**Software Requirements and Design Document**

**For**

**Group Phaze 5**

Version 2.0

**Authors**:

Faith Miller

Mackenzie Knight

Eduardo Antonini

Tristan Garcia

Katie Rombeiro

**1.** **Overview (5 points)**

A simplified version of the game Phase 10 put into a web-game version that allows for 2-6 players to play. This new game is called Phaze 5 and it includes the game play of two of the phases from the original Phase 10 game. The game is run through localhost using the Atom platform. The programming languages React, Javascript, HTML and CSS make up the game implementation and the front-end implementation.

# **2.** **Functional Requirements (10 points)**

A functional requirement that was completed during this iteration for the frontend was figuring out how to update the state of the number of players in the game (1). This was a high priority because it is necessary for us to know the number of people that will be playing the game. While completing this requirement, we figured out how to print out the corresponding number of player username textboxes on the Home Page depending on the number of players submitted. For example, if a user inputs that there will be 2 players and clicks the submit button, then two textboxes will appear for the user to type in the username for each player. Knowing how many players and what the names of each player is essential because this depends on how many player views will be needed when playing the game and the name of the current player will appear on the top of the screen on the Play Game page. Another functional requirement that was completed during this iteration for the frontend was figuring out how to create modals to take in and grab player input of game information (2). This was a high priority because taking in user information from the web app and grabbing it within the frontend code is essential for communication with the backend to play the game. Another functional requirement is making the web app look appealing to the eye (3). During this iteration, this was a low to medium priority. We did our best to make the interface of the app look good while also keeping it simple enough to make sure we are able to finish the web app on time. We will make the appearance of the interface more of a priority once we have a functioning app.

A functional requirement that was completed during this iteration for the backend is getting the game to function as it functions in a normal game of Phase 10 (4). These functional requirements were a high priority, because the game must function properly, and it must mimic the actual game of Phase 10. There are many smaller functional requirements that go into getting the game to function properly, but the essential ones are outlined below. The system must add and remove cards from a player's hand and allow them to make the appropriate plays for each phase by placing cards down on their “board.” Players must also be able to play cards on other players’ boards once they have completed their phase for that round. Each completion of a phase must match the criteria for that round. For example, if a player is on phase 1, two sets of three must be played. Both sets of three must include at least three cards of all the same number or wild cards. When playing on another player’s board, any cards must also match their specific play for their phase. For example, if someone wants to play on another player’s set of three and they have placed down three fives, the only cards they are allowed to play are fives and wild cards. Another smaller functional requirement that goes into getting the game to function properly is saving the state of each player between rounds. For example, if two players finish phase 1 during the first round and two players do not, those states must be saved. Also, the number of points against each player must be saved between rounds.

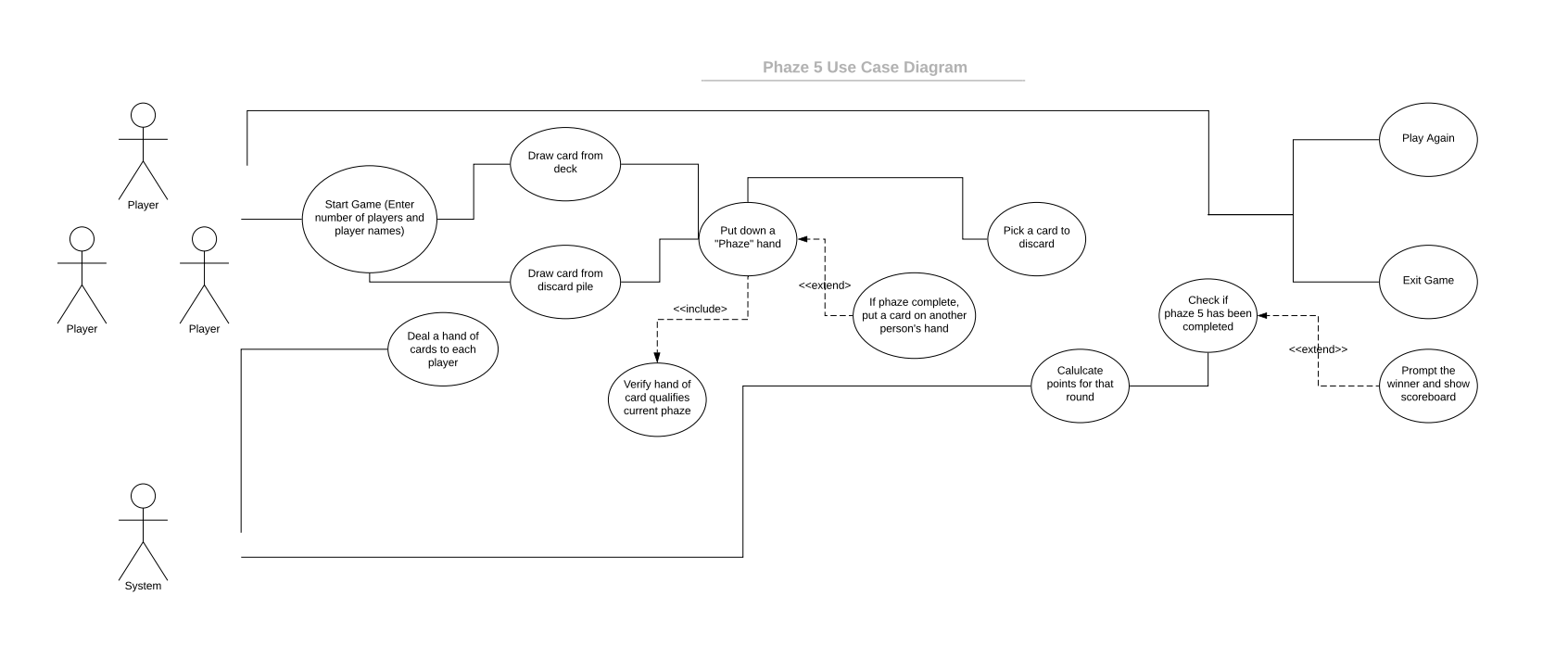
# **3.** **Non-functional Requirements (10 points)**

