**Localization Tool**

**Planning Document**

Assignment 1

SP1 2017 Introduction to C# and Tools Development

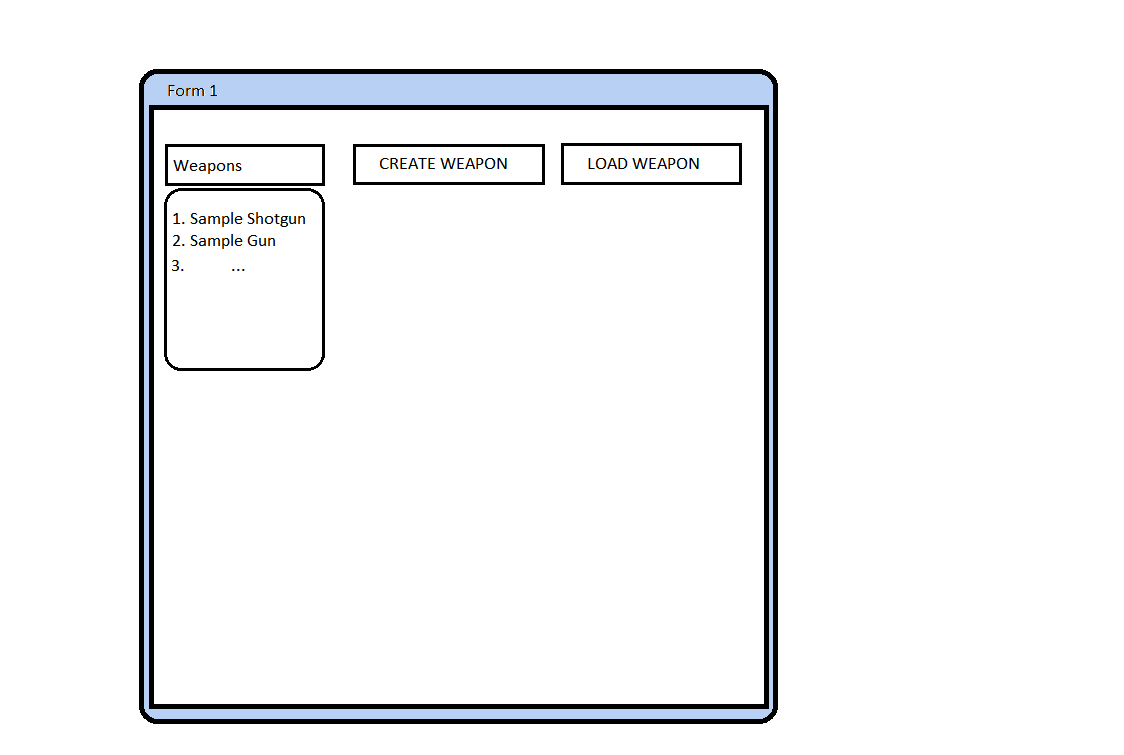
Academy of Interactive Entertainment

|  |  |
| --- | --- |
| Author | Sarthak Saxena |
| Contact | Sarthaksaxena42@yahoo.com |

