**.NET Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *CLC Milestone 6: Final Project and Presentation: Completed Minesweeper Game* | |
| **Date:** | *10/14/2018* | |
| **Revision:** | *1.7* | |
| **Team:** | 1. *Michael Weaver* | |
| 1. *Mark Piland* | |
| 1. *Fredrick Ondieki* | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *Game Board Formatting* | *Michael* | *2* | *?* | | *Game Board Functionality Change* | *Michael* | *2* | *?* | | *Save Feature Ajax Call and Research* | *Michael* | *2* | *0* | | *Documentation* | *Michael* | *1* | *0* | | *Project Reset and Registration / Login Fix* | *Mark* | *4* | *0* | |  |  |  | *0* | |  |  |  | *0* | |  |  |  | *0* | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | *https://github.com/battousairurik/CST-247* | |
| **Peer Review:** | *Y/N* | Yes |

**Planning Documentation**

**Agile Scrum Product Backlog:**

[*https://github.com/battousairurik/CST-247/blob/master/Week%201/Project%20Backlog.xls*](https://github.com/battousairurik/CST-247/blob/master/Week%201/Project%20Backlog.xls)

**Agile Scrum Sprint Backlog:**

[*https://github.com/battousairurik/CST-247/blob/master/Week%201/Sprint%20Product%20Log.xlsx*](https://github.com/battousairurik/CST-247/blob/master/Week%201/Sprint%20Product%20Log.xlsx)

**Agile Scrum Burn Down Chart:**

[*https://github.com/battousairurik/CST-247/blob/master/Week%201/Burndown%20Sheet.xlsx*](https://github.com/battousairurik/CST-247/blob/master/Week%201/Burndown%20Sheet.xlsx)

**Agile Retrospective Results:**

*The following table should be completed after each Retrospective on Things That Went Well (Keep Doing). An alternative to the following table is to use a Mind Mapping tool such as Coggle. If you use a Mind Mapping tool you must include a URL or Image File.*

|  |
| --- |
| **What Went Well** |
| **Once I (Michael) pointed out there is a way to create a project where Visual Studio generates the Login and Registration, that portion went smoothly** |
|  |
|  |

*The following table should be completed after each Retrospective on Things That Didn’t Go Well (Stop Doing) and What Would Be Done Differently Next Time with an Action Plan to Improve (Try Doing and Continuous Improvement). An alternative to the following table is to use a Mind Mapping tool such as Coggle. If you use a Mind Mapping tool you must include a URL or Image File.*

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
| **Save Game Feature incomplete** | **Complete next week** | **N/A** |
| **Load game feature incomplete** | **Complete next week** | **N/A** |
| **Lack of JavaScript understanding to finish project** |  | **N/A** |
| **Controllers cannot pass objects between multiple views** | **Nothing can be done about the nature of the controllers** | **N/A** |

**Design Documentation**

**Install Instructions:**

*Step by step instructions for setting up your database, configuring, and deploying/installing your application. This section should also include detailed instructions for what configuration files are required by your application, what configuration settings need to be adjusted for various runtime (development or production) environments, and where the files need to be deployed to. This section should also contain detailed instructions for how to clone your application source code from BitBucket and deploy the application to an externally hosted site.*

*Retrieve Project from GitHub*

*Follow the included link in this documentation to the BitBucket account.*

*Preferably clone the entire repository.*

*Open the Minesweeper folder. Click on the executable file (Minesweeper.exe – Shortcut)*

*Retrieve Code from GitHub*

*Follow the included link in this documentation to the BitBucket account.*

*Within Repository, open the Minesweeper Folder.*

*Open Minesweeper Project folder.*

*Retrieve necessary source code.*

*Running from the Web Browser.*

*For local deployment extract files to local deployment folder, such as MAMP.*

*Open and run MAMP.*

*Start MAMP server and navigate the directory to find the Login Page.*

*Open the Login page.*

**General Technical Approach:**

*You should, in words, describe your approach and design here. You should also summarize any meeting notes, brainstorming sessions, etc. that you want to retain thru the design of your project.*

Week 1

This Week consisted of uploading the preexisting project to the team repository, followed by a team meeting and generation of user stories for future work. The goal of this week was to update each team member with the goals and expectations of this class and ensure that all future work can be accomplished in a timely manner.

