*Use Case ID*: **SM801**

*Scenario*: **Start a new Group Game**

*Actor:* User

*Pre-Conditions:*

1. User is logged in

2. User is part of a group with more than 1 member

*Description:*

1. User opens the app
2. User selects the “groups” option on main menu
3. System switches to Groups screen
4. User selects the group they wish to play with from drop down menu.
5. System fetches the group player list
6. User selects the players for the game and confirms
7. System logs a new game instance
8. System loads game scoring screen

*Post Conditions:*

1. A new game entry has been stored for the group.
2. Group statistic for games played increases by 1

*Alternate Courses of Action:*

1. In step D.5, if exactly 2 players are not selected, the system shall prompt the user to enter the correct number of players.

*Exceptions:*

1. System fails to retrieve group player list

*Related Use Cases:*

Start a new Quick Game, Enter Score for Round, Edit Group

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency:* About 3 uses per week

*Criticality:* High. Essential to repeated group play

*Risk:* Medium. Misuse is logged, group access not restricted

*Constraints:*

Must log game results in group history within 5 seconds of match completion

Must update statistics of involved players within 3 seconds of match completion

--------------------------------------------------------------------------------------------------------------------------------

**Modification History:**

*Owner:* Luke Jeries

*Initiation date:* 10/12/2017

*Date last modified:* 10/12/2017

*Use Case ID:* **SM802**

*Scenario*: **Add a member to a Group**

*Actor:* User

*Pre-Conditions:*

1. User is logged in
2. User is part of a group

*Description:*

1. User opens the app
2. User selects the “groups” option on main menu
3. System switches to Groups screen
4. User selects the group they wish to add a player to
5. System fetches the group player list
6. System switches to group game staging screen
7. User selects the add new player option
8. System prompts User for new player’s ID
9. User enters new player’s ID and desired nickname
10. System adds the new player to the player list

*Post Conditions:*

1. Added User’s group list includes new group
2. Group player list includes new member

*Alternate Courses of Action:*

1. In step D.9, the user may leave the player ID blank to add a member with no account

*Exceptions:*

1. Invalid player ID is entered, no new player is added to group

*Related Use Cases:*

Edit Group

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency*: Low About 3 uses per week

*Criticality:* High. Essential to repeated group play

*Risk:* Medium. Vulnerable to invite spam

*Constraints:*

New member’s group list must include new group within 45 seconds

Must update member list of all players in group within 30 seconds

--------------------------------------------------------------------------------------------------------------------------------

**Modification History:**

*Owner:* Luke Jeries

*Initiation date*: 10/12/2017

*Date last modified:* 10/12/2017

*Use Case ID:* **SM803**  
*Scenario:* **View Group Score History**  
 *Actors:* User  
 *Pre-conditions:*

1. User is part of at-least 1 group

2. User is logged in  
 *Description:*

1. User opens the app
2. User Clicks Stats Button on Main Menu
3. System brings User to Statistics Screen
4. User Clicks on Groups drop-down box
5. System provides a list of groups the user is a part of
6. User selects a group they wish to see the history for
7. User clicks the History button under the drop down box
8. System pulls score screens from previous games the selected group has played
9. System brings user to a new screen and lists the score screens by session number and date played
10. User clicks the session he wants to view the score screen for
11. System shows the score screen for that particular session

*Post-conditions:*

1. User Remains at score screen until he clicks home button or back button  
*Alternative courses of action:*  
1. 7.b User can click on the group drop down box again, then goes to step 4  
*Exceptions:*

1. If system cannot pull the score screens display an error message

2. If system takes to long to pull score screens display an error message and ask  
 user if they want to continue waiting or go back to main menu  
*Related use cases:*  
 Scoring a game

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**  
 *Frequency:* Medium, Could be used often, 8 or more uses per week

