Software Requirements Specification Document

The Dungeoneer's Assistant

Version 2

09/17/19

Wizards of the Midwest ™

Submitted in partial fulfillment of the requirements of
IT 326 Software Engineering

Table of Contents

Revision History

Date	Description	Author	Comments
9/18/19	Version 1	Chloe Glass, William Paton, Seth Tummillo, Ryan Busch	Details for the project were established
9/19/19	Version 1.1	^	Requirements adjusted, UML diagram added, grammar
10/10/2019	Version 2	"	Added some diagrams and partially completed section 4

Document Approval

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

Signature	Printed Name	Title	Date
WP	William Paton		9/19/2019
RB	Ryan Busch		9/19/2019
ST	Seth Tummillo		9/19/2019
AT	Austin Tran		9/19/2019
CG	Chloe Glass		9/19/2019

1. Introduction

This document is designed to clearly portray the features, constraints, and technicalities involved with implementing the "Dungeoneer's Assistant" software suite via web browser. It has been organized to convey the purpose of the software and the guidelines that the developers should follow to construct and test the functional requirements. It also includes clarifications for terms, and considerations for those unfamiliar with the subject of D&D. This SRS serves as a blueprint for the "Wizards of the Midwest" team to design the project and ensure it adheres to the desired designs to the best of their ability.

1.1 Purpose

The SRS will describe the nature of the final product, list and describe the requirements necessary to build the project, and layout the dependencies, constraints and potential conflicts for the benefit of the Wizards of the Midwest and Dr. Rishi Saripalle.

1.2 Scope

1. Product
    a. The Dungeoneer's Toolkit
        i. Map Generator
        ii. Encounter Generator
        iii. Character Creator/ Tracker
        iv. Campaign Timeline
2. Functionality
    a. It will simplify tasks typically performed by hand on pencil and paper by both players and the DM. Examples of tasks are:
        i. Character Tracking
        ii. NPC Tracking
        iii. Creating and managing encounters
        iv. Keeping track of events
        v. Managing Inventories and Spells
    b. It will save resources created for several campaigns on a user account basis
    c. It will not be used as a tool to run an entire game digitally; this is a tool to aid an "in-person" game of Dungeons and Dragons
3. Goal
    a. The primary goal of this software is for it to be used before or during a game session. The software should make running a game easier for both the user running the game and the other users playing it.

1.3 Definitions, Acronyms, and Abbreviations

Ability Score: Each PC, NPC, and Monster has 6 Ability Scores that make the foundation of what they are capable of. These scores are:

Strength: measure of physical power

Dexterity: measure of agility and speed

Constitution: measure of damage resistance and endurance
Intelligence: measure of intellect and logic
Wisdom: measure of perception and common sense
Charisma: measure of personality and persuasion

Campaign: Overarching story, experienced over multiple sessions

Character Class: Each PC or NPC has a Character Class that determines how they interact with their possessions, skills, and world elements. These could include features that only that class has access to, or limitations on how that class may interact with in-game elements. An example being a Cleric having the unique ability to learn spells that can regenerate health but is restricted on using heavy weapons.

D&D(Dungeons and Dragons): Fantasy tabletop roleplaying game consisting of a single DM and up to several players.

DC(Difficulty Class): The difficulty of an action that a player is trying to accomplish.

DM(Dungeon Master): A role in a game of D&D who narrates the story, accepts player requests, generates challenges, enforces game mechanics, and generally controls all aspects of the game's direction and pace.

DMG(Dungeon Master's Guide): A document written by WotC with a focus on DM roles. These roles include rules specific to the DM, an expanded list of items and monsters, and general DM advice.

Encounter: An in-game event where the players interact with the world or engage in combat.

Item: Any physical possession that is owned by a PC or NPC.

Monster: Works similarly to an NPC but is not any of the race options for players and is usually involved in a combat encounter.

NPC(Non-Player Character): Any in-game character controlled by the DM, and not by any player.

One-Shot: Campaign designed to last one session.

PC(Player Character): Fictional character created by a player so the player can interact with the game world, other PCs, NPCs, and Monsters.

PHB(Player's Handbook): A document written by WotC with a focus on character creation, player rules, classes, races, spells, and a selection of items, monsters and general player advice.

Player: Any non-DM participant in the game, who controls a PC and determines their actions within the game.

Race: Each PC or NPC has a Race that determines physical characteristics as well as modifications to their abilities. For example, a Half-Orc would have unique physical characteristics based on their depiction in the game, as well as benefits to game stats such as Strength and Constitution.

