READERS GUIDE PORTFOLIO

S3

Welles,Nick N.H.M.

2022 - 2023

# Version control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Changes | Author | Date |
| V0.1 | Initial version. | Nick Welles | 19-10-2022 |
| V0.2 | Added parts of Individual project & group project. | Nick Welles | 02-11-2022 |
| V0.3 | Added [UX Design](#UXDesign)  Added [Data Persistence](#DataPersistence)  Added [UX Design](#_UX_Design) (Group)  Added [Software Design](#_Software_design_1) (Group) | Nick Welles | 03-11-2022 |
| V0.4 | Added column in in [learning outcomes](#_Learning_outcomes)  Added page numbers | Nick Welles | 8-11-2022 |
| V0.5 | Added [Research Business process](#_Business_Process)  Added [CI](#_CI) | Nick Welles | 16-11-2022 |
| V0.5.1 | Added [Quality Assurance](#_Quality_Assurance) | Nick Welles | 29-11-2022 |
| V0.6 | Updated [Software Design](#_Software_design):   * Added asynchronous functionality * Changed Software Architecture | Nick Welles | 30-11-2022 |
| V0.7 | Added [UX Design](UX%20Design.docx) explanation file  Added [CD](#_CD)  Updated [Agile](#_Agile)  Added [My contribution](#_My_contribution) | Nick Welles | 5-12-2022 |
| V0.7.1 | Added [Software tests](Research/Software%20tests.docx) | Nick Welles | 7-12-2022 |
| V0.8 | Added [Cultural differences and ethics](#_Cultural_differences_and)  Updated [Software tests](Research/Software%20tests.docx)   * Added End to end testing   Updated [My contribution](#_My_contribution) | Nick Welles | 8-12-2022 |
| V0.8.1 | Added Reflection of outcomes  Updated [Software Design](#_Software_design):   * Added Messaging and events | Nick Welles | 13-12-2022 |
| V0.8.2 | Updated [Software tests](Research/Software%20tests.docx)   * Added important tests for my project | Nick Welles | 14-12-2022 |
| V0.9 | Updated [Quality Assurance](#_Quality_Assurance) | Nick Welles | 22-12-2022 |
| V0.9.1 | Updated [Agile](Research/Agile.docx) | Nick Welles | 23-12-2022 |
| V0.9.2 | Updated [Agile](Research/Agile.docx)  Updated [Cultural differences and ethics](Research/Cultural%20differences%20and%20ethics.docx) | Nick Welles | 30-12-2022 |

# General information

* For course: S-DB-IPS3 and S-DB-GPS3
* Class: S3-DB03
* Coaches: Marc M.H. van Grootel, Hans J.B.H.M Heumen
* Date: December 30th 2022
* Version: 0.9.2

Portfolio for semester 3 of the bachelor's program of IT from Fontys University of Applied Sciences.

# Table of content

[1 Version control 1](#_Toc122612867)

[2 General information 1](#_Toc122612868)

[3 Table of content 2](#_Toc122612869)

[4 Introduction 3](#_Toc122612870)

[5 Learning outcomes 3](#_Toc122612871)

[6 Research 5](#_Toc122612872)

[6.1 Security 5](#_Toc122612873)

[6.2 Agile 5](#_Toc122612874)

[6.3 Techstack 5](#_Toc122612875)

[6.4 Business Process 5](#_Toc122612876)

[6.5 Cultural differences and ethics 6](#_Toc122612877)

[7 Individual project (Instruweb) 6](#_Toc122612878)

[7.1 Description 6](#_Toc122612879)

[7.2 Software design 6](#_Toc122612880)

[7.3 UX Design 7](#_Toc122612881)

[7.4 Data Persistence 7](#_Toc122612882)

[7.4.1 ORM 7](#_Toc122612883)

[7.4.2 Database 8](#_Toc122612884)

[7.5 Quality Assurance 9](#_Toc122612885)

[7.6 Software Release Management 10](#_Toc122612886)

[7.6.1 CI 10](#_Toc122612887)

[7.6.2 CD 10](#_Toc122612888)

[8 Group project (World of Content web- scraper/crawler) 11](#_Toc122612889)

[8.1 Software design 12](#_Toc122612890)

