# RPG Character Creator Documentation

Ethan Buchanan and Steven Garcia-Alamilla

**Abstract**

**The RPG Character Creator is a tool that lets users create their own characters for RPG games, from Tabletop RPG enthusiasts to casual gamers. This project is currently in the planning stage and is still being worked on.**

## 1. Introduction

Character creation is a very important part of any RPG experience. The RPG Character Creator is designed to provide an easy way to plan out future RPG journeys, whether you are an experienced RPG veteran or new to the RPG genre, the RPG Character Creator will provide a fun and engaging way to customize certain aspects of your character.While it is not necessary to plan out your character in advance, doing so allows stronger, more focused characters that can make your experience playing through the game better. While we can't provide a RPG character creator that covers every RPG system, we do hope our more generalized system allows the user to create a character that covers many different games.

### 1.1. Background

If you are unfamiliar with the RPG (Role-Playing Game) genre, there are a few things you should know. Firstly, RPGs allow you to play the role of a fictional character in a narrative-driven story. You are then given the ability to create and customize you character choosing who and what you want to play as. Some different attributes that you might be able to customize would include:

- Name
- Appearance.
- Class
- Race
- Background
- Skills and Abilities

### 1.2. Impacts

While this project might not be impactful to everyone,we hope that the RPG Character Creator will have a major impact with people who enjoy playing RPG games. It would allow for users to plan out their character for their next RPG journey, which can enhance their overall experience.

### 1.3. Challenges

A few challenges that we anticipate through the course of this project include:

- Not being that acquainted with WPF Applications enough to do implement functionality.
- Creating a GUI that makes it easy for the user to navigate through.
- "Broken windows" in our code.
- Being able to finish the project on time.

A way we would overcome these challenges is by following the stages in the waterfall software development life cycle (SDLC). The SDLC makes sure that as we progress through the different phases of this project we keep in mind what has to be accomplished in order to move forward to the next stage.

## 2. Scope

The completion of the project would mean that the user can create a single or multiple RPG characters by customizing the characters Name, Class, Attributes, Stats and much more. The RPG Character Creator would show the list of characters that have been created allowing for the creation/deletion of characters and even modifying characters after they have been created. A few stretch goals we have include:

- Different rule sets for different types of RPGs (example: Fallout/ DnD 5e)
- Allow for the user to toggle a DM mode which allows them to add custom skills, abilities, races, and classes to the character creator allowing for more customization of characters.

| Use Case ID | Use Case Name | Primary Actor | Complexity | Priority |
|---|---|---|---|---|
| 1 | Create New Character | User | Med | 1 |
| 2 | Fill out Character Bio/Information | User | Low | 2 |
| 3 | Choose Character Portrait | User | Hard | 3 |
| 4 | Choose Character Class | User | Low | 2 |
| 5 | Choose Character Race | User | Low | 2 |
| 6 | Choose Character Background | User | Med | 2 |
| 7 | Choose Character Skills | User | Med | 3 |
| 8 | Choose Character Abilities | User | Med | 3 |
| 9 | Choose Character Alignment | User | Low | 2 |
| 10 | Finish Character on Overview Screen | User | Med | 1 |
| 11 | Edit Character | User | Med | 4 |
| 12 | Delete Character | User | Med | 5 |
| 13 | Export Character | User | Hard | 6 |

TABLE 1. RPG CHARACTER CREATOR USE CASE TABLE

## 2.1. Requirements

We want to make sure that our project meets the user's needs, so we looked at similar RPG character creator websites and looked at a variety of different video game character creators to understand what is required to create a fulfilling character creator that does not make the user want for more customization. We found that users prefer something that's easy to use, responsive, and gives the user many different options. Based on what we found users preferred, we created these requirements for our application.

