

# Team 3 Test Plan

## MSCI 342 Homework 3 Part 2

By: Danial Betres, Victoria Li, Jason Wang, Ishani Adityan, Priyanka Brodie, Ayat Turar

## Testing Sections

Player Login

Cards

Transaction

Admin Login

Admin Navigation

Admin Export Page

Admin Card Set Page

Admin Cards

## Functional Requirements

### Player Login

**Requirement:** FUN-PLAYER-LOGIN-1

**Test case name:** Bad Login

**Author:** Danial Betres

**Description:** Attempting to login when the username input field contains non-alphanumeric characters should not work

**Preconditions:** Open a new tab

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/>
2. Enter any non-alphanumeric characters like “!@#\$\$%^&\*()”
3. Click on join

**Expected output:** Nothing should happen as the user won't be able to click on the “join” button since it's disabled.

**Requirement Name:** FUN-PLAYER-LOGIN-2

**Test Case Name :** Bad Login

**Author:** Danial Betres

**Description:** The “login” button shall be disabled when the username input field is empty.

**Preconditions:** Open a new tab

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/>
2. Leave the username input field blank
3. Click on join

**Expected output:** Nothing should happen as the user won't be able to click on the "join" button since it should be disabled.

**Requirement Name:** FUN-PLAYER-LOGIN-3

**Test Case Name:** Good Login

**Author:** Danial Betres

**Description:** The "login" button shall be enabled when the user has inputted a username that contains only alphanumeric characters.

**Preconditions:** Open a new tab

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/>
2. Enter a username that contains only alphanumeric characters
3. Proceed to click on "join".

**Expected output:** the application should allow the player to click on the join button since the inputted username follows all constraints

**Requirement Name:** FUN-PLAYER-LOGIN-4

**Test Case Name:** Login with less than three players currently in the system

**Author:** Danial Betres

**Description:** The application shall redirect the user to a new page that displays "loading" upon clicking the "login" button.

**Preconditions:** Ensure there are no more than two players currently waiting to play the game

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/>
2. Enter a valid username (only alphanumeric characters)
3. Click on join

**Expected output:** The application should direct the user to the loading page since there is still less than four players in the game

**Requirement Name:** FUN-PLAYER-LOGIN-5

**Test Case Name:** Open instructions modal

**Author:** Danial Betres

**Description:** The application shall display a modal with instructional text about the game when the "instructions" button is clicked.

**Preconditions:** Open a new tab

**Executions steps:**

1. Go to <https://team-3.herokuapp.com/>
2. Click on the instructions modal which is found on the login page

**Expected output:** A modal should pop up with a list of all the instructions that are required to play the game

**Requirement Name:** FUN-PLAYER-LOGIN-6

**Test case name:** Close instructions modal

**Author:** Danial Betres

**Description:** The user shall be able to close the instruction modal by clicking on either the top right x button or the “close” button.

**Preconditions:** Open a new tab, go to <https://team-3.herokuapp.com/> and have the instructional modal open

**Execution steps:** Click on the top right x button or the “close” button

**Expected output:** The modal should close and the user should see the login page again

**Requirement Name:** FUN-PLAYER-LOGIN-7

**Test Case Name:** Login with already used username

**Author:** Danial Betres

**Description:** An error shall be displayed on the main screen if the user tries to log in with an already used username.

**Preconditions:** Have another player already log in to the game with a valid username. Open a new tab as well

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/>
2. Log in with the same username the previous player logged in with
3. Click on “join”.

**Expected output:** An error should show up informing the player that the username is already taken

## Cards

**Requirement:** FUN-CARDS-1

**Test case name:** Display of Cards on Main Game Page

**Author:** Priyanka Brodie

**Description:** Upon entering the game room, each user should see four cards on their page. This is the initial state of the game.

**Preconditions:** User should be a player and should be on the player login page of the application.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/> and login with a username to enter the game room.
2. Wait for the three other players to login to the application.
3. As another option, you can open four tabs on your local machine, go to the following link <https://team-3.herokuapp.com/> on each tab, and enter a username for each user.

**Expected output:** Once all players have entered the game room, the application should display four cards on each user’s page.

