GenesisNPC : NPC Generation and Management

ENSE 477 : Project

Date : April 12, 2017

Project Members : Christian Koback - 200324208

# Executive Summary

Most role playing games include non-player characters within them to provide the storyline to the game or exist for players’ entertainment to kill. The other systems currently available for use for character generation can only create one non-player character at a time and must be exported to an external file because once the webpage for the character generation changes, the created non-player character is inaccessible.

GenesisNPC is a new system which I have created for my fourth year project to generate several non-player characters at one time, the generated characters can have a probability of being based off a specific template, and templates which non-player characters are based off are highly flexible.

There have been several documents created over the course of the project, which include a Requirements Document to state what the system is to complete and check the feasibility of the ship, System Design Document which describes the architecture and decomposition of the project, an Object Design Document where trade-offs within the system were created and the file organization and documentation of the project, and a Test Case Document which describes the most important manual tests that the system must be able to handle.

The documents mentioned have been updated to fit the end result, which was primarily due to the navigation and design of the project has changed to improve the user experience and to prevent an information overload for the user. Some difficulties that occurred during the project was due to amount of work necessary to complete it which occurred primarily because there is no automated method to test dynamic creation of html elements, and there are minor differences between web browsers which change the style I must use for the webpages.

Fortunately, the project has been completed and GenesisNPC is now open for public use. The final features include allowing users create a customizable character template, can generate multiple characters from the character template, and can use multiple templates for character generation by using a probability of template use for character generation. Alternatively, there are plenty of ideas to develop for future users which include adding a table to the template creation page to increase the similarity of it to a Dungeons&Dragons character sheet.

Table of Contents

[Executive Summary i](#_Toc479788844)

[Table of Contents **Error! Bookmark not defined.**](#_Toc479788845)

[Table of Figures iv](#_Toc479788846)

[1. Introduction 1](#_Toc479788847)

[1.1 Background 1](#_Toc479788848)

[1.2 Other System Functionality 1](#_Toc479788849)

[1.3 Current System Functionality 1](#_Toc479788850)

[2. Documents 2](#_Toc479788851)

[2.1 Requirements Document 2](#_Toc479788852)

[2.1.1 Success Criteria 2](#_Toc479788853)

[2.1.2 Functional Requirements 2](#_Toc479788854)

[2.1.3 Non-Functional Requirements 3](#_Toc479788855)

[2.2 System Design Document 11](#_Toc479788856)

[2.2.1 Software Architecture 11](#_Toc479788857)

[2.2.1.1 System Decomposition 12](#_Toc479788858)

[2.2.1.2 Persistent Data Management 12](#_Toc479788859)

[2.3 Object Design Document 13](#_Toc479788860)

[2.3.1 Object Design Trade-Offs 13](#_Toc479788861)

[2.3.2 Interface Documentation Guidelines 14](#_Toc479788862)

[2.3.3 Packages 17](#_Toc479788863)

[2.4 Test Case Document 18](#_Toc479788864)

[2.4.1 Approach 18](#_Toc479788865)

[2.4.2 Test Cases 19](#_Toc479788866)

[3. Business Analysis 29](#_Toc479788867)

[3.1 Software Maintenance 29](#_Toc479788868)

[3.2 Future Developments 29](#_Toc479788869)

[4. Reflection 30](#_Toc479788870)

[5. Conclusion 30](#_Toc479788871)

# Table of Figures

[Figure 1: Overall use case diagram. 5](#_Toc479788952)

[Figure 2: Login and Main Webpage sequence diagram. 8](#_Toc479788953)

[Figure 3: Design for Login page. 9](#_Toc479788954)

[Figure 4: Template Creation webpage design. 10](#_Toc479788955)

[Figure 5: Main Page Design. 10](#_Toc479788956)

[Figure 6: System organization with Roles. 11](#_Toc479788957)

[Figure 7: Structure of files dealing for each webpage within the package. 17](#_Toc479788958)

# Introduction

This document provides insight into the GenesisNPC project which I have completed for a 4th year project. In addition, this section describes the need for the system and the functionality of the system.

