Software Requirements Specification for Chess Connect: Online tools combined with on-board vision to improve and share your game

Team #4,
Alexander Van Kralingen
Arshdeep Aujla
Jonathan Cels
Joshua Chapman
Rupinder Nagra

October 4th, 2022

Contents

Table of Revisions

Table 1: Revision History

Date	Developer(s)	Change
2022-10-04	Jonathan Cels	Template creation and document formatting
2022-10-04	Jonathan Cels	Non-functional requirements
2022-10-05	Joshua Chapman	Constants, Monitored Variables, Controlled Variables
2022-10-05	Alexander Van Kralingen	Added Context Diagram
2022-10-05	Joshua Chapman	Problem Description, Assumptions
2022-10-05	Jonathan Cels	Scope, Intended Reader, Stakeholders
2022-10-05	Rupinder Nagra, Jonathan Cels	Functional Requirements
2022-10-05	Arshdeep Aujla	Constraints
2022-10-05	Alexander Van Kralingen	Added Use Cases/Scenarios. Fixed system context diagram position. Added likely change.
2022-10-05	Rupinder Nagra	Purpose, Likely Changes and Unlikely Changes
2022-10-05	Alexander Van Kralingen	Added Undesired Event Handling section and a likely change.
2022-10-05	Arshdeep Aujla	Characteristics of Intended User, Stakeholders
2022-10-05	Arshdeep Aujla	Reflection
2022-10-05	Alexander Van Kralingen	Added FSM

1 Units, Terms, Acronyms, and Abbreviations

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	**unit**	**SI**
V	electric potential	volt
A	current	ampere
Ω	resistance	ohm
s	time	second
°C	temperature	centigrade
J	energy	joule
W	power	watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)

1.2 Abbreviations and Acronyms

symbol	description
A	Assumption
CSA	Canadian Standards Association
DD	Data Definition
FIDE	International Chess Federation or Fédération Internationale des Échecs
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
LCD	Liquid Crystal Display
LED	Light-Emmitting Diode
MCU	Micro Controller Unit
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
SS	Skills for Success
T	Theoretical Model
VnV	Verification and Validation
WCAG	Web Content Accessibility Guidelines

1.3 Mathematical Notation

1.4 Terminology and Definitions

Term	Definition
Legal Move	Moving a single chess piece from one square on the board to another in a way that follows the rules of chess.
Resign	To forfeit or surrender a game of chess.
Draw	To tie a game of chess.
Draw by Agreement	When both players agree to draw instead of continuing to play.
Capture	When a piece moves into the space of an opposing piece following the rules of chess, removing that piece from the game.
Check	A condition that occurs when a player's king is under threat of being captured.
Checkmate	A condition that occurs when a player's king is in check **and** has no legal moves to escape. This ends the game, and the player who delivered the checkmate wins.
Stalemate	A type of draw that occurs when the king is **not** in check, but no piece can be moved without putting the king in check.

2 Introduction

2.1 Document Purpose

The purpose of this document is to provide a set of requirements for a system that will integrate over-the-board chess with the online world of chess to assist in learning the game in a flexible manner. This includes a detailed description of our functional and non-functional requirements, including the performance and attributes of our design. The document also includes a project overview showcasing the behaviour with event handling and system diagrams, and the likely and unlikely changes we expect to encounter throughout the development of our product.

2.2 Characteristics of Intended Reader

The document is written with the purpose of guiding development for the Chess Connect team. The intended readers of this document are the developers of Chess Connect, Dr. Spencer Smith, and Nicholas Annable, the teaching assistant assigned to this project. The document is thus written for an audience that is well-versed in formal specification at

a university level. This includes models, diagrams, and mathematical notation. Readers should also have a university-level understanding of electrical circuit knowledge.

2.3 Characteristics of Intended User

This project will assist chess players of any level that are looking for a tool to help them learn and study the game. For beginners, the board serves as a learning tutorial and a general introduction to the game, while intermediate and advanced players can use the engine move recommendations to study new lines, puzzles, and specific positions to enhance their chess skills.

2.4 Stakeholders

The professor and TAs for the course SFWRENG 4G06 as they will be providing the feedback which will directly affect the project's development. Society is also a stakeholder because this project will provide the members of society another gateway to be involved in the community through the game of chess. Lastly, this project will also be relevant to chess tournament organizers looking for a method to easily broadcast and share their games online in real-time.

