**COM S/SE 319 : Software Construction and User Interfaces**

**Spring 2019**

**Group No. 11: 2nd Release Report**

**1. Successful Implemented Story Cards for Demo 2:**

* Story Card 8: Update .gitignore file

-Name of the Story: Update .gitignore file

-Assigned Team Member: Stamatios Morellas

-Tasks accomplished for this story card:

i. Add proper reactjs app gitignore

* Story Card 9: Make cells clickable

-Name of the Story: Make cells clickable

-Assigned Team Member: Shivam Vashi

* Story Card 10: Update the CSS layout on the main menu page

-Name of the Story: Update the CSS layout on the main menu page

-Assigned Team Member: Stamatios Morellas

-Tasks accomplished for this story card:

i. Format the buttons

ii. Link the other pages to the buttons

* Story Card 11: Add Changelog
* Name of the Story: Add Changelog
* Assigned Team Member: Stamatios Morellas
* Story Card 12: Finish and populate database

-Name of the Story: Finish and populate database

-Assigned Team Member: Andrew Smith, Minji Park

-Tasks accomplished for this story card:

i. Design Database

ii. Api's connect to database

iii. Populate database with data

* Story Card 13: Highscore implemented

-Name of the Story: Highscore implemented

-Assigned Team Member: Andrew Smith, Minji Park

-Tasks accomplished for this story card:

i. Make High Score api

ii. Make new score checking api to check if the new score is in Top 10

iii. Write tests for High Score api

* Story Card 14: ViewField implemented and stored

-Name of the Story: ViewField implemented and stored

-Assigned Team Member: Andrew Smith

-Tasks accomplished for this story card:

i. Method for View Field implemented

ii. Store in database

iii. Make update view field method

* Story Card 15: Get server code running on server

-Name of the Story: Get server code running on server

-Assigned Team Member: Andrew Smith, Minji Park

-Tasks accomplished for this story card:

i. Npm installed or alternative

ii. Start server code

* Story Card 16: Finish implementing other HTML pages

-Name of the Story: Finish implementing other HTML pages

-Assigned Team Member: Stamatios Morellas

-Tasks accomplished for this story card:

i. Implement leaderboards page

ii. Implement select difficulty page - Waiting

iii. Implement about page

iv. Implement play game page

* Story Card 17: Make CSS on main game have a 90s aesthetic

-Name of the Story: Make CSS on main game have a 90s aesthetic

-Assigned Team Member: Shivam Vashi

* Story Card 18: Implement a counting timer to score the game

-Name of the Story: Implement a counting timer to score the game

-Assigned Team Member: Stamatios Morellas

* Story Card 19: Update Cell and Board Component classes

-Name of the Story: Update Cell and Board Component classes

-Assigned Team Member: Stamatios Morellas

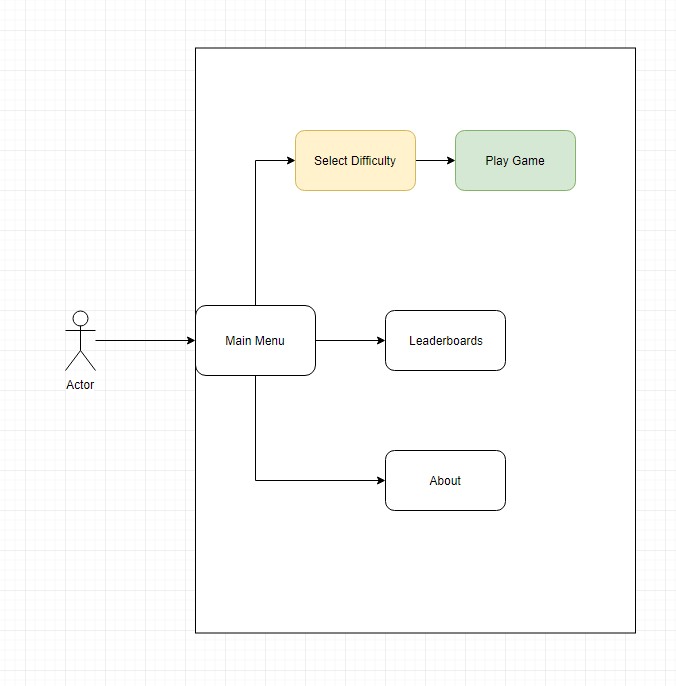
-Tasks accomplished for this story card:

i. Start implement the board class

ii. Implement the cell class

iii. Import the server api to each class

**2. Design Documentation (UML Diagram for Story Cards):**

****

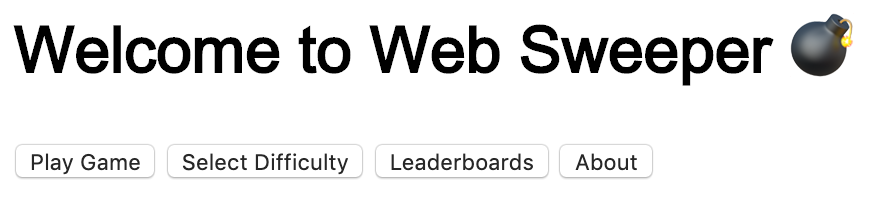
*Figure 1: Use case diagram of what the user can access currently*