Session: Stretch of time that players gather.

"Splitting the Party" - The PCs separate to explore different areas.

WotC(Wizards of the Coast): The current publisher of D&D 5th Edition.

XP(Experience Points): Points that are earned by the players at the discretion of the DM in order to level up and improve their player characters.

1.4 References

Wizards of the Coast. Players Handbook. 5ed. Hasbro, 2014.
Wizards of the Coast. Dungeon Master's Guide. 5ed. Hasbro, 2014.

Sources can be found online as pdf's or physical copies can be purchased or rented from retailers.

1.5 Overview

The rest of the SRS is broken into two sections: General Description and System Requirements. The General Description section is made up of five parts that address outside influences on the creation of the product.
The System Requirements Section contains requirements that the group will fulfill by the end of the project, and design constraints that come up as 'company' restraints.

2. General Description

The goal with our web application is to make managing and creating campaigns easier while also being intuitive for users to interact with. The website will provide tools to aid campaign creation such as an encounter generator and Map Generator, as well as providing tools to aid campaign tracking such as timelines and character sheets. Many of these terms will not sound familiar to those unacquainted with the game or the genre of role-playing games (RPG) in general.

2.1 Product Perspective

Many tools exist to help track the vast amount of information that goes into a game of D&D, but our product is aware of what the competition does; and our product will implement features that exist more uniquely while being more user-friendly, or features that have yet to be implemented into an online tool. The specifics can be viewed in the 'Specific Requirements' section below.

2.2 User Characteristics

Users will have an education level of Junior High or above. They will have minimum technical skills and will already have a fundamental understanding of how a game of D&D runs. Users must also be familiar with website applications and generally know how to navigate through one based on various widgets, such as buttons and scrolling.

2.3 System Environment

Our product will be a website hosted on a server via virtual box provided by ISU IT department and will be implemented/programmed in Python. Our database of choice is MongoDB. Any device capable of using a web browser will be able to execute the product; however, we will design the product mainly for use on a computer screen as there will be no mobile-friendly version. It is still possible to use on mobile devices but will be less user-friendly and/or intuitive with smaller screen estate available.

2.4 General Constraints

Our product will essentially be a collection of tools to aid a user. There will be no interactions between users. User information will be stored server side in a database. Modules should be able to operate independently of each other but may include functions to export data to another module.

2.5 Assumptions and Dependencies

We assume that the user is using our application on one of the more widely used web browsers such as Google Chrome, Edge, Firefox, and Safari.

3. Specific Requirements

3.1 Functional Requirements

The function requirements fall into five categories: Encounter Generator, Timeline Generator, Character Generator/Tracker, Map Generator, and Account.