## Background

In most games, there exist special characters called NPCs which stands for non-player characters. These characters within games tend to not be controlled by a specific player, but exist within the game to provide storyline or enemies to a game to increase the fun for players.

## Other System Functionality

Currently, other systems require the user to generate NPCs based on specific templates which are created by the developer of the system which forces users to use a different system for every different game the user plays. In addition, only one NPC can be generated at a time and must be exported to a document for the user to keep the generated NPC for later use.

## Current System Functionality

The GenesisNPC system allows users to create customized templates where users can base their generated characters from. In addition, users can generate multiple characters at one time and can view characters at later dates.

# Documents

This section discusses the important and unique sections mentioned in each document used in the development of this system with updated information. These documents include the Requirements Document, the System Design Document, the Object Design Document, and the Test Case Document.

## Requirements Document

This document describes the requirements needed for this system.

### Success Criteria

The success criteria for this system are for the user to be able to generate a large amount of NPCs using one or more templates. In addition, an individual who generates an NPC must be able to share the characters with other individuals.

### Functional Requirements

These are requirements on how the proposed system should work.

A user can:

* create a character template for NPC generation
* save a created character template for later use
* generate one NPC from one character template
* generate several NPCs from one character template
* generate one NPC from several character templates
* generate several NPCs from several character templates
* modify the generated character’s level and changed character statistics are updated & saved
* share their created NPCs with other users/individuals to view
* view/use their previously created NPCs

### Non-Functional Requirements

#### Usability

* All currently generated characters should be able to be viewed quickly
* All previously generated characters should load and be viewed quickly
* Reliability
* Needs to become inaccessible less than once a week.

#### Interface

* Will be a web application

#### Legal

* Will be using open source software
* Will be managed by the web domain developer

#### System Models

**Scenarios**

**Scenario 1:** User generates a NPC

* User creates a character template
* User generates a character based on template

Optional: save generated character

**Scenario 2:** User generates several NPC

* User either creates one or several character templates
* User generates several characters based on given templates

Default: generate characters based on 100% / #of characters to generate

Optional: save generated characters

**Scenario 3:** User generates several NPC with additional area %chance

* User creates several character templates that have an area probability
* User generates several characters based on given templates & area probability

Optional: save generated characters

**Scenario 4:** User views previous generated NPC

* User logins into account
* User navigates/finds character they want to view
* User opens the character display and views character

**Scenario 5:** User sends NPC to other individual to view.

* User logins into account
* User navigates/finds character they want to view
* User sends character ID to other individual to view character with

#### Use case model

The functionality possible by various user role models is shown in Figure 1. User 1 in Figure 1 is a user with an account and User 2 is a guest user.

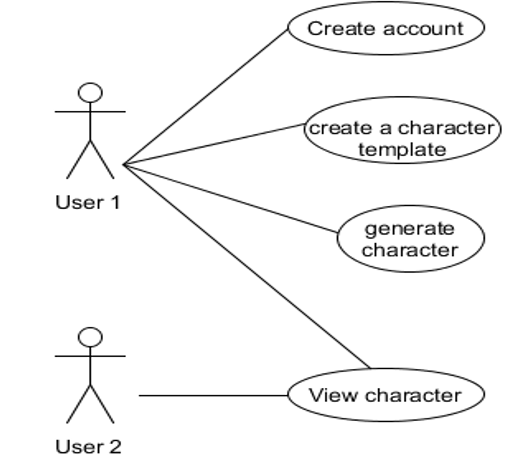


Figure 1: Overall use case diagram.

