**Reader's Guide portfolio Jorn Kersten**

Student: Jorn Kersten

Student number: 483331

Date: 18-10-2022

Semester: S-DB-IPS3-S3-DB03

# Version control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Modification |
| 1.0 | Jorn Kersten | 18-10-2022 | Initialisation of the readers guide |
| 1.1 | Jorn Kersten | 28-11-2022 | Updating text on individual project |
| 1.2 | Jorn Kersten | 13-12-2022 | Added following items to individual project: [UX design](#_UX_design), [Researches](#_2.2_Researches), [Software design](#_2.3_Software_design), [ORM](#_2.5_ORM)  Initialized [group project](#_Group_project) and [reflection](#_Reflection) |
|  |  |  |  |

Table of contents

[Version control 2](#_Toc121928321)

[Introduction 5](#_Toc121928322)

[Learning outcomes 6](#_Toc121928323)

[1.1 You design and build **user friendly**, **full-stack** web applications. 6](#_Toc121928324)

[1.2 You use software **tooling and methodology** that continuously monitors and improve the software quality during software development. 6](#_Toc121928325)

[1.3 You **choose** and implement the most suitable agile software development method for your software project. 6](#_Toc121928326)

[1.4 You **design and implement** a (semi)automated software release process that matches the needs of the project context. 7](#_Toc121928327)

[1.5 You **recognize** and **take into account** cultural differences between project stakeholders and ethical aspects in software development. 7](#_Toc121928328)

[1.6 You analyse (non-functional) requirements, elaborate (architectural) designs and validate them using **multiple types of test techniques**. 7](#_Toc121928329)

[1.7 You analyse and describe **simple** business processes that are **related** to your project. 8](#_Toc121928330)

[1.8 You act in a **professional manner** during software development and learning. 8](#_Toc121928331)

[Individual project 9](#_Toc121928332)

[2.1 Description 9](#_Toc121928333)

[2.2 Researches 9](#_Toc121928334)

[2.2.1 Agile (wip) 9](#_Toc121928335)

[2.2.2 Frameworks 9](#_Toc121928336)

[2.2.3 Database 10](#_Toc121928337)

[2.2.4 Security (wip) 10](#_Toc121928338)

[2.3 Software design (wip) 10](#_Toc121928339)

[2.4 UX design 10](#_Toc121928340)

[2.5 ORM 11](#_Toc121928341)

[2.5 Quality assurance (wip) 12](#_Toc121928342)

[2.6 CI/CD (wip) 12](#_Toc121928343)

[2.6.1 CI 12](#_Toc121928344)

[2.6.2 CD 12](#_Toc121928345)

[Group project 13](#_Toc121928346)

[3.1 Description 13](#_Toc121928347)

[3.1.1 About World of content 13](#_Toc121928348)

[3.1.2 Assignment 13](#_Toc121928349)

[3.2 Workflow 13](#_Toc121928350)

[3.3 Software design 14](#_Toc121928351)

[3.4 UX design 14](#_Toc121928352)

[3.5 My contribution 15](#_Toc121928353)

[3.6 What I’m proud of 16](#_Toc121928354)

[Reflection 16](#_Toc121928355)

# Introduction

This document serves as the readers guide for my semester 3 portfolio. The portfolio contains the developed products from this semester, using these products I will show proof I have gained enough knowledge to fulfil the requirements of the learning outcomes. In this guide you will find a short summary of each product per learning outcome. Thereafter there are a few links to the files where a longer worked out version of the product can be found.

--------------------------------------------------------------------------------------------------------------------------------------

In this document you will find a few sections. The learning outcomes section contains all the learning outcomes and a description of what knowledge is required to show the understanding of the learning outcome. The research section…

--------------------------------------------------------------------------------------------------------------------------------------

# Learning outcomes

## 1.1 You design and build **user friendly**, **full-stack** web applications.

**Clarification:**

|  |  |
| --- | --- |
| **User friendly:** | You apply basic User experience testing and development techniques. |
| **Full-stack:** | You design and build a full stack application using commonly accepted front end (JavaScript-based framework) and back end techniques (e.g. Object Relational Mapping) choosing and implementing relevant communication protocols and addressing asynchronous communication issues. |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You design and build **user friendly**, **full-stack** web applications.”

[Software design](#_2.3_Software_design)

[UX design](#_2.4_UX_design)

[ORM](#_2.5_ORM)

## 1.2 You use software **tooling and methodology** that continuously monitors and improve the software quality during software development.