[8.2 UX Design 12](#_Toc122612891)

[8.3 My contribution 12](#_Toc122612892)

[9 Reflection 14](#_Toc122612893)

[9.1 Reflection 13-12-2022 14](#_Toc122612894)

[9.2 What I have learned 15](#_Toc122612895)

[9.3 What went well 15](#_Toc122612896)

[9.4 What can be improved 15](#_Toc122612897)

# Introduction

This document fills in as a reading guide for my portfolio from semester 3 of the Four year certification program in Information Technology at Fontys University of Applied Sciences. The portfolio contains the items that I have created during the semester, demonstrating that I have acquired adequate information to meet the set necessities in the last fulfilment levels. This guide gives a concise outline of every item and segment, and points the reader towards the files which contain the fully worked-out versions of the subject.

# Learning outcomes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # |  |  |  |  |
| 1 | Web application | You design and build **user-friendly**, **full-stack** web applications. | **User friendly**: You apply best practices when creating user interfaces and basic user experience testing and development techniques.  **Full-stack**: You design and build a full stack application using a commonly accepted front end JavaScript framework and back end application implementing relevant communication protocols, persistence of data by usage of ORM and addressing asynchronous communication issues. | See:  [UX Design](#_UX_Design_1)  [ORM](#_ORM)  [UX Design (group)](#_UX_Design)  [Software Design](#_Software_design) |
| 2 | Software quality | You use software **tooling and methodology** that continuously monitors and improve the software quality during software development. | **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. | See:  [Security](#_Security)  [Quality Assurance](#_Quality_Assurance)  [My contribution (group)](#_My_contribution) |
| 3 | Agile method | You **choose** and implement the most suitable agile software development method for your software project. | **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. | See:  [Agile](#_Agile)  [Group project](#_Group_project_(World) |
| 4 | CI/CD | You **design and implement** a (semi)automated software release process that matches the needs of the project context. | **Design and implement**: You design a release process and implement a continuous integration and deployment solution (using e.g. Gitlab CI and Docker). | See:  [CI](#_CI)  [CD](#_CD) |
| 5 | Cultural differences and ethics | You **recognize** and **take into account** cultural differences between project stakeholders and ethical aspects in software development. | **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. | See:  [Cultural differences and ethics](#_Cultural_differences_and) |
| 6 | Requirements and design | You analyse (non-functional) requirements, elaborate (architectural) designs and validate them using **multiple types of test techniques**. | **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. | See:  [Techstack](#_Techstack)  [Software Design](#_Software_design)  [Software Design (group)](#_Software_design_1) |
| 7 | Business processes | You analyse and describe **simple** business processes that are **related** to your project. | **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). | See:  [Business process](#_Business_Process) |
| 8 | Professional | You act in a **professional manner** during software development and learning. | **Professional manner**:  You develop software as a team effort according to a prescribed software methodology and following team agreements. You are able to track your work progress and communicate your progress with the team.  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. | See:  [Security](#_Security)  [Software Design (group)](#_Software_design_1) |

# Research

Over the span of this semester, I have carried out a ton of groundwork into new innovations, business cycles and strategies.

## Security

When you develop and publish a Web application, you expose it to a potentially malicious environment. There are those who scour the Internet looking for security vulnerabilities in order to exploit them and cause damage or steal sensitive data. When you are developing a Web application, you naturally want to make sure that your application has as few security problems as possible, preferably none at all! the assignment for this research was to properly research exactly which security problem fits your project. My answer to that, all of them. you obviously don't want your application to have vulnerabilities if you don't pay attention to such security problems.

In this research document I have done research to a lot of available risks given, and how to prevent them. This research relates to learning outcomes 2 and 8: Software Quality and Professional.

[View file](Research/Security.docx)

## Agile

Software development is in many cases upheld by Agile practices. During this semester I have utilized an Agile method called Scrum for my individual project and my group project. I also used the story mapping method to map out all the user stories that I made for my individual project. Yet, there are numerous other Agile methods accessible to utilize and each has their own utilization cases and advantages. I have carried out some analysis into the meaning of Agile, the various methods that are accessible, and the way things are utilized by and by. This item demonstrates my capability at learning outcome 3: Agile method.