|  |  |
| --- | --- |
| Use Case Element | Description |
| Use Case Number | 1 |
| Use Case Name | Create Account |
| Use Case Description | The user creates an account to link to character template and characters within the database. |
| Primary Actor | User |
| Precondition | User owns an email for account confirmation. |
| Trigger | User clicks on ‘register’ button on login |
| Basic Flow | User gets to login screen, clicks on ‘register’, enters email& login info, opens email to verify their account |
| Alternate Flows | User logins with valid login info, Invalid login for user, user doesn’t validate their account within time period. |

|  |  |
| --- | --- |
| Use Case Element | Description |
| Use Case Number | 2 |
| Use Case Name | Create Character Template |
| Use Case Description | The user creates a character template with which they generate a NPC with. |
| Primary Actor | User |
| Precondition | User has already logged into their account, user is using temporary permissions (temp user). |
| Trigger | User clicks on ‘create character template’. |
| Basic Flow | User clicks on button, user enters all template parameters and values and saves to database. |
| Alternate Flows | User logins with valid login info, Invalid login for user, user doesn’t validate their account within time period. |

|  |  |
| --- | --- |
| Use Case Element | Description |
| Use Case Number | 3 |
| Use Case Name | Generate Character |
| Use Case Description | The user generates a character using the saved character template. |
| Primary Actor | User |
| Precondition | User has already logged into their account, user is using temporary permissions (temp user). Character template must be generated. |
| Trigger | User clicks ‘create character’. |
| Basic Flow | User chooses character template, user clicks ‘generate’ and generates character. |
| Alternate Flows | User chooses one template and generates multiple characters, User chooses multiple templates and generates multiple characters. |

|  |  |
| --- | --- |
| Use Case Element | Description |
| Use Case Number | 4 |
| Use Case Name | View character |
| Use Case Description | User 1 logs into account, User 1 finds saved character to share to User 2, User 1 sends character specific ID to User 2, User 2 views character. |
| Primary Actor | User |
| Precondition | User 1 is logged into an account, User 2 has access to internet. |
| Trigger | User1 clicks on ‘view character’. |
| Basic Flow | User 1 navigates to character, user 1 gets special character ID, user 1 sends the character ID to user2. |

#### Dynamic models

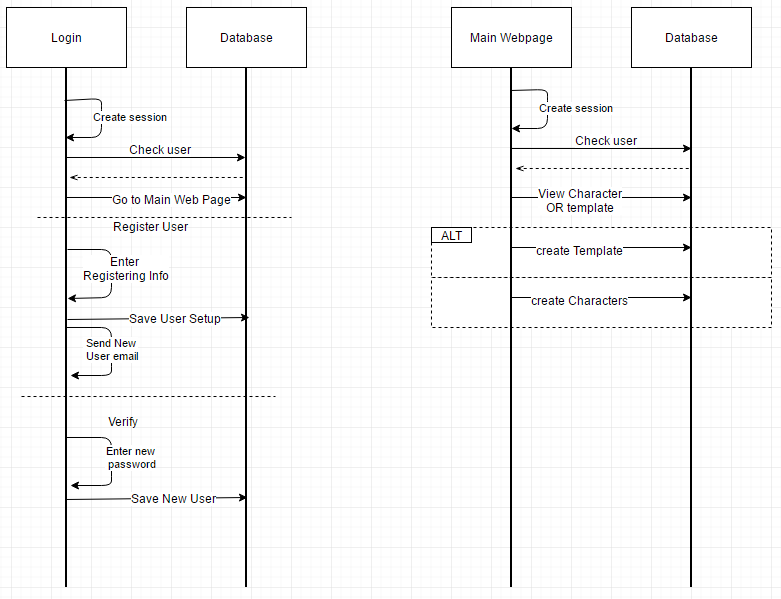


Figure : Login and Main Webpage sequence diagram.

The flows contained within the login and main webpage are shown in Figure 2.

#### User interface – Navigational Paths and Screen Mock-ups

The login design for the system can be found in Figure 3, which is the first page the

user is shown when they use the website and the design for the template creation page is shown in Figure 4. These pages I have dictated as the most important webpages because the majority of the functionality are on these pages.

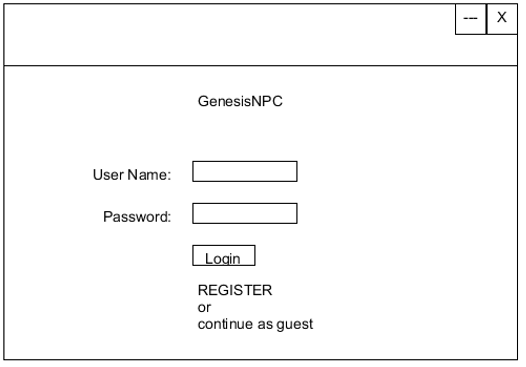


Figure : Design for Login page.