Week 2

This week was spent refactoring the existing minesweeper project to remove redundant classes. After which a login and registration page was built to allow users to register their minesweeper account and play the game. Finally documentation was filled out to track progress on the project.

Week 3 and 4

Discovering that the original approach to the project was incorrect, we began a new project from scratch as a correct ASP.net MVC 5 project. Then created login and registration pages via the use of models and visual studios capability to automatically generate classes. Next we designed views and controllers to handle the gameplay and navigation, albeit hit a few walls preventing 100% game functionality.

Week 5

The group was fairly busy this week so getting group meetings was nigh impossible. Understanding the flow of logic for game navigation proved to be difficult and, in the end, prevented us from making progress. Button events and Ajax requests do not work as intended and calling Script Functions doesn’t seem to work either.

Week 7

We decided to create a new project and have Visual Studio pre-generate the Individual User Authentication scheme, this way users can register and login without issue. After that we transferred all the old files to the new project and did some formatting. Michael placed the “New Game” hyperlink in the Valid Authentication Bar, so only users logged in can access the Game Board to play the game.

**Key Technical Design Decisions:**

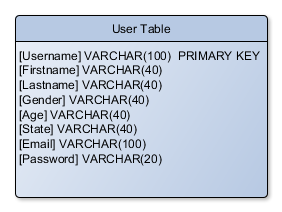
*Any final technical design decisions, such as framework decisions, etc., should be documented here. This should list the technology/framework, its purpose in the design, and why it was chosen.*

*The Login and Registration pages are designed with HTML and C#*

*The core Minesweeper program is written in C# and broken into MVC standard architecture.*

**ER Diagram:**

*Image file of your ER database diagram.*

**

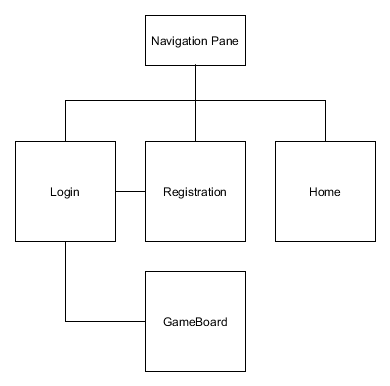
**DDL Scripts:**

*This should contain a link to BitBucket where the DDL script can be downloaded from.*

Coming in a future update

**Sitemap Diagram:**

*Image file of your Sitemap diagram.*

**

**Security Design:**

*This section should outline the design for how authentication and authorization was supported. This section should also contain all of the roles and privileges that are supported by the design.*

Authentication for the login and registration is handled by Required Fields in the Model, as Data Members enable Views to require users to enter valid data. Passwords require a minimum number of characters, at least one number, and one special character.

**Third Part Interface Design:**

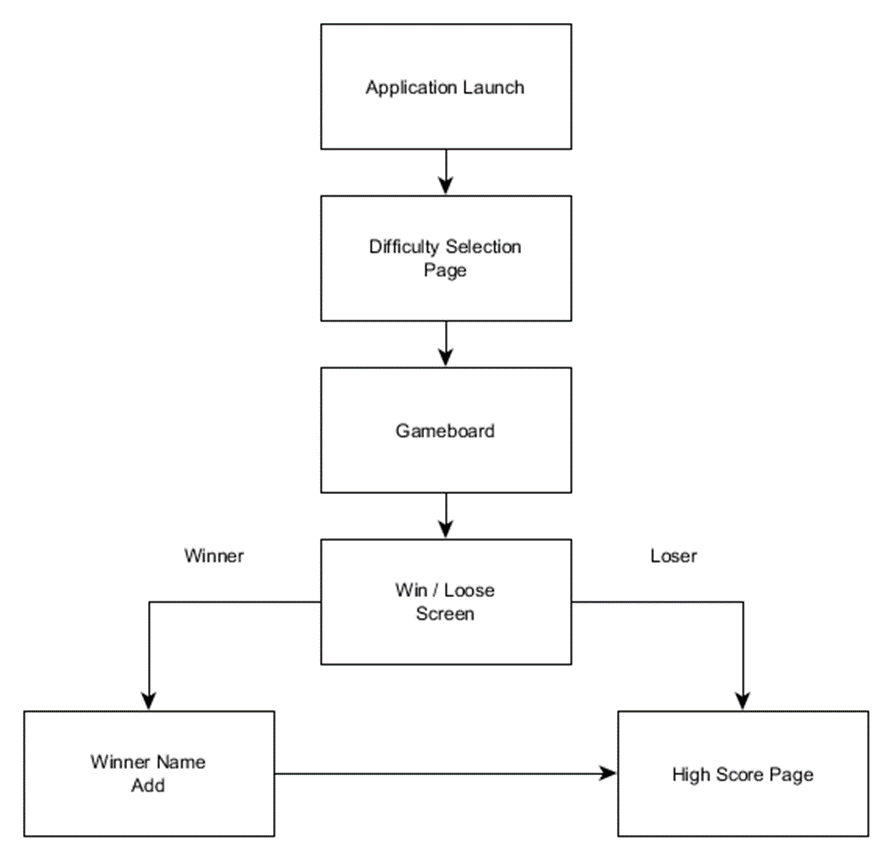
*This section should fully document any Third Party Service Interface API’s, how to access the service, what parameters are required by the API, and the detailed JSON data format specification that could be used by a third party developer to integrate with the service and API.*

