**Software Requirements Specification:**

**Shattered Rails**

**Version 1.0**

**02/12/24**

Ankith Goutham

Mahin Haque

Jayanth Nama

Lilac Sabri

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 02/12/2024 | Version 1.0 | Ankith Goutham  Mahin Haque  Jayanth Nama  Lilac Sabri | First iteration for the SRS document. |
| 2/19/2024 | Version 1.1 | Ankith Goutham  Mahin Haque  Jayanth Nama  Lilac Sabri | Make corrections based on instructor and GTA feedback. |
| 04/13/2024 | Version 1.2 | Mahin Haque | Add additional information regarding the scope of the 2nd game, add new user interface images, and new constraints. |
|  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
| AnkithG | Ankith Goutham | Team Lead | 02/12/24 |
| MahinH | Mahin Haque | Backend Lead/Documentation Lead | 02/12/24 |
| JayanthN | Jayanth Nama | Presentation Lead/ Frontend Lead | 02/12/24 |
| LilacS | Lilac Sabri | Presentation Lead/Front-end Lead | 2/12/24 |

**Table of Contents**

[**Revision History** 2](#_Toc159253691)

[**Document Approval** 2](#_Toc159253692)

[**1. Introduction** 5](#_Toc159253693)

[**1.1 Purpose** 5](#_Toc159253694)

[**1.2 Scope** 5](#_Toc159253695)

[**1.3 Definitions, Acronyms, and Abbreviations** 5](#_Toc159253696)

[**1.4 References** 7](#_Toc159253697)

[**1.5 Overview** 7](#_Toc159253698)

[**2. General Description** 8](#_Toc159253699)

[**2.1 Product Perspective** 8](#_Toc159253700)

[**2.2 Product Functions** 9](#_Toc159253701)

[**2.3 User Characteristics** 9](#_Toc159253702)

[**2.4 General Constraints** 10](#_Toc159253703)

[**2.5 Assumptions and Dependencies** 10](#_Toc159253704)

[**3. Specific Requirements** 12](#_Toc159253705)

[**3.1 External Interface Requirements** 12](#_Toc159253706)

[**3.1.1 User Interfaces** 12](#_Toc159253707)

[**3.1.2 Hardware Interfaces** 15](#_Toc159253708)

[**3.1.3 Software Interfaces** 15](#_Toc159253709)

[**3.1.4 Communications Interfaces** 15](#_Toc159253710)

[**3.2 Functional Requirements** 15](#_Toc159253711)

[**3.2.1 FR1: Register** 15](#_Toc159253712)

[**3.2.2 FR2: Login** 17](#_Toc159253713)

[**3.2.3 FR3: Logout** 17](#_Toc159253714)

[**3.2.4 FR4: Home** 18](#_Toc159253715)

[**3.2.5 FR5: Path/Choice Selection** 19](#_Toc159253716)

[**3.2.6 FR6: Battle System** 19](#_Toc159253717)

[**3.2.7 FR7: Adjust Text Speed** 20](#_Toc159253718)

[**3.2.8 FR8: Pause Text** 21](#_Toc159253719)

[**3.2.89 FR9: Adjust Text Font Style** 22](#_Toc159253720)

[**3.2.10 FR10: Adjust Text Font Color** 22](#_Toc159253721)

[**3.2.11 FR11: Adjust Background Color** 23](#_Toc159253722)

[**3.2.12 FR12: Adjust Font Size Increase** 24](#_Toc159253723)

[**3.2.13 FR13: Adjust Font Size Decrease** 24](#_Toc159253724)

[**3.2.14 FR14: Text to Speech** 25](#_Toc159253725)

[**3.2.15 FR15: Profile** 26](#_Toc159253726)

[**3.2.16 FR16: Save** 27](#_Toc159253727)

[**3.2.17 FR17: Game Completion Bar** 27](#_Toc159253728)

[**3.2.18 FR18: Change User Information** 28](#_Toc159253729)

[**3.2.19 FR19: Exit Warning** 29](#_Toc159253730)

[**3.2.20 FR20: Email Verification** 30](#_Toc159253731)

[**3.2.21 FR21: Show Help** 31](#_Toc159253732)

[**3.2.22 FR22: User Password Recovery** 32](#_Toc159253733)

[**3.2.23 FR23: User Username Recovery** 33](#_Toc159253734)

[**3.3 Non-Functional Requirements** 34](#_Toc159253735)

[**3.3.1 Performance** 34](#_Toc159253736)

[**3.3.2 Reliability** 34](#_Toc159253737)

[**3.3.3 Availability** 34](#_Toc159253738)

[**3.3.4 Security** 34](#_Toc159253739)

[**3.3.5 Maintainability** 35](#_Toc159253740)

[**3.3.6 Portability** 35](#_Toc159253741)

[**4. Design Constraints** 35](#_Toc159253742)

[**4.1 Hardware Constraints** 35](#_Toc159253743)

[**4.2 Program Design Constraints** 36](#_Toc159253744)

[**4.3 UI Constraints** 36](#_Toc159253745)

[**4.4 Video Game Rating/Audience Constraints** 37](#_Toc159253746)

[**5. Logical Database Requirements** 37](#_Toc159253747)

[**6. Analysis Models** 38](#_Toc159253748)

[**6.1 Data Flow Diagram (DFD)** 38](#_Toc159253749)

# 

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to provide a comprehensive Software Requirements Specification for Shattered Rails. This document's contents will provide a clear and detailed understanding of all necessary features being implemented into the Shattered Rails application. It will provide a complete list of requirements regarding this application’s software design, implementation, and testing. The document will also include a specified list of design constraints, logical requirements for data storage capabilities, data retention, data integrity, etc. Additionally, it will also detail the flow of data within the entire system through a Data Flow Diagram (DFD). The intended audience for this document is for developers and individuals involved in application development.

## **1.2 Scope**

Shattered Rails will provide users with two text-based adventure games experiences where the first game allows users to make choices that will lead to different endings and the second offers expanded combat mechanics offering various boss battles. The first game includes branching storylines where the player is given 2 types of choices at each scenario, one being stealth-based, and the other being action-based with each choice leading to unique paths and endings. There will be 15 different endings categorized into good, bad, and secret endings. It will also offer a branching storyline system for users where the choices they make will affect the ending they will reach. Additionally, there will be a sequel to the game, where it will expand the universe offering a more dynamic combat system while expanding the story of the first game. This sequel will more of a battle-centric approach which offers the players the chance to test their tactical strategies as they must keep track of their health and magic points (MP) while using an expanded arsenal to combat enemies.