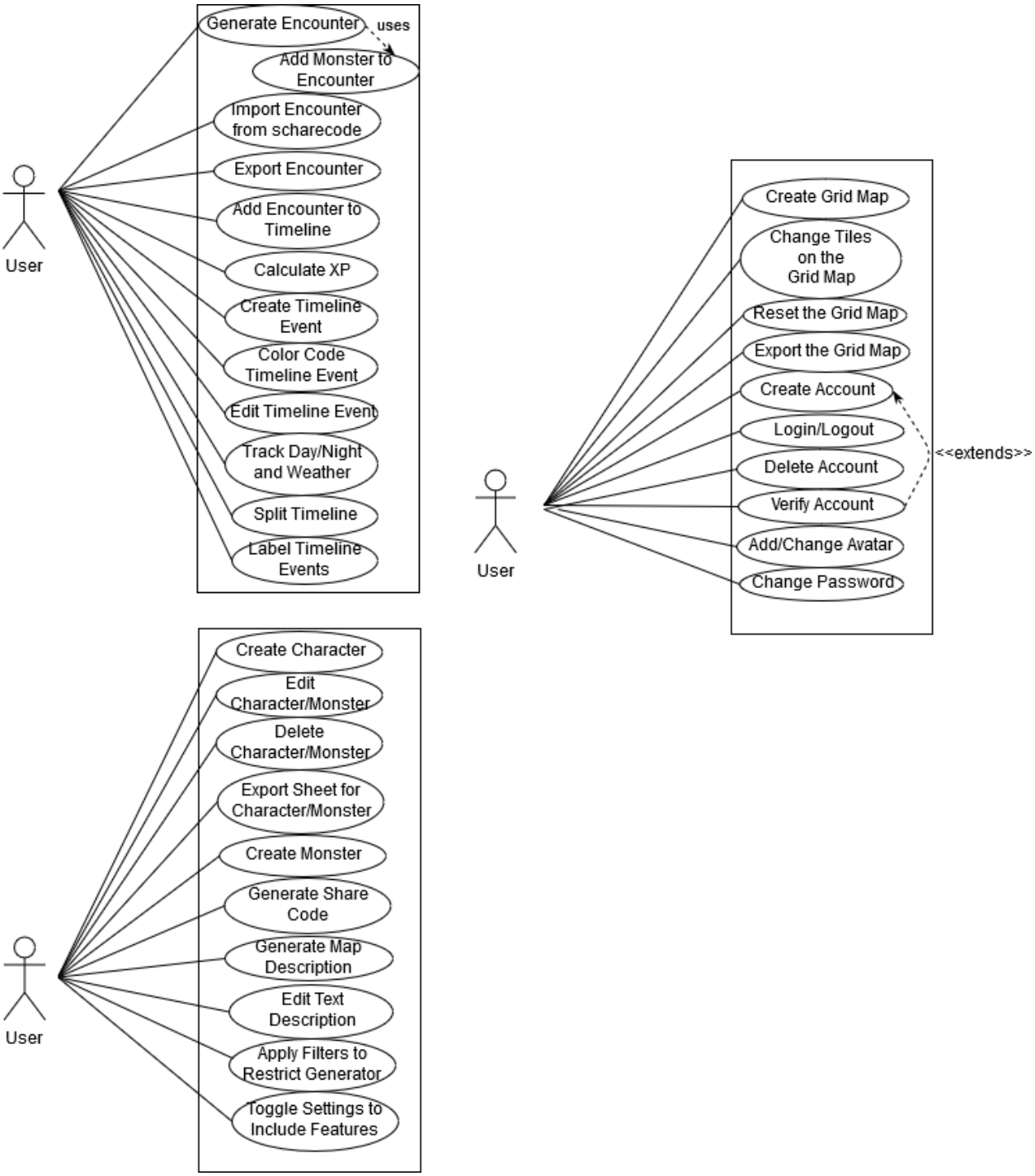
The Encounter Generator allows the user to randomly select a monster, submit a monster to the options, and enter criteria to reduce the options. The Encounter Generator will also determine rewards appropriate for defeating the monster.

The Timeline Generator provides a timeline for the user to create events on. Once the events have been created, the user can edit, delete, or label the events. Additionally, the user can track the day/night cycle and the weather or split the timeline to indicate simultaneous occurrences.

The Character Generator/Tracker allows the user to create a character or a monster, edit or delete them, export the character sheet to a PDF, and generate a share code so other users can access that character.

The Map Generator generates a text description based on a grid map. The user can edit the text. The user can select features they would like to include in the description or filters to restrict the generator. The tiles of the grid map can be clicked to change them. There will be a button so the map can be reset to its original state and an export button.

Account lets the user create an account, which cause an email to be sent for verification, delete the account, change their password, add and change an avatar, and login and logout.



3.1.1 Add Multiple Enemies to Encounter

Category - Encounter Generator

Description – The user will click a button that will display a menu for selecting a monster type. What this menu displays will be based on the biome selected and will appear as a dropdown menu. Keeping biome in the default position will display all possible enemies

Actor(s) – The user

Trigger – The user will click a button to add a new enemy

Conditions

Pre – The user must be in the encounter generator module

Post – The user must submit or cancel the enemy interface to either add it to the pool or cancel it. If they hit 'Submit', an enemy will be added. If they click 'Cancel', the menu will close and nothing will happen.

3.1.2 Generate an Encounter

Category - Encounter Generator

Description – The user will generate an encounter based on chosen fields such as the biome selector, difficulty level, enemies, and loot. After the requirements for the encounter are chosen, the program will generate an encounter to fit those requirements

Actor(s) – The user

Trigger – The user choses options from certain fields and then clicks a button to generate the encounter

Conditions

Pre – Variables are decided upon such as difficulty level, biome, encounter size, etc...

Post – The user finalizes the encounter or clicks out of the menu

3.1.3 Import Encounter from Share Code

Category – Encounter Generator

Description – Using a share code the user can generate an encounter with certain parameters

Actor(s) – The user

Trigger – The user clicks a button after inserting a valid share code

Conditions

Pre – The user must have a valid share code

Post – An encounter will be generated

3.1.4 Export an Encounter that has been Generated

Category - Encounter Generator

Description – We want to be able to export possible encounters using a share code. Essentially the system will generate a share code based on parameters in the encounter

Actor(s) –The user

Trigger – The user will click a generate share code button

Conditions

Pre – The user must be in the encounter generator module

Post – The user must click the chosen difficulty or close out of the encounter generator

3.1.5 Add Encounter to the Timeline

Category - Encounter Generator

Description – The user will click a button that will bring up a menu to then add a generated encounter to the timeline.