**Executive Summary**

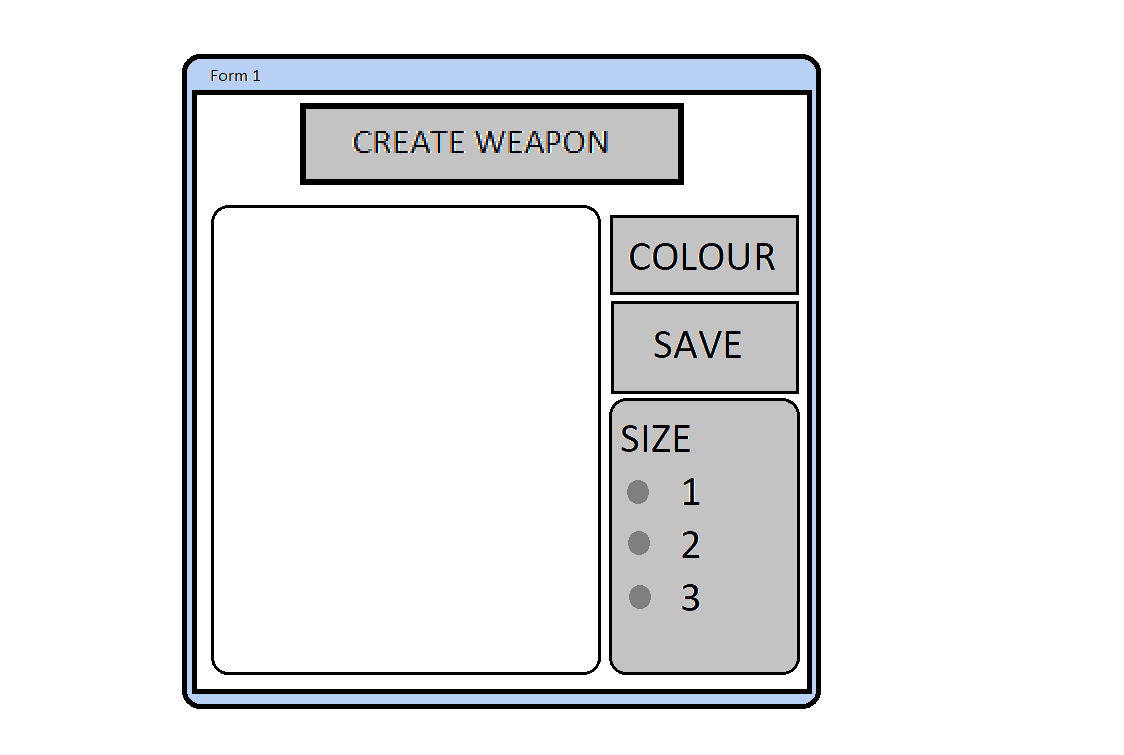
This document  
  
  
  
**Screen Mock-Ups**

**Main Window**

Figure 1. Shows the main interface for the tool.  


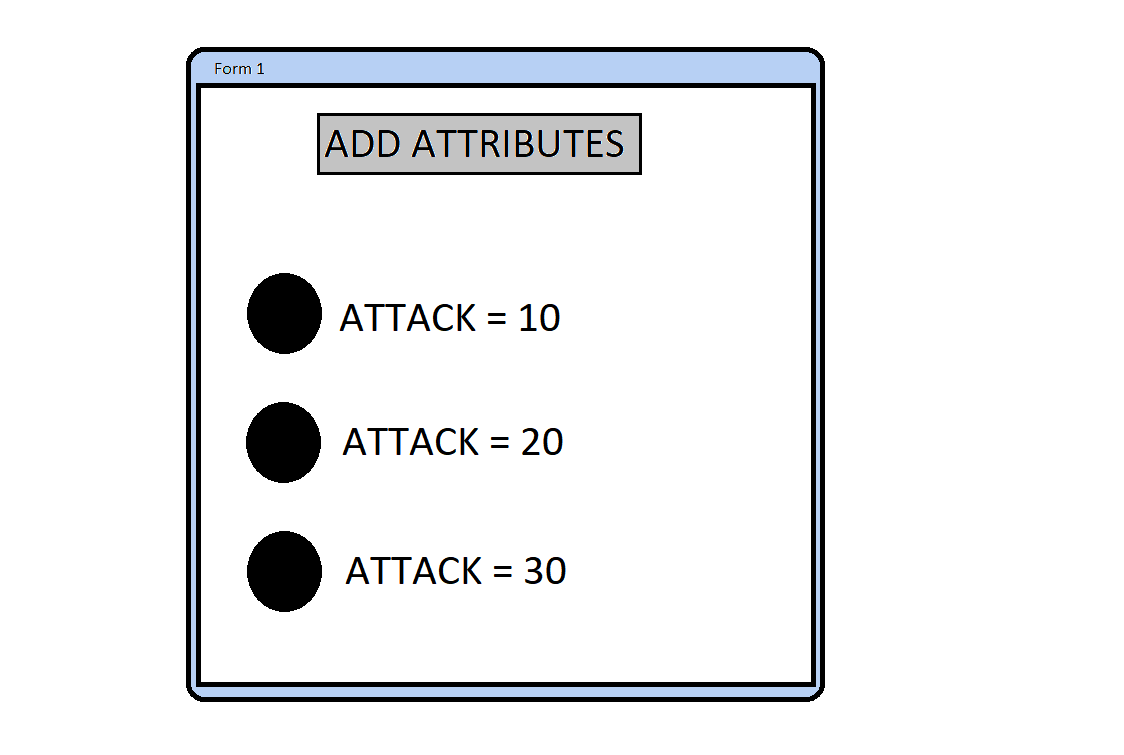
**Figure 1. Shows the main interface for the tool.**

