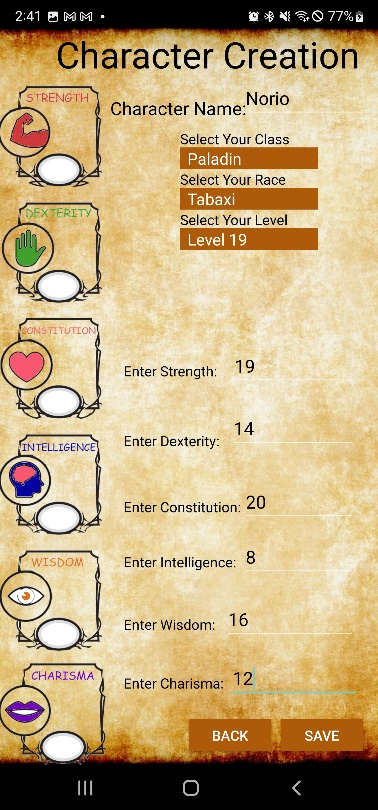
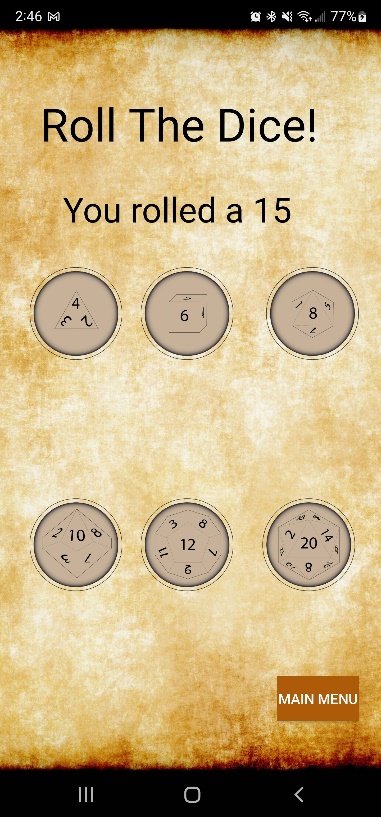
THE GO TO COMPANION APP for D&D: Beholder's Almanac

Save yourself the carpel tunnel! Character sheets, Dice Roller, Music Player  
Its easy to import your existing characters into the app using a simple touch screen interface. The dice roller allows you to roll all the dice you need and instantly displays the result for you. The music player will be further developed to allow a user to select what ambiance they want, setting the proper mood for the table!

A screenshot of a video game

Description automatically generated with medium confidence 

Contents

[1. Project Level Summary Evaluation - Requirements 4](#_Toc106885429)

[2. IEEE 830 Software Requirements Specifications 4](#_Toc106885430)

[2.1 Introduction 4](#_Toc106885431)

[2.1.1 Purpose 5](#_Toc106885432)

[2.1.2 Document Conventions 5](#_Toc106885433)

[2.1.3 Intended Audience and Reading Suggestions 5](#_Toc106885434)

[2.1.4 Product Scope 5](#_Toc106885435)

[2.1.5 References 5](#_Toc106885436)

[2.2 Overall Description 6](#_Toc106885437)

[2.2.1 Product Perspective 6](#_Toc106885438)

[2.2.2 Product Functions 6](#_Toc106885439)

[2.2.3 User Classes and Characteristics 6](#_Toc106885440)

[2.2.4 Operating Environment 6](#_Toc106885441)

[2.2.5 Design and Implementation Constraints 7](#_Toc106885442)

[2.2.6 User Documentation 7](#_Toc106885443)

[2.2.7 Assumptions and Dependencies 7](#_Toc106885444)

[2.3 External Interface Requirements 7](#_Toc106885445)

[2.3.1 User Interfaces 7](#_Toc106885446)

[2.3.2 Hardware Interfaces 8](#_Toc106885447)

[2.3.3 Software Interfaces 8](#_Toc106885448)

