

Project Plan

Trivia Music Web-based game

Teens and young adults

Date	:	05 April 24
Version	:	0.2
State	:	In progress
Author	:	Tony

Version history

Version	Date	Author(s)	Changes	State
0.1	18 March 24	Tony Jiang	Initialize document	In progress
0.2	05 April 24	Tony Jiang	Add testing strategy	In progress

Distribution

Version	Date	Receivers

Contents

1. Project assignment.....	3
1.1 Context	3
1.2 Goal of the project	3
1.3 Scope and preconditions	3
1.4 Strategy	3
1.5 End products.....	3
2. Activities and time plan.....	5
2.1 Phases of the project.....	5
2.2 Time plan and milestones.....	5
3. Testing strategy and configuration management	6
3.1 Testing strategy	6
3.2 Testing environment and required resources	6
3.3 Configuration management	6

1. Project assignment

1.1 Context

My individual project is a music trivia web-based game that tests players' knowledge across various genres and time periods. To win, players must select the correct answer quickly to accumulate points. At the end of each round, the player with the highest score emerges victorious. Participants have one minute to identify either the song title or the artist, depending on the game mode, after listening to a 10-second sample. The game offers the option to play solo, with friends, or against random opponents in multiplayer mode.

1.2 Goal of the project

The goal of this project is to scale the web-based game to handle around ten thousand users without experiencing any downtime or encountering any performance issues, both during the game session and creating the session, regardless of whether traffic is low or high traffic. Additionally, I aim to ensure website security, protect users' information, and eliminate cheating. Last but not least, I aim to create a fun gaming experience for the players to enjoy.

1.3 Scope and preconditions

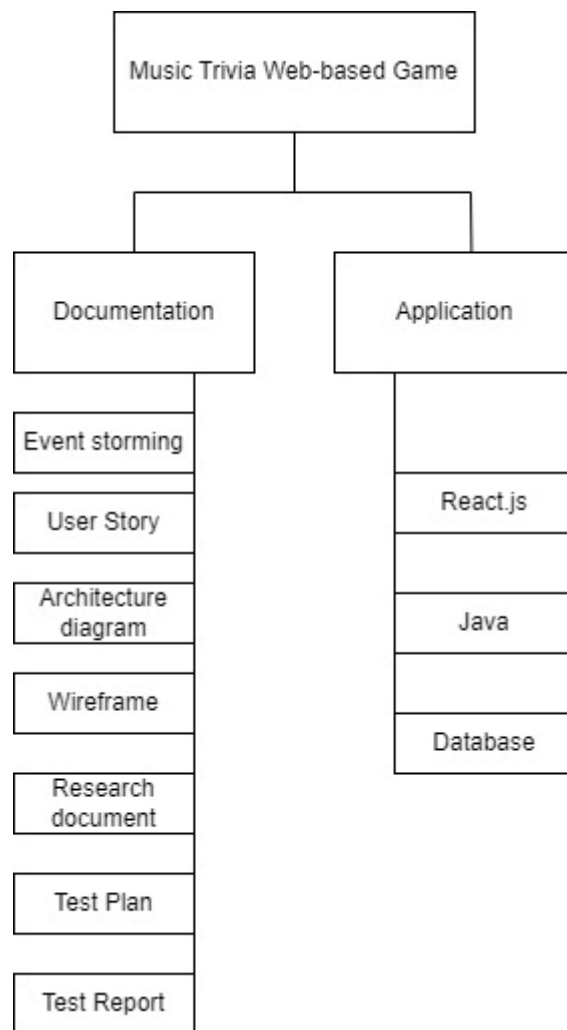
<<What activities and which end products (to what extent or quality) belong to the project, and which don't.>>

Inside scope:	Outside scope:
1 User story.	1
2 C4 architecture.	2
3 Wireframe.	3
4 Test plan.	4
5 Research document.	5
6 Scale the web application around ten thousand users.	6
7 Experience minimal low performance issues on the web app.	7
8 Implement authentication and authorization.	8
9 Implement CI/CD pipeline	

1.4 Strategy

For this project, I'm adopting a Scrum approach combined with DevOps. With Scrum, I can demonstrate the progression of the project during each sprint iteration, gather feedback, and continuously improve. Additionally, I will integrate continuous integration for automated testing and continuous deployment for automated deployment practices during each sprint. This process will enable faster, more efficient, and reliable software development.

1.5 End products



2. Activities and time plan

2.1 Phases of the project

<< Describe the main phases of your project. Even in a scrum project, you should specify at least the components at the beginning and end phases like problem analysis in the beginning, as well as handover, evaluation, reflection, and wrap up at the end.

For internship projects, reserve sufficient time for developing/maintaining the portfolio/thesis.
>>.

The phases of the project are in sprints. Each sprint I have to deliver and present the product to my technical teachers. Each sprint takes 3 working weeks before the end of the internship. The whole project should be finished in 5 sprints. Below it is possible to see the supposed milestones and correlating activities for the project.

- Sprint 1: Initial planning and defining - Initial planning of the project and defining the structure of the project.
- Sprint 2: Setup environment and initial implementation - setup the environment of the project after defining the structure as planned and start implementing.
- Sprint 3: improve and continue implement – improve on the feedback from sprint 2 and continue to implement.
- Sprint 4: improve and continue implement – improve on the feedback from sprint 3 and continue to implement
- Sprint 5: finish the project – Finish with everything for the project.

2.2 Time plan and milestones

<< For a waterfall project you can indicate the phases and milestones below (can be adapted as required).

For an agile project, describe how the artefacts are planned. E.g., length of sprint (with justification), organization of stand up, demo, retrospective.
>>

Phasing	Start date	Finish date
1 Initial planning and defining	4 March 24	22 March 24
2 Setup environment and initial implementation	25 March 24	12 April 24
3 improve and continue implement	15 April 24	8 May 24
4 improve and continue implement	13 May 24	31 May 24
5 finish the project	3 Jun 24	21 Jun 24

3. Testing strategy and configuration management

3.1 Testing strategy

- Unit test: This test will be used to test the code behaviour. The unit test will be done for each functionality. I will make sure to have at least 80 percent coverage for Unit test.
- Integration test: This test will be used to test the behaviour of the system components working together. I will make sure that the frontend, backend, and database are working together properly.
- Acceptance test: This test will be used to check if the stories acceptance criteria were met.

3.2 Testing environment and required resources

Environments:

- The tests are performed on a local machine (laptop or pc).
- The unit tests/integration are run on a CI/CD pipeline.
- Postman is used to test HTTP API requests and examine their responses.

Required resource tools:

- IntelliJ for Backend development.
- Visual code for Frontend development.
- Database TBD.
- Postman for HTTP request.

3.3 Configuration management

The branching strategy is git flow.

Tools:

- GitHub with the following branches:
 - Main branch: This will contain the last working version of the prototype. Merge to the main branch will be done only when the developed feature is fully working and tested successfully.
 - Development: Up to date branch where all the features are merged and tested.
 - Hotfix: Fixes related to main branch to quickly resolve and push.
 - Feature branch: There will be a branch for each feature.
 - Documents: contains all the documents of the project.