**.NET Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | CLC Milestone 2: Login and Registration Modules | |
| **Date:** | 1/16/2021 | |
| **Revision:** | 2.0 | |
| **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 between the team went well |
| Breakdown of the work went well |
|  |

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
| **Some of the ui still need styling** | **As we go forward continue adding to it** | **2/17/21** |
|  |  |  |
|  |  |  |

**Design Documentation**

**Install Instructions:**

This project is being designed with ASP.NET MVC. The required database can be configured with the provided DDL script. To run the application, you must load the project with Visual Studio. Once the project it loaded it can be started with F5. Currently the project only consists of a Login page and Registration page. These pages can be accessed with:

localhost:[port]/Registration

localhost:[port]/Login

**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

Milestone 1: 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.

Milestone 2: For milestone 2 we implemented registration and login pages. Each page uses a separate controller and model for handling the handling of routing and data. Both the Registration and Login pages send their information to their controllers with HTTP POST for security. The form data from both pages is validated in the controller with the use of validation rules on the data models. Services were added for handling all business logic concerned with storing registration data and validating credentials for login.

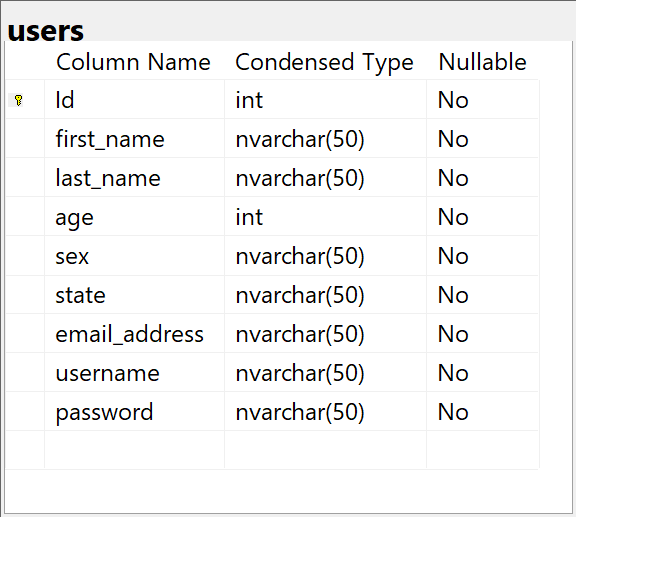
**Key Technical Design Decisions:**

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.

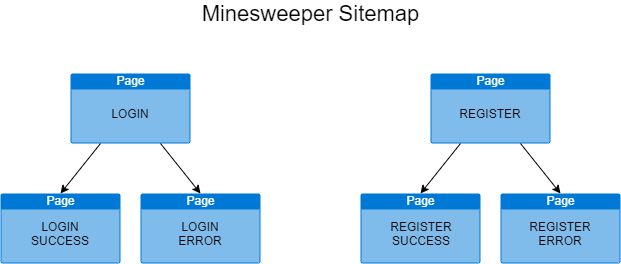
**ER Diagram:**

**

**DDL Scripts:**

[*https://github.com/darthxvaderxd/CST-247-Project/blob/master/Planning%20and%20Design/DDL%20File/*](https://github.com/darthxvaderxd/CST-247-Project/blob/master/Planning%20and%20Design/DDL%20File/)