Buttons such as the “Create Weapon” allows the creation of weapons. “Load Weapon” loads in saved weapons from list of “Weapons” list. By pressing on the one of the items in list, then pressing “Load Weapon” it will load weapons image, and attributes assigned to it from a local saved folder.  
  
**Create Window**Allows the user to create a weapon by hand and save it referring to figure 3.   
Buttons such as “Colour” will allow the user to pick from a range of colours to draw weapon. As well as the button “Save” opens up a save window referred in figure 3. from which you may save your creation. The “Size” button is to add depth to your weapon, so the user feels more enticed into creating something with a different size so each weapon looks creative and unique. Once the user is happy with their image they can “Save” or press “Create Weapon” to open up attributes screen referred on Figure 3.



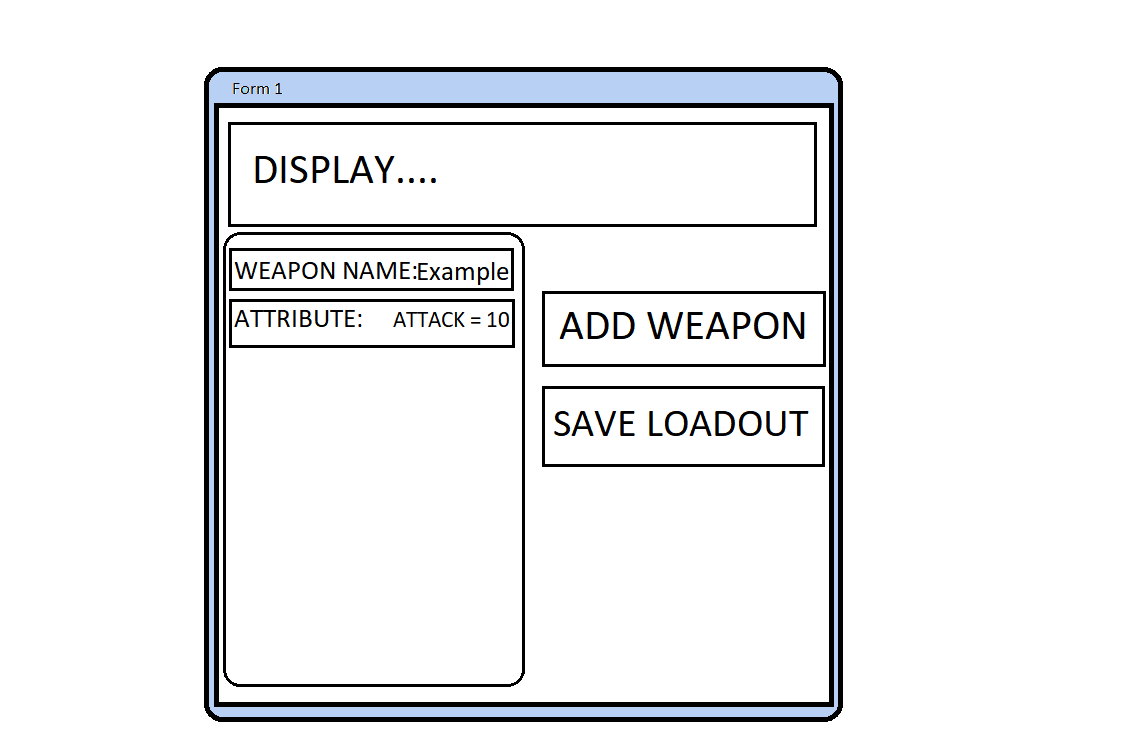
**Figure 2. Open up window to draw Weapon**