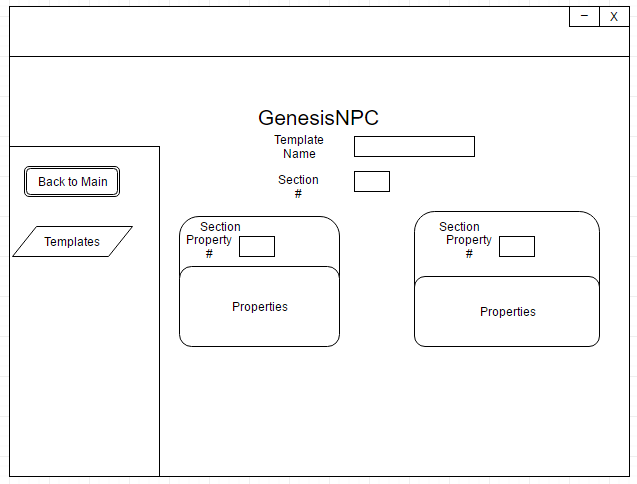
 

Figure : Template Creation webpage design.

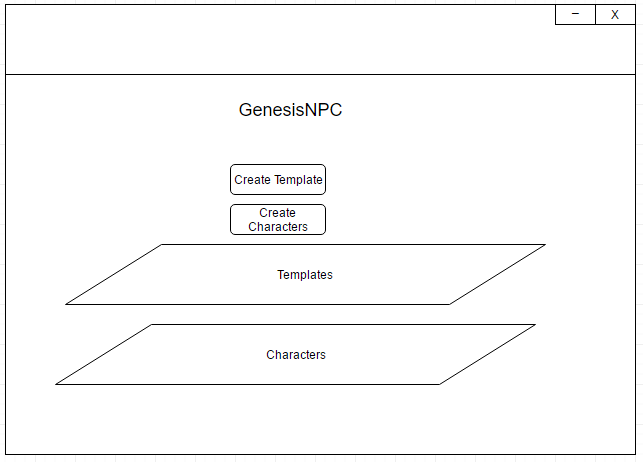


Figure : Main Page Design.

## System Design Document

This section describes how the GenesisNPC system is organized and its architecture.

### Software Architecture

As mentioned in the Introduction, there are two user roles which are:

* 1. Registered User
  2. Guest User

Both of these roles have various functionalities and therefore have slightly different workflows; however, the organization of code for both roles is the same. The organization of the files and of the system is broken down into 3 main sections which are client, server, and database as seen in Figure 6.

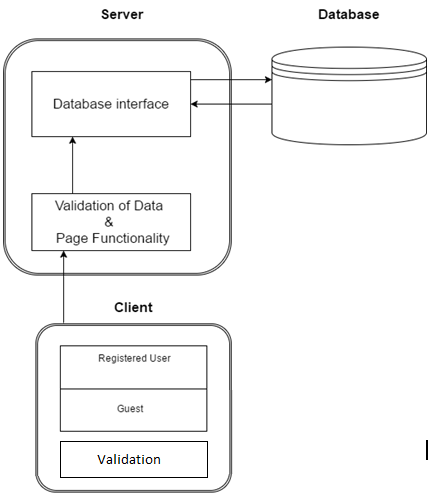


Figure : System organization with Roles.

### System Decomposition

The client is responsible for verifying if the user's input is valid and modifies the display for a more enjoyable and less overwhelming experience when entering information into the system. This also involves implementing the majority of the styling of the web pages, interactive input elements, and the handling of website events.

The server keeps track of the user's session, and whether the user is a registered user or a guest user. All database connections are also primarily handled by the server after secondary information checks are completed, as well as completing additional computing to allow less stress on the client's machine and increase the security of data. In addition, the server ensures all information entering the database is unique which ensures all data is unique for each individual registered user, and for each guest that uses the system.