The first game will allow the user to select their difficulty choice in the beginning to dictate the difficulty of their playthrough. The user will play through the game making choices on how they want to proceed after the game provides context on the current situation. Some choices in the game can lead to a combat scenario against the enemy where they will fight in turns against the enemy. The user can attack or stay still and the turns will alternate between the user and the enemy until one of them is defeated. The user can customize their experience by modifying colors on the screen, changing text to the users, being able to adjust speed of text output, and the choice to use text to speech. Additionally, the game will offer a progress bar allowing the user to track the different endings they have already visited with the progress completion saving if they were to logout or visit another page and then return to the game. The game will also keep track of where you are in the story with the autosave feature allowing users to leave the page and come back to where they were in the path before they left. Images and sound effects will be used throughout the game to create a more engaging experience.

The second game in the beginning will allow the user to select their difficulty and their class that they want to play as. The user will then be given a set of five bosses to choose from and can defeat them in any order. The user will now be given new ways to fight the opponent such as attack, defend, stand still, spell, run and persuade. The spell attack runs on MP and each class has a limited amount with some having more than others. The user will be given the same UI customization tools that were provided in the first game.

Additionally, the project will provide registration and login functionality. The user can also use the forget name and forget password features to retrieve the credentials they may have forgotten. After the user registers, they will be prompted to verify their email address before they can login to the application. The user can also view their profile where they can edit their personal information and change their password. A help page will be provided to the user to learn more about how to play the games and more about the text-based adventure genre.

This project will not include 3D graphical gameplay and will rely on text as the main form of gameplay. In addition, the game will contain no background music during gameplay.

## **1.3 Definitions, Acronyms, and Abbreviations**

|  |  |  |
| --- | --- | --- |
| **Term** | **Abbreviation** | **Definition** |
| Django | N/A | Django is a python framework that makes creating websites using python a lot simpler. It comes with features such as a login system, database connection, etc. |
| Hyper Text Transfer Protocol | HTTP | Communication between client computers and web servers is done by sending HTTP requests and receiving HTTP responses. |
| Software Requirements Specification | SRS | SRS is a comprehensive document that contains essential elements that provide a clear and detailed understanding of the project’s objectives, scope, user characteristics, functional, and non-functional requirements, design constraints, and more. It adheres to principles such as correctness, traceability, verifiability, and prioritization. |
| Text Based Adventure | N/A | It is an online game that uses a text-based user interface, where the worlds are described in a narrative format and the player submits simple commands to interact with the worlds. |
| Application Programming Interface | API | API is the way that computer systems or components communicate with each other. A Web API can extend the functionality of the browser, simplify complex functions, and provide easy syntax to complex code, meaning it is an application programming interface for the web. A Browser API can extend the functionality of a web browser, and a server API can extend the functionality of a web server. |
| SQLite | N/A | Embedded, server-less relational database management system. |
| Data Flow Diagram | DFD | DFDs are a way of representing a flow of data through a system or a processor. It provides the inputs and outputs of each entity and the process. |
| Magic Points | MP | MP is a resource often used in a fantasy game to control the amount of magic and spells a person can cast. If a person runs out of MP they can no longer cast magic. |

## **1.4 References**

* Google Chrome Hardware Requirements: <https://support.google.com/chrome/a/answer/7100626?hl=en#:~:text=Windows,or%20later%20that's%20SSE3%20capable>
* Safari Hardware Requirements:

<https://support.apple.com/kb/sp552?locale=en_US>

* Django Documentation:

<https://docs.djangoproject.com/en/5.0/>

* ESRB Ratings:

<https://www.esrb.org/ratings-guide/>

* Definitions:

[*https://www.w3schools.com/*](https://www.w3schools.com/)

* Web Technology Documentation
  + <https://developer.mozilla.org/en-US/docs/Web/HTML>
  + <https://developer.mozilla.org/en-US/docs/Web/CSS>
  + <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

## **1.5 Overview**

The Software Specification Requirements document covers details essential for the software application development, beginning with general factors that provide context through purpose, goals, user characteristics, constraints, assumptions, and dependencies. Interface requirements are also part of the document, including all forms of external interfaces the software will interact with, such as user interfaces, hardware, software, and communication interfaces, along with their respective protocols, formats, and standards. Functional requirements are outlined to describe the software's intended functionalities and features, encompassing input data, processing logic, output results, error handling, dependencies, and prioritization. Non-functional requirements focus on the software's quality aspects, including performance, reliability, availability, security, maintainability, and portability, setting the criteria and constraints for achieving these attributes. Design constraints specifically address limitations in the software's design and front-end development. Database requirements are detailed for data storage, retrieval, management, and integrity. Lastly, analysis models, particularly a data flow diagram, are included in the document to demonstrate the understanding of the data flow within this software application and its interaction with the user.

# **2. General Description**

Shattered Rails is a web-based application that offers two text-based adventure games. The user is taken through a sign in process to enter the game’s page, from which the user can select to play one of the two games. The user creates an account so that the saved data can be linked to the user’s account. The games are text based as mentioned before, so the user will read whatever text is being output by the game about the story, and then in certain parts of the game choices will be given to the user to advance the story in different ways.

For the first game, the user will be given multiple buttons on the screen which correspond to the choices that the user can take in the story, and when clicking a button, the game will go down that path. Paths go down to individual endings. There are 16 endings to this game. There are 2 types of decisions users can pick, one that is “stealth” based and one that is “action” based. The stealth-based decisions output text to further the story. The action-based decisions result in a battle system with enemy AI to win and progress the story. The action-based decisions are recommended for players who have experience with the text-based adventure genre and avid enjoyers of the field of text-based adventures, whilst the stealth-based choices are recommended for newcomers. The endings are split into 3 types: good, bad, and special endings, based on the user’s choices throughout the game. The game gives the user a tracker on how many endings the user has already seen and how many new endings that the user still must see when replaying. Completing the game fully allows for 100% completion meaning all endings have been viewed.

In the second game, the user will be given a choice in selecting their difficulty and what class they want to play as with each class having its own unique benefits and drawbacks. They will then be able to fight a boss from a set of five bosses and can fight them in any order as many times as the user wants.

## **2.1 Product Perspective**

This game is independent and self-contained, as it is an original entity/IP. There is no relationship between this game and others, meaning that it does not take place in a larger system, the system ends with it. This game juxtaposed to others in its genre is unique due to its story/setting, and its gameplay including battle elements.

One thing that Shattered Rails has that other games that are text-based adventures don’t have is the stealth/action-based choice system. Most text-based adventures are point and click games meaning the user points to parts of the screen and sees what that part has to say. Along with this, most text-based adventures don’t have any other forms of interaction with the story and characters as its simple button presses for the most part, with it ending in a linear one ending path. This game, however, has the features of choosing whether to act on the choices in a way that offers challenges with the enemy battle system, allowing for a more engaging and interactive experience rather than just a continuous journey of inputting a command and waiting for a response. With our text-based adventure, we also have many options that are helpful for the user such as text font/size/speed/color or even text to speech customization. Most games in this field do have customization, but not to the extent that Shattered Rails has because it is all customizable to the user, allowing the user to customize their experience to their specific needs.