Section Currently Not Applicable.

No third party material is currently used in version 1.1.

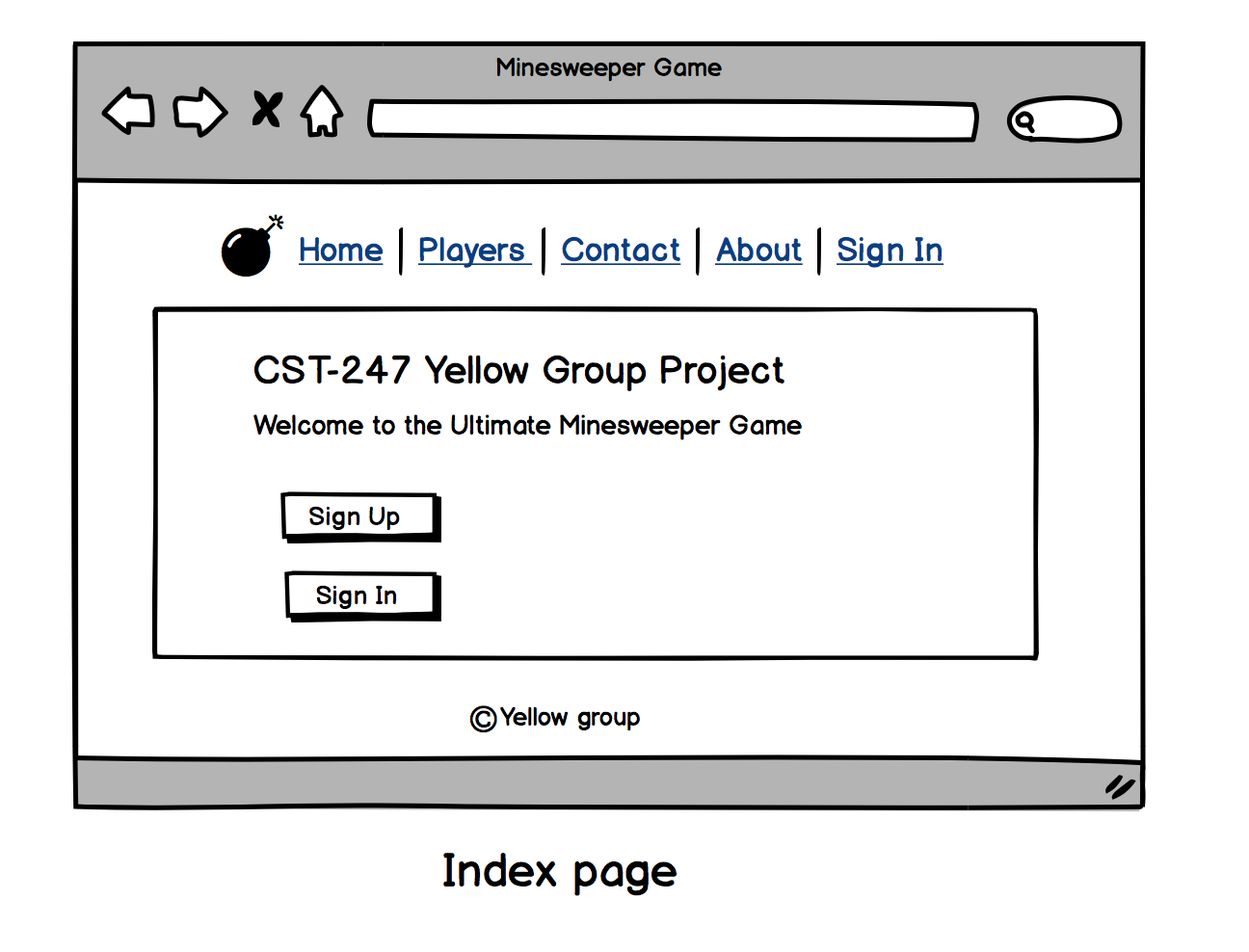
**Flow Charts:**

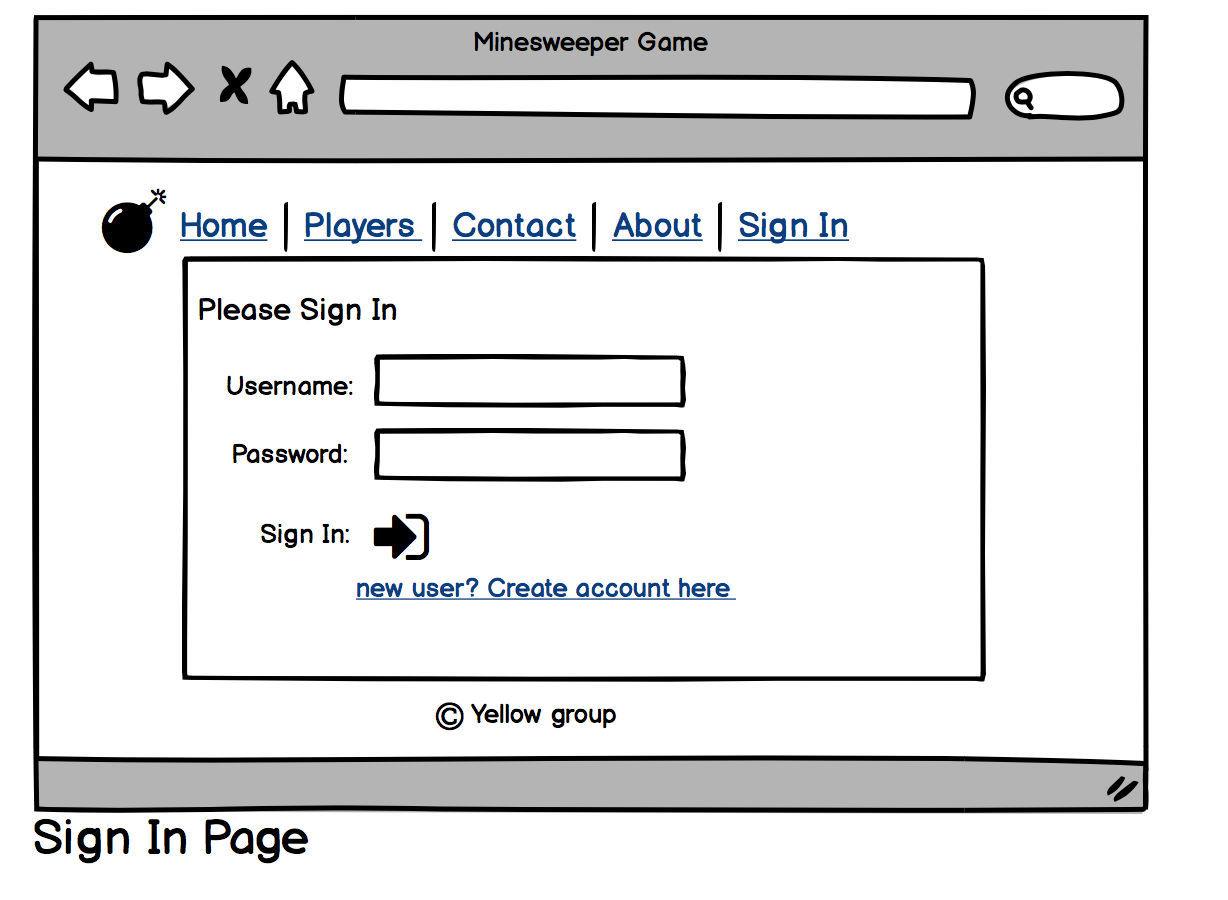
*You should insert any flow charts here. Flow charts should document algorithms or workflow that will be implemented in your program. At a minimum this should contain a flow chart of the Minesweeper game logic.*