[View file](Research/Agile.docx)

## Techstack

Of course, a Web application does not spring up out of nowhere. For this, you use various languages to build the most complete application possible. Since we had a number of workshops concerning possible choices for a front-end and backend framework, I researched what would possibly be the best choice for a front-end framework, backend framework, in what database the data would be stored and possibly an API in between. This product helps prove my proficiency at learning outcome 6: Requirements and design.

[View file](Research/Techstack.docx)

## Business Process

In this section, I examined business processes and how they are applied in software engineering. I also created an example of a business process.

In addition, I worked out my own business process here. This is the business process of my individual project.

[View file](Research/Business%20process.docx)

## Cultural differences and ethics

In this section I examined the definition about cultural differences and ethics in general, and in software engineering. I also gave my own experience regarding the two.

[View file](Research/Cultural%20differences%20and%20ethics.docx)

# Individual project (Instruweb)

Inspired by: [Bax-shop](https://www.bax-shop.nl/)

## Description

Instruweb is my individual project. It is a web-based shop for purchasing music instruments or things that musicians need to play their gigs or what not.  
The goal for me was to learn how to build a web shop from the ground up without using a pre-build content management system. This seemed like a good challenge to me for semester 3.

I’ve made use of the Agile method while working on this project. Since GitHub has various tools that use the Agile method I chose this. from the Agile method, I used the Scrum form because I was already familiar with this and it complements the learning outcomes.

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

A good software engineer can easily adapt to new programming languages, techniques and frameworks. This is also something I always strive for. As an ICT person, you never stop learning!

Having used .NET, C# and as frameworks ASP.NET a lot in recent years, in this semester I have chosen to build the front-end with Angular, and build the API/Back-end with Java with the framework Quarkus (REST API)

Since I had no previous experience with Java, I was informed by my subject teachers by what are the advantages and disadvantages of the framework Quarkus and Java in general. To get started with the knowledge, there was an online workshop to get acquainted with Java. This workshop had been given by an outside company. In addition, our subject teacher Hans gave another online workshop explaining how to write simple API calls within Quarkus.

As described above, I chose Angular as the frontend framework. After doing some research, this framework seemed to fit best with my way of programming, especially the SOLID principles and the description of my assignment. Although Angular is less popular than Vue and React, I still had a better feeling about Angular. I have worked with Angular before in 2018 when I was an intern at an IoT company in Arcen.

Angular didn't have a lot of examples but fortunately enough can be found on the Internet and the documentation is well written.

## Software design

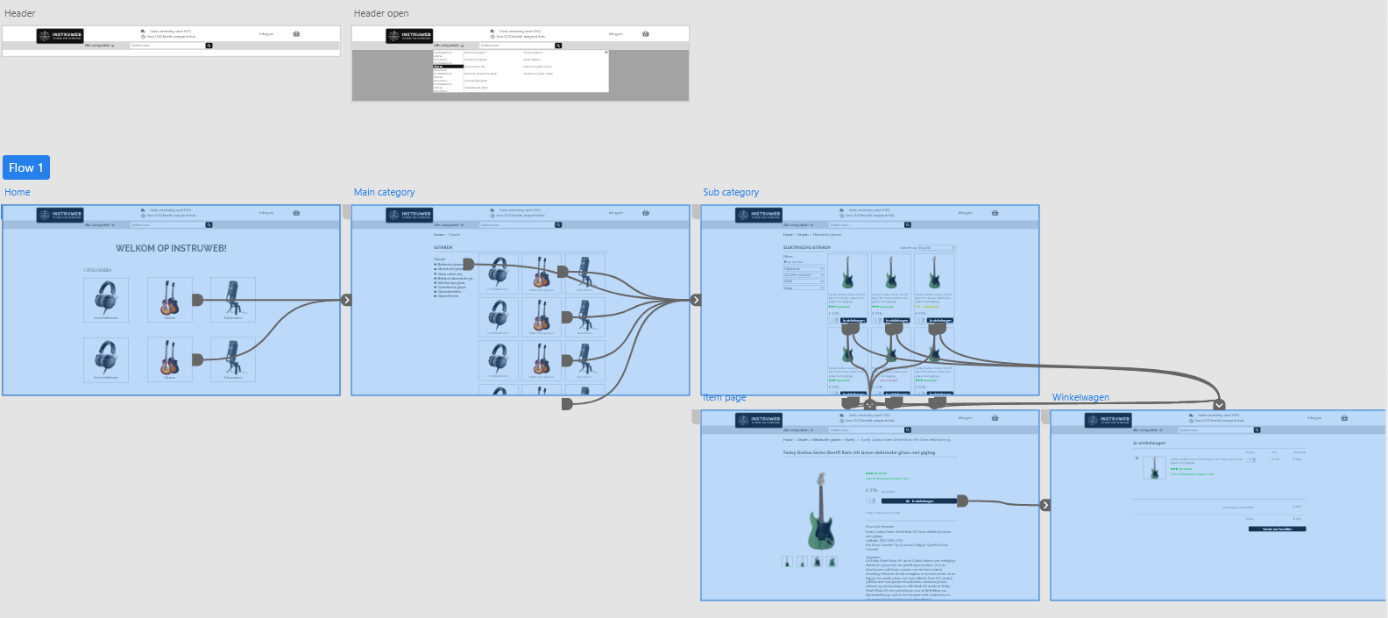
For this project, I created such user stories. To make it a little easier for me in terms of layout also a concept diagram and some images related to the software architecture.  
Information and explanations about async functionality, messaging and events can also be found here. With, of course, how these were applied in my individual project.