[2.3.4 Communications Interfaces 8](#_Toc106885449)

[2.4 System Features 9](#_Toc106885450)

[2.4.1 System Feature #1 (Character Sheet) 9](#_Toc106885451)

[2.5 System Feature #2 (Dice Roller) 9](#_Toc106885452)

[2.6 Other Nonfunctional Requirements 9](#_Toc106885453)

[2.6.1 Performance Requirements 9](#_Toc106885454)

[2.6.2 Portability 9](#_Toc106885455)

[2.6.3 Security Requirements 9](#_Toc106885456)

[2.6.4 Software Quality Attributes 9](#_Toc106885457)

[2.6.5 Business Rules 10](#_Toc106885458)

[Appendix A: Glossary 10](#_Toc106885459)

[Appendix B: Use Case Diagram 11](#_Toc106885460)

[1 OOAD Diagrams 12](#_Toc106885461)

[1.1 Updated Use Case Diagram 12](#_Toc106885462)

[2 12](#_Toc106885463)

[2.1 Use Case Specifications 12](#_Toc106885464)

[2.7 1.2.1 Use Case #1: Character Sheet 13](#_Toc106885465)

[2.8 13](#_Toc106885466)

[2.9 1.2.2 Use Case #2: 14](#_Toc106885467)

[2.10 1.2.3 Use Case #3: 15](#_Toc106885468)

[2.11 1.2.4 Use Case #4: 16](#_Toc106885469)

[2.2 Class Diagram 17](#_Toc106885470)

[2.3 Sequence Diagrams 18](#_Toc106885471)

[2.4 State Machine Diagram 20](#_Toc106885472)

[3 Project Level Summary Evaluation – Analysis and Design 20](#_Toc106885473)

[4 Risk Assessment Matrix 21](#_Toc106885474)

[5 Project Level Summary Evaluation – Testing 21](#_Toc106885475)

[5.1 Testing Evaluation 21](#_Toc106885476)

[6 Test Plan 22](#_Toc106885477)

[6.1 Test Plan Identifier 22](#_Toc106885478)

[6.2 Introduction 22](#_Toc106885479)

[6.3 Test Items 22](#_Toc106885480)

[6.4 Features to be Tested 22](#_Toc106885481)

[6.5 Features not to be Tested 23](#_Toc106885482)

[6.6 Approach 23](#_Toc106885483)

[6.7 Items Pass/Fail Criteria 23](#_Toc106885484)

[6.8 Test Deliverables 23](#_Toc106885485)

[6.9 Environmental Needs 23](#_Toc106885486)

[6.10 Responsibilities 24](#_Toc106885487)

[6.11 Staffing and Training Needs 24](#_Toc106885488)

[6.12 Schedule 24](#_Toc106885489)

[6.13 Risks and contingencies 24](#_Toc106885490)

[7 Test Cases 24](#_Toc106885491)

[Naming Standard 24](#_Toc106885492)

[Test Case #1 24](#_Toc106885493)

[Test Case #2 25](#_Toc106885494)

[Test Case #3 25](#_Toc106885495)

[Test Case #1 26](#_Toc106885496)

[1 Updated Project Plan 26](#_Toc106885497)

# Project Level Summary Evaluation - Requirements

What would you do better or differently?

* Digitize the experience via mobile app
* Ease of access for all ages
* Simplify complex rules using built in macros

What was the quality of the tools and techniques?

* Primary tool used is Microsoft office. Text is properly formatted.

Are the requirements clear?

* Requirements listed:
  + Android application
  + Information sheet functionality
  + Dice roller function
* General asset requirements listed

What are the strengths and weaknesses?

* Strengths
  + Easy to understand
  + Clear outline of project
  + User-base is wide and has large market share
* Weaknesses
  + Design schema unclear
  + Make be unclear to those unfamiliar with tabletop roleplaying games

What are your recommendations or suggestions?