*Criticality: Medium-High, depending on the user this could be essential or not*

*important at all*  
*Risk:* Low-medium, only apparent risk would be too much time used to implement

this use case

*Constraints:*  
System must pull game score histories in under 8 seconds  
System provides list of groups in under 2 seconds

--------------------------------------------------------------------------------------------------------------------------------

**Modification History:**  
 *Owner:* Taylor Shephard  
 *Initiation date:* 10/13/2017

*Date last modified:* 10/13/2017

*Use Case ID:* **SM804**  
*Scenario:* **Rename Group**  
 *Actors:* User  
 *Pre-conditions:*

1. User is part of at-least one group

2. User is logged in

3. User must have group edit permissions

*Description:*  
 1. User Starts App  
 2. User clicks on Group button on Main Menu  
 3. System brings user to group screen  
 4. User clicks on Groups drop down box  
 5. System pulls and lists groups the user is currently apart of in drop down box  
 6. User clicks on group he wishes to change name for from drop down box  
 7. System pulls information for selected group and displays on screen  
 8. User clicks edit name button located to the right of the group name

(will look like a little pencil icon)  
 9. System Displays a text box and asks user to type in new name  
 10. User types in the new name for selected group  
 11. User clicks ok  
 12. System checks if written name is the same as another groups  
 13. Group name is not taken so system sets selected group name to name typed

in  
14. System displays that the name of the group has been changed  
*Post-conditions:*  
1. The selected group's name is set to the new one  
*Alternative courses of action:*  
13.b Group name is taken so system displays that the name is already taken and asks user to enter a name, then goes to step 12  
*Exceptions:*  
1. If system cannot pull group info in specified time display error message

2. If system cannot change name of group display error message  
*Related use cases:*  
Creating a Group, Editing a Group, Adding member to Group.

------------------------------------------------------------------------------------------------------------------------------**Decision Support:**  
 *Frequency:* Very low, no more then once a week  
 *Criticality:* Low, not very essential   
 *Risk:* Medium, could cause issues or be problematic if the group name is

changed abusively  
*Constraints:*  
System must pull group info in less than 2 seconds.

------------------------------------------------------------------------------------------------------------------------------**Modification History:**

*Owner:* Taylor Shephard  
 *Initiation date:* 10/13/2017  
 *Date last modified*: 10/13/2017

*Use Case ID:* **SM805**

*Scenario:* **View active game**.

*Actor:*SM user.

*Pre-conditions:*

1. User must be logged into the system

2. Another user must have previously created an active game

*Description:*

1. Use case begins when SM user opens the app
2. The system shall provide the SM user with the menu
3. The SM user shall choose the **Join Game** button
4. When the request is received, the system shall prompt the SM user to scan a game code (possibly QR code) via the camera.
5. The SM user shall provide the code.
6. The system shall validate the code with the DB.
7. Use case ends when the system can find game and present the SM user with the **Gameplay Screen** in view-only mode.

*Post-conditions:*

1.The number of join-game requests stored in the system has increased by one.

2. The request has been saved in the system.

*Alternative Courses of Action:*

1. In step 4, the user has the option to cancel the request.
2. In step 6, if the code is incorrect an error message will be displayed.

*Exceptions:*

1. If there is no internet connection, the code won’t be able to check with the database.

*Related Uses Case:*

Creating a Game.

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency:* On average 2 requests are made weekly by SM user.

*Criticality:* Low. This feature is not a show stopper, but we would like to implement this to make the app stand out from others.

*Risk:* Medium. Implementing this use case requires camera accessibility for QR code (if we do a number, then there is less risk).

*Constraints:*

Non-functional requirements – Users must be able to access the new game screen within 5 seconds. With proper internet connection, code must validate within 10 seconds.

--------------------------------------------------------------------------------------------------------------------------------

**Modification History:**

*Owner:* Cristina Bindschatel

*Initiation date:* 10/12/17

*Date last modified:* 10/12/17

*Use Case ID:* **SM806**

*Scenario:* **Scorekeeping a Game**.

*Actor:*SM user.