**Clarification:**

|  |  |
| --- | --- |
| **Tooling and methodology:** | Carry out, monitor and report on unit integration, regression and system tests, with attention for security and performance aspects, as well as applying static code analysis and code reviews. |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You use software **tooling and methodology** that continuously monitors and improve the software quality during software development.”

Link 1 (ex. UX design)

## 1.3 You **choose** and implement the most suitable agile software development method for your software project.

**Clarification:**

|  |  |
| --- | --- |
| **Choose:** | You are aware of the most popular agile methods and their underlying agile principles. Your choice of a method is motivated and based on well-defined selection criteria and context analyses. |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You **choose** and implement the most suitable agile software development method for your software project.”

[Link 1 (see SCRUM board)](#_2.1_Description)

## 1.4 You **design and implement** a (semi)automated software release process that matches the needs of the project context.

**Clarification:**

|  |  |
| --- | --- |
| **Design and implement:** | You design a release process and implement a continuous integration and deployment solution (using e.g. Gitlab CI and Docker). |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You **design and implement** a (semi)automated software release process that matches the needs of the project context.”

Link 1 (ex. UX design)

## 1.5 You **recognize** and **take into account** cultural differences between project stakeholders and ethical aspects in software development.

**Clarification:**

|  |  |
| --- | --- |
| **Recognize:** | Recognition is based on theoretically substantiated awareness of cultural differences and ethical aspects in software engineering. |
| **Take into account:** | Adapt your communication, working, and behaviour styles to reflect project stakeholders from different cultures; Address one of the standard Programming Ethical Guidelines (e.g., ACM Code of Ethics and Professional Conduct) in your work. |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You **recognize** and **take into account** cultural differences between project stakeholders and ethical aspects in software development.”

Link 1 (ex. UX design)

## 1.6 You analyse (non-functional) requirements, elaborate (architectural) designs and validate them using **multiple types of test techniques**.

**Clarification:**

|  |  |
| --- | --- |
| **Multiple types of test techniques:** | You apply user acceptance testing and stakeholder feedback to validate the quality of the requirements. You evaluate the quality of the design (e.g., by testing or prototyping) taking into account the formulated quality properties like security and performance. |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You analyse (non-functional) requirements, elaborate (architectural) designs and validate them using **multiple types of test techniques**.”

Link 1 (ex. UX design)

## 1.7 You analyse and describe **simple** business processes that are **related** to your project.

**Clarification:**

|  |  |
| --- | --- |
| **Simple:** | Involving stakeholders, predominantly sequential processes with one or two alternative paths. |
| **Related:** | Business processes during which the software that you are developing will be used (business processes that the software must support by fully or partially automating them).  or  Business processes needed for the success of your software development project (e.g., product release, market release, financial assurance). |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You analyse and describe **simple** business processes that are **related** to your project.”

Link 1 (ex. UX design)

## 1.8 You act in a **professional manner** during software development and learning.

**Clarification:**

|  |  |
| --- | --- |
| **Professional manner:** | You actively ask and apply feedback from stakeholders and advise them on the most optimal technical and design (architectural) solutions. You choose and substantiate solutions for a given problem. |

**Proof:**

In the following places I will show I have gained enough knowledge of the learning outcome:

“You act in a **professional manner** during software development and learning.”

Link 1 (ex. UX design)

# Individual project

## 2.1 Description

My individual project that I have created is a website on which prices of equal products from multiple different supermarkets are compared. A user is able to conveniently see the price of an product and clearly sees if the prices is the cheapest from all supermarkets. They’ll also be able to see where the price at a specific supermarket has risen, stayed the same or decreased in comparison to the week before.

If the user wants to, they can login to the website and create a personal list of items they would like to see. They then to can filter by supermarkets within their own list. Outside of just comparing a list, they can also create a shopping list here the cheapest overall suggestion to order the selected products is shown to the user.

I have used the SCRUM framework for keeping track of my tasks, and I’m using git as a tool to keep a SCRUM board in which I can easily keep track of progress and a sprint planning. I have a total of 5 sprints of 3 weeks each. [My SCRUM board](https://github.com/users/Jorn-Kersten/projects/2/views/4).

## 2.2 Researches

### 2.2.1 Agile (wip)

Dit nog doen maat (onderzoeken nog)

### 2.2.2 Frameworks

Before creating the front-end, I researched a few frameworks and scoured the internet to see which framework would work best for me. I then weighed every advantage and disadvantage and made a choice based on what I thought would best suit my application.