Actor(s) – The user

Trigger – The user will click a button to generate a menu.

Pre – The encounter that the user wants to share to the timeline must exist

Post – The encounter will be added to the timeline

3.1.6 Calculate XP Based on Encounter Parameters

Category - Encounter Generator

Description – Calculate the XP that is to be rewarded to the party for the successful completion of the encounter

Actor(s) – The user clicks a button to generate XP based on the encounter that has been generated

Trigger – The user clicks a button to generate XP

Conditions

Pre –The encounter must be generated by the user

Post – The user must save the encounter or close out of the encounter generator

3.1.7 Create Events on the Timeline

Category - Timeline Generator

Description – The user will click a button to create an event, then a form will pop up. Fields cover information such as which session the event occurred, when in the campaign it occurred, and what happened during the event

Actor(s) – The user

Trigger – The user will click a button to create a new event

Conditions

Pre – The user must be on the Timeline page

Post – The user must click a 'Create' or 'Cancel' button at the bottom of the form

3.1.8 Color Code Events

Category - Timeline Generator

Description – This will allow the user to apply a color to a timeline event. There will be a key on the page for the user to reference. This could be expanded to allow the user to determine the colors and meanings for the color-coding system

Actor(s) – The user

Trigger – The user will either click a button on the timeline event or enter a mode that allows them to edit the colors of the timeline event

Conditions

Pre – The timeline event must already exist

Post – The user must click out of the menu or click a button to close the menu

3.1.9 Edit Timeline Event

Category - Timeline Generator

Description – The user will click a button on the timeline event that will bring up a form like that of the 'Create Event' menu. The user will be able to alter the information of the fields. When they are done, they can save their changes, cancel their changes, or delete the timeline event

Actor(s) – The user

Trigger – The user will click a button on the timeline event

Conditions

Pre – The timeline event must already exist

Post – The user must click a 'Save', 'Delete', or 'Cancel' button at the bottom of the form

3.1.10 Track Day/Night and Weather

Category - Timeline Generator

Description – An indicator will be displayed alongside the timeline, showing the time of day and the weather

Actor(s) – The user

Trigger – The user will click a button or an alternative widget to toggle the time of day and weather

Conditions

Pre – The timeline event must already exist

Post – The user must click out of the menu or click a button to close the menu

3.1.11 Split Timeline to Follow Individuals

Category - Timeline Generator

Description – The user can enable a timeline split to indicate the party has split or perhaps track the actions of important NPCs. This will create a new timeline event in the same row as the parallel event. Deleting the split events will return the timeline to normal

Actor(s) – The user

Trigger – The user clicks on a button either in the edit menu or along the timeline

Conditions

Pre – There must be at least one timeline event

Post – The user must click a 'Create' or 'Cancel' button at the bottom of the form

3.1.12 Label Events by Character

Category - Timeline Generator

Description – When the user clicks the button, a menu will pop up. The user will then choose an icon to tag the event either with the name of a character or an image that represents that character

Actor(s) – The user

Trigger – The user must click a button on a timeline event

Conditions

Pre – The timeline event must exist or be in the process of being created

Post – The user must click out of the menu

3.1.13 Create Character

Category - Character Generator/Tracker

Description – Creating a character sheet from a template to be used in a game of D&D

Actor(s) – The user

Trigger – User clicks on the "Character Creator" module they are given the option to begin creating a new character

Conditions

Pre – The user must be prepared to create their character in a single sitting, along with general knowledge of what they wish to create

Post – The user must choose whether they'd like to save this character to their account, or export their creation to a PDF

3.1.14 Edit Character/Monster

Category - Character Generator/Tracker

Description – After a character has been created with permanent attributes, the user is able to revisit their character/monster and edit dynamic attributes (Ability Scores, Item Inventory, Health)

Actor(s) – The user

Trigger – After the user clicks on the "Character Creator" module, they are given a list of characters (if any) that they've previously created. These can be selected for editing

Conditions

Pre – The user must have already built a character and saved it to their account

Post – Any changes the user has made must be saved to their character

3.1.15 Delete Character/Monster

Category - Character Generator/Tracker

Description – After a character/monster has been created, the user is able to delete them from their account