### 2.1.1. Functional.

- The user should be able to create a new character.
- The user should be able to customize their character's stats, skills, abilities, choose from a large list of classes, choose from a wide variety of different races, and their character's background
- The user should be able to manage their characters by viewing, editing, and deleting them

### 2.1.2. Non-Functional.

- The application should be easy to use allowing the user to easily navigate through the GUI making it a more pleasant experience.
- Users will be able to use RPG Character creator without it feeling choppy or unresponsive.

## 2.2. Use Cases

Use Case Number: 1
Use Case Name: Create New Character
Description: The user starts the process for creating a new character.They will click on the "Create New Character" button this will take User to the character creator menu starting with the character information screen.
Basic flow for the process:

1) User navigates to the "Create new Character" button
2) User left-clicks on the button.
3) User is taken the character creation screen.

Termination Outcome: The user is now in the process of creating a new character. The user begins on the character info screen.

Use Case Number: 2
Use Case Name: Fill Out Character Bio/Information
Description: The user is now underway with creating their character. They have been taken to the first tab of character creation: the character information screen where they will fill in the Character's bio and basic information like name, age, gender, etc.
Basic flow for the process:

1) User is now on Character Bio screen
2) User can fill in each field for character info.
3) User can fill in character's back story.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.


Figure 1. Use Case 1: What a user would see with no characters created.

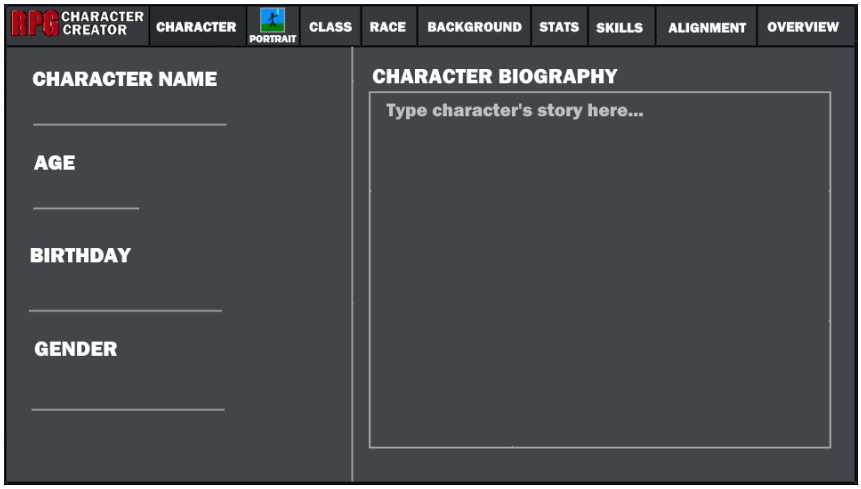


Figure 2. Use Case 2: What a user would see with no character info screen.

Use Case Number: 3
Use Case Name: Choose Character Portrait.
Description: Once the user has reached this screen, they can choose a portrait that will represent their character.
Basic flow for the process:

1) User is now on Character Portrait screen
2) User can choose from an already collected group of portraits.
3) User can import their own portrait for their character.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 4
Use Case Name: Choose Character Class.
Description: Once the user has reached this screen, they can choose a class for their character.
Basic flow for the process:

1) User is now on Character Class screen
2) User can scroll up and down the list to see what class they want.
3) By clicking on a class it will select what class they want to be.
4) Clicking on a class also shows a description of what each class is as well as the class's primary and secondary stats, class skills, and class abilities.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

