**.NET Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *FR4.5 FR6* | |
| **Date:** | *10-24-2020* | |
| **Revision:** | *5.0* | |
| **Team:** | 1. *Matt Sievers* | |
| 1. Crystal Sciarrino | |
|  | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | As a developer I need an endpoint I can call from the controller on an onclick to save the current Board to a DB | *Matt Sievers* | *6* | *0* | | As a developer I need an endpoint so I can load a game in progress to continue playing | *Matt Sievers* | *4* | *0* | | As a developer, I need to recreate the project to allow for WebAPI to be implemented for a new GameServiceController | *Matt Sievers* | *3* | *0* | | As a developer I need a new/load game view in between login and difficulty and have all the views updated based on workflow | *Matt Sievers* | *1* | *0* | | As a developer, I need a gameTimer to increment each second to show how long the game took before win/loss | *Crystal Sciarrino* | *1* | *0* | | As a team, we need to do weekly scrum paperwork | *Matt Sievers / Crystal Sciarrino* | *2* | *0* | | As a developer, I need to add a click counter to count the number of cells clicked on before game win/loss is met | *Crystal Sciarrino* | *4* |  | | As a developer, I need to update the views for Game and \_Gameboard to have gameboard load a partial view using Ajax | *Matt Sievers* | *1* |  | | BUG-As a developer, I need to resolve the bug with partial view for gameboard not showing right click flags until left click is pushed | *Crystal Sciarrino* |  | *3* | | BUG-As a developer, I need to resolved the bug about game win/loss partial view now showing up if gameOver == true | *Crystal Sciarrino* | *3* |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | https://github.com/Malleas/CLCMinesweeperApp2 | |
| **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 Board:**

[**https://www.meistertask.com/app/project/sSO4jU4u/cst-247**](https://www.meistertask.com/app/project/sSO4jU4u/cst-247)

**Agile Scrum Product Backlog:**

https://github.com/Malleas/CLCMinesweeperApp/tree/master/CLCMinesweeperApp/Planning%20and%20Design/Week%206

**Agile Scrum Sprint Backlog:**

*https://github.com/Malleas/CLCMinesweeperApp/tree/master/CLCMinesweeperApp/Planning%20and%20Design/Week%206*

**Agile Scrum Burn Down Chart:**

*https://github.com/Malleas/CLCMinesweeperApp/tree/master/CLCMinesweeperApp/Planning%20and%20Design/Week%206*

**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** |
| **Able to get the remaining gameboard logic implemented.** |
| **Great working sessions as we did peer coding this week as well. Working on the partial view as a team sharing code and screens. This was done this week as well as we continue to work on the project as a team. Most of the Stats work and some bug fixes were worked on in tandem.** |
| **Fixed the page refresh for the gameboard** |

*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** |
| Broke the right click, only shows up after a left click | Bug logged, will continue to work on it | **10/31** |
| **Again, assignment and CLC were not mirrored. Found that you can build an app with webAPI turned on for the REST services.** | **None, project has been rebuilt. Please note, new repo address for App2.** |  |
| **Simple things again took the most time. Deserializing the json data or hacking a hidden input to capture data to send to an on click.** | **Better, more concise assignments that teach skills needed to complete the CLC.** |  |

**Design Documentation**

**Install Instructions:**

*Connect and pull from master from repo. Refer to DDL scripts before running as a local DB will need to be created and table added. Click debug menu option and select Run without debug. Backup provided of DB for full install as needed.*

*Please note that two additional tables have been added since previous build so refer to the DDL scipts to build those tables in a local database.*

**General Technical Approach:**

*Instead of creating a gameserviveDAO, rebuilt the project with webAPI turned on for MVC projects. This made the implementation of the services much easier. Created new DB’s to store the new stats and games. Added a simple button counter and timer, although since these were in JS, they proved to be harder to implement as this is a C# class and our knowledge on JS is limited.*

**Key Technical Design Decisions:**

*Nothing ground breaking for this, as stated the project was rebuilt using WebAPI for MVC to allow for easy building of services. We also chose to abstract the different levels of data to their own tables to not create a large flat file.*

