

VR-CHESS

VR-Chess Team Members

Team Member	Role
Eric Smrkovsky	Project Manager
Christian Leon	Back End Developer
Brett Harris	Graphics Design and Visuals
Jose Fernando Jimenez Chavez	Lead Technical Designer
Jacob Miller	Front End Developer

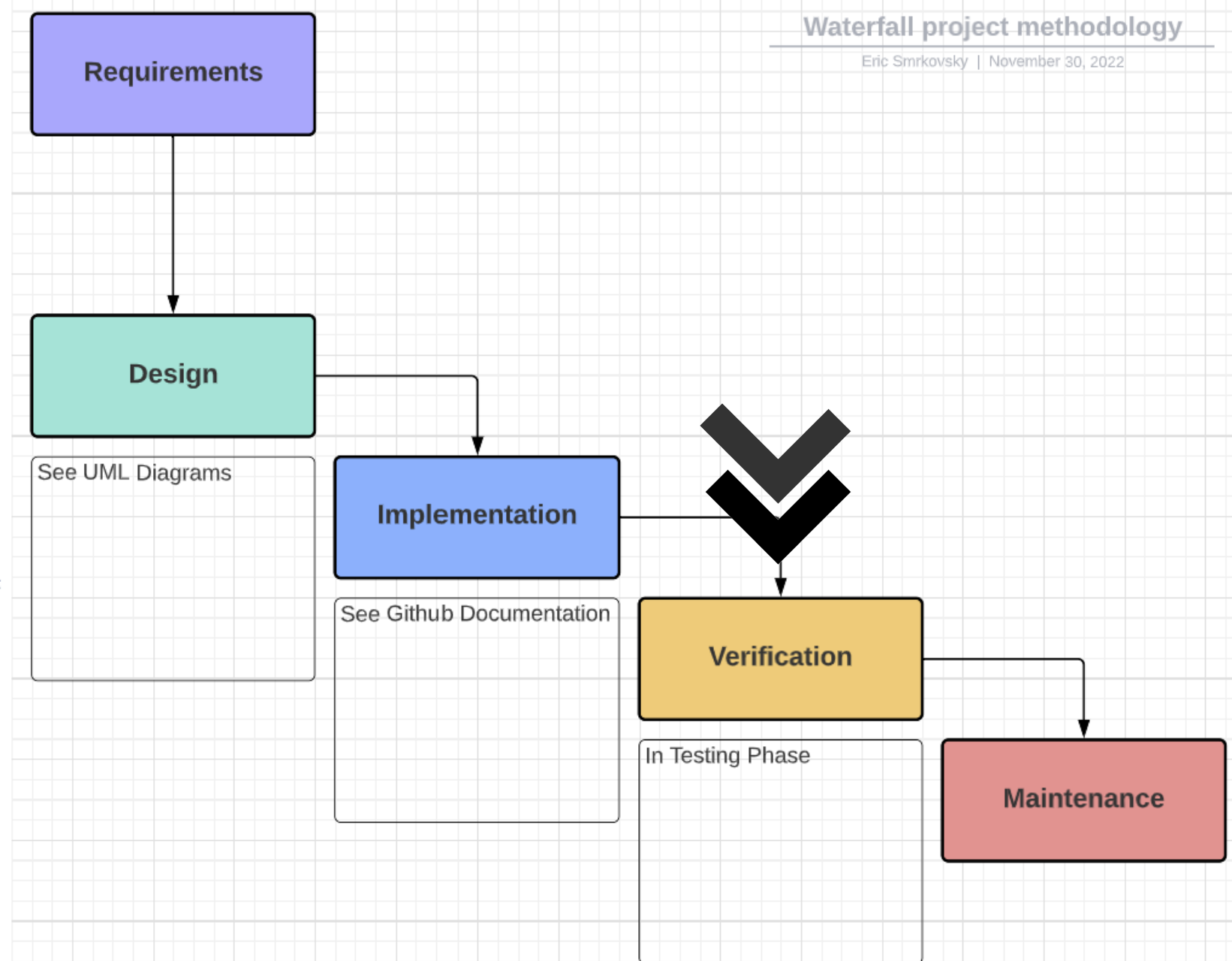
Welcome to the final presentation for VR-Chess
In the verification step we asses the product

