JAR Chess Game Use Case Survey

# Actor Summaries

## Player

Players will use the game to play matches and access saved historical matches.

* Speed - The application needs to be responsive or players will find other games to play.
* Ease of use - The UI needs to be intuitive. All components of the application need to be easily understood and managed without training or external documentation. Most players won’t bother putting that much effort to learn to use the app.
* Security - While there is not much for sensitive data, the app needs to protect players’ username\password combinations and email addresses.
* Privacy – Players don’t want their information in the hands of anyone they didn’t give it to, and they won’t want us monitoring their location or other private data without cause and consent.
* Availability – If any aspect of the game is not available at any given time, that will make the players look elsewhere for entertainment.

## Opponent (secondary actor)

The opponent is entity that the player is playing against.

# Use Case Summaries

## Login to Account

The goal of this use case it to allow the player to log into their account in order to access additional online functionality. The player tells the system that they wish to login. The system asks for login credentials. The player provides the information. The system verifies the information and logs the player in.

## Play Match

## The goal of this use case is to play a chess match. The player tells the system what type of match they wish to play. The system finds an opponent if necessary. The system starts the match. Each participant takes turns making moves until a match ending condition occurs. The result of the match is shown to the participants.

## Manage Account

The goal of this use case is to permit a user to view and make changes to their member account. The user prompts the system to manage their member account. The system displays the user’s current member account data. The user prompts the system to make changes in their member account. The system validates the input proposed by the user and retains the effective changes. See Appendix A for details.

(JAR Chess Use Case Diagram on next page)

# Use Case Diagram

Change log:

* Removed the logged in and logged out actors
* Removed the Friend secondary actor
* Simplified the Use Case Diagram and updated the Use Case Summaries.
* Added this change log
* Changed version from 1.0 to 2.0