**Use Case UC1: Register User**

**Scope:** Chad game application

**Level:** User account

**Primary Actor:** Person.

**Stakeholders and Interests:**

* Person: Wants to add themselves as a user to the system so they can play games

**Preconditions:** Person must have an email that has not already been used in another registration

**Success Guarantee:**

* The person has an account as a registered user and may login to the system using their email and password. The system will remember the account even if the system is shut down or crashes.

**Main Success scenario:**

1. Person enters a valid email that is not used by another user
2. Person enters a nickname that has not been used by another user
3. Person enters a password to secure their account
4. A user account is created for the person

**Extensions:**

4a. An invalid email was entered, the person is notified of the issue and the system resets to step 1

4b. The entered email is already being used, the person is notified and offered the options to recover the account currently using the entered email or to return to step 1

4c. The entered nickname is already in use, the person is asked to enter a different nickname

**Special Requirements:**

**Technology and Data variations List:** none.

**Frequency of Occurrence:** Account creation happens once per user.

**Miscellaneous:** If user selects to allow the app to save data to their device the login process is automatic, otherwise they must do so manually.

**Use Case UC2: Login User**

**Scope:** Chad game application

**Level:** User account

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: Wants to log in to the system

**Preconditions:** User must have completed Register User use case

**Success Guarantee:**

* The user is logged in and may access system features other then login

**Main Success scenario:**

1. User enters the email linked to their account
2. User enters the password set for the account
3. The system preforms the login operation for the user

**Extensions:**

3a. The email entered was not correct, the system will notify the user and ask them to login again

3b. The password entered is incorrect, the system will notify the user and ask them to login again

3c. The system is currently down, the user is notified and asked to try again later

**Special Requirements:**

**Technology and Data variations List:** none.

**Frequency of Occurrence:** Often

**Miscellaneous:** Find out how this interacts with the google authentication system we are using

**Use Case UC3: Create Game**

**Scope:** Chad game application

**Level:** User games

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: can start and play games with other users

**Preconditions:** Login User must be completed

**Success Guarantee:**

* A new game is created and the user will be a player of that game

**Main Success scenario:**

1. A user requests the system makes a new game
2. The game is created and the requesting user is added as the first player of the game

**Extensions:**

2a. The game creation system is currently off line, the user is notified and asked to try again later

**Special Requirements:**

**Technology and Data variations List:**

**Frequency of Occurrence:** Very often

**Miscellaneous:**

**Use Case UC4: Invite Users to Created Game**

**Scope:** Chad game application

**Level:** User games

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: wants to allow other users to join their games

**Preconditions: T**he user has at least one game where they are the only player

**Success Guarantee:**

* An invitation is presented to all invitees on the invitation list and they may then Accept Invitation

**Main Success scenario:**

1. The user selects a game where they are the only player
2. The user selects one or more users to become the second player in the game
3. Once satisfied with the invitation list the user requests the invitations to be sent
4. The system sends an invitation to all users on the list

**Extensions:**

4a. The invitation system is down, the user is notified and asked to repeat the invitation later

**Special Requirements:**

**Technology and Data variations List:**

**Frequency of Occurrence:** Very often

**Miscellaneous:**

**User Case UC5: Accept Invitation**

**Scope:** Chad game application

**Level:** User games

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: wants to accept or reject the invitation to the game. If accepting the invitation the user wishes to join and play the game they are being invited to.

**Preconditions:** The user must have received an invitation to a created game

**Success Guarantee:**

* The user is added as a player in the game and may make moves after inviting user has played

**Main Success scenario:**

1. The user accepts the invitation
2. The user is assigned as the second player of the game
3. The game is started and moves may be made
4. The inviting user is notified that the invitation has been accepted

**Extensions:**

1a. The user rejects the invitation and will no longer be able to join the game

1b. The user preforms not action on the invitation for greater then the timeout period, the invitation is removed and the user will no longer be able to join the game

2a. The game already has a second player, the user is notified and will no longer be able to join the game

2b. The game system is down, the user will be notified and asked attempt to join the game later

**Special Requirements:**

**Technology and Data variations List:**

**Frequency of Occurrence:** Very often

**Miscellaneous:**

**Use Case UC6: Access User Games**

**Scope:** Chad game application.

**Level:** User games

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: must be able to access games he/she is part of. The user must be able to save and make progress.

**Preconditions:** Login User is completed, user must have at least one active game

**Success Guarantee:**

* The user may make moves in the game they accessed

**Main Success scenario:**

1. User selects a game from the list of active (unfinished, non-abandoned) games
2. User reenters the game and may make moves if it is their turn

**Extensions:**

2a. The game system is down, the user is notified and asked to try again later

**Special Requirements:**

**Technology and Data Variations List:**

**Frequency of Occurrence:** Often

**Miscellaneous:**

**Use Case UC7: Leave game**

**Scope:** Chad game application.

**Level:** User games

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: wants to leave a game in progress but be able to rejoin and make progress later

**Preconditions:** User has completed Login User and is actively playing a game

**Success Guarantee:**

* User is no longer playing the game but may reenter the game with at most one move being made in the meantime

**Main Success scenario:**

1. User requests to leave the game
2. User is marked as not actively playing the game
3. If it is the users turn while they are away the game will not progress until the user completes Access User Games for the game

**Extensions:**

3a. The user does not complete Access User Games for greater then the timeout period, the game is marked as abandoned and may not be accessed

**Special Requirements:**

**Technology and Data Variations List:**

**Frequency of Occurrence:** Often

**Use Case UC7: Quit game**

**Scope:** Chad game application.

**Level:** User “goal”.

**Primary Actor:** User.

**Stakeholders and Interests:**

* User: must be able to quit games he/she is part of. The game must be deleted from the list of games he/she is part.
* Company: fast and easy process for the user.