**Requirement:** FUN-CARDS-2

**Test case name:** Number of Images Corresponding to Cards on Main Game Page

**Author:** Priyanka Brodie

**Description:** The user should always have an image associated with a card that is in their possession (i.e. if the user has four cards, the application should display four images).

**Preconditions:** User should be a player and should be on the player login page of the application.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/> and login with the correct credentials to enter the game room.
2. Wait for all other players to login to the application.
3. As another option, you can open four tabs on your local machine, go to the following link <https://team-3.herokuapp.com/> on each tab, and enter a username for each user.
4. Initially, you should see four images corresponding to four cards on your page.
5. Give a card to another player and ask them to accept the card.

**Expected output:** Once the player has accepted the card, the application should display three images corresponding to the three cards that are in your possession on your page.

## Transactions

**Requirement:** FUN-TRANSACTION-1

**Test Case Name:** Accept Proposed Transactions

**Author:** Victoria Li

**Pre-conditions:** The “player” database table should be allow the current players to be able to join the game.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set
3. Get another player to drag a card to your user’s droppable bin.
4. A card should display in the droppable area. Click the “accept” button.

**Expected Output:** The card that was dragged to the user’s droppable bin should appear in the current droppable areas with an accept and reject button. Clicking on the accept button should send a POST request to accept the trade.

**Requirement:** FUN-TRANSACTION-1

**Test Case Name:** Reject Proposed Transactions

**Author:** Victoria Li

**Pre-conditions:** The “player” database table should be allow the current players to be able to join the game.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set
3. Get another player to drag a card to your user’s droppable bin.
4. A card should display in the droppable area. Click the “reject” button.

**Expected Output:** The card that was dragged to the user’s droppable bin should appear in the current droppable areas with an accept and reject button. Clicking on the reject button should send a POST request to accept the trade.

**Requirement:** FUN-TRANSACTION-2

**Test Case Name:** Proposed card moves from Sender's Cardset into Recipient's Cardset

**Author:** Victoria Li

**Pre-conditions:** The "player" database table should be allow the current players to be able to join the game.

**Execution Steps:**

1. 1.Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set
3. Get another player to drag a card to your user's droppable bin.
4. A card should display in the droppable area. Click the "accept" button.

**Expected Output:** The current user (recipient) should see the proposed card added to their cardset. The other user (sender) should see the proposed card removed from their cardset.

**Requirement:** FUN-TRANSACTION-3

**Test Case Name:** Proposed card moves from Sender's Cardset into Recipient's Cardset

**Author:** Victoria Li

**Pre-conditions:** The "player" database table should be allow the current players to be able to join the game.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set
3. Get another player to drag a card to your user's droppable bin.
4. A card should display in the droppable area. Click the "accept" button.

**Expected Output:** The current user (recipient) should see the proposed card added to their cardset. The other user (sender) should see the proposed card removed from their cardset.

**Requirement:** FUN-TRANSACTION-4

**Test Case Name:** POST request 401 error when "request\_trade" is called

**Author:** Victoria Li

**Pre-conditions:** The "player" database table should be allow the current players to be able to join the game. Set up some API development environment (i.e. Postman) for API endpoint testing.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set.
3. Use POSTMAN and send a POST request to /cards/request-trade with card\_id, recipient\_id, sender\_id fields as NULL.
4. Press submit

**Expected Output:** The response that comes back on Postman should be 401 unauthorized error.

**Requirement:** FUN-TRANSACTION-5

**Test Case Name:** POST request 200 status when “request\_trade” is called

**Author:** Victoria Li

**Pre-conditions:** The “player” database table should be allow the current players to be able to join the game. Set up some API development environment (i.e. Postman) for API endpoint testing.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set.
3. Go to the database and query for any two rows in player\_card table.
4. Get the user id and card for a row and use its as sender\_id and card\_id
5. Get the other user id from the second row.
6. Use POSTMAN and send a POST request to /cards/request-trade with card\_id, recipient\_id, sender\_id fields as the variables found from the previous step.
7. Press submit

**Expected Output:** The response that comes back on Postman should be 200 and a new row should be added to the transaction table with type as “inProgress”.

