**Project Plan**

***Disx***

|  |
| --- |
| **Date : March 18th 2024** |
| **Version : v0.1** |
| **Author : Deren Serce** |

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)

[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 & environment 8](#_Toc42673526)

[4.3 Configuration management 8](#_Toc42673528)

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

[5.1 Risk and mitigation 9](#_Toc42673531)

# Project assignment

## Context

Disx is an emerging music platform designed to connect music enthusiasts, facilitate discussions, and enable users to discover new music. It aims to create a vibrant online community where users can share their passion for music, engage in meaningful discussions, and explore a wide range of musical genres.

## Goal of the project

The goal of the Disx project is to develop and launch a user-friendly and engaging music platform that fosters community interaction, enhances user engagement, and promotes music discovery. The project aims to create a preferred situation where music enthusiasts can connect seamlessly, share their favorite tracks, participate in discussions, and discover new music recommendations. By offering a dynamic and interactive platform, Disx seeks to add value to the music community by providing a centralized hub for music lovers to connect and engage with each other.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Development of core features such as user registration, profile creation, music discussion forums, and music ranking systems. | 1. Physical music distribution or sales. |
| 1. Tests | 1. Licensing agreements with music labels or artists. |
| 1. Research |  |
| 1. UI/UX design |  |
| 1. Documentation |  |
| 1. Github version control |  |

## Strategy

The project will adopt an agile approach, specifically Scrum methodology, to ensure flexibility and adaptability throughout the development process. This iterative approach allows for continuous feedback and collaboration between stakeholders, enabling quick adjustments to evolving requirements and priorities. Scrum's emphasis on delivering working software in short iterations aligns well with the project's goal of creating a responsive and user-centric music platform.

## Research questions and methodology

### **Main Question**

“What strategies can be employed to effectively enhance user engagement and activity in online music communities?”

### **Sub-questions**

* “What are the key factors influencing user engagement within online music communities?”
* “How can artificial intelligence assist in identifying and recommending key factors to enhanceuser engagement?”
* “What are the most effective community management techniques for fostering sustained user activity and participation in online music communities?”

## End products



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Deren Serce* | *D.S.* | *Project executor* | *Monday, Wednesday, Thursday* |

# Activities and time plan

## Phases of the project

*For the Disx project, which follows an agile approach using Scrum methodology, the main phases are outlined below:*

1. **Initiation Phase:**
   * Define project goals, scope, and deliverables.
   * Conduct initial research and analysis to understand user needs and market trends.
   * Set up project infrastructure and tools.
2. **Planning Phase:**
   * Create a product backlog containing all desired features and user stories.
   * Prioritize backlog items based on business value and stakeholder input.
   * Estimate effort and complexity for each backlog item.
   * Define sprint goals and select backlog items for the initial sprint.
3. **Execution Phase (Sprints):**
   * Sprint Planning: Select backlog items for the sprint and define sprint goals. This phase typically occurs at the beginning of each sprint.
   * Sprint Execution: Development team works on implementing selected backlog items according to the sprint plan.
   * Daily Stand-up Meetings: Short daily meetings to discuss progress, challenges, and plans for the day.
   * Sprint Review: Demonstrate completed work to stakeholders and gather feedback. Review sprint goals and adjust backlog priorities if necessary.
   * Sprint Retrospective: Reflect on the sprint process, identify areas for improvement, and make adjustments for future sprints.
4. **Closure Phase:**
   * Handover: Ensure smooth transition of deliverables to stakeholders or operational teams.
   * Evaluation: Assess project outcomes against initial goals and success criteria.
   * Reflection: Conduct a retrospective to review project processes, identify lessons learned, and document best practices for future projects.
   * Wrap-up: Finalize project documentation, archive project artifacts, and celebrate team achievements.

## Time plan and milestones

***Agile Artifacts:***

* Sprint Length: Three-week sprints will be adopted to strike a balance between short feedback cycles and manageable workloads.
* Stand-up Meetings: Since I will be executing this project on my own, no stand-ups will be done.
* Sprint Demo: At the end of each sprint, a demo will be conducted to showcase completed work to stakeholders(teachers) and gather feedback.
* Retrospective: Since I will be executing this project on my own, no retros will be done.

# Testing strategy and configuration management

## Testing strategy & environment

For the Disx project, the testing strategy will encompass unit testing, continuous integration (CI), and code quality checks using GitHub Actions and potentially integrating with tools like SonarQube.

1. Unit Testing

* Goal: Achieve high code coverage and ensure individual components function as expected.
* Approach: Write unit tests for each function, method, or component in the codebase using a testing framework such as Jest (for JavaScript) or pytest (for Python).
* Automation: Unit tests will be automated and integrated into the CI pipeline to run on each code push or pull request.

2. Continuous Integration (CI):

* Goal: Detect and address integration issues early in the development process.
* Approach: Utilize GitHub Actions as the CI/CD tool to automate build, test, and deployment processes.
* Automation: CI workflows will be set up to trigger unit tests, linting, and code quality checks on every code change.

3. Code Quality Checks:

* Goal: Maintain code quality standards and identify areas for improvement.
* Approach: Integrate code quality analysis tools like ESLint (for JavaScript) or Flake8 (for Python) into the CI pipeline to enforce coding standards and identify potential issues.
* Automation: Code quality checks will be automated and integrated into the CI workflow to provide immediate feedback to developers.

4. SonarCloud:

* Goal: Ensure code maintainability, reliability, and security.
* Approach: Integrate SonarQube as a code quality analysis tool to perform static code analysis and identify code smells, vulnerabilities, and security risks.
* Automation: SonarQube analysis will be triggered as part of the CI pipeline, providing continuous feedback on code quality metrics and suggesting improvements.

Automation of these testing processes through GitHub Actions enables rapid feedback to developers, streamlines the development workflow, and ensures consistent quality standards across the codebase.

## Configuration management

*Project Approach for Version Management:*

Version Management Tooling:

* GIT will be used as the version control system for the Disx project due to its popularity, flexibility, and robust branching capabilities.

Branching Strategy:

* Feature Branching: The project will adopt a feature branching strategy where each new feature or user story will be developed in a dedicated feature branch. This allows for parallel development of features without disrupting the main codebase.
* Main Branch: The main branch (often called "master" or "main") will represent the stable version of the codebase. Changes from feature branches will be merged into the main branch through pull requests after thorough testing and code review.

Overall, the project approach for version management will emphasize transparency, and continuous improvement to ensure the successful development and delivery of the Disx music platform.

# Finances and risk

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Delay | Setup deadlines | Make new planning |
| 1. Lack of knowledge | Research, peer/teacher communication | Working with peers |
| 1. Hardware issues | Regularly software updates, handling equipment gently | Borrowing items from ISSD |
| 1. Loss of data | Using cloud, VSC, making back-ups | Using back-up |