Aaron Alexander
101107249

COMP2406 Assignment 5

To setup the program:

un-zip file
open termianl inside the directory
run "npm install"
start Mongo DB Daemon
run "node database-initializer"
run "node store-server.js"
open browser to "http://localhost:3000/"

How to test Program: (say how to test)

Registration, Logging In/Out

My login page works as both a login page and a registration page. If a user enters and a new username name the program creates and new user with the given username and password and allocates 10 random cards to the user. If a users enters a username of an account that already exists the system checks if the password

Test Registration:

1)navigate to "http://localhost:3000/" and enter a new username and create a user.
2) enter existing username and to create user (and see error message)

Test Login:

1) enter existing username and correct password
2) enter existing username and enter incorrect password (and see error message)

Test Logout

1) once logged in, click logout button at top of page.

Friends

Test searching for friends and sending friend requests

1) once logged in search for other users by entering text into the search bar and clicking search. Users with names including test from the search bar will appear below with a button allowing users to send friend requests, click the button to send friend requests. You will not receive a notification if your request is accepted. It see new friends you must refresh you page

Cannot become friends with the same person twice

1) send a friend request to the user who is in your friend list and a message will be displayed. You also cannot send a friend request to yourself

Real time friend proposal

1)Log into one account and log in to another account from another browser(incognito) and send an friend request to one of the users logged in, You will send the friend request appear in real time

Accept and reject friend requests

1) Click accept on a friend request and view alert message(you must refresh the display to see the newly added friend)
2) Click reject on a friend request and view alert message

View other friends cards and friends

1) click on the friend name inside of your friends list and you will be able to see their friends and cards.

Trades

Propose trade request to friends.

1) Under propose a trade request select yours cards that you would like to offer, select a friend from the drop down and select his cards that you would like to request, then hit the submit trade request button to submit the trade

Real time trade request

1)Log into one account and log in to another friend account from another browser(incognito) and send a trade request to one of the users logged in, You will send the trade request appear in real time

View trade request

1) Click on a trade request under pending trade requests and you will be able to view the trade details

Accept trade request

1) Click on trade request, then click accept at the bottom of the trade request page. See message. User who sent request will not be notified and will have to refresh display to use changes

Reject Trade Request

1) Click on trade request, then click decline at the bottom of the trade request and see the message. User who sent request will not be notified

Invalid Trade Request

1) Login to an account and make sure you have at least two friends, if not create and add these friends. Create a trade request with one of your friends, Create another trade request with another friend and offer the same cards. Login to the first friend account and accept the trade request, then login to the second friend account and accept the trade request. The trade will be invalid, view error message and the trade will be canceled.

Design Decision

How data is stored

Inside the database there are collections for Cards, Users, Friend Requests and Trade Requests.

Card contains a artist(string), name(string) , cardClass(string), rarity(string), attack(number) and health(number)

User contains a username(string), password(string), friends(array of user object ids) and cards(array of card object ids)

Friend Request contains to(user objectId), from(user objectId) and cardsRequested(array of card object ids)

Trade Request contains to(user objectId), from(user objectId), cardsRequested(array of card object ids) and cardsOffered(array of card object ids)

Server Routes

The store-server has a userRouter(/users) cardRouter(/cards) friendRequestRouter(/friendRequests) and tradeRequestRouter(/tradRequests)

Justification / General explanation

This program creates users and gives them cards.

When users request a friend it creates a friend request and it is stored in the database. When users are logged in they are polling to search the database from friendRequest where the to is the current user is logged in. The friend request is sent to the client. If it is accepted the users are added to the others friend list and the friends request is deleted from the database

When users request a trade it creates a trade Request with the users involved and the cards involved and store it in the database. When users are logged in they polling to search the database from tradeRequests where the to is the current user logged in. The friend request is sent to the client. If it is accepted the cards are swapped and the trade request is deleted from the database.