## **2.2 Product Functions**

Shattered Rails has multiple functions at its disposure. The game is designed to allow the user the ability to choose between choices in any given scenario. This goes to a designated path towards an ending, giving the user the ability to experience different outcomes in the story. The game will also keep track of the already existing endings the user has seen and display the percentage of the game completed. The game will also track the save state of the user in the background allowing users to leave the page and return to where they left off. In addition, the game also has a dynamically changing background based on what background the story is set in, so that the user can be visually immersed in the story. Having the stealth/action-based system in play, the user can battle enemies in combat scenarios. Both games will offer user interface customization to give the user the tools to personalize the experience to their needs.

## **2.3 User Characteristics**

Shattered Rails is intended for users that are not exposed to the field of text-based adventures. This is intended as a beginner/easy game as an entry point into this field, so if users like this game, they can go on and play other text-based adventures since they have some experience from Shattered Rails. However, players who have played text-based adventure games in the past can also play this game since we offer a more challenging way to play the game through battle system mechanics. The user age range would be around the age of a college student so people in their early to late 20’s as this game uses language that is more suitable to people of this generation. There will also be violence in the game, but it won’t be too extreme, because of this the game will have a mature rating not allowing people under 17 to experience the game.

Since the application requires only clicking on a mouse to interact with buttons, anyone who has the physical ability to do so can play the game. We have accommodated some features to include people that may be disabled or have challenges to play the game. Some of these features include a text to speech option, allowing the user to choose colors/fonts/sizes of the text, and zooming in/zooming out option on the screen. The basic game without any further adjustments is perfectly playable by those that are physically abled, but there are players of video games that are color-blind hence why we give the option to choose the color of text fonts/backgrounds/terminal etc. There are also gamers that are more visually challenged meaning they need bigger fonts, hence why we added the option to increase font size. Some people cannot read fast as well, which is why the option of text speed was added into the game. Some people would like to hear the dialogue instead of reading text, as reading text might be too strenuous to the eyes, so that is why the feature of text to speech was added to the game in order to satisfy those customers. Users should be knowledgeable on how to access webpages so they can click on the buttons and access all the pages of the game. For example, a user that is 80 years old and doesn’t know how to navigate through a web page by clicking links/buttons would not be in the scope of the intended user, as they wouldn’t be able to even start the game. The user should also be able to read at an 8th grade literacy level as the language of the game is simple English.

## **2.4 General Constraints**

When designing the system, it is paramount to think about the boundaries of the game, as it is a game that does not have multiple users playing at the same time, it doesn’t need to be very intensive on computer hardware. Since the game also has significant amounts of dialogue, one constraint is that chunks of text must be separatable for text to have white space for the user’s eyes to easily read said text. The formatting of text and information is something that must be paid extra close attention to so that there is an easy path for the eyes to go down to read the story without extra eye strain. The game code design should be modular and in components so if one component breaks the whole application will not. It also will allow for new features to be added easily without negatively affecting other components.

Another constraint for this project will be budget constraints as this project is developed by college students with no budget allocated for the project. This can hinder development due to the reliance on free open source software when a paid alternative could be better.

There is also the aspect of schedule constraints, 3-4 months’ worth of time to work on a game result in features that cannot be added due to lack of time. Usually, text-based adventure games take at minimum 6-12 months and can go further depending on scope and ambition. Comparing Shattered Rail’s development time to these other full-fledged game schedules makes clear that development of game filters out a lot of features that could be possibly added to make the game more complex.

In addition to the time constraint, there is also a development constraint in that there are only four developers working on this project with limited experience with the tools in this project. The developers must learn the tools alongside developing new features which can slow development and limit the scope of the project.

Another aspect of Shattered Rail’s production that is a constraint, is that the game is a beginner/entry level game, meaning that the difficulty has to be adjusted in order to make sure that it is suitable for general/easy audiences. There should also be extra difficulty added for the enemy AI battle system so that it is catered towards veterans and people who crave more difficult scenarios. This difficulty is something that is a constraint that requires delicate attention and care so it can cater to both audiences that are in mind.

## **2.5 Assumptions and Dependencies**

This project assumes that the user is using one of 2 primary software operating systems: Windows and Mac. Since these are the 2 most popular and widely used operating systems in the world, they have been used as the basis for building the game. Other systems such as Linux can also support this game but it has not been tested on such software. This game also assumes you have internet access as that is needed for the game to connect to the database which saves user information. Another assumption is that the user is using browsers such as Google Chrome, Safari, Brave, Microsoft Edge, etc. As long as the hardware being used can access all of these browsers and can function such, the game can be successfully played without any errors, as the game is built on HTML/JavaScript/CSS which are not linked to any one browser.

As we listed prior, this game’s hardware specifications are the same that are used for Safari and Google Chrome’s browser usage. This game assumes that the user’s device is advanced enough to meet these criteria. If the user does not meet these criteria, the game will be unplayable. The project also assumes that the user’s internet connection is stable and strong enough to sustain website usage, so that the user can access the game, it’s account options, and also that the user can play the game for a long period of time should they wish to do so. If the user’s internet connection does not satisfy the criteria, not only can the user not play the game, they wouldn’t be able to access the website in the first place, and accordingly, they wouldn’t be able to register/login.

On that note, Shattered Rails assumes that the user has already registered an account and subsequently logged in, as in order to get to the game’s page, one would need to have an account to login to, so that they get prompted to the game’s page. This also assumes that the user and their details are in the database already. Since the user needs to login to view the game’s contents, the database is vital to the game’s access. The game’s save feature also assumes that the user’s progress is able to be written to the database. Since the user can come back to the game at any time to where they last left off, the save function depends on the database to correctly write the save data. This database information is also crucial to access the user’s profile. In order to see the user’s profile settings, the database needs to store this information such that when the user clicks on the profile page, they are able to access the records that exist for them.

Since this game requires the usage of SQLite, the database that is inbuilt into Django, this game assumes that SQLite is running efficiently so that the game can cater to the user’s register/login and profile information. The game also assumes that the user has already input their credentials in terms of profile registration into the database, so that the game can properly start with recording user’s save data and information. This game assumes that the database can connect to internet in order to access and modify the user’s profile data, as a previous assumption stated.