VR-CHESS

VR-Chess Team Members

Team Member	Role
Eric Smrkovsky	Project Manager
Christian Leon	Back End Developer
Brett Harris	Graphics Design and Visuals
Jose Fernando Jimenez Chavez	Lead Technical Designer
Jacob Miller	Front End Developer

- FR1: Store User Information
- FR2: Account Creation
- FR3: User Login
- FR4: Store Password Securely
- FR5: Multiple Environments
- FR6: Multiple Chess Board Designs
- FR7: Choose Piece Color
- FR8: Google Authentication
- FR9: Horoku Cloud Deployment
- FR10: Multiple Avatars
- FR11: Store User Preferences
- FR12: Start VR Scene
- FR13: Access User Preferences
- FR14: Record Game Move History
- FR15: Store Game Move History Log
- FR16: View Game Logs
- FR:22 Control Avatar in Virtual Environment
- FR23: Take a Seat at Table by peice selection
- FR24: Board: Preset Positions of Pieces
- FR25: Board Game Logic: King
- FR26: Board Game Logic: Queen
- FR27: Board Game Logic: Bishop
- FR28: Board Game Logic: Knight
- FR29: Board Game Logic: Pawn
- FR30: Board Game Logic: Rook
- FR31: Board: Piece Movement
- FR32: Board Game Logic: Kill
- FR33: Board: Store Killed Pieces



Welcome to the final presentation for VR-Chess
In the verification step we asses the product