**Preconditions:** user is identified and authenticated.

**Success Guarantee:**

* User can quit and shutdown a game. User is no longer playing the game.

**Main Success scenario:**

* The user logs into the system. Once the user is identified, the user can choose to start a game and after some time is able to successfully quit the game.

**Extensions:** none.

**Special Requirements:**

* Touch screen UI large enough. Text must be visible. Recover data if system crashes.

**Technology and Data Variations List:** none.

**Frequency of Occurrence:** nearly continuous.

**Miscellaneous:** can the user quit games on different platforms e.g. iPad, laptop or smart phone.

**Use Case UC8: Un-register from the system**

**Scope:** Chad game application.

**Level:** User “goal”.

**Primary Actor:** User

**Stakeholders and Interests:**

* User: must be able to quit from the system. The profile information must be safely deleted from the system
* Company: easy and fast to quit. It no longer has access to the personal information of the user.

**Preconditions:** user is identified and authenticated.

**Success Guarantee:**

* The user no longer has access to any features of the game.

**Main Success scenario:**

* The user logs into the system. Once the user is identified, the user can view his/her profile. The user deletes the profile.

**Extensions:** none.

**Special Requirements:** none

**Technology and Data Variations List:** none.

**Frequency of Occurrence:** nearly continuous.

**Miscellaneous:** none.

**Use Case UC9: Record of games played**

**Scope:** Chad game application.

**Level:** User “goal”.

**Primary Actor:** User.

**Stakeholders and Interests:**

* System records games played by the user. Game must record who the user was, the start date and time of the game, end date of the game, and the end result/

**Preconditions:** user is identified and authenticated.

**Success Guarantee:**

* The user can look at his/her history of past games played.

**Main Success scenario:**

* User checks his his/her history of past games played and can see when the game started, when it finished, and who won/lost/tied.

**Extensions:** none.

**Special Requirements:**

* A menu that allows you to access “History”. Text must be visible. Recover data if system crashes.

**Technology and Data Variations List:** none.

**Frequency of Occurrence:** nearly continuous.

**Miscellaneous:** none.

**Use Case UC10: User Profile**

**Scope:** Chad game application.

**Level:** User goal.

**Primary Actor:** User.

**Stakeholders and Interests:**

* User has their own profile which has a nickname picked out by the user; must be unique.
* User profile must allow user to see their history of games.

**Preconditions:** user is registered and authenticated.

**Success Guarantee:**

* User can see their nickname displayed in game and in their profile.
* The user can look at his/her history of past games played.

**Main Success scenario:**

* User has the ability to display their nickname during game play and can visibly see it under their profile and check their history of past games.

**Extensions:** none.

**Special Requirements:**

* Nickname must be unique and no two users can have the same nickname. Text must be visible. Recover data if system crashes.

**Technology and Data Variations List:** none.

**Frequency of Occurrence:** Continuous.

**Miscellaneous:** none.

**Use Case UC11: players required (game play)**

**Scope:** Chad game application.

**Level:** User goal.

**Primary Actor:** Users.

**Stakeholders and Interests:**

* Game must have two users to start a game.

**Preconditions:** two users must be present to start a game, otherwise error.

**Success Guarantee:**

* One user invites another and opponent accepts.

**Main Success scenario:**

* User invites another user being his opponents to start game, once opponent accepts the game will start.

**Extensions:** none.

**Special Requirements:**

* Opponent must accept invitation to play otherwise the game will not start.

**Technology and Data Variations List:** none.

**Frequency of Occurrence:** Whenever someone invites another user to play a game.

**Miscellaneous:** none.

**Use Case UC12: Game start**

**Scope:** Chad game application.

**Level:** User goal.

**Primary Actor:** Users.

**Stakeholders and Interests:**

* Once an opponent accepts invitation to start a game another player cannot join pre-existing game.

**Preconditions:** Two users are in a game against each other.

**Success Guarantee:**

* User trying to join game getting a message saying he/she can’t join.

**Main Success scenario:**

* User trying to join another person’s game that is already started will get a message saying he/she can’t join and ask if they would like to invite user to start a game with them

**Extensions:** none.

**Special Requirements:**

* An error message on user’s screen that is attempting to join a game in motion.

**Technology and Data Variations List:** none.

**Frequency of Occurrence:** Only when user is trying to join a game that already exists and is being played by two users.