**ER Diagram:**

*n/a for this requirement*

**DDL Scripts:**

*CREATE TABLE [dbo].[Player]*

*(*

*[USERID] INT NOT NULL PRIMARY KEY IDENTITY(1,1),*

*[FIRSTNAME] VARCHAR(50) NOT NULL,*

*[LASTNAME] VARCHAR(50) NOT NULL,*

*[GENDER] VARCHAR(50) NOT NULL,*

*[AGE] INT NOT NULL,*

*[STATE] VARCHAR(2) NOT NULL,*

*[EMAILADDRESS] VARCHAR(100) NOT NULL,*

*[USERNAME] VARCHAR(50) NOT NULL,*

*[PASSWORD] VARCHAR(50) NOT NULL*

*)*

*CREATE TABLE [dbo].[Stats] (*

*[Id]     INT IDENTITY (1, 1) NOT NULL,*

*[Time]   INT NOT NULL,*

*[Clicks] INT NOT NULL*

*);*

*CREATE TABLE [dbo].[Game] (*

*[Id]        INT           IDENTITY (1, 1) NOT NULL,*

*[Gameboard] VARCHAR (MAX) NOT NULL*

*);*

**Sitemap Diagram:**

*N/A not required for this FR*

**Security Design:**

*Added hashing to the password entry at new user creation so we’re not storing actual passwords into the db.*