3 Problem Description

Online chess has functionality for both beginners and experienced players to learn and practice the game. However, these forms of learning emphasize a visual style of learning using a standard keyboard and mouse, while physical boards place emphasis on tactile learning when learning or studying the game. The highest-rated chess players often use a combination of the two styles to optimize their play. However, no option exists for players of any skill level to integrate their over-the-board and online play with one solution.
Chess Connect plans to centralize these two mediums of studying the game in order to provide flexibility and remove constraints for new players in learning how to play chess.

4 Assumptions

A1. Users of the board have knowledge of the starting positions and will set them up before the start of a new game.

A2. Users will not make illegal moves outside of beginner mode.

A3. Game termination as dictated by the rules of chess FIDE (2018)

A4. During gameplay, users will not take back moves. When a piece is placed on the board, that move is final.

A5. Users will have the knowledge of bluetooth setup and connection.

5 Constraints

C1	The cost should not exceed CAD $750
Rationale	This is the maximum budget alloted to this project as per the course requirements.

C2	The project must be completed by the winter semester of 2023
Rationale	This is the allocated time for this project as per the course requirements.

6 Scope

The system is called Chess Connect, and will include a software application and physical hardware device. The hardware will take the form of a chess set, and will collect and relay move and piece data. The device will convey the best moves for the specific board position, and will convey legal moves for specific pieces.The device will be connected to the software application, relaying and receiving relevant data. The software application will model and track the physical device, and will broadcast the data in an accessible format. The application will be constrained to a 2-dimensional model of the hardware device, showing a top-down view of the game.

In-scope items for the system include the following:

1. Modeling and tracking a chess game played using the Chess Connect hardware
2. Displaying and broadcasting the game state on the Chess Connect software application
3. Giving users an option to choose between beginner mode, engine mode, and normal mode
    - Beginner mode will display legal moves for individual pieces when a chess piece is picked up, and will warn the players when an illegal move is made
    - Engine mode will display the best moves as determined by a chess engine for the position

- Normal mode will disable the engine and beginner mode features. This is intended for regular play between experienced players

The following items are deemed to be **out of scope**:

1. FIDE (International Chess Federation) standards for tournament appropriate chess equipment
2. Tracking and support for alternate chess variants such as Chess960, Atomic Chess, King of the Hill. More information found here: List of Chess Variants.
3. Proper tracking of alternate starting positions like puzzles
4. Proper tracking of illegal moves and rule violations when warnings are ignored

7 Project Overview

7.1 System Context Diagram

The context of the system invloves two integrated but separate system components, as well as two distinct end users.