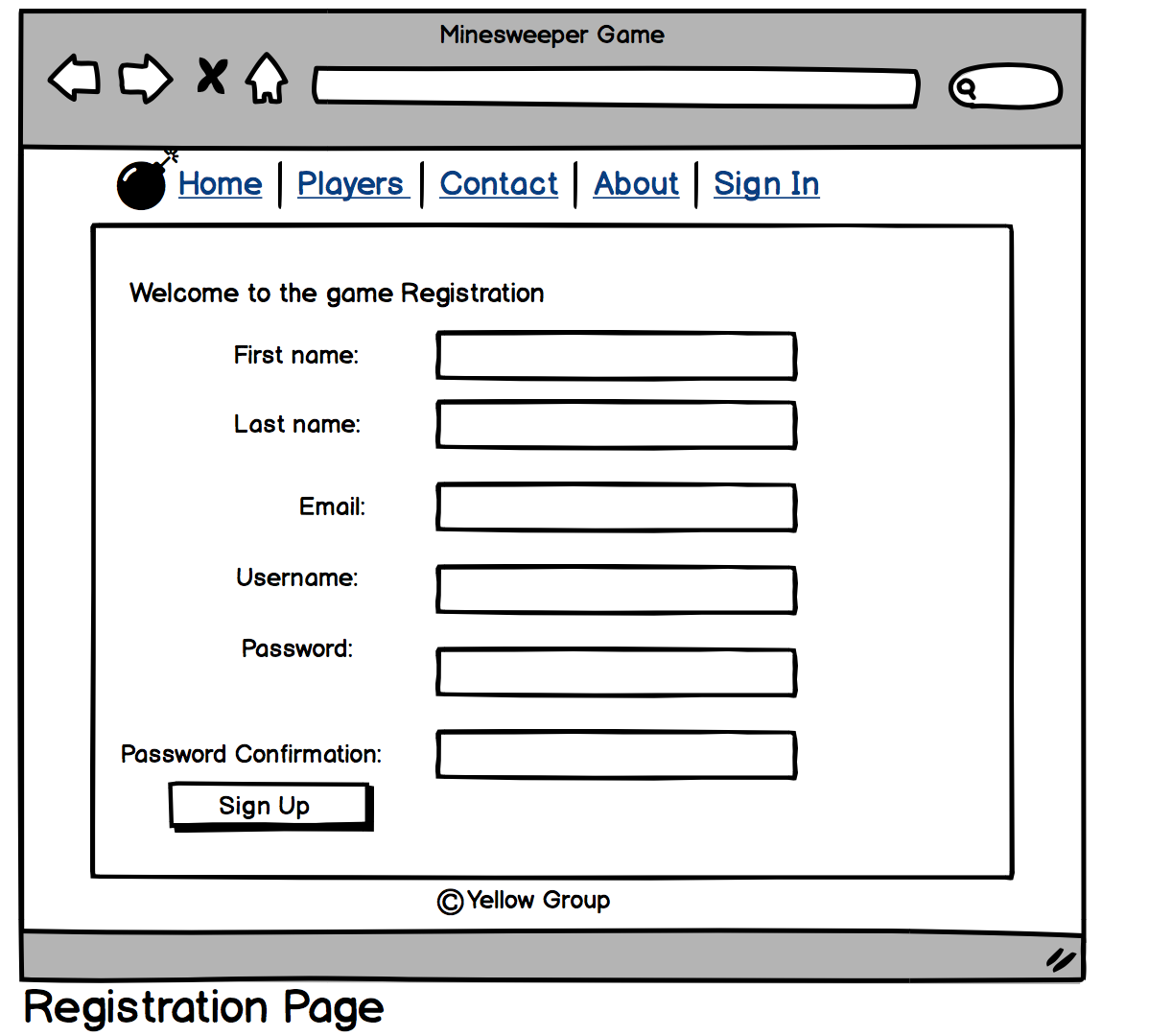


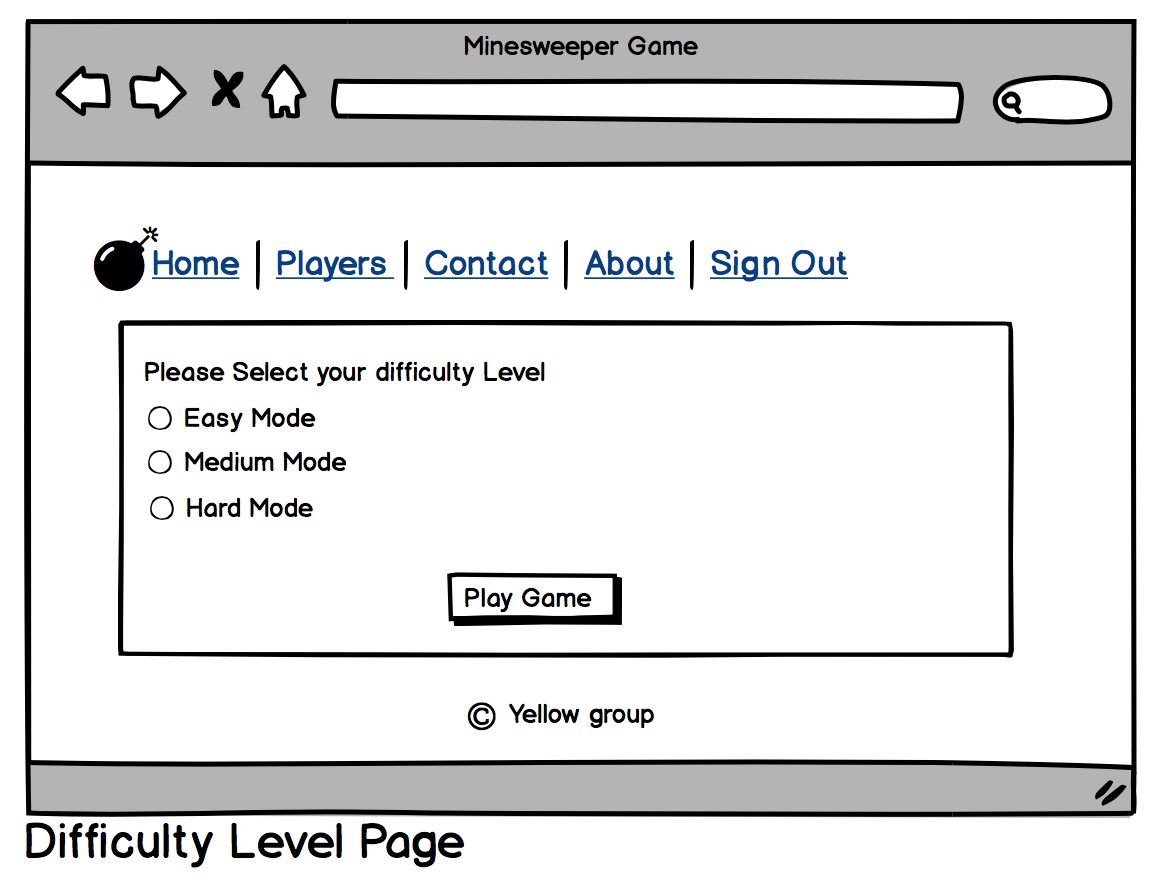
**User Interface Diagrams:**

*You should insert any wireframe drawings or white board concepts that were developed to support your application. If you have no supporting documentation please explain the rational why you are able to leave this section as N/A.*





