

Project Plan

Find your Garageband

Fontys University of Applied Sciences | Eindhoven

ICT S3

Date	:	March 24, 2023
Version	:	2.0
State	:	Unfinished
Author	:	Fabiënne Leidekker

Version history

Version	Date	Author(s)	Changes
0.1	25-02-2023	Fabiënne Leidekker	Edited 1.1, 1.2, 1.3, 1.4
0.2	27-02-2023	Fabiënne Leidekker	Edited 2.1, 2.2, 3.2
1.0	03-03-2023	Fabiënne Leidekker	Edited 1.5, 1.6, 3.1, 4.1
1.1	12-03-2023	Fabiënne Leidekker	Edited 1.2, 1.3, 3.2, 4.1, 4.2, 4.3, 5.1
2.0	16-03-2023	Fabiënne Leidekker	Edited 1.2, 1.3, 4.1, 4.3 (now 4.2) removed 4.2

Distribution

Version	Date	Receivers
1.0	03-03-2023	Márcio Dantas and Felipe Ebert
2.0	24-03-2023	Márcio Dantas and Felipe Ebert

Contents

1.	Project assignment	4
1.1	Context	4
1.2	Goal of the project.....	4
1.3	Scope and preconditions	4
1.4	Strategy.....	4
1.5	End products.....	5
2.	Project organisation	6
2.1	Stakeholders and team members.....	6
2.2	Communication	6
3.	Activities and time plan	7
3.1	Phases of the project	7
3.2	Time plan and milestones.....	7
4.	Testing strategy and configuration management	8
4.1	Testing strategy	8
4.2	Configuration management.....	8
5.	Finances and risk	9
5.1	Risk and mitigation	9

1. Project assignment

1.1 Context

As an amateur musician, it can be difficult to find people to jam with. The musicians application makes it possible for a musician, such as a guitarist, to find a band to play with. It will also work the other way around, so a band can look for their missing member. In addition, a user can chat with people to ask for tips during their own learning process. And last but not least, a user could also search for a music teacher.

1.2 Goal of the project

The goal of this project is to help musicians connect with other people through their shared passion for creatin music.

1.3 Scope and preconditions

Inside scope:	Outside scope:
1 CI/CD (precondition)	1 Maintaining after semester
2 MySQL database (precondition)	2 Hosting other than FHICT
3 Java with SpringBoot (precondition)	3 Making it native
4 JavaScript with React (precondition)	4 Making it responsive
5 Hibernate as ORM (precondition)	5

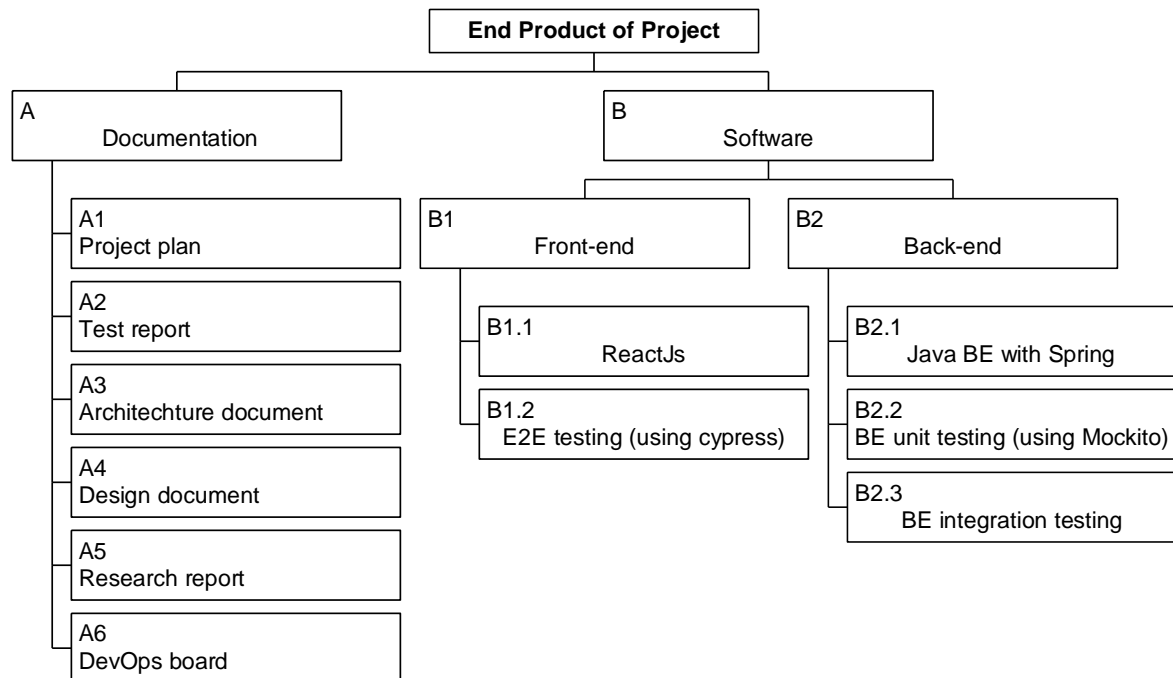
1.4 Strategy

For this individual project I will be using an Agile Scrum method. This is partly because I'm already familiar with it and because we will also be using it during the group project. Other reasons why I am going to use Scrum are because of the benefits. With scrum you divide your project into sprints that are about three weeks long. With each sprint you deliver a small deliverable of your product. After those weeks you or your group will have an retrospective to look back at the past sprint. "Is everything we wanted done?", "What do we need to improve on during the next sprint?". This is a very nice way to reflect of your process and progress. If the stakeholder doesn't like the way the app is going, there is the next sprint to improve on that.