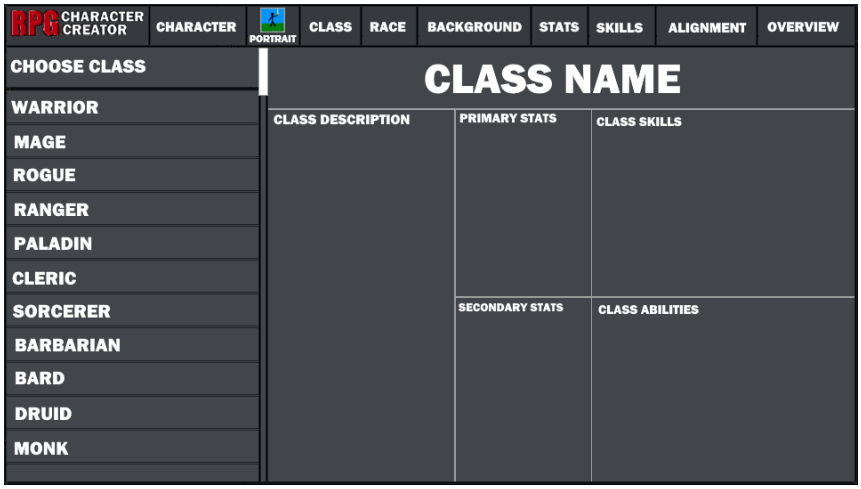


Figure 3. Use Case 4: What a user would see on the Character Class screen

Use Case Number: 5
Use Case Name: Choose Character Race.
Description: Once the user has reached this screen, they can choose a race for their character.
Basic flow for the process:

1) User is now on Character Race screen
2) User can scroll up and down the list to see what race they want.
3) By clicking on a race it will select what race they want to be.
4) Clicking on a race also shows a description of what each race is as well as the race's traits.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

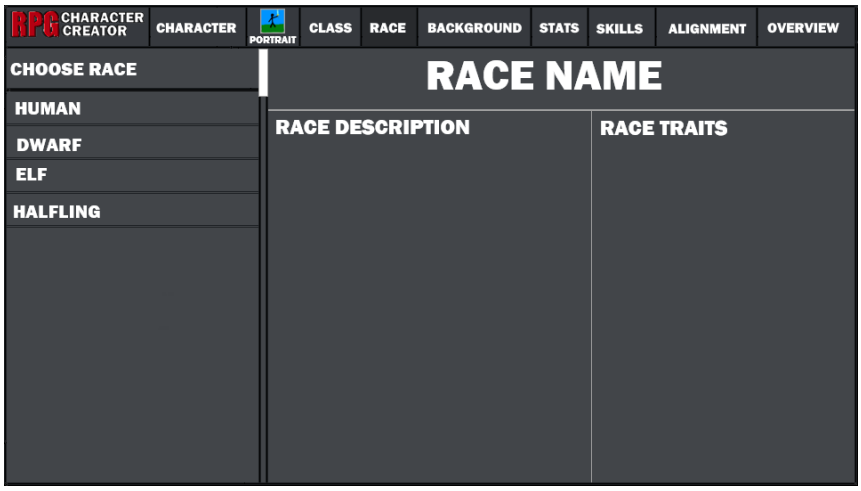


Figure 4. Use Case 5: What a user would see on the Character Race screen

Use Case Number: 6
Use Case Name: Choose Character Background.
Description: Once the user has reached this screen, they can choose a background for their character.
Basic flow for the process:

1) User is now on Character Background screen

2) User can scroll up and down the list to see what background they want.
3) By clicking on a background it will select what background they want to be.
4) Clicking on a race also shows a description of what each background is as well as what skills the background provides.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 7
Use Case Name: Choose Character Skills.
Description: Once the user has reached this screen, they can choose what skills they want their character to have. There is universal skills and class specific skills.

Basic flow for the process:

1) User is now on Character Skills screen
2) User can scroll up and down the list to see what skills they want.
3) By clicking on a skill it will select what skills they want to add to their character, they can add multiple skills.
4) Clicking on a skill also shows a description of what each skill entails.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 8
Use Case Name: Choose Character Abilities.
Description: Once the user has reached this screen, they can choose what abilities they want their character to have. There is universal abilities and class/race specific abilities.

Basic flow for the process:

1) User is now on Character Abilities screen
2) User can scroll up and down the list to see what abilities they want.
3) By clicking on a skill it will select what abilities they want to add to their character, they can add multiple abilities.
4) Clicking on a abilities also shows a description of what each abilities entails.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 9
Use Case Name: Choose Character Alignment.
Description: Once the user has reached this screen, they can choose what abilities they want their character to have. There is universal abilities and class/race specific abilities.