**Miscellaneous:** none.

**Use Case UC13: Determine which Player Starts First**

**Scope:** Chad game application

**Level:** User-goal.

**Primary Actor:** User

**Stakeholders and Interests:**

* System must determine who starts the game.
* User with White chess pieces must move first.

**Preconditions:** User authenticated and other user has accepted invite.

**Success Guarantee:**

* Game has started.

**Main Success Scenario:**

* Game is started, user with white chess pieces make the first move.

**Extensions:** N/A

**Special Requirements:** N/A

**Technology and Data Variations List:** N/A

**Frequency of Occurrence:** Once per game.

**Miscellaneous:** N/A

**Use Case UC14: Determine Player Turn:**

**Scope:** Chad game application

**Level:** User-goal

**Primary Actor:** User

**Stakeholders and Interests:**

* User will take turns moving one chess piece at a time.
* Turn ends once move has been made.

**Preconditions:** Both users must have started/accepted an invitation to game

**Success Guarantee:**

* The other appropriate user can now make their move.

**Main Success Scenario:**

* User makes move, system will allow other user to make move once current user has finished.

**Extensions:** N/A

**Special Requirements:** N/A

**Technology and Data Variations List:** N/A

**Frequency of Occurrence:** Continuous. After a user has finished their turn.

**Miscellaneous:** N/A

**Use Case UC15: Make Moves in Active Games**

**Scope:** Chat game application

**Level:** User-goal

**Primary Actor:** User

**Stakeholders and Interests:**

* User wants to be able to access only their active games.

**Preconditions:** User must have been invited to or started games.

**Success Guarantee:**

* User cannot access games that are not active

**Main Success Scenario:**

* User logs in, sees list of active games, given option to play active games.

**Extensions:** N/A

**Special Requirements:** N/A

**Technology and Data Variations List:** N/A

**Frequency of Occurrence:** Continuous

**Miscellaneous:** N/A

**Use Case UC16: Makes Move when Player’s Turn**

**Scope:** Chat game application

**Level:** User “goal”.

**Primary Actor:** User

**Stakeholders and Interests:**

* Users want to be able to make a move if and only if it’s their turn.

**Preconditions:** Game must have already been initiated.

**Success Guarantee:**

* Only the current user can make a move when it’s their turn.

**Main Success Scenario:**

* User starts game, white makes first move, opposing user’s turn to make move

**Extensions:** N/A

**Special Requirements:** N/A

**Technology and Data Variations List:** N/A

**Frequency of Occurrence:** Often

**Miscellaneous:** N/A

**Use Case UC17: Allow Valid Moves Only**

**Scope:** Game-play Mechanics

**Level:** Game-play Restriction

**Primary Actor:** User

**Stakeholders and Interests:**

* Game Engine: wants to ensure the game is being played according to the rules
* Opponent: wants to know that only valid moves are used against them.

**Preconditions:** a game has started

**Success Guarantee:**

* Invalid moves if tried are rejected and the games state does not change

**Main Success scenario:**

* User attempts to preform an invalid move
* System identifies the move as incorrect
* The games state is returned to whatever state it was in before the move was made
* The user is notified the move was invalid

**Extensions:** none.

**Special Requirements:**

* If the game rules are changed newly valid moves must not trigger the invalid move behavior

**Technology and Data Variations List:**

* Any move that is not allowed by the rules must be identified
* Any move that is allowed by the rules must not be treated as invalid

**Frequency of Occurrence:** occasional

**Miscellaneous:** N/A

**U****se Case UC18: Save the State of Active Games**

**Scope:** User Games

**Level:** Games In Progress

**Primary Actor:** User

**Stakeholders and Interests:**

* User: wants to resume an unfinished game after leaving the game for some amount of time

**Preconditions:** a game has been started but not finished and one of the players leaves the game

**Success Guarantee:**

* The user may resume playing the game as normal

**Main Success scenario:**

* The user leaves an unfinished game
* The user rejoins the game and may play there next move as if they had not left

**Extensions:**

* The opponent may make their next move if the user had finished their turn before leaving the game
* The system will update the game's state and when the user returns the game will show the opponent's move

**Special Requirements:** N/A

**Technology and Data Variations List:** N/A

**Frequency of Occurrence:** Occasional

**Miscellaneous:** N/A

**U****se Case UC19: Determine End of Game and Game Result**

**Scope:** Game-play Mechanics

**Level:** End of Game

**Primary Actor:** User

**Stakeholders and Interests:**

* User: wants to be notified when the game is finished and what was the outcome.
* User Statistics Recorder: wants to know the outcome of the game so it may update the number and outcomes of games the user has played

**Preconditions:** The game must be over according to the rules

**Success Guarantee:**

* The user is notified of the outcome of the game

**Main Success scenario:**

* A user makes a move that ends the game
* Both players are notified if they won, lost, tied or if the game ended in a draw according to the rules

**Extensions:**

* A player may forfeit the game without a game ending move occurring
* The game will end
* The player that forfeited will be notified they lost and the opponent will be notified they won

**Special Requirements:** N/A

**Technology and Data Variations List:** N/A

**Frequency of Occurrence:** Often

**Miscellaneous:** N/A