**Project Plan**

***Access Control Visualization***

*Simac IDS*

|  |
| --- |
| **Date** **:** **18/03/2021** |
| **Version** **:** **1.0** |
| **State** **:** **First draft, awaiting feedback** |
| **Author** **:** **Georgi Dimitrov, Aleks Petrov, Steven Verberne &**  **Sytse Walraven** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1.0 | 18/03/2021 | Georgi Dimitrov Aleks Petrov Steven Verberne Sytse Walraven | Initial project plan, with the following features: Project assignment, organization, activities & time plan, testing strategy & configuration management, and risk analysis. | First draft, awaiting feedback |
|  |  |  |  |  |
|  |  |  |  |  |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
| 1.0 | 18/03/2021 | Christian van Deuren, Georgios Metaxas |
| 1.1 | 25/06/2021 | Christian van Deuren, Georgios Metaxas |

**Table of contents**

[1. Project assignment 4](#_Toc66883168)

[1.1 Context 4](#_Toc66883169)

[1.2 Goal of the project 4](#_Toc66883170)

[1.3 Scope and preconditions 4](#_Toc66883171)

[1.4 Strategy 4](#_Toc66883172)

[1.5 Research questions 4](#_Toc66883173)

[1.6 End products 5](#_Toc66883174)

[2. Project organization 6](#_Toc66883175)

[2.1 Stakeholders and team members 6](#_Toc66883176)

[2.2 Communication 6](#_Toc66883177)

[3. Activities and time plan 7](#_Toc66883178)

[3.1 Phases of the project 7](#_Toc66883179)

[3.2 Time plan and milestones 7](#_Toc66883180)

[4. Testing strategy and configuration management 8](#_Toc66883181)

[4.1 Testing strategy 8](#_Toc66883182)

[4.2 Test environment and required resources 8](#_Toc66883183)

[4.3 Configuration management 8](#_Toc66883184)

[5. Risk and mitigation 9](#_Toc66883185)

[5.1 Purpose and Need not well-defined: 9](#_Toc66883186)

[5.2 Incomplete project design and deliverable definition: 9](#_Toc66883187)

[5.3 Difficulty in defining and understanding project schedule: 9](#_Toc66883188)

[5.4 Risk factors related to disputes: 9](#_Toc66883189)

[5.5 Communication issues: 9](#_Toc66883190)

[5.6 Risk related to errors: 10](#_Toc66883191)

[5.7 Lack of control over staff priorities: 10](#_Toc66883192)

[5.8 Resource suppliers’ risk: 10](#_Toc66883193)

[5.9 Escalating project conflicts not reported timely: 10](#_Toc66883194)

[5.10 Delay in projects: 10](#_Toc66883195)

# Project assignment

## Context

Christian van Deuren – our product owner – is the development & operations manager at Simac IDS (Identification Solutions), a branch of the IT company Simac that focusses on access control and identification solutions.

## Goal of the project

One of Simac IDS’s solutions, Pronto, is a real-time access control system. It offers real-time information about any location at any time, and gives on-site access based on this information. This solution, however, is a bit lacking on the visual side of configuration and monitoring. To address this issue, Christian has provided as with the following request: Can we make the Access Control System configuration visually more attractive.  
Preferably, this solution allows for the customer to create their own maps and floorplans and add access control areas to them. Furthermore, it would allow for the configuration of readers and controllers, and monitoring these readers, controllers, and areas. This would make Pronto more accessible to new and existing customers.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| Creating maps and floorplans (open to ideas, but importing is preferred) | Post-finilasation product support |
| Drawing zones and dragging/dropping controllers and readers and editing these later if necessary. |  |
| Monitoring of zones and searching functionality for finding controllers and readers on the maps. |  |
| Software must be General Data Protection Regulation (GDPR) complicit. |  |
| Multilingual software for even broader accessibility. |  |

## Strategy

For this project we will be working with the scrum methodology, but with a new scrum master each sprint. This allows for the clear oversight that daily communication from scrum offers, with the additional benefit of giving everyone the ability of developing and displaying their leadership skills. A possible additional strategy we want to integrate in our scrum process is pair programming. This allows for direct code review and sharing of ideas and knowledge during the process.