Basic flow for the process:

1) User is now on Character Alignment screen
2) User can mouse over each alignment which will give them a description of what kind of character falls in to each alignment.
3) Clicking on a alignment choose that alignment for their character.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation. At this point if the user went in order, they would go to the Overview tab.

Use Case Number: 10
Use Case Name: Finish Character on Overview Screen
Description: Once the user has reached this screen, they are on the final section. This screen gives an overview of what they selected and they can finally add the character to the system.

Basic flow for the process:

1) User is now on Character Overview screen.
2) User can now see all of what they have chosen for their character.
3) Clicking on ”Finish Character Creation” will add the character to the main menu.

Termination Outcome: Once the user clicks the ”Finish Character Creation” button they are taken to the main menu, exiting the character creation screen.

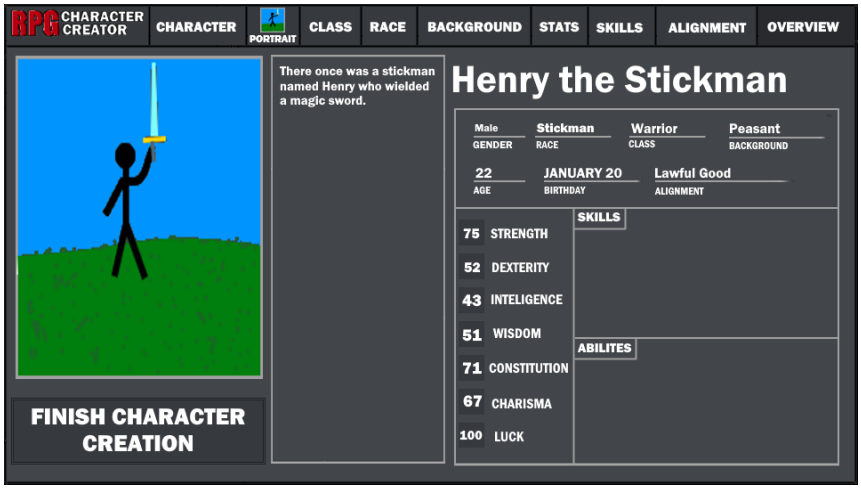


Figure 5. Use Case 10: What a user would see on the Character Overview screen

Use Case Number: 11
Use Case Name: Edit Character
Description: If there are characters in the system then the user can go back and edit their characters allowing them to go back through the character creation screen.

Basic flow for the process:

1) User is on Main Menu screen
2) User can click on character they want to select in left tab.
3) User can then navigate to bottom left of screen and press the Edit button.
4) This will then take them back to the character creation menu.

Termination Outcome: Once the user has edited the character to how they want they will go back to the Overview screen and press ”Finish character Creation” button again.

Use Case Number: 12
Use Case Name: Delete Character
Description: If there are characters in the system that the user wants to delete they can by using the delete character function.

Basic flow for the process:

1) User is on Main Menu screen
2) User can click on character they want to select in left tab.
3) User can then navigate to bottom left of screen and press the delete button.
4) A confirmation menu will then pop up and the user will need to press ”Yes” if they want to delete the character and ”No” if they want to cancel deleting the character.

Termination Outcome: Once the user has closed the confirmation menu, the character will be deleted and the App functions as normal.

Use Case Number: 13
Use Case Name: Export Character
Description: If there are characters in the system that the user wants to export to a file then they can.

Basic flow for the process:

1) User is on Main Menu screen
2) User can click on character they want to select in left tab.
3) User can then navigate to bottom left of screen and press the export button.
4) A menu showing where they want to export the character file to will show up and they choose where.

Termination Outcome: Once the user has chosen the location then, they character sheet should be saved to their computer.