**Requirement:** FUN-TRANSACTION-6

**Test Case Name:** Most Objectively Imbalanced Set of Cards Distributed to a User

**Author:** Victoria Li

**Pre-conditions:** The “player” database table should be allow the current players to be able to join the game.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set.
3. Go to database and query the player\_card table i.e. “SELECT \* FROM PLAYER\_CARD”

**Expected Output:** Verify that each player has 4 cards each with a unique main value.

**Requirement:** FUN-TRANSACTION-7

**Test Case Name:** Players Disabled during an “inProgress” Transaction

**Author:** Victoria Li

**Pre-conditions:** The “player” database table should be allow the current players to be able to join the game.

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. Wait until 3 other players join the game, the application shall be redirected to the main page with 4 cards distributed to your card set.
3. Drag a card to one of the droppable bins, a trade request for a transaction has been requested

**Expected Output:** A “IS\_PENDING\_TRANSACTION” message has been emitted to all players and disables all users from dragging and dropping cards. The only user that can perform any call to action is the recipient user which can accept or reject a trade.

## Admin Login

**Requirement:** FUN-ADMIN-LOGIN-1

**Test Case Name:** Admin-Login-Good

**Author:** Jason Wang

**Description:** A login attempt with the proper login credentials will pass

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL and enter the string “RobD” for the username and “cardadmin342” as the password. Then clicks the login button.

**Expected output:** The user is directed to the admin homepage.

**Requirement:** FUN-ADMIN-LOGIN-1

**Test Case Name:** Admin-Login-Bad

**Author:** Jason Wang

**Description:** A login attempt without the proper login credentials will fail

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL and enter the string “test” for both the username and password. Then clicks the login button.

**Expected output:** An error message should appear, but the user remains in the admin login page.

**Requirement:** FUN-ADMIN-LOGIN-2

**Test Case Name:** Admin-Login-Fields-Good

**Author:** Jason Wang

**Description:** When the login page loads successfully, there should be 2 textboxes for the entry of username and password.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL

**Expected output:** A page should appear with two textboxes labelled for username entry or password entry.

**Requirement:** FUN-ADMIN-LOGIN-2

**Test Case Name:** Admin-Login-Fields-Bad

**Author:** Jason Wang

**Description:** If the login page does not load successfully, the page will be missing the 2 textboxes for the entry of username and password.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL

**Expected output:** A page will not appear with two textboxes labelled for username entry or password entry.

**Requirement:** FUN-ADMIN-LOGIN-3

**Test Case Name:** Admin-Login-Redirect-Good

**Author:** Jason Wang

**Description:** A successful login attempt will redirect the user to the admin homepage

**Preconditions:** The user has access to the admin URL and the login credentials

**Execution steps:** The user inputs the admin URL and enter the string “RobD” for the username and “cardadmin342” as the password. Then clicks the login button.

**Expected output:** The user is directed to the admin homepage.

**Requirement:** FUN-ADMIN-LOGIN-3

**Test Case Name:** Admin-Login-Redirect-Bad

**Author:** Jason Wang

**Description:** A failed login attempt will not redirect the user

**Preconditions:** The user has access to the admin URL and the login credentials

**Execution steps:** The user inputs the admin URL and enter the string “RobD” for the username and “cardadmin342” as the password. Then clicks the login button.

**Expected output:** The user remains in the admin login page.

**Requirement:** FUN-ADMIN-LOGIN-4

**Test Case Name:** Admin-Login-Empty-Good

**Author:** Jason Wang

**Description:** If the user enters a username and password, the login button will be enabled.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL and “test” as both the username and password.

**Expected output:** The login button will be enabled (not greyed out & clicking on it will result in an error message)

**Requirement:** FUN-ADMIN-LOGIN-4

**Test Case Name:** Admin-Login-Empty-Bad

**Author:** Jason Wang

**Description:** If the user leaves either the username or password boxes empty, the login button will be disabled.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL

**Expected output:** The login button will be disabled (greyed out & clicking it will do nothing)

**Requirement:** FUN-ADMIN-LOGIN-5

**Test Case Name:** Admin-Login-Filled-Good

**Author:** Jason Wang

**Description:** If the user enters a username and password, the login button will be enabled.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL and “test” as both the username and password.



