MSC IT+ MASTERS TEAM PROJECT
Team 31

2424921G, Nicholas Gibb
2412377R, Philip Rodger
2413547K, Berlian Khatulistiwa
2026124M, Ryan Maloney

Contents (of Documentation):

1. Introduction
2. Method
3. MVC Diagrams
4. Users
5. User Stories
6. Story Cards
7. Release plan / Development Sprints
8. Burndown Chart Examples
9. Website Design
10. Assumptions
11. Testing
12. Deficiencies
13. Conclusion
14. Screenshots
15. Appendix

Introduction

The following development document is a companion to our Top Trumps Team project, along with this document is included a full file of the burndown chart, (an example of the file can be found in this example).

Various applications were used in the creation of this project such as:

- *Slack*, for team communication.
- *GitHub,* for the code to be housed and allowing all members to clone it and contribute.
- *Trello,* for the design of the burndown chart and overall timekeeping requirements.
- *Lucid Chart,* for the creation of the MVC diagrams.

Method

As stated in the project brief, we followed the Scrum Agile development method, the development was a little longer than the 30-day cycle (outside interference was the reason for this) from 9/01/2019 to 18/02/2019. The process still consisted of two sprints over that time.

Total Days Available:
41 (Amount of inclusive days between January 9th (day of project briefing) and February 18th (deadline of the project).

Story points Available per project:
22

Total Story Points to complete System Implementation:
22

Total User Stories:
9

Number of Sprints:
2

Estimated Sprint Lengths:
Sprint 1: 21 days
Sprint 2: 14 Days

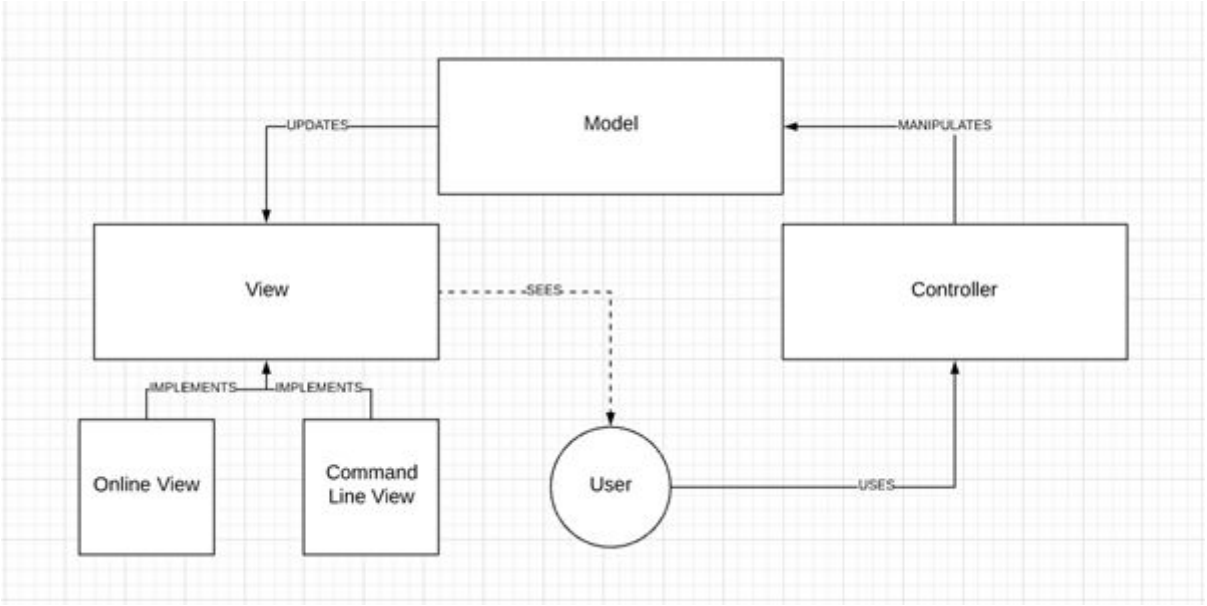
Actual Spring Lengths:
Spring 1:21 Days
Sprint 2: 14 Days

Estimated story points team consumed in each sprint:
Sprint 1: 13 points
Sprint 2: 9 points

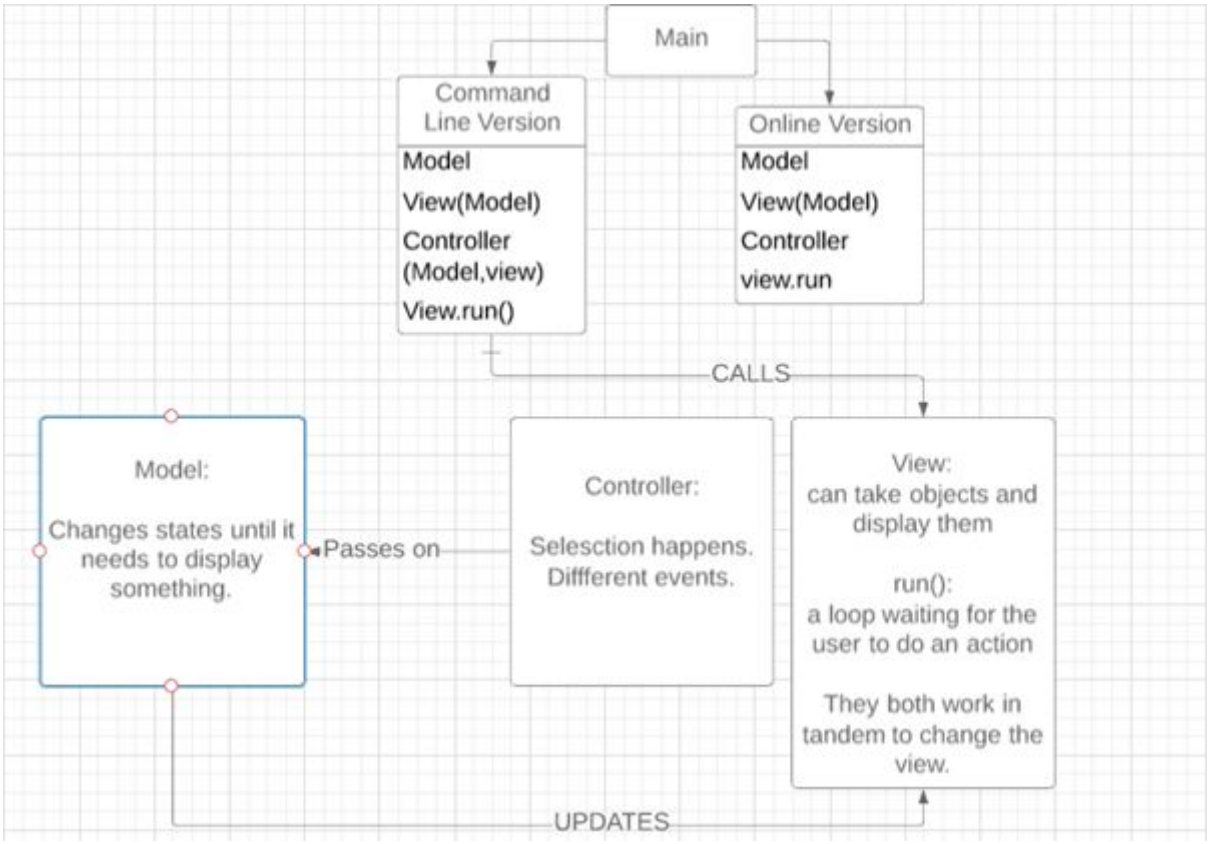
Actual amount of story points team consumed in each Sprint:
Sprint 1: 13
Sprint 2: 9

By the end of the last iteration, all 20 available story points will have been consumed. By this point, the project will be fully implemented. Release plan consists of 2 sprints with the first sprint focused on implementing the command line version and the 2nd sprint on implementing the web version.

MVC Diagrams



(A basic MVC of how the application will be implemented. Most of these (excluding the Online View) will be completed during the first sprint.



(In-depth look at the model)

Users

Below is a group of users we are basing the stories on.
The two main users for the program would be the player user, this user is focused on the running of the program, recording their games and viewing text files containing the information about the game. (i.e. Rounds won, rounds lost etc.)

The other is the developer user.
This user could be seen as one or a few different developers focussing on different parts of the code such as the database model and the web application development.

User Stories

These stories have the following points:

Must-Have

- Play Top Trumps on the command line version - **estimated 5 points, actual = 5** (This would be considered the Epic of the work, we break down this later into individual points
- Play Top Trumps on the web version - **estimated 4 points, actual = 4** (As above this is considered another epic, there are various steps to make this work.

