**Project Plan**

***JAZZ&CLASSICS***

*Fontys Rachesmolen*

<<

*This template can be used for all projects, especially software engineering projects. Chapters or parts that are not applicable can be removed.*

*Text in italic is background information and must be removed in the final version of your project plan.*

*Note that this is a template and can be changed for own purposes, e.g. you can adapt the layout to the layouts as used at the company of your internship.*

*For your project name, think of a name that highlights the most relevant aspect of your project, and specify whether it is about graduation internship or third year internship.*

>>

|  |
| --- |
| **Date : 08/09/2023** |
| **Version : 1** |
| **State : Eindhoven** |
| **Author : Alec Sola Castermans** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
|  |  |  |
|  |  |  |

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)

[2.2 Communication 6](#_Toc42673521)

[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 8](#_Toc42673526)

[4.2 Test environment and required resources 8](#_Toc42673527)

[4.3 Configuration management 8](#_Toc42673528)

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

[5.1 Project budget 9](#_Toc42673530)

[5.2 Risk and mitigation 9](#_Toc42673531)

# Project assignment

## Context

*There is a wide variety of jazz and classical music listeners around the world, however not many get to see it live where you can really feel the music. What if there was a website that helped you get to see these shows?*

## Problem Statement

*There was recently a study made by researchers from Cambridge University that stated that older people tend to listen more to classical music and jazz while the younger generation lean more towards R&B, pop or hip hop. With all the new technology it’s every time more challenging for older generations to acquire tickets because it’s difficult for them to use the internet, some theatres no longer offer in-person ticket sales, opting for only-online option. To continue, if someone wants to buy tickets for a determined show, even if they know how to use the internet, they will probably have to do unnecessary research that is eventually going to waste time.*

*The problem in this case is the lack of visibility this matter has, many people, including the younger generations are open to visit these theatres, however since it takes so much time to just book one seat, it discourages them and tend to leave it behind.*

*Due to this, society is losing the love for this beautiful type of music that has been around for centuries. Furthermore, studies from The National Library of Medicine even postulate that it has health benefits including reduction of anxiety which avoids irregular heartbeat and among others, it can stimulate the brain in order to help concentration.*

## Goal of the project

*The primary objective of this project is* ***helping****, make things easier and more* ***efficient****. However, its overarching goal is to* ***restore the love*** *for this music, a timeless art that allows us to travel through time and experience the world from a different perspective.*

*Since it’s not possible to return to the past and make theatres sell tickets locally again, the solution for this matter is to make* ***a large website*** *that displays the* ***schedule*** *of the most famous theatres in Europe and facilitates the* ***purchase*** *of tickets for these stage plays.*

*The advantage of this project when finished is that people won’t have to worry anymore in doing unnecessary research. Instead of having to go through different websites to see different performance venues, they will have everything they need in just one, this makes* ***research more efficient.*** *This benefit will make live shows more accessible starting a chain reaction that brings many other advantages like for example, the hiring of new employees or the tourism rate increase.*

## 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. Ticket sales | 1. Advertisement |
| 1. Scheduling of events 2. Login & Register 3. A Home Page for relevant news related to the website. 4. A separate page to see the different famous people that attended to the theatres. 5. Discount Codes 6. Season Tickets | 1. Returning system (for tickets) 2. Chat helper 3. Customer service |

*x*

*Pre-Conditions:*

*Repository in Git.*

*IntelliJ*

*Java.*

*Spring boot.*

*Runners.*

*No DTO’s.*

*Database SQL.*

## Strategy

*There are many strategies I could follow in order to deliver this project, the main two options are waterfall and agile scrum.*

*I don’t think one is better than the other, I think it depends on the task. Therefore, for many reasons, I’ve decided to approach the agile way of working.*

*The reason I’ll be working this way is because my project is going to be flexible, it’s going to be in constant change, I want my product to align to the stakeholder’s preferences. Consequently, I will regularly update my work in order to seek feedback from the client and make the necessary adjustments to present it later again. This is the pattern that will be followed.*

*Waterfall hasn’t convinced me because it’s not a secure way of working unless the idea is very clear. If I were to work this way, once I would present my final product, if the client isn’t satisfied, I would have to make major changes which may take a long time. Asking for feedback periodically, allows me to readjust the project in phases, making the process and progress secure.*

## End products



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Bart Rabeling*  *Senne de Ruiter*  *Erik van der Schriek* | *B.T.*  *S.M de*  *H.J.D* | *Technical teacher*  *Semester coach*  *Technical teacher* | *Monday*  *Wednesday & Friday*  *Wednesday & Friday* |

## Communication

*Although I haven’t consulted this matter with my teachers, the ideal way of communication would be me asking for feedback every week or every time I have updated/added something relevant to my project. I prefer the meetings being in physic because the feedback is easier to understand, nevertheless, if I have a question which I feel is not that complex I will send an e-mail, I understand that teachers need their time as well.*

# Activities and time plan

## Phases of the project

*All projects follow a determined structure, I’m conscious that earlier in this document I stated that my project would be flexible, however this doesn’t mean that I don’t have certain requirements to go over.*

***Beginning.***

* ***Problem Analysis*** *is the first step, realising what the issue is.*
* ***Ideation*** *of a potential solution.*
* *Make a* ***project plan*** *based on the solution.*

***Handover.***

* *Setting* ***deliverables.***
* ***Backlog Development.***
* ***Testing.***
* ***Sprint Planning****. I select determined parts I would like to display and submit to the stakeholders.*
* ***Sprint Execution*** *by having stand-up presentations/meetings displaying what was submitted.*

