

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

# FINAL ITERATION

## Use Cases

1: As a user, I want to log into the ~~website~~ application so that my game history is saved. My conditions of satisfaction are being able to see my wins and losses for the games I have played.

### **Use Case for User Story 1:**

- a. Title: Logging In and Persistent Information
- b. Actor: Any User (registered or non-registered)
- c. Basic Flow:
  - i. User starts our desktop application.
  - ii. User clicks the "Sign in with Google" button
  - iii. User's web browser prompts a Google signin and the user goes through the steps to authorize the application for OAuth.
  - iv. The web browser shows a confirmation message that signing in was successful or shows an error message if signing in was not successful.
  - v. If signing in was not successful nothing happens within the application. Otherwise, the application advances to the home screen where it either shows a welcome message if you are a new user and shows your game history and game invites if you are a returning user.

2: As a user, I want a simple UI so that I can play tic-tac-toe with my mom! My conditions of satisfaction are that there are no ads, we log in via email, and the app is hosted on a website.

### **Use Case for User Story 2:**

- a. Title: User plays tic-tac-toe with opponent
- b. Actor: A registered user
- c. Description: User attempts to play tic-tac-toe with her mom
- d. Basic Flow:
  - i. User is a 21-year old Computer Science major at Columbia University. She opens the Game Fame application and clicks on "Sign In with Google" button.
  - ii. Once she has given her authentication information, she proceeds to click on "Create New Game."
  - iii. She selects "Tic-Tac-Toe," puts her mom's email address in, and selects "Create Game."
  - iv. User notices that the UI of the board is very simple and does not contain any advertisements. She makes her move and waits for her mom to play!

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

3: As a user, I want a menu of available games so that I can see all the games that I can play with my friends. My conditions of satisfaction are a grid or list of playable games.

**Testing for User Story 3:**

- a. Title: User can see a list of available games
- b. Actor: Signed-in User
- c. Precondition: User is signed into the application
- d. Basic Flow:
  - i. User clicks the "Create Game" button at the top menu of the application and is brought to the game creation screen.
  - ii. User observes the "Game Type" select, which lists all of the available games. When a game is selected, the application screen is changed to provide a description of the game's rules.

~~4: As a user, I want descriptions of games so that I can understand games that I am not familiar with. My conditions of satisfaction are that every game must contain a description as well as instructions. As a user, I want descriptions of games under the game being played so I am aware of how to play while viewing the game board.~~

**Use Case for User Story 4:**

- a. Title: User views description of games
- b. Actor: Registered User
- c. Description: User wants to see how to play games being hosted on the Game Fame application
- d. Basic Flow:
  - i. User is an 8-year old student in elementary school, looking to play games with his friends. He heard of an application called Game Fame but is unsure of how to play the games that are hosted on there.
  - ii. Upon opening the application on his desktop, he clicks the "Sign In with Google" button and puts his authentication information in.
  - iii. He sees the "Create Game" button and selects it.
  - iv. User now sees a drop-down menu which lists all the games available on the platform.
  - v. Upon selecting game of choice, he sees the description of the game underneath the board of the game he chose!

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

**5: As a user, I want to be able to switch between games that I am playing so that I am not forced to play only one game at a time with just on friend. My conditions of satisfaction are I must have a list of active games so that I can easily switch between games.**

**Use Case for User Story 5:**

- a. Title: Multiple Game Instances
- b. Actor: Paulo, Tracy, Malcolm
- c. Description: Paulo switches between two ongoing game
- d. Basic Flow:
  - i. Paulo is a 17 year old high school junior. He likes to stay in school after classes have ended to play games with his best friends Tracy and Malcolm.
  - ii. Paulo and his friends go to the school's computer lab and all launch the Game Fame application.
  - iii. Tracy enjoys playing tic-tac-toe and Malcolm likes connect-four, therefore they each start their favorite games with Paulo respectively.
  - iv. Paulo receives both invites and begins playing with Tracy.
  - v. After a while, Malcolm makes a move in connect-four, and therefore Paulo clicks on the homescreen button to go back to a list of all of his ongoing games.
  - vi. He sees that Malcolm has made a move and and click on their ongoing game. This action opens up their game and they continue playing.

**6. As a user, I want to have an in-game text chat so that I can talk to my friends during games without having to use an external messaging service. My conditions of satisfaction are that messages display in real-time, have timestamps, and are persistent.**

**Use Case for User Story 6:**

- a. Title: In-game Chat
- b. Actor: Leslie
- c. Description: Leslie plays games with her friends and sends them messages.
- d. Basic Flow:
  - i. Leslie is 25-year old software engineer working at Clarifai. She likes to destress during a tough day by playing tic-tac-toe with her friends and talking to them at the same time.
  - ii. She opens up Game Fame and clicks "Sign In with Google."
  - iii. Upon submitting her authentication details, she is prompted to click "Create Game," where she selects tic-tac-toe from the drop down list and inputs her friend's email address.
  - iv. Once she creates the game, she sees a message board at the bottom of the tic-tac-toe board and begins playing and sending messages to her friend!

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

For the final iteration of the project, we fixed up the bugs that existed from the second iteration and added one new feature. Two important bug fixes were:

- (1) Adding move-locking logic for Connect Four - a user can only play if it is his/her turn;
- (2) Message rendition for Connect Four - we were unable to render messages properly in the Connect Four page but now everything is working properly.

