**.NET Application Programming**

**Project Status and Design Report**

**http://scrumblr.ca/CLC%20Milestone**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *CST 247 CLC Project* | |
| **Date:** | *10/25/2020* | |
| **Revision:** | *7.0* | |
| **Team:** | 1. *Zack Chambers* | |
| 1. Wyatt Surratt | |
|  | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *Please see below and attached documents under “Planning and Design”folder.* |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | <https://github.com/zchambers3/CST247> | |
| **Peer Review:** | *Y/N* | 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:**

*This needs to contain a URL to BitBucket Scrum Product Backlog Artifact.*

In “Planning and Design” folder.

**Agile Scrum Sprint Backlog:**

*This needs to contain a URL to BitBucket Scrum Sprint Backlog Artifact. This current week’s progress should be reflected in the above section of this Design Report.*

In “Planning and Design” folder.

**Agile Scrum Burn Down Chart:**

*This needs to contain a URL to BitBucket Scrum Burn Down Chart Artifact.*

In “Planning and Design” folder.

**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** |
| Communication, planning and testing. |
|  |
|  |

*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** |
|  |  |  |
|  |  |  |
|  |  |  |

**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.*

Note: Removed “bin” and “packages” in order to be able to upload zip file to GitHub. We hope this does not cause any issues.

1. Download “CST247\_MineSweeperCLC-master” zip from <https://github.com/zchambers3/CST247>.
2. Create a new localdb “MinesweeperCLCProject”.
   1. Under “Tables” add the follow new tables. You can find details of what to add below under “Other Documentation”.
      1. Users
      2. Grids
      3. Cells
      4. Stats
3. Run “MinesweeperProjectCLC247” through “IIS Express (Google Chrome)”.
4. Register yourself.
5. Select a difficulty.
6. Login.
7. Play Minesweeper.

**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.*

We discussed what we could use from our previous minesweeper build. We have both been keeping track of our work and communicating through LoudCloud.