# **3. Specific Requirements**

## **3.1 External Interface Requirements**

### **3.1.1 User Interfaces**

#### **3.1.1.1 Home Page**

The Home Page will contain the title of our game, Shattered Rails, a small description of our game to entice the user’s attention to join the text-based adventure that this game offers. Top part of the page will display the navigation bar allowing you to switch to other pages. The “Login” button will take you to the login page, The “Register” button will take you to the register page. The “Settings” button will open a popup to the right of the screen allowing the user to customize the user interface. The left card provides a description of the first game and the card on the right provides a description of the second game. The bottom card is a side game built in to allow the user to play tic tac toe.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.1. Home Page

#### **3.1.1.2 Register Page**

The Register Page contains four fields for the user to input their information, “Email”, “Username”, “Password”, and “ Confirm Password”. Each field will be required for the user to input before they can submit. The user’s credentials will be securely stored in the database after inputting the information and clicking the “Register” button. The “Login” button, “Home” button, “Register” and “Settings” button are also present on this page where the login button redirects the user to the login page where they can use their credentials to log in, the home button will redirect the user back to the home page, the register button will take you to the registration page (current page), and the settings button will open a popup to allow the user to customize the user interface. After the user clicks submit to register they will be taken to a page telling the user to verify their email address.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.2. Registration Page

#### **3.1.1.3 Login Page**

The Login Page contains two fields for the user to input their credentials in, the “username” and “password”. After creating an account on the register page and verifying their email address, the user will be able to use their chosen username and password to sign-in to the application by clicking the “Login” button. If the user enters the valid username and password, then they will be redirected to the game page of the application where they can embark on their adventure of Shattered Rails. However, if the user enters an invalid username/password then the system will prompt an error message requesting the user to try again with a different username or password. Additionally, there are “Register” and “Home” page buttons, where if clicked, the user will be redirected to the corresponding pages. Clicking on the “Login” button will take you to the login page (current page) and clicking on “Settings” will open a popup allowing the user to customize the user interface.A screenshot of a computer login

Description automatically generated

Figure 3.1.1.3. Login Page

#### **3.1.1.4 Verify Email Notice Page**

This page displays a message to the user telling them to check the email that they registered with to get a email verification link where the email will be confirmed and will be allowed to login. A screen shot of a computer

Description automatically generated

Figure 3.1.1.4. Confirm Email Notice

**3.1.1.5 Game Selection Page**

The game selection page will display a nav bar on top with the options “Home”, “Profile”, “Logout”, and “Settings”. The home button will redirect the user to the home page, the profile button will redirect to the profile page, the logout button will log you out of the application removing access to most of the pages, and the settings button will open a popup allowing the user to customize the user interface. There are two cards in the middle with the left one taking you to the starting page before the first Shattered Rails game and the right card taking you to the starting page before the second Shattered Rails game. The bottom red card is a side game allowing you to play a game of rock paper scissors with an AI opponent.

Screens screenshot of a computer

Description automatically generated

Figure 3.1.1.5 Game Selection Page

**3.1.1.6 Start Page Before First Game**

This page is the start page before being able to play the first game. The “Start Mission” button will take the user to the first game.

A screen shot of a computer screen

Description automatically generated

Figure 3.1.1.6 Start Page Before First Game

**3.1.1.7 Start Page Before Second Game**

This page is the start page before being able to play the second game. The “Start Mission” button will take the user to the second game.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.7 Start Page Before Second Game

**3.1.1.8 Help Page**

The next three figures display different parts of the help page. The “Profile” button will redirect the user to the profile page, the “Game Selection” button will redirect to the game selection page, and the “Logout” button will log the user out restricting access to most pages. The tree diagram shows the different branches the user can take in the first game based on the decisions the user makes. Figure 1.1.1.8.2 shows some of the explanations offered by the help page. More can be seen as the user further scrolls down. In the bottom of the page, there are three side games with the first being Odd or Even where the user has to guess if the randomly generated number is odd or even. The second side game is Above or Below 5 where the user has to guess if the randomly generated number is above or below 5. The final side game on this page is a coin flip game where the user has to guess if the flipped coin is heads or tails.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.8.1 Help Page Displaying Game 1 Branches

A screenshot of a computer

Description automatically generated

Figure 3.1.1.8.2 Help Page Explaining Mechanics

A screenshot of a computer program

Description automatically generated

Figure 3.1.1.8.3 Help Page Side Games

**3.1.1.9 Profile Page**

Clicking on the “Profile” button on other pages will lead to the profile page. The “Profile” button will redirect the user to the profile page, the “Game Selection” button will redirect to the game selection page, the “Logout” button will log the user out restricting access to most pages, the “Home” button will redirect to the first landing page of the website where they see the descriptions of the game, and the “Settings” button will open up a popup where the user can customize the user interface. There is a side bar on the left side with two options being “Edit Profile” and “Edit Password”. Edit profile is set as the default when users first come to this page. Edit profile allows them to enter in a name, change their email address, change their username, or any combination of the three. An error message will be displayed if the email or username us taken and a success message will be shown if the profile is successfully updated. The edit password section allows the user to change their password by first typing in their current password, entering the new password, and retyping the new password to ensure it matches. If the current password is incorrect or if the new password and new password confirmation don’t match an error message will be shown. A success message will be displayed if the password is successfully changed.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.9.1 Edit Profile Page

A screenshot of a computer login

Description automatically generated

Figure 3.1.1.9.2 Edit Password Page

**3.1.1.10 Shattered Rails 1**

This page shows Shattered Rails 1. The “Profile” button will redirect the user to the profile page, the “Game Selection” button will redirect to the game selection page, the “Logout” button will log the user out restricting access to most pages, and the “Settings” button will open up a popup where the user can customize the user interface. The small bar in the top left shows the user their progress of the game and will be filled with green as the percentage of the game completed increases. The middle terminal is where the text from game will be outputted. The pause button pauses the flow of text and clicking on it will resume the flow of text. The restart game button will restart the game from the beginning. The black field below the two buttons only shows in the beginning and prompts the user to enter a name for their character.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.10 Shattered Rails 1

**3.1.1.11 Shattered Rails 2**

The next three figures show different components of Shattered Rails 2. The “Profile” button will redirect the user to the profile page, the “Game Selection” button will redirect to the game selection page, the “Logout” button will log the user out restricting access to most pages, and the “Settings” button will open up a popup where the user can customize the user interface. The “Pause” and “Restart Game” buttons return from the first game with the same functionalities. The first figure shows five different bosses the user can choose to fight from. The second figure shows the battle mechanics of Shattered Rails 2 with expanded mechanics compared to the first game. The first game only offered “Attack” and “Stay Still” while now Shattered Rails offers the same choices plus “Defend”, “Spell”, “Run”, and “Persuade”. The spell attack uses the MP resource and the amount of MP is dependent on the class chosen. The third figure shows the different classes a user can pick from in the beginning four different choices. Each class has their own statistics affecting things such as attack strength, MP amount, and persuasion chance.

A screenshot of a computer

Description automatically generated

Figure 3.1.1.11.1 Boss Select in Shattered Rails 2

A screenshot of a computer

Description automatically generated

Figure 3.1.1.11.2 Battle Mechanics in Shattered Rails 2

A screenshot of a computer

Description automatically generated

Figure 3.1.1.11.3 Class Select in Shattered Rails 2

### **3.1.2 Hardware Interfaces**

The application will rely on having a reliable browser that allows web applications to run without any additional problems during the process. A. reliable browser meaning suggests that it does not crash often and can handle tasks required by the web application. Recommended browsers like chrome and safari is what the web application is best suited for. This web application does not rely on any specific hardware interfaces.

