**CST-350 Project Status and Design Report**

**.NET Application Programming**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Topic 5: REST Services in ASP .NET Core | |
| **Date:** | *12/24/2023* | |
| **Revision:** | *3.0* | |
| **Team:** | 1. Dakoda Meade | |
| 1. Jamie Lewis | |
|  | |
|  | |
| **Milestone Task Summary: 8** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | As a user, I would like to be able to register an account so that I can login. | *Dakoda Meade* | *8* | *0* | | As a user, I would like to be able to login so that I can access the game | *Jamie Lewis* | *6* | *0* | | As a user, I would like to start a game so that I can see the game board. | Jamie Lewis | 4 | 0 | | As a user, I would like to click the buttons on the game board so that I can interact with the game. | Dakoda Meade | 7 | 0 | | *As a developer, I would like to remove full page refreshes when a user clicks a cell on the game board so that I can save memory and create a smoother user experience.* | *Dakoda Meade* | *6* | *0* | | *As a user, I would like to right click on the game board to place a flag so that I can keep track of potential bombs on the board not be able to left click that cell.* | *Dakoda Meade* | *4* | *0* | | *As a user, I would like it to display whether I win or lose a game so that I can see my results when the game is over.* | *Jamie Lewis* | *2* | *0* | | *As a developer, I want to maintain a clean code base by performing occasional refactoring and cleanup* | *Jamie Lewis* | *8* | *0* | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **Repository URL:** | https://github.com/jtlewis81/CST-350/tree/main/CLC%20Project | |
| **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:**

<https://onedrive.live.com/edit.aspx?resid=8B6EC91F88275E78!14618&cid=8b6ec91f88275e78&CT=1701265493598&OR=ItemsView>

In the project backlog tab.

**Agile Scrum Sprint Backlog:**

<https://onedrive.live.com/edit.aspx?resid=8B6EC91F88275E78!14618&cid=8b6ec91f88275e78&CT=1701265493598&OR=ItemsView>

In the burn down chart tab.

**Agile Scrum Burn Down Chart:**

<https://onedrive.live.com/edit.aspx?resid=8B6EC91F88275E78!14618&cid=8b6ec91f88275e78&CT=1701265493598&OR=ItemsView>

In the burn down chart tab.

**Agile Retrospective Results:**

*The following table should be completed after each Retrospective on things that went well (keep doing).*

|  |
| --- |
| **What Went Well** |
| **Communication** |
|  |
|  |

*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. An Action Plan is a short statement describing what you will do differently. The due date for the plan is when you will implement the change.*

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
|  |  |  |
|  |  |  |
|  |  |  |

**Design Documentation**

**Install Instructions:**

*Step-by-step instructions for setting up your database and 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.*

First go to link provided: <https://github.com/jtlewis81/CST-350/tree/main>

This is our git repository that has all the files for this project. From here we click the green code button and download a zip from this.

A screenshot of a computer

Description automatically generated

Then open visual studio and open SQL server object explorer.

A screenshot of a computer program

Description automatically generated

Create a new database called Minesweeper as show. Run the SQL file in the documentation folder. This will add a table called Users and a user to that table.

Then open the Minesweeper solution file in the application folder. Then you should be able to run the application. A web browser will appear and it is ready to use.

**Key Technical Design Decisions:**

*Any final technical design decisions, such as framework decisions, addressing its purpose in the design and why it was chosen.*

**ER Diagram:**

*The ER diagram shows the design of database tables and foreign key relationships. Include an image file of your ER database diagram.*

*A screenshot of a computer

Description automatically generated*

Diagram of the current and only database table, the dbo.Users table.

**DDL Scripts:**

*The DDL script is the SQL export text file from a database. This should contain a link to a private repository wherefrom the DDL script can be downloaded.*

<https://github.com/jtlewis81/CST-350/blob/main/CLC%20Project/Documentation/dbo.Table.sql>

**Sitemap Diagram:**

*The Sitemap shows a navigation path that the user can take through the application. Include an image file of your Sitemap diagram.*

*A diagram of a computer program

Description automatically generated*

Current 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.*

A user can register an account, which stores their information in a SQL database. They are authenticated at login by comparing the username and password to existing database records. User login persistence and logout is still in the works.

**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.*

A diagram of a flowchart

Description automatically generated

Game logic flowchart

**User Interface Diagrams:**

*You should insert any wireframe drawings or whiteboard concepts that were developed to support your application.*

A screenshot of a computer login

Description automatically generated A screenshot of a login form

Description automatically generatedA grid of squares on a white background

Description automatically generated

User Interface Diagram mockups

**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.*

A screen shot of a computer screen

Description automatically generated

Diagram for the models

A diagram of a computer

Description automatically generated with medium confidence

Class diagrams for the Controllers and Views

A screenshot of a computer

Description automatically generated

Services class diagrams

**Pseudo Code:**

*You should provide a URL references to any code stubs and pseudocode. If you have no supporting documentation, please explain the rationale for why you are able to leave this section as N/A.*

Milestone 1: Code for this project, so far, has been adapted from the examples set forth in Activity 2, copies of which can be accessed by:

* Go to the root of the GitHub repository linked above
* Navigate to the Topic 2/Activity 2/Applications folder
* Explore the Activity applications

**Other Documentation:**

*You should insert any additional drawings, storyboards, whiteboard pictures, project schedules, tasks lists, etc. that support your approach, design, and project. If you have no supporting documentation, please explain the rationale for why you are able to leave this section as N/A.*