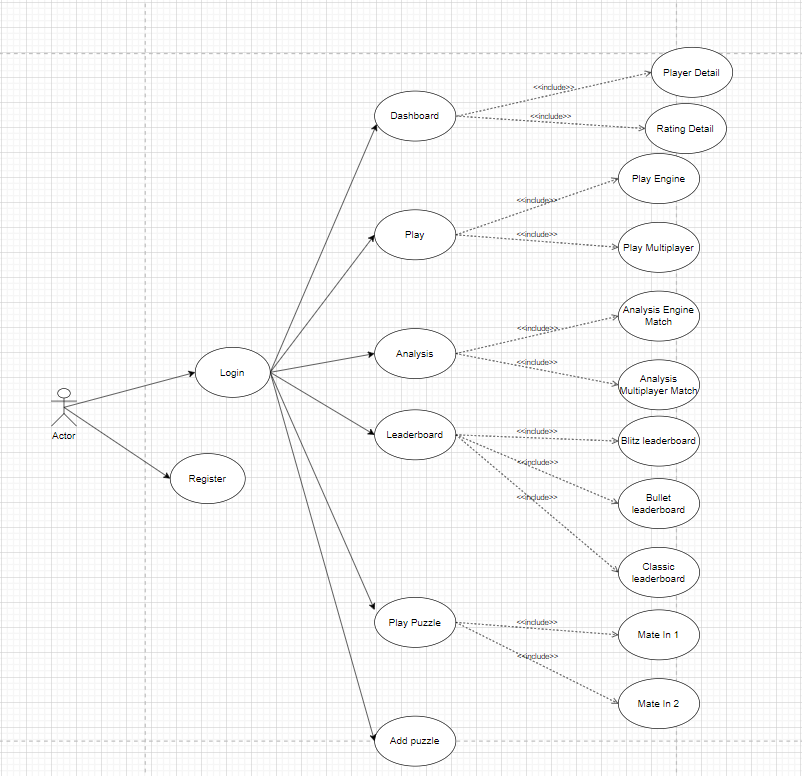
This is a complete class diagram of chess game play. The MainChessController receives FEN position and calculates whether the game can be played or not. If a game can be played, then it finds all the legal moves to be played. If the player makes valid moves or if chess engine returns valid move. It updates the FEN position and repeats the process.

* 1. **Sequence Diagram:**

****

This is a sequence diagram showing how players can play chess on a browser. First a chess board with piece position is displayed. If it is players turn to move. Players can engage with bowser and make a move. If the engine turns to move, engine will evaluate the position and make its move.

* 1. **Use Case Diagram:**

****

This is Use Case diagram which shows all the features provided to user. Players should register and then registered users can login. Players can view all the features at first layers like dashboard, play, analysis, leaderboard, play puzzle and add puzzle in side-navigation.

* 1. **ERD Diagram of database:**

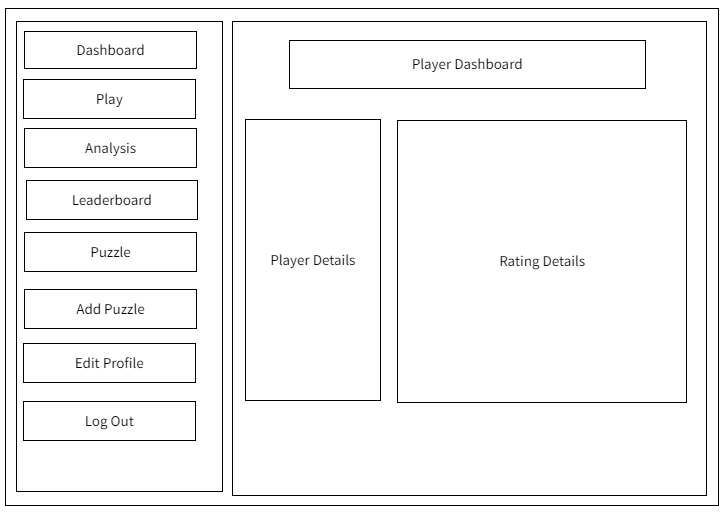
**A screenshot of a computer

Description automatically generated**

This is a database design for the system. It has places for all the necessary information that needs to be recorded from player information to puzzles.

1. **System Interface Design**
   1. **Wireframes**

Following wireframes were produced after system analysis and designs:

****This is wireframe design for player dashboard presentation. Profile details and rating details are planned to be displayed on the right side and navigation is planned to display at left.

**A screenshot of a game

Description automatically generated**

This is a wireframe for giving options for selecting match against online players or match with engine.