### **3.1.3 Software Interfaces**

For this application, we will not be using any external software for the project such as any external APIs.

### **3.1.4 Communications Interfaces**

The web application communicates through HTTP requests with the browser and server where the user can perform actions like logging into their account or registering for an account, etc. HTTP requests will be sent from the browser, and an HTTP response will be sent from the server with an affirmation of the request from the browser.

## **3.2 Functional Requirements**

This section will explain the functional features of the Shattered Rails game.

### **3.2.1 FR1: Register**

|  |  |
| --- | --- |
| **ID** | FR1 |
| **Title** | Register |
| **Priority** | High |
| **Description** | The register feature allows the users that don’t have an account to register. This feature will prompt the user to enter their full name, username, email, and password. While registering there will be certain specifications for the password to be at least eight characters, one capital letter and one special character. Additionally, the form cannot be left empty before successful submission. |
| **Inputs** | * Name * Username * Email * Password |
| **Processing** | 1. User clicks on the register button. 2. This function checks if all the fields have appropriate values. 3. It sends the data to Django database. 4. Database sends a message to the application about successfully storing the data. 5. It sends an email for account verification. 6. The application would prompt the user to enter a four-digit code. 7. User enters that four-digit code for successful registration. |
| **Outputs** | Showing a message to the user that registration was successful and redirecting them to the login page. |
| **Error Handling** | If registration is unsuccessful, the user will not be able to access the game application and will not be able to login. |
| **Forward Dependencies** | FR2 |
| **Backward Dependencies** | N/A |

### **3.2.2 FR2: Login**

|  |  |
| --- | --- |
| **ID** | FR2 |
| **Title** | Login |
| **Priority** | High |
| **Description** | The login feature will allow users to login into their own personal account of Shattered Rails where users can save their progress in the game. |
| **Inputs** | * User enters their credentials for username and password. |
| **Processing** | * The system verifies the given credentials for username and password with Django’s database records. |
| **Outputs** | 1. If user is logged in successfully, the system will redirect the user to the game page. 2. If the login is unsuccessful, the system will display an error message, prompting user to try again. |
| **Error Handling** | * System will handle exceptions by informing the user that the login service is currently unavailable. |
| **Forward Dependencies** | * Successful implementation of this feature will route the user to the game page (FR5). |
| **Backward Dependencies** | * FR1 * Database must be set up to store the user’s account information and must be secure. |

### **3.2.3 FR3: Logout**

|  |  |
| --- | --- |
| **ID** | FR3 |
| **Title** | Logout |
| **Priority** | High |
| **Description** | The logout feature will allow users to logout of the Shattered Rails game and remove access to the game until you log back in again or register a new account and log back in. |
| **Inputs** | * The logout button. |
| **Processing** | * Set the user status as logged out. |
| **Outputs** | 1. Return to the login page. 2. Remove access from the game page and the account page. |
| **Error Handling** | * If log out process fails, then an error message will be displayed to the user encouraging them to try again later. |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | FR2 |

### **3.2.4 FR4: Home**

|  |  |
| --- | --- |
| **ID** | FR4 |
| **Title** | Home |
| **Priority** | High |
| **Description** | The home is the first page the user sees, it has a small description for our game to grab the user’s attention. It displays two buttons at the bottom of the screen, prompting the user to login or register to play Shattered Rails. |
| **Inputs** | * User interacts with the navigation elements on this page specifically the home and register buttons. |
| **Processing** | * The system processes the user’s request to go to the home page of the game and displays the content. |
| **Outputs** | * Home page content displayed which includes the title of the game, as well as a small description of the game and two buttons (login, register) for navigation. |
| **Error Handling** | * The system will display an error message to inform the user about the error that occurred. |
| **Forward Dependencies** | * Successful implementation of the home page will allow the user to navigate through the other pages of Shattered Rails much easier. |
| **Backward Dependencies** | N/A |

### **3.2.5 FR5: Path/Choice Selection**

|  |  |
| --- | --- |
| **ID** | FR5 |
| **Title** | Path/Choice Selection |
| **Priority** | High |
| **Description** | The Path/Choice Selection is the buttons that correspond to paths that the user can take in scenarios, once clicked upon, the user will be prompted down that path. This is the basis of the game since prompting the user for more and more choices and paths leads to endings. |
| **Inputs** | * The z/x/c/v/b buttons on bottom of screen. |
| **Processing** | * The user’s button input matches the corresponding path function. |
| **Outputs** | * The corresponding path function gets triggered and gets shown to the user. |
| **Error Handling** | * If a wrong button is pressed, the user is prompted with an error message to choose a valid option. |
| **Forward Dependencies** | * FR6 * FR16 * FR17 |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.6 FR6: Battle System**

|  |  |
| --- | --- |
| **ID** | FR6 |
| **Title** | Battle System |
| **Priority** | High |
| **Description** | The Battle System mechanic is activated when the user chooses an “Action-based” choice in a scenario. The game will transition into a battle system where the user can attack the enemy or defend themselves, and then the enemy will either attack/defend itself based on what it’s AI chooses to do. The user and the enemy take turns attacking/defending until either one’s health points drop to 0, leading to a victory/loss of the user |
| **Inputs** | * The attack/defend buttons on bottom of screen. |
| **Processing** | * The user’s attack/defend button input triggers the related function to attack/defend |
| **Outputs** | * User either attacks/defends against the enemy |
| **Error Handling** | * If a wrong button is pressed, the user is prompted with an error message to choose a valid option. |
| **Forward Dependencies** | * FR16 * FR17 |
| **Backward Dependencies** | * FR1 * FR2 * FR5 |

### **3.2.7 FR7: Adjust Text Speed**

|  |  |
| --- | --- |
| **ID** | FR7 |
| **Title** | Adjust Text Speed |
| **Priority** | Medium |
| **Description** | * This feature allows the player to adjust the speed at which text is displayed during the game. |
| **Inputs** | * Using the slider control to adjust text speed, and moving the slider left or right to decrease or increase speed. |
| **Processing** | * User must position the slider to the precise point according to their preferences. |
| **Outputs** | * Text is displayed at the precise adjusted speed of the slider. |
| **Error Handling** | * If there is an error, then the text-speed may be too fast or too slow for user to read. |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR6 |