These diagrams/models and explanations and related information can be found in the software design document for Instruweb. This section is part of proving my proficiency at learning outcome 6: Requirements and Design.

[View file](Software%20design.docx)

## UX Design

UX design is a field of work devoted to assisting a user with clearing their path through a digital process or product with insignificant exertion and greatest worth. Since the UX design field is consistently advancing and somewhat new, we could hold returning to this definition and changing it — users, their concerns, and how visual developers tackle them will change that rapidly.



As seen in the image above, I created a fairly simple (UX) design with a flow in it. The user is instructed in advance where to navigate to and only when the user finds it clear enough they agree to the design and it is built in as a web application.

The design together with the flow was made using the Adobe XD program. There are various design tools (like Figma) that offer almost the same functionality. This section helps prove my proficiency at learning outcome 1: Web application.

More explanation can be found in the UX design document.

[View file](UX%20Design.docx)

## Data Persistence

Persistence in computer science is defined as ‘the characteristic of state that outlives the process that created it’. Take, for example, writing code to write data to external files for future use. The code which wrote these files has since stopped running, but the files that were created still exist. That is persistence.

### ORM

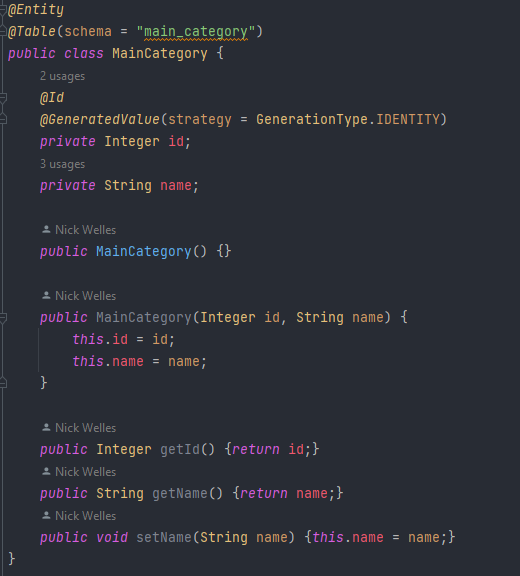
Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm. When talking about ORM, most people are referring to a library that implements the Object-Relational Mapping technique, hence the phrase "an ORM".

An ORM library is a completely ordinary library written in your language of choice that encapsulates the code needed to manipulate the data, so you don't use SQL anymore; you interact directly with an object in the same language you're using.

In the case of Java and Quarkus, I used the Hibernate function. Quarkus has a library on this called Panache. Our subject professor Hans also gave a short online workshop on this at the time.

*@ApplicationScoped  
public class* MainCategoryRepository *implements* PanacheRepository<MainCategory> {}

Described above, this class (MainCategoryRepository) implements the PanacheRepository part with the MainCategory object as reference. In it, all the properties of the table are defined. (see image below)



This section helps prove my proficiency at learning outcome 1: Web application.