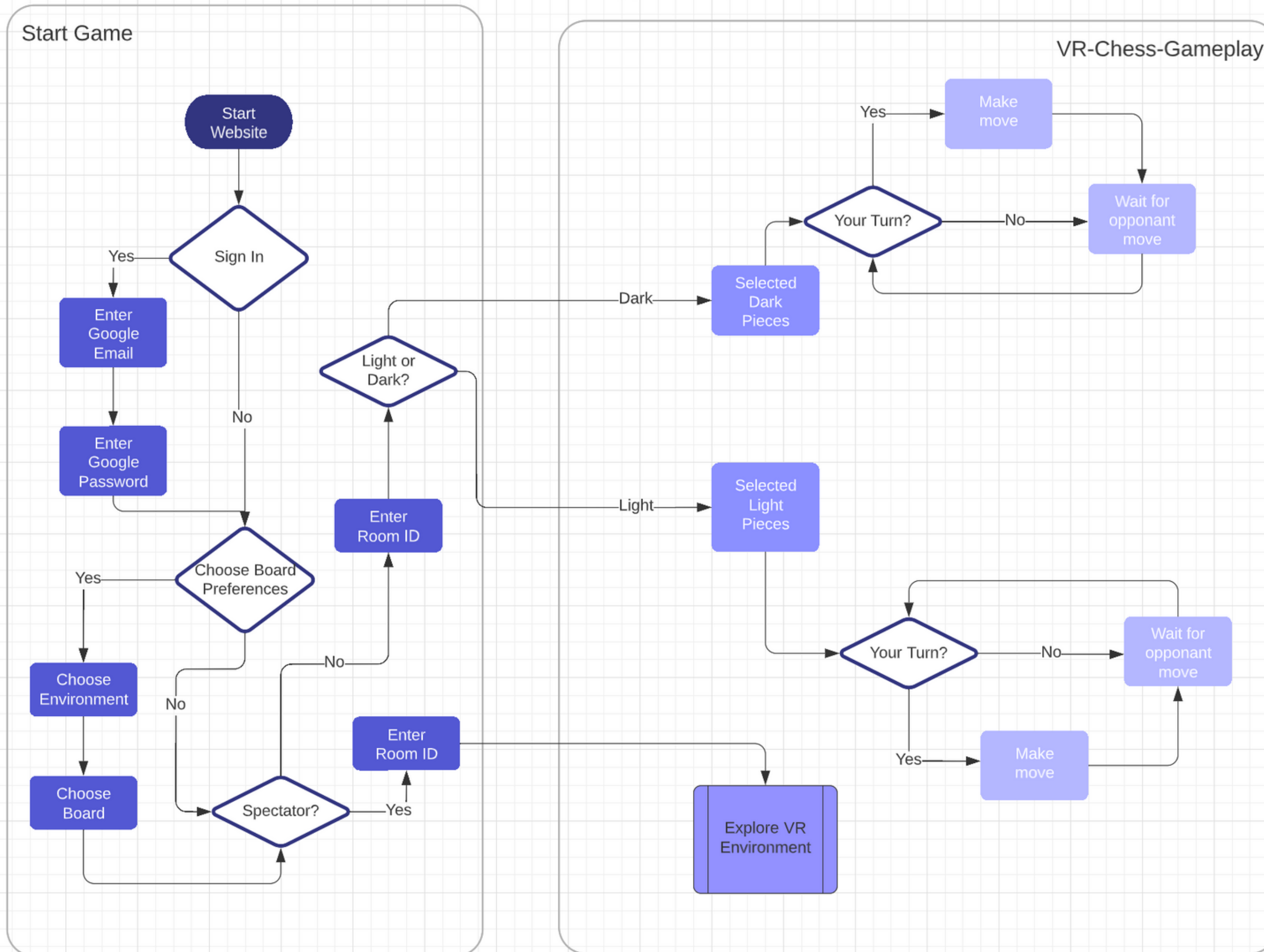
VR-CHESS

The Value of VR-Chess

- Relaxation/Mental Health
- Connect with Friends
- Practice Chess Skills
- Available Cross-Platform
- Simplified Access

User Journey Flowchart

Eric Smrkovsky | November 30, 2022





We have fully developed the semi-immersive VR-Chess experience.