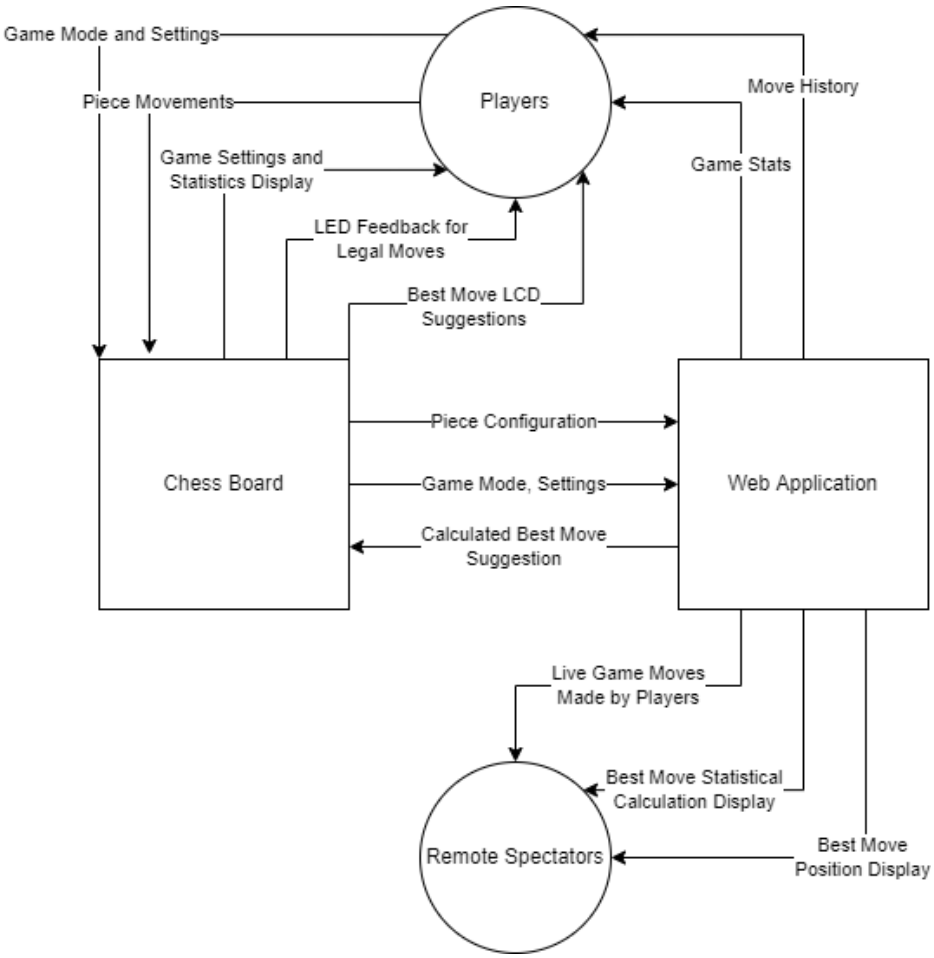

Figure 1: Overall System Context

7.2 Behaviour Overview

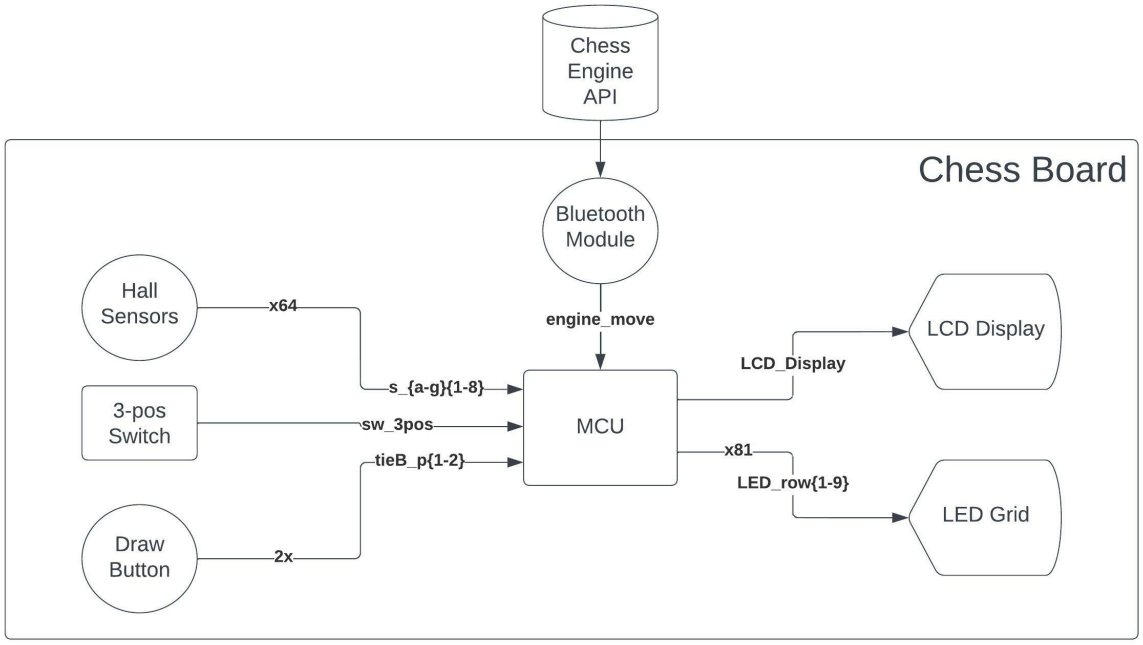

Figure 2: On-board System Context

7.3 Normal Operation

7.3.1 Description

The normal operation of the chess board involves configuring game mode and settings, reading the physical positions and identifiers (magnetic strength) of the pieces, lighting up corresponding LEDs and determining legal moves on the microcontroller. There will be three game modes: Normal Mode (no LED feedback), Engine Mode (best moves calculated by a chess engine and displayed by an LCD display) and Beginner Mode (legal moves displayed by LED when a piece is picked up). The microcontroller will also be simultaneously transmitting data to the server via Bluetooth and receiving responses in the form of "best" moves from the server. The server will be calculating these best moves in real time depending on the configuration of the pieces on the board sent from the chess board, and sending it back over Bluetooth every time it is queried while the game is in "Engine Mode". The server will also be communicating all of this information to a web application where users can tune in and watch the pieces and see game stats in real time.