In addition, it is also useful for keeping track of tasks. For my personal project I will be using a DevOps board for all my tasks.

1.5 End products

Project Breakdown Structure (PBS)



2. Project organisation

2.1 Stakeholders and team members

Name	Abbreviation	Role and functions	Availability
Fabiënne Leidekker 481376@student.fontys.nl	FL	Developer	Monday through Friday, from 9.00 - 16.00
Márcio Dantas m.paixaodantas@fontys.nl	MD	Teacher	Monday, from 12.00 - 16.00 and Friday, from 12.00 - 16.00
Felipe Ebert f.ebert@fontys.nl	FE	Teacher	Wednesday, from 12.00 - 16.00 and Friday, from 9.00 - 12.00

2.2 Communication

On Monday afternoon, Tuesday, Wednesday afternoon, Thursday and Friday I will be working on my personal project. At the start of each day I will have a little stand-up where I take a look at my scrum board and see what I still need to do.

Every week on location I will try to speak with one of the teachers to show my progress and get feedback to know if I'm still going in the right direction.

3. Activities and time plan

3.1 Phases of the project

The canvas planning of this semester has given a schedule in which you tackle a few new things each sprint, instead tackling all subjects randomly. This creates a more structured way of working. In the beginning I will be focusing on a few things, like creating a RESTful back-end application, setting up my project management plan and writing some documentation.

Future sprints will focus more creating a front-end, software documentation and reports, testing, security and web sockets. In the last sprint, I will mainly work on putting the dots on the I, so that I can finish the project completely and (hopefully) well.

3.2 Time plan and milestones

As mentioned before, I will be working with the Agile scrum method. The semester is split into 6 sprints, where every sprint is 3 weeks long. The sixth sprint's purpose is mostly for finishing touches.

Phasing	Effort	Start date	Finish date
Sprint 1	Start developing my idea, setup Back-end + coding, initialization of CI	06-03-2023	03-03-2023
Sprint 2	Design document (C4 model, reason product, like react), CORS configuration, Initial front-end setup, Initial research document	06-03-2023	24-03-2023
Sprint 3	Update C4, Database implementation, unit testing (using Mockito), Using SonarQube, Research document complete	27-03-2023	14-04-2023
Sprint 4	Authentication and authorization, Design document (diagram of CI, match architecture diagram to BE), CI and SonarQube (> 80% tested and passing)	17-04-2023	12-05-2023
Sprint 5	Final design document, Websockets, Security report, MVP, CI and SonarQube (> 80% tested and passing)	15-05-2023	02-06-2023
Sprint 6	Final UX feedback report + result improvements UI, Final MVP, CI/CD	05-02-2023	23-06-2023

4. Testing strategy and configuration management

4.1 Testing strategy

There are a few testing strategies that I envision for my project. For the back-end I will be writing **unit tests**. With unit test I can test individual units of my software to ensure it functions as intended. I'm also thinking of **integration testing**. Unlike unit testing, integration testing tests multiple units and the interactions between them. For both the front- and back-end testing, I will be doing end-to-end testing. **E2E testing** will test my app from the web browser through to the back end of my application and searches for deviations from expected outcomes in both input and output. So it basically makes sure that my entire app is functioning as a cohesive whole. Last but not least, **acceptance testing**. This type of testing assesses if the software meets the end user's expectations.

To test the quality of my code I will use **SonarQube**. It reviews my code to detect bugs and code smells (indication of weaknesses in design, program could be more prone to bugs and failures). It reports on a lot of things, like duplicated code, coding standards, unit tests, code coverage, code complexity, comment and security recommendations.

4.2 Configuration management

For version management I'm using GitLab. Because I'm creating this project individually, I won't be using branches, probably only when I want to try different implementations of something or try something more experimental.

5. Finances and risk

5.1 Risk and mitigation

Risk*	Prevention activities	Mitigation activities
1 32. Impacted individuals aren't kept informed	Have a feedback meeting with the stakeholder (teachers) multiple times each sprint	Have a feedback meeting with the stakeholders (teachers) as soon as possible
2 44. Architecture lacks flexibility	Implement software principles, like SOLID	Have a teacher review my code, and lecture me about the changes that need to be made
3 45. Architecture is not fit for purpose	Constantly work and improve my project, including code AND documentation	Take a few steps back and look at the parts my project is failing
4 50. Design fails peer review	Research design implementations and test those with actual people	Redesign the parts of the design that are failing
5 55. Technology components have security vulnerabilities	Test the security implementation	Isolate the component and research why it's vulnerable
6 64. Components or products aren't maintainable	Document every software and design choice.	Search which components aren't maintainable and make it maintainable
7 121. User interface is low quality	Make sure to research about implementation of UI	Remove the UI parts that cause the issue and replace them with a better design

*The risks are taken from website: <https://management.simplicable.com/management/new/130-project-risks>