**Expected output:** The login button will be enabled (not greyed out & clicking on it will result in an error message)

**Requirement:** FUN-ADMIN-LOGIN-5

**Test Case Name:** Admin-Login-Filled-Bad

**Author:** Jason Wang

**Description:** If the user leaves either the username or password boxes empty, the login button will be disabled.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the admin URL

**Expected output:** The login button will be disabled (greyed out & clicking it will do nothing)

**Requirement:** FUN-ADMIN-LOGIN-6

**Test Case Name:** Admin-Refresh-Good

**Author:** Jason Wang

**Description:** If properly implemented, if the user refreshes the admin homepage or returns to the homepage after logging in previously in the same session, the user will remain on/be redirected to the admin homepage.

**Preconditions:** The user has access to the admin URL and login credentials.

**Execution steps:** The user inputs the admin URL, login credentials, then clicks the login button. Once on the admin homepage click the browser's "refresh" button.

**Expected output:** The admin homepage should refresh.

**Requirement:** FUN-ADMIN-LOGIN-6

**Test Case Name:** Admin-Refresh-Bad

**Author:** Jason Wang

**Description:** If improperly implemented, if the user refreshes the admin homepage or returns to the homepage after logging in previously in the same session, the user will be redirected to the admin login page.

**Preconditions:** The user has access to the admin URL and login credentials.

**Execution steps:** The user inputs the admin URL, login credentials, then clicks the login button. Once on the admin homepage click the browser's "refresh" button.

**Expected output:** The user will be redirected to the admin login page.

## **Admin Navigation**

**Requirement:** FUN-ADMIN-NAVIGATION-1

**Test case name:** Admin Navigation Visible on Add Cardset Page

**Author:** Priyanka Brodie

**Description:** Upon clicking on the hamburger menu button on the admin Add Cardset page, a navigation pane should appear on the left-hand side of the page.

**Preconditions:** User should be an admin and should be on the Add Cardset page in the Admin Portal.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/admin/cardset>.
2. Click on the hamburger menu button at the top left-hand side of the page.

**Expected output:** A navigation pane should appear on the left-hand side of the page containing one tab called “Export”.

**Requirement:** FUN-ADMIN-NAVIGATION-1

**Test case name:** Admin Navigation Visible on Export Page

**Author:** Priyanka Brodie

**Description:** Upon clicking on the hamburger menu button on the admin Export page, a navigation pane should appear on the left-hand side of the page.

**Preconditions:** User should be an admin and should be on the Export page in the Admin Portal.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/admin/exportcsv>.
2. Click on the hamburger menu button at the top left-hand side of the page.

**Expected output:** A navigation pane should appear on the left-hand side of the page containing one tab called “Cardsets”.

**Requirement:** FUN-ADMIN-NAVIGATION-2

**Test case name:** Admin Navigation Pane Directs Users to the Export Page

**Author:** Priyanka Brodie

**Description:** Upon clicking on the “Export” tab in the navigation pane found on the Add Cardset page, the application should redirect users to the Export page.

**Preconditions:** User should be an admin and should be on the Add Cardset page in the Admin Portal.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/admin/cardset>.
2. Click on the hamburger menu button at the top left-hand side of the page.
3. Once the navigation pane appears, click on the “Export” tab.

**Expected output:** The application should redirect the user to the Export page in the application.

**Requirement:** FUN-ADMIN-NAVIGATION-3

**Test case name:** Admin Navigation Pane Directs Users to Add Cardset page

**Author:** Priyanka Brodie

**Description:** Upon clicking on the “Cardsets” tab in the navigation pane found on the Export page, the application should redirect users to the Add Cardset page.

**Preconditions:** User should be an admin and should be on the Export page in the Admin Portal.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/admin/exportcsv>.
2. Click on the hamburger menu button at the top left-hand side of the page.
3. Once the navigation pane appears, click on the “Cardsets” tab.

**Expected output:** The application should redirect the user to the Add Cardset page in the application.

## Admin Export Page

**Requirement:** FUN-ADMIN-EXPORT-1

**Test case name:** Export Data Button Functionality

**Author:** Priyanka Brodie

**Description:** Upon clicking on the “Export Experiment Data” button on the Export page, the application should download a CSV file onto the user’s local machine.

**Preconditions:** User should be an admin and should be on the Export page in the Admin Portal.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/admin/exportcsv>.
2. Click on the “Export Experiment Data” button.