### **3.2.8 FR8: Pause Text**

|  |  |
| --- | --- |
| **ID** | FR8 |
| **Title** | Pause Text |
| **Priority** | Medium |
| **Description** | User pauses the text that is being continuously outputted so that they can see previous text. |
| **Inputs** | * The pause button at the bottom of the screen. |
| **Processing** | * Once the pause button is hit, it triggers the pause() function, which stops flow of text. |
| **Outputs** | * The outflow of text on screen gets stopped in place. |
| **Error Handling** | N/A |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.89 FR9: Adjust Text Font Style**

|  |  |
| --- | --- |
| **ID** | FR9 |
| **Title** | Adjust Text Font Style |
| **Priority** | Medium |
| **Description** | This feature allows the user to change the text-font and to choose from over 70 different text fonts to their desired font option. It will then update the font that the user has selected by making the user’s option visible, after selecting a specific font. |
| **Inputs** | * Drop-down menu to choose from multiple text fonts. |
| **Processing** | 1. Capture the user's selection from the drop-down menu. 2. Apply the selected font style to the text displayed in the game. |
| **Outputs** | * Text displayed in the user’s chosen font style. |
| **Error Handling** | If there is an error, the user will not be able to change the text font style. |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR6 |

### **3.2.10 FR10: Adjust Text Font Color**

|  |  |
| --- | --- |
| **ID** | FR10 |
| **Title** | Adjust Text Font Color |
| **Priority** | Medium |
| **Description** | This feature allows the user to adjust the text font color of the text in the AI-text based adventure. Once the user chooses a color, the button color updates to the user’s chosen color. |
| **Inputs** | * Large-spectrum color palette to choose the text font color. |
| **Processing** | The user will select from the color-palette to a precise color of font.  It will then automatically update the font color. |
| **Outputs** | * The text is displayed in the chosen font color. |
| **Error Handling** | * The user will be unable to adjust the text font color. |
| **Forward Dependencies** | * N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR6 |

### **3.2.11 FR11: Adjust Background Color**

|  |  |
| --- | --- |
| **ID** | FR11 |
| **Title** | Adjust Background Color |
| **Priority** | Medium |
| **Description** | This feature allows the user to adjust the entire background color of the application. |
| **Inputs** | * Users select from a large-spectrum color palette to adjust the background color. Once the user chooses a color, the button color updates to the user’s chosen color. The background color is set to default white, so the user can adjust according to their preferences. |
| **Processing** | 1. The user will select a precise color according to their preferences. 2. It will capture the user's selection from the color palette. 3. Apply the selected color to the background of the application. |
| **Outputs** | * The background is displayed into the chosen font color. |
| **Error Handling** | * N/A |
| **Forward Dependencies** | * N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR6 |

### **3.2.12 FR12: Adjust Font Size Increase**

|  |  |
| --- | --- |
| **ID** | FR12 |
| **Title** | Adjust Font Size Increase |
| **Priority** | Medium |
| **Description** | This feature adjusts the size of the font when clicking on the “+” button. The font increases more and more based on the number of clicks. |
| **Inputs** | * Clicking on the "+" button to increase the font size. The more clicks the bigger the font. |
| **Processing** | 1. Receive user input by detecting the click event on the "+" button. 2. Increment the font size by a predetermined value or percentage. 3. Apply the increased font size to the text displayed on the application interface. |
| **Outputs** | * Updates the font size of the application based on the number of clicks. |
| **Error Handling** | * Font size cannot be increased. |
| **Forward Dependencies** | * N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR6 |

### **3.2.13 FR13: Adjust Font Size Decrease**

|  |  |
| --- | --- |
| **ID** | FR13 |
| **Title** | Adjust Font Size Decrease |
| **Priority** | Medium |
| **Description** | This feature adjusts the size of the font when clicking on the “-” button. The font decreases more and more based on the number of clicks. |
| **Inputs** | * Clicking on the "-" button to increase the font size. The more the clicks the smaller the font becomes. |
| **Processing** | 1. Receive user input by detecting the click event on the "-" button. 2. Increment the font size to decrease by a predetermined value or percentage. 3. Apply the increased font size to the text displayed on the application interface. |
| **Outputs** | * Updates the font size of the application based on the number of clicks on the “-” button. |
| **Error Handling** | * Font size cannot decrease. |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR6 |

### **3.2.14 FR14: Text to Speech**

|  |  |
| --- | --- |
| **ID** | FR14 |
| **Title** | Text to Speech |
| **Priority** | Medium |
| **Description** | This requirement implements a feature that reads the text input given on the screen into speech output. |
| **Inputs** | * Text given on the screen. |
| **Processing** | 1. The system processes text written on the screen and reads it by an automated voice using a text-to-speech service. 2. Utilizes web speech API to process the text on the page into speech audio. |
| **Outputs** | * Spoken audio corresponding to the text. |
| **Error Handling** | * Errors could occur when initializing the speech interface. An error message is implemented for users to try again. |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.15 FR15: Profile**

|  |  |
| --- | --- |
| **ID** | FR15 |
| **Title** | Profile |
| **Priority** | Medium |
| **Description** | The user profile feature allows logged in users to view their profile and see the information that they registered with displayed on the page. |
| **Inputs** | * The profile page button. |
| **Processing** | * The information will be retrieved from the database such as name, username, and email. |
| **Outputs** | * The profile page will be displayed to the user. |
| **Error Handling** | * If the profile page is unable to retrieve the information from the database, then an error message should be displayed to the user encouraging them to try again later. |
| **Forward Dependencies** | * Change User Information (FR18), the user must be on the profile page before they can modify their account information. |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.16 FR16: Save**

|  |  |
| --- | --- |
| **ID** | FR16 |
| **Title** | Save |
| **Priority** | Medium |
| **Description** | The save feature will allow the user to manually save their progress in the game allowing them to come back later and continue from where they last left off. |
| **Inputs** | * The save button. |
| **Processing** | * The current spot in the game and progress of branches completed will be stored in the database. |
| **Outputs** | * The current spot in the game and game progress on the page will match the same information stored in the database. |
| **Error Handling** | * If the game is unable to save, then an error message will be displayed encouraging the user to try again. * If the game is unable to save due to a weak or nonexistent network connection, then an error message will encourage the user to try again later or to establish a network connection. |
| **Forward Dependencies** | * The database will rely on the save feature to maintain the most up to date information about the user on where they are at in the game. |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.17 FR17: Game Completion Bar**

|  |  |
| --- | --- |
| **ID** | FR17 |
| **Title** | Game Completion Bar |
| **Priority** | Low |
| **Description** | This feature will show the progress bar for the percentage of the game complete keeping track of all the paths the user has visited. |
| **Inputs** | * N/A |
| **Processing** | * The information will be stored in the database and will be retrieved when the user logs in. |
| **Outputs** | * N/A |
| **Error Handling** | * N/A |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 * FR5 * FR6 |

