**Project Plan**

***DM-Dashboard***

*Dungeon Masters around the World.*

|  |
| --- |
| **Date : 27-09-2023** |
| **Version : V2.0** |
| **State : Initial Start** |
| **Author : Ryan van Wegen** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| V1.0 | 27-09-2023 | Ryan v. Wegen | Initial start of the Project Plan Document, worked on Context & Goal of the project |  |
| V2.0 | 28-09-2023 | Ryan v. Wegen | Worked on Scope and Preconditions, Strategy, Research Questions and Methodology, End Products, Stakeholders, Timeplan and Milestones, Risk and Mitigation | Note: sections outlined in red to be updated when it’s known. |
|  |  |  |  |  |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
|  |  |  |
|  |  |  |

Contents

[1. Project assignment 4](#_Toc42673512)

[1.1 Context 4](#_Toc42673513)

[1.2 Goal of the project 4](#_Toc42673514)

[1.3 Scope and preconditions 4](#_Toc42673515)

[1.4 Strategy 4](#_Toc42673516)

[1.5 Research questions 4](#_Toc42673517)

[1.6 End products 4](#_Toc42673518)

[2. Project Organisation 6](#_Toc42673519)

[2.1 Stakeholders and team members 6](#_Toc42673520)

[2.2 Communication 6](#_Toc42673521)

[3. Activities and time plan 7](#_Toc42673522)

[3.1 Phases of the project 7](#_Toc42673523)

[3.2 Time plan and milestones 7](#_Toc42673524)

[4. Testing strategy and configuration management 8](#_Toc42673525)

[4.1 Testing strategy 8](#_Toc42673526)

[4.2 Test environment and required resources 8](#_Toc42673527)

[4.3 Configuration management 8](#_Toc42673528)

[5. Finances and Risk 9](#_Toc42673529)

[5.1 Project budget 9](#_Toc42673530)

[5.2 Risk and mitigation 9](#_Toc42673531)

# Project assignment

## Context

I’m Ryan van Wegen, a student of S3\_CB02 and need to create a full stack application for this semester. I’m a gamer by heart and decided to do something I was very interested in, because this would help me finish the application and try to think outside of the box and come up with a lot of functionality. I decided to create something for the world of Dungeons & Dragons (D&D or DnD).

Dungeons and Dragons is a fantasy tabletop role-playing game. Each player in the game creates their own character and together with other players you embark upon adventures within a fantasy setting. A DM serves as the game’s referee and storyteller, while maintaining the setting in which the adventure occurs, and playing the role of the inhabitants of the game world, known as non-player characters (NPCs). The characters form a party, and they interact with the setting’s inhabitants and each other. Together they solve problems, engage in battles, explore, and gather treasure and knowledge. In the process, player characters earn experience points (XP) to level up and become increasingly powerful over a series of separate gaming sessions. Players choose a class when they create their character, which gives them special perks and abilities every few levels.

## Goal of the project

The goal of the project is to create an application that will help not just myself but a lot of other Dungeon Masters (DM) around the world that are playing Dungeons & Dragons (D&D or DnD). During a game of D&D there is always going to be combat, where the players fight other monsters in the story or other NPCs. During this combat as a DM, it’s always hard to keep track of the combat and what every monster or NPC can do and how much Health Points (HP) everyone has left. I thought it would be a good idea to create a program that keeps track of everything there is to do during combat and be able to customize it.

## Scope and preconditions

*<<What activities and which end products (to what extent or quality) belong to the project, and which don’t.>>*

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. A use-case that requires user authentication with the system (e.g., logging in as a user) 2. A use-case that requires user authorization (JWT role-based authorization in the backend) |  |
| 1. A use-case where the user retrieves detailed data about a database object (e.g., a specific invoice) 2. A use-case where the user retrieves aggregated data (e.g., total sales in the last quarter). 3. A use-case where the user searches using filters (e.g., get list of invoices above EUR100,00 in the past 5 days) 4. A use-case where the user modifies an existing database object 5. A use-case where the user deletes an existing database object 6. A use-case where the user creates a new database object 7. A use-case that is implemented via Websockets (e.g., chat or notification) 8. The program has to be as user-friendly as possible. 9. The application should be sufficient enough to keep me busy for a semester. 10. The application has a Javacript framework based front-end. 11. The application has a RESTful API with relational databased back-end. 12. ORM will be used for database access. 13. React will be used for the front-end 14. Java Spring Boot REST API back-end 15. Database T.B.A. |  |

## Strategy

For this project I will go with the Agile methodology for a couple of reasons:

1. Agile is highly adaptive. Which means it allows for changes to be made even late in the development process. This is useful because I might get better understanding later about certain topics of the project and coding wise.

2. Agile has incremental development, which means I can release a functional part of the software in short cycles. This will help me because my teachers will be able to guide me during each sprint.

3. Agile has continuous User Involvement. Agile encourages regular and continuous collaboration with end-users. Since I know a couple of DMs it will be good to get feedback during the entire process. This can help the development process and ensure the tool meets the specific needs and preferences.

4. Agile also has some sort of early delivery. This works well as a school project because you always must deliver something each sprint. And can be advantageous for the stakeholders wanting to use the program as soon as possible.