**Expected output:** A CSV file should start downloading onto the user’s local machine.

**Requirement:** FUN-ADMIN-EXPORT-2

**Test case name:** Delete Data Button Functionality

**Author:** Priyanka Brodie

**Description:** Upon clicking on the “Delete Experiment Data” button on the Export page, the application should clear the experiment data from the server/database.

**Preconditions:** User should be an admin and should be on the Export page in the Admin Portal.

**Execution steps:**

1. Go to <https://team-3.herokuapp.com/admin/exportcsv>.
2. Click on the “Delete Experiment Data” button.

**Expected output:** All of the data in the Experiment table in the database should be cleared. This includes the Recipient\_ID, Sender\_ID, Card\_ID, Round\_ID and Type (In Progress, Accepted, Rejected) columns.

**Requirement:** FUN-ADMIN-EXPORT-3

**Test case name:** Data test - good

**Author:** Ishani Adityan

**Description:** The CSV file containing data from the last experiment shall contain the following columns: Recipient\_ID, Sender\_ID, Card\_ID, Round\_ID, and Type (In-Progress, Accepted, Rejected), and shall not be a null file when exported.

**Preconditions:** A previous experiment must have been completed so that the file won’t be null.

**Execution steps:**

1. On <http://localhost:3000/admin/exportpage>
2. Click the button
3. A file will be downloaded onto your computer

**Expected output:** The file downloaded should have values for the columns: Recipient\_ID, Sender\_ID, Card\_ID, Round\_ID, and Type (In-Progress, Accepted, Rejected)

## Admin Card Set Page

**Requirement:** FUN-ADMIN-CARDSET-1

**Test Case Name:** Save cards to database

**Author:** Ayat Turar

**Description:** The application shall save the input values and pictures to the corresponding 'Cardset' table in the database, once the user clicks the "Save cards" button.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards.

**Execution steps:** Go to <https://team-3.herokuapp.com/admin/cards> , click on any card set and after inputting all the cards click on "Save cards"

**Expected output:** The cards will be saved to the database and would show up upon the refresh of the page.

**Requirement:** FUN-ADMIN-CARDSET-2

**Test Case Name:** Clicking on the card set name to view the cards

**Author:** Ayat Turar

**Description:** Click on 'Cardset 1' Cardset Page to view the set of 16 cards.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards.

**Execution steps:** Go to <http://localhost:3000/admin/cards> , click on the card set tab to view the cards

**Expected output:** 16 cards will be shown corresponding to the clicked card set tab.

**Requirement:** FUN-ADMIN-CARDSET-3

**Test Case Name:** Clicking on the card set name to view the cards

**Author:** Ayat Turar

**Description:** Click on 'Cardset 2' Cardset Page to view the set of 16 cards.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards.

**Execution steps:** Go to <http://localhost:3000/admin/cards> , click on the card set tab to view the cards

**Expected output:** 16 cards will be shown corresponding to the clicked card set tab

**Requirement:** FUN-ADMIN-CARDSET-4

**Test Case Name:** Clicking on the card set name to view the cards

**Author:** Ayat Turar

**Description:** Click on 'Cardset 3' Cardset Page to view the set of 16 cards.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards.

**Execution steps:** Go to <http://localhost:3000/admin/cards> , click on the card set tab to view the cards

**Expected output:** 16 cards will be shown corresponding to the clicked card set tab.

**Requirement:** FUN-ADMIN-CARDSET-5

**Test Case Name:** Validation on saving the cards

**Author:** Ayat Turar

**Description:** Prevent the user from saving the cards if no all of them contain all the required features.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards and all the required fields for 16 cards to save the cards

**Execution steps:** Go to <http://localhost:3000/admin/cards> , click on the card set tab to view the cards, fill all the required fields and click on “Save Cards”.

**Expected output:** The user will not be able to save the cards unless the required fields are present

**Requirement:** FUN-ADMIN-CARDSET-6

**Test Case Name:** Choosing a specific card set for a main experiment

**Author:** Ayat Turar

**Description:** The user will be able to choose a specific card set for a main experiment when clicking on “Choose for an experiment”.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards, fill all 16 cards for cardset and click on “Choose for experiment”.

**Execution steps:** Go to <http://localhost:3000/admin/cards> , click on the card set tab to view the cards, fill all the required fields and click on “Choose for an experiment”.