Actor(s) – The user

Trigger – After the user clicks on the "Character Creator" module, they are given a list of characters/monster (if any) that they've previously created. These can be selected for deletion

Conditions

Pre - The user must have already built a character and saved it to their account

Post – N/A

3.1.16 Export a Character Sheet

Category - Character Generator/Tracker

Description – After a character has been created, the user is able to export their character to a pdf that they can print and use in their game

Actor(s) – The user

Trigger – After the user clicks on the "Character Creator" module, they are given a list of characters (if any) that they've previously created. These can be selected for exporting

Conditions

Pre – The user must be familiar enough with the PDF format to download and print their sheet

Post – N/A

3.1.17 Create a Monster

Category - Character Generator/Tracker

Description – Creating a monster using from a template for use by the DM in a game of D&D

Actor(s) – The user, specifically a DM

Trigger – The user clicks on the "Character Creator" module they are given the option to begin creating a monster

Conditions

Pre – The user must be familiar enough with the format of monsters and be able to complete this sheet in a single sitting

Post – The user must save their changes

3.1.18 Generate a Character/Monster Share Code

Category - Character Generator/Tracker

Description – Generate a string that can be used by the Character/Monster Generator to instantly create a predeveloped Character/Monster to aid with sharing creations

Actor(s) – The user

Trigger – After the user clicks on the "Character Creator" module, they are given a list of characters (if any) that they've previously created. These can be selected for Share Code generation

Conditions

Pre – The user must already have a Character/Monster that they can develop their share code for

Post – The user must copy this code to their clipboard to either share it or store it in a text file

3.1.19 Generate a Map Description

Category - Map Generator

Description – Generate a list of key details from a map to assist players with their immersion

Actor(s) – The user

Trigger – In the Map Generator tool, there will be a "Generate Description" button the user will click

Conditions

Pre – The user must have a map created

Post – The user must click "OK" on the description pop-up to continue with Map Generation

3.1.20 Edit Text Description

Category - Map Generator

Description – Edit the description generated by Map Description

Actor(s) – The user

Trigger – There will be a button under the "Generate Description" button called "Edit Description" that must be clicked after generating a description to edit that description

Conditions

Pre – The user must have already generated a description and be ready to edit that description

Post – The user must save their changes to the description

3.1.21 Apply Filters to Restrict Generator

Category - Map Generator

Description – The user can select a biome or conditions, which will affect what the generator displays for the text description

Actor(s) – The user

Trigger – There will be a menu with fields on it.

Conditions

Pre – The user must be on the Map Generator page

Post – The user must click an 'Apply' or 'Cancel' button

3.1.22 Toggle Settings to Include Features

Category - Map Generator
Description – Toggling the visibility of certain elements on the map

Actor(s) – The user

Trigger – The user will click on a checkbox for each toggle (show monsters, show terrain, show traps, etc.)

Conditions

Pre – The user must already have a map generated, with the graphics they wish to toggle

Post – The user may choose to de-toggle those features to reenable their visibility

3.1.23 Set Effects Based on Biome/Conditions

Category - Map Generator
Description – If certain biomes are selected, then there will be effects on the player

Actor(s) – The user

Trigger – Based on filters that the user has selected, the text description will include effects on characters

Conditions

Pre – The text description must have been generated and filters selected

Post – N/A

3.1.24 Create a Grid Map

Category - Map Generator
Description – Create a blank grid map template to begin editing your map
Actor(s) – The user

Trigger – The main page will have a Map Generator button, and after clicking that you are given the option to create a new map

Conditions

Pre – The user must click Generate Map, have general map-making knowledge, and be ready to finish their creation in a single sitting

Post – The user must save their changes

3.1.25 Change Tiles on the Grid Map

Category - Map Generator
Description – Change each tile to represent a specific terrain tile

Actor(s) – The user

Trigger – The user will right click the desired tile they wish to change and select from a dropdown list

Conditions

Pre – The user must understand what they want before changing

Post – The user must save their changes

3.1.26 Reset the Grid Map

Category - Map Generator
Description – Reset the map back to a template

Actor(s) – The user

Trigger – There will be a button on the Map Generator screen called "Reset Grid"

Conditions

Pre – The user must be ready to reset their map

Post – The user must save their changes

3.1.27 Export the Grid Map

Category - Map Generator
Description – Export the Grid Map to PDF form for printing or sharing

Actor(s) – The user

Trigger – There will be a button on the Map Generator screen called "Export Grid" which will convert the map to a PDF and