The front-end framework I chose is Angular, I chose this framework instead of Vue.js or react. I chose Angular because it was completely new to me, this framework was most similar to C# which I was already comfortable with. I was eager to learn something new. Since I already have knowledge of React, that was going to be very difficult, which is why I didn't go for React. Beyond that, both React and Vue.js have the entire stack of a page in one file. Therefore, programming in React and Vue.js does not allow me to follow SOLID principles, which I would like to do.

For the back-end, I researched which programming language is best for creating an API. Here, I mainly looked at security of the programming language. I also considered the way the language works.

The two programming languages I compare are C# and Java. I briefly told you about both languages and then wrote down their features. Then I put the differences of both languages together, from this I made the choice to use Java.

This because I wanted to learn something new and because there was also a brief explanation from school about a framework using Java, which I wanted to learn and use.

[Software characterisation research](Research%20documents/software%20characterisation%20research.docx) page 4 and 8

### 2.2.3 Database

**SQL-based vs NoSQL-based**

When researching the what database I’d like to use, I found out there are two different ways of saving data in a database. An SQL-based database and an NoSQL-based database. Both have their advantages and disadvantages but based on the research I did on both of them, I found a NoSQL-based database wouldn’t be an option in my implementation.

[Software characterisation research](Research%20documents/software%20characterisation%20research.docx) page 11

**What database did I choose**

After finding out what type of database I should use in my project, I searched for what database options I had with using an SQL-based database. I compared MySQL, PostgreSQL, and MariaDB. The reason for it was that I had experience with MySQL and PostgreSQL and MariaDB both are kind of similar to MySQL.

After the research I had done I choose to go with MariaDB, as MariaDB doesn’t add many more features but it is faster in processing SQL query’s. The reason I didn’t choose PostgreSQL is that it adds a few more features like the possibility to add arrays as a column definition, which I didn’t need within my project.

[Software characterisation research](Research%20documents/software%20characterisation%20research.docx) page 13

### 2.2.4 Security (wip)

Iets over security nog invullen OWASP enzo zeg ma (OWASP uitleggen en met .dot framework onderzoeken)

## 2.3 Software design (wip)

I created user stories based on the established requirements. I divided these into functional and non-functional user stories.

Based on these user stories, I first created a conceptual model to get a clearer picture of which components I am going to have in my website. After creating the conceptual model, I created a software architecture that shows how components communicate with each other.

Software design architecture (dit nog maken vriend)

## 2.4 UX design

There are all kinds of ways to arrive at a good website design. Different design methodologies, also called UX design, have been developed for this purpose. For this, I used a method that follows five steps. Starting with identifying the target audience, to make the right product you need to know for whom you are making it. Then I worked out what features were needed for my product, this I worked out in paper sketches.

**[image]**

In the image above, you can see the wireframes and the clickable prototype, these were created with Figma.

Since these sketches are not a whole, I worked it into wireframes to get an idea of the navigation structure and formatting of the data. I then worked this into a visual prototype, all the pages are linked together here so they can be clicked through.

Clicking through is done with testing. I asked three users if they could navigate to certain pages. If all three users could do this without help, it means the design is effective and I can start implementing it.

[UX design](UX%20desgin.docx)

## 2.5 ORM

Object Relational Mapping also known as ORM, can be seen as a layer between object-oriented programming (OOP) and a relational database. When interacting with a database using OOP languages, you'll have to perform different operations like creating, reading, updating, and deleting (CRUD) data from a database. By design, you use SQL for performing these operations in relational databases.

While using SQL for this purpose isn't necessarily a bad idea, the ORM and ORM tools help simplify the interaction between relational databases and different OOP languages.

ORM tools can actually be written with any programming language. So this is a very simple way to combine OOP with a relational database. There is also a tool for this for Quarkus; our teacher Hans gave a simple workshop for this. I used the hibernate function, this is a function of the Quarkus library called Panache.



In the image above, a class is called that uses Panache to create a PanacheRepository based on the Product class. This class defines all the desired proportions that can be found in the table Product.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## 2.5 Quality assurance (wip)

Testen enzo ff nog maken hier makkertje jorn (over testen enzo)

## 2.6 CI/CD (wip)

### 2.6.1 CI

Nog invullen/maken (front- en back-end)

### 2.6.2 CD

Nog invullen/maken (front-end alleen)

# Group project

## 3.1 Description

### 3.1.1 About World of content

World of Content is driven by innovation. When we started, we were driving the e-commerce industry by introducing smart content management services that were easier, faster and more flexible than what the unwieldy, conventional providers could deliver. Now, a not-too-long period later, World of Content continues to work at a breakneck pace on innovative products that bring us closer to the dot on the horizon; automated global delivery of the perfect content tailored to the individual.

