**Cloud 9 Final Project**

Benjamin M. Brandhorst

University of Maryland Global Campus

SDEV 400 7980 Secure Programming in the Cloud

Doctor James Robertson

May 4th, 2020

**Role-Playing Game Character Generator Overview and User Guide**

This application allows a user to create, edit, and delete a role-playing game (RPG) character like are used in pen and paper games such as Dungeons and Dragons (D&D) or Multiuser Dungeons (MUDs). Additionally, users can view a list of all saved RPG characters and export their character information to a text file in the specified S3 bucket. Figure 1 gives an overview of the initial menu selection options.

A screenshot of a cell phone

Description automatically generated

**Figure 1 – Application Main Menu**

Users must select a menu item using the numbers 1-6. If the user inputs anything else they are warned and given the option to return the menu again. Figure 2 demonstrates incorrect menu selection input.

A screenshot of a cell phone

Description automatically generated

**Figure 2 – Incorrect menu selection input**

If users select the “Create a new character” option they are asked to enter their adventurer’s name and character class before being presented with a set of randomly generated RPG statistics. Each stat is generated within a range of 3 to 18 (simulating the range three six-sided dice could give). As seen in Figure 3, users are asked if they wish to keep these stats.A screenshot of a cell phone

Description automatically generated

**Figure 3 – Character stat generation**

Figure 4 demonstrates how the application will only accept “Y”, “yes”, “N”, or “No” as inputs to continue.

A screenshot of a cell phone

Description automatically generated

**Figure 4 – Character stat incorrect input**

Once stats are accepted, users are brought to an additional menu where they are given 50 gold pieces to spend on a list of 15 starting items (see Figure 5). Each time a menu item is selected, the specified amount of gold is deducted, and that equipment is added to the RPG character equipment list.

A screenshot of a cell phone

Description automatically generated

**Figure 5 – Starting gear list**

Figure 6 shows how any selection other than the sixteen menu options simply brings the user back to the menu.

A screenshot of a cell phone

Description automatically generated

**Figure 6 – Incorrect starting gear selection demonstration**

Figure 7 shows the output when an item is selected for purchase. Additionally, Figure 8 shows the output when a user selects an item they do not have enough gold to purchase.

A picture containing table, water

Description automatically generated

**Figure 7 – Starting gear purchase**

**A screenshot of a cell phone

Description automatically generated**

**Figure 8 – Starting gear purchase attempt when short gold**

Once users are finished purchasing items, they should select the “Exit and save character information” option. This saves the character information to a DynamoDB table along with a list of their starting gear. Users are then brought back to the main application menu.

Users can also edit existing characters. To do so they must know the name of the character they wish to edit. Figure 9 shows what happens if users try to edit a character that does not exist in the table. Figure 10 demonstrates attempting to edit a stat that does not exist.

A screenshot of a cell phone

Description automatically generated

**Figure 9 – Editing a character that does not exist**

A screenshot of a cell phone

Description automatically generated

**Figure 10 – Editing a stat that does not exist**

If a user identifies the character and the stat they wish to edit, they are given the option to change the value to something else. Once complete, users are asked if they wish to edit any other stats as can been seen from Figure 11.

A close up of a sign

Description automatically generated

**Figure 11 – Successful stat edit**

If a user does not know which characters are saved in the database, they can selection “View all saved characters” for a listing of all of the characters (see Figure 12).

A screenshot of a cell phone

Description automatically generated

**Figure 12 – Viewing all saved characters**

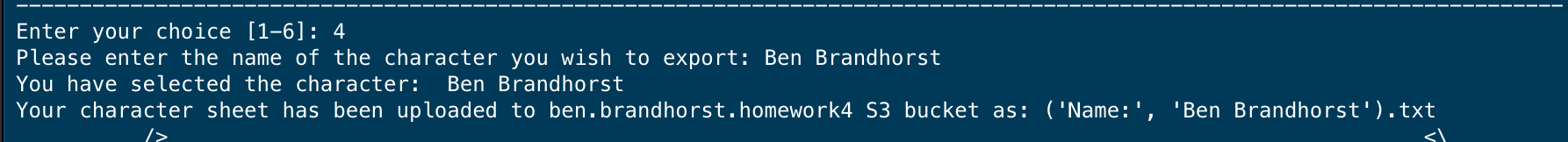
Users also have the option to export existing characters to a text file in an S3 bucket. This is more of a proof of concept rather than something that is actually useful. If I had more time to work on formatting the output, the idea would be the possibility of filling an RPG character sheet with the information generated from the application. As it is, the export option simply fills a text file with the character information in a hastily formatted way that allows identification of the different stats but is not visually pleasing. Again, the user must select a character that exists in the DynamoDB table. If they input a nonexistent character, they are warned and brought back to the main menu (see Figure 13).

A screenshot of a cell phone

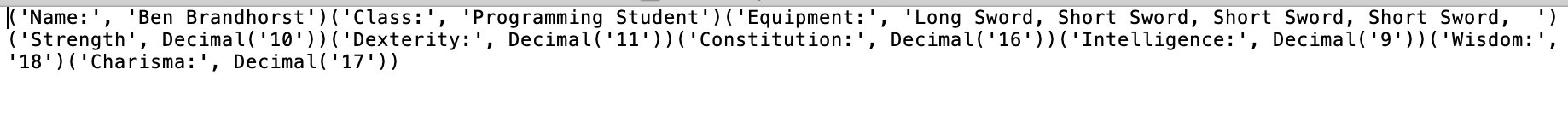
Description automatically generated

**Figure 13 – Exporting a character that does not exist**

Figure 14 shows application output when users select an existing character for export. Figure 15 shows the S3 Bucket text file contents after being exported.



**Figure 14 – Successfully exporting a character**

****

**Figure 15 – S3 Bucket text file contents after export**

Finally, users can opt to delete an existing character. As before, if a name which doesn’t exist is input, users are warned and returned to the main menu (see Figure 16).

A screenshot of a cell phone

Description automatically generated

**Figure 16 – Attempting to delete a nonexistent character**

However, if an existing character is selected, users are notified that the character has been deleted and asked if they want to delete another (Figure 17).

A screenshot of a cell phone

Description automatically generated

**Figure 17- Successfully deleting a character**

The last menu selection item from the main menu simply exits the program, as can be seen in Figure 18.

A screenshot of a cell phone

Description automatically generated

**Figure 18 – Exiting the program**