**.NET Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Milestone 6: Final | |
| **Date:** | 04/22/2018 | |
| **Revision:** | 1.6 | |
| **Team:** | *1.* Kaleb Eberhart | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **Team**  **Member** | **Tasks Performed** | **Hours**  **Worked** | **Hours Remaining** | | *Kaleb Eberhart* | Code revision, api revision, api documentation | *23* | *0* |   Detailed information about project approach is available in SCRUM document uploaded in bitbucket. | |
| **GIT URL:** | <https://github.com/kalebbe/CS3/Final_milestone> | |
| **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/kalebbe/CS3/Final\_Milestone/tree/master/Planning\_and\_Design/Final\_Sprint\_Back\_Log.xls*](https://github.com/kalebbe/CS3/Final_Milestone/tree/master/Planning_and_Design/Final_Sprint_Back_Log.xls)

**Agile Scrum Sprint Backlog:**

*https://github.com/kalebbe/CS3/Final\_milestone/tree/master/Planning\_and\_Design/Final\_Sprint\_Back\_Log.xls*

**Agile Scrum Burn Down Chart:**

*https://github.com/kalebbe/CS3/Final\_milestone/tree/master/Planning\_and\_Design/Final\_Sprint\_Burn\_Down.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**  I had a lot of fixes to make with this milestone and I’m working solo on every project this semester, so I’m just happy that everything is completed. Nothing really went well with this milestone because there were 3 other projects to complete at the same time. |
| **Fulfilled requirements**  All requirements were fulfilled. |

*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** |
| Bugs | There are bugs left in my project that I would like to get rid of, but ran out of time. All the bugs I have found were all with extra features though, so it wasn’t a priority. | n/a |
| **Time Management** | This was pretty much explained in what went “well”. This final push felt like I was working on a new 10 hour project every day between everything due. | **n/a** |

**Design Documentation**

**Install Instructions:**

1. Download Visual Studios 2015+ (not going to go into detail with this).
2. Pull the project code from the BitBucket repository and save it locally.
3. Import project into Visual Studios (create project from existing sources).
4. Download ddl from BitBucket and import the database to your project.
5. Run the project through Visual Studios.
6. Register a user.
7. Login your user.
8. Play the game.
9. Repeat.

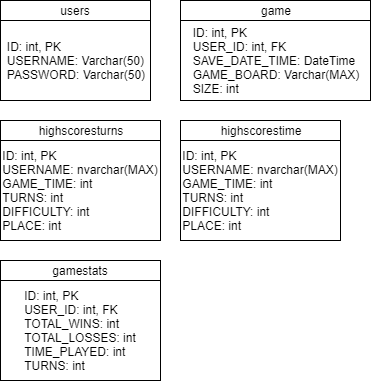
**General Technical Approach:**

MVC layered architecture with partial page views for consistent navbar and headers.

**Key Technical Design Decisions:**

We will be using Bootstrap Studios to create the header/footer for each page and the registration, login, and index pages. We will be integrating this in Razor to work with .NET. As far as our framework, we will be using .NET MVC for this project.

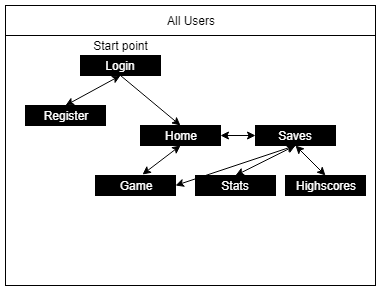
**ER Diagram:**



**DDL Scripts:**

*https://bitbucket.org/kaleb\_be/csharp3/src*

**Sitemap Diagram:**

**

**Security Design:**

Keeping in line with CST-236, we will be using C# and javascript to check the user’s information when they register and every time they log in. Passwords must be at least 8 characters, contain one number, and one letter. Emails (if used) will need to be unique and real email addresses. Also, users will not be able to access the game without being logged in (subject to change).

Milestone 2 additions:

-hashing passwords to prevent data leak.

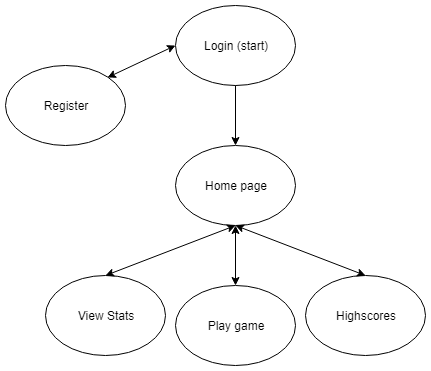
-razer form processing to prevent injection attacks.

-Emails are not used and usernames are used for login.