## Research questions and methodology

1. What are the key needs and preferences of Dungeon Masters regarding combat management in D&D?

* It’s important for the DM to keep track of all the monster’s, npc’s and player’s health.
* DMs need to keep track of what abilities everyone has.
* DMs need to keep track of what items everyone has.
* DMs need to keep track of all the stat bonusses on each item and character.
* DMs need to be able to keep track of the initiative of everyone during combat.

The methodology for that question is that I myself am a Dungeon Master, so I know most of these things. I will also scour the internet for additional information or ask some of the other Dungeon Masters I know what else I could be missing out on.

1. How can Java and React be used effectively to create a seamless and responsive user interface?

The methodology will be mostly experimenting with different integration approaches, but also to gather feedback from the teachers during the project and maybe classmates.

1. How should the dashboard be designed for optimal usability and a positive user experience for other DMs.

The methodology will be to conduct usability testing with prototype versions of the dashboard, gather feedback on user interactions and iterate on the design based on that user’s input.

## End products

1. Project Initiation

- Define Project Scope

2. Research and Planning

- User Needs Analysis

- Technology Research (Java & React)

- Create Project Plan

3. Development Environment Setup

- Set up Java Development Environment

- Set up React Development Environment

- Establish Version Control (Git)

4. Backend Development (Java)

- Implement backend logic

- Implement User Authentication

- Define Database Objects and Relationships

5. Frontend Development (React)

- Design User Interface (UI)

- Implement UI Components

- Integrate with Backend Services

- Implement User Authentication in the Frontend

6. Database Operations

- Retrieve Database Object

- Modify existing Database Object

- Delete existing Database Object

- Create new Database Object

8. Search Functionality

- Implement Search Functionality with filters

9. WebSocket Implementation

10. Iteration and Refinement

11. Documentation

12. Deployment

# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Dungeon Masters* | *DMs* | *End Users: They utilize the final product.*  *And their responsibility (including myself) is to provide feedback on usability and functionality* | *N.A.* |
| *Teachers* | *N.A.* | *Project Manager: Oversees the project from initiation to closure* | *Monday, Wednesday, Friday 9 am – 4pm* |
| *Ryan van Wegen* | *Ryan* | *Project Manager: Leads the overall project. Responsibilities include planning, coordination and communication.* | *N.A.* |
| *Ryan van Wegen* | *Ryan* | *Full stack developer: Develops the Frontend and Backend logic.* | *N.A.* |
| *Ryan van Wegen* | *Ryan* | *Product Owner: Manages the product backlog.* | *N.A.* |

# Activities and time plan

## Phases of the project

There are quite a couple of phases of the project but since we are using agile these phases may overlap, and they may go through several iterations of development and testing in shorter cycles:

1. Initiation
2. Planning
3. Design
4. Developent
5. Testing
6. Deployment
7. Documentation
8. Feedback and Iteration
9. Completion and Review

There is also going to be continuous improvement, meaning that throughout the project I will allow for some adjustments based on the changing requirements or new insights.

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 1. Sprint 1 |  | Sept 7 | Sept 22 |
| 1. Sprint 2 |  | Sept 28 | Oct 13 |
| 1. Sprint 3 |  | Oct 14 | Nov 10 |
| 1. Sprint 4 |  | Nov 11 | Dec 1 |
| 1. Sprint 5 |  | Dec 2 | Dec 22 |
| 1. Sprint 6 |  | Dec 23 | Jan 19 |

# Testing strategy and configuration management

## 

## Testing strategy

*<<Which testing strategy do you envision? E.g., on which levels will testing take place? Consider that you could choose unit, component, integration, system, or acceptance testing.*

*Justify your strategy, and also set goals where relevant. E.g., percentage code coverage for the relevant unit tests. For each of the planned tests, indicate what will be automated and what not.*

*Also think of quality testing setups like, e.g., Sonarqube.*

*>>*

## Test environment and required resources

*<< Describe the test environment. E.g., do you envision a DTAP (Development, Testing, Acceptance, Production) environment. Can you make use of a CI/CD environment or will you develop your own?*

*It often helps to use a picture to visualize the test environment.*

*If you already know, describe which resources are required for realization and testing. Think of hardware, cloud environments and specific tooling required for development and testing.*

*>>*

## Configuration management

*<< Describe the project approach with respect to version management (e.g. your GIT repository). This might include things like tooling, branching strategy, promotion-, release- and baseline strategy.*

*Also, when relevant, think of a mechanism to deal with change requests and problem reports.>>*

# Risk and mitigation

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Unforeseen technical challenges during the integration of Java and React | Keep practicing with Java and React and learn about new things | Get help from the teachers or other people who are more experienced with the programming languages |
| 1. Changes in user requirements | Conduct a comprehensive requirement analysis before the project starts | Implement an Agile development approach to accommodate for the changes |
| 1. Insufficient resources (e.g., testers) | Keep in communication with other DMs | Adjust project timeline based on the resource availability |
| 1. Not enough communication with the teachers | Go to school enough to see them each week | Schedule when I’m going to be at school |
| 1. The gradual expansion of project scope beyond the initial requirements | Clearly define the scope during the planning phase | Prioritize changes based on project goals and deadlines |
| 1. Insufficient testing leading to undetected bugs | Conduct enough testing during the process | After a version release do enough user testing |