Main Features

3d Graphics

Turn Based Game Logic

Online Networking

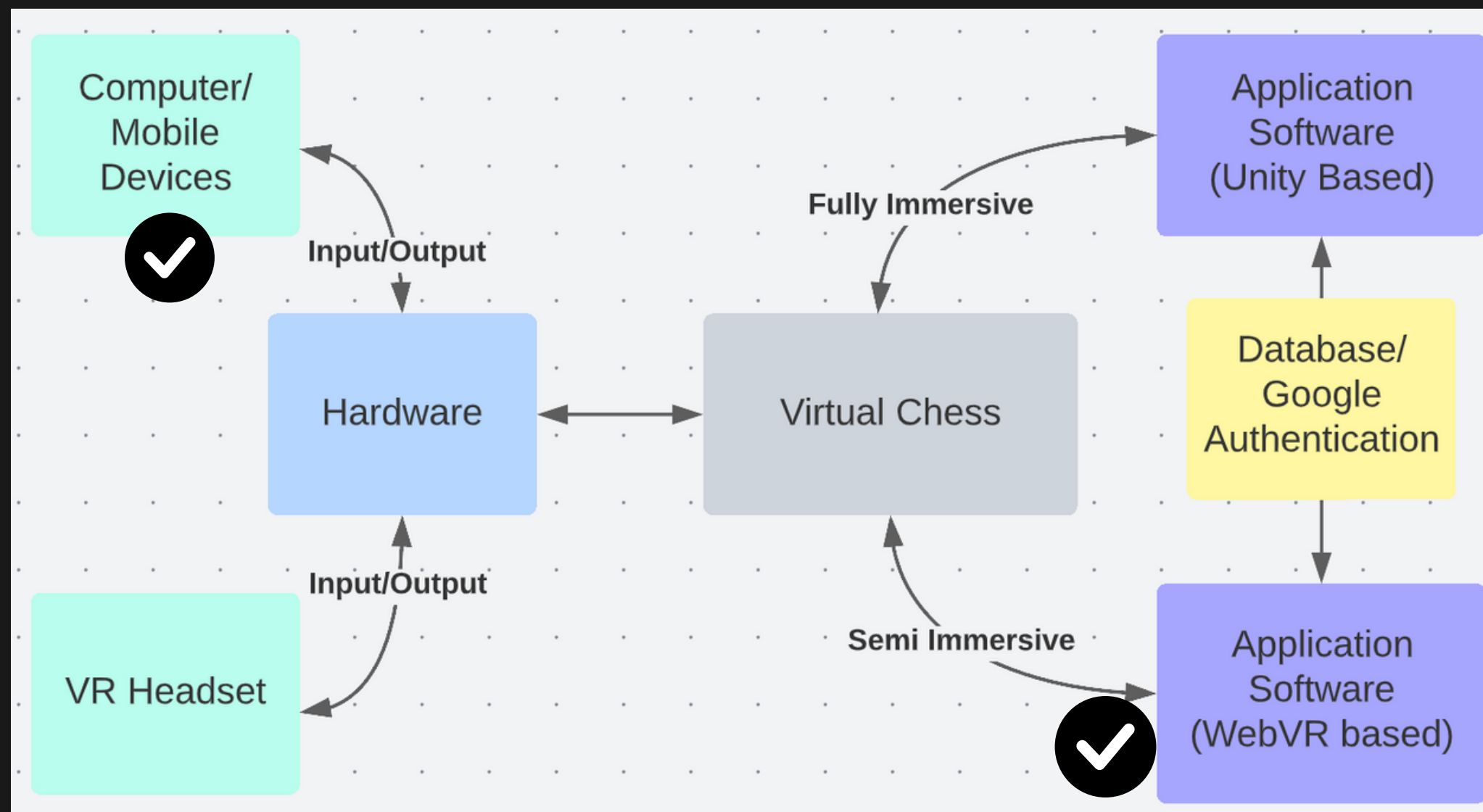
Google Authentication

Multiple Environments

Multiple Piece Designs

Spectator Mode

Dynamic Room Loading



Functional Requirements

- **FR5: Multiple Environments**
 - Description: The user must be able to choose from three or more choices for the VR environment.
 - Dependency: FR3
- **FR6: Multiple Chess Board Designs**
 - Description: The user must be able to choose from two or more Chess board designs available.
 - Dependency: FR3
- **FR7: Choose Piece Color**
 - Description: The user must be able to choose from white, black, green, red, yellow, and blue for the piece color.
 - Dependency: FR3
- **FR8: Google Authentication**
 - Description: The software must utilize the Google Login API.
 - Dependency: None
- **FR9: Horoku Cloud Deployment**
 - Description: The software must utilize the Google Login API.
 - Dependency: None
- **FR10: Multiple Avatars**
 - Description: The user must be able to choose from multiple avatars.
 - Dependency: FR3
- **FR11: Store User Preferences**
 - Description: The software must store the user's preferences.
 - Dependency: FR5, FR6, FR7, DB
- **FR12: Start VR Scene**
 - Description: The user must be able to start the VR scene.
 - *Dependency: FR5, FR6, FR7, FR10, FR13
- **FR13: Access User Preferences**
 - Description: The user must be able to choose to load saved preferences.
 - Dependency: FR5, FR6, FR7, FR10, FR11

- **FR23: Take a Seat at Table by peice selection**
 - Description: The user must be able to sit virtually at the chess table when the peice is chosen.
 - Dependency: FR22
- **FR24: Board: Preset Positions of Pieces**
 - Description: The board must have the game pieces present when the user sits at the chess table.
 - Dependency: FR23
- **FR25: Board Game Logic: King**
 - Description: King piece logic must be designed to follow standard Chess rules.
 - Dependency: Chess Game Rules
- **FR26: Board Game Logic: Queen**
 - Description: Queen piece logic must be designed to follow standard Chess rules.
 - Dependency: Chess Game Rules