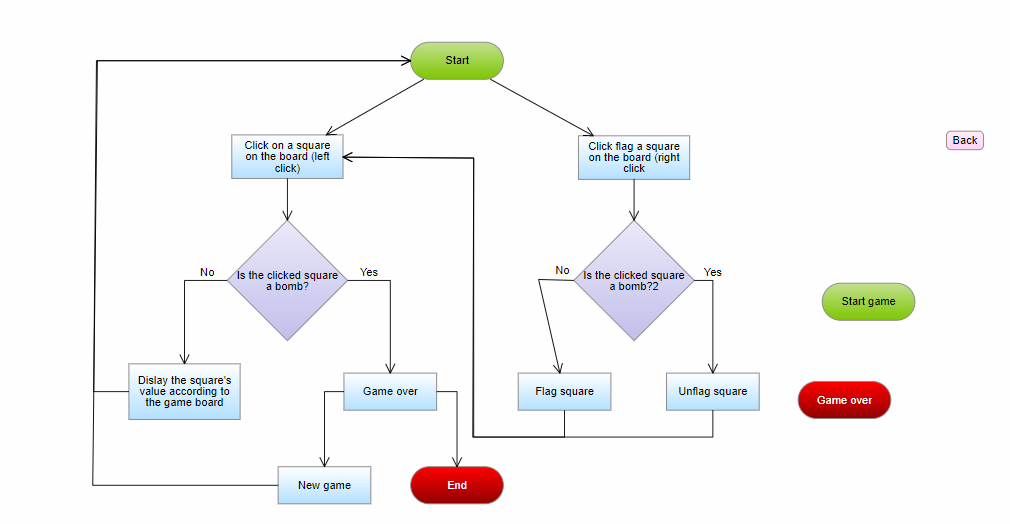
**Key Technical Design Decisions:**

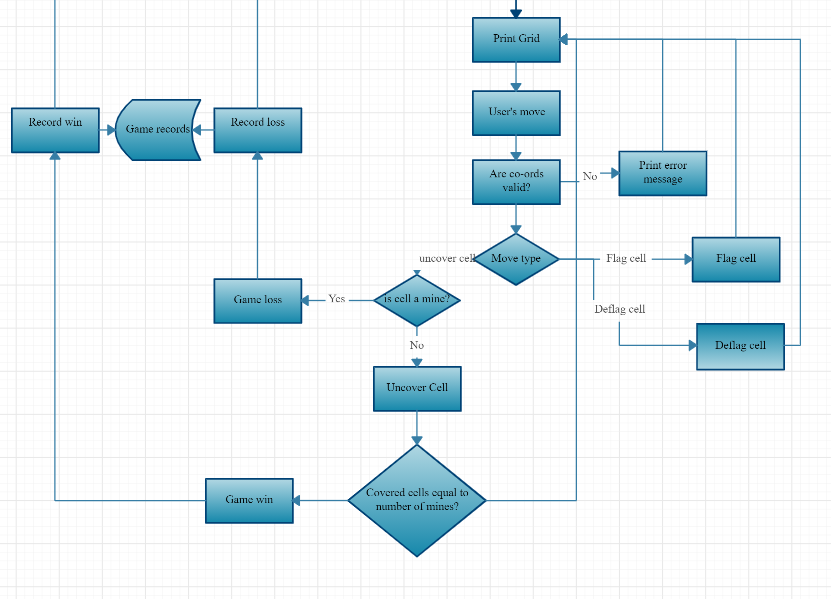
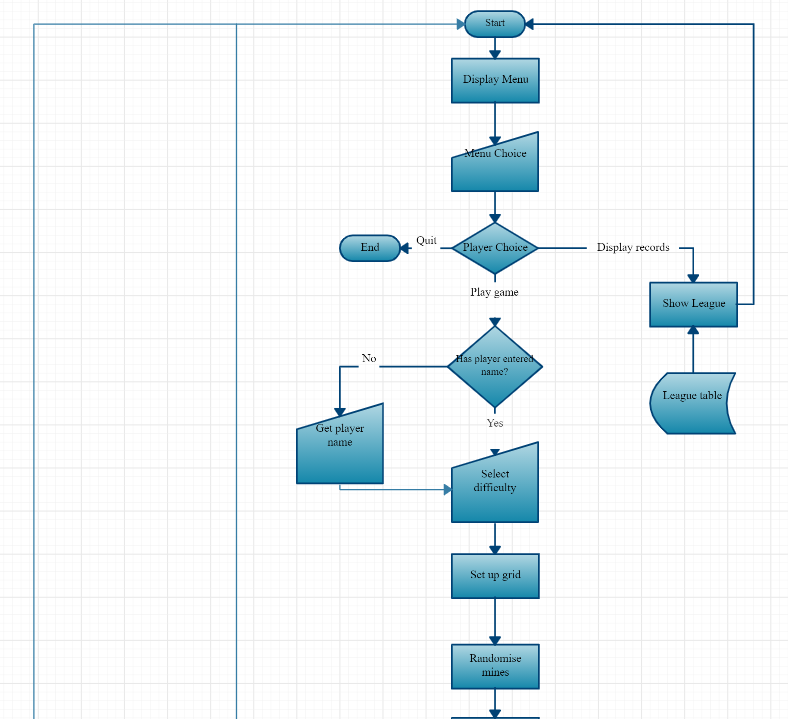
*Any final technical design decisions, such as framework decisions, etc., should be documented here. This should list the technology/framework, its purpose in the design, and why it was chosen.*

Basic design based on the deliverables from instructions provided.

**ER Diagram:**

*Image file of your ER database diagram.*





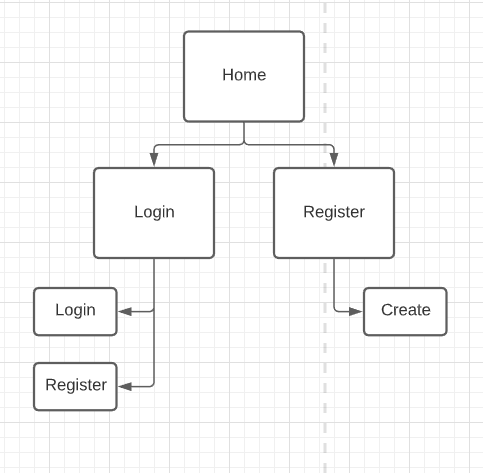
**DDL Scripts:**

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

<https://github.com/zchambers3/CST247>

**Sitemap Diagram:**

*Image file of your Sitemap diagram.*



**Security Design:**

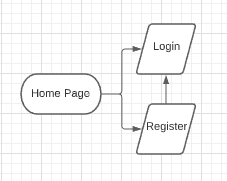
*This section should outline the design for how authentication and authorization was supported. This section should also contain all of the roles and privileges that are supported by the design.*

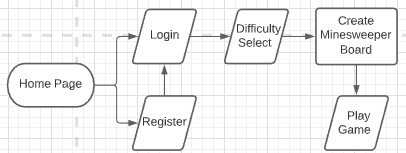
**Third Part Interface Design:**

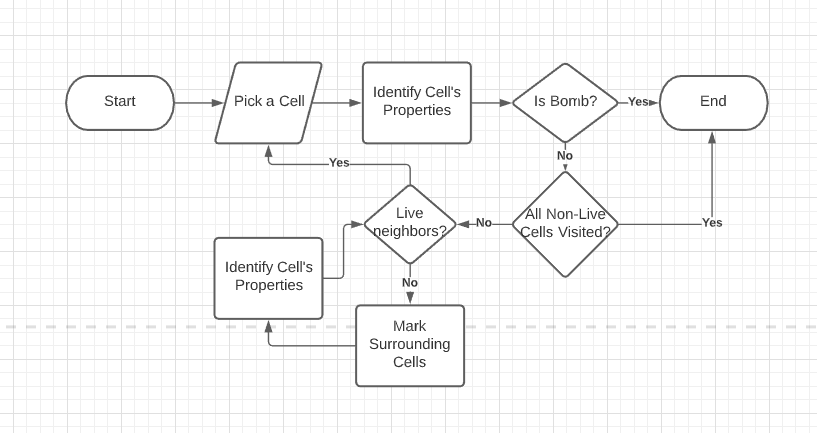
*This section should fully document any Third Party Service Interface API’s, how to access the service, what parameters are required by the API, and the detailed JSON data format specification that could be used by a third party developer to integrate with the service and API.*