* Screenshots of the project outline would be helpful
* A design schema showing the functional

# IEEE 830 Software Requirements Specifications

## Introduction

Created by Ernest Gygax and David Arneson, Dungeons and Dragons debuted 1974 as a fantasy role-playing game with over 50 million players since its release. With a game that was released before the VHS was established there have been numerous updates and quality of life adjustments over the life of the game. But to its core the game is still a pencil and paper game with dice as its core mechanic. The issue arises as games become more complicated, characters have more stats and enemies become more numerous keeps tracking of all these things with just a pencil and paper can at times feel antiquated and clunky. Making sure you’re properly keeping track of all of this can take away from the purpose of the game, the roleplaying.

### Purpose

2,500,000,000. There are over 2.5 billion active Android users worldwide, with more than 3 million devices across the globe using the Android operating system. You will be hard pressed to find someone that doesn’t always have a smart phone in their pocket. The solution is to create a mobile app that takes away a lot of the inconvenience and clunkiness of pencil and paper and puts the work and functionality in the hands of the end-user utilizing a mobile app to track character sheet information and added functionality of a dice roller.

### Document Conventions

Bolded words or phrases will be used to indicate features of the project. Italics will be used to indicate user interactions. Under-lined sections are to indicate features or tools yet to be implemented.

### Intended Audience and Reading Suggestions

The intended audience for D&D podcasts is people who are considering venturing into playing Dungeons and Dragons or who are already playing. The application can also be used be anyone who has a need for a dice roller for any tabletop game.

### Product Scope

The main deliverable for this project is a working Android application that will allow an end user to input information to create their character sheet. This should show the character’s name, hit points, speed, armor, initiative, and main stats (e.g., Strength, Intelligence, Dexterity… etc.) Ideally I want the character sheet to be updated easily with new information and display the updated information quickly. The second function of the app is to act as a random dice roller. I want the function to allow the user to select how many sides the dice has (e.g., 4,6,20, 100 etc.) as well as how many dice should be rolled at once. The function should be able to display each dice individually and display a total output sum of the dice if multiple is rolled at once. To accomplish this there will be assets that need to be made for the application logo, icon, user-interface, and dice faces.

### References

<https://www.polygon.com/23023498/hasbro-buys-dnd-beyond-pdf-wotc-fandom>

<https://en.wikipedia.org/wiki/Dungeons_%26_Dragons>

<http://www.acaeum.com/ddindexes/setpages/basic.html>

<https://github.com/fexed/RPG-Character-Sheet>

<https://dnd.wizards.com/>

## Overall Description

### Product Perspective

This will be a standalone product designed to be an android application on a personal cellphone, tablet, or computer.

### Product Functions

* Savable/Editable Character Sheet
* Dice Roller

### User Classes and Characteristics

* General User
  + The general user will have all the standard functions of the application like the character sheet and the dice roller.
  + Standard ads will be displayed, and they will have a limit to how many character sheets they have at any one time
* Power User
  + This user will have all the functions of the previous user but will also no longer be subject to ad rolls within the application.
  + This user will have 2x the character sheet limit as the general user.

### Operating Environment

Android Platform Version 5.0 Lollipop API 21. This will allow the app to run on approximately 98.6% of devices.   
A picture containing timeline

Description automatically generated

### Design and Implementation Constraints

Primary limitation will be multiple different forms of hardware to test the application on. Android studio, the primary IDE to be used allows for emulators of multiple devices so this may help limit testing issues.

### User Documentation

User documentation will be included in the Google play store application information section as well as listed within the settings section of the application.

### Assumptions and Dependencies

Improper integration with Android Studio or system failure.

## External Interface Requirements

### User Interfaces

The primary aim of the application is to ensure ease of access and simplicity in form. The original launch screen will show the three main interfaces. The character sheet, dice roller and general settings of the application.   
A picture containing text, envelope, businesscard

Description automatically generated

### Hardware Interfaces

The general GUI will utilize each device’s built-in touch screen functionality to allow the user to interface with the application. The application will eventually have sound so the device’s speaker will be utilized to generate sound.

