.

**.NET Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | CLC - Milestone 5: Save/Restore Game Progress and REST API Features | |
| **Date:** | 12/12/2020 | |
| **Revision:** | 1.4 | |
| **Team:** | 1. Kurt Newcomb | |
| 2. Isaac Tucker | |
|  | |
|  | |
| **Weekly Team  Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team Member** | **Hours Req** | **Hours Remaining** | | Make game only accessible to logged in users | Isaac | 4 | 0 | | Update header so Login and Register change once someone is logged in | Isaac | 4 | 0 | | Make it so users can save their gamestate and load from last save (if not win/lose) when they return. | Isaac | 10 | 0 | | Provide REST services so that game scores can be retrieved. | Kurt | 10 | 0 | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | https://github.com/MrAizakku/CST247CLC.git | |
| **Peer Review:** | Yes | 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:**

*The Product backlog is available in the Git repository:**https://github.com/MrAizakku/CST247CLC.git*

*The document is located within the* [*PlanningDesign*](https://github.com/MrAizakku/CST247CLC/tree/main/PlanningDesign) *folder.*

**Agile Scrum Burn Down Chart:**

*The SCRUM burn down chart is available in the Git repository:**https://github.com/MrAizakku/CST247CLC.git*

*The document is located within the* [*PlanningDesign*](https://github.com/MrAizakku/CST247CLC/tree/main/PlanningDesign) *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** |
| 1. Team communication seems to be going well; all parties reached out within a timely manner and joined a shared medium of communication (discord). |
| 1. Work distribution is going well; each team member grabbed a portion of the assignment and started working on that portion to ensure entire assignment completion. |
|  |

*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** |
| Some minor issues with the REST services and searching user | Use Joins so the return Table has data that works with REST string limitations | 12/12/2020 |
| Some members are finding difficulty securing time to contribute to CLC progress due to life distractions. | Ensure time is protected as to not be interrupted by outside influences. | Everyday |
| Some users experienced bugs and couldn’t get components to work | Review code via live stream on Discord to work out bugs. | 12/12/2020 |

**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.*

At this stage in the program development process there is nothing to install.

**SQL Database:**

User Table:

CREATE TABLE [dbo].[Users] (

[UserID] UNIQUEIDENTIFIER DEFAULT (newid()) NOT NULL,

[FirstName] NVARCHAR (50) NOT NULL,

[LastName] NVARCHAR (50) NOT NULL,

[Sex] NVARCHAR (50) NOT NULL,

[Age] INT NOT NULL,

[State] NVARCHAR (15) NOT NULL,

[Email] NVARCHAR (50) NOT NULL,

[Username] NVARCHAR (50) NOT NULL,

[Password] NVARCHAR (50) NOT NULL,

[GameString] TEXT NULL,

PRIMARY KEY CLUSTERED ([UserID] ASC)

);

CREATE TABLE [dbo].[Scores] (

[Id] UNIQUEIDENTIFIER DEFAULT (newid()) NOT NULL,

[userID] UNIQUEIDENTIFIER NOT NULL,

[score] INT NOT NULL,

[difficulty] NVARCHAR (10) NOT NULL,

[clicks] INT NOT NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

);

**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 through the design of your project.*

At this stage in the program development process we are simply following the instructions as they appear within the milestone assignment. Specifications for attributes within the user class are as defined by the milestone instructions. Views are kept simple to ensure program operation is the main focus. Bootstrap by Twitter has been installed using Nuget to enhance overall views. Minesweeper view has been created with a corresponding controller to make the minesweeper game fully functional with win/lose conditions. Right clicking is also enabled to flag cells.

**Key Technical Design Decisions:**

*At this point we are focusing on the required milestone requirements, and will make QoL updates as the project progresses throughout the course of the class. I.e. a check for an already existing user during the registration.*

**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 Party 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:**

*The User Interface Diagrams, or Wireframes, are available in the Git repository:**https://github.com/MrAizakku/CST247CLC.git*

*The document is located within the* [*PlanningDesign*](https://github.com/MrAizakku/CST247CLC/tree/main/PlanningDesign) *folder.*

**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 rationale 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 rationale why you are able to leave this section as N/A.*

**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.*