**A screenshot of a computer screen

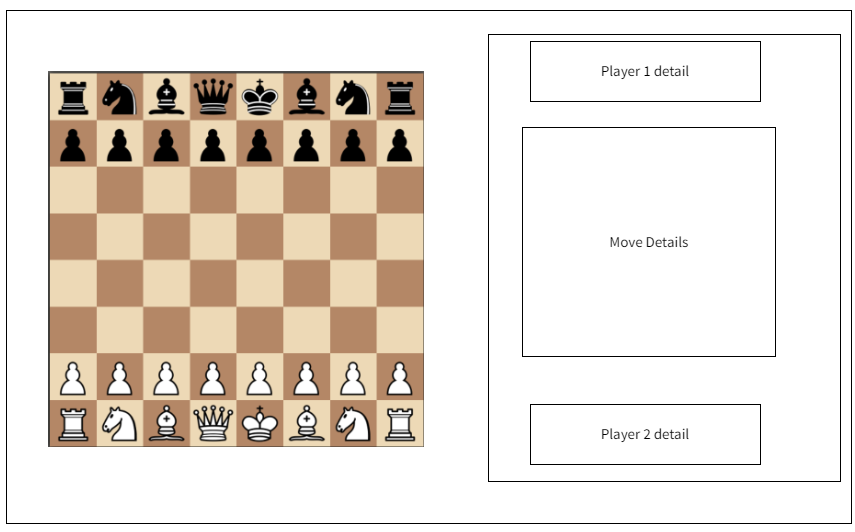
Description automatically generated**

The wireframe shown here is the planning for showing past match history of player. Players can view recent matches with engine and multiplayer.

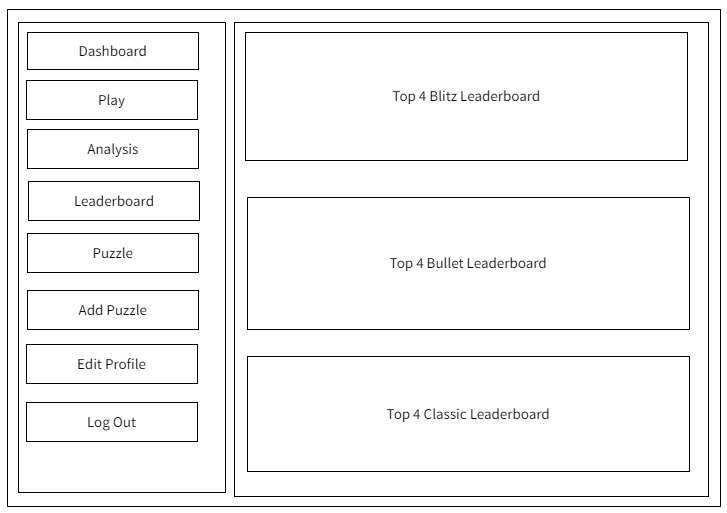
**A screenshot of a puzzle

Description automatically generated**

This wireframe has plans for displaying puzzles inserted in the system. Players will be viewing all the details of puzzles in sequence.

****

This wire frame has representation for match ground between players or engine. This wire frame contains planning for board information, player information and move details.



This wire frame has representation for player leaderboard for various categories. Players curtain details will be viewed on basis of rating.

* 1. **Mockups**

**A screenshot of a game dashboard

Description automatically generated**

This is mockups for player dashboard. Players can view all of their statistics here.

**A screenshot of a game

Description automatically generated**

This is mockups for selection of match type. Players can select a match against stockfish or play with other players online.

**A screenshot of a video game

Description automatically generated**

This mockup shows a list of recent matches with engine and recent matches with online multiplayer.