**Expected output:** The user will not be able to save the cards unless the required fields are present

**Requirement:** FUN-ADMIN-CARDSET-7

**Test Case Name:** Get the active card set being used

**Author:** Ayat Turar

**Description:** Upon clicking on “Choose for Experiment”, the users will be notified with a status of which card set is being used.

**Preconditions:** Admin needs to provide the credentials to view and modify the cards, fill all 16 cards for cardset and click on “Choose for experiment” to be notified of the active card set name being used

**Execution steps:** Go to <http://localhost:3000/admin/cards> , click on the card set tab to view the cards, fill all the required fields and click on “Choose for an experiment”. The text field should be shown on the top right indicating which card set is being used.

**Expected output:** The user will be notified of an active card set being used.

## Admin Cards

**Requirement:** FUN-ADMIN-CARDS-1

**Test Case Name:** Admin Display text fields

**Author:** Ishani Adityan

**Description:** Admin user can input text into the indicated ‘primary’ and ‘secondary’ text boxes shown in the page.

**Preconditions:** Admin user has logged in correctly to view this page.

**Execution steps:**

1. Go to <http://localhost:3000/admin/cardset> , this is the ‘Cardset page’.
2. Click on the text boxes for one ‘card’ and enter values.

**Expected output:** Text box will allow user to type in values to both the text fields.

**Requirement:** FUN-ADMIN-CARDS-2

**Test Case Name:** Admin card image field

**Author:** Ishani Adityan

**Description:** Admin user upon clicking on the image box is taken to pop-up to select image from computer to input into image field.

**Preconditions:** Admin user has a .jpg/.png image stored in their computer to import to the page.

**Execution steps:**

1. Go to <http://localhost:3000/admin/cardset> , this is the ‘Cardset page’,
2. Click on image box.

**Expected output:** A pop-up prompt will show up letting you browse through your files to select pictures. Upon selection, it will display in the box chosen.

**Requirement:** FUN-ADMIN-CARDS-3

**Test Case Name:** updates to Admin card-bad

**Author:** Ishani Adityan

**Description:** The Admin user can not update an existing input values or image field before saving the card deck.

**Preconditions:** The values are currently not empty when completing this step. The user has not already clicked saved and submitted the values to the database.

**Execution steps:**

1. Continue from previous step, remain on <http://localhost:3000/admin/cardset> .
2. Using the card that you have already entered values and a picture for.
3. Try to go back to the field and change any value.

**Expected output:** System will allow the texts to be changed and new image selection to occur.

**Requirement:** FUN-ADMIN-CARDS-4

**Test Case Name:** Missing values

**Author:** Ishani Adityan

**Description:** Admin user tries to click ‘Save’ without entering any values or uploading an image for one of the cards.

**Preconditions:** None.

**Execution steps:**

1. Remain on <http://localhost:3000/admin/cardset> .
2. On a admin card, enter the image and ‘primary’ value.
3. Do not enter ‘secondary’ value.
4. Move onto the next card in the set.

**Expected output:** System will show a “Required” text in red at the bottom of the ‘secondary’ value that has no input.

**Requirement:** FUN-ADMIN-CARDS-5

**Test Case Name:** Edit without permission

**Author:** Ishani Adityan

**Description:** Clicking the ‘Edit’ icon let’s the user change the values, without having it clicked it will not allow values to be written.

**Preconditions:** There are pre existing values in the card.

**Execution steps:**

1. On the <http://localhost:3000/admin/cardset>

2. Try to 'edit' values by just hovering and clicking over the fields
3. Then click on the 'Edit' button and try again.

**Expected output:** System will not allow user to enter any input when button is not clicked first.

**Requirement:** FUN-ADMIN-CARDS-6

**Test Case Name:** Image Upload - Bad

**Author:** Ishani Adityan

**Description:** Admin user trying to upload anything other than an image file should not be allowed.

**Preconditions:** The test user has a folder of images and non image files.

**Execution steps:**

1. In the <http://localhost:3000/admin/cardset>
2. Click on the image field. This prompts a pop-up to your folder.
3. Try to choose a non image file to upload.

**Expected output:** System will not show the non-image files (as they are not accepted), only image files will be selectable from the folder.