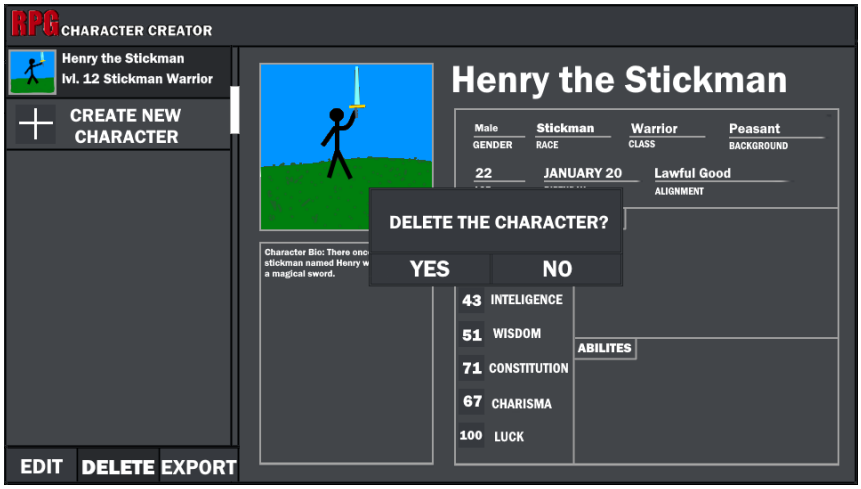


Figure 6. Use Case 12: What a user would see on when deleting a character

## 2.3. Interface Mockups


Figure 7. Might have intro menu maybe not

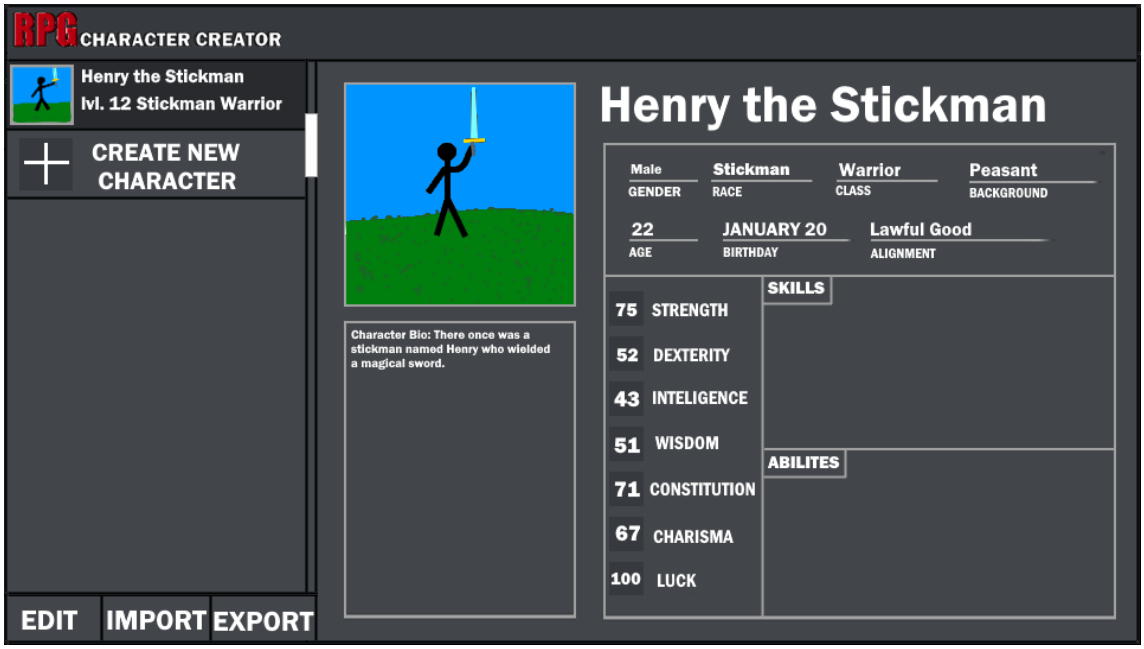


Figure 8. What the prgroam would look like with character added

## 3. Project Timeline

**Project Proposal Draft (February 04, 2023)** - First, we brainstormed ideas for a project that would be worth pursuing, and then we drafted a project proposal. This proposal included a project name, an abstract, an introduction, and a scope. Finally, we created a Github repository and uploaded the proposal to it.