The feature we added in our final iteration is:

- (1) Proper game invite system: We now allow a user to invite friends using their email addresses, sending them an invite as soon as they open up the application. If accepted, the users can play the game together!

## SECOND ITERATION

### Test Plan

Since we are managing application state with [Redux](#), the unit testing for our application is performed by testing that the actions dispatched by the action creators are as they should be. The testing for this application was done with [Jest](#) and the files that our test suite runs are located within the "test/actions" directory of our project. The equivalence partitions and boundary conditions for each group of action creators are as follows:

Authentication:

- Signing In:
  - Equivalence Partitions: Emails that are currently registered as users within our database; emails that are not currently registered as users within our database.
  - Boundary Conditions: None since emails are strings and this operation is simply a lookup.
  - NOTE: this set of tests is the only one that is particularly difficult to test since it is dependent upon the user interacting with a separate application (their web browser) to sign into our application. (Users sign in with Google.) We have made progress towards writing complete tests for this (by simulating a user), but since they do not fully work at the moment, we have not committed these changes.
- Signing Out:
  - Equivalence Partitions: Application states where the users are signed in (valid); Application states where the users are not signed in (also valid). Note that if the users are not signed in, performing the sign out action is equivalent to doing nothing and should succeed.
  - Boundary Conditions: None since the equivalence partitions span all possible application states.

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

### Connect Four

- Retrieve Board from Database
  - Equivalence Partitions: Keys that have a game associated with them (valid); Keys that do not have games associated with them (invalid).
  - Boundary Conditions: None since game keys are strings and this operation is simply a lookup.
- Making a Move
  - Equivalence Partitions: Board states where a move for the chosen column is permitted (valid); Board states where a move for the chosen column is not permitted (invalid).
  - Boundary Conditions: Board states that are empty; Board states where the column clicked is one move away from being full; Board states where the column clicked is full.
- Creating a Board
  - Equivalence Partitions: The set of all application states (should make a new board no matter what).
  - Boundary Conditions: None since this should always succeed.

### Game Info

- Retrieving Game Info
  - Equivalence Partitions: Emails that are currently registered as users within our database; emails that are not currently registered as users within our database.
  - Boundary Conditions: None since emails are strings and this operation is simply a lookup.

### Tic Tac Toe

- Retrieve Board from Database
  - Equivalence Partitions: Keys that have a game associated with them (valid); Keys that do not have games associated with them (invalid).
  - Boundary Conditions: None since game keys are strings and this operation is simply a lookup.
- Creating a Board
  - Equivalence Partitions: The set of all game keys (valid).
  - Boundary Conditions: None since this should always succeed.
- Resetting a Board
  - Equivalence Partitions: The set of all game keys (valid).
  - Boundary Conditions: None since this should always succeed.

## Branch Coverage

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

While researching coverage tools, we learned that a coverage tool called [Istanbul](#) is built into Jest, so we activated it on our test suite. The results are listed below:

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	68.75	49.15	62.16	69.23	
app/actions	68.15	49.09	61.11	68.66	
ConnectFour.js	84	53.06	70.59	85.14	... 93,212,216,217
auth.js	19.23	0	25	19.23	... 48,52,53,55,59
games.js	75	100	50	75	14,15
tictactoe.js	69.23	50	85.71	69.23	... 17,18,22,26,27
app/constants	100	100	100	100	
firebase.js	100	100	100	100	
internals/scripts	71.43	50	100	71.43	
CheckBuiltExist.js	71.43	50	100	71.43	19,27

From these results, we can observe that the unit tests ("app/actions" section) have some work to do in terms of branch coverage, but this is most likely due to the technical difficulty in testing asynchronous Firebase code (e.g. making sure that updates are detected within the database, which raises timing issues) – this is something that we will be working on within the next iteration. We did not write any test cases for loops because our tested actions do not rely on loops ([this is due to an asynchronous design pattern recommended by Firebase](#)). Additionally, it is worth noting that the abysmally low coverage ratPaulo switches between two ongoing gamees for "auth.js" are due to the issue mentioned above about the action being dependent on the results from an external application (due to OAuth).

## Functionality Changes

We will be adding three key features to our product for the second iteration demo.

- (1) Messages within games: We have a chat between the players of the game which stores all logs (even conversations from prior game) and brings them into the latest one. We accomplish this by adding a new category to our Firebase backend which stores user1&user2, where user1 and user2 are the respective players, under the "conversations" child node. For the next iteration, we will be adding functionality for users to be able to delete their messages from the chat (which will delete them from the other user's end as well).
- (2) Locking players after move: In order to have a fair game, we need to ensure that the user who just made a move cannot play for the other user. We accomplish this by having an extra parameter in our backend game logic which keeps track of the userID of the last player who made a move. Once that is stored, we check the backend before letting the next player make a move, ensuring that only the alternate player is allowed to play. For

Team Name: GameFame

Team Members: Sarthak Arora (sa3522), Gharvhel Carre (gc2767), Eleanor Lee (sl3948), Aleksander Zieba (arz2116)

the next iteration, we plan on providing functionality for an “undo move” if both players agree to it.

- (3) Improving appearance of boards: The tic-tac-toe board was too small and the connect-four board colors were a bit too bright. Changing the appearance to make the games look better is very important if we want people to keep using our product!