Should Have:

- Write a test log to file that contains snapshots of the program's state as it runs on the command line version - **estimated 3 points, actual 3**
- Store results of played games in a database - **estimated 3 points, actual 3**
- View overall game statistics of results of played games on command line version - **estimated 2 points, actual 2**
- View overall statistics of results of playing games on the web version - **estimated 2 points, actual 2**
- Play multiple games simultaneously on different web browsers - **estimated 1 points, actual 1**

Could Have:

- Cards have pictures - **estimated 1 points, actual 1**
- Create sleek web version design - **estimated 1 points, actual 1**

Story cards not implemented:

- Have AI opponents that make more sensible decisions
- Select a difficulty level for the AI players
- Slow down or speed up the pace of the AI opponents moves

Story Cards

Story Card 1

Front Side:
Story Name: Play Top trumps on the command line version
Story: "As a user, I need to be able to play the game on the command line
Conversation: User enters input into the terminal that starts a game
Priority: Must
Cost: 5 Points

Back Side:
Tests: -Game auto resolves once the user loses
-Can play through a whole game

Story Card 2

Front Side:
Story Name: Play top trumps on the web version
Story: "As a user, I need to be able to play the game on a web browser"
Conversation: Use starts a game with a button from a game selection screen
Priority: Must
Cost: 4 points
Sprint: 2

Back Side:
Tests: -Can play through a whole game on a browser
-Game auto resolves once the user loses

Story Card 3

Front Side:
Story Name: Write a test log to file that contains snapshots of the program's state as it runs on the command line version
Story: "As a user, I want to a test log of the game as I am playing it on the command line"
Conversation: When a game is played a file called "Testlog" is created or overwritten with information from the game
Priority: Should
Cost: 3 points
Sprint: 1

Back Side:
Tests: -Test log correctly contains information about a game that was played
-Run with a -t flag

Story Card 4

Front Side:
Story Name: Store results of played games in a database
Story: "As a user, results of games I play should be stored"
Conversation: Game automatically stores results of the game into a database once a game is completed
Priority: Should
Cost: 4 points
Sprint: 1

Back Side:
Tests: -Data is created once a game is completed
-New data from new games are made on top of old data

Story Card 5

Front Side:
Story Name: View overall game statistics from the results of played games on command line version
Story: "As a user, I'd like to be able to see the results of past games on the command line"
Conversation: Console prints out stats from previous games when an option is selected.
Priority: Should
Cost: 2 Points
Sprint: 1

Back Side:
Tests: -Statistics are formatted correctly
-Statistics reflect the actual game that was played

Story Card 6

Front Side:
Story Name: View overall statistics of results of playing games on the web version
Story: "As a user, I'd like to be able to see the results of past games on the web version"
Conversation: Game selection screen has a button that will bring the user to a page that contains relevant stats
Priority: Should
Cost: 3 points
Sprint: 2

Back Side:
Tests: -Statistics are formatted correctly
-Statistics reflect the actual game that was played

Story Card 7

Front Side:
Story Name: Play multiple games simultaneously on different web browsers
Story: "As a user, I'd like to be able to play multiple games at the same time"
Conversation: User should be able to open a window to enter game selection screen even if another window with an in-progress game is being played

Priority: Should
Cost: 1 point
Sprint: 2

Back Side:
Tests: -Starting a new game doesn't cause any currently in progress games to crash

Story Card 8

Front Side:
Story Name: Cards have pictures
Story: "As a user, I'd like the cards in the games to have pictures"
Conversation: Each card will have a unique picture
Priority: Could
Cost: 1 point
Sprint: 2

Back Side:
Tests: -Image that appears on card corresponds with the card name

Story Card 9

Front Side:
Story Name: Create a sleek web version design
Story: "As a user, I'd like the web version's design to be sleek and modern"
Conversation: User's card will be distinguishable from cards of other players
Priority: Could
Cost: 1 point
Sprint: 2

Back Side:
Tests: -Sleek design does not impair functionality

Release plan / Development Sprints

Sprint 1:

Estimated:

- This sprint will complete implementation of the command line version.
- Sprint 1 will Implement all user cards related to the command line the game.
- Total Story points consumed: 13 points

Actual:

- Total Story points consumed: 13 points

Sprint 2:

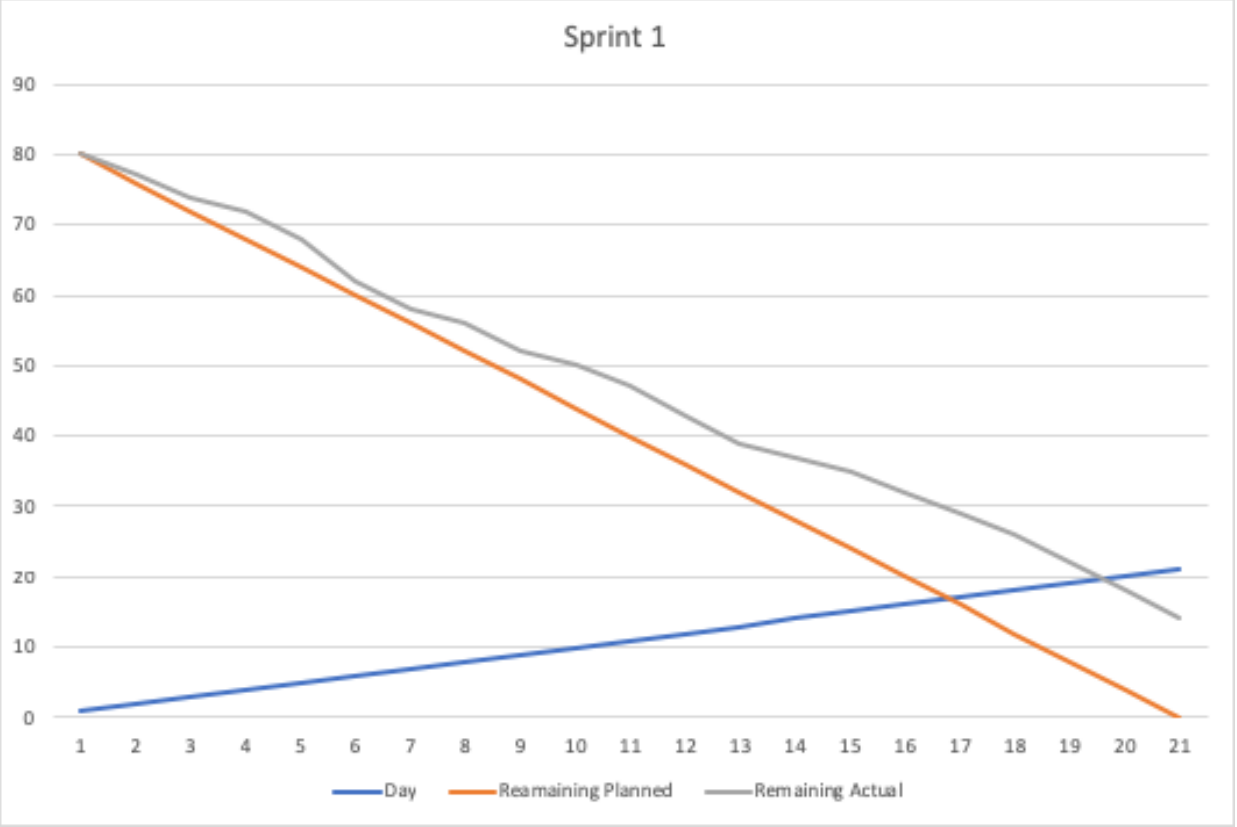
Estimated:

- This spring will implement the web version of the game. As the web version merely implements functionality that was built in the command line version, we anticipate this sprint to consume fewer points.
- Sprint 2 will implement the user cards for the web version of the game.
- Sprint 2 will also complete cosmetic user cards for the web version
- Total story points consumed: 9 points

Actual:

- Total story points consumed: 9

Burndown Chart Example

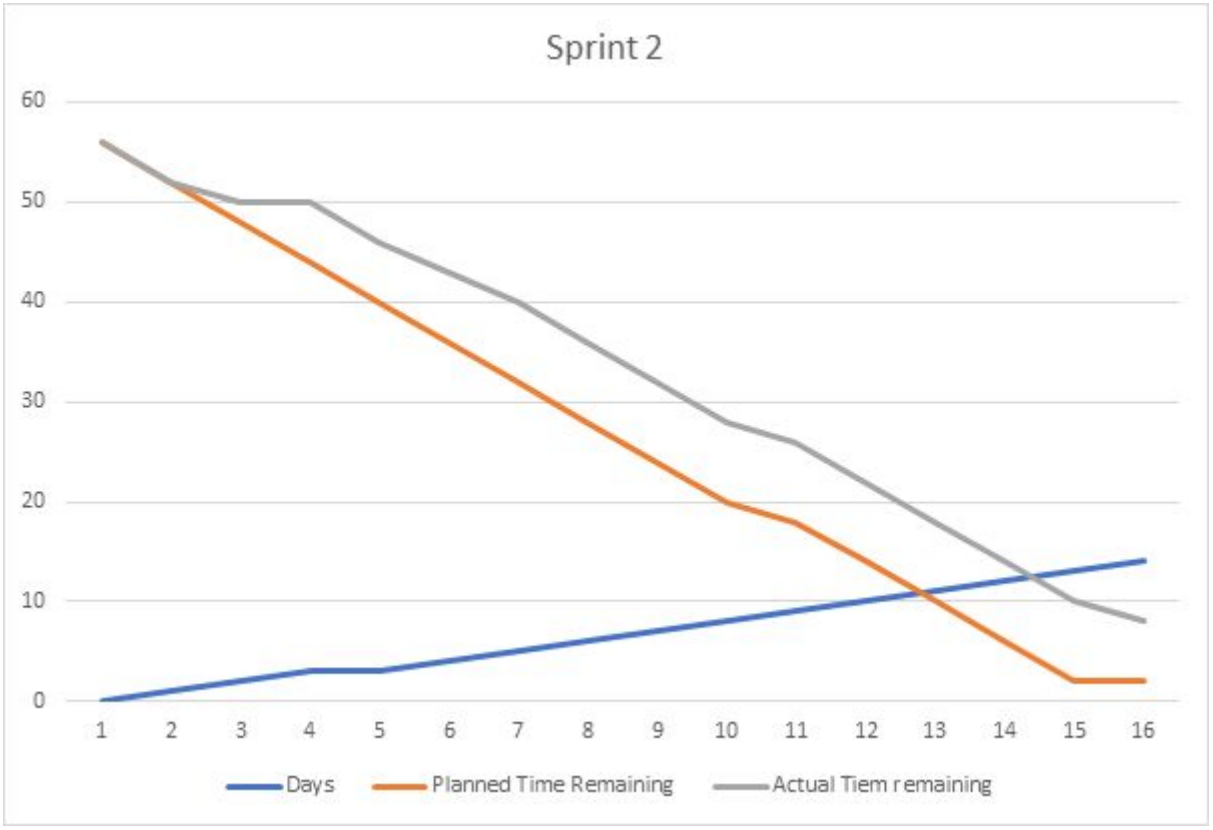


Break down of sprint 1:

This sprint was focused primarily on the development of the command line version. As you can deduce from the chart, we allotted ourselves 21 days for the first sprint. As our first meeting was rather quick and we could parse out the different roles with ease, it showed the team how quickly this first sprint would go.

The first sprint was command line focused as this was the base game that we wanted to complete as quickly as possible. The back end is the crux of the program running, we reasoned that to give this more time as in a realistic setting, the front end service or web build would be constantly updated with various decks, UI upgrades etc.
Due to time constraints of being students, we can't distribute as much time to this one project as a developer would. So the tasks run on four hour work time. With some days as seen above, we did the work quickly and other days where a break was taken or work for other courses was completed instead.

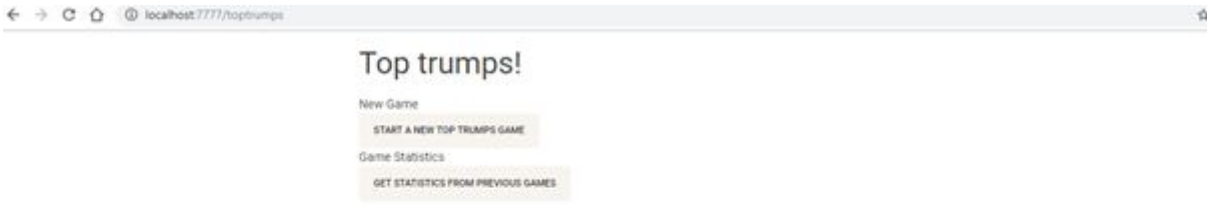
Overall the sprint was a huge success and brought about a well designed command line game that met and exceeded our expectation, resulting in the completion of all sprint related stories within the time frame.



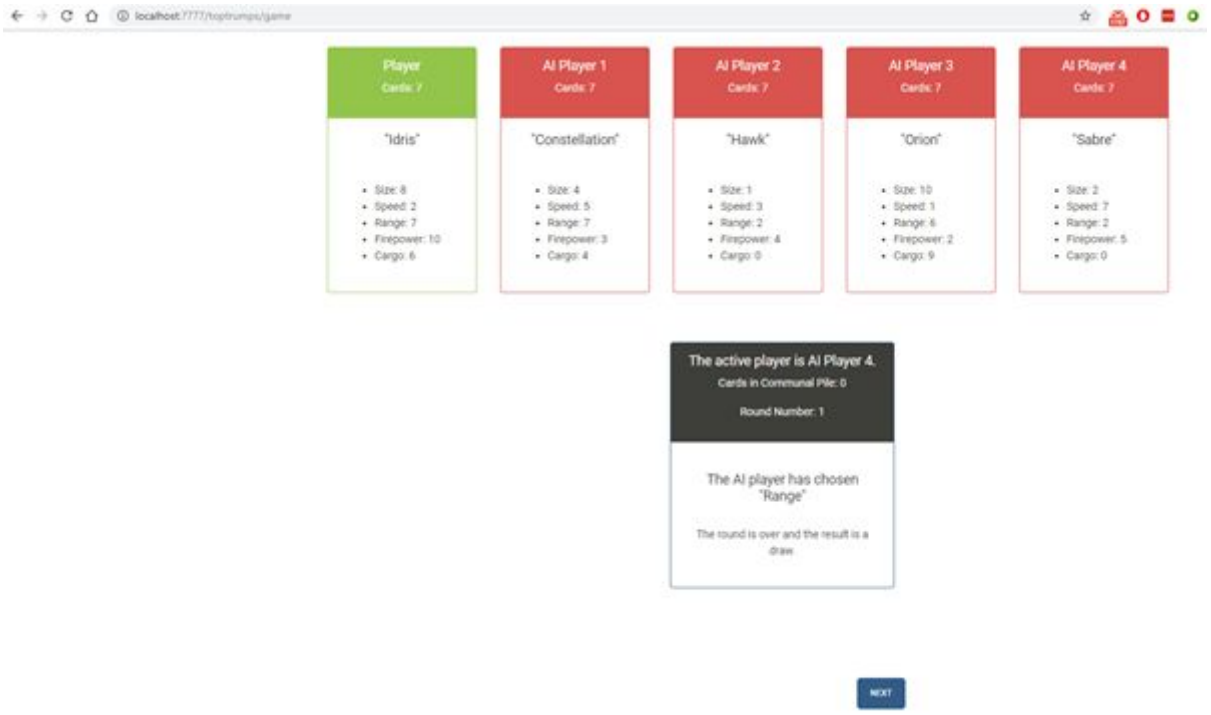
Break down of Sprint 2:

In this sprint, the main focus was the web build of the game. We decided to focus upon this section during this sprint as the web build was dependent on the functionality of the command line version. We then would have time once we converted the code to work in a web browser. After the game functioned in the browser we then spent the remaining time on this sprint on ‘dressing up’ the game page, making it more exciting to look at. Although we did meet some issues related to syncing when utilising the restful API and the front end of the gameplay in the online version, most of the functionality that was detailed in the user stories for the sprint where met within the time frame outset so that the project could be delivered on time. The current software design would allow for these issues to be addressed and fixed if more time was given to it.

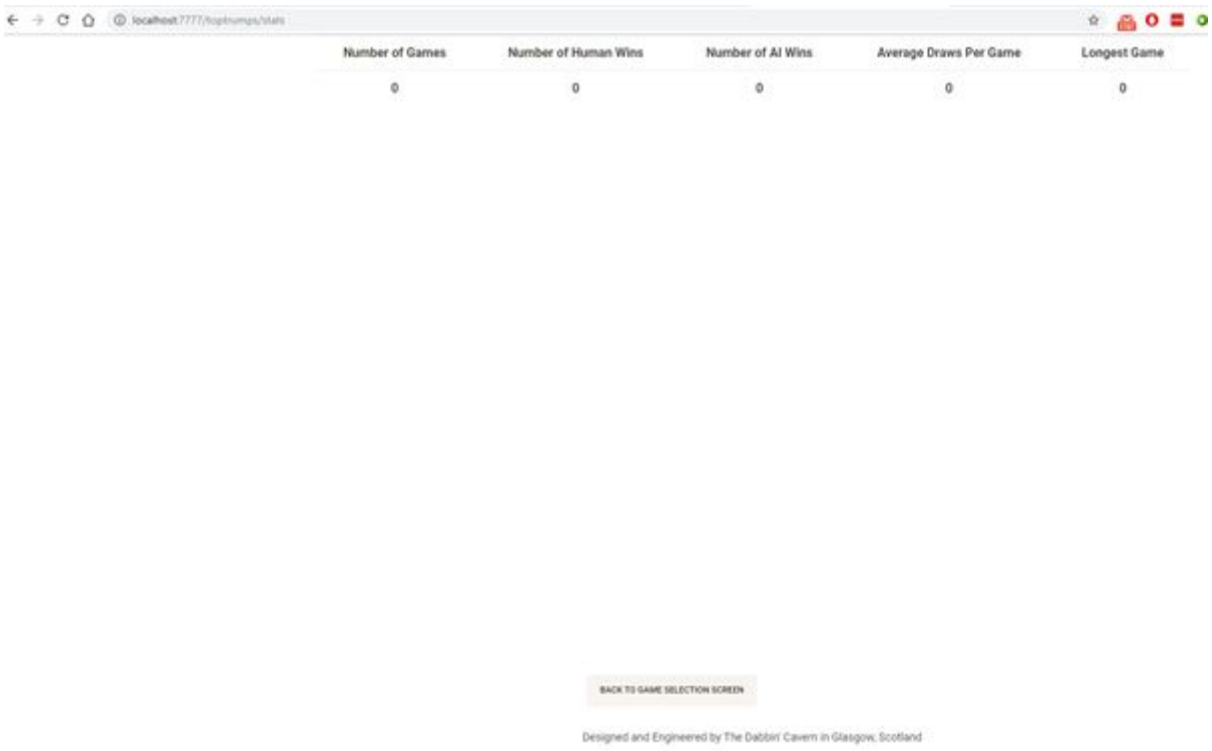
Website Design



The front page has a very simple design with minimal colours, (we will see this throughout our web design, the UI is very clean with bold colours to easily identify the different players and sections of the game.)