- **FR39: Board Game Logic: Piece Ownership**
 - Description: Only the player who owns the pieces can move them.
 - Dependency: FR31
- **FR40: Dynamic Rooms**
 - Description: The software must allocate resources for a new room when new game is started.
 - Dependency: FR23
- **FR41: Dynamic Room Code**
 - Description: The software must assign a room code for other players to join game.
 - Dependency: FR40

Contributions: Eric

Project Managment

- Documentation
- Project Board
- Presentations
- Evaluating Requirements
- UML Design
- Project Idea/Planning
- Team Managment

Big Thanks to the
VR_Chess Team!

Contributions: Brett

Graphics and Visuals

- Developed models with Blender
- Used A-Frame Environment Component and custom lighting for backgrounds
- Set movement dynamics with Three.js
- Developed custom dynamic scene loading components for A-Frame

Challenges:

- Ran into limitations with each framework
- 10% of time modeling setting up pretty scenes, 90% of time programming and debugging the implementation
- Learning Blender, Three, A-Frame, and JS on the fly

Contributions: Jacob

Movement & Gameplay

Repo Component

Player turns
Player position
Spectator mode

Chess-Piece Component

Board position
Special moves

Piece-Interaction Component

General movement
Move validation
Capture & Graveyard

Challenges:

- Learning JavaScript's DOM & Event-Listeners along with AFrame's Entity-Component System
- Constantly debugging a project that grew increasingly larger & interdependent

Contributions: Jose

Lead Technical Designer:

- Set up Networked A-Frame project and integrated existing code to create multiplayer experience, hosted on Heroku. Also added Google OAuth login. Minor A-frame scene work
- Lead team to complete core gameplay functionality first
 - Used NAF as it was easy to integrate into existing codebase
 - Used Heroku as it was the quickest to deploy an instance, and also make automatic deployments from github a breeze, we were actually able to fix a small issue during milestone 2 during someones presentation, would not have been as easy without CD tools
 - Added Google OAuth for optional logins.
- Problems?
 - Was lead to believe Glitch would be a great option, it's not for anything more than a minor 1 page/a-frame scene project. Very quickly outgrew it.
 - Very strange networked template issue.
 - Felt as if we were "boxed in" to node.js and made trying to add on a simpler PHP log in system a little more difficult.

Contributions: Chris

Database & Login System

- Utilized mongoDB to create database for users.
- Created Schema to store usernames, passwords, and email address.
- Created forms to login, logout, and change password.
- Utilized "POST" method for transmitting this data as it is more secure than "GET" method.
- Utilized Express for managing these types of requests.
- Interconnected files with VR Chess.
- Logging in to play can be made optional.

Contributions: Chris Database & Login System

Register Now

E-mail

Username

Password

Confirm Password

Register

Already Registered? Login



QUERY RESULTS: 1-2 OF 2	
	<pre>_id: ObjectId('638761d9e5eba43ade403469') unique_id: 1 email: "chris@gmail.com" username: "chris" password: "test" passwordConf: "test" __v: 0</pre>
	<pre>_id: ObjectId('63876827aa40f800165b54a5') unique_id: 2 email: "josh@gmail.com" username: "josh" password: "test" passwordConf: "test" __v: 0</pre>

Watch Live Demo

Jacob VS Brett

Eric in Spectator Mode



THANK YOU!
PLEASE TRY
PROOF OF
CONCEPT!

Google "vr chess csci 150"

Click:

[https://github.com > Ericesmrk > VR-Chess](https://github.com/Ericesmrk/VR-Chess)

[Ericesmrk/VR-Chess: Term project for CSCi 150 ... - GitHub](#) ✓

Then click ->



click photo to play with prototype

!!!VR CHESS!!!