



Design Document

OGBEIDE
GODFREY .OSAS

Contents

Introduction2

 Versions.....2

 Project Description3

 Technologies3

Frontend Framework3

Database3

Architecture.....4

SonarQube Code Quality5

Test Plan.....6

 User Acceptance Test6

 Test Strategy6

Wireframes.....6

 Home Page.....6

 Category Page8

 Login Page.....8

 Dashboard (Admin)9

CI/CD Pipeline.....10

Project unit tests implemented11

Project SonarQube implemented.....13

Introduction

Versions

Version	Changes
V1	Project description, Technologies, Planning added

V2	Technologies updated, Frontend Framework Research added, User stories, Prioritization, wireframes added.
V3	Frontend Framework Research updated APA Style, C4 model added, ERD added, CI/CD added
V4	C4 model finalized, CI/CD Updated, SonarQube Quality Analysis added, Test Strategy added
V5	User Acceptance Test Added, Burndown charts added
V6	SonarQube Analysis Updated, ERD Updated, C3 updated.

Project Description

This project is for my individual track project for the semester. The project is about an online Webshop store called Boutique Web shop Shopping cart, which deals in selling different types of sneakers, Suits, Cologne, Summer and Winter Wears, Corporate Wears, Shoes, Pants trousers, Jeans, to help t users to keep up with the latest or brands of Clothing, Colognes, Jeans, Business Attires. and hopefully buying them. The website will also be used by the store administrator who handles stock maintenance, as well as updating the webstore with the latest sneakers, wears, and also have Sales-Manager who Keeps tracks of Revenues and viewing sales which have been made in the webstore.

Technologies