**Attributes Screen**The user will be guided to a list of attributes to place on their weapon image, to finalize. A list of attributes will be displayed, and the player has to pick between which option to keep on the weapon. This is displayed through Figure 3.

****

**Figure 3. Shows adding attributes to Weapon**

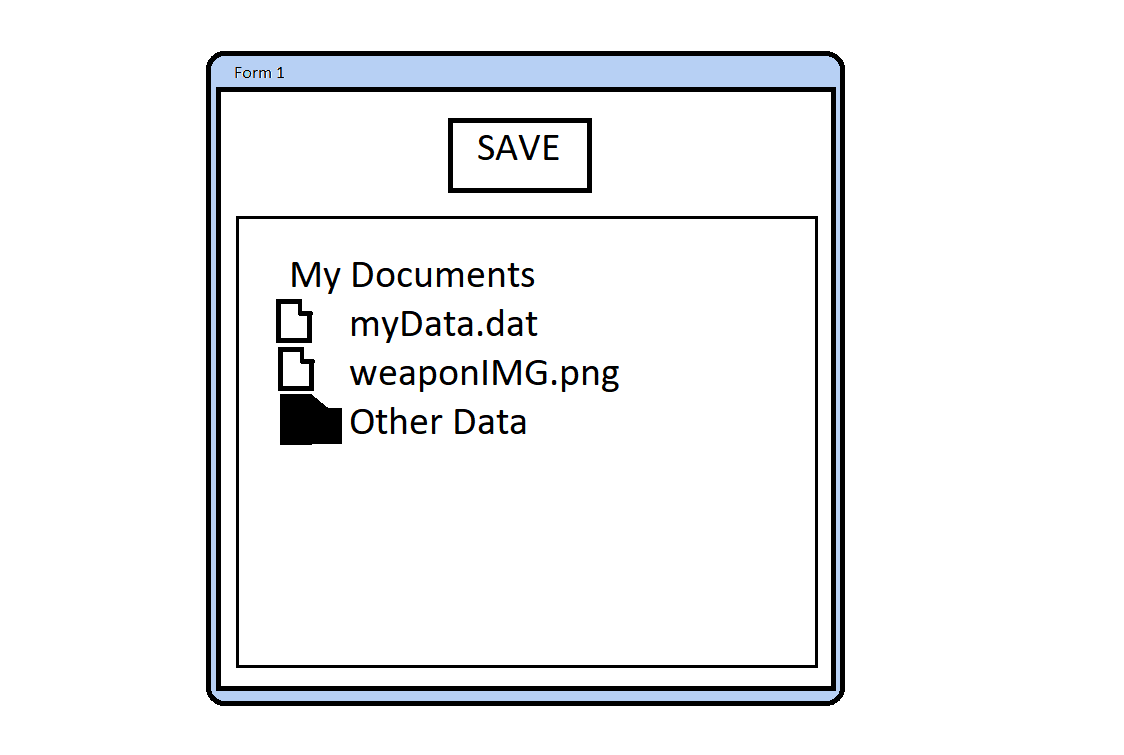
Buttons displayed are the attributes that are accessible to user. The Button “Add Attributes” allows the user to finalize their choice after picking one of the three mock-up attributes. This will guide the user to their next screen in Figure 4. **Finalize Weapon Screen**The weapon screen displayed in Figure 4. allows the user to finalize the weapon created solely based on the user’s desire.