**Project Proposal Update (February 28, 2023)** - The update to the project proposal involved identifying the use cases for the project, as well as defining its functional and non-functional requirements. In addition, several interface mockups were created to show our vision for the project.

**GUI Day (March 9th, 2023)** - Showcased our interface mockups to the class and gathered feedback on potential additions or modifications to our interface mockups.

Coding of the actual project began during these two periods beginning with the implementation of a basic UI and building upon it leading up to demo day.

**Demo Day (April 6th, 2023)** - Presented the current status of our project to our classmates and Dr. Ericson.

**Project Update (April 10th, 2023)** - We created a timeline to showcase the progression we have made so far, which involved adding project structure and outlining UML diagrams to project documentation.

**Penultimate Writeup(April 21, 2023)** - We plan to revisit our work and make necessary updates to our documentation to ensure that it is accurate and up-to-date by the specified time.

**Presentation Draft (April 21, 2023)** - We plan on presenting our slides, and giving a small demo of our current project to receive feedback before our final presentation.

**Final presentation(April 29, 2023)** - We plan on finishing our project and giving our final presentation to the class.

**Final Written Report (April 28, 203)** - We plan on having our final written report that will contain all needed information to show what we have accomplished.

## 4. Project Structure

First, we needed to determine our target audience. In our case, we decided that this RPG character creator would be geared towards everyone, including both experienced and beginner RPG players. We wanted to make the process as smooth as possible for the user, so we decided on having an easy-to-navigate user interface. We designed the RPG character creator with several sections, including Character which includes the basic character information such as name, age, birthday, and gender. There is also a portrait section where users can choose from pre-selected images or upload their own. The race, background, and alignment section allows users to choose from multiple options, and their information is filled in accordingly. There is also a stats and skills section and an overview page that lets users review their character's details before finalizing the creation process.

In order to stay organized throughout the development stage, we have decided to implement the builder design pattern in our RPG character creator application. By using this pattern, we can easily create complex objects (Characters in our case) by building them step by step. The builder pattern offers the advantage of creating multiple types of objects using the same interface, which will be helpful for our application since users can create a variety of RPG characters.We also implemented the abstract factory design pattern which will be used to create different color themes for the application, easily allowing user to switch between different themes without affecting the functionality or structure of the application.

In addition to the design patterns, we have also implemented the Model-View-ViewModel (MVVM) architecture in our RPG character creator application. MVVM helps to cleanly separate the GUI and program logic allowing for better maintainability and testability.

