DnD Toolkit

SWDV:691

Austin Glass

Contents

[Background 3](#_Toc202109637)

[Problem Description 3](#_Toc202109638)

[Personas 3](#_Toc202109639)

[Solution 4](#_Toc202109640)

[Justification 4](#_Toc202109641)

[Minimum Viable Product 4](#_Toc202109642)

[Core Features 4](#_Toc202109643)

[Minimal Feature Set 5](#_Toc202109644)

[High Level Architecture 5](#_Toc202109645)

[High Level Data Management 6](#_Toc202109646)

# Background

Among the most well-known of tabletop roleplaying games (TTRPG) is Wizards of the Coast’s Dungeons and Dragons (D&D). At its core, the game is simple. A dungeon master (DM) speaks a situation to the player(s) who act out their own player character’s (PC) response. For uncertain outcomes, the use of random chance is handled by rolling dice.

The game does have many rules and components that require memorization, effort, preparation, and organization. Many tools and applications have been created to assist in immersing players in the game. While many of these tools have proven effective such as Roll20 or D&D Beyond, these tools have a steep learning curve and often price many users out of the market. There remains a gap to provide D&D players with the necessary tools so that they can fully enjoy a seamless experience without getting bogged down by details.

# Problem Description

Organization in D&D is difficult. Creating multiple tools that require steep learning curves doesn’t reduce the struggle for DMs and players. It is necessary to create a solution that provides a way to keep organized, reduce mental burden, and be easy to learn.

## Personas

#### Dungeon Master Dave

Dungeon Master Dave represents the DM. He juggles many roles during a session of D&D. Dave creates the world and plays any non-player character (NPC). Dave **keeps track of the world** he creates such as new locations and their social structures. Dave must think on his feet as the PCs often do things requiring Dave to **quickly think of a new NPC or encounter.** Of course, when things don’t go according to plan, Dave also has to handle combat where he **keeps track of multiple enemies including their turn order and hit points.**

#### Patrick the Player

Patrick is a player. He controls one character that he spent hours creating a backstory and physical appearance for. Patrick gets to experience the world that the DM has created, but Patrick has to **keep track of his own character** since the DM is not. Patrick creates his character with all the stats. He adventures along the story **following new quests with ever-changing objectives.** He may even pick up a few magical items along the way.

## Solution

The solution is a simple, mobile-first web application. The application will help players and DMs stay organized throughout a campaign and allow them to focus on the story while it handles some of the mentally taxing aspects of play. It will accomplish this in 4 key ways.

1. Built-in Character Sheets: A player can input a few key attributes, and the application will use business logic to handle the rest of the calculations.
2. Adventure Log: Any locations, objectives, etc. can be added so that players do not have to remember what they were doing after weeks of not playing.
3. Random Generators: Random generators allow for balanced combat encounters using rules based on PC level and party size as well as NPC generation.
4. Initiative Tracker: The DM can input all combatants and monitor turn and hit points all in one location at the palm of his hand.

## Justification

This application is designed as a quality-of-life tool to help D&D players and DMs stay organized and focused on storytelling. While similar tools exist, many are overly complex or costly. This project aims to offer a simpler, more accessible alternative without trying to compete as a commercial product

# Minimum Viable Product

## Core Features

#### Built-in Character Sheets

Built-in character sheets allow for a player to input key information such as level, proficiencies, and ability scores. The application will then take that data to determine ability and skill modifiers. It will also determine commonly used stats such as armor class (AC), proficiency bonus, initiative bonus, and passive perception.

#### Adventure Log

The adventure log will serve as the memory center for the application. It will store 4 pieces of information: Locations, NPCs, Session Recaps, and Quests. The purpose is to reduce the amount of information that needs to be remembered from week to week. While the PCs would likely remember something like the town mayor’s name, the players may have heard it once and forgotten it. This feature will allow players to quickly reference important aspects of the world in a familiar, blog-like interface.

#### Random Generators

1. Random NPC Generator: This creates an NPC with a name and appearance so that PCs can always know who they are speaking with. The DM can generate a new NPC with the push of a button.
2. Random Encounter Generator: This takes the party size/level and creates a unique encounter that is properly balanced according to the D&D 5th edition rule set.

#### Initiative Tracker

The initiative tracker is designed to allow the DM to easily keep track of combat. The feature will allow the DM to input combatants, and the application will automatically order combatants, track the turn order, and track hit points.

## Minimal Feature Set

The application must have 5 features: Login, Built-in Character Sheets, Adventure Log, Random Generators (NPC and Encounter), and Initiative Tracker. This feature set encompasses the core features while expanding the list to include login functionality. This is required so players can have their character sheets saved and accessible.

## High Level Architecture

The application will use a modern full-stack JavaScript architecture optimized for performance and scalability with the following components:

#### Frontend/Backend

Frontend and backend development will be handled with Next.js. This JavaScript framework allows seamless handling of frontend user interface and backend API routes.

The frontend renders the responsive, mobile-first such as character sheets, adventure logs, and other tools. It will provide interactive user interface (UI) components such as those for generating NPCs and encounters.

The backend handles API requests for authentication, data operations, and content generation. The business logic lives here to generate random content and calculate stats.

#### Hosting Service

The application will be hosted by Vercel. This is the preferred hosting service for Next.js applications. It can be directly integrated with GitHub for CI/CD.

#### Database

MongoDB Atlas will be used to store data for the application. This will manage persistence so that users can store the data of their campaign and continue to come back to it later. MongoDB is a document database rather than a relational database. This is appropriate since the data to be stored is not inherently related

#### Authentication

Authentication will be managed by NextAuth.js. This is designed for Next.js applications and provides an integration for user authentication that is robust and easy to implement.

#### Architecture Diagram

A diagram of a software application

AI-generated content may be incorrect.

## High Level Data Management

Since MongoDB is not a relational database, there is no need to detail table relationships. Below are the key data that need to be managed.

* User data such as username and password
* Character data such as ability scores and proficiencies
* Session data such as dates and recaps
* Location data primarily text descriptions
* Quest data primarily text descriptions
* NPC data primarily text description
* Monster data such as experience points for generating encounters
* NPC data for random generation such as images and names
* Combatant data such as initiative and hit points