**Project Plan**

***MetaRate***

*Individual project*

|  |
| --- |
| **Date : 3-3-2023** |
| **Version : 1.0** |
| **State : In progress** |
| **Author : John Ooi** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1.0 | 3-3-2023 | John | Starting | In progress |
| 1.2 | 16-3-2023 | John | Added risks + test strategies | In progress |
| 1.3 | 24-3-2023 | John | Improved some stuff | In progress |
| 1.5 | 18-6-2023 | John | Updated and finished it | Finished |

Contents

[1. Project assignment 4](#_Toc129941811)

[1.1 Context 4](#_Toc129941812)

[1.2 Goal of the project 4](#_Toc129941813)

[1.3 Scope and preconditions 4](#_Toc129941814)

[1.4 Strategy 4](#_Toc129941815)

[1.5 Research questions and methodology 5](#_Toc129941816)

[1.6 End products 5](#_Toc129941817)

[2. Activities and time plan 6](#_Toc129941818)

[2.1 Phases of the project 6](#_Toc129941819)

[2.2 Time plan and milestones 6](#_Toc129941820)

[3. Testing strategy and configuration management 7](#_Toc129941821)

[3.1 Testing strategy 7](#_Toc129941822)

[3.2 Test environment and required resources 7](#_Toc129941823)

[3.3 Configuration management 7](#_Toc129941824)

[4. Risk 8](#_Toc129941825)

[4.1 Risk and mitigation 8](#_Toc129941826)

# Project assignment

## Context

MetaRate is a web application that provides information related to video games. With this application, users can find out more about a game. The main focus of this application is allowing users to rate and review video games. Admins will be able to add new games to the database, users will then be able to rate that game and review it. For each game there will be a chat, which will enable users to meet people who share their interests.

## Goal of the project

The new preferred situation would be a fully functional MetaRate website that allows users to browse, search, and filter reviews and ratings, as well as submit their own reviews and ratings. The website will also feature user profiles and game profiles.

The advantages of this project include providing a valuable service to users and establishing a brand in the entertainment industry.

The ICT product offers the possibility to aggregate and display reviews and ratings from various sources, such as professional critics, user reviews, and social media. The project will realize this possibility by developing a data pipeline to collect, clean, and store reviews and ratings, as well as a website and API to display and interact with the data.

## 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. Creating a database to store user reviews and ratings. | 1. Developing mobile applications for iOS or Android platforms |
| 1. Testing the website for usability, functionality, and security. | 1. Building a machine learning model to predict user ratings. |
| 1. Deploying the website to a web server or cloud platform. | 1. Providing legal advice on copyright or trademark issues. |
| 1. A fully functional website that allows users to search for and review software products. |  |
| 1. Incorporating APIs to get information |  |

The application will be a full stack application. HTML, CSS, JavaScript and React will be used for the frontend. The backend will be built with the Springboot Framework, written in Java.

## Strategy

For the MetaRate project, an Agile approach, specifically Scrum, will be used. It provides a flexible and adaptive approach to software development, which is especially useful for complex projects with evolving requirements.

At the beginning of each sprint, a set of tasks will be selected from the backlog, which are then implemented and tested during the sprint. At the end of each sprint, an assessment with the client is held to discuss the completed work, and a retrospective is held to reflect on the process and identify areas for improvement.

## End products

A diagram of a project

Description automatically generated with medium confidence

# Activities and time plan

## Phases of the project

At the start of every sprint, a backlog of tasks will be created, everything will be planned out and everything will be made according to the planning of when the features that need to be implemented will be taught.

Every week there will be a meeting with both teachers to ask for feedback regarding implementations and in case there are any questions.

At the end of each sprint a small demonstration of what has been implemented will be done and any outstanding issues will be addressed. In case of any unsatisfaction for some implementations, those implementations will be added to the backlog and will be improved upon.

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 1. Sprint 1 | 5/10 | Week 1 | Week 3 |
| 1. Sprint 2 | 6/10 | Week 4 | Week 6 |
| 1. Sprint 3 | 6/10 | Week 7 | Week 9 |
| 1. Sprint 4 | 8/10 | Week 10 | Week 12 |
| 1. Sprint 5 | 8/10 | Week 13 | Week 15 |
| 1. Sprint 6 | 7/10 | Week 16 | Week 18 |

# Testing strategy and configuration management

## Testing strategy

For this project, the testing strategy will focus on unit testing. Unit tests will be written for the components inside the business layer of the application.

The goal for testing is the following: ensure that all components are functioning correctly and integrate seamlessly with each other.

All unit tests will be automated. This will ensure that tests are consistent and can be run easily as part of the CI/CD pipeline. Manual testing will also be performed during the development process.

Integration, user acceptance and end to end testing.

## Test environment and required resources

A CI/CD pipeline will be used for development, testing, and deployment. The pipeline will be defined in a YAML file, which will describe the various stages and steps in the pipeline. Whenever something new gets pushed to the git repository, the pipeline will get triggered and everything will be automatically tested.

To realize and test the project, the following resources will be required:

* CI/CD tooling: A CI/CD tool such as GitLab CI/CD will be used to implement the pipeline.
* Database: A database will be required for storing application data, and a suitable database engine such as MySQL will be used.
* Runner: The pipeline jobs will be carried out using a GitLab runner.
* Shell: Shell scripts will be used to carry out various pipeline processes. The pipeline will be defined using a YAML file.

## Configuration management

The project will use the Git version control system and store the code on the GitLab. The Gitflow branching strategy will be adopted, which involves maintaining a "main" branch for production-ready code and a "develop" branch for ongoing development. Feature branches will be created off of the "develop" branch, and merged back into "develop" via pull requests once the feature is completed and reviewed.

# Risk

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Technical issues during development | Regular testing and quality assurance of software development | Backup plan for rolling back to previous versions |
| 1. Unforeseen changes in project requirements | Close collaboration with stakeholders to clarify project scope | Re-evaluation of project requirements and timeline |
| 1. Inability to meet project deadlines | Establish clear channels of communication and regular updates | Schedule regular meetings with stakeholders for feedback |
| 1. Lack of technical expertise or skills | Identify skills gaps and allocate resources for training | Consult a teacher for help |
| 1. Poor user adoption or satisfaction | Conduct user testing and gather feedback throughout development | Address issues identified through user feedback |