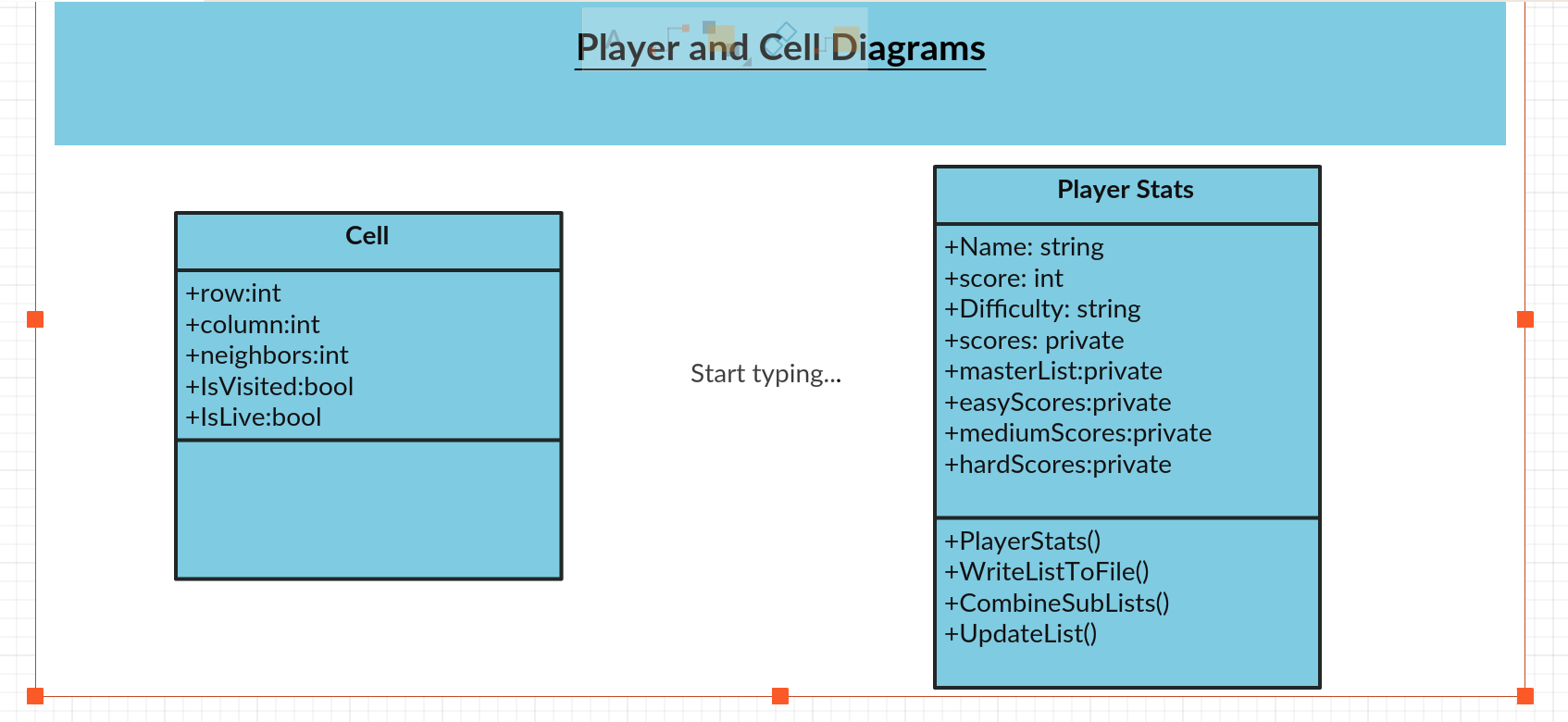




**Class Diagrams:**

*You should insert any class diagrams here. Your class diagrams should be drawn correctly with the three appropriate class compartments, + and – minus to indicate accessibility, and the data types for the state/properties as well as method arguments and return types. If you have no supporting documentation please explain the rational why you are able to leave this section as N/A.*





**Pseudo Code:**

*You should provide BitBucket URL references to any code stubs & pseudo code. If you have no supporting documentation please explain the rational why you are able to leave this section as N/A.*

Section Currently Not Applicable.

Code from previous project has no pseudocode and current portions of the project have not yet been undertaken. Refactoring has not yet begun. Pseudocode will fall into place as the project takes wing.

**Other Documentation:**

*You should insert any additional drawings, storyboards, white board pictures, project schedules, tasks lists, etc. that support your approach, design, and project. If you have no supporting documentation please explain the rational why you are able to leave this section as N/A.*

Section Currently Not Applicable.

Do not currently have any additional documentation as the project is still in its design phase.

**Results of SCRUM Retrospective:**

Week 2

After reflecting on how the first week went the team all agreed that meeting twice a week is in our best interest. The first meeting to handle refactoring issues and work out any additional issues with the project that had arisen. The team also determined that holding weekly meetings to keep each other on the same track is a great idea and will be something handled each week.

Week 5

This week was a busy week that the group did not manage to make enough progress on the project. A lack of understanding and communication has prevented the group from functioning at its peak possibility. For future sessions the group would need to communicate better, discover its weaknesses, and work towards mitigating the time needed to learn the material.

Week 6

This week the group worked towards fixing the registration error and including the Save / Load game features via Ajax and function calls. Participation increased though communication was a bit lacking given how busy each teammate was.

Week 7

The group was fairly disconnected this week, we each had our own tasks to handle. Registration and Login was corrected, though a lack of progress on the game functionality stifled us. I personally discovered that controllers are created from scratch when they are called, meaning they cannot carry any sort of class level variable without having it declared as a singleton.