Conditions

Pre – The user must be familiar with the PDF format and have the space and permissions to download the PDF file

Post – The user must access their PDF for printing or file-sharing

3.1.28 Create Account

Category - Account
Description - This requirement will allow a user to register an account on our website.
Actor(s) – The user

Trigger – Clicking button to create account

Conditions

Pre - None

Post – The user will have an unverified account

3.1.29 Login/Logout

Category - Account
Description - The user should be able to login to our website to gain access to features and then be able to log out to close their session
Actor(s) – The user

Trigger – The user clicks on a login button

Conditions

Pre – The user must have an existing account

Post – The user will be able to access website functions if verified

3.1.30 Delete Account

Category - Account
Description - The user should be able to delete their account if they wish. This should also require confirmation stating what they are about to do.
Actor(s) – The user

Trigger – Press 'Delete Account' button

Conditions

Pre – The user must be logged in and have an existing account

Post – The user will no longer have an account associated with our website. All information related to their account will be permanently deleted.

3.1.31 Verify Account via Email

Category - Account
Description – System sends email to user to confirm account creation
Actor(s) – The user

Trigger – Triggered after account creation

Conditions

Pre – The user must create an account

Post – The user will have a verified account

3.1.32 Add and Change Avatar

Category - Account
Description – Allow users to customize their account
Actor(s) - The user

Trigger – The user clicks on their character portrait

Conditions

Pre – Must have a verified account

Post – Portrait must be changed

3.1.33 Change Password

Category - Account
Description – Allow user to change their password
Actor(s) – The user

Trigger – The user clicks change password button

Conditions

Pre – The user must have a verified account

Post – The user's password should be different

3.3 Non-Functional Requirements

3.3.1 Performance

Each webpage and module should load in a reasonable time without interfering with the end-user's other processes.

3.3.2 Reliability

Any creations that users have produced must be reproducible by the tools whenever the source file is uploaded to its corresponding tool. Our website uptime depends upon the quality of the virtual box received by ISU IT, which could fail due to hardware abnormalities or an online error when the virtual box may be having networking issues.

3.3.3 Availability

The website and its database must be able to fetch webpages and account information with 100% accuracy at any point in time. However, if there were networking or technical issues on the side of MongoDB this would be out of the control of our system. Our product will back up the data stored by users interacting with the system in case changing databases is necessary in the future.

3.3.4 Security

Account access shall be password restricted to prevent users from altering the accounts of others. Passwords should meet requirements and will be encrypted.

3.3.5 Maintainability

The system must include a reference document for each module to document how the code operates.

3.3.6 Portability

The system must be able to run on all major web browsers regardless of operating system or end-user hardware. Our website will be thoroughly tested in multiple web browsers to ensure users interacting with the system have the best experience regardless of their choice of browser. Our code will be published in collaboration on GitHub will is accessible from any computer given one of the members provides their credentials, which allows us to easily port the programming progress if for example a different way of hosting the server was desired.

3.4 Design Constraints

This toolkit must be used to ease the use of WotC documents, but not replace them.
The functionality of this toolkit relies on the uptime of our servers and consistent internet access from the end-user.
The navigation of the site must not result in fail state where the user is unable to continue editing their creations or reach a desired webpage.

4. Design & Development

This section is to outline how our team plans to execute the design and development of the "Dungeoneer's Assistant" application. It will cover the software process model that we chose to follow and justify it. It will also show a diagram of this model that will provide details to our processes such as our sprint schedule and meeting frequency.

Our team plans to execute our application through the Scrum model and its multiple implementations of weekly and/or bi-weekly sprints. Our group is most familiar with this model from past experiences and thought it would be best fit given our other options. More specific details below.

4.1 Software Process Model

4.1.1 Introduction

This section will briefly describes the process model chosen and any background information required to understand the rest of this section.

4.1.2 Process Model

The process model we have chosen for the development of our application is the Scrum Model. Scrum allows us to pick a feature to be expanded upon in sprints which we have chosen to last 1-2 weeks. We also have daily scrum standups which allow consistent updates and feedback to be recorded and monitored on weekdays Monday through Friday. Currently we hold meetings on Wednesdays at Milner at 3:30 PM which can last anywhere from thirty minutes to two hours, depending upon what needs to be discussed and accomplished as a group.

4.1.3 Plan Description

Our focus for the first couple of sprints is the planning and designing of our application, while the following sprints will focus more heavily on the actual code integration and the setup of a collaborative environment with GitHub to share our code repositories respectively. We expect to hold virtual/online meetings during later sprint stages since we all have our own components to work on and are all comfortable with the online communication space platform.