**A screenshot of a game

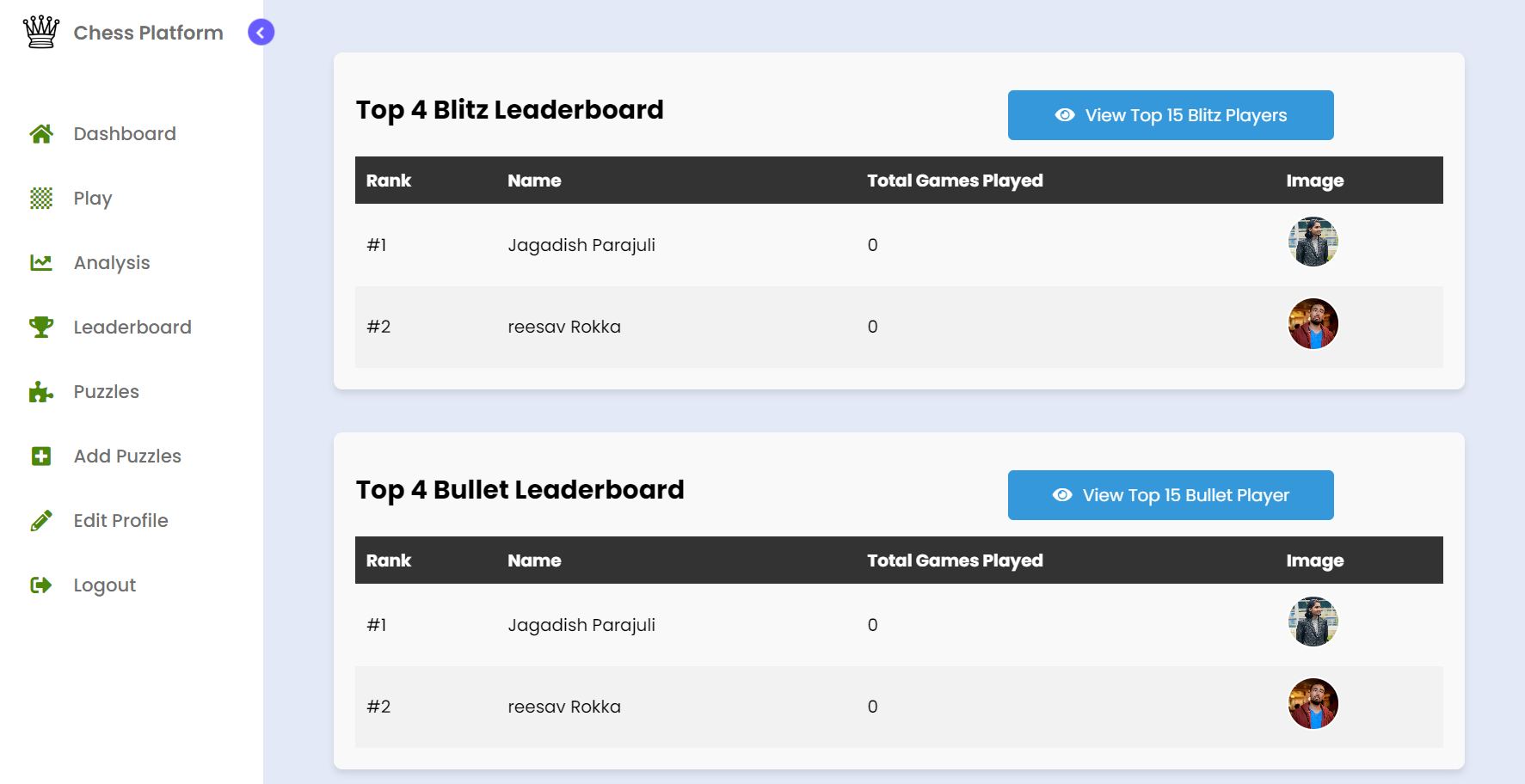
Description automatically generated**

This mockup shows a list of puzzles. System can have as many puzzles as inserted. Players can see the board position, puzzle description and puzzle status.

**A screenshot of a game

Description automatically generated**

This is mockup for match ground. Match ground either online or single player will be having same design as shown in figure.

****

This mockup has a list of top players with various categories. Players can view the top four players according to their rating.

1. **System Build and Technical Notes**

Mockups and wireframes were created to illustrate the user interface and functionality of our chess platform, assuring a clear blueprint for our chess platform.

* 1. **Database Connectivity**

After demonstration of wireframe and mockups, we developed a reliable database connection, employing MySQL as our preferred database management system (DBMS), to efficiently store and manage game and user information. This created the groundwork for the later technical implementation, laying the groundwork for a well-structured and unified system construction. We will investigate the complexities of our system development experience in this field, delivering useful technical insights and highlights.

**A screen shot of a computer code

Description automatically generated**

This is a section of .env file present in my project which creates connection between Laravel application and MySQL database.

* 1. **WebSocket Connection**

The Laravel broadcasting function, which relies on WebSocket technology, is utilized to improve participation and real-time communication inside our chess platform. We have smoothly established two critical events into our system using this feature: "MatchMaking" and "PlayerMadeMove."

A black screen with white text

Description automatically generated

The image shown here is channel registration for both MatchMaking and PlayerMadeMove detection events.

* + 1. **MatchMakingEvent:**

Our system actively looks for couples’ players with opponents of comparable skill levels or ratings via the MatchMakingEvent. This event makes use of Laravel's broadcasting features to spread matchmaking requests and replies among connected clients as smoothly as possible. This guarantees that players are paired with opponents who are both challenging and balanced. WebSocket technology enables the immediate and dynamic transmission of matching data, allowing participants to quickly select their preferred mode and begin their games.

* + 1. **PlayerMadeMove:**

Our chess platform will be known for its real-time gameplay, and the PlayerMadeMove event is at the core of that experience. This event is initiated whenever a player makes a move on the chessboard, and it serves several functions. After each move has been made, it updates the game state in real time for both players, guaranteeing that they see their opponent's move as soon as it occurs.