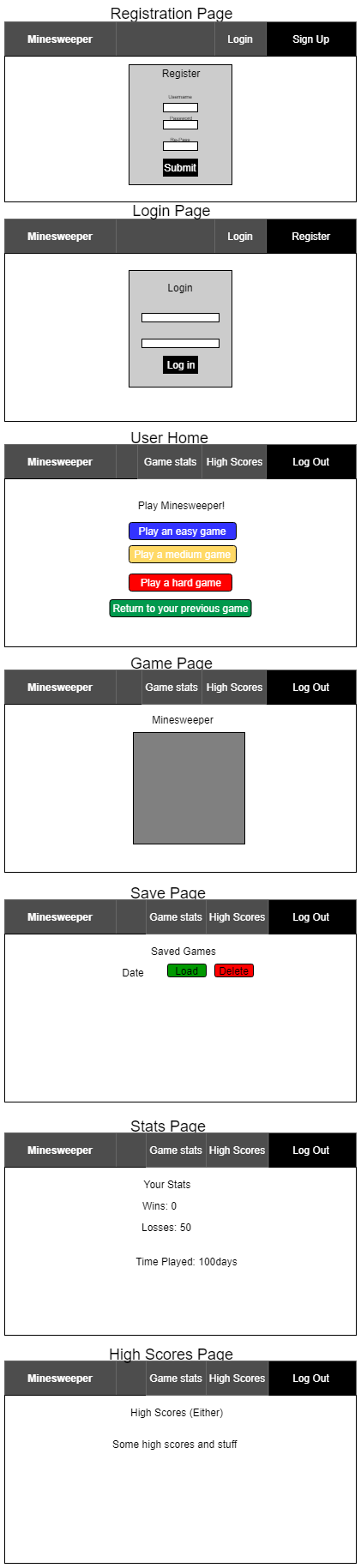
**Third Party Interface Design:**

Not sure if Bootstrap Studios counts for this, but that’s what we used to create our views.

**Flow Charts:**

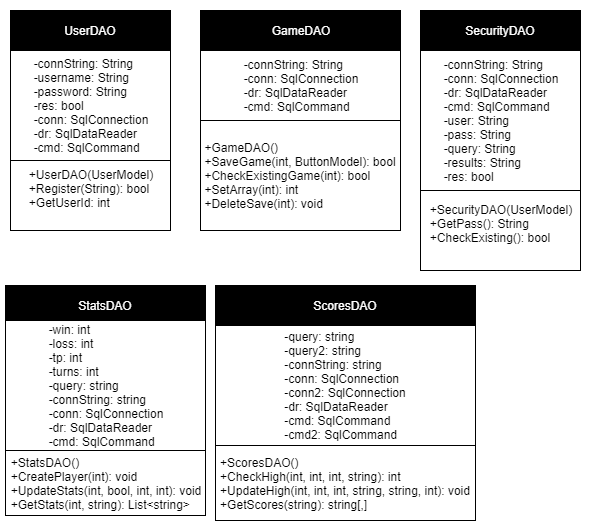
**

**User Interface Diagrams:**

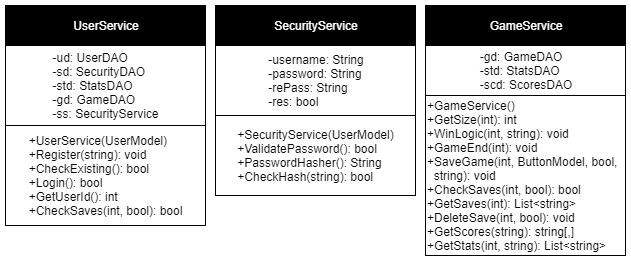
**

**Class Diagrams:**

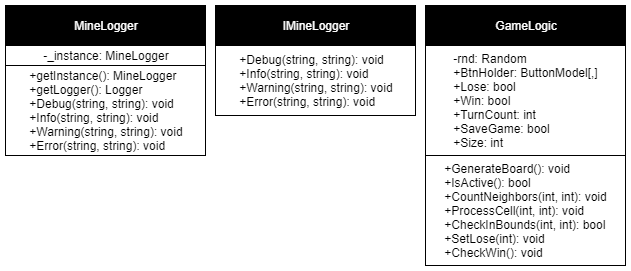
**A. Data**

****

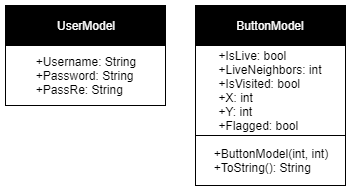
**B. Business**

****

**C. Utilities**

****

**D. Models**

****

**API Documentation:**

Done through swagger:

<https://app.swaggerhub.com/apis/Kaleb_be/Minesweeper-new/1.0.0-oas3>

**Loom video:**

Kaleb: <https://www.useloom.com/share/dcab775f77714008aedde672b274422c>