**3. UI description with Screenshots:**

These are all Screenshots for successfully implemented story cards for 2st Release:

**Story Card 3: Main Menu Screen**

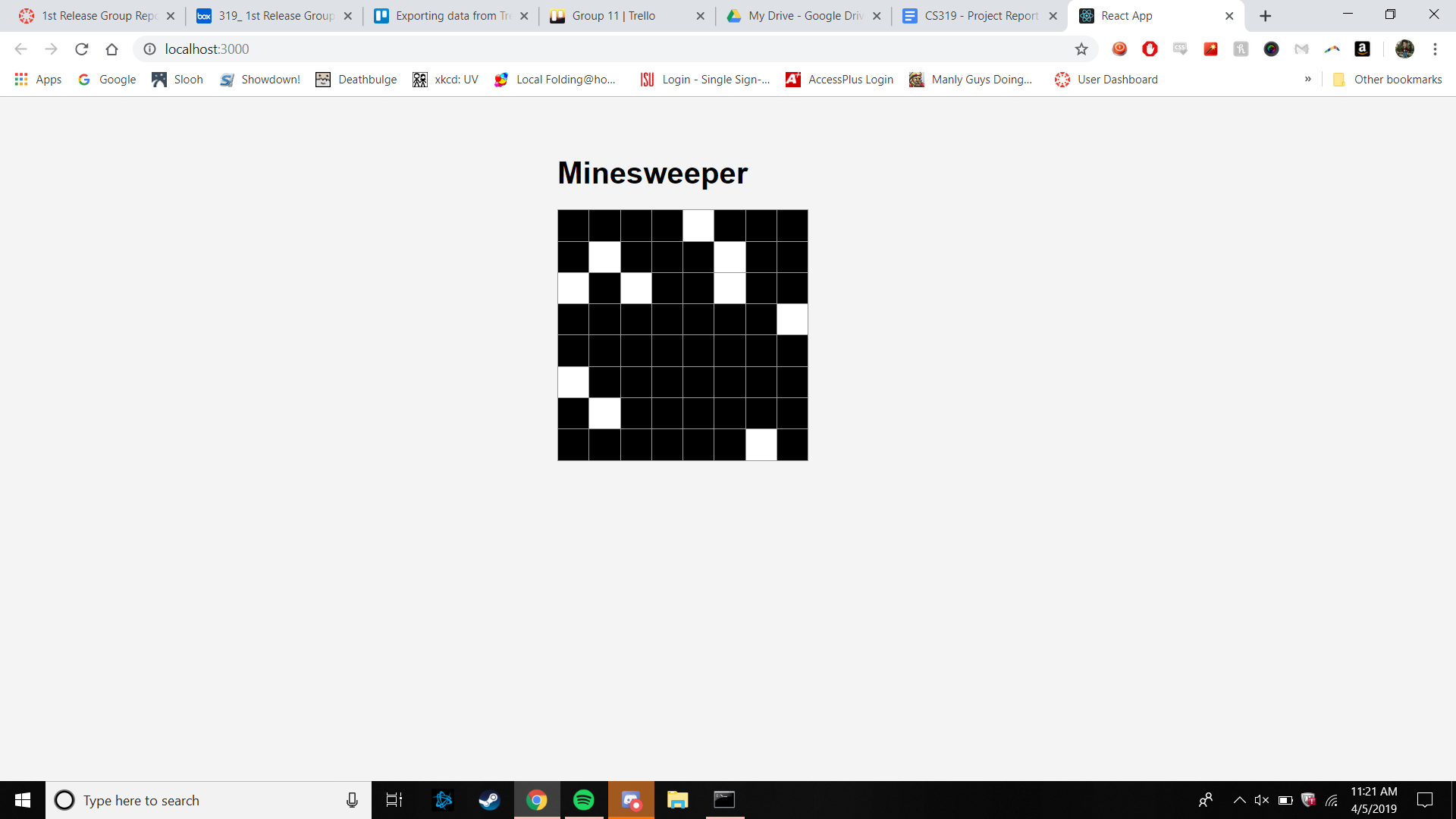
*This is the initial screen that the user will see when the game is started in the web browser. This is simply a mockup, and it has not been styled with CSS yet (this will be done in the near future).*