* 1. **Chess Programming in JavaScript:**

This section covers the fundamentals of integrating chess functionality into our system. This section explores key topics including FEN (Forsyth-Edwards Notation), board visualization, move generation, game status checks, player permissions, and Stockfish integration. It provides a formal and systematic overview of how JavaScript is employed to create an interactive chess experience within our web-based application.

* + 1. **FEN Position:**

FEN (Forsyth-Edwards Notation) is a chess notation system used to express the current position of pieces on a chessboard. It's a short, human-readable format for representing several aspects of a chess position. A FEN string is made up of numerous components that are separated by spaces and together give a meaningful picture of the game’s state.

The key components within a FEN string are as follows:

* **Piece Placement:**

The placement of chess pieces on the board is represented by this component. Uppercase characters for white pieces (K for king, Q for queen, R for rook, N for knight, and B for bishop) and lowercase letters for black pieces (k, q, r, n, b) are used. Empty squares are denoted by numerals (1-8) that indicate successive empty squares.

* **Active Color:**

Indicates which player's turn it is to move, with 'w' representing white and 'b' representing black.

* **Castling Availability:**

Describes whether both players can still cast, with 'K' and 'Q' denoting kingside and queenside castling for white, and 'k' and 'q' for black.

* **En Passant Target Square:**

If a pawn has recently moved two squares, this indicates the square that the opposing pawn may take en passant. If such an option does not exist, a hyphen ('-') is used.

* **Halfmove Clock:**

This is the number of half-moves that have occurred since the last pawn move or capture. It is used to calculate the fifty-move rule in chess, which can result in a draw.

* **Fullmove Number:**

The number of full moves made in the game, often beginning at one and rising after each pair of moves (one move by each player).

* + 1. **Board Visualization:**

One key part of designing our chess application is seeing the chessboard itself. To do this, we created a board visualization system with two unique classes. The first class is intended to generate an empty chessboard, which will serve as the basis for the game to evolve. The second class, on the other hand, uses FEN (Forsyth-Edwards Notation) position data to produce an HTML representation of the chessboard dynamically. This HTML element acts as the graphical interface via which players engage with the game, displaying the current location of the chess pieces and allowing them to move. This section digs into the specifics of how these classes work together to provide an immersive and user-friendly chess game experience.

A chess board with chess pieces

Description automatically generated

Empty board and board with initial board setup with FEN position: “rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1”

* + 1. **Valid Move Generation**

When our system receives a FEN (Forsyth-Edwards Notation) position, it translates the positional data into an array representation of the chessboard. This array serves as the foundation for establishing each player's legitimate movements. The system examines the current situation of the pieces, evaluates permissible movements based on chess rules, and creates a list of potential possibilities for the active player using an elaborate algorithm. This procedure guarantees that players are presented with correct and permitted move options, preserving the game's integrity, and improving the overall user experience.

A chess board with different chess pieces

Description automatically generated

Valid moves of white queen, white king, black night and black bishop are shown in above figure.

1. **Test Strategy:**

All the test cases provided here are available in the tests directory of the project folder. The field test cases have all the function name of test activity, description has the objective of test cases and expected outcome has result of test carried out.

* 1. **Feature Testing:**

Feature testing is an important aspect of assuring the stability and dependability of our Chess platform. These tests evaluate the platform's ability to navigate between important views and ensure that users are automatically routed to the expected interfaces. The tests include essential elements such as the dashboard, play view, analysis view, leaderboard, puzzle view, and edit profile view, ensuring that they all work properly.

|  |  |  |
| --- | --- | --- |
| Test cases | Description | Expected Outcome |
| testDashboardRedirection | Redirects to dashboard view. | HTTP status code: 200 (OK) |
| testPlayView | Redirects to play view. | HTTP status code: 200 (OK) |
| testAnalysisView | Redirects to analysis view. | HTTP status code: 200 (OK) |
| testLeaderboardView | Redirects to leaderboard view. | HTTP status code: 200 (OK) |
| testPuzzledView | Redirects to puzzle view. | HTTP status code: 200 (OK) |
| testEditView | Redirects to edit profile view. | HTTP status code: 200 (OK) |

* 1. **Integration Testing:**

This section introduces a collection of integration tests for evaluating user-related features. These tests focus on scenarios such as user registration, login, authentication, and dashboard access to ensure that the interactions between these features are effective.

|  |  |  |
| --- | --- | --- |
| Test cases | Description | Expected Outcome |
| testNewUserRegister | Accesses the registration page. | HTTP status code: 200 (OK) |
| testUserLogin | Accesses the login page. | HTTP status code: 200 (OK) |
| testUserLoginAuthen | Authenticates an existing user and accesses the dashboard. | HTTP status code: 200 (OK) |
| testUserRegistration | Registers a new user and checks if the registration process redirects correctly. | HTTP status code: 302 (Redirect) |

1. **Conclusion**