### Persistent Data Management

This is a major area for this project since there will be a large amount of data getting saved by various users. For this reason, there will be no saved materials for any guest users and any guest user information will be removed monthly. At this time I have not decided on a life span for registered user information.

## Object Design Document

This section describes the structure of the GenesisNPC system in greater detail. Specifically, trade-offs, documentation of the system, and file organization of the system.

### Object Design Trade-Offs

There are multiple trade-offs considered in this project, and the majority of the trade-offs deal with increasing the user experience of using the system. Some of these trade-offs include flexibility of user input for template creation vs input validation, ease of character viewing vs crowding of windows, ease of other user’s character viewing vs security.

**User Input Flexibility vs Template Automation**

Since the scope of use for this project is for generic use among games, this project has given flexibility of character templates a higher priority than automated template creation. The flexibility allows this project to be usable in a large amount of situations.

**Character Viewing Pop-ups vs Character Viewing In-page**

This system uses pop-ups to provide increased ease to switch between large amount of characters’ statuses and information. Viewing the characters in-page within the system only supports viewing one character at a time and I have chosen pop-ups to display characters because multiple characters can be shown at once.

**Viewing Other User’s Characters by:**

**User & Character Name vs Unique Code from Creator**

This system uses the functionality of the visitor using a specific user’s name and the user’s character or template name to view the user’s specific character or template because it reduces the complexity of the system. However, a feasible alternative is to generate a unique identification number per character and include permissions of who can view what user characters.

### Interface Documentation Guidelines

These documentation guidelines are provided and should be followed to ensure developers to become easily intimate with the system. The commenting process is the same within all php, javascript, and css files.

#### File Header Comments

The documentation for file headers should contain a summary of what functionality the file contains, what the name of the file is, what project the file is included in, and name of primary developer. Use /\* to start the comment and \*\ to end the comment.

/\*

programmer ---> Christian Koback

program ---> Genesis NPC : NPC Generator & Management

Date started : September 2016

Date stopped development: April 2017

Purpose : Capstone (4th year project)

file: characterDisplay.php

file description: ---> Show list of created templates and characters

---> provide a central location for user to access template and character creation

\*/

#### Function Comments

The function comments should be used if the function name is not descriptive enough about what the objective of the function is. It is also recommended to include function comments if the function is complex, have a large amount of parameters, or there are very specific preconditions needed for the function. Use /\* to start the comment and \*\ to end the comment.

/\* changes user's account to active, changes the user's password, sets the verification info to 0 for no re-use \*/

function setupUser($email, $hash, $pass, $passSalt){

…

}

#### Inline Comments

Inline comments are not recommended to be used but are necessary if there is a complex function call or complex functionality within a function. Use // to comment out the entire line for the necessary inline comment.

//handle redirecting to viewing page if template/character name is clicked

if (isset( $\_POST["viewTemplate"]) && !empty($\_POST["viewTemplate"])

{

blah;

}

#### Coding Conventions

The coding conventions used in this project are to be followed for easy readability.

Indentation: 4 spaces   
Whitespace:

* one whitespace between commas and additional parameters
* one whitespace on both sides of operators
* all conditional loops are to have braces around it’s body code
* all conditional statements are to be used within parentheses

All variables and functions are to be camel-cased where the first letter of the first word is lowercase and all additional words in the variable name have the first letter capitalized.

### Packages

The project is of a monolithic structure where there is one main package where all the code in the system lives. However within the package, the files are organized in separate folders of php files, css files, and javascript files for easy management. The files within the package for each webpage is as shown in Figure 7.

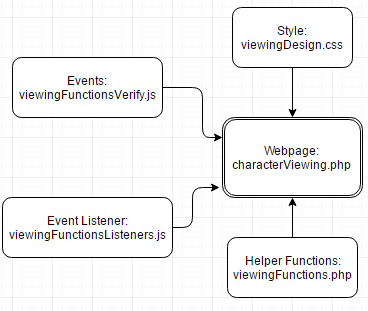


Figure : Structure of files dealing for each webpage within the package.