### 4.1. UML Outline

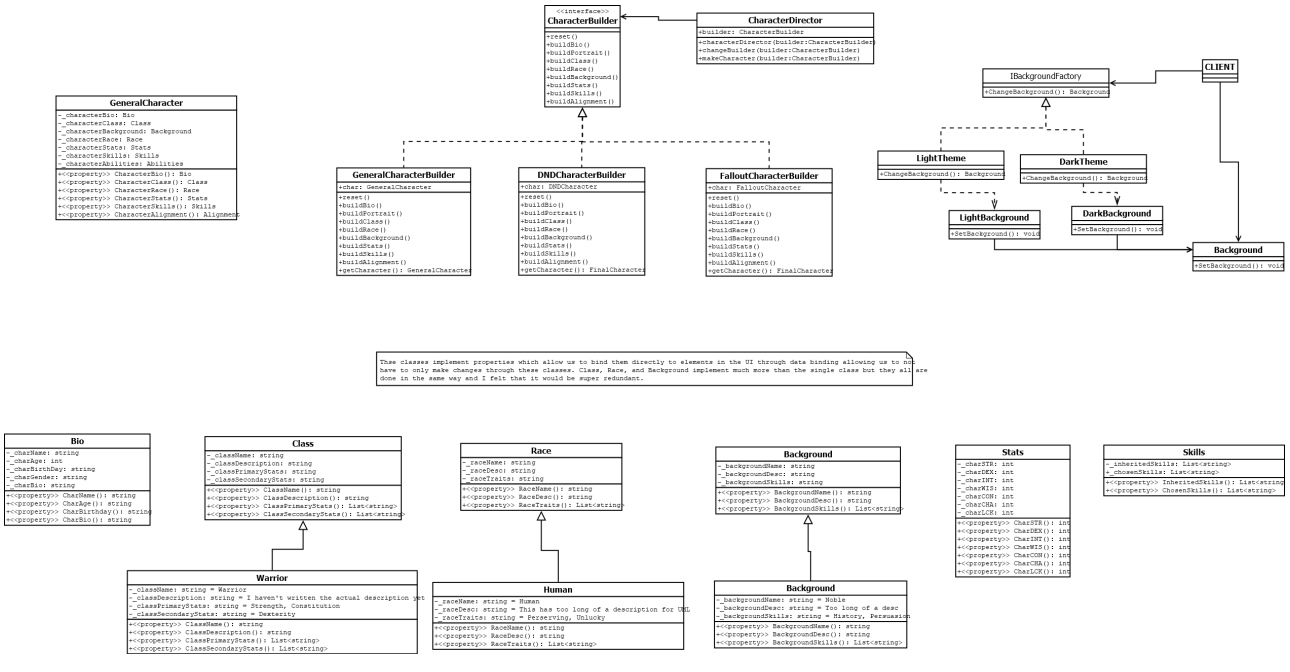


Figure 9. The entire UML for the project

The following UML is an Abstract Factory Design pattern see Figure 10

The BackgroundFactory is an abstract class that defines a factory method, 'CreateBackground()' , which returns a Background object.

The LightTheme and DarkTheme classes are concrete implementations of the BackgroundFactory class. Each class provides a different implementation of the CreateBackground() method, which creates a Background object that has a different color theme.

The Background class is a product of the BackgroundFactory class. It defines a set of properties and methods that allow clients to interact with the background object, such as getting and setting the background color.

Clients of the BackgroundFactory pattern use the factory method to create a Background object without having to know the implementation details of the factory or the product.

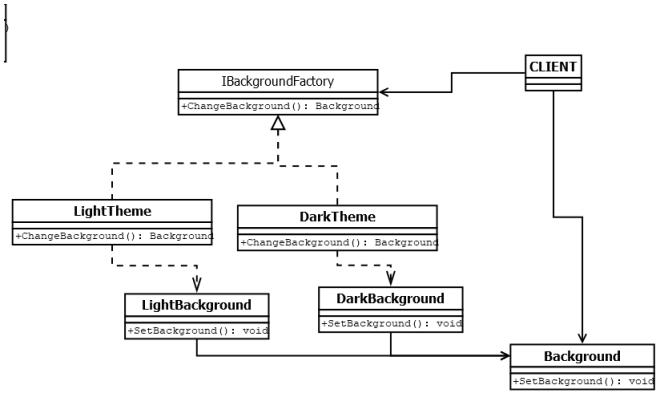


Figure 10. Possible implementation for a factory method that allows for us to easily change the theme of the UI.

The following UML is a Builder Design pattern see Figure 11.

The CharacterBuilder interface specifies the steps for building a character.

The GeneralCharacterBuilder class implements this interface and is responsible for building the character.

The CharacterDirector class instructs the GeneralCharacterBuilder on the steps to building a character. The GeneralCharacter object is the complex object, and the CharacterBuilder interface defines the steps required to construct it.

The GeneralCharacterBuilder class implements these steps in its methods.

