**Project Plan**

***Crowdfunding page***

**

<<

*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 : Date** |
| **Version : Version** |
| **State : State** |
| **Author : Mateo González Alonso** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1.0 | 1/10/2024 | Mateo | Teacher Review | Draft |
|  |  |  |  |  |
|  |  |  |  |  |

**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)

[5.2 Risk and mitigation 9](#_Toc42673531)

# Project assignment

## Context

This platform allows users to create and back crowdfunding projects. Creators can list projects, set funding goals, and describe their purpose. Backers can browse projects and pledge funds using fake payments (for the moment). The application integrates user authentication, role-based authorization (admin, creator, backer),

## Goal of the project

The goal of this project is to develop a full-stack application that empowers users to fund and create projects with ease. In today’s competitive environment, traditional funding methods can be restrictive and inaccessible. This platform will provide a democratized space where users can bring their ideas to life by setting up projects with descriptions, images, and funding goals, while allowing funders to track their contributions and see the impact they are making.

The new platform will create a seamless and transparent experience for both project creators and supporters. Users will benefit from intuitive project management tools and secure transaction processing, while funders can track progress and outcomes in real-time. This environment will foster a strong community of creators and supporters, driving engagement and increasing the chances of project success.

For the company, this project adds significant value by expanding its service offerings and establishing a presence in the growing crowdfunding market. The platform not only creates a new revenue stream but also positions the company as an innovator in the digital space, attracting a diverse user base and enhancing brand reputation. Additionally, the platform’s scalability and integration capabilities will ensure long-term growth and adaptability.

## 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. Real time funded money display | 1. Mobile UI compatibility |
| 1. Documentation and user guides | 1. Developing real payments |
| 1. User friendly UI | 1. Regarding legal payments or User data manipulation |
| 1. User can create projects | 1. Multi-Language Support |
| 1. User can fund other projects |  |
| 1. Admins can delete projects |  |

*<< Indicate any preconditions. E.g., think of technology choices that have already been made by the company. Note that you are also expected to retain a critical, but constructive, mindset for choices already made >>*

## Strategy

*<< Describe the strategy for your project (the approach). E.g., waterfall, or an agile approach like scrum, and justify the choice. >>.*

The strategy I will implement is the Scrum Agile methodology, as it provides the flexibility needed to adapt to changing requirements throughout the project. This approach is ideal for my work because it allows for incremental development, enabling continuous refinement and improvement based on feedback and evolving needs. By working in sprints, I can ensure steady progress while retaining the ability to make adjustments if any requirements shift.

To facilitate effective project management, I will use Jira, a comprehensive tool that will help me break down the project into smaller, manageable tasks. By organizing and prioritizing tasks within Jira, I can ensure that every aspect of the project is covered, and my responsibilities are clear. Jira’s tracking features will allow me to monitor my progress, maintain visibility on what has been completed, and identify any bottlenecks or challenges early on.

Using Jira, I aim to keep my workflow transparent and organized, making it easier to meet deadlines and ensure that the project stays on track. This approach will help me maintain focus and productivity, ensuring that the project is completed efficiently and effectively.

## Research questions and methodology

*<<*

*Describe the research questions that are most relevant to your project. For each research question, describe the approach and/or methodology. Use the Dot Framework to specify strategies and methods - see* [*http://www.ictresearchmethods.nl*](http://www.ictresearchmethods.nl) *for details.*

*Note that research is not only part of the initial phases (like analysis) of the project, but runs throughout the whole project. E.g., in the realization phases, you will probably do research in the Workshop and Lab context.*

*Realize that during the project your research questions may change, and that new ones will come up. That normal for any project, and is not a problem as long as you involve the right stakeholders, and keep your deliverables updated.*

*>>*

## End products

*<< A Product Breakdown Structure (PBS) lists the end products that you realize, including a description of each product. In software engineering, the products are more than just the project plan and the application itself. E.g., requirements documents, architecture documents, research reports and test reports are all end products. These are all important products that are required for effective handover. They are also necessary for further maintenance and follow up-projects. The PBS can change during the course of the project.>>*

* Frontend (User Interface):
  + Dashboard layout
    - Friendly and simple UI to both see and create the projects
  + Event display components
    - Real time updates
    - On click details: the user could interact with some of the components to see extra information
* Backend:
  + Data collection:
    - API integration to get the desired data
    - Handling errors when obtaining data
  + Data processing:
    - Filtering or discarding faulty data
  + Data storage:
    - Data could be stored if needed
  + Real-time Data Handling
    - Web-sockets implementation for real-time updates on funding
* Documentation
  + Architecture documentation:
    - Overview of the system design
    - Technology choices
  + User guide:
    - How to interpret the dashboard
    - How to interact with it
  + API documentation:
    - Description of the APIs used for data integration
    - Endpoints, data formats, etc
* Logs
  + Dashboard usage logs
  + Error logs
* Testing
  + Unit testing:
    - Test of individual components
  + Integration testing
    - Ensure that API and real-time data work cohesively
    - Ensure that the data is received in a proper format
    - If the format is not correct, discard the data or remove the errors
* Deployment
  + Web deployment
  + Docker deployment

