**Functions Needed:**

(input & output)

Auction

* Is the auction ended?
  + **Input:** ticket id
  + **Output:** boolean
* Is there an auction for this ticket? DONE
  + **Input:** ticket id
  + **Output:** boolean
* Get ticket by ticket id DONE
  + **Input:** ticket id
  + **Output:** ticket object
* Get auction by ticket id DONE
  + **Input:** ticket id
  + **Output:** 
    - On success: auction object
    - On failure: Error
      * No auction exists
      * Auction ended
* Get all auctions associated with a user “DONE”
  + **Input:** user
  + **Output:** list of auctions
* Get all auctions associated with a game DONE:3
  + **Input:** game id
  + **Output:** list of auctions

Incentive Based Rewards/Penalties

* (Admin) Penalize all users who didn’t attend a game DONE
  + **Input:** game id
  + **Output:**
    - New request: JSON output on Success
      * Can just be {“success”: “success”} or something
    - On success: n/a
    - On failure: error
* Reward user for attending game (transfer $$, set ticket to attended)
  + **Input:** ticket id
  + **Output:** 
    - On success: new user balance
    - On failure: error
      * Already attended
      * Transfer failed
* Get tickets owned by user whose games occur after today (active tickets) DONE
  + **Input:** user
  + **Output:** list of tickets
* Get game type as string DONE
  + **Input:** game type enum
  + **Output:** game type as string

DONE

* Get tickets owned by user whose games occurred before today DONE
  + **Input:** user
  + **Output:** list of tickets
* Get # of lottery tickets for game DONE
  + **Input:** game id
  + **Output:** (int) number of lottery tickets for this game
* Get users entered in lottery for game DONE
  + **Input:** game id
  + **Output:** list of users entered in lottery

Admin

* Transfer HTK to account DONE
  + **Input:** user to transfer to, amount, message
  + **Output:**
    - On success: user’s new balance
    - On failure: error messages
      * Hokipoki has insufficient funds???
      * Not sure what else will go wrong
* Execute lottery DONE
  + **Input:** game id
  + **Output:** JSON (however you guys did the other ones)
    - On success: n/a
    - On failure: error messages
      * Lottery not opened
      * Lottery already executed
      * Game already passed
      * No tickets available for lottery
* Open lottery DONE
  + **Input:** game id
  + **Output:** JSON (however you guys did the other ones)
    - On success: n/a
    - On failure: error messages
      * Lottery already opened
      * Lottery already executed
      * Game already passed
      * No tickets for game

Getters

* Transfer history Niles
  + **Input**: user
  + **Output**: all transfers to/from a user’s account.
    - Per transfer: date, amount (positive for sell, negative for buy?), memo (“VT vs. UNC”)
* Get all games on/after current date DONE
  + **Input:** date
  + **Output:** list of games with dates >= input date
* Get all tickets owned by a user DONE
  + **Input:** user
  + **Output:** list of tickets.
    - Game id, ticket type (for\_lottery), ticket sellback price
* Get game by id DONE
  + **Input:** game id
  + **Output:** game json info

Boolean Checks

* Does game have tickets available for purchase and game is not past DONE
  + **Input:** game id
  + **Output:** bool
* Does game have tickets available for lottery and lottery is open DONE
  + **Input:** game id
  + **Output:** bool
* Check if user has already bought game ticket DONE
  + **Input:** user, game\_id
  + **Output:** bool
* Check if user is in lottery DONE
  + **Input:** user, game\_id
  + **Output:** bool

Actions

* Buy a ticket DONE
  + **Input:** user, game id
  + **Output:** 
    - On success: New balance for user
    - On failure: Error messages
      * User already owns a ticket for this game
      * User doesn’t have enough HTK
      * Game tickets are not available
      * Game has already passed
* Sell ticket DONE
  + **Input:** user, ticket id
  + **Output:** 
    - On success: New balance for user
    - On failure: Error messages
      * User does not own this ticket
      * Game has already passed
      * Sale period is not open
* Enter lottery DONE
  + **Input:** user, game id
  + **Output:**
    - On success: No info needed
    - On failure: Error messages
      * No lottery tickets available
      * User already entered the lottery
      * Before lottery open date
      * After lottery close date
* Leave lottery DONE
  + **Input:** user, game id
  + **Output:** 
    - On success: No info needed
    - On failure: Error messages
      * Before lottery open date
      * After lottery close date
      * User doesn’t own lottery ticket
* Execute lottery DONE
  + **Input:** game id
  + **Output:** bool

Student Pages

(can only be accessed if user is logged in as a student)

**[libgoblin]** **[command]** doesn’t exist

Account

* Display user id **[get\_user\_info(user)]**
* Current balance of HTK **[get\_balance(user)]**
* Transaction history

Buy

* Games available to buy from (all future games) + ticket price **[games]**
  + game date
  + game start time
  + lottery open/close dates
  + Game title (e.g. “Virginia Tech vs. University of North Carolina”)
  + Game location
* Buy button **[buy name user, uint64\_t ticket\_id]**
  + Handled as a pop-up window
    - “Cancel” button
    - Warn of new balance
    - Confirm purchase
    - Success screen
    - “Go back” button

Lottery

(keep similar to current lottery interface)

* Lottery dates for current and future games
* For available lottery dates:
  + If not entered, “Enter Lottery” button   
    **[enterlottery name user, uint64\_t game\_id]**
  + If entered, “Leave Lottery” button   
    **[leavelottery name user, uint64\_t game\_id]**
  + If lottery executed, display results

Tickets

* Display all owned tickets
  + Game information
  + Sell button -> pop-up window similar to Buy workflow
    - Sell **[sell name user, uint64\_t ticket\_id]**
* Ticket history

Admin Pages

Accounts

* View student accounts
  + Balance
  + Tickets
* Create new account

Transfer

* Choose student account
* Transfer amount

Games

* List of existing games
  + Option to execute lottery/end lottery
  + View lottery entries for that game
* Create new game