**Project Plan**

***IvantiMarkt***

*Ivanti*

|  |
| --- |
| **Date : 23.09.2021** |
| **Version : 1** |
| **State : State** |
| **Author : Group 3** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1 | 23.09.2021 | Group 3 |  |  |
| 2 | 08.11.2021 | Group 3 | Test plan moved to here |  |
|  |  |  |  |  |

**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)

# Project assignment

## Context

The company that we will be working with is Ivanti. They reached to us in order to create a marketplace where customers can download packages. Users also have the possibility to become content creators by uploading packages.

## Goal of the project

The goal of the project is to make a web application which will serve as a marketplace for Ivanti.

The marketplace will display an overview of all the available packages, the user being able to search for packages or filter the results, depending on his desires.

A customer can become a content creator by uploading his own package on the marketplace and managing the use of his package (free or charged).

With this project we strive to achieve a pleasant experience for the upcoming users, who might be landing on Ivanti's page for the first time and an unique exposure for each of our frequent users, making each interaction special. We hope that with the help of this web interface we will be able to thin the barrier between the user and provider, thus making further projects easier to connect with our versed users and supporters.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Frontend using Angular and JavaScript | 1. Test database |
| 1. Backend using Java | 1. Website content |
|  | 1. Style guide |

## Strategy

For this Ivanti Marketplace project, we will use an agile scrum approach, because we find it essential tobe in constant contact with the teammates and to follow the progress. The requirements are not yet final, so we will continuously add or improve the existent ones, based on the client’s needs. The sprints are organized every 3 weeks, where we will have a demo for the product owner.

## 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.>>*

*Documents: Project Plan, Product Backlog,, Jira Board,*



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Contact name (and specify further detail as needed, e.g., email or tel nr).* | *Abbreviation can help, e.g., when using the name in tools like Jira or MS project.* | *See above.* | *When is the person available for your project (define this in the way most relevant for your project, e.g., which days are available, the amount of time, or in which phase of the project).* |
| Dennis Smits  [dennis.smits@ivanti.com](mailto:dennis.smits@ivanti.com) | Dennis Smits | Product owner | Available online, via mail. He will be present on some workshop classes, depending on his schedule. |
| Nicole Zuurbier-Munneke  [n.munneke@fontys.nl](mailto:n.munneke@fontys.nl) | Nicole Zuurbier-Munneke | Mentor, teacher | Monday between 9:00 – 12:00 and Thursday between 13:00 – 16:00, during the workshop classes. Available for questions online, via mail or teams. |
| Ana Maria Nitu  [a.nitu@student.fontys.nl](mailto:a.nitu@student.fontys.nl) | Ana | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |
| Bianca Onea  [b.onea@student.fontys.nl](mailto:b.onea@student.fontys.nl) | Bianca | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |
| Vid Barbaro  [v.barbaro@student.fontys.nl](mailto:v.barbaro@student.fontys.nl) | Vid | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |
| Tomislav Tonchev  [t.tonchev@student.fontys.nl](mailto:t.tonchev@student.fontys.nl) | Tomislav | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |
| Plamen Peev  plamen.peev@student.fontys.nl | Plamen | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |
| Matei Mitran  [m.mitran@student.fontys.nl](mailto:m.mitran@student.fontys.nl) | Matei | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |
| Ioan Nicolae Popa  i.popa@student.fontys.nl | Ioan | Developer | Monday, Thursday group meetings, when we have group project workshop. Every day for individual work. Available for meetings scheduled within the group, online or offline. |

## Communication

This project we need to keep close contact not only with the teammates, but also with the product owner and the teacher.

For the organization of the tasks, project requirements, user stories, sprints, progress, tasks assigning, we will use Jira Software. As a group, the majority of the meetings will take place offline, in the Fontys building, at least 3 times a week.

For the short announcements or any kind of discussion within the group, we use a WhatsApp group with all the teammates.

The communication with the product owner will be either offline, when he is present in the Fontys building, or online, via email.

We have workshops with the teacher twice a week, in which we can present our work, and she can help us with any impediment or question. The workshops take place in the building.

# Activities and time plan

## Phases of the project

The project is divided in 6 sprints, every 3 weeks, where we have the presentation and demo for the product owner.

The meetings within the group will be every week, at least 3 times a week, where we discuss what we worked on, the current situation and the future plan for working on the project.

## Time plan and milestones

For this project, we are required to work during 6 sprints. The sprint has a length of 3 weeks, during which we, as a team, are expected to improve the product by adding new features. During this time, the university provides us with classes and workshops that have the scope of extending our knowledge within this field.

During each sprint, we decide on what we have to work on, we get familiar with the requirement we have to fulfill and discuss our progress and ideas.

At the end of the sprint, we have to present our work to the product owner, where we have the chance of receiving constructive feedback and making changes where needed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 1. Sprint A | - | 20.09.2021 | 08.10.2021 |
| 1. Sprint B | - | 11.10.2021 | 05.11.2021 |
| 1. Sprint C | - | 08.11.2021 | 26.11.2021 |
| 1. Sprint D | - | 29.11.2021 | 17.12.2021 |
| 1. Sprint E | - | 20.12.2021 | 21.01.2022 |

We want to give maximum effort for each sprint

# Testing strategy and configuration management

## Introduction

The Test Plan is designed to test the main functionality of Ivanti’s website project. The tests are broad and not specific as we are still at the start of the project. But they are going to give the main idea of whether the units and API work fine.

## 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.*

*>>*

This willl be completed later in the development process, when we will start testing the software.

### Unit tests

We have made several unit tests so we can make sure whenever something in a certain unit (class) is changed it won’t break the whole application. Also, to do that we first run the tests and make sure they pass and afterwards we run the application.

### Integration tests

As we recently studied about CI/CD we will try to actively include it in our project because this feature is pretty useful for big projects where we have many tests on pieces of code and it will be easier to have a person developing some new feature instead of testing something which can be tested by a computer. Also, unit tests can be run using CI/CD. It is easy to update and maintain the existing pipelines.

### Acceptance tests

User acceptance tests are one of the most expensive as we can say. Thus, it is because we have to gather people who have no idea how the application works and give them instructions on how to use it or just let them explore it on their own. So it is time-consuming but this is also the most efficient way to see whether their unusual/unexpected actions are going to break the app. There we have some steps and expected results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test description** | **Step #** | **Test Steps** | **Expected Result** | **User Story** |
| 1 | Verify that the user can make an account  (Postman) | 1 | The application is launched | Your account was successfully made.  And the account is in the database. |  |
| 2 | Navigate to Sign up menu  Fill in all required fields |
| 2 | Verify user can log in  (Postman) | 1 | Navigate to log in | 1. User hasn’t made a registration yet. 2. User logs in successfully 3. User clicks forgotten password |  |

### SonarQube

Using SonarQube we can see easily how maintainable and usable is our code. We can see security vulnerabilities, warnings of probable bugs and more.  
 --- picture after including it

## 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.*

*>>*

*As we already described we are going to try to make use of CI/CD so the testing of the components and their work can be easier and continuous.*

*This willl be completed later in the development process, when we will start testing the software.*

## 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.>>*

*This willl be completed later in the development process, when we will start testing the software.*