**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.*

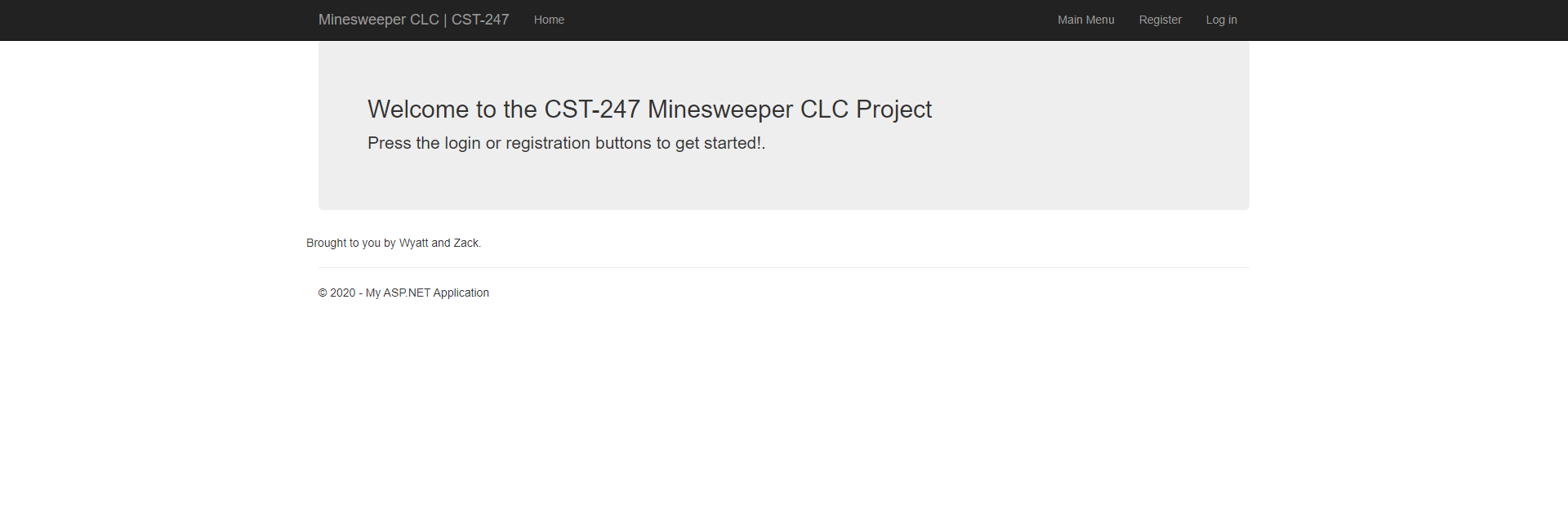


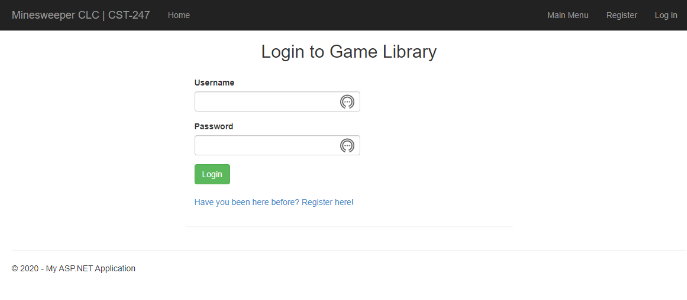
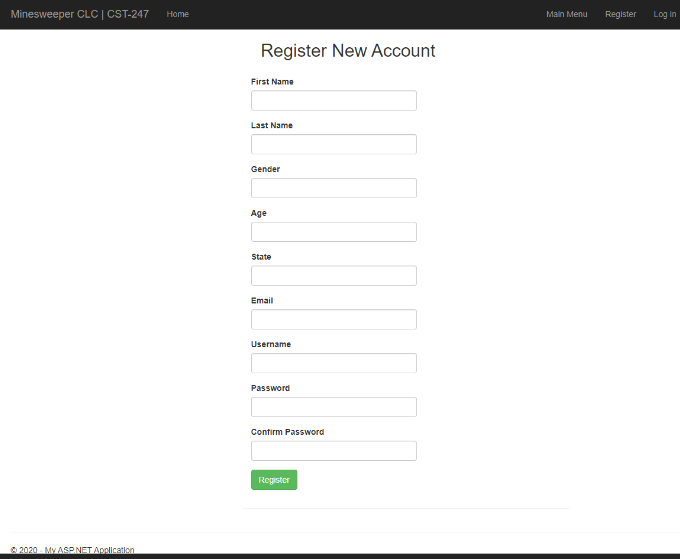