Each php file refers to either a webpage or contains helper functions required by a webpage. In all cases the name of the file should be descriptive enough to give the user an idea about what the contents are to be used for, and any php files containing helper functions and associated javascript and css files should contain names which logically relate them to the webpage where the functionality is used. The helper function name typically contains part of the caller webpage’s file name. All file names should be camel-cased with the first letter of the first word lowercase.

The general behaviour and validation of the webpage should be handled within the associated javascript files where the php files handle the display of information to the user and handling any database functionality.

## Test Case Document

This document describes the approach taken to tests and the test cases which are created for the GenesisNPC system.

### Approach

The approach I have taken to testing for this system is unit testing javascript and php calculations, while testing of javascript and php generated html is tested manually through a web portal.

### Test Cases

#### Login - Guest

**Description:** User does not need a valid account to use the system.

**Preconditions:** none

**Postconditions:** User is logged in as a guest profile.

**Flow of Events:**

* User navigates to website.
* Clicks on “login as guest”.
* User is successfully logged in.

#### Login - Creator

**Description:** Guest can view complete template.

**Preconditions:** none

**Postconditions:** User is now logged in and is known as Creator.

**Flow of Events:**

* User navigates to website.
* User enters correct username and password.
* Ensure both are case sensitive.
* Ensure only letters and numbers are allowed for both.
* User is redirected to the main application page. Login is successful.

#### Register

**Description**: User can register and enter as Creator.

**Preconditions:** none

**Postconditions:** User is registered as a valid account.

**Flow of Events:**

* User navigates to website.
* User enters in personal information.
* User receives notice via email.
* User validates their account.
* User can now log in to their account.

#### Create a General Template

**Description:** Creator can create a new customized template.

**Preconditions:** User is logged in.

**Postconditions:** Template is created and added to the database.

**Flow of Events:**

* User navigates to template creation page.
* User ensures template type is general.
* User ensures all fields are blank.
* User enters all necessary information.
* Ensure no template/section/property names are missing.
* Ensure all associated property values are missing.
* User saves the template.

#### Create a Pathfinder Template

**Description:** Creator can create a new pathfinder template with default pathfinder categories.

**Preconditions:** User is logged in.

**Postconditions:** Pre-saved information is filled out in the various templates fields. Template is created and added to the database.

**Flow of Events:**

* User navigates to template creation page.
* User ensures template type is pathfinder.
* User ensures all necessary section/property names are filled and correct amount of fields are shown.
* User enters all necessary information.
* Ensure no template/section/property names are missing.
* Ensure all associated property values are missing.
* User saves the template.

#### Create a Dungeons&Dragons Template

**Description:** Creator can create a new dungeons&dragons template with default dungeons&dragons categories.

**Preconditions:** User is logged in.

**Postconditions:** Pre-saved information is filled out in the various templates fields. Template is created and added to the database.

**Flow of Events:**

* User navigates to template creation page
* User ensures template type is pathfinder.
* User ensures all necessary section/property names are filled and correct amount of fields are shown.
* User enters all necessary information
* Ensure no template/section/property names are missing
* Ensure all associated property values are missing
* User saves the template

#### View Template

**Description:** Creator can view complete template after creating a template.

**Preconditions:** User is logged in. User has created a template.

**Postconditions:** All fields displayed hold valid values. All information is shown.

**Flow of Events:**

* User navigates to webpage with template names displayed in a list.
* Main webpage and character creation page.
* User clicks on template name and template pops up for display.

#### Create Character – Single template, Single character

**Description:** Creator can create a single character with a single template.

**Preconditions:** User is logged in. User has created a template.

**Postconditions:** One character is created and saved into database.

**Flow of Events:**

* User navigates character creation webpage.
* User chooses 1 template to use.
* User chooses template from template list to the left.
* User chooses to create 1 character.
* User ensures they generated 1 character.

#### Create Character - Multiple templates, Single character

**Description:** Creator can create a single character with a chance of it being based off one of multiple templates.

**Preconditions:** User is logged in. User has created a template.

**Postconditions:** One character is created and saved into database.

**Flow of Events:**

* User navigates character creation webpage.
* User chooses 2 or more template to use.
* User chooses templates from template list to the left.
* User chooses to create 1 character.
* User ensures they generated 1 character.