### Database

For my project, I researched different forms of storing data in databases. I was already familiar with MySQL, but since I have been working with these for several years, I thought it would be a better challenge to go for another way. I researched PostgreSQL and MariaDB. It came out that PostgreSQL was just a little too complex for the application I wanted to build so I chose MariaDB.

Research can be found in the Techstack document under the Database section.

[View file](Research/Techstack.docx#Database)

## Quality Assurance

To understand exactly what is meant by quality assurance, I first researched many types of software testing methodologies. I worked this all out in a research paper. Eventually I did not apply them all, but I did find out for myself which ones were applicable to my project. In the Quality Assurance document I’ve documented out all the tests I’ve applied to my project.

[View Quality Assurance file](Quality%20Assurance.docx)

To ensure that the software I wrote for my individual project was actually reliable, I wrote a number of integration tests. Both for the front-end application and the backend application.

Regarding the front-end, it is more to see if the components in Angular are actually created with certain content on the pages. For the backend, it is more testing the application with mock data and whether it can then find and pass the data. This is where the HTTP endpoints are tested for.

In the front-end, I used Karma testing, which comes with Angular.  
In the backend, I used Quarkus Tests. With this you can easily test the services and endpoints.

In [this](Quality%20Assurance.docx) file you can view a more detailed explanation about the kind of tests I wrote for the front- and backend.

[View file](Quality%20Assurance.docx)

For the backend, I linked to the file directory where all resources and services are tested. The image below shows a test example of creating a product.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Here I create a mock-product with some data.  
From the given() function I will test an HTTP endpoint with which I pass the mock-product as JSON body in the request. When the complete endpoint is executed, I expect a status code 201 from it. That is the code indicating that a new object has been created.

Together with these test I also made a few security tests. These can be found [here](https://github.com/Instruweb/Backend/blob/main/src/test/java/com/instruweb/resource/UserResourceTest.java).

## Software Release Management

### CI

To ensure that the software is easy to test, I applied CI. To do this, I used GitHub Actions. This is a free tool that you can use to do CI/CD. I created a workflow for this that every time a push is made to the main branch the software will build and test. I used an additional branch, development, during the development process. By using another branch you can be sure that you don't interfere with the main/master branch and if CD is included it won't constantly deploy. I did CI for the front-end as well as the backend.

Afbeelding met tekst, schermafbeelding, monitor, scherm

Automatisch gegenereerde beschrijving

[View front-end CI file](https://github.com/Instruweb/Front-end/blob/main/.github/workflows/ci.yml)

[View backend CI file](https://github.com/Instruweb/Backend/blob/main/.github/workflows/ci.yml)

### CD

To ensure that the integrated code is actually deployed, I used a CD workflow.

My front-end application is deployed to Google Firebase after a fully approved merge and push between the development and main branches. For this, I have an npm package (firebase-tools) installed in my application. I also created a new workflow file for this purpose.

Afbeelding met tekst, schermafbeelding, monitor, scherm

Automatisch gegenereerde beschrijvingThe deployed application can be found on <https://instruweb-cd.web.app>

[View front-end CD file](https://github.com/Instruweb/Front-end/blob/main/.github/workflows/cd.yml)

# Group project (World of Content web- scraper/crawler)

World of Content is driven by innovation. When they started, they 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 them closer to the dot on the horizon; automated global delivery of the perfect content tailored to the individual.

World of Content came to Fontys explaining what kind of product they wanted that would make the work of a brand easier.

They were looking for a web scraper or web crawler that searched for information from a brand (e.g., Red Bull) on a retailer's website (e.g., Jumbo). Once this information was found, they wanted to see if this data found on the website also matched the data that the brand (i.e., Red Bull) provided to the retailer (i.e., Jumbo).

We were asked to make a maintainable and scalable application that they could use.

We made use of the Agile method while working on this project. Since GitHub has various tools that use the Agile method we chose this approach. from the Agile method, we used the Scrum form because we where going to use that in our individual project and we were already familiar with this and it complements the learning outcomes.