### Software Interfaces

The primary software interfaces will be between the application and the user’s current Android API. There will also be integrated ad feeds utilizing google.

### Communications Interfaces

The communication functions required by this application will be primarily electronic forms to record and keep track of the character sheets. An active network connection will allow the ads to display properly in the application. Within the settings section of the application there will be a section for general users to email bugs or issues with the software.

## System Features

### System Feature #1 (Character Sheet)

The first will allow the user to input information to create their character sheet. This should show the character’s name, hit points, speed, armor, initiative, and main stats (e.g., Strength, Intelligence, Dexterity… etc.) Ideally I want the character sheet to be updated easily with new information and display the updated information quickly.

Functional requirements

* + Electronic forms, savable and editable

## System Feature #2 (Dice Roller)

The second function of the app is to act as a random dice roller. I want the function to allow the user to select how many sides the dice has (e.g., 4,6,20, 100 etc.) as well as how many dice should be rolled at once. The function should be able to display each dice individually and display a total output sum of the dice if multiple is rolled at once.

Functional requirements

* + Random number generator integrated into each icon which varies depending on face value selected

## Other Nonfunctional Requirements

### Performance Requirements

The application should be light weight and extremely responsive even on older machines.

### Portability

Since this application is utilizing the Lollipop API 98.6% of Android users should be able to use this system.

### Security Requirements

Data used and created within this application stays on the end user’s device. A future design will include the option for the user to login to the application allowing them to access their character sheets from multiple platforms at once.

### Software Quality Attributes

Adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Highest priority placed on usability.

### Business Rules

General and Power users will be able to perform the main functions of the program. Administrators will have admin access to program’s scripting and ad sense.

# Appendix A: Glossary

* API: Stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.
* Character Sheet: A D&D character sheet is the player's bible as they play throughout the adventure. It has absolutely everything a player needs to tell them their character's health, class, name, level, stats, items, background, and even spell slots if you choose a class that uses magic.
* D&D(DnD): Dungeons & Dragons is a fantasy tabletop role-playing game originally designed by Gary Gygax and Dave Arneson. It was first published in 1974 by Tactical Studies Rules, Inc. It has been published by Wizards of the Coast since 1997.
* GUI: A GUI uses windows, icons, and menus to carry out commands, such as opening, deleting, and moving files. Although a GUI operating system is primarily navigated using a mouse, a keyboard can also be used via keyboard shortcuts or the arrow keys.
* IDE: An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger

# Appendix B: Use Case Diagram

**

# OOAD Diagrams

# Updated Use Case Diagram

# 

****

# Use Case Specifications

## 1.2.1 Use Case #1: Character Sheet

## 

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: **Character Sheet** | | ID: character\_sheet | Importance Level: High |
| Primary Actor: General User | Use Case Type: Standard | | |
| Stakeholders and Interests: General User | | | |
| Brief Description: The user is brought to the character sheet page where they have the option to start a new character sheet, load an existing sheet or return to the main menu. | | | |
| Trigger: Button press from the main menu.  Type: Mouse left click or touch screen tap | | | |
| Relationships:  Association: Main Menu  Include: Return to main menu  Extend: Add new character, Load existing character  Generalization: Main Application Function | | | |
| Normal Flow of Events:   1. User clicks Character sheet from Main Menu. 2. User selects either new character, existing character, or main menu. 3. Functions proceeds depending on which option user chooses. | | | |
| Sub Flows:  S-1: Using mouse clicks   1. User hovers mouse cursor over preferred option and left clicks. 2. Depending on button chosen application proceeds to next operation.   S-2: Using touchscreen   1. User taps preferred option using either their hand or stylus. 2. Depending on button chosen application proceeds to next operation. | | | |
| Alternate/Exceptional Flows: none | | | |