The main game page:
As above we have chosen the same design choices, with this format the deck can be changed out and fitted with a new topic. Later on in the development process, one may add the photos of the ship from the game. However, these were not included in the file.



A very basic stats page, this allows the player to track the data of the games played in a clear and concise format.

Assumptions

We had little to few assumptions surrounding the development of the top trumps system, one of the main assumptions was that many different users would be playing this game on varying browsers.

So we needed to make sure the game ran on various browsers, as the game runs on the web it is very unlikely that the game would need much power to run, as such most devices could run this game.

Testing

We have been testing the program throughout the lifespan of the project. The screenshots below are from the testlog.java file that we created within our model class. To keep track of game stats, card stats etc.

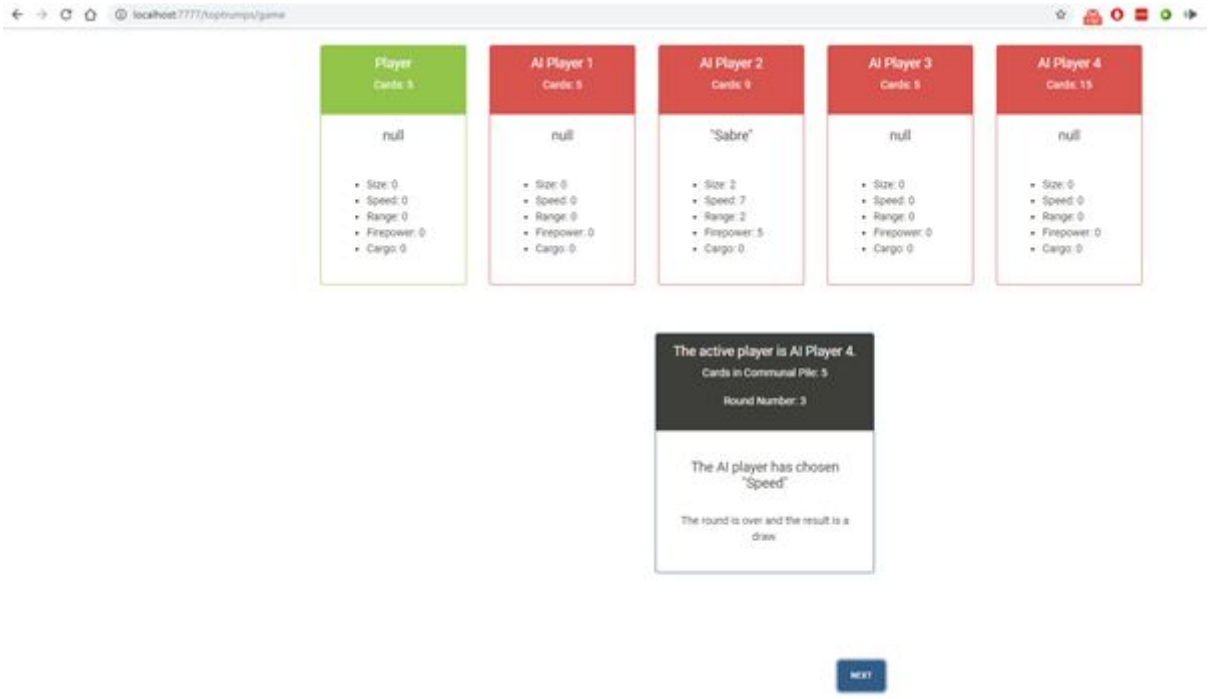
To test the RESTful API the game can be played in a headless state (without the front end) at the following URL: toptrumps/response?selection=3&update=true
Selection and update parameters are used to communicate user choices and progress the game. This was used to confirm the API was returning what was expected and also confirmed that the issues seen on the front end - old data being displayed - was due to the JavaScript and not the backend.

We had run other tests whilst building the code, in particular, some JUnit Testing, as we are covering this methodology in software engineering, we also took the time to review some web development testing services.

Deficiencies

At the time of submission, there are some deficiencies with the online version of the game.

In particular there was a bug with the Javascript which meant that players (especially the human player) were often one update behind the actual state of the game as per the REST response. It may be possible to fix this bug by changing the ordering of when particular elements of the page are updated.



Initially, in our planning phase, we had accounted for adding images to the cards but were unable to implement it into the game.

Related to the website version, the cards that have not yet been drawn are not hidden but are shown as “null” cards. This could be improved by adding a condition in the Javascript where if the card is null, to hide the card section.

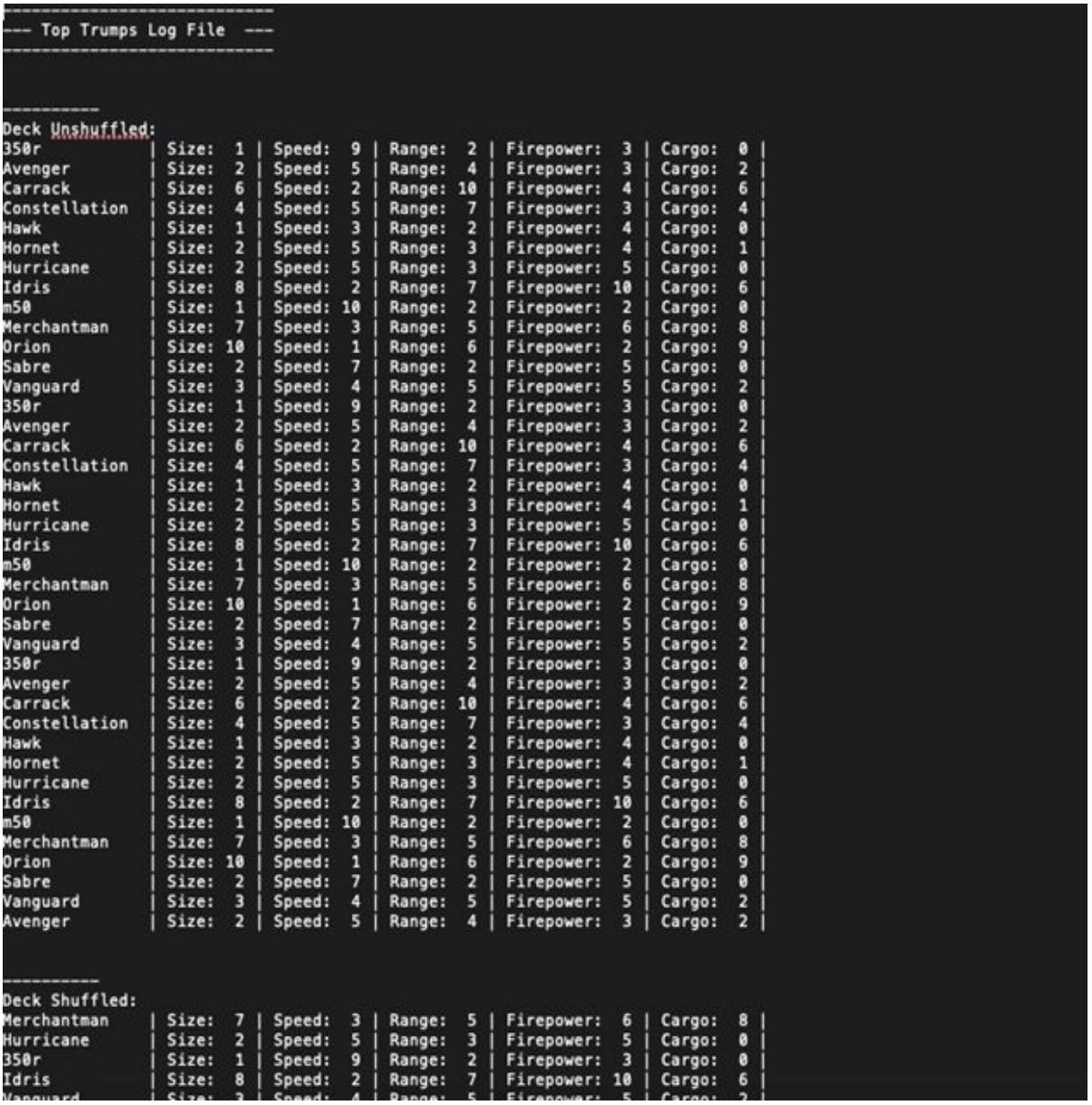
Additionally the website version does not currently support concurrent users, as the game state is currently tied to the RESTful API. To fix this a further class could be created to store different instances of the MVC and introduce a parameter to the RESTful API which uniquely identified a user, for example a cookie.