Weapon Screen   


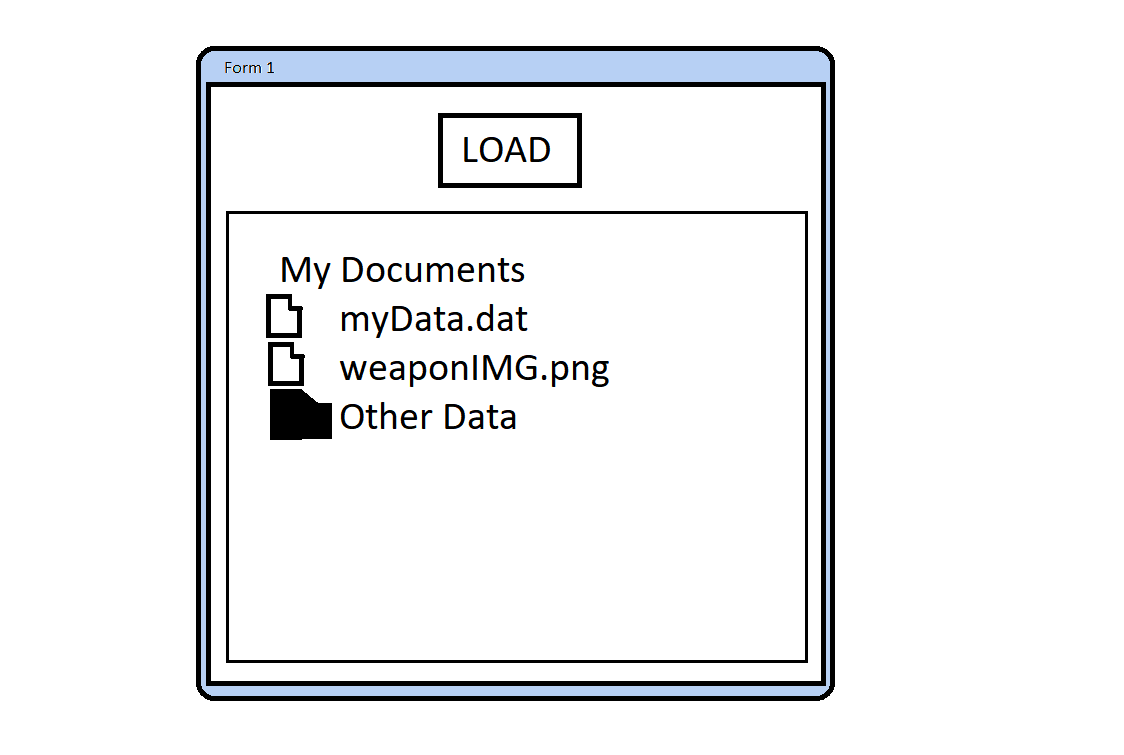
**Figure 4. Displays Final Created Weapon**

Buttons displayed as “Example” will be editable so that the user can manually write their desired name for the weapon. The button “Add Weapon” allows the user to finalize their weapon and add it to the weapons list in the Main Menu referred in Figure 1. The user may choose to save their entire load out which will lead them to the Save Screen referred in Figure 5. **Dialogues**

Saving and Loading dialog boxes are displayed when the save or load buttons (respectively) are pressed. They will be pressed at a further time in the tool, as shown in Figure 3 Save dialog box, and Figure 4 with the Load dialog box.