## Research questions

To ensure we reach an endproduct that not only pleases the product owner but also our own expectancies for finishing this project in a successful way, we’ve decided to focus on these 4 ICT research methods the gain an apply knowledge to our project.

### Design pattern research

We need to find the relevant UX/UI design standards to please the PO, since the main focus of this project is on visualization. For the back-end, we will also need to find up-to-date research and documentation on how to structure our code, database and possible other solutions.

### Interview/stakeholder analysis

We must always be sure that we are on the same line as the product owner when it comes to recreating their desires during this project. Whenever we introduce a new feature or design element, it should be discussed with the PO as soon as possible to make sure we waste as little time as possible.

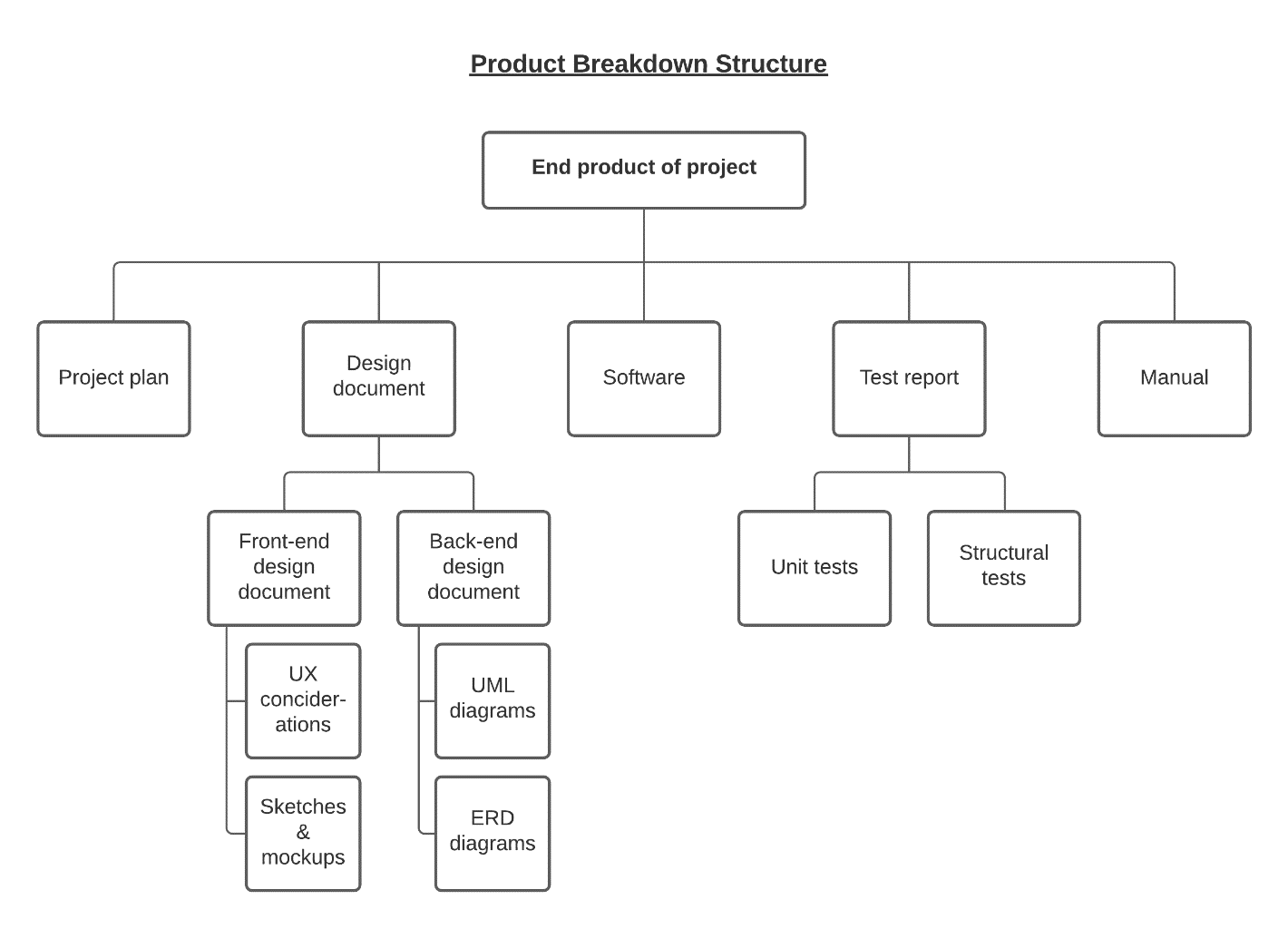
### Unit test/system test

For ensuring code quality and usability of our end product, we will need to apply constant unit test during the process (with automated CI/CD when possible) to ensure code quality and consintency, and to make sure that the product is intuivtive for the actual end users. Researching the optimal methods of testing will be an important step at the beginning of this project.

### Guideline conformity analysis

Since GDPR compliance is important when handling personal information, we will need to research how we should handle personal information and what we can/can’t share, store and modify. If we research these rules early on in the project, we can make sure that we don’t have to modify all our code at the end.

## End products



# Project organization

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Email** | **Role and functions** | **Availability** |
| Christian van Deuren | Christian.van.deuren@simac-ids.com | Product Owner | On request via email |
| Georgios Metaxas | g.metaxas@fontys.nl | Semester Coach /  Group project teacher | Every Thursday from 9.00 - 16.00 & via Teams |
| Georgi Dimitrov | Georgi.dimitrov@student.fontys.nl | Developer | Every Thursday, Friday from 9.00 -16.00 |
| Aleks Petrov | aleks.petrov@student.fontys.nl | Developer | Every Thursday, Friday from 9.00 -16.00 |
| Steven Verberne | Steven.verberne@student.fontys.nl | Developer | Every Thursday, Friday from 9.00 -16.00 |
| Sytse Walraven | s.walraven@student.fontys.nl | Scrum Master | Every Thursday, Friday from 9.00 -16.00 |