7.3.2 Finite State Machine

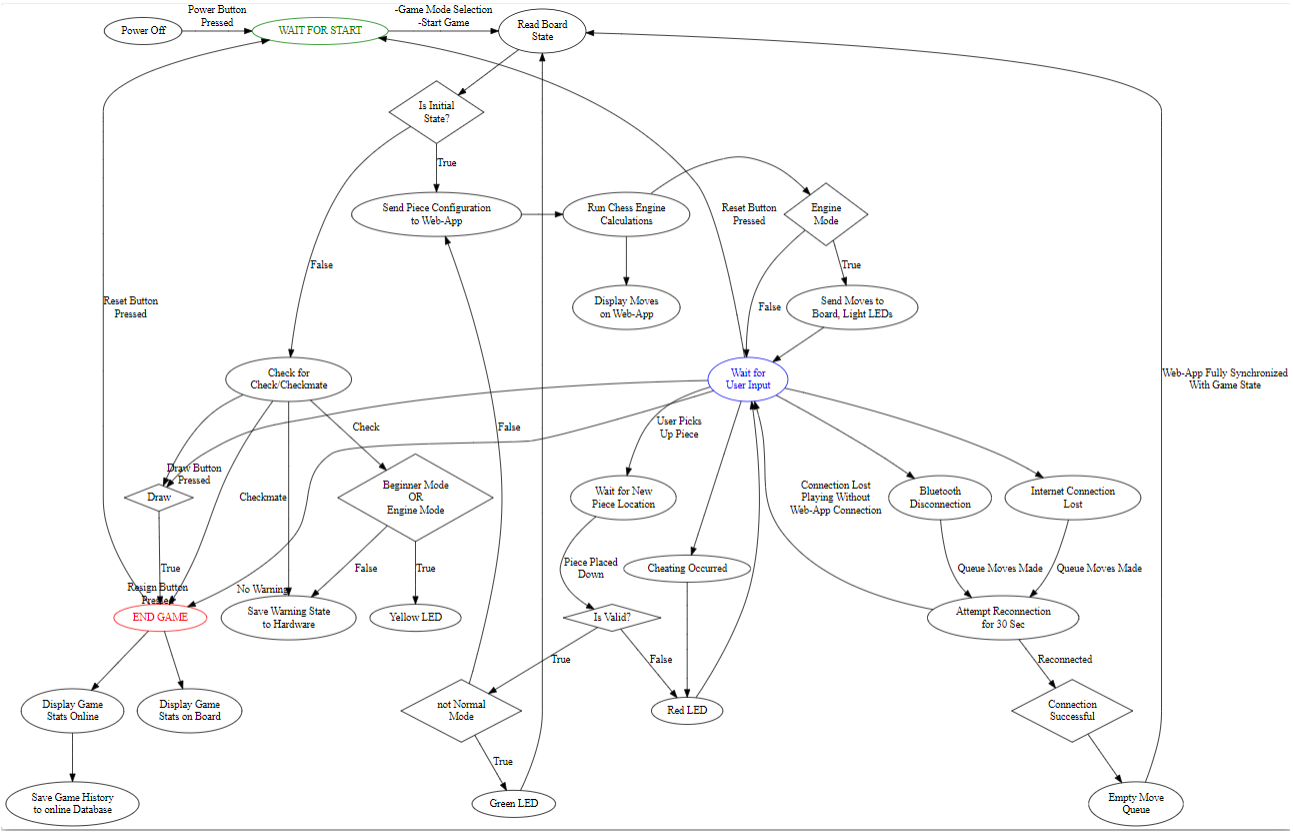

Figure 3: Finite State Machine Detailing Normal System Behaviour

7.3.3 Use Cases/Scenarios

7.3.3.1 Normal Mode

Users play the game without any LED feedback. This would typically be a formal match involving players that are familiar with the game. Moves would still be broadcast to the web application, and move history will still be recorded for future reference.

7.3.3.2 Engine Mode

Users play the game with LCD feedback indicating which pieces to move to which positions that would be statistically the most probable move to win the game. This game mode would be most likely played as a way to improve one's skill level and to gain a deeper understanding of the game.

7.3.3.3 Beginner Mode

Users play the game with LED feedback on lifting a piece. The piece lifted will be determiend by the hardware and corresponding legal moves will be lit up on the board. This game mode is designed to help beginners learn the game of chess. Legal moves will involve more complex strategies such as “en passant” and “castling”.

7.3.3.4 Studying Past Games

The web application will have a record of the moves made in past games. This can be used to study the game moves and understand scenarios which could have been handled differently to produce a more favourable outcome.