**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.*

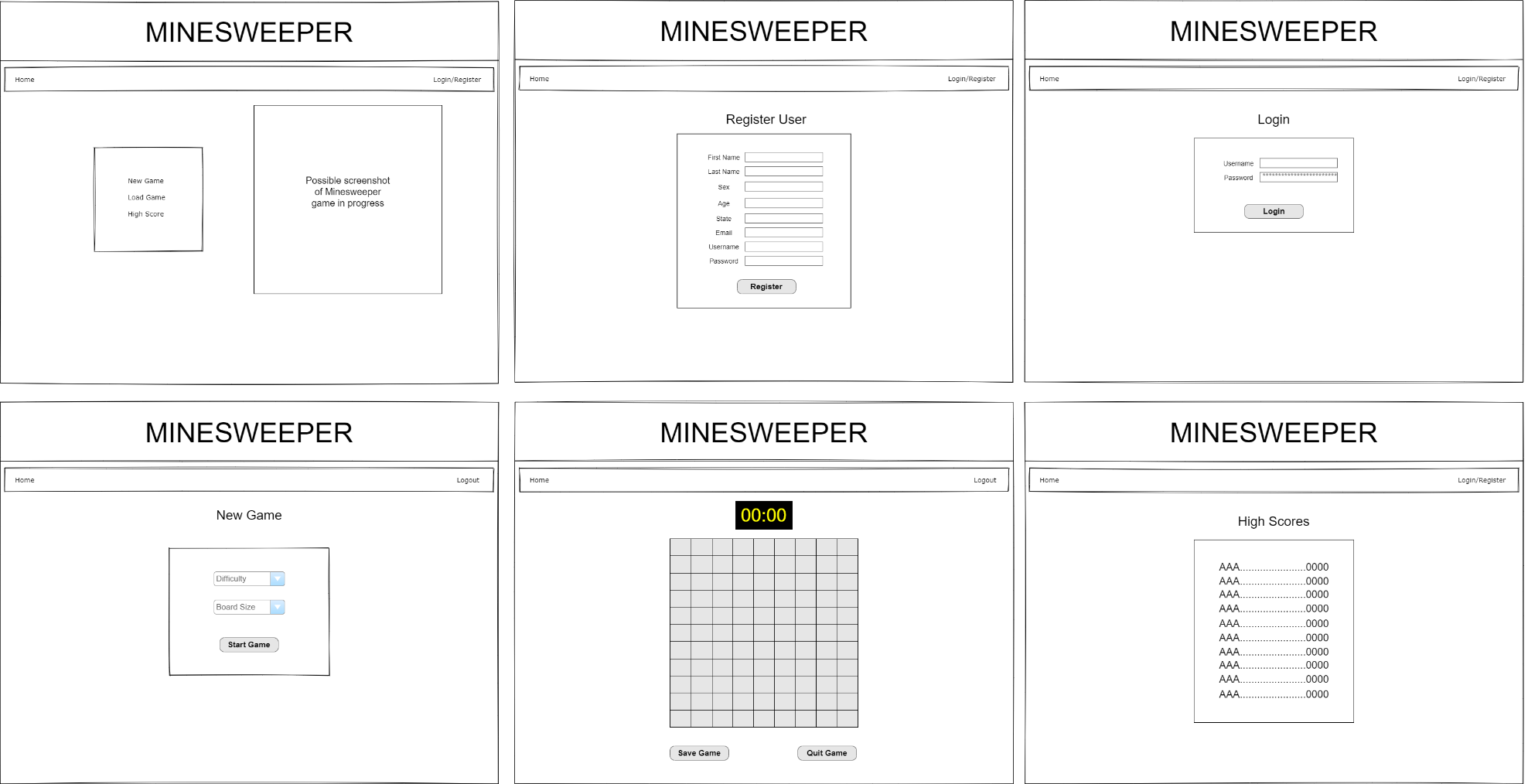
**Third Part Interface Design:**

*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.*

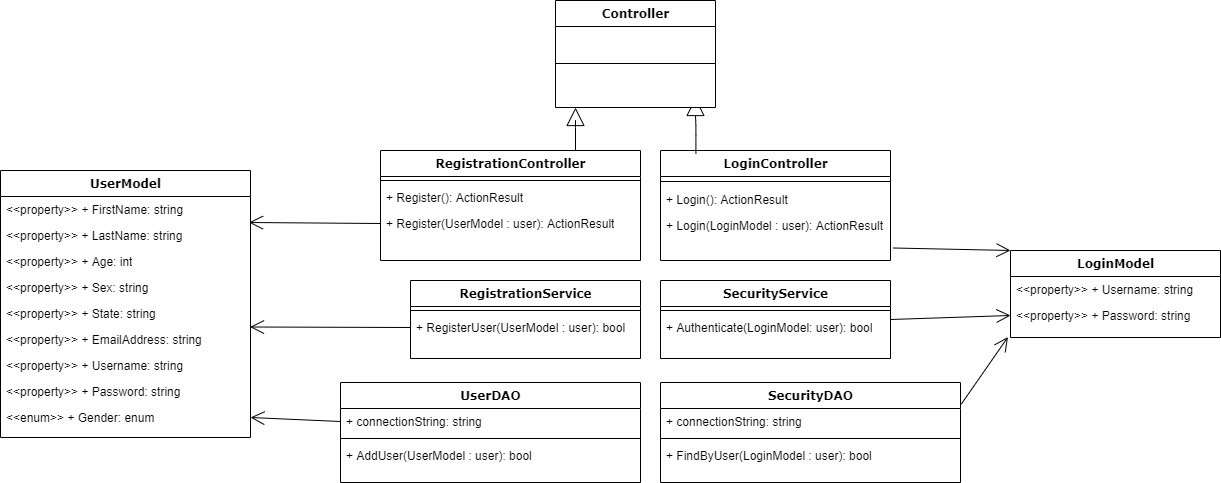
**Flow Charts:**

*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:**



**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.