### **3.2.18 FR18: Change User Information**

|  |  |
| --- | --- |
| **ID** | FR18 |
| **Title** | Change User Information |
| **Priority** | Low |
| **Description** | This feature will allow the logged in user to change their information such as name, username, email, and password. |
| **Inputs** | * User information such as email, name, username, current password, new password, and confirm new password. |
| **Processing** | 1. The information will be sent to the backend where it can be processed to ensure it meets requirements. 2. The password will be checked if it is different from the current password and if it meets the requirements of at least 8 characters with at least one capital letter, at least one number, and at least one special character. 3. The new email will be checked for if it is different from the current email and will check if the new email already exists in the database. 4. The new username will be checked for if it is different from the current username and it will check the database to ensure that the new username isn’t taken by another user. 5. The name will be allowed to be change without any prerequisites or restrictions. |
| **Outputs** | * The updated information will be displayed on the profile page. |
| **Error Handling** | * If the user tries to submit a new password that doesn’t meet the criteria, an error message will be displayed telling them to enter a password that meets the requirements. * If the new email address already exists in the database, then the user will receive an error message telling them to try a different email since that email is already taken by another user. * If the new username already exists in the database, then the user will receive an error message telling them to try a different username since that username is already taken by another user. * If the user enters in their current password incorrectly then the other changes will not go through, and an error message will be displayed saying to retype their correct password before the new changes can be applied. * If the new password and the retyped new password don’t match, then an error message will be displayed telling the user that the two passwords don’t match. * If the new changes can’t be applied for any other reason than an error message will be displayed saying that the changes could not be applied and for the user to try again later. |
| **Forward Dependencies** | * The database must be functional before the new changes can be applied. |
| **Backward Dependencies** | * FR1 * FR2 * FR15 |

### **3.2.19 FR19: Exit Warning**

|  |  |
| --- | --- |
| **ID** | FR19 |
| **Title** | Exit Warning |
| **Priority** | Low |
| **Description** | This feature will display a message to the user warning them that game progress could be lost if they don’t save. |
| **Inputs** | * Trying to close the tab on the web browser. * Clicking on the back page arrow on the web browser. * Clicking the log out button. |
| **Processing** | * Displaying a warning message. |
| **Outputs** | * A warning message with the option to save and exit. |
| **Error Handling** | * N/A |
| **Forward Dependencies** | * The game status will be saved to the database. |
| **Backward Dependencies** | * FR1 * FR2 * FR16 |

### **3.2.20 FR20: Email Verification**

|  |  |
| --- | --- |
| **ID** | FR20 |
| **Title** | Email Verification |
| **Priority** | Low |
| **Description** | Following the user's submission of the specified criteria within the registration form, the verification process begins. Upon the user's activation of the registration command, a unique four-digit verification code is sent to the user’s designated email address, and the code gets stored within the database for reference. The application interfaces with the user, prompting them to input the four-digit code retrieved from their email. This action initiates the validation process within the application's database and starts a process to match the user’s entered code, with the one stored in the database. If the code does not match, then the user can choose to resend another unique code to their email. Once the user inputs the correct code from their email, there will be a message that the verification was successfully validated/completed, and the user’s registration information will be stored in the database. After this point, the application will redirect the user to the login page enabling them to login with their information. |
| **Inputs** | * User registration information from the registration form. * User's email address. |
| **Processing** | 1. User submits registration form. 2. Application generates a four-digit code. 3. Application sends the code to the user's email. 4. User inputs the received code into the application. 5. Application checks if the input code matches the code stored in the database. 6. If the code matches, the user's registration information is stored in the database. 7. Application redirects the user to the login page. |
| **Outputs** | * Confirmation message indicating successful verification. |
| **Error Handling** | * If the code input by the user does not match the code stored in the database, display an error message and prompt the user to enter the correct code. * If there are issues with sending the code to the user's email, such as email server problems, display an appropriate error message and provide instructions to the user on what to do next. |
| **Forward Dependencies** | * FR2 * FR16 |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.21 FR21: Show Help**

|  |  |
| --- | --- |
| **ID** | FR21 |
| **Title** | Show Help |
| **Priority** | Low |
| **Description** | This feature allows the user access to a help-page. This feature enables the user to find out more about how to navigate through the game, and what all of the game features mean and how to use them. The help page will also include a step-by-step tutorial. |
| **Inputs** | * Help Page Button |
| **Processing** | * The user is redirected to the home page. |
| **Outputs** | * The user will now be on the help page. |
| **Error Handling** | If there are technical issues preventing the loading or display of the help page, the application should display an error message and provide instructions on what to do next. |
| **Forward Dependencies** | N/A |
| **Backward Dependencies** | * FR1 * FR2 |

### **3.2.22 FR22: User Password Recovery**

|  |  |
| --- | --- |
| **ID** | FR22 |
| **Title** | Forgot Password |
| **Priority** | Low |
| **Description** | This feature will allow the user to change their password after they have forgotten their password credential. |
| **Inputs** | * The forgot password button. * The email of the associated account * The new password you would like for the account. * Confirm the new password for the account. |
| **Processing** | 1. An email will be sent to the email address of the user with a link to where they can put their new password in. |
| **Outputs** | 1. The password for the email associated with the user will be changed in the database. 2. The user will now be able to login with the new password and not the old password. |
| **Error Handling** | 1. If the new password and the new password confirmation don’t match, then an error message will be displayed to the user. 2. The user will be prompted to enter a new password and to retype that new password. |
| **Forward Dependencies** | * FR1 |
| **Backward Dependencies** | * FR2 |

### **3.2.23 FR23: User Username Recovery**

|  |  |
| --- | --- |
| **ID** | FR23 |
| **Title** | Forgot Username |
| **Priority** | Low |
| **Description** | This feature will allow user to change their username after forgetting their username credential. |
| **Inputs** | * The forgot username button. * The email of the associated account * The new username you would like for the account. * Confirm the new username for the account. |
| **Processing** | * An email will be sent to the email address of the user with a link to where they can put their new username in. |
| **Outputs** | 1. The username for the email associated with the user will be changed in the database. 2. The user will now be able to login with the new username and not the old username. |
| **Error Handling** | 1. If the new username and the new username confirmation don’t match then an error message will be displayed to the user. 2. The user will be prompted to enter a new username and to retype that new username. 3. If the new username is the same as the old username then an error message will be displayed. |
| **Forward Dependencies** | * FR1 |
| **Backward Dependencies** | * FR2 |

## **3.3 Non-Functional Requirements**

This section of the document will cover the non-functional requirements for the Shattered Rails application. These are overarching goals we will strive for in the in the application.

### **3.3.1 Performance**

This application will not have any issues with performance. Changing in between pages should take less than 2 seconds. The performance of the application might be affected by a weak internet connection.

### **3.3.2 Reliability**

This application will work without any interruptions by having no crashes or freezes in the application. This application will also have error handling mechanisms for errors and exceptions and a subsequent error message will be displayed to the user to help them fix the error or guide them in the right direction.

### **3.3.3 Availability**

The availability of this application will be 24/7. However, if there is downtime needed for maintenance, it shall not exceed 1 hour and should be limited to once a month.

### **3.3.4 Security**

The application has a system where users can register their account through inputting their name, email, username and password to be able to access the game application.

For security, there are specific criteria for the password, which are at least eight characters, one capital letter, and one special character. These password criteria are set in place, to not have the user password become easily guessable, protecting the users account, and reducing the chances of other users accessing other accounts.