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

Jorn and I were mainly responsible for creating the web crawler and scraper. First, we created a crawler that will look by product name on a website for all existing URLs with that product name in them. Then we store those URLs in a small text file. Next, we are going to scrape these URLs. We then look at certain elements on the page. From those, we retrieve the bare text and compare this retrieved bare text with the data in the database. All this was written with Python version 3.10.

## Software design

Together with my group we setup user stories. To make it a little easier for ourselves, we created a concept diagram. From the EER we created an ERD, that way we could set up a database and that way we also know how to set up the ORM.

These diagrams/models and related information can be found in the software design document in the Group project directory. This section is part of proving my proficiency at learning outcome 6: Requirements and Design and learning outcome 8: Professional.

[View file](Group%20project/Software%20Design.docx)

## UX Design

In the group project, we spent a fair amount of time making all the designs and the process involved. Below in the image are all the designs we created, and below that is the final design we continued development with.

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



Tom (who was our contact during development) reviewed the designs and also the design department at WoC gave us tips, as well as compliments on the designs they liked the most.  
Ward was the one in charge of making the most of the designs. Jarno and I each made our own iterations so that we would have more options for the final result/iteration.

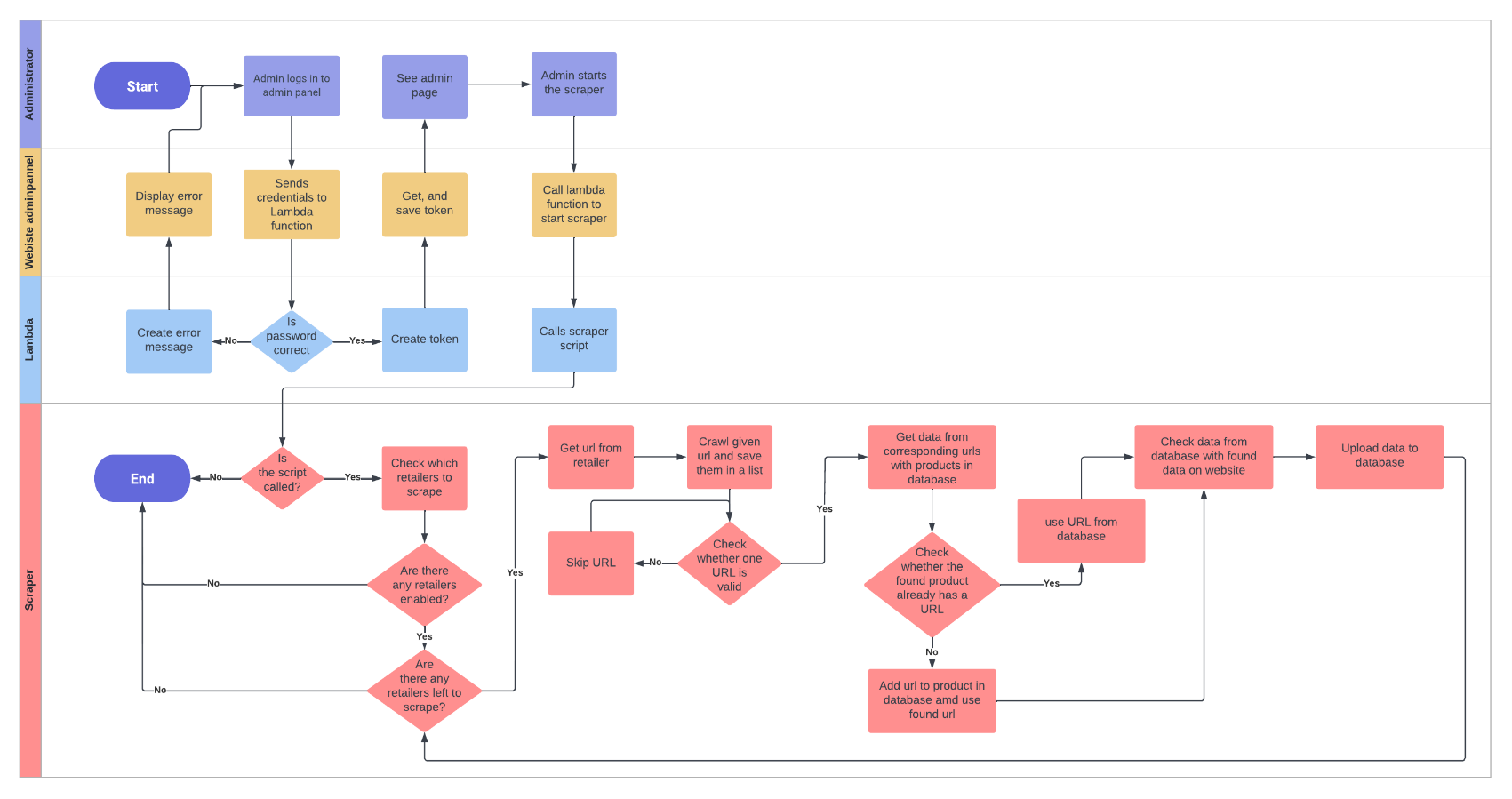
This section helps prove my proficiency at learning outcome 1: Web application. And helps to prove my proficiency at learning outcome 8: Professional.

