

Project Plan

Guidance

Philips

Date	15-09-2020
Version	0.3
State	For approval
Author	Guidance → Nick Beijerbacht, Dylan Arts, Martijn Bassa, Denis Atanasov, Dimitar Petrov & Yessin Ajouaou

Version history

Version	Date	Author(s)	Changes	State
0.1	14-09-20	Nick Beijerbacht, Dylan Arts, Martijn Bassa, Yessin Ajouaou, Denis Atanasov	Updated points 1., 2. and 5.	In Progress
0.2	15-09-20	Nick Beijerbacht, Dylan Arts, Martijn Bassa, Yessin Ajouaou, Denis Atanasov, Dimitar Petrov	Updated points 1., 2., 3., 4., 5. and table of contents	In progress
0.3	05-10-20	Nick Beijerbacht, Dylan Arts, Martijn Bassa, Yessin Ajouaou, Denis Atanasov, Dimitar Petrov	Reworked points: 1,2,3,4 and 5	In progress

Distribution

Version	Date	Receivers
0.2	15-09-20	Kleijnen, Carli C

Table of content

Version history	3
Table of content	5
1. Project assignment	6
1.1 Context	6
1.2 Goal of the project	6
1.3 Scope and preconditions	Fout! Bladwijzer niet gedefinieerd.
1.4 Strategy	7
2. Project organisation	8
2.1 Stakeholders and team members	8
3. Activities and time plan	10
3.1 Phases of the project	Fout! Bladwijzer niet gedefinieerd.
3.2 Time plan and milestones	Fout! Bladwijzer niet gedefinieerd.
4. Testing strategy and configuration management	Fout! Bladwijzer niet gedefinieerd.
4.1 Testing strategy	12
4.3 Configuration management	12
5. Finances and risk	Fout! Bladwijzer niet gedefinieerd.
5.2 Risk and mitigation	13

1. Project assignment

1.1 Context

Our company 'Guidance' consists of a team of 6 enthusiastic students that want to do whatever they can to produce an outstanding product to help others.

The project is a safety measure for big buildings that want a safe route through their building during the times of a pandemic or fire, so all the employees can travel through the building safely.

The product becomes a web application that can be displayed on a monitor in the building or on a user's telephone. The idea is that the web application displays a heat map of the floor plan. In warm places the population is relatively high. In the event of a pandemic, the application indicates a route to the outside with the least population. In the event of a fire, the application indicates the shortest route to the nearest emergency exit with the least congestion.

The heat map is generated by the application estimating the user location. The estimate is made on the basis of the data retrieved from the routers.

1.2 Goal of the project

The aim of the project is to provide safe passage for people inside the building to the outside of the building, this could be during a pandemic or during a fire outbreak.

Although the current Covid virus pandemic is (hopefully) almost over, this does not mean that The team can stop taking precautions. The team must prepare for a possible next pandemic. With this project minimizing the spreading of the virus within large buildings is a priority. For example business premises or educational institutions.

It can also be used for other purposes, for example in the event of a fire outbreak. At that point, the user will be led outside to the nearest (emergency) exit via a route with the least congestion.

Because companies or educational institutions want to improve safety within the organization, they will want to purchase and install this product.

1.3 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 Automatic routing application, focused on minimal user interaction and more on background algorithmic calculations in order to provide safe routing aid on demand or in emergency situations. It also consists of	1 Algorithm to extract a usable grid out of a floorplan
2 Setup guide	2
3 Technical documentation	3
4 Project plan	4

<< 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

It has to be a web application.

From Philips the team will need the following items: a GitHub repository, a deployment server and expertise.

>>

1.4 Strategy

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

The team is gonna work with scrum. The team thinks scrum is a good way to keep track of the state of the project. This way members will be more motivated when they know others are also working hard on their parts within the project. Scrum helps looking back to what has been done so adjustments can be made to make sure the next sprint goes more fluid.

To show this to the users The team will make a UI as simple and straightforward as possible. This way people will need the least possible attention for our application as possible. This so that if there is an added feature like fire detection and people need to get out of the building as fast as possible they can put almost all their attention in the real world instead of looking on their phone while running.