4.1.4 Plan Diagram

This elaborates on the plan description in the form of a diagram with the exact details related to the plans such as sprint schedule, meeting timeframes, cycle time (1 week, 2 weeks, etc.)

4.2 Sprint 1

4.2.1 Description

The focus of our first sprint is to create the class, sequence, activity, and state diagrams to help visualize our application and see if any necessary changes became apparent as we finalize our models. We plan on revising our diagrams after the first sprint, there may be small tweaks and/or improvements in the second or third sprint(s). Revisions to our SRS have been made to make any unclear ambiguities more specific and concise.

4.2.2 Satisfying Requirements

This sprint is focused on creating rough versions of diagrams, which will be improved as we continue to figure out what content will be needed to achieve our requirements

4.2.3 Design

4.2.3.1 Description

We focused on a simple design for the first sprint, as we felt the features we were implementing would be much further expanded on and integrated down the line in order to work with the rest of our use cases.

4.2.3.2 Standards, Frameworks and Tools

Tools we used in the creation of the design of the sprint were modeling tools from Microsoft Visio, and a free modeling website, gliffy.com. Standards we followed in the design of features of the sprint was UML.

4.2.3.3 Class Diagram

The class diagram showed in Figure 1 is an early draft. The variables for each class have yet to be discussed, as do the types for the methods.