7.3.3.5 Tournament Play

Since the moves made in the game will be broadcasted live to the internet via the web application, tournaments can be held with spectators watching the game in real time.

7.4 Undesired Scenario Handling

7.4.0.1 Earthquake Scenario

Also known as the “sore loser scenario”, when some or all of the pieces are removed from their spaces within a short amount of time, the game will be ended as a draw.

7.4.0.2 Power Loss

When the power to the board is cut, the web application will end the game as a draw. The board will also have to be reconfigured back into the default starting positions for all pieces.

7.4.0.3 Bluetooth Disconnection

On bluetooth disconnection, the board will attempt to reconnect for 30 seconds. Upon the expiration of this timer, the game will end as a draw.

7.4.0.4 Internet Disconnection

On internet disconnection, the server will keep a history of past moves made. On reconnection, the game will be updated to reflect these moves as they are sent through the queue to the hosted interface.

7.4.0.5 Cheating

If pieces are moved or removed out of turn, the spaces which were affected will be lit up with red LEDs. When the board is returned to the state before cheating occurred, the game will resume as normal.

8 System Level Variables

8.1 Constants

Constant	Unit	Value
Chess board width	inches	12
Chess board length	inches	12
Chess board tile width	inches	1.5
Chess board tile length	inches	1.5
Supply Power to Board	Volts	110 VAC

8.2 Monitored Variables

Variable	Units	Description
s_a{1-8}	Volts	States of tiles a1 - a8 on the board. They are analog signals converted to digital and the state of the tile is determined. The possible states of each tile is empty, black/white pawn, black/white rook, black/white knight, black/white bishop, black/white queen, black/white king.
s_b{1-8}	Volts	States of tiles b1 - b8 on the board. ” ”
s_c{1-8}	Volts	States of tiles c1 - c8 on the board. ” ”
s_d{1-8}	Volts	States of tiles d1 - d8 on the board. ” ”
s_e{1-8}	Volts	States of tiles e1 - e8 on the board. ” ”
s_f{1-8}	Volts	States of tiles f1 - f8 on the board. ” ”
s_g{1-8}	Volts	States of tiles g1 - g8 on the board. ” ”
sw_3pos	Volts	The three-position switch is located on top of the board. It toggles between the beginner advice, engine advice and normal modes.
tieB_p{1-2}	Volts	The “draw” push-button for each player is located on the top of the board on their respective sides. When both players press their button the game is a draw.
engine_move	chess notation	The chess engine API provides best moves into the system.

8.3 Controlled Variables

Variable	Units	Description
LED_row{1-9}	Volts	A total of 81 LEDS will be located under the board. They are on the corner of each tile and illuminate based on conditions of the inputs.
LCD_Display	Volts	An LCD Display is located on the chess board to indicate best moves delivered by the engine.

9 Requirements

9.1 Functional Requirements

The system has two states, the Game Active State and the Game Inactive State. When in the Game Active State, the system can be in one of three possible "user modes". The user modes are Normal Mode, Engine Mode, and Beginner Mode.

Figure 4: Game State Diagram

9.1.1 Game Active State

9.1.1.1 Chess Board

GA1. Pressing the Resign/Draw button will change the system to the Game Inactive State.

GA2. Pressing the New Game button will have no effect.

GA3. Allow users to switch between user modes, choosing Normal Mode, Engine Mode, or Beginner Mode by using the User Mode switch on the side of the board.

GA4. The chess board follows the **Chess Board** section for the respective user mode being played.

9.1.1.2 Data Transfer

GA5. Data transfer follows the **Data Transfer** section for the respective user mode being played.

9.1.1.3 Web Application

GA6. Entering the Game Active State will reset the game state to the starting position.

GA7. Game termination of type stalemate or checkmate shall change the system to the Game Inactive State.

GA8. The web application follows the **Web Application** section for the respective user mode being played.

9.1.2 Game Inactive State

9.1.2.1 Chess Board

GI1. Pressing the New Game button will change the system to the Game Active State.

GI2. Using the User Mode switch at the side of the board will have no effect until the system is in the Game Active State.

GI3. Pressing the Resign/Draw button will have no effect.

9.1.2.2 Data Transfer

GI4. The board state data is not sent to the web application when a piece is moved.

9.1.2.3 Web Application

GI5. The web application will display the final game state upon game termination.

GI6. The web application will display a message with the game termination type (stalemate, checkmate, resignation, draw).

9.1.3 Normal Mode

9.1.3.1 Chess Board

NB1. The system shall store the position, colour, and type of the previously moved piece in the micro-controller.