The CharacterDirector class acts as a director, which looks over the construction process and can create a specific type of character by using a specific implementation of the CharacterBuilder interface.

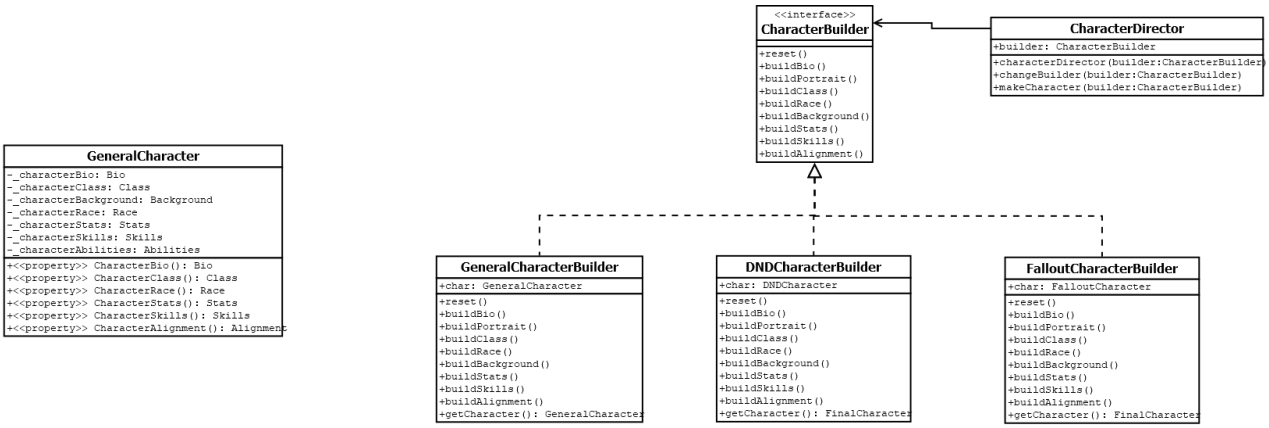


Figure 11. The Builder of the project, the generalCharacterBuilder creates a general character and through the director builds a generalCharacter with each of its variables being assigned through the builder.

## 4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

# 5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

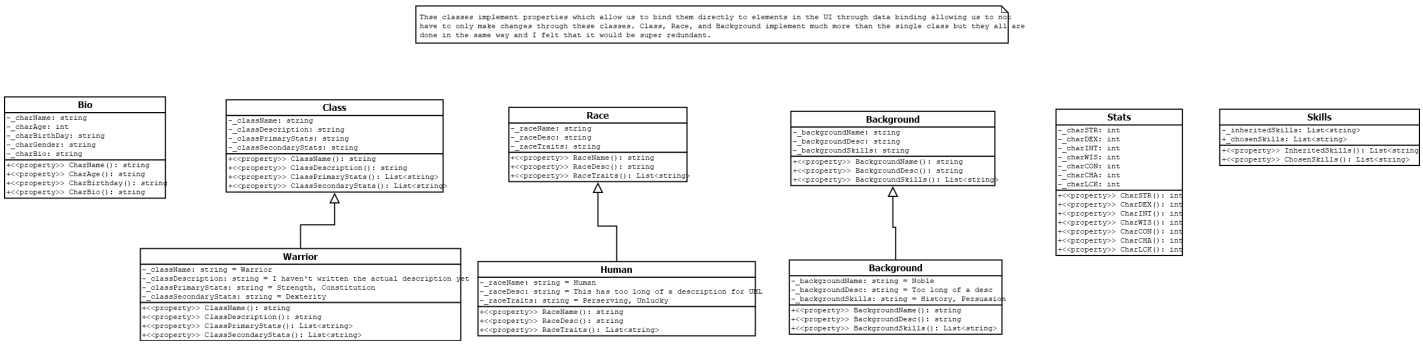


Figure 12. The generalCharacter is made of these classes, these classes store information and allow for us to easily update the UI through the use of data binding and properties.

## 5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

# References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.