**Source:** <https://worldofcontent.com/nl-nl/about-us/>

### 3.1.2 Assignment

At the beginning of the semester, World of content gave us an assignment. This assignment is to create a dynamic scraper.

The product they we for World of content need to improve their current product. Their current product makes it very easy for a brand to put products for sale on retailers' websites. Different retailers need different types of data about the same product from a brand. World of content has a platform that puts this in a clear overview. A brand then only has to fill in the data requested in that overview instead of having to ask and check separately for each retailer.

After all the data is filled in, our assignment comes in handy. For some retailers, it can take a while for adjustments entered by brands on the World of content website to be implemented. A brand can always go and manually check whether their product on a retailer's website matches the indicated data on the World of content website. This takes a very long time and should happen very often if the brand sells a lot of products.

The solution to this: creating a scraper. Data from retailers' websites should be automatically retrieved and compared with a brand's indicated data. When the indicated data is found on the retailer's website, this data should be displayed in a clear overview.

For example, if the title is found on the retailer's website, the overview will show behind the title that it was found. The stated data of the brand and the found text on the retailer's page are displayed side by side so that this can be checked again by the brand.

If something similar is found, it should be shown in that same overview. It will then say that something was found, but it does not match the indicated data on the World of content website. If nothing at all is found, this is also indicated in the overview.

## 3.2 Workflow

Throughout the project, we worked with the project group in an agile manner. We used the scrum method during this project. For this we used GitHub as a tool. in GitHub, every sprint we keep track of what tasks we have, how far we are with these and what still needs to be done. We had already worked with this a bit, and we wanted to work with this in our individual project as well. In addition, this also helps with demonstrating the learning outcomes.

[View our Scrum board on GitHub](https://github.com/orgs/WJJCN/projects/1)

The task can be split into two directions. Creating the scraper and creating the website that displays the results. I worked with Nick to create the scraper. We started by writing crawler that retrieves all URLs on a website that are related to products on that website. We list these URLs and then compare them with the products in the database.

If a URL matches a product from the database, we retrieve the URL. From the data we get back, we extract certain elements of the page. This is always just text, we then compare this text with the data of the product that matched the URL. Before we started, we researched in which programming language a scraper could be most easily and best written. From this came Python, we then chose to write everything with Python 3.10.

## 3.3 Software design

Just after we got the assignment from World of content, we converted the requirements into user stories with the group. Then, to make it clearer for ourselves, we made a concept diagram. We based an ERD on this so that we could easily set up a database with it. Because not all data always have the same properties, we opted for a NoSQL database.

These diagrams/models and related information can be found in the software design document in the Group project directory.

[View files](Software%20documents/Software%20architecture%20(group))

## 3.4 UX design

At the beginning of the project, we spent a lot of time with the whole group to create the design of the website. I didn't create a design myself, but I gave a lot of feedback to people who did. Whenever we disagreed with the group about something in the design, we discussed it together. If we disagreed, we immediately made two options. If we were not sure what to do we asked the stakeholder what he wanted with that piece of the design.

The image below shows the final design.

[View design file](https://www.figma.com/file/QXiSsczGjRt8JTmbToN7Wz/World-Of-Content-Live-Score)



## 3.5 My contribution

Nick and I took on the task of making the scraper together. Together, we had to ensure that data is automatically and dynamically retrieved and then compared.

To get this done, we split the task: "Create a scraper that automatically and dynamically retrieves and compares data" into a three sub-tasks.

We then looked at which programming language we could best use to create the scraper. We did a little research on this. Most websites pointed to Python, with one library specifically: Beautifullsoup 4. We then chose to use the same Python version throughout the project. We are using an AWS server, on which we have lambda functions. Of these lambda functions, we cannot choose which Python version we use, here the Python version is 3.10. So this is also the version we used for our scraper.

The first task is to ensure that we retrieve the necessary links from a merchant's website. These are all links that belong to products, there are sometimes cookbooks or other unnecessary links on a retailer's website.

In the second task, we want to compare the URLs found with the products in the database. If after the beginning of the URL, i.e. from the first "/", the name of the product is found in the URL then that is probably the URL of the product.

Then comes the final task, retrieving and actually comparing data. The HTML source code is retrieved and then put into a Beautifullsoup object. This allows easy retrieval of elements from the HTML page, these are all p-tags, span-tags, h1-tags, h2-tags and li-tags. All text between these tags is then compared one by one and then stored in a list. When all tags are compared, the length of the found text is compared to the correct text from the database. Then everything is stored in the database so it can be displayed to the brands.

# Reflection

## 4.1 What I’m proud of