

HazzardHulen.io

by: Anders Bundgaard, Casper Beese Nielsen, Oliver Vestergaard,  
Simon Hjortshøj Larsen, Steen Schütt & Thomas Lemqvist

November 19, 2016

# Contents

<b>1</b>	<b>The Software</b>	<b>3</b>
<b>2</b>	<b>Technology</b>	<b>3</b>
2.1	Client Side . . . . .	3
2.2	Server Side . . . . .	3
<b>3</b>	<b>UML Sequence Diagram</b>	<b>3</b>
<b>4</b>	<b>UML Object Diagram</b>	<b>3</b>
<b>5</b>	<b>Walkthrough of key Code-snippets</b>	<b>4</b>
5.1	The Connection between the client and server . . . . .	4
5.2	The calls from the Client to the Server . . . . .	4
<b>6</b>	<b>How to Run</b>	<b>4</b>
<b>7</b>	<b>References</b>	<b>5</b>

# 1 The Software

HazardHulen.io is a multiplayer blackjack table, where players can bet virtual currency to participate in a round of blackjack. The game uses the same ruleset as regular blackjack played in casinos, with a few details left out for simplicity. Players play against an artificial intelligence which acts as the dealer, and win double their bet if they beat the AI. If a player loses, his bet is lost. All players get a set amount of virtual money when they join the table, and have to place a bet of at least a minimum amount to play. If a player does not place a bet within a certain time period in between rounds, he is labeled as inactive, and cannot take part in the next round.

## 2 Technology

This section describes the technologies used to implement this system.

### 2.1 Client Side

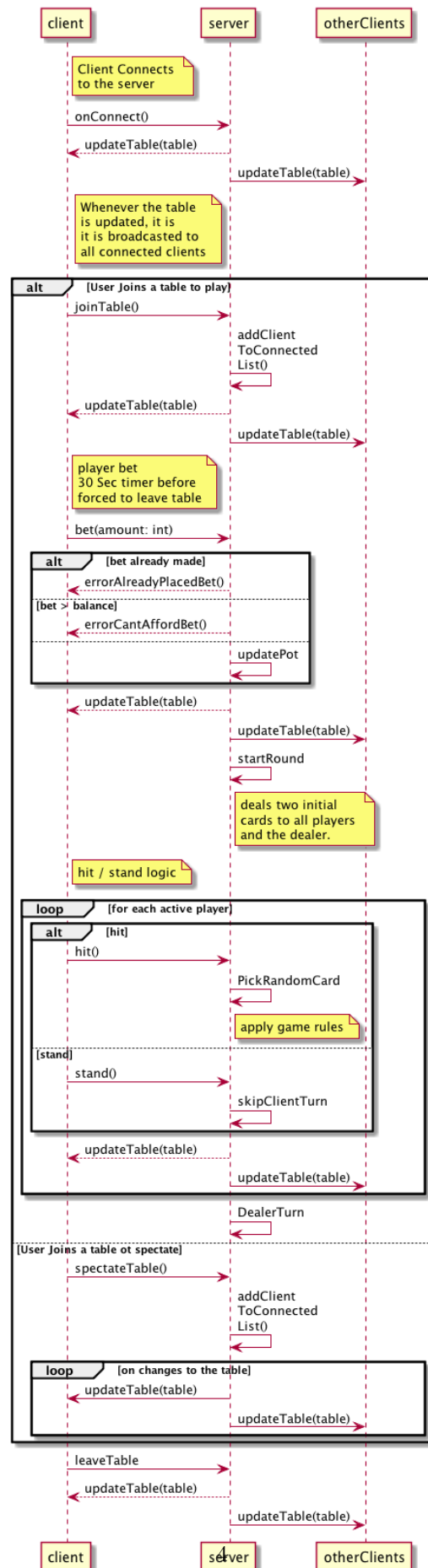
The client is written in *HTML* and *jQuery*, which uses *socket.io*, which is a web socket library, to connect to the server.

### 2.2 Server Side

The server is written in *JavaScript* and uses *socket.io* to communicate with the clients.

The server utilizes the duplex connection, provided by *socket.io* to synchronize the state of the game across all clients

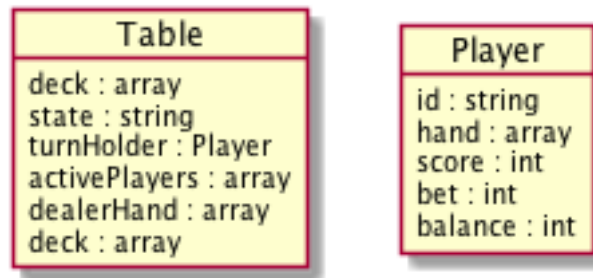
### 3 UML Sequence Diagram



The sequence diagram, shown above, shows all calls made between the server and the client.

The server is responsible for synchronizing the table across all clients. It uses the table object (*as seen in the object diagram below*) for this, in the `updateTable` method.

## 4 UML Object Diagram



The object diagram contains the two key objects used in this project, *Table* and *Player*.

\* The types set in the fields in the objects, are not according to UML standards.

## 5 Walkthrough of key Code-snippets

### 5.1 The Connection between the client and server

```
var express = require('express');
var app = express();
var server = require('http').createServer(app);
var io = require('socket.io')(server);

// ... omitted for brevity

server.listen(13337);
```

The code snippet above is responsible for making the server serve the clients the last line, sets the port, on which is listened on for requests to the server.

```
var port = 13337;

// ... omitted for brevity,

var socket = io.connect("http://localhost:" + port + "/");
```

The code snippet above is taken from the client, and makes it connect to the server, in this case it connects to the localhost, on the port 13337.

## 5.2 The calls from the Client to the Server

```
io.on('connection', function (client) {  
  
  // ... omitted for brevity  
  
  client.emit('setId', id);  
  client.emit('updateTableState', table);  
  
  client.on('joinTable', function () {  
  
    // ... omitted for brevity  
  
  });  
  
  // ... omitted for brevity  
  
});
```

The code snippet above is taken from the server, and it is responsible for taking commands from the client.

## 6 How to Run

To run the server, you have to have npm (it can be downloaded along with node here: <https://nodejs.org/en/download/>).

Run the following commands to start the server:

```
$ npm install
```

Install bower by running this command, in the command line to install bower: (from: <https://bower.io>)

```
$ npm install -g bower
```

To start the server, now you simply just has to run the following command.

```
$ npm start
```

Now the server should be up and running.

To connect a client to the server, go to <http://localhost:13337/>

## 7 References

### npm packages

<https://nodejs.org/en/download/>

<https://bower.io>

<https://lodash.com>

<http://socket.io>

<https://www.npmjs.com/package/monolog>

## Helpful Links

<http://www.programwitherik.com/socket-io-tutorial-with-node-js-and-express/>

## Information about Blackjack

<https://en.wikipedia.org/wiki/Blackjack>