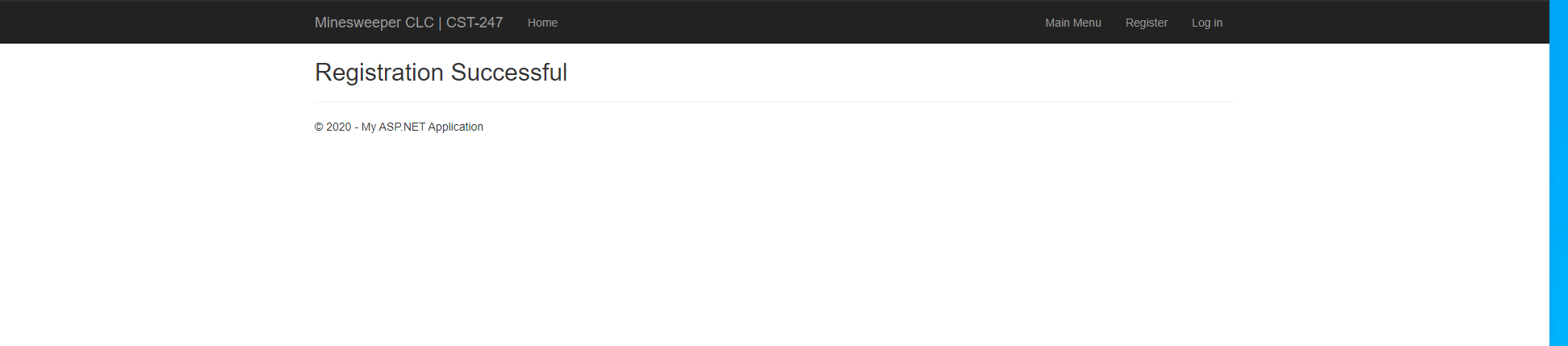


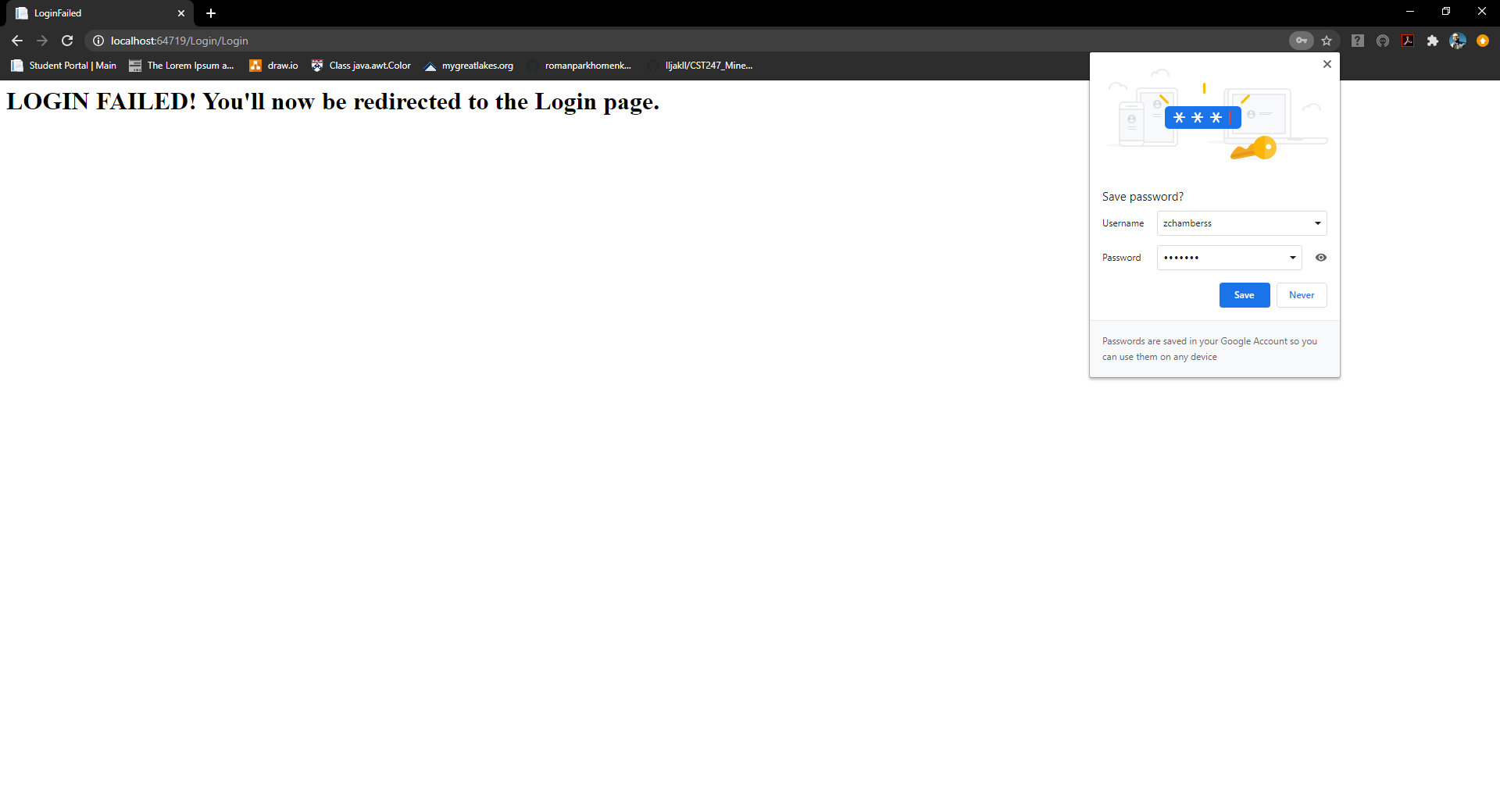
**User Interface Diagrams:**

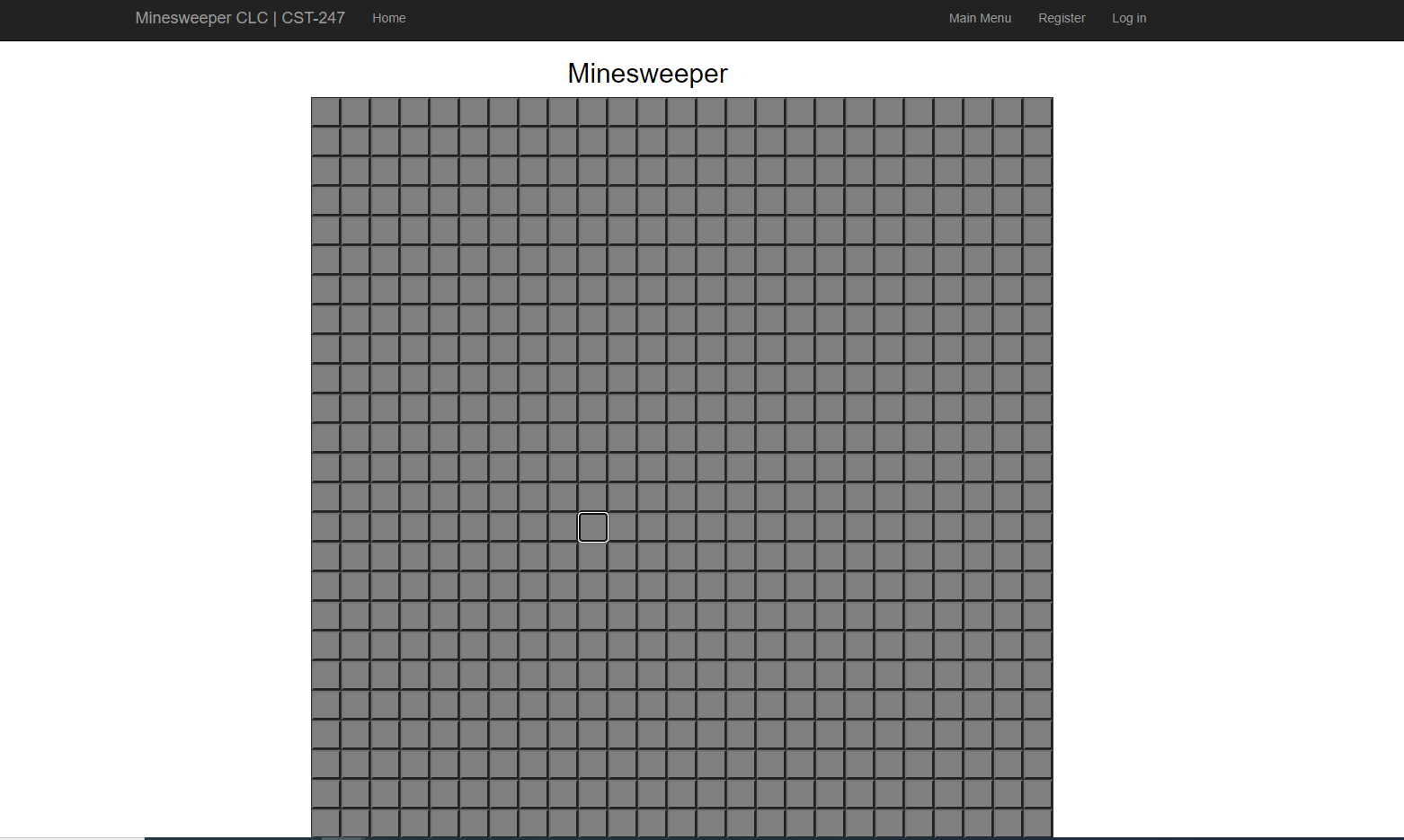
*You should insert any wireframe drawings or white board concepts that were developed to support your application. If you have no supporting documentation please explain the rational why you are able to leave this section as N/A.*