****

*Figure 1: Screenshot of the first iteration of the main menu layout*

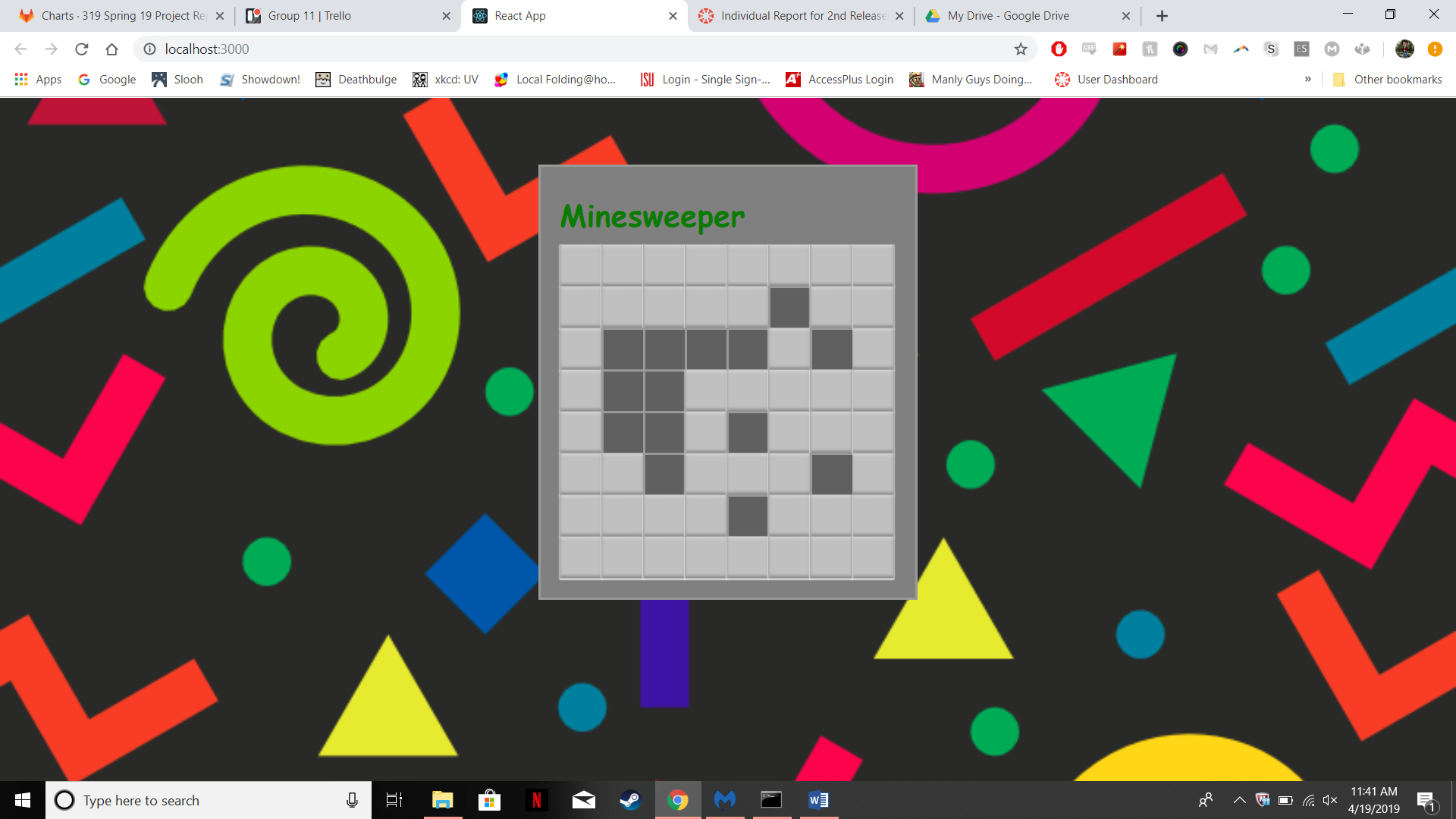
**Story Card 4: Create Grid Layout**

*On startup, the board should render*

****

*Figure 7: Story Card 4: As a user I want to be able to see the grid when I start the game.*

**Story Card 9: Make cells clickable**

**

*Figure 8: When the user clicks a cell, it should be clicked and look that way*

**4. Testing:**

Unit tests for server APIs

* GET new game API: more unit tests are add for each difficulty (easy, medium, and hard), and the unit test checks if the api returns correct response status code (200 OK) and minefield data after the actual database operation is successfully done.
* GET leaderboard API: more unit tests are add for each difficulty (easy, medium, and hard), and the unit test checks if the api returns correct response status code (200 OK) and a data of the top 10 highest scores from the database table.
* POST check score API: unit tests for each difficulty to check if new score is going to be in Top 10 scores. The tests validate if the api returns correct response code (200 OK) and the result that shows whether the score is in Top 10.

**5. Plan for 3nd Release:**

List of selected Story Cards for next iteration i.e., for 3nd Release.

* **Story Card 20:** *Method to add username to leaderboards*
* **Story Card 21:** *Method to update the view field*
* **Story Card 22:** *Add visualization for bombs*
* **Story Card 23:** *Add APIs for the server end code for Front end to call*