## 1.2.2 Use Case #2:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: **Dice Roller** | | ID: dice\_roller | Importance Level: High |
| Primary Actor: General User | Use Case Type: Standard | | |
| Stakeholders and Interests: General User | | | |
| Brief Description: The user is brought to the dice roller page where they have the option to roll different dice depending on which dice face they choose. After selecting the dice, a display will populate with the results. | | | |
| Trigger: Button press from the main menu.  Type: Mouse left click or touch screen tap | | | |
| Relationships:  Association: Main Menu  Include: Return to main menu  Extend: Roll Dice, Display dice output  Generalization: Main Application Function | | | |
| Normal Flow of Events:   1. User clicks Dice Roller from Main Menu. 2. User selects a dice face, or main menu. 3. If dice face is selected a text output appears showing results of roll. 4. Functions proceeds depending on which option user chooses. | | | |
| Sub Flows:  S-1: Using mouse clicks   1. User hovers mouse cursor over preferred option and left clicks. 2. Depending on button chosen application proceeds to next operation.   S-2: Using touchscreen   1. User taps preferred option using either their hand or stylus. 2. Depending on button chosen application proceeds to next operation. | | | |
| Alternate/Exceptional Flows: none | | | |

## 1.2.3 Use Case #3:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: **Music Player** | | ID: music\_player | Importance Level: low |
| Primary Actor: General User | Use Case Type: Standard | | |
| Stakeholders and Interests: General User | | | |
| Brief Description: The user can turn on background music from the main menu by clicking the “ON” button. Upon leaving page, clicking “OFF” or the song playing completely the music player will end freeing up system resources. | | | |
| Trigger: Button press from the main menu.  Type: Mouse left click or touch screen tap | | | |
| Relationships:  Association: Main Menu  Include:  Extend: Turn Music On, Turn Music Off  Generalization: Main Menu Application Function | | | |
| Normal Flow of Events:   1. User clicks Character sheet from Main Menu. 2. User selects either new character, existing character, or main menu. 3. Functions proceeds depending on which option user chooses. | | | |
| Sub Flows:  S-1: Using mouse clicks   1. The user can turn on background music from the main menu by clicking the “ON” button. 2. Upon leaving page, clicking “OFF” or the song playing completely the music player will end freeing up system resources.   S-2: Using touchscreen   1. User taps preferred option using either their hand or stylus. 2. Depending on button chosen application proceeds to next operation. | | | |
| Alternate/Exceptional Flows: none | | | |

## 1.2.4 Use Case #4:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: **Settings** | | ID: settings | Importance Level: medium |
| Primary Actor: General User | Use Case Type: Standard | | |
| Stakeholders and Interests: General User | | | |
| Brief Description: The user is brought to the settings page where they have the option to submit a bug report and view application data. | | | |
| Trigger: Button press from the main menu.  Type: Mouse left click or touch screen tap | | | |
| Relationships:  Association: Main Menu  Include:  Extend:  Generalization: Main Application Function | | | |
| Normal Flow of Events:   1. User clicks Settings from Main Menu. 2. User can send bug report to administrator to review. 3. User can review current build number for application and general application information. | | | |
| Sub Flows:  S-1: Using mouse clicks   1. User hovers mouse cursor over preferred option and left clicks. 2. Depending on button chosen application proceeds to next operation.   S-2: Using touchscreen   1. User taps preferred option using either their hand or stylus. 2. Depending on button chosen application proceeds to next operation. | | | |
| Alternate/Exceptional Flows: none | | | |
|  | | | |

# Class Diagram



# Sequence Diagrams





# State Machine Diagram



# Project Level Summary Evaluation – Analysis and Design

* What would you do better or differently?
  + Taking more time in initial planning would have helped outline the project scope and resources needed.
  + Some variables have similar names which can cause confusion if documentation isn’t done properly.
* Do they have a good understanding of key risks for their projects and how to mitigate risk (external and internal)?
  + Yes risks have been documented and are minimal due to likely non-public release.
* What additional risks would you consider for the project (minimum of three)?
  + Time risk may be present as application is currently being developed on an accelerated schedule.
  + Market risk exists if application goes public.
  + Performance risks may be applicable to users with outdated firmware.