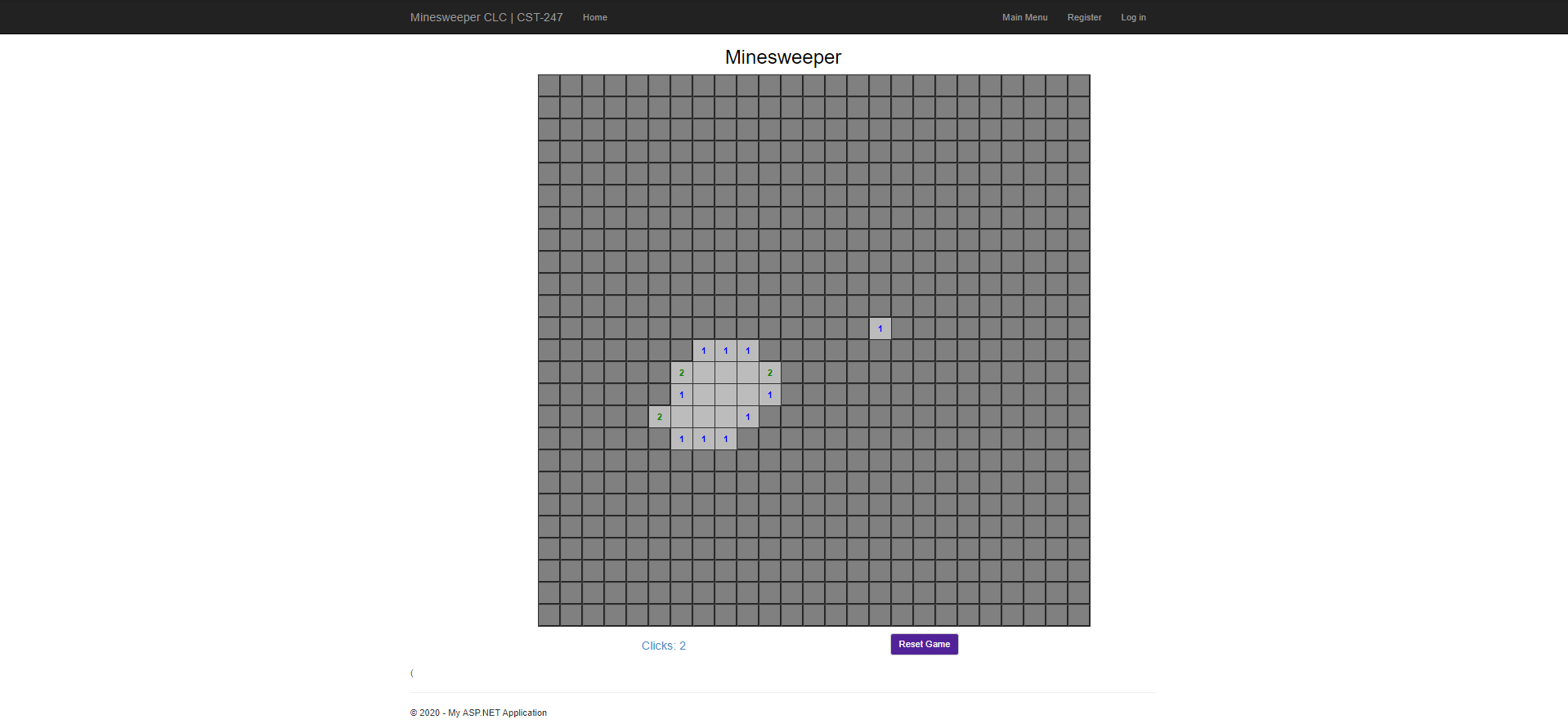


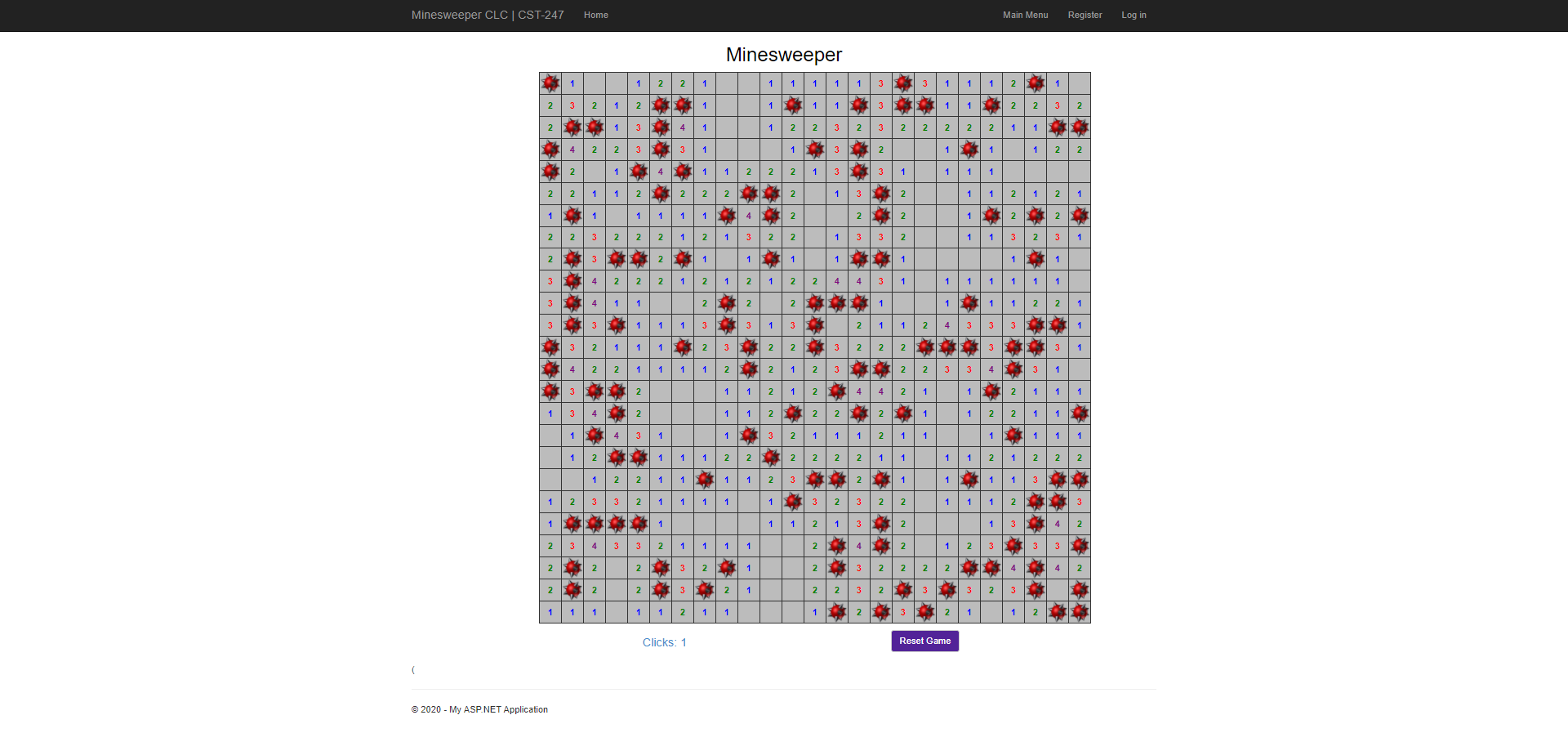




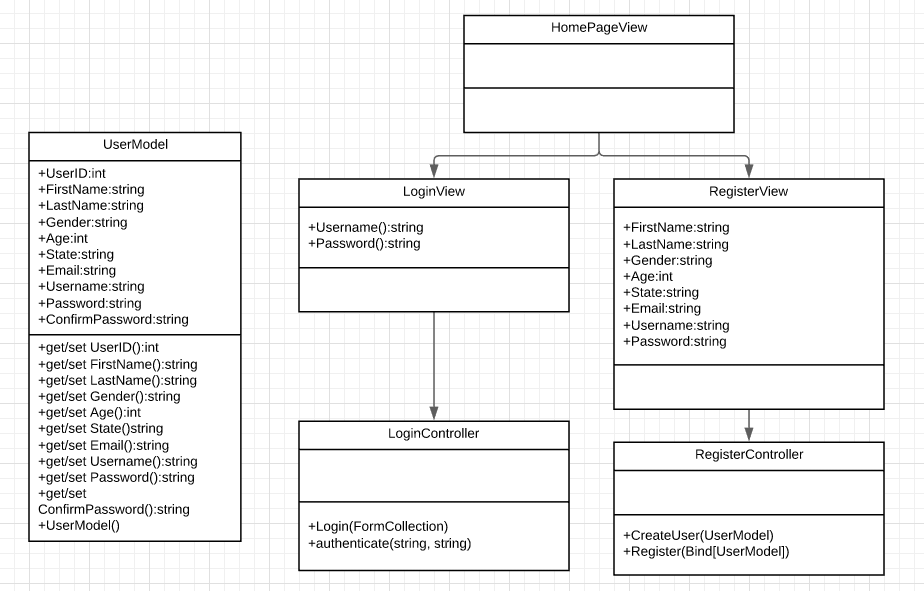


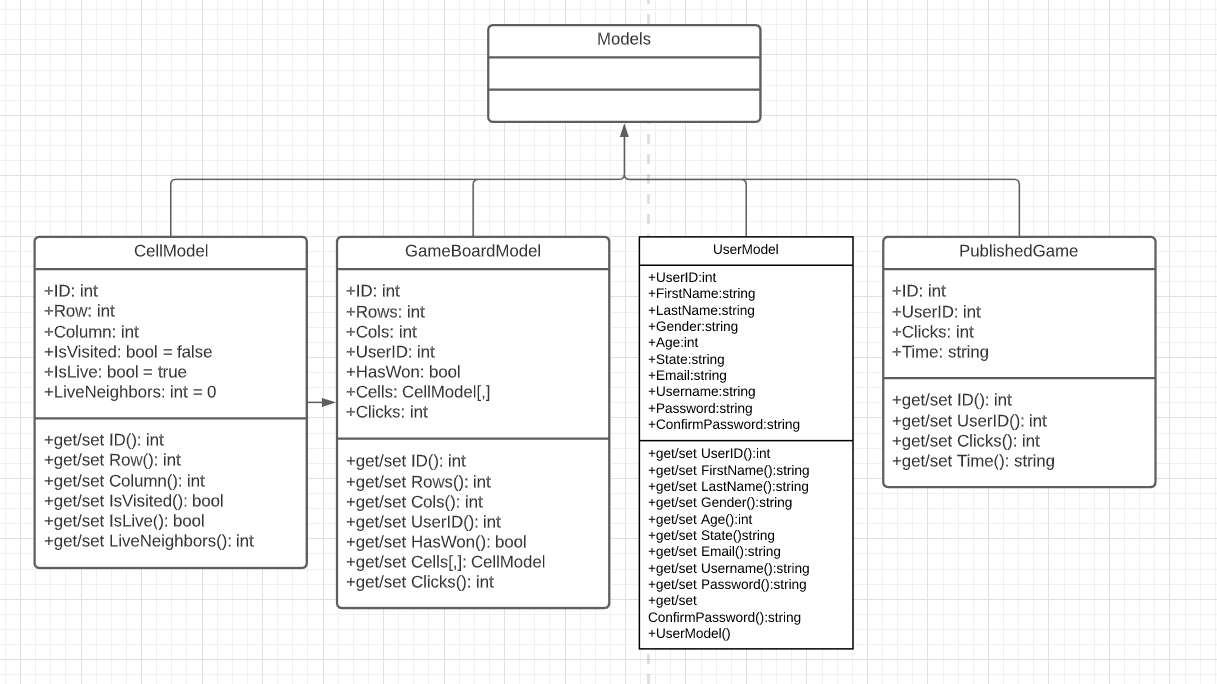




**Class Diagrams:**

*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.*

<https://github.com/zchambers3/CST247>

**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.*

We have discussions in the group forum. Share our progress on different aspects of the project.

Database Design: (Users, Grids, Cells, Stats)

CREATE TABLE [dbo].[Users] (

[ID] INT NOT NULL IDENTITY PRIMARY KEY,

[FirstName] VARCHAR (50) NOT NULL,

