Project Plan

Red Scar

Date	:	**09.02.2023**
Version	:	**1**
State	:	**Draft**
Author	:	**Viktor Skachkov**

Version history

Version	Date	Author(s)	Changes	State
1	09.02.2023	Viktor Skachkov	Starting the project plan	Draft
2	13.02.2023	Viktor Skachkov		

Distribution

Version	Date	Receivers

Contents

1. Project assignment

1.1 Context

My task is to create a full-stack application (in my case, I will make a site where a company can sell their videogames and additions to them) with a frontend, a backend and a database. For the frontend, I should use Javascript React, Java for the backend and I can choose which type of database I want to implement, but I have decided to use MySQL.

The project needs to include at least the following:
- user authentication
- user authorization with at least two types of users (like admins and customers) and separation of privileges
- feature to retrieve detailed data about a database object (like invoice)
- feature to retrieve aggregated data (like total sales for a specific period of time)
- filter to search with
- feature to modify an existing database object
- feature to remove an existing database object
- feature to create a database object
- implementation of Websockets (like chat or notification)

1.2 Goal of the project

This website will allow the company to sell their videogames and will add other functionalities like reviews, to see which games are well-liked and which aren't and what can be improved, support, to help users who have troubles, and the possibility to buy additions to the games like expansions. This way, the company hopes to increase the trust of the customers in its products and surpass the competitors.

1.3 Scope and preconditions

Inside scope:	Outside scope:
1 Frontend (with Javascript React)	1 Hosting the application on a server
2 Backend (with Java)	2
3 Database (MySQL)	3
4 CI/CD environment	4
5 Testing (with Cypress for the frontend and unit tests for the backend)	5
6 Sonarqube inclusion	6
7 Dockerization (three separate containers for the frontend, backend and database)	7

1.4 Strategy

I will use Scrum (an Agile approach) for my project. It means that the working period would be separated in sprints, and I will split my work for each sprint and ask for feedback from my teachers. This way, if they are unhappy with what I am doing, they can tell me early on and I can fix my mistakes.

If I use the Waterfall approach, I won't be able to do that because I would work on the project without receiving any feedback and just present it to the teachers by the end of the semester. For that reason, Agile Scrum is the better choice.

1.5 Research questions and methodology

1. How to connect the frontend to the backend? – API requests are used. The frontend sends requests to a specific address while the backend provides those addresses and carries out a specific task once the request from the frontend reaches the method.

2. How to connect the backend to the database?
3. How to utilize Spring and Dependency Injection in the backend?
4. How to dockerize my three components (frontend, backend, database) of the project?
5. How to connect Sonarqube to GitLab?
6. How to implement authorization and authentication?
7. How to implement a reliable CI/CD environment in GitLab?
8. How to include reliable tests?

1.6 End products

1. Java Backend
2. Javascript Frontend
3. Database
4. Docker containers
5. SonarQube
6. Continuous Integration and Tests
7. Design Document
8. Security Reports
9. UX feedback document
10. Backlog tasks
11. Applied research document
12. Project plan

2. Project organisation

2.1 Communication

The communication with the teachers will be done either in person at the university, through Teams or through emails.

I have to update the teachers on the progress I have made at least two times per sprint, but it would be better if I ask for feedback more often than that.

During the meetings, I will write down notes if the teacher has a lot of feedback that I can't fully memorize about things I should improve on, and I would prefer if the meetings took place in person because this way, they the feedback is more comprehensive for me.

3. Activities and time plan

3.1 Phases of the project

Sprint	Actions
Sprint 1	Planning the project, starting the backend and creating some RESTful services and a CI/CD environment
Sprint 2	Applied research document, starting the frontend, CORS configuration for controllers and designs and architecture for the project
Sprint 3	Design document, database setup and mockito testing, SonarQube, research document
Sprint4	Include CI in design document, make sure that there are no problems with continuous integration, include authentication and authorization
Sprint5	Finish design document and project, security report based on the OWASP top 10 security risks, include websocket feature
Sprint6	Finish the project, the security reports and the testing for the application

3.2 Time plan and milestones

Phasing	Effort	Start date	Finish date
Sprint 1	Starting project	06.02.2023	03.03.2023
Sprint 2	Starting frontend	06.03.2023	24.03.2023
Sprint 3	Database setup, SonarQube	27.03.2023	14.04.2023
Sprint 4	Continuous integration, authorization and authentication	17.04.2023	12.05.2023
Sprint 5	Security report, websocket feature	15.05.2023	02.05.2023
Sprint 6	Finish everything concerning the project	05.05.2023	23.05.2023

4. Testing strategy and configuration management

4.1 Testing strategy

I will include test for everything in both the frontend and the backend. I will mostly use unit testing because it's important to test every single part of the application and unit testing inspect the smallest parts of an application. I will also include system testing to check how the application behaves as a whole because it's important to know how the system will work and not just test its different units.

I will try to have a 100% coverage with the tests and I will make my tests automated. I will also include Sonarqube in my project which is a quality testing setup that inspects the quality of the code and provides results which allow the developer to see what could be improved.

4.2 Test environment and required resources

I will use a CI/CD environment on GitLab to automatically run the tests when pushing alterations. I will need to install a GitLab Runner for that purpose and register it.

I will also use Sonarqube to test the quality of the code and connect it to GitLab, so the process would become automatic.

4.3 Configuration management

I will make frequent updates to the code on GitLab because this way, there would be less changes per request and thus, the possibility for clashes also becomes lower.

I will also create secondary branches to ensure where I would make changes and then make merge requests to the main branch to decrease the possibility of errors.

The CI/CD environment would also help with the version management because it would automatically run all the tests and inform the developer if they still pass or not. In case they pass, then everything is alright, but in case one or more of them fail, then the developer needs to make changes to the code again.

5. Finances and risk

5.1 Project budget

The project will not need any budget because everything I will be using is for free. I will use IntelliJ as my environment and include different dependancies that I need aong the way and also use MySQL Workbench for my database.

5.2 Risk and mitigation

Risk	Prevention activities	Mitigation activities
1 Unavailability of the teachers	-	We can communicate through email.
2 I am unavailable	I will make sure to focus solely on my studies and not on anything outside of them.	I will try to organize my time in case of something emergent.
3 I get stuck at something I don't understand and waste a lot of time	I will ask the teachers to help me.	I will try to do everything as fast as possible to have a lot of time in cases like that.

Risk Matrix

The numbers of the risks will be used in the Risk Matrix instead of their full names.

	Low	Moderate	High
Very Likely			3
Likely	1, 2		
Unlikely			
Occurrence/Impact	Low	Moderate	High