## My contribution

In WoC's project, Jorn and I were assigned to the backend. Together we made sure that the data from WoC was compared with the data that is on the website of one of the retailers (e.g. Jumbo, AH etc).  
To accomplish this we created two scripts in Python (version 3.10). By using the BeautifulSoup library we can enter a URL and then start crawling for URLs throughout the website. We use these URLs to start retrieving the data when a product is located in the URL. We retrieve these products from our database which consists of data we got from WoC.

Together with Jorn we made a business process diagram of the complete scraper seen in the image below. More of these flows and designs can be found in the documentation repository of our group project.

[View repository](https://github.com/WJJCN/Documentation/tree/main/Documentatie/Afbeeldingen)



[View image in full size](https://raw.githubusercontent.com/WJJCN/Documentation/main/Documentatie/Afbeeldingen/WoC%20-%20Business%20process%20diagram%20scraper.png)

I'm most proud of the feature that goes to see if a product in the database is also located in one of the URLs the crawler found. The function is very big, and yes, it should be divided in multiple functions, but in our case that would destroy the flow of the script.

In the picture below is a small portion of the function which I will explain.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

This part of the function checks if the product which I got from the database already has a URL, if so, the comparer (which compares the data from the database with the page, written by Jorn) gets the product and the URL. If the URL from the product is empty, the script will update the product in the database with the found URL. And if both statements are incorrect, the script will give the comparer the found URL by the crawler with the complete product.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

*The product with the found URL.*

[View function](https://github.com/WJJCN/WJJCN-Backend/blob/backend-development/linkscraper/main.py#L241)

# Reflection

## Reflection 13-12-2022

1. **Web application**

Proficient: I understand well how it works from start to finish. New features I get implemented fairly easily and this is also fun to do.

1. **Software Quality**

Beginning: I already have a fair amount of unit/integration testing in both my backend and front-end. These are not yet enough tests to achieve this learning outcome with a proficient. I still need to be able to add performance testing, regression testing & a static code analysis.

1. **Agile method**

Proficient: Both individually, and in the group project, we made good use of the agile methodology. Every sprint we did a review and a retrospective. We also started each day with a stand-up. The scrum board was neatly maintained. And for my individual project, I also used story mapping.

1. **CI/CD**

Proficient: In both my front and backend, I used CI. CD only in the front-end because for the backend it is quite difficult to find a good platform for students without paying too much for a hosting. I did start making docker images for the backend CD.

1. **Cultural differences and ethics**

Beginning: Because so far we haven't had a workshop or anything in terms of good guidance on this, I don't know whether to put beginning or proficient here. I have done research on both subjects and this is also reflected in the portfolio. It also contains things I have experienced myself and how I have applied them in my individual project.

1. **Requirements and design**

Proficient: For this, I used feedback from our stakeholder WoC in the group project during sprint deliveries. For both the individual project and the group project, we set up (functional and non-functional) requirements, used a UX design and even more software architect models things regarding software.

1. **Business process**

Proficient: For both my individual project and in the group project, we created different business processes. Not all are equally complex but they show what the products do in such a process.

1. **Professional**

Proficient: While making the group project, we behaved professionally towards our stakeholder WoC. We also made a cooperation contract with the group together and we always followed well the methodologies we agreed on at the beginning.

## What I have learned

…

## What went well

…

## What can be improved

…