***Evaluation.***

* ***Sprint Review.***
* *Review* ***Meeting.***
* *Evaluation of the stakeholder.*

***Reflection.***

* ***Retrospective.*** *Look back into what should have been different and what was done correctly.*

***Wrap Up.***

* *Consider what could be added in the* ***future.***

## 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** | **Effort** | **Start date** | **Finish date** |
| 1 | Ideation | 08/09/2023 | 08/09/2023 |
| 2 | Prototype | 08/09/2023 | 08/09/2023 |
| 3 | Project Plan | 08/09/2023 | 08/09/2023 |
| 4 | Git Board (Planning) | 10/09/2023 | 10/09/2023 |
| 5 | RESTful API Setup (Class Layering, Dependency Inj.) | 17/09/2023 | 22/09/2023 |
|  | **Sprint Submission 1** | **22 Sept** | **22 Sept** |
| 6 | Design Document Version 1 | 22/09/2023 | 13/10/2023 |
| 7 | Backend (CORS) | 22/09/2023 | 13/10/2023 |
| 8 | Frontend setup | 22/09/2023 | 13/10/2023 |
|  | **Sprint Submission 2** | **13 Oct** | **13 Oct** |
| 9 | Design Document Version 2 | 13/10/2023 | 10/11/2023 |
| 10 | Backend Database Setup | 13/10/2023 | 10/11/2023 |
| 11 | Sonarqube | 13/10/2023 | 10/11/2023 |
|  | **Sprint Submission 3** | **10 Nov** | **10 Nov** |
| 12 | Design Document Version 3 | 10/11/2023 | 01/12/2023 |
| 13 | Authentication and Authorization (Backend) | 10/11/2023 | 01/12/2023 |
| 14 | Authentication and Authorization (Frontend) | 10/11/2023 | 01/12/2023 |
| 15 | Continuous Integration and Sonarqube | 10/11/2023 | 01/12/2023 |
|  | **Sprint Submission 4** | **1 Dec** | **1 Dec** |
| 16 | Final Design Document | 01/12/2023 | 22/12/2023 |
| 17 | Websocket feature | 01/12/2023 | 22/12/2023 |
| 18 | Minimum Viable Product | 01/12/2023 | 22/12/2023 |
| 19 | Continuous Integration and Sonarqube | 01/12/2023 | 22/12/2023 |
|  | **Sprint Submission 5** | **22 Dec** | **22 Dec** |
| 20 | UX Feedback Report | 22/12/2023 | 19/01/2024 |
| 21 | Minimum Viable Product | 22/12/2023 | 19/01/2024 |
| 22 | Continuous Integration and Sonarqube | 22/12/2023 | 19/01/2024 |
| 23 | Web Performance review document | 22/12/2023 | 19/01/2024 |
|  | **Sprint Submission 6** | **19 Jan** | **19 Jan** |

# Testing strategy and configuration management

## 

## Testing strategy

*There are many ways of testing that could be implemented in this project.*

*Firstly, I could use End-to-End tests where I would test the whole structure of my project, it would have more code integration than other testing strategies, but on the other hand, if something goes wrong it would take a lot of time to find out where the problem is. Secondly, I could use Integration tests in order to see the behaviour of components working together, this would be less complex, however, it would still take some time. Lastly, I could use Unit testing, this way I would observe the behaviour of specific methods, this is fine-grained technical testing.*

*It’s not that one way of testing is better than the other one, it really depends on the person, however for me, I will choose* ***Unit testing*** *for this project. By using Unit testing I’m able to analyse what a method really is doing once I’ve coded it, for me it’s an organised way of working and it will allow me to keep everything in track without having to worry about other methods.*

## Test environment and required resources

*For this project I envision a DTAP environment.*

* ***Development:*** *Write and test the code locally.*
* ***Testing:*** *Testing the code more comprehensively, the objective is to find all remaining bugs and search for a fix.*
* ***Acceptance:*** *Stakeholder validates the project.*
* ***Production:*** *Project is accessible to all users.*

*To add I will make use of CI/CD pipelines in GIT, this is already created in my repository.*

## Configuration management

Version Control Management is crucial in a project, it allows to keep all changes in track making it easier to solve issues later on.

I will have a repository in Git due to it’s popularity, robustness and extensive tooling support.

As of the branching strategy which is very important because it helps define how each feature, improvement, or bug fix is handled, I will use a Main Branch to represent the production-ready code. I will create Feature Branches for new features or enhancements and lastly, I will have a Release Branch in order for me to prepare what’s going to be released.

Next, implementing a promotion strategy can help me move code changes in different stages(development, testing, production). For this, it’s very handful to make use of CI/CD pipelines or runners, this way I ensure that the code is built, tested and deployed consistently.

Maintaining a release documentation process in order to detail what changes are included after each release is important as well.

Lastly, I will use GitBoard issues in order to deal with change requests and problem reports, this way I can prioritize them and organise my project the best way possible.

# Finances and risk

## Project budget

*There is no budget for this project, no money will be invested.*

## Risk and mitigation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Impact** | **Probability** | **Prevention activities** | **Mitigation activities** |
| 1. Missing deadlines. | Big | 10% | Work in front of schedule so that there is always time. | Talk with the teacher/client and explain why the deadline was missed. |
| 1. Communication problems | Medium | 30%/ | Have an alternative way of communicating (e-mail, teams) | Let know that communication should have been better. |