#### Create Character - Multiple templates, Multiple characters

**Description:** Creator can create a multiple characters with a chance of each character being based off one of multiple templates.

**Preconditions:** User is logged in. User has created a template.

**Postconditions:** Multiple characters are created and saved into database.

**Flow of Events:**

* User navigates character creation webpage.
* User chooses 2 or more template to use.
* User chooses templates from template list to the left.
* User chooses to create 2 or more characters.
* User ensures they generated the correct amount of characters.

#### Create Character - Single template, Multiple characters

**Description:** Creator can create a multiple characters with all characters based off one template.

**Preconditions:** User is logged in. User has created a template.

**Postconditions:** Multiple characters are created and saved into database.

**Flow of Events:**

* User navigates character creation webpage.
* User chooses 1 template to use.
* User chooses template from template list to the left.
* User chooses to create 2 or more characters.
* User ensures they generated the correct amount of characters.

#### View Character - Creator

**Description:** Creator can view complete character after creating a character.

**Preconditions:** User is logged in. User has created a character.

**Postconditions:** User can see all character data.

**Flow of Events:**

* User navigates to main page.
* User chooses to view a character.
* User enters in the character ID and chooses enter.
* User is given chance to edit character

#### View Character - Guest

**Description:** Guest can view complete character.

**Preconditions:** User is logged in. User has a different user’s character ID.

**Postconditions:** User can see all character data.

**Flow of Events:**

* User navigates to main page.
* User chooses to view a character.
* User enters in the character ID and chooses enter.
* User is not given chance to edit character

#### Edit Character - Creator

**Description:** Creator can edit a character they created.

**Preconditions:** User is logged in. User has created a character.

**Postconditions:** edited fields are saved in the database.

**Flow of Events:**

* User navigates to display character page.
* User enables editing.
* User edits various fields and saves changes.

#### View Template- Guest

**Description:** Guest can view complete template.

**Preconditions:** User is logged in. User has a different user’s template ID.

**Postconditions:** User can see all template data.

**Flow of Events:**

* User navigates to main page.
* User chooses to view a template.
* User enters in the template ID and chooses enter.
* User is not given chance to edit template.

# Business Analysis

This section describes some marketing aspects of the GenesisNPC system which includes the maintenance and the future developments that may occur for the GenesisNPC system.

## Software Maintenance

Since this system relies largely on databases and database memory, one part of the maintenance will deal with memory. In an effort to reduce the memory usage, all guest-role character templates and characters are deleted immediately after the guest user logs out of the system and all guest templates and characters are to be cleaned out of the database every once a month to handle the situations where guests do not press the logout button.

There is no specific maintenance in place for templates and characters created by users with user accounts. The reason behind this is to allow users with accounts larger flexibility with character and template storage.

## Future Developments

Thanks to user feedback and marketing, several features that may be added are as follows:

* add exporting functionality to characters and templates
* add table to template creation to allow greater similarities to Dungeons&Dragons character sheets
* add post-template functionality for changing pre-generated characters easily
* add basic template for easy template creation for various games like Pathfinder and Dungeons&Dragons
* add extra functionality so templates and characters can be viewed by others more securely

# Reflection

Some things I have learned over the period of this project are that even a simple website can become extremely difficult and a team or group of people may have made the project easier to complete and implement features and time management and estimating amount of work that is doable within a certain amount of time is extremely important for the completion of a project on time. In addition, all dynamic creation of html documents and dynamic use of websites are extremely time-consuming because there are no testing tools.

# Conclusion

GenesisNPC can generate multiple characters at once and the template creation is highly dynamic to provide the highest amount of user experience to the users of the system, and the templates and characters for users with accounts are saved for future use. In addition, over the course of the project I have created a requirements document which describes the requirements of the system, system design document which describes the architecture of the system, object design document which describe the documentation and trade-offs within the project.

There are plenty of future developments which can be added to make it more attractive to users who play games like Dungeons&Dragons or Pathfinder, like adding exporting functionality to templates and characters and adding more functionality for templates to ensure better customization of characters.