[LastName] VARCHAR (50) NOT NULL,

[Gender] VARCHAR (50) NULL,

[Age] INT NOT NULL,

[State] VARCHAR (50) NULL,

[Email] VARCHAR (50) NOT NULL,

[Username] VARCHAR (50) NOT NULL,

[Password] VARCHAR (50) NOT NULL

);

CREATE TABLE [dbo].[Grids] (

[ID] INT IDENTITY (1, 1) NOT NULL,

[Rows] INT NOT NULL,

[Cols] INT NOT NULL,

[HasWon] VARCHAR (5) DEFAULT ('false') NOT NULL,

[Clicks] INT DEFAULT ((0)) NOT NULL,

[UserID] INT NOT NULL,

PRIMARY KEY CLUSTERED ([ID] ASC),

CONSTRAINT [UserID] FOREIGN KEY ([UserID]) REFERENCES [dbo].[Users] ([ID]) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE [dbo].[Cells] (

[ID] INT IDENTITY (1, 1) NOT NULL,

[Col] INT NOT NULL,

[Row] INT NOT NULL,

[IsLive] VARCHAR (5) NOT NULL,

[IsVisited] VARCHAR (5) NOT NULL,

[LiveNeighbors] INT NOT NULL,

[GridID] INT NOT NULL,

PRIMARY KEY CLUSTERED ([ID] ASC),

CONSTRAINT [GridID] FOREIGN KEY ([GridID]) REFERENCES [dbo].[Grids] ([ID]) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE [dbo].[Stats] (

[ID] INT NOT NULL IDENTITY PRIMARY KEY,

[Clicks] INT NOT NULL,

[Time] VARCHAR(50) NOT NULL,

[StatsUserID] INT NOT NULL, CONSTRAINT StatsUserID FOREIGN KEY (StatsUserID)

REFERENCES Users (ID)

ON DELETE CASCADE

ON UPDATE CASCADE

);