*Pre-conditions:*

1. User must be logged into the system

2. User must be the creator/admin of a game

3. The game has begun, or is in progress

*Description:*

1. Use case begins when SM user selects an element to update via toggle arrows or a checkbox (time out, 8 on break, 8 ball and run, 8 out of turn, scratch on 8, or win)
2. The interface updates the changes.
3. A request is sent to the database to save the new score.

*Post-conditions:*

1. The request has been saved in the system.

*Alternative Courses of Action:*

*Exceptions:*

1. If there is no internet connection, the score won’t be able to update with the database.

*Related Uses Case:*

Creating a Game.

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency:* On average 5 requests are made weekly by SM user.

*Criticality:* High. This is the main feature of the entire app.

*Risk:* Medium. Making sure all of the options work properly will be the biggest challenge.

*Constraints:*

Non-functional requirements – Users must see the interface update instantly.

When a winner is selected, the game must end and move onto the next one. A function we would also like to implement is that the game automatically ends when either of the players maximum wins are met.

--------------------------------------------------------------------------------------------------------------------------------

**Modification History:**

*Owner:* Cristina Bindschatel

*Initiation date:* 10/12/17

*Date last modified:* 10/12/17

Use Case ID: **SM807**

Scenario: **Log in**

*Actor:* User

*Pre-conditions:*

1. User has account registered.

*Description:*

1. Use case begins when user opens app.
2. User clicks on log in.
3. User types in credentials.
4. System validates credentials.
5. User is taken to home screen.

*Exceptions:*

1. User is not registered.
2. User types in wrong credentials.

*Post Conditions:*

1. The user is logged in to account.
2. User is given access to create group, view history, quick play.

*Related Use Cases:*

Validate account.

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency*: High. User’s will login every time they open the app.

*Criticality:* High. User’s need account to use the app.

*Risk:* High. Logging in will allow users to use app functionality.

*Constraints:*

Non-functional requirements.

--------------------------------------------------------------------------------------------------------------------------------

Use Case: **SM808**

Scenario: **Create group**

*Actor:* User

*Pre-conditions:*

1. User has logged into account.

*Description:*

1. Use case begins when user clicks Create Group button.
2. The UI shows user’s name with a blank box for a score.
3. The UI shows Add Player option.
4. The user clicks add player to group.
5. The UI shows blank space to type in Player name.
6. Once the name is entered, it appears in the group under the previous player name.
7. Once all names have been entered, the user selects start game option.

*Exceptions:*

1. User does not want to create a group.

*Post Conditions:*

1. The game has been started for the group of players that was created.

*Related Use Cases:*

Log in, Start Game

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency:* Medium. The number of users creating groups to keep track of score is

medium to high.

Criticality: High. The is one of the main functions of the program.

Risk: High. Implementing the use case allows the users to create groups for tracking

Scores.

*Constraints:*

Non-Functional Requirements.

--------------------------------------------------------------------------------------------------------------------------------

Use Case: **SM809**

*Scenario:* Check Personal Game History

*Actor:* User

*Pre-Conditions:*

1. Have account
2. Played games previously
3. User is logged into account

*Description:*

1. Use case begins when user clicks View Stats/History button.
2. The UI shows the user the View Statistics Screen.
3. The user clicks the Personal Game History button.
4. The UI shows the personal history of the user.

*Exemptions:*

1. User has no personal game history.

*Post Conditions:*

1. The UI shows the View Statistics Screen.

*Related Use Cases:*

Log in, View Personal Statistics, View Group Statistics, View Group Game

History

--------------------------------------------------------------------------------------------------------------------------------

**Decision Support:**

*Frequency:* Medium. The number of users checking personal history is medium to high.

*Criticality:* High. The is one of the main functions of the program.

*Risk:* High. Implementing the use case allows the users to track their personal history.

*Constraints:*

Non-Functional Requirements.

--------------------------------------------------------------------------------------------------------------------------------