The software quality and performance must be stable enough to support the multiplayer game in its entirety. The system must be reliable enough to store player information, such as the running score against each player and a state indicating what phase each player is on in any given round. Safety and security are not requirements for our game because there is no sensitive information stored in the game. Additionally, there are no efficiency requirements for our game, even though we have been coding as efficiently as possible.

While coding the user interface of the project, we made sure to try and fix any Warnings that were shown in the Inspect on Google Chrome because this allows us to understand that the performance of our program is as stable as possible. If there are any unfixed Warnings, then they could potentially cause a performance or reliability issue. In addition, the usernames of the players are only being saved for the time that the current game of Phaze 5 is running for. So, when a new game is started, the usernames are wiped and replaced with usernames. So, if a user chooses a personal username, it will not be saved in any database. This will allow for all users to feel safe when playing our game.

**4.** **Use Case Diagram (10 points)**

Link to PDF version of diagram: <https://www.lucidchart.com/publicSegments/view/bb20ea11-da9a-4a5b-aa11-6dc4c68ffb69/image.pdf>



**5.** **Class Diagram and/or Sequence Diagrams (15 points)**

**

**6.** **Operating Environment (5 points)**

More and more it looks like we won’t attain our original plan of deploying our game on a real website to allow the players to interact remotely. Regrettably, because of a number of technical difficulties that we failed to foresee, the group is now working under the assumption that we will end up with a game that requires the users to pass a laptop around so each one can take their turns at playing. As far as the operating environment of our game, it should not make a difference what kind of computer would run it — Windows, Mac, or Linux —, as long as that computer is compatible with some browser able to run our game in the localhost. By the time we complete the project, the game will surely run on a Mac computer through the Safari or Chrome browsers, and a Windows computer through the Chrome browser. Other operating systems and internet browsers are candidates to the list of supported environments, but thorough compatibility cannot be assured at this time.

**7.** **Assumptions and Dependencies (5 points)**

There are still a lot of assumed factors that could affect the end result of our project. One of the issues we faced was how we originally planned to use five phases for the game. It came to our attention through the coding process that that many phases would take hours to complete in just one test due to the long length of the game. We had to switch to a smaller amount of phases to use within the game to lower the time it takes for all of us to test our code, whether it be front-end UI code or back-end game implementation code. Another issue we currently are facing (after the completion of the back-end code) is due the overall lack of knowledge of React code, therefore causing us to worry on how to get the “connection” between the front-end UI and the back-end game implementation and Player class data. Another assumed factor is human resource availability, due to the switch to online classes for the rest of the semester we have limited in-person communication therefore causing all of the team members to use the communication application Zoom to talk about the project and show portions of code. Some mandatory dependencies are researching more on the React language so we can all comfortably write the code for the connection aspect of the game program. This is pretty time-dependent due to it being the final step of the project and our forever decreasing time limit on the availability to work on the project. Still, we do not use specific code from another project but continue to turn to the use of the internet for explanation of certain aspects within the project that we are not fully knowledgeable of.