After the registration process, the application has a security protocol where users would receive a link via email. Upon receipt, users are prompted to click the link in their inbox from the shattered rails email, which will take them to the log in page. This serves to deter the creation of multiple accounts by a single individual, creating a more secure and authentic environment to the users.

After this process is complete, the user can then login with the registered information to authenticate.

### **3.3.5 Maintainability**

Utilize control systems like GitHub to track changes, manage collaboration, and have access to previous versions if necessary. This helps in maintaining stability and functionality through tracking changes.

Maintain regular updates to address bugs, introduce new features, and improve performance based on user feedback.

Implement consistent and descriptive naming for variables, functions, and classes to enhance code readability and maintainability. In addition, the design of the code will be modular to allow for the application to be split into smaller functions improving readability of the code for future developers.

Documentation of the application will maintained during initial development such as a README document which will allow future developers to reference to it when implementing future modifications.

### **3.3.6 Portability**

Conduct compatibility testing on a variety of devices, including PCs, consoles, mobile devices, and web browsers, to ensure consistent performance and functionality across different platforms.

Design responsive user interfaces that adapt to varying screen sizes and resolutions, accommodating the diverse range of devices on which the application will be used.

Continuously gather feedback from users experiencing bugs across different platforms to identify and address portability issues or compatibility challenges, prioritizing improvements based on user challenges.

# **4. Design Constraints**

## **4.1 Hardware Constraints**

* Hardware requirements for running Google Chrome (Preferred Browser for Windows):
  + Windows:
    - Windows 10 or later or Windows Server 2016 or later.
    - An Intel Pentium 4 processor or later that's SSE3 capable.
  + Mac:
    - macOS Catalina 10.15 or later.
  + Linux:
    - 64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, or Fedora Linux 32+.
    - An Intel Pentium 4 processor or later that's SSE3 capable.
* Hardware requirements for running Safari (Preferred Browser for Mac):
  + Any Mac running Mac OS X Leopard 10.5.7 or Mac OS X Tiger 10.4.11 and Security Update 2009-002.
  + Mac with an Intel processor or a Power PC G5, G4, or G3 processor and built-in FireWire® 256MB of RAM.

## **4.2 Program Design Constraints**

The primary design constraint of this project is that since it is a primarily text-based game, there needs to be extra emphasis placed on gameplay to be stimulating as if it is not, the game is just reading text and choosing between dialogues. Since the goal of the project to be engaging and memorable, the gameplay goes beyond the standard of choice selection and adds another part, like the enemy battle system. Another constraint is that this project is a web-based game without any visual based software such as adobe flash which is discontinued, so the game will be low on developer created visuals. This leads to relying on previously existing media (images, music, etc.) to use in the project, and this media has to be checked for its copyright/ license usage so it can be used without any legal problems. Due to this there must be extra focus on integration of aspects of CSS/JavaScript/HTML that allows for extra visual effects and a pleasant viewing experience for the user. In addition, the sound effects included in the game must be ensured that they are copyright free. A constraint that pops up is that each choice in a scenario is one of 2 choices, a stealth/action-based choice. This limits choices to being on or the either option, so there will only be 2 types of gameplay choices needed to program into the game.

## **4.3 UI Constraints**

On the development side, there are constraints when it comes to the UI of the website/game presentation. Due to the game being placed in a web browser for the whole game, UI is limited to a fixed viewpoint for the user, there will be one thing to view the whole game. The terminal is contained to being in a rectangular shape so that the user can read all the text being output whilst also choosing whichever button input on the outside of the terminal such as choices, or even options such as pause/text speed/help etc. As said before, the UI must be accessible to people that have certain disabilities or impairments, allowing for them to experience the game to the fullest extent. With this in mind, the UI has to not only suit physically abled individuals but also members of the userbase that have certain accommodation needs, leading to a double job of catering to both audiences. Another constraint is the fact that the default theme is good enough that if a user does not require any further changes to the format, they can be content viewing content in the default option.

## **4.4 Video Game Rating/Audience Constraints**

Another constraint is that the design of the game has to be limited by the video game rating system. For video games, there is an organization called ESRB (Entertainment Software Rating Board), which gives certain ratings to games based on the subject matter/materials that is in the game. Shattered Rails is not extremely dark in nature, but there are mature themes in it, so it would be rated as a T(Teen) or M(Mature) rating, similar to PG-13/R ratings for movies. Shattered Rails must be constricted to fit into these rating systems, and to not get an extremely mature rating as that would restrict coverage and userbase exponentially. Along with the rating system, this is a project that is intended for demonstration to a university audience, meaning that the subject matter must also be aligned with the values of Wayne State University, alongside what is acceptable to show to college students and college faculty, in a classroom setting, and also in a presentation setting for school officials/organizations, outside stakeholders, and the general public.

# **5. Logical Database Requirements**

This section of the document will explain how the database will store user information and information pertaining to the game in the database.

Shattered Rails will use the SQLite database built into the Django framework to store user information that will be used for account authentication and will store game information connected to the user. The logical database requirements will be outlined below on how the data will be organized, the measures taken to ensure data integrity and retention, and where it will be used. The figure below provides a visual view of the design of the database through an ER Diagram.

A black background with white circles

Description automatically generated

Figure 5.1. Database ER Diagram

* User Profile
  + User profile will be stored in the SQLite database.
  + Information that will be stored includes name, email address, username, password, and verification status.
* Game Information
  + Game information relating to the current logged in user will be stored in the SQLite database.
  + Will be an extension of the User table.
  + Information that will be stored will be the in-game name, save state of the game, the amount of health you have, and percentage of the game completed.

The database will be accessed in five different circumstances. The first will be when a new user registers for the game which will store the registration information and the verification status. The second instance will be when the user verifies their email address it will access the database and verify the status of the account. The third instance will be when the user logs in, it will access the database to retrieve the information relating to the game such as save state and percentage completed. The fourth instance will be when you click on the save button which will access the database to update the current game state. Similarly, the fifth instance will be when you reach checkpoints throughout the game which will access the database to update the current game state. The sixth instance where the database will be accessed is when one of the branches in the story is completed, which will access the database and update the percentage of the game completed.

For the data retention, we will be dropping users that are left unverified after 24 hours. This is to ensure that the database does not get filled with empty accounts that don’t delve deeper into the application. The verification process will be an email sent to the email that was used during registration where you will click on a link to verify the account. In addition, to keep password secure for accounts we will be using an encryption method for passwords after the user registers for the application.

Data integrity for this project will be maintained by input validation for the data and by managing access controls. Input validation allows for information to follow a certain standard. For example, establishing password character requirement allows for strong passwords to be stored in the database. Access controls will be limited to only certain users so that only authorized users can manipulate the database and will establish a trail if the database is manipulated maliciously.

# **6. Analysis Models**

## **6.1 Data Flow Diagram (DFD)**