## Communication

The communication in our group will be done through MS Teams. Any additional or new features, problems will be discussed with the Product Owner. We will meet at least twice a week with development group via MS Teams or at the Fontys Campus to keep on track with our progress. Meetings with the product owner will occur at the end of each sprint, or if necessary, during the sprint as well.

# Activities and time plan

## Phases of the project

### Sprint A

* Initiating: Form a project group and assign roles
* Problem Analysis: Analyse the problem, setup requirements and create backlog
* Planning: Plan sprint deliverables
* Setup: Setup working environment and tooling

### Sprint B - D

* Development: Work on current sprint backlog
* Demo: Present a demo of the current progress to the Product Owners
* Reflection: Review sprint and peers

### Final Sprint

* Development: Work on current sprint backlog
* Handover: Deliver the final product to the Client
* Evaluation: Final product evaluation by the Client
* Wrap up: Finishing up the project

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| Sprint A | Setting up the project e.g. project plan and tooling | 08.02.2021 | 26.03.2021 |
| Sprint B | Working on MVP | 29.03.2021 | 16.04.2021 |
| Sprint C | Continue development from backlog | 19.04.2021 | 16.05.2021 |
| Sprint D | Continue development from backlog | 17.05.2021 | 04.06.2021 |
| Final Sprint | Finishing up the project | 07.06.2021 | 23.06.2021 |

# Testing strategy and configuration management

## Testing strategy

We chose Structural Testing Strategy, because in this way we can test every new function and component that is coded and, in the future, we can set up Git pipelines in order to test our code, before pushing in into the repository and furthermore to the production server.  
  
We’re planning to have 100%-unit test coverage and all the tests are going to be run after pushing into the repository.

Integrating Sonarqube is going to be a real advantage for our static code analysis in our project which is going to help us to have more quality code and to guide us to follow the best principles of coding.

## Test environment and required resources

We envision to use DTAP environment, because the project flow is going to be a lot smoother and even if untested code has been pushed into the repository it won't make it into the production severs.