In further iterations, these deficiencies would be addressed.

Conclusion

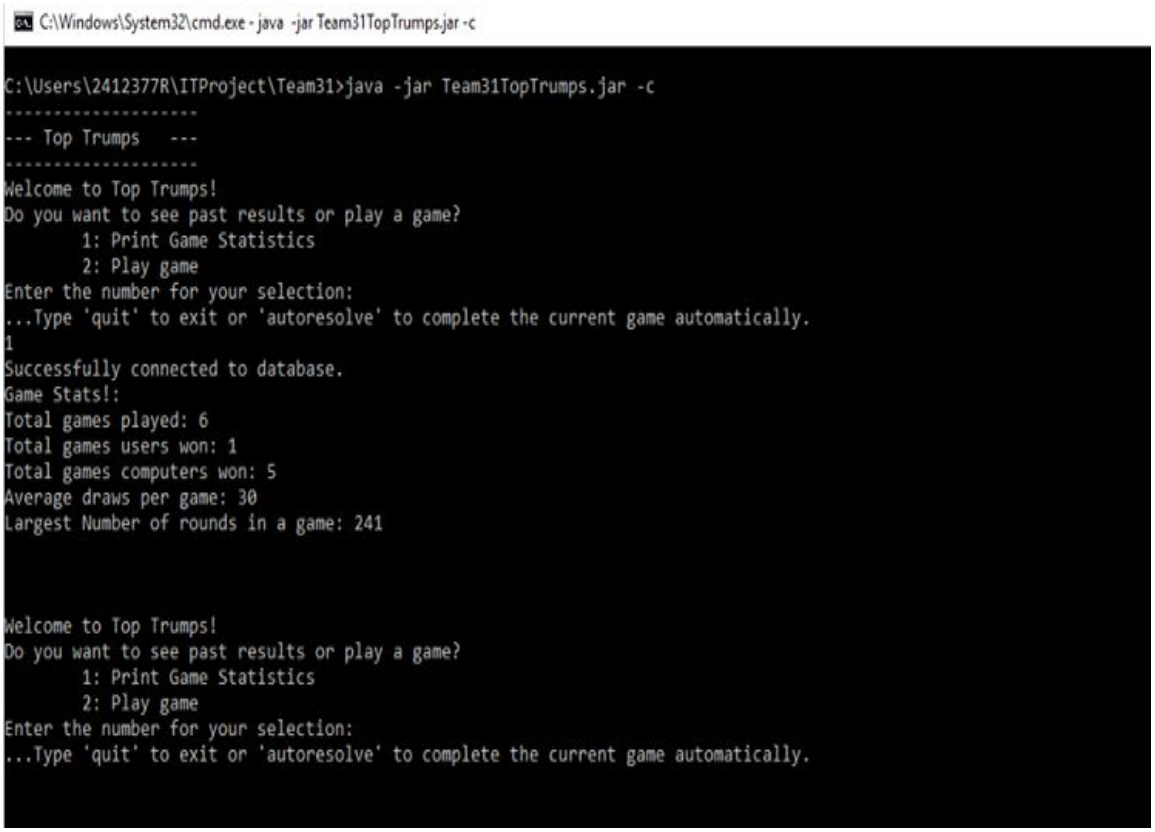
In conclusion, we have reached the end of this process with a working command line version of the game, the web version is in the process of being fixed and will continue to inspect this long after the project has been submitted. Despite the stated deficiencies, the overall quality of the software design is well done and is of a high standard overall.

Screenshots

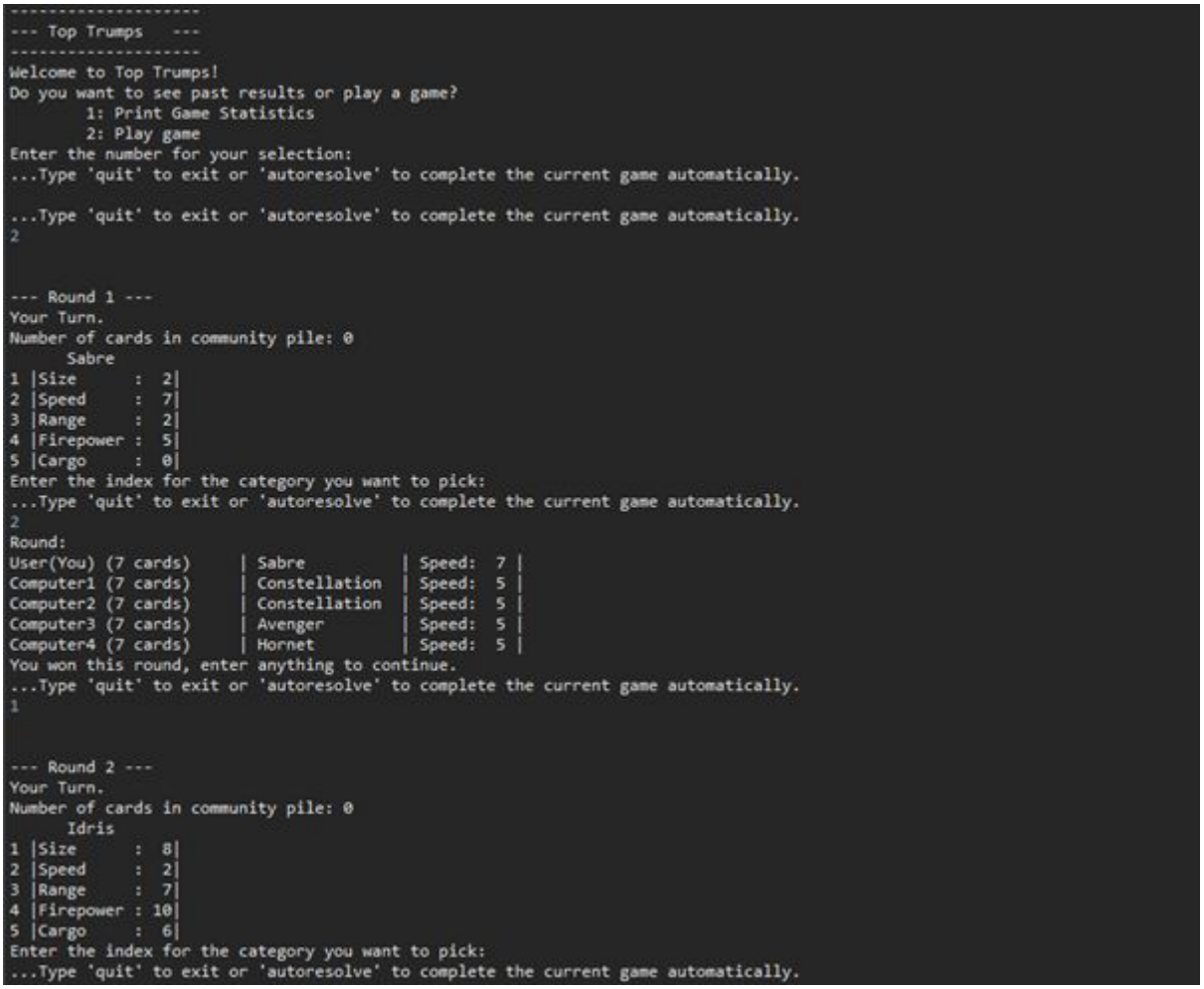


```
---------------------------
--- Top Trumps Log File  ---
---------------------------


----------
Deck Unshuffled:
350r           | Size:  1 | Speed:  9 | Range:  2 | Firepower:  3 | Cargo:  0 |
Avenger        | Size:  2 | Speed:  5 | Range:  4 | Firepower:  3 | Cargo:  2 |
Carrack        | Size:  6 | Speed:  2 | Range: 10 | Firepower:  4 | Cargo:  6 |
Constellation  | Size:  4 | Speed:  5 | Range:  7 | Firepower:  3 | Cargo:  4 |
Hawk           | Size:  1 | Speed:  3 | Range:  2 | Firepower:  4 | Cargo:  0 |
Hornet         | Size:  2 | Speed:  5 | Range:  3 | Firepower:  4 | Cargo:  1 |
Hurricane      | Size:  2 | Speed:  5 | Range:  3 | Firepower:  5 | Cargo:  0 |
Idris          | Size:  8 | Speed:  2 | Range:  7 | Firepower: 10 | Cargo:  6 |
m50            | Size:  1 | Speed: 10 | Range:  2 | Firepower:  2 | Cargo:  0 |
Merchantman    | Size:  7 | Speed:  3 | Range:  5 | Firepower:  6 | Cargo:  8 |
Orion          | Size: 10 | Speed:  1 | Range:  6 | Firepower:  2 | Cargo:  9 |
Sabre          | Size:  2 | Speed:  7 | Range:  2 | Firepower:  5 | Cargo:  0 |
Vanguard       | Size:  3 | Speed:  4 | Range:  5 | Firepower:  5 | Cargo:  2 |
350r           | Size:  1 | Speed:  9 | Range:  2 | Firepower:  3 | Cargo:  0 |
Avenger        | Size:  2 | Speed:  5 | Range:  4 | Firepower:  3 | Cargo:  2 |
Carrack        | Size:  6 | Speed:  2 | Range: 10 | Firepower:  4 | Cargo:  6 |
Constellation  | Size:  4 | Speed:  5 | Range:  7 | Firepower:  3 | Cargo:  4 |
Hawk           | Size:  1 | Speed:  3 | Range:  2 | Firepower:  4 | Cargo:  0 |
Hornet         | Size:  2 | Speed:  5 | Range:  3 | Firepower:  4 | Cargo:  1 |
Hurricane      | Size:  2 | Speed:  5 | Range:  3 | Firepower:  5 | Cargo:  0 |
Idris          | Size:  8 | Speed:  2 | Range:  7 | Firepower: 10 | Cargo:  6 |
m50            | Size:  1 | Speed: 10 | Range:  2 | Firepower:  2 | Cargo:  0 |
Merchantman    | Size:  7 | Speed:  3 | Range:  5 | Firepower:  6 | Cargo:  8 |
Orion          | Size: 10 | Speed:  1 | Range:  6 | Firepower:  2 | Cargo:  9 |
Sabre          | Size:  2 | Speed:  7 | Range:  2 | Firepower:  5 | Cargo:  0 |
Vanguard       | Size:  3 | Speed:  4 | Range:  5 | Firepower:  5 | Cargo:  2 |
Avenger        | Size:  2 | Speed:  5 | Range:  4 | Firepower:  3 | Cargo:  2 |
350r           | Size:  1 | Speed:  9 | Range:  2 | Firepower:  3 | Cargo:  0 |
Avenger        | Size:  2 | Speed:  5 | Range:  4 | Firepower:  3 | Cargo:  2 |
Carrack        | Size:  6 | Speed:  2 | Range: 10 | Firepower:  4 | Cargo:  6 |
Constellation  | Size:  4 | Speed:  5 | Range:  7 | Firepower:  3 | Cargo:  4 |
Hawk           | Size:  1 | Speed:  3 | Range:  2 | Firepower:  4 | Cargo:  0 |
Hornet         | Size:  2 | Speed:  5 | Range:  3 | Firepower:  4 | Cargo:  1 |
Hurricane      | Size:  2 | Speed:  5 | Range:  3 | Firepower:  5 | Cargo:  0 |
Idris          | Size:  8 | Speed:  2 | Range:  7 | Firepower: 10 | Cargo:  6 |
m50            | Size:  1 | Speed: 10 | Range:  2 | Firepower:  2 | Cargo:  0 |
Merchantman    | Size:  7 | Speed:  3 | Range:  5 | Firepower:  6 | Cargo:  8 |
Orion          | Size: 10 | Speed:  1 | Range:  6 | Firepower:  2 | Cargo:  9 |
Sabre          | Size:  2 | Speed:  7 | Range:  2 | Firepower:  5 | Cargo:  0 |
Vanguard       | Size:  3 | Speed:  4 | Range:  5 | Firepower:  5 | Cargo:  2 |
Avenger        | Size:  2 | Speed:  5 | Range:  4 | Firepower:  3 | Cargo:  2 |


----------
Deck Shuffled:
Merchantman    | Size:  7 | Speed:  3 | Range:  5 | Firepower:  6 | Cargo:  8 |
Hurricane      | Size:  2 | Speed:  5 | Range:  3 | Firepower:  5 | Cargo:  0 |
350r           | Size:  1 | Speed:  9 | Range:  2 | Firepower:  3 | Cargo:  0 |
Idris          | Size:  8 | Speed:  2 | Range:  7 | Firepower: 10 | Cargo:  6 |
Vanguard       | Size:  3 | Speed:  4 | Range:  5 | Firepower:  5 | Cargo:  2 |
```