## Nonfunctional Requirements

**Requirement:** NF-SCALABILITY-1

**Test case name:** Support Four Players

**Author:** Priyanka Brodie

**Description:** The card trading game should be able to support a maximum of four players at a single time.

**Preconditions:** User should be a player and should be on the player login page of the application.

**Execution steps:**

1. Open four tabs on your local machine, go to the following link <https://team-3.herokuapp.com/> on each tab, and enter a username for each user.
2. Press the "Login" button to enter the application. Do this for each of the four users.

**Expected output:** The application should direct users to the main game page of the application where they can start playing the game. Each user should see three other player boxes on their game page.

**Requirement:** NF-PERFORMANCE-2

**Test Case Name:** Provide real-time, client/server communication using socket.io

**Author:** Victoria Li

**Pre-conditions:** The code should have socket.io installed on both server and client side

**Execution Steps:**

1. Navigate to <https://team-3.herokuapp.com/> and login with a username
2. The application should display a loading screen as users wait for the rest of the players to join.
3. Once 3 other users have joined, all users should be redirected to the main page with default cardset and droppable bins.
4. For one user, click the "WE'RE DONE" button.

5. The application should display a modal to all users that prompt a “you are not done just yet” dialog.

**Expected Output:** All users should be emitted a “BEGIN\_GAME” message from the server to the client using socket.io almost immediately after the 4th user joins the game . This enables all the users to be redirected to the main page to see all the cards. All users should be emitted a “ROUND\_COMPLETE” message when the “WE’RE DONE” button is clicked and all users should be able to see the modals on their screen almost immediately after the button is clicked.

**Requirement:** NF-PERFORMANCE-3

**Test Case Name:** NF-Real Time-Hand-Good

**Author:** Jason Wang

**Description:** If properly implemented, when a transaction is successful, the cards in the hands of the users participating in the transaction will automatically update.

**Preconditions:** The user has access to the player login URL

**Execution steps:** The user inputs the player URL and player usernames in four tabs, in the first tab click and drag a card from the player’s hand to the first trade bin in the top left corner. In the second tab, select the “accept” icon.

**Expected output:** The card in the proposed trade should automatically enter the 2<sup>nd</sup> tab’s “hand” while the card will disappear from the 1<sup>st</sup> tab’s hand.

**Requirement:** NF-PERFORMANCE-3

**Test Case Name:** NF-Real Time-Hand-Good

**Author:** Jason Wang

**Description:** If improperly implemented, if a transaction is successful, the cards in the hands of the users participating in the transaction will automatically update.

**Preconditions:** The user has access to the admin URL

**Execution steps:** The user inputs the player URL and player usernames in four tabs, in the first tab click and drag a card from the player’s hand to the first trade bin in the top left corner. In the second tab, select the “accept” icon.

**Expected output:** The hands of all of the player tabs will not update (will continue displaying the same cards).

**Requirement Name:** NF-SECURITY-4

**Test Case Name:** Administrator username and password requirements

**Author:** Danial Betres

**Requirement Description:** The admin user shall require a username and a password containing at least 6 characters, with one capital and one number, to enter the admin page

**Preconditions:** None

**Execution steps:**

1. Go to <http://localhost:3000/admin/cardset>
2. Enter a username and password that does not satisfy the requirements. For example less than 6 characters in length or no capital or numbers in the field.



**Expected output:** An error should be produced informing the user of the exact violation they are committing so that the user know what needs to be fixed.

**Requirement:** NF-CAPACITY-5

**Test Case Name:** Upload Capacity - Bad

**Author:** Ishani Adityan

**Description:** A file that's more than 1MB will not be allowed to be uploaded as an image for the cards.

**Preconditions:** Test user has an image file that is more than 1MB in size.

**Execution steps:**

1. Remain on <http://localhost:3000/admin/cardset> .
2. Click on image field
3. Try to chose the image file in your folder that's greater than 1MB.

**Expected output:** The prompt will not close indicating that another file should be chosen (as this one exceeds system capacity).

**Requirement:** NF-SECURITY-6

**Test Case Name:** Secure admin login

**Author:** Ayat Turar

**Description:** The application will provide a login functionality for the admin user and store his/her credentials securely.

**Preconditions:** Obtained admin credentials

**Execution steps:** Navigate to <http://localhost:3000/admin/> login. Obtain the admin credentials from the developer team and input the email and password.