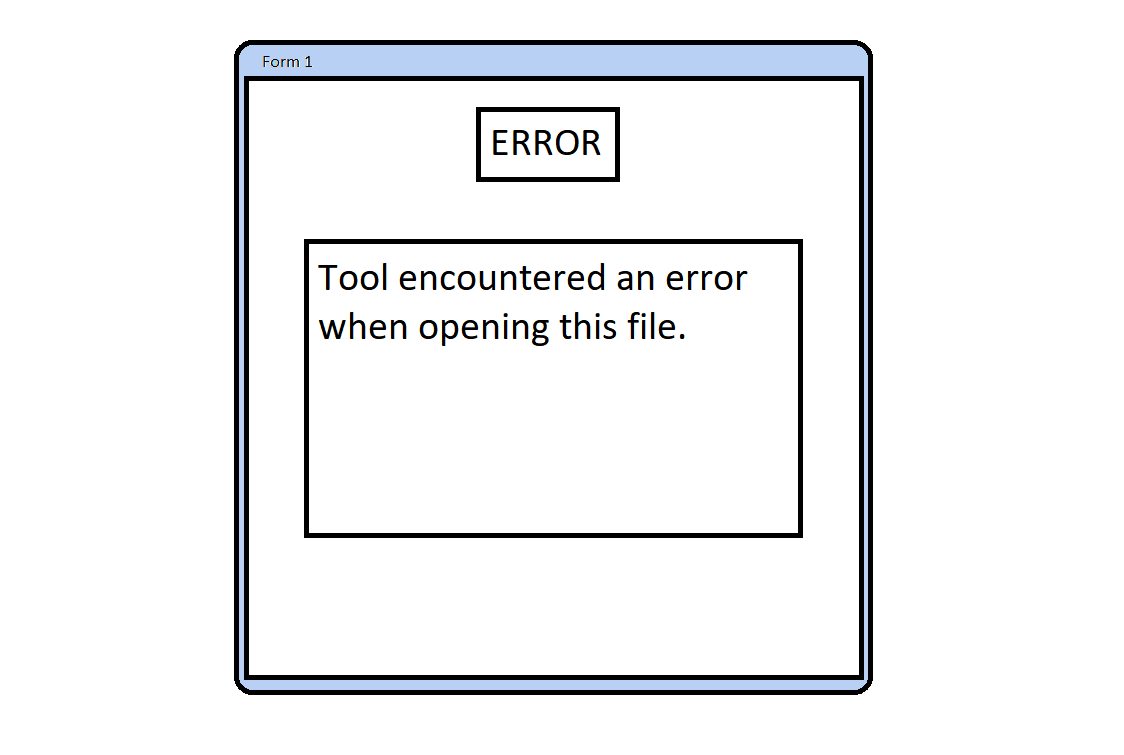


**Figure 5. Save Screen When Saving A Weapon’s Image and Data**



**Figure 6. Shows Loading in Weapon Attributes, and Weapon IMG**

If the user tries to load a file that is with incorrect format or corrupted the Error dialogue box shown in Figure 4 will be displayed.



**Figure 7. Error displayed when loading incorrectly**

**Class Diagram**

Figure 5. Class Diagram

The class diagram shows the main classes used in the tool.

The main program will store a collection of UIText objects, which contain the English word or phrase along with a collection of translations for the languages specified.

Ancillary classes for dialog boxes and other utility classes are not shown.

**State Diagram**

The State Diagram shown in Figure 6 shows the various program states and the transition between them.

The user will spend the majority of their time in the ‘Enter Text’ state, only transitioning to the other program states when saving or loading the application data.

Figure 6. State Diagram

**Sequence Diagram**

The sequence diagram shown in Figure 7 shows how UIText elements are created or updated in response to the user entering text via the main program user interface.

Figure 7. Sequence Diagram showing the updating of UIText objects in response to user input.

**Save Data Format**

Data will be saved in a comma-separated value format (.csv)

The first entry will be the English phrase or word, followed by translations in the desired languages. Where a translation for a specific language hasn’t been added, a space will be inserted instead.

An example of a data file saved by this program is shown in Figure 8.

Figure 8, Saved data format