We will need the actual tests, Git pipelines, Docker, a server in which the application should be hosted, and a script (to pull, run the tests, rebuild the project, rebuild the Docker image) to have an ease when we should push the repository safely into the production server.

## Configuration management

We’re going to have the main project and the most up to date working version in the master branch.

Every feature is going to have separate branch and when there is a new update that has to be delivered in production, the branch is going to be merged from the developer branch into the master one. If there is a big new functionality or refactor that has to be made, it is going to be developed into a separate branch.

# Risk and mitigation

## Purpose and Need not well-defined:

The first project risk example is the risk related to the need and purpose of the project. This is a medium type of risk, but it can get transferred to the high project risk category if the project is impacted by this factor.

Mitigation: It is important for any organization to complete a project plan if it has not been provided beforehand. Also, the need and purpose of the project must be mentioned and defined accurately.

## Incomplete project design and deliverable definition:

The second project risk example is incomplete project design and deliverable definition. It is a low-risk factor but can eventually turn out to be a high-risk factor if not controlled beforehand.

Mitigation: It’s important that all project members take the project design into a consideration beforehand. Doing that the risk can be prevented efficiently.

## Difficulty in defining and understanding project schedule:

Every project must have a specific time period to be completed. If there is no set schedule or if there is difficulty in understanding the project schedule, then this project risk example will arise. It is included in the low-risk category but can turn out to cause a medium risk to the project.

Mitigation: Team meetings are important in such cases. It is advisable for all to conduct schedule meetings with the team members. This will help them manage time efficiently and avoid missing tasks.

## Risk factors related to disputes:

A project is handled by many people and it is likely to happen that disputes can arise due to different thoughts and different expectations. So, therefore, this is included in the project risk examples.

Mitigation: The way to avoid such risks is to conduct meetings on a regular basis and let all the team members and project related personnel participate so that the issues can be discussed openly, and a relevant solution is provided as soon as possible.

## Communication issues:

One of the other project risk examples includes the communication channel between the people related to the project. Due to lack of communication, there will be no clarity, and instead, confusion will arise which will be stressful for the efficient running of the project.

Mitigation: To prevent such risks, the communication plan must be established considering the audience, frequency, and goal of the project. Along with the plan the right channel of communication is to be established through emails, phone calls, in written and so on.

## Risk related to errors:

Another project risk example is related to the errors. The team members must not be forced to complete tasks in a limited time period as this will increase the possibility of getting errors. This type of project risk also arises when the team is working under pressure.

Mitigation: To avoid such risks, it is important for the project team to have enough time to complete the tasks and space must be given to the workers to do the task as per their skills and efficiency levels. The task duration must not be reduced certainly as it will increase the possibility of getting errors.

## ****Lack of control over staff priorities:****

The next project risk example is related to the staff members. If a project fails to create a backup for team members, then the project will be delayed which is indeed a negative aspect that may give rise to other risk factors.

Mitigation: To prevent this risk factor, a project manager must take the initiative to brief out the importance of the project to the other managers. The manager should schedule the dates of performing each task and provide backup for every team member. In case anyone leaves the project team, time must not be wasted in finding another candidate suitable for the profile. Instead, a backup must be kept ready to avoid such risks.

## ****Resource suppliers’ risk:****

Another project risk example include risk related to the resources. If the project does not acquire the required resources on time, the project will face many problems.

Mitigation: To avoid such risks, resources must be gathered before starting the project by analyzing the requirements efficiently.

## ****Escalating project conflicts not reported timely:****

This gives rise to another project risk examples. According to this, conflicts arising in the team and outside the team are not handled timely due to which the conflicts arise. This is a low to medium risk factor if it causes an impact on the project.

Mitigation**:** Regular project team meetings must be conducted, and conflicts must be looked out and resolved.

## Delay in projects:

Delay in competing for earlier project causes this risk to occur in the current project.

Mitigation: To avoid such risks, trackers must be made to analyze the task completion and delays must be reported immediately so that the relevant action plan can be made to complete the tasks as soon as possible.