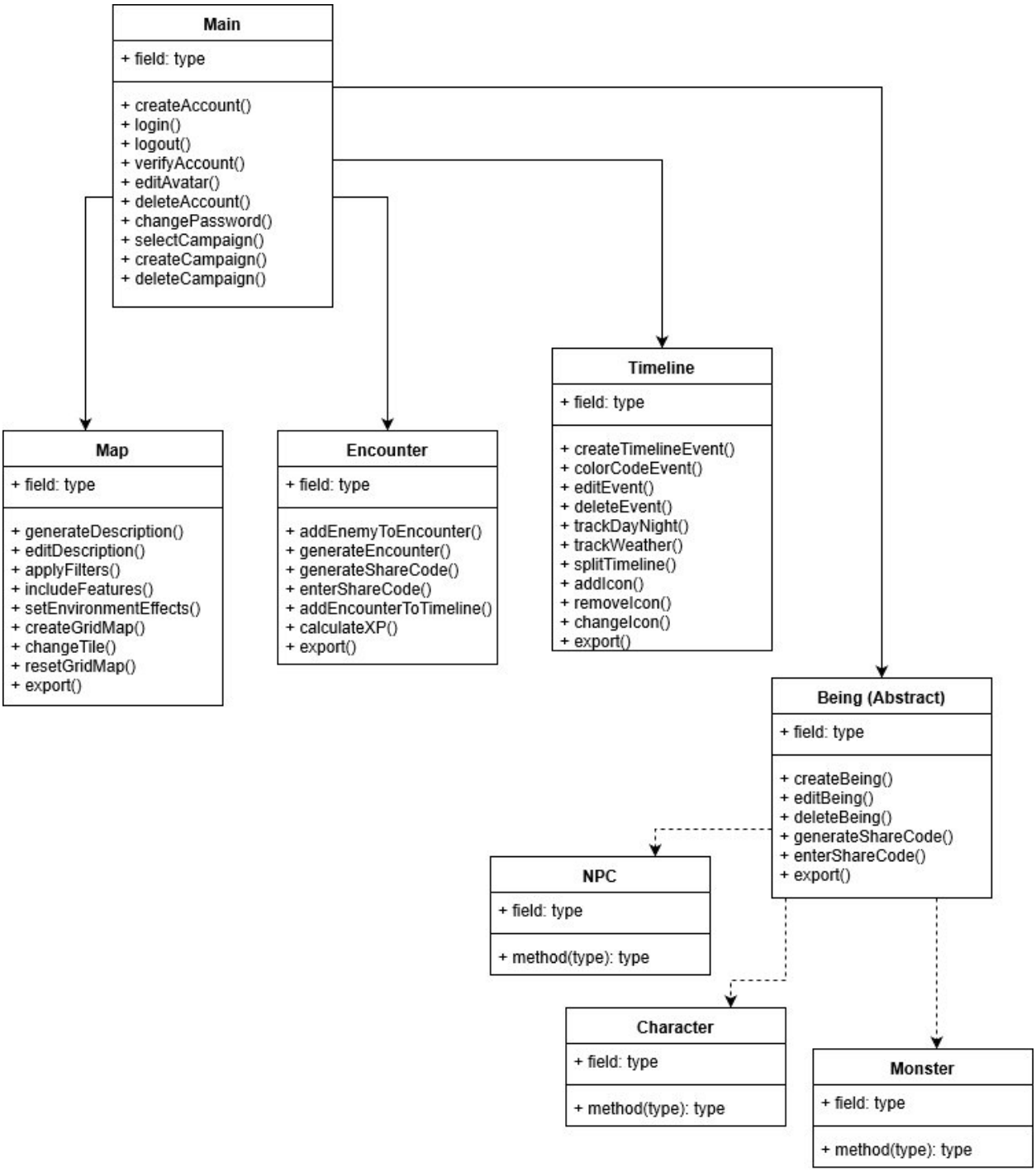


Figure 1: Class Diagram

4.2.3.4 Activity Diagram

The activity diagram shown in Figure 2 explains the activities involved and their flow for ordering an equipment.

Figure 2: Equipment checkout Activity Diagram

4.2.3.5 Sequence Diagram

The sequence diagram shown in the Figure 3 illustrates the messages exchanged between the object of type "MaintaenanceEmployee" and object of type "System".

Figure 3: Checkout Sequence Diagram

4.2.3.6 Navigation Graph

The Navigation Graph shown in Figure 4 shows the navigation between pages on the site.

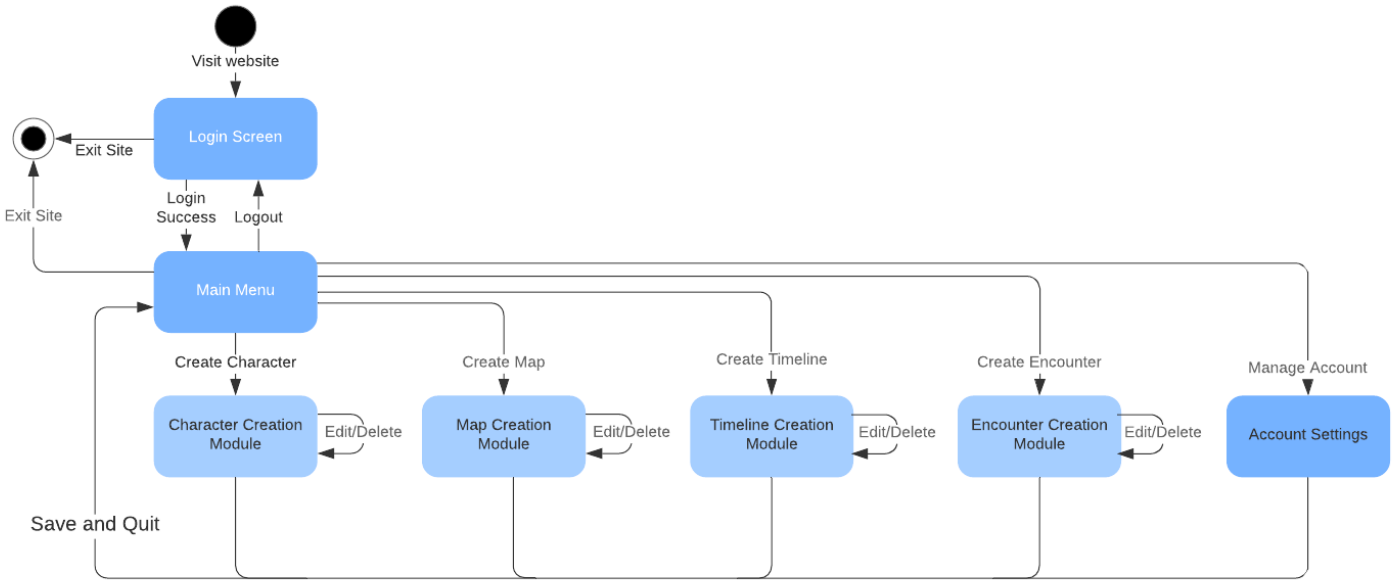


Figure 4: Site Navigation Graph

4.2.4 Development

4.2.4.1 Description

The features we developed in the first sprint was a simple class diagram and setting up the database for persistence of objects. These aspects we felt were vital to the later successful implementation of the core aspects of the application.

4.2.4.2 Standards, Frameworks and Tools

For the development aspect of the sprint, we used a Java IDE in order to create the classes we needed. We used the latest version of NetBeans, version 8.0.2, and Eclipse version 4.4.

4.2.4.3 Screenshots