**.NET Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | CLC Milestone 1: Initial Scrum Planning | |
| **Date:** | 1/16/2021 | |
| **Revision:** | 1.0 | |
| **Team:** | 1. Shawn Fradet | |
| 1. Richard Williamson | |
|  | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | Setup the database so that we can save data. So that we can meet requirements of other stories and tasks | Shawn | 2.0 | 0 | | Create login and registration controllers. So that we can have ability to have functionality to allow for FR-2 User story. | Shawn | 1.0 | 0 | | Create models for user and registration. So that when a user registers their user data is stored in the database and when logging in it will validate against the database. | Shawn | 1.0 | 0 | | Create views that are consumed by controllers. So that our users can interact with the site to create users and login. | Richard | 2.0 | 0 | | Ensure controllers and views validate data. So that we can ensure data integrity. | Richard | 0.25 | 0 | | Update design documentation. So that we can relay how we are meeting the plan. | Richard and Shawn | 1.0 | 0 | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | <https://github.com/darthxvaderxd/CST-247-Project> | |
| **Peer Review:** | *Y* | We acknowledge that our team has reviewed this Report and we agree to the approach we are all taking. |

**Planning Documentation**

**Agile Scrum Product Backlog:**

<https://github.com/darthxvaderxd/CST-247-Project/blob/master/agile_artifacts/SprintProductLog.xlsx>

**Agile Scrum Sprint Backlog:**

<https://github.com/darthxvaderxd/CST-247-Project/blob/master/agile_artifacts/SprintBackLog.xls>

**Agile Scrum Burn Down Chart:**

<https://github.com/darthxvaderxd/CST-247-Project/blob/master/agile_artifacts/SprintBurnDown.xlsx>

**Agile Retrospective Results:**

|  |
| --- |
| **What Went Well** |
| Communication. Both members of the team were responsive and open to communicating about the path forward for the project. |
| Preparation. Both members had access to their Minesweeper code from the previous course. |
| Division of work. Members were receptive to splitting up work fairly. Work was divided equally for Sprint 1. |

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
|  |  |  |
|  |  |  |
|  |  |  |

**Design Documentation**

**Install Instructions:**

Milestone 1 covered the initial Scrum planning for getting the project path laid out. No code has been created, therefore there are no install instructions at this time.

**General Technical Approach:**

The approach for this group project is to reuse Minesweeper code from a previous class and adapt it to a web application. We will be following the guidelines provided by the class to incorporate upgrades to the initial code. These updates are to include:

* A secure site
* User Registration/Login
* Saving game state and stats
* Displaying game stats and results with HTML
* Retrieving game stats and results with a Web Service Interface

During our Milestone 1 meeting we laid out the user stories for the different milestones and divided work for the first Sprint. The first Sprint will cover creating code for an initial user registration and login screen.

**Key Technical Design Decisions:**

This project will be completed using ASP.NET MVC as the basis for the web application. ASP.NET MVC provides a framework that will let us build Controllers to work with data between a View and a Model. The View will use Razor pages and AJAX for the dynamic and responsive content needed for a game board. The Controllers and Models will be coded with C#.

For data storage this project will use SQL Server. ASP.NET MVC will communicate with the SQL Server through ADO.NET connections.

In later milestones a Web API will be introduced to retrieve game stats and results using REST.

**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.*

**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.*

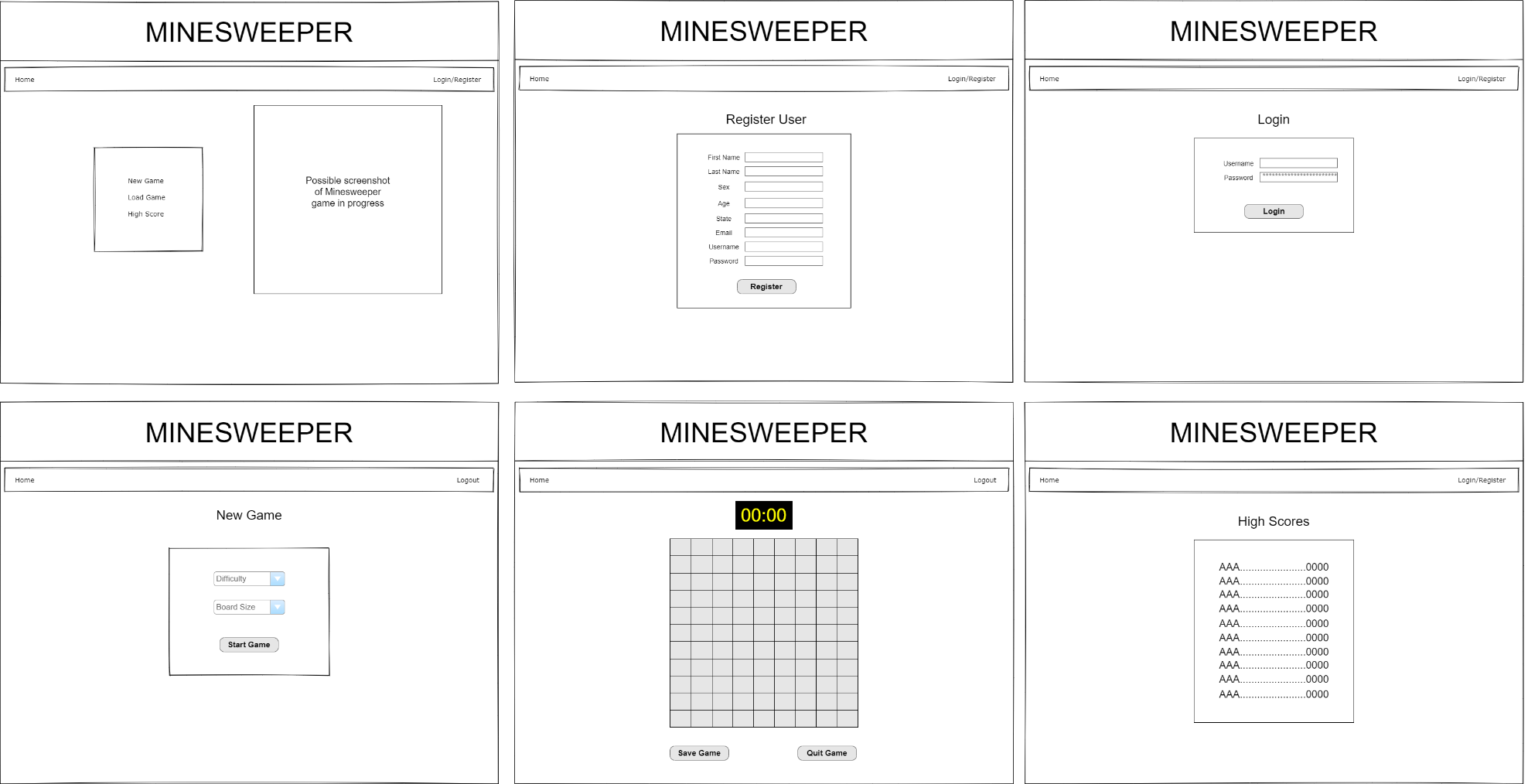
**Third Part Interface Design:**

*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.*

**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:**



**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.*

**Other Documentation:**

Team will be meeting daily on a Discord channel to update progress for Scrum.