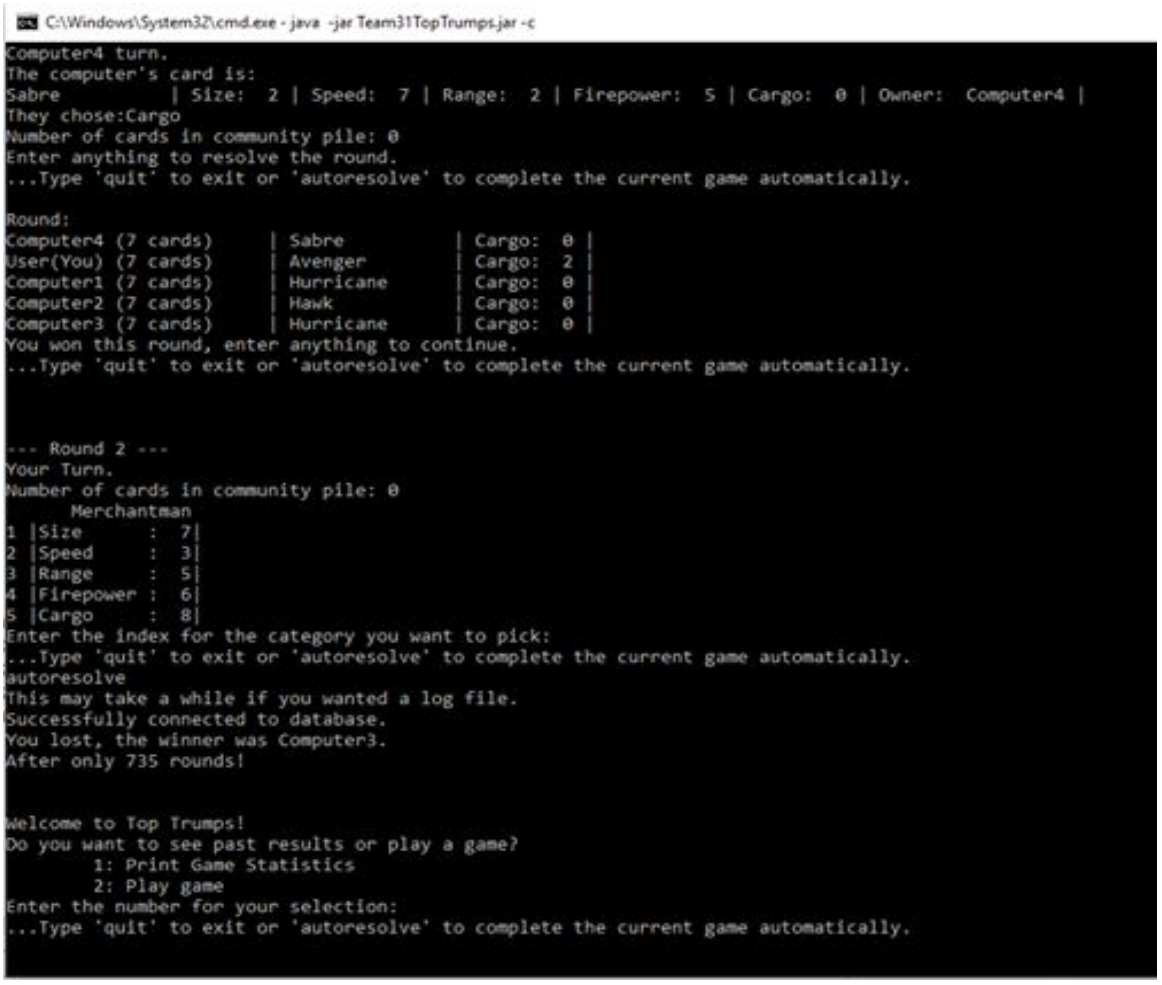
1.1 Screencap of testlog.txt



1.2 Command line start page with game stats

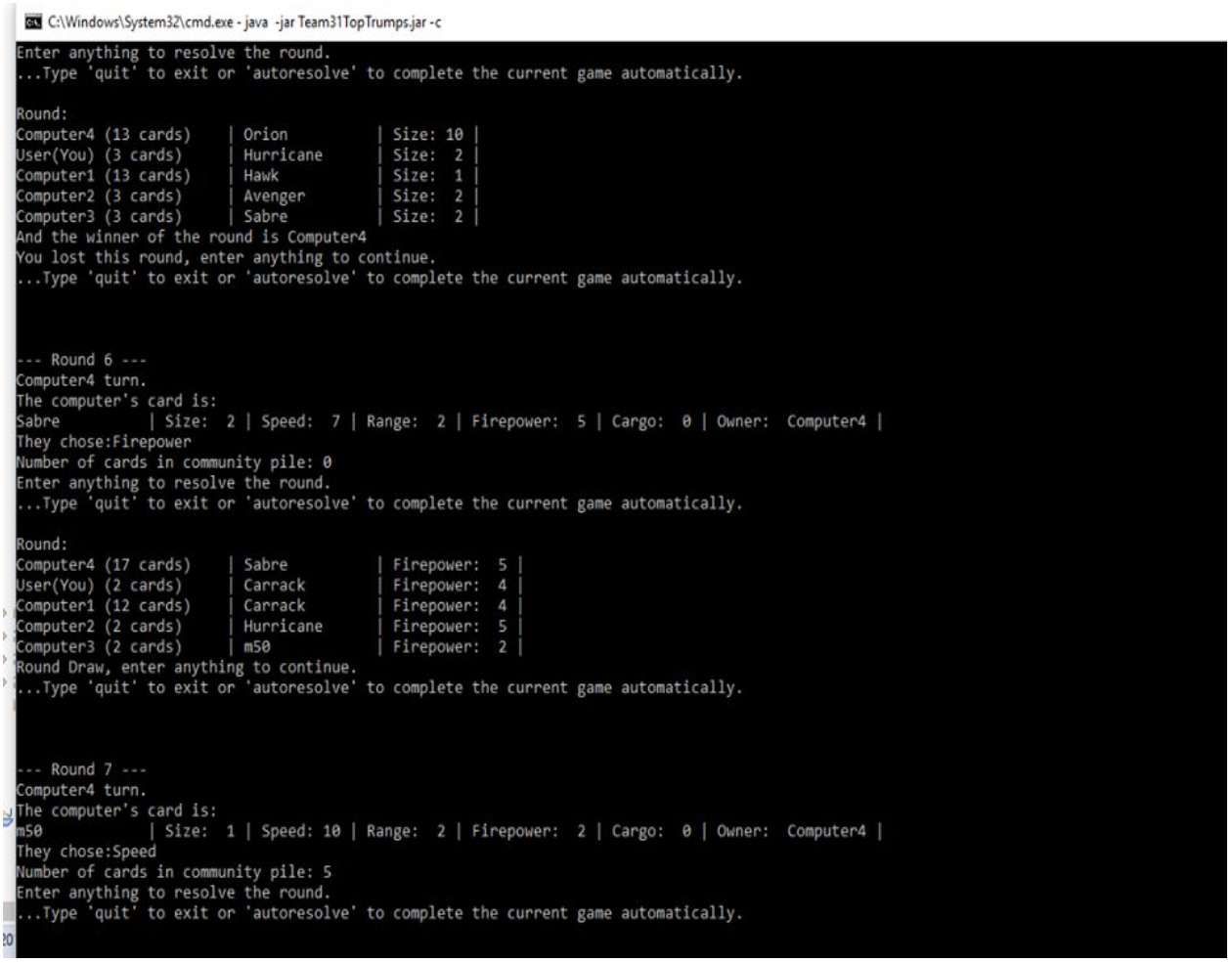


1.3 Command Line version running.