NB2. Allow each player the ability to resign by holding down the Resign/Draw button for ENDTIME seconds located on their side of the board.

NB3. Allow the players the ability to draw by having them both hold down the Resign/Draw button for ENDTIME seconds located on their side of the board.

9.1.3.2 Data Transfer

ND1. The system shall send the micro-controller output over the chosen data transfer method as an input to the web application.

9.1.3.3 Web Application

NA1. The web application will receive data related to the previous move.

NA2. The web application will display the updated game board configuration with the data of the previous move.

NA3. In event of game termination (stalemate, checkmate, resignation, draw), the web application will display a message with the method of game termination. The system shall change to the Game Inactive State.

9.1.4 Engine Mode

9.1.4.1 Chess Board

EB1. The system shall store the position, colour, and type of the previously moved piece in the micro-controller.

EB2. Allow each player the ability to resign by holding down the Resign/Draw button for ENDTIME seconds located on their side of the board.

EB3. Allow the players the ability to draw by having them both hold down the Resign/Draw button for ENDTIME seconds located on their side of the board.

EB4. The system shall show the top engine moves on the LCD display.

9.1.4.2 Data Transfer

ED1. The system shall send the micro-controller output over the chosen data transfer method as an input to the web application.

ED2. The system shall send the web application engine moves to the LCD display over the chosen data transfer method.

9.1.4.3 Web Application

EA1. The web application will receive data related to the previous move.

EA2. The web application will display the updated game board configuration with the data of the previous move.

EA3. The system shall input the current chess game position to a chess engine.

EA4. The system shall use the chess engine to evaluate the position and calculate the best engine moves.

EA5. The system shall display the calculated engine moves on the web application.

EA6. In event of game termination (stalemate, checkmate, resignation, draw), the web application will display a message with the method of game termination. The system shall change to the Game Inactive State.

9.1.5 Beginner Mode

9.1.5.1 Chess Board

BB1. The system shall store the position, colour, and type of the previously moved piece in the micro-controller.

BB2. When a user picks up a piece, allow them to view all legal moves with green LED lights on valid squares.

BB3. If an illegal move is made, the tile that the piece is moved onto will display a red LED light.

BB4. Allow each player the ability to resign by holding down the Resign/Draw button for ENDTIME seconds located on their side of the board.

BB5. Allow the players the ability to draw by having them both hold down the Resign/Draw button for ENDTIME seconds located on their side of the board.

9.1.5.2 Data Transfer

BD1. The system shall send the micro-controller output over the chosen data transfer method as an input to the web application.

9.1.5.3 Web Application

BA1. Allow user to view instructions regarding the rules of chess on web application.

BA2. The web application will display the updated game board configuration with the data of the previous move.

9.2 Nonfunctional Requirements

9.2.1 Look and Feel Requirements

9.2.1.1 Appearance Requirements

LF1. The product shall use white, black, grey, and brown as its primary colours.

LF2. The product shall use green, red, and blue as its secondary colours.

9.2.1.2 Style Requirements

LF3. The product shall look and feel similar enough to traditional chess boards and chess pieces that the target audience will recognize the product as a chess set when encountering it for the first time. The level and speed of audience recognition achieved by the design shall be described following the procedure given in Section 5.2.1 of the VnV (Verification and Validation) Plan.

9.2.2 Usability and Humanity Requirements

9.2.2.1 Ease of Use Requirements

UH1. The system shall require the user to place chess pieces fully on their intended squares.

UH2. Physical hardware components of the system will not impede the user during play.

9.2.2.2 Personalization and Internationalization Requirements

UH3. The system will only display information in English.

UH4. The system will only use the Arabic numerals.

9.2.2.3 Learning Requirements

UH5. The product shall be able to be used by members of the public over with no previous training. Details on the learnability of the system shall be described following the procedure given in Section 5.2.2 of the VnV Plan.

9.2.2.4 Understandability and Politeness Requirements

UH6. All symbols and words shall be similar to historically used Chess symbols. Wall (2003)

9.2.2.5 Accessibility Requirements

UH7. The system shall follow guidelines for correct size and colour contrast ratio for text to the background as stated in the WCAG.

9.2.3 Performance Requirements

9.2.3.1 Speed and Latency Requirements

PR1. The average time between a user placing down a piece and the visual model response shall be small.

PR2. The maximum time between a user placing down a piece and the visual model response shall be small.

PR3. The average time between a user picking up a piece and the visual board indicator response shall be small.