# Project organisation

## Stakeholders and team members

*<<Indicate all stakeholders and team members for your project. For each stakeholder indicate the role for your project. Note that the role that a person has for your project is different from the function the person has. E.g., someone with the function “department manager of department X” can have the role of product owner for your project.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| Mateo González Alonso  glzalomateo@gmail.com | MGA | Developer of the application | Available on request |

## Communication

*<< Indicate the meetings and other channels of communication that you have established, or that you use for your project. Think of communication with all stakeholders including company supervisor, teachers, etc.*

*In which manner does each communication take place? Think of the goals, the location (or whether it should be online), the timing and frequency, and the attendee list>>*

As an individual developer, my communication will be primarily with my teachers, either in person, via email, or through Microsoft Teams. In-person meetings will be used for discussing complex issues and receiving direct feedback.

Email will handle formal communications, such as progress updates and submitting deliverables, while Microsoft Teams will be used for quick questions and virtual meetings when necessary. These channels will ensure I stay aligned with project goals and receive the guidance needed to succeed.

# Activities and time plan

## 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 development phases of my project will align with the course sprints, allowing me to progress at the same pace as we're taught the relevant technologies. This synchronization ensures that I can immediately apply what I learn in class to my project, reinforcing my understanding and building each phase on a solid foundation. By focusing on specific aspects of the project that correspond with the current sprint such as front-end development during front-end sprints. I can effectively integrate new skills as they are taught, leading to a more cohesive and well-executed final product.

# Testing strategy and configuration management

## Testing strategy

*<<Which testing strategy do you envision? E.g., on which levels will testing take place? Consider that you could choose unit, component, integration, system, or acceptance testing.*

*Justify your strategy, and also set goals where relevant. E.g., percentage code coverage for the relevant unit tests. For each of the planned tests, indicate what will be automated and what not.*

*Also think of quality testing setups like, e.g., Sonarqube.*

*>>*

The primary testing strategy for this project will focus on component testing,. However, integration testing will also play a crucial role to ensure seamless communication between the front-end and back-end systems. The strategy will follow a multi-level approach incorporating unit and integration testing to ensure overall quality, functionality, and robustness of the solution.

1. Unit Testing

* Unit testing will focus on individual functions and classes, ensuring that each piece of code works as expected in isolation.
  + Unit tests provide the foundation for detecting and fixing bugs early, ensuring the correctness of the code before moving on to more complex testing stages.
* Goal:
  + Achieve at least 80-90% code coverage for unit tests, especially for critical components such as data manipulation logic, input validation, and UI utilities.

2. Integration Testing

* Integration tests will validate the interactions between the front-end and back-end services, ensuring that data is passed and displayed correctly, APIs and real-time data are consumed without errors, and overall system behavior is maintained.
  + Given that the dashboard relies heavily on communication between front-end and back-end systems, it's essential to verify that all integrated services and components work harmoniously. This will also help in identifying issues like data format mismatches, API errors, and timing issues.
* Goal:
  + Ensure that all critical data flows (e.g., fetching data from the server, updating components) work seamlessly with minimal latency.
  + Test error handling for common failure cases like API timeouts or authentication failures.

## Test environment and required resources

*<< Describe the test environment. E.g., do you envision a DTAP (Development, Testing, Acceptance, Production) environment. Can you make use of a CI/CD environment or will you develop your own?*

*It often helps to use a picture to visualize the test environment.*

*If you already know, describe which resources are required for realization and testing. Think of hardware, cloud environments and specific tooling required for development and testing.*

*>>*

I will be using a pipeline in GitHub in order to include a CI/CD so I can use it for my testing environment

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

## Configuration management

*<< Describe the project approach with respect to version management (e.g. your GIT repository). This might include things like tooling, branching strategy, promotion-, release- and baseline strategy.*

*Also, when relevant, think of a mechanism to deal with change requests and problem reports.>>*

I will mainly work in a single branch in GitHub

# Finances and risk

## Project budget

*<< If specific budget is required for your project, indicate it here, and also what needs to be done to get budget approval. Think of hardware, applications, libraries, development environments, etc.*

*Regular costs that have already been covered, like an internship compensation, do not need to mentioned.*

*>>*

## Risk and mitigation

*<< Investigate and define all risks affecting the project. For each risk indicate what has been done, or will be done during the project, to prevent the risk from being actualized, and define the mitigation actions, such as what you plan to do if the risk actually eventuates. Think both from an organizational perspective about risks (e.g. sudden unavailability of the company mentor) and also from a content perspective (e.g. what happens if your research shows that it is a better to purchase an application than to develop it as a major part of your internship).*

*In a more elaborate version, you can also label the risks with their chance of occurrence and impact. The advice is to focus on risks that have both a real chance of eventuating and some considerable impact. Direct risks, like what to do if your company supervisor is not available anymore, should always be described, as they have happened in the past quiet regularly.*

*>>*

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
|  |  |  |
|  |  |  |
|  |  |  |