>>

2. Project organisation

2.1 Stakeholders and team members

Name	Abbreviation	Role and functions	Availability
Ajouaou, Yessin Y.	<i>Yessin</i>	<i>Member of Guidance, Developer</i>	<i>~2.5 days a week</i>
Arts, Dylan D.	<i>Dylan</i>	<i>Member of Guidance, Developer, UX</i>	<i>~2.5 days a week</i>
Atanasov, Denis D.H.	<i>Denis</i>	<i>Member of Guidance, Developer(Back- end, Support)</i>	<i>~2.5 days a week</i>
Bassa, Martijn M.A.	<i>Martijn</i>	<i>Member of Guidance, Developer</i>	<i>~2.5 days a week</i>
Beijerbacht, Nick N.H.P.	<i>Nick</i>	<i>Member of Guidance, Developer (Front-end and Back-end)</i>	<i>~2.5 days a week</i>
Petrov, Dimitar D.Z.	<i>Dimitar</i>	<i>Member of Guidance, Developer</i>	<i>~2.5 days a week</i>

Heuts, Alexander	<i>Alexander</i>	<i>Product Owner</i>	<i>1 day a week</i>
Bergmans, Liesbeth	<i>Lisebeth</i>	<i>Expert</i>	<i>1 day a week</i>
Kleijnen, Carli C.	<i>Carli</i>	<i>Stakeholder</i>	<i>3 days a week</i>

3. Activities and time plan

3.2 Time plan and milestones

<< For a waterfall project you can indicate the phases and milestones below (can be adapted as required).

Sprints will last 3 weeks each and there will be a total of 6 sprints. The 3 weeks is just enough time to finish most of the goals for a sprint. The team will do a stand up at the start of every week where each member will discuss what has been done, what there is to do and what the possible issues exist that they can run into. At the end of each sprint they will demo (if possible) the product to the product owner and listen to incoming feedback. After the demo a retrospective meeting will be held, in which they will discuss the positives and negatives of our work process.

Phasing (Sprints)	Start date	Finish date
1 Phase 1 (Sprint 1)	21 September	6 October
2 Phase 2 (Sprint 2)	12 October	3 November
3 Phase 2 (Sprint 3)	9 November	24 November
4 Phase 2 (Sprint 4)	30 November	15 December
5 Phase 3 (Sprint 5)	4 January	19 January

<< 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.

Sprints will last 3 weeks each and in total there will be 6 sprints. The 3 weeks is just enough time to finish most of the goals for a sprint. A stand up is held every day, the team discusses what each individual has worked on, they explain if they need help or more time. A stand up consists of 2 minutes per individual so that is a total of 12 minutes in our team. The team will do

a standup to keep the PO and experts updated on how far the team has gotten with the work they agreed with the PO.

>>

4. Testing strategy and configuration management

4.1 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.

>>

4.3 Configuration management

<< Describe the project approach with respect to version management. 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.

The documentation will have version management, this way it'll be clear what work has been done and when it was done. Every delivery the team will hand in a version, all changes after the delivery will be the next version of the document. The team will use continuous integration. The team will integrate their code early and often to the main branch. People will work on their individual parts, this way people can constantly update their changes without interference. The idea is to automate tests to be able to run for every change that is made to the main repository, hereby capturing issues early and reducing disruptions for the team. They do need to give clear notes of what they changed or added. In case of a request or report the team will handle that during the times the team works together in voice chat. The team will use separate branches for each member, which also helps with CI.

>

5. Finances and risk

5.2 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.

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 quite regularly.

>>

Risk	Prevention activities	Mitigation activities
1 Members of the team not sticking with the group schedule and/or missing meetings	Have a conversation with them and adjust future scheduled meetings accordingly to everybody's availability	Talk to the stakeholders and the rest of the team. Collectively decide how to deal with the situation.
2 The route algorithm not functioning properly	Researching in detail if the algorithm is possible before starting the project.	Reconsider the project (Chokepoint)
3 Software data loss due to device malfunction or damage	Keep up-to-date version of the software on multiple devices and/or remote repositories	Change approach to data storage in order to avoid future data loss and proceed with the

		latest available version of the software.
--	--	---