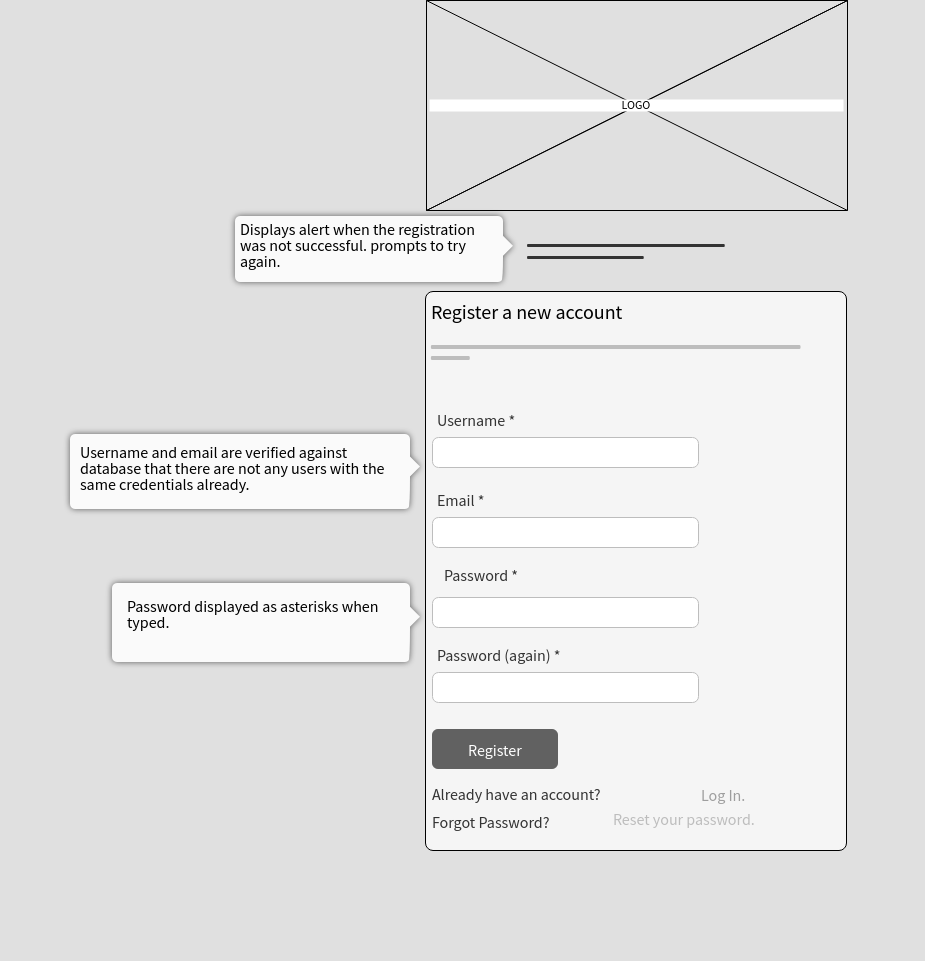
**Third Part Interface Design:**

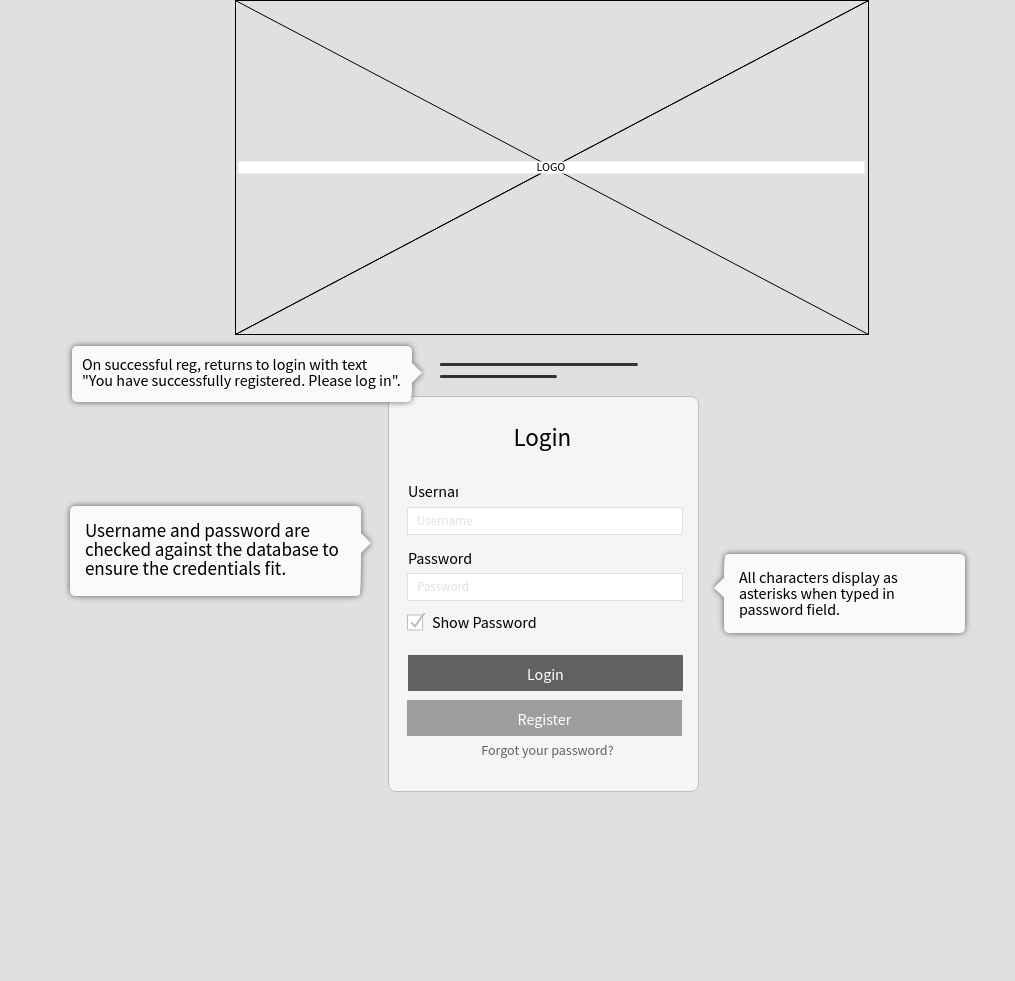
*NA*

**Flow Charts:**

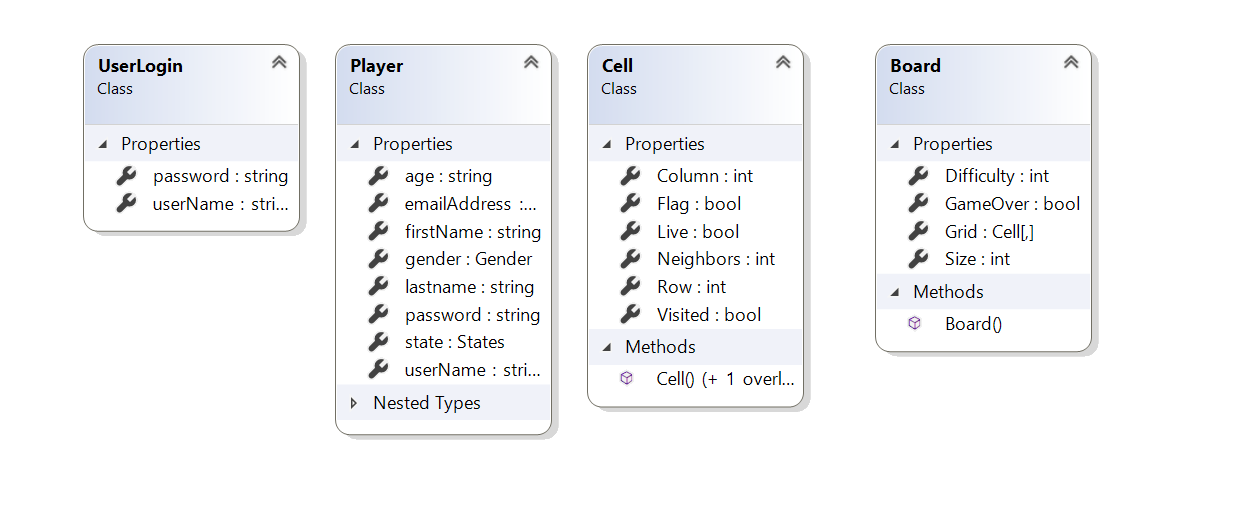
*NA*

**User Interface Diagrams:**





**Class Diagrams:**



**Pseudo Code:**

*N/A*

**Other Documentation:**

*N/A*