PR4. The maximum time between a user picking up a piece and the visual board indicator response shall be small. The degree of speed for PR1 through PR4 shall be described following the procedure given in Section 5.2.3 of the VnV Plan.

9.2.3.2 Health and Safety-Critical Requirements

PR5. The system shall be properly grounded according to the Canadian Electrical Code. CSA (2021)

PR6. The maximum power on any single wire shall be within the safety limits described in the Canadian Electrical Code.

9.2.3.3 Precision or Accuracy Requirements

PR7. The software application game state will model the game state on the Chess Connect hardware with a high degree of accuracy. The level of accuracy shall be described following the procedure given in Section 5.2.4 of the VnV Plan.

9.2.3.4 Reliability and Availability Requirements

PR8. The product shall be available with a high degree of uptime. The level of availability shall be described following the procedure given in Section 5.2.5 of the VnV Plan.

9.2.3.5 Robustness or Fault-Tolerance Requirements

PR9. The software application shall maintain the game state if the connection between the software and hardware systems is interrupted.

9.2.3.6 Capacity Requirements

PR10. The software shall require computer memory to function effectively. The level of memory capacity required shall be described following the procedure given in Section 5.2.6 of the VnV Plan.

9.2.3.7 Scalability or Extensibility Requirements

PR11. The product must support the addition of new features and components.

9.2.3.8 Longevity Requirements

PR12. The product must be supported while the application remains deployed.

PR13. The product will depend on the continued support of packages and libraries.

9.2.4 Operational and Environmental Requirements

9.2.4.1 Expected Physical Environment

OE1. The hardware and software systems shall be close enough to each other to facilitate communication. The degree of proximity required shall be described following the procedure given in Section 5.2.7 of the VnV Plan.

OE2. The area shall be clear of potentially dangerous or harmful environmental factors.

9.2.4.2 Requirements for Interfacing with Adjacent Systems

OE3. The system shall interface with an external server to make requests to a chess engine.

9.2.4.3 Productization Requirements

OE4. The product shall be deployed to a public website where users may access it.

9.2.4.4 Release Requirements

OE5. The product will be tested for bugs and issues. These issues will be fixed and the application will be redeployed accordingly.

9.2.5 Maintainability and Support Requirements

9.2.5.1 Maintenance Requirements

MS1. The product shall be maintained actively by the developers until the Chess Connect team graduates.

9.2.5.2 Supportability Requirements

N/A

9.2.5.3 Adaptability Requirements

MS2. The software application will be able to be hosted on Apple, Windows, and Linux devices.

MS3. The product shall be accessible from any web browser.

9.2.6 Security Requirements

9.2.6.1 Access Requirements

SR1. Only the Chess Connect team are able to modify the software system.

9.2.6.2 Integrity Requirements

SR2. The product will not store game data after a game has concluded.

9.2.6.3 Privacy Requirements

SR3. The product will not store or collect user data.

9.2.6.4 Audit Requirements

SR4. Requirements shall be easy to follow and verify against both the system and the VnV plan in order to facilitate regular inspections.

9.2.6.5 Immunity Requirements

N/A

9.2.7 Political and Cultural Requirements

9.2.7.1 Cultural Requirements

PC1. The product will not use and terms or symbols that are deemed offensive to any culture.

9.2.7.2 Political Requirements

N/A

9.2.8 Legal Requirements

9.2.8.1 Compliance Requirements

LR1. The system shall comply with the Canadian Electrical Code CSA (2021).

9.2.8.2 Standards Requirements

LR2. The product shall follow WCAG.

10 Likely Changes

LC1. Reviewing past games allows players to evaluate their own positions with a better understanding on how to play in future games. We will use databases such as MySQL and MongoDB in order to store these games for users to view.

LC2. An additional user mode, Study Mode, that will allow users to set up puzzles that do not start from the default position. It will allow users to practice a specific phase of the game, including the ability to take back moves without having to play out a full game with an opponent. This mode is meant to be used individually, also allowing a user to use and study the board without the necessity of another player.

LC3. Online chess has extensive communities on existing platforms. To promote code reusability, the Chess Connect online platform will interface with popular existing websites. Users of the board have the capability to share their games with a larger community via these platforms in addition to the custom web application. These sites include, but are not limited to, Chess.com and Lichess.

LC4. An additional game setting to enable either white or black to be a "computer" opponent will be implemented in Engine Mode. The user will be able to set a skill level for the computer, and moves will be displayed as squares lit up by the on-board LEDs.