# Risk Assessment Matrix

Risk scale: 1 = low, 2 = fair, 3 = medium, 4 = probable, 5 = high

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risks** | Likelihood | Preventive Measure | Response | Impact |
| Hardware Failure | 2 | * Test on multiple platforms to ensure wide functionality. | * Utilize bug report within settings | * Schedule * Budget |
| Cost Risk | 3 | * Establish budget based off initial requirement scope * Update budget routinely throughout development | * Find other resources of money to help support your project | * Budget |
| Appropriate API | 2 | * API 28 has been chosen for wide coverage 98.6% and functionality. * Always look for cost, support, and reviews | * Unsure marketing targets devices with at least API 28. | * Requirements * Schedule * Budget |
| Time Constraints | 5 | * Make a schedule that allows for periodic meetings to update advancement | * Alternating paperwork and technical work each week * Set weekly goal to motivate and challenge us | * Schedule |
| Technology Risks | 3 | * Test on multiple platforms to ensure wide functionality. | * List API requirements on Play store | * Launch |
| Marketing Risks | 3 | * Use surveys, questionnaires, and reviews to get customers opinion of product | * Rework product by changing requirements | * Requirements |

# Project Level Summary Evaluation – Testing

# Testing Evaluation

* What would you do better or differently?
  + Document program functions properly to ensure ease of reading
  + Label in progress sections of code and layouts
  + Implement authentication via Firebase earlier
* Do you get a good understanding of their testing process or the thoroughness of their testing? Yes or no? Explain either way.
  + Yes, records were kept of previous stages of program allowing me to go back and review test stages as they were being done and the updates to the code and functionality along the way.
* Is there traceability from requirements?
  + Technically yes, development has been ongoing as requirements and use cases have been established.
* Do they have a test plan and test cases? If so, describe the quality. How do they apply to the three testing concepts of verify, validate, and explore?
  + Yes test plan was created to allow multiple users to interact with the application and review functionality. Documenting the experience testing to verify each section of the application was working correctly. Validation and exploration of application was also done during the development cycle as well as during the testing phase.
* How would your new functional requirements for the project impact the testing process?
  + The new functional requirements added to this project will require additional testing in the way of user authentication and cloud storage utilizing Firebase. Separate testing for each may be required to ensure functionality.

# Test Plan

## Test Plan Identifier

BeholdersAlamanc\_0001

## Introduction

The objective of this test plan is to test the functionality of ‘Beholder’s Almanac’. This test plan was created for Professor Ostrowski CEIS 200 Software engineering class and focuses on the Android application Beholder’s Almanac documentation based on the criteria for Lab 3 and the IEEE 829 standard format. This test plan covers component/class based testing for four test cases.

## Test Items

Main Menu functionality, Character Creation, Dice Roller, Music Player.

## Features to be Tested

* Main Menu
  + Allows user to navigate each section of the app
  + Ensure Main menu launch page appears correctly on multiple device types
* Character Creation
  + Allows user to create a new character utilizing drop down lists and entry forms
  + Saves entered data to be stored both on device and cloud
  + Allows user to retrieve saved character data
* Dice Roller
  + Allows user to roll dice by clicking on dice face (D4, D6, D8, D10, D12, D20)
  + Displays roll utilizing random number generator for user.

## Features not to be Tested

* Settings Section still in development
* User authentication & Login still in development
* Cloud storage utilizing Firebase still in development

## Approach

* Emulators will be used to test multiple device types (Android phone small, large, tablet)
* Physical devices will also be used to ensure real life functionality and testing user experience.

## Items Pass/Fail Criteria

Below are the stipulations that apply to our testing process to assure that all our testing is based on the same principles and requirements outlined.