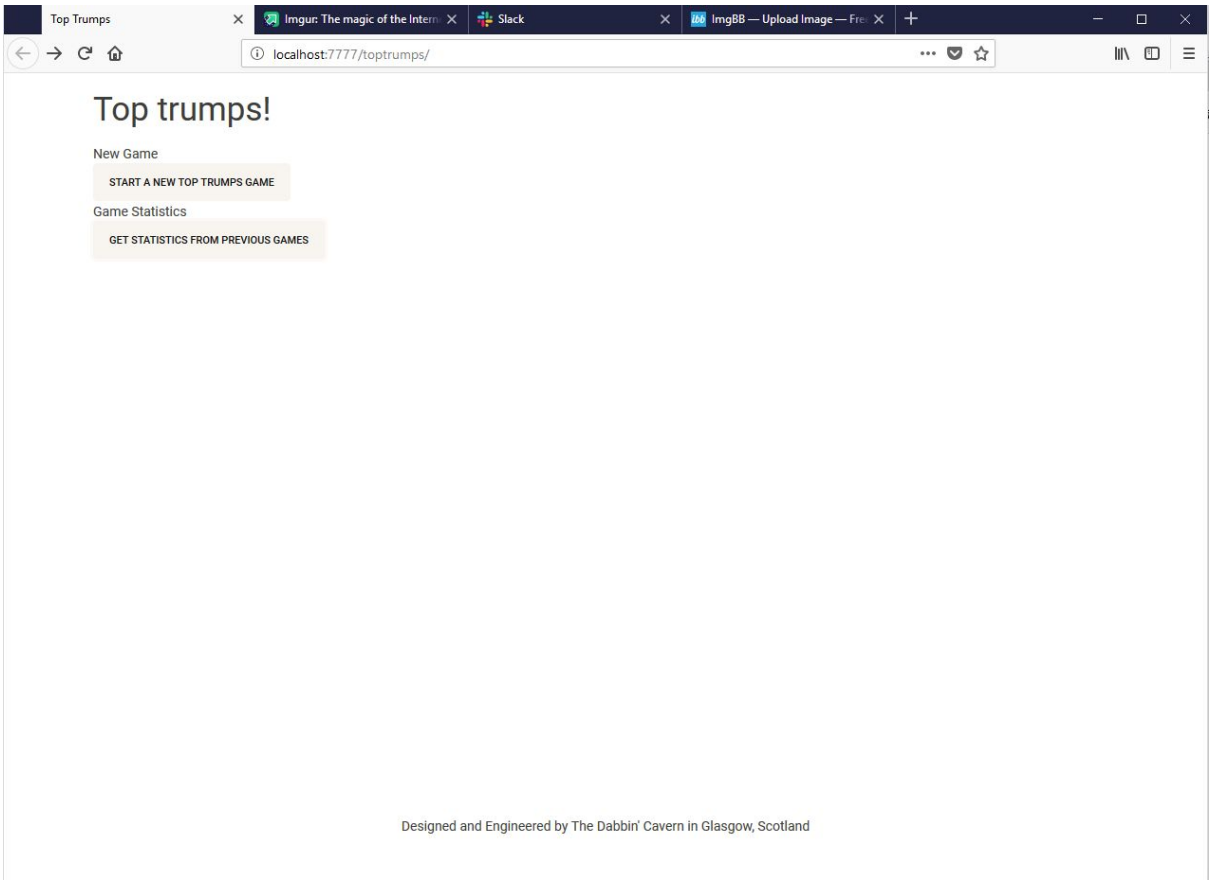


1.4 Command Line showing a game played through with the user losing.

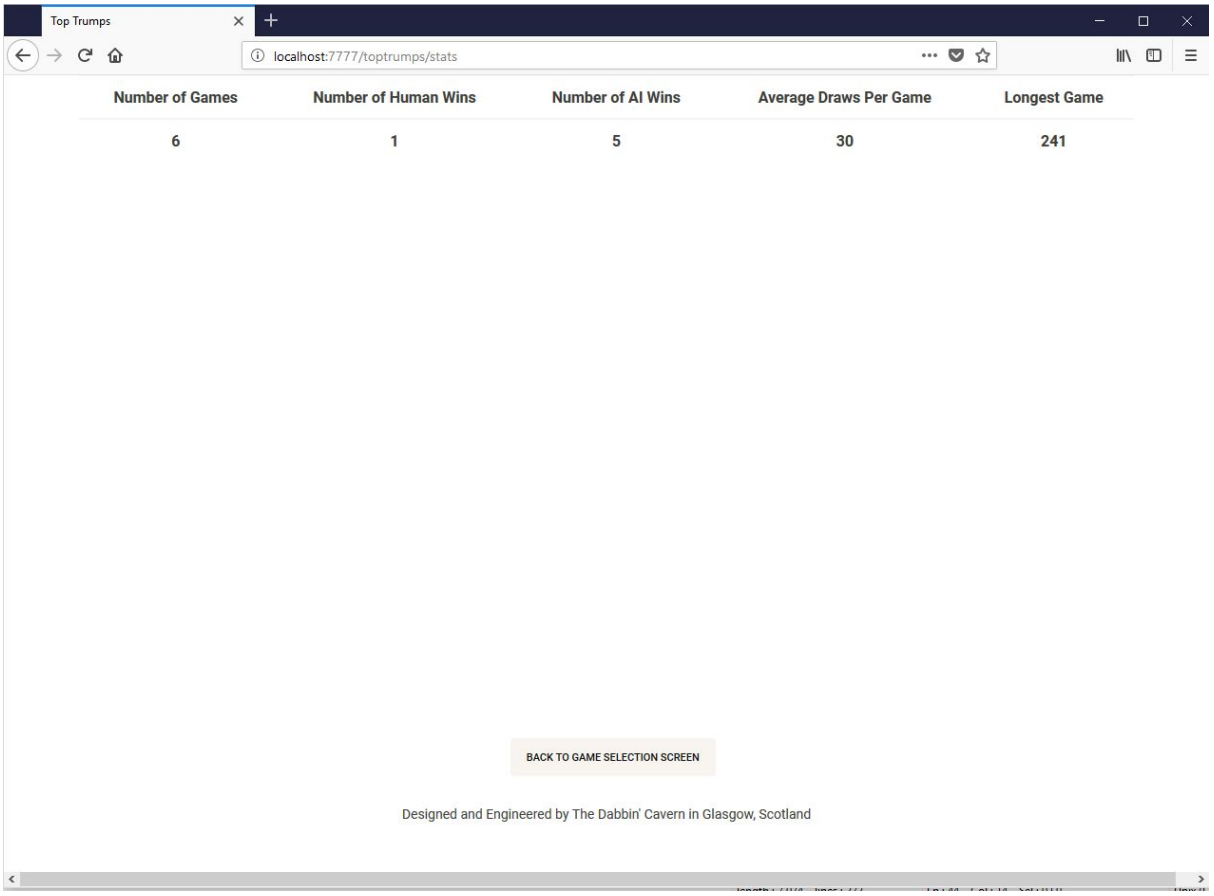
1.5 Command line showing win condition and auto resolve function



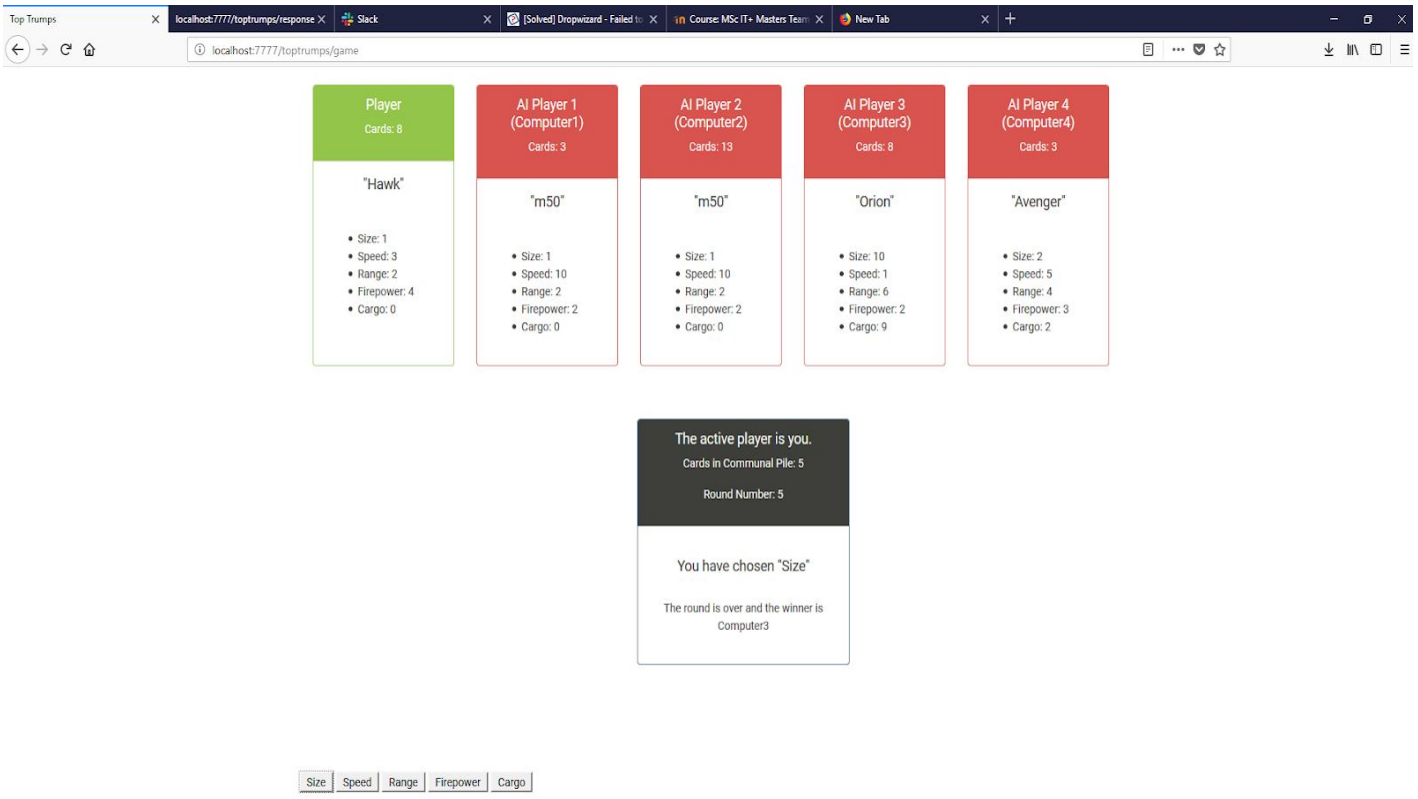
1.6 Command line game showing a round drawn condition with cards in the community pile



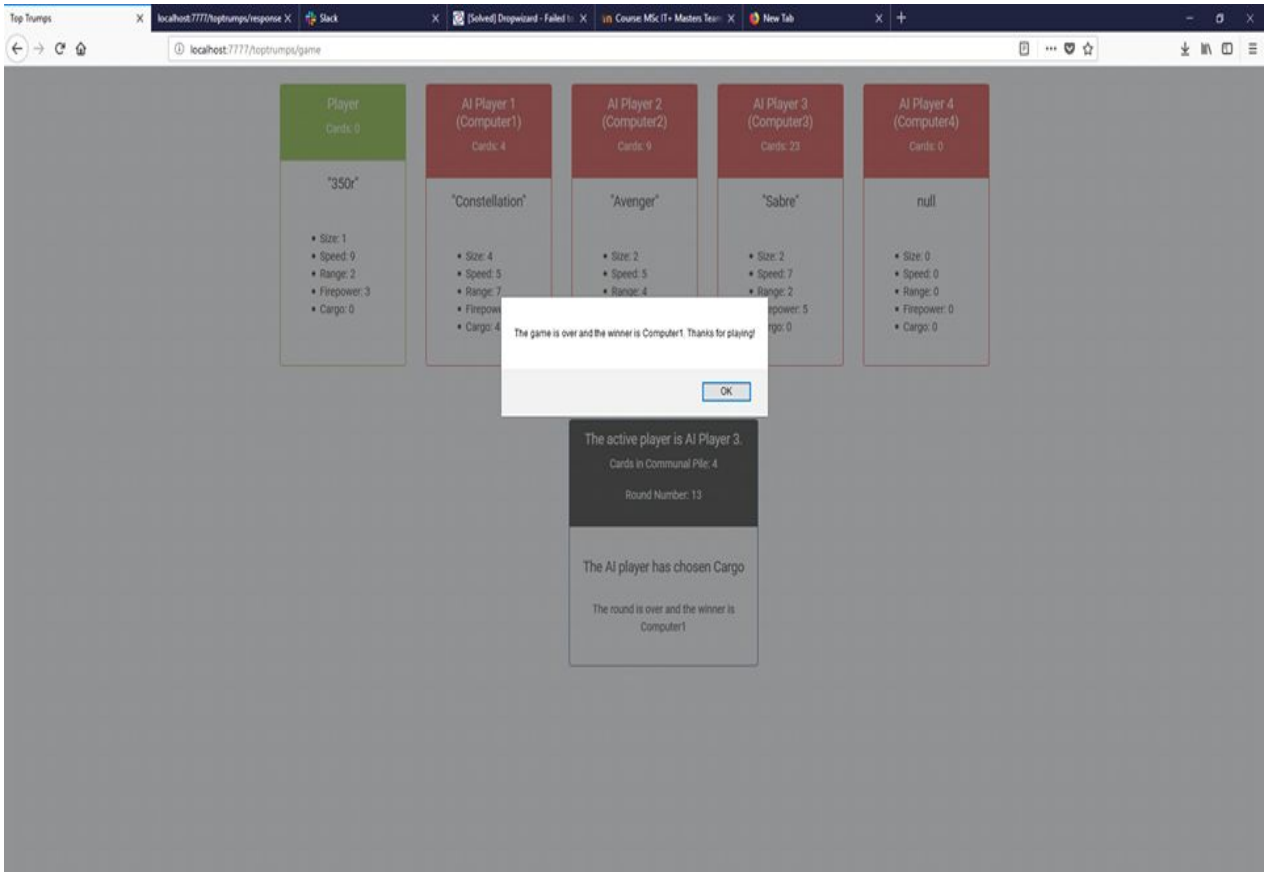
2.1 Online version start page



2.2 Online version stats screen



2.3 Example of a working game of the online version



2.4 Example of online version end game.

Appendix

1.1 Database diagrams

Game

GameID
Number of Rounds
Game Winner
Number of Draws
Player 1 rounds won
Player 2 rounds won
Player 3 rounds won
Player 4 rounds won
Player 5 rounds won

1.2 Database Credentials

Username: “m_18_2413547k”;
Password: “2413547k”
Database name: “jdbc:postgresql://yacata.gla.ac.uk:5432/”;