LC5. The backend/database programming languages and frameworks previously listed are subject to change depending on the use of the data transfer functionality, in which comptibility to the hardware would play a role in picking our technology stack.

LC6. The load capacity of our system is initially set low for the first edition of this application. As demand for the application increases, the load capacity must also increase to support a larger user base.

LC7. The method of data transfer is subject to change depending on interfacing between the hardware and software. Examples of these are Bluetooth and USB connected to the board.

LC8. The dimensions of the chess board are subject to change based the 3D-printed designs we use for the pieces.

LC9. When pieces are lost, power is cut or data transmission is cut, the current result is ending the game as a draw. Adding the ability for the board to start from any state would allow the players to resume from where they left off. This would require several other options such as "which colour starts", "pausing the game" and "resume from last saved configuration".

11 Unlikely Changes

UC1. Since overarching idea of the project is integration between over the board and online chess, a web application will be necessary.

UC2. There must be sensors to locate the pieces in order to solve tracking and object detection.

UC3. The most popular and easy to use chess engine is Stockfish, which is open source and includes a well-written API documentation. This will likely be the engine we will be using for predicting future moves.

UC4. As a learning application, giving beginners the ability to learn quickly is a big priority, so the application will include a beginner mode to help familiarize players with the movement of the chess pieces.

UC5. Although the back-end/database technologies are subject to change, the front-end will make use of our past members' experience in user interface design. For this reason, the technologies previously mentioned for front-end are unlikely to change.

12 Traceability Matrix

A1	GI1
A2	BB3
A3	BB2, BB3
A4	UH1
BA1	UH5
7.3.2.1	NB1, NB2, NB3, ND1, NA1, NA3,
7.3.2.2	EB1, EB2, EB3, EB4 ED1, ED2, EA1, EA2, EA3, EA4, EA5, EA6
7.3.2.3	BB1, BB2, BB3, BB4, BB5, BD1, BA1, BA2
7.4.0.4	PR9
7.4.0.5	PR9
PR11	LC1, LC2, LC3, LC4, LC5, LC6, LC7, LC8, LC9
LF2	BB2, BB3

A Reflection

A.1 Skills for Success and Learning Approaches

SS1. Proper unit tests must be made for the integration between the web application and microcontroller. The web application will be made using React so this a skill that will have to be learned. This required knowledge will be acquired by reading documentation on open-source projects with similar use of technology. This will be done by member Jonathan Cels, Rupinder Nagra, and Alexander Van Kralingen.

SS2. The back-end system will be created using Python and MongoDB. These skills will have to be learned by reading online documentation and blog forums. This will be done by Rupinder Nagra.

SS3. Continuous integration/deployment will have to be integrated into the project. This skill will be learned by review the tutorials related to this topic as well as reading online documentation. This will be done by member Alexander Van Kralingen.

SS4. The microcontroller will need to be setup to use Bluetooth to communicate between the web application and the microcontroller. This will be learned by reading the documentation for the Bluetooth hardware chosen. The skills required for this will be learned by members Jonathan Cels, Joshua Chapman, and Alexander Van Kralingen.

SS5. The microcontroller will need to be programmed with the rules of chess. The rules of chess can be found through online resources. This will be done by member Joshua Chapman.

SS6. Utilisation of a Hall sensor to identify unique chess pieces using magnets is required. Knowledge of installing and soldering hardware components will have to be learned. This knowledge will be acquired by reading blog posts by people who have used these components and watching tutorial videos on media platforms such as YouTube. This will be done by member Arshdeep Aujla and Joshua Chapman.

SS7. The relation between the inputs and outputs will have to be mapped using Karnaugh Maps and State Machine Tables. This will be learned by reviewing related course materials from previous/current years. This will be done by member Arshdeep Aujla.

References

CSA. *Canadian Electrical Code*. CSA Group, 2021.

Interal Chess Federation FIDE. Fide laws of chess. https://www.fide.com/FIDE/handbook/LawsOfChess.pdf, 2018.

List of Chess Variants. List of chess variants, Sep 2022. URL https://en.wikipedia.org/wiki/List_of_chess_variants.

Bill Wall. History of chess sets and symbols. https://www.chesscentral.com/pages/chess-sets-pieces-boards/a-history-of-chess-pieces-and-chess-sets.html, 2003. Accessed: 2022-10-04.

WCAG. Web content accessibility guidelines 2 overview. https://www.w3.org/WAI/standards-guidelines/wcag/#intro, 2018. Accessed: 2022-10-05.