* Each individual functional requirement must be tested all throughout the application to be considered acceptable.
* Make sure that each function of the application is functional and bug-free.

## Test Deliverables

Below are the deliverables presented after all testing is complete.

* Test plan
* Test cases
* Testing matrixes
* Summary/review reports

## Environmental Needs

The list below includes all the tools needed to successfully test all our functional requirements.

* An Android device with minimum API that meets our system requirements
* Touch screen compatible hardware
* Internet connection that meets appropriate requirements
* Customers to test application and give feedback

## Responsibilities

Test plan should be prepared by Test Lead. Preparation and execution of tests should be carried out by testers.

## Staffing and Training Needs

Plan training video to improve the skills of resources in the project to achieve the desired goals.

## Schedule

Testing is being performed throughout the entire development cycle. From the start each main function was broken down and tested individually before new functionality was added. Once a new function was added additional testing was done to not only check the new function but also the previous functions to ensure no code breakage occurred. Most testing was done on emulator formats along with physical devices.

## Risks and contingencies

In case of a wrong budget estimation, the cost may overrun. Contingency Plan – Establish the scope before beginning the testing tasks and pay attention in the project planning and track the budget estimates constantly.

# Test Cases

Naming Standard

MNU = Menu CHA = Character Sheet DIE = Dice Roller MUS = Music Player

## Test Case #1

|  |  |
| --- | --- |
| Test Identifier: | MNU\_\_0001 |
| Requirements Addressed: | Ensure that the user has the ability to access all main sections of application from Main Menu |
| Prerequisite Conditions: | Player must be at the “Main Menu” screen |
| Test Input: | Click Button Options from Main menu |
| Expected Test Results: | Each function page loads up click |
| Instructions for Conducting Procedure: | Click a button to access the desired function |
| Features to be Tested: | Button functionality |
| Requirements Traceability: | Character Sheet, Dice Roller, Music Player |
| Rationale for Decisions: | Allows user quick access to all app functions from launch page. |

## Test Case #2

|  |  |
| --- | --- |
| Test Identifier: | CHA\_\_0001 |
| Requirements Addressed: | Ensure that the user has the ability to access Character sheet function |
| Prerequisite Conditions: | Player must be at the “Character Sheet” screen |
| Test Input: | Click Button Options from Character Sheet |
| Expected Test Results: | Each function page loads up click |
| Instructions for Conducting Procedure: | Click a button to access the desired function |
| Features to be Tested: | Button functionality |
| Requirements Traceability: | New Character, Load Character, Main Menu |
| Rationale for Decisions: | Allows user to create and load character sheets |

## Test Case #3

|  |  |
| --- | --- |
| Test Identifier: | DIE\_\_0001 |
| Requirements Addressed: | Ensure that the user has the ability to access all Dice roller functions. |
| Prerequisite Conditions: | Player must be at the “Dice Roller” screen |
| Test Input: | Click Dice facer from Dice Roller screen |
| Expected Test Results: | Each function page loads up click |
| Instructions for Conducting Procedure: | Click a button to access the desired function |
| Features to be Tested: | Button functionality |
| Requirements Traceability: | Dice Face rolls |
| Rationale for Decisions: | Allows user quick access to all app functions from dicer roller page |

## Test Case #1

|  |  |
| --- | --- |
| Test Identifier: | MUS\_\_0001 |
| Requirements Addressed: | Ensure that the user has the ability to access all music functions from main menu |
| Prerequisite Conditions: | Player must be at the “Main Menu” screen |
| Test Input: | Click Button Options from Main menu |
| Expected Test Results: | Each function page loads up click |
| Instructions for Conducting Procedure: | Click a button to access the desired function |
| Features to be Tested: | Button functionality |
| Requirements Traceability: | Character Sheet, Dice Roller, Music Player |
| Rationale for Decisions: | Allows user quick access to all app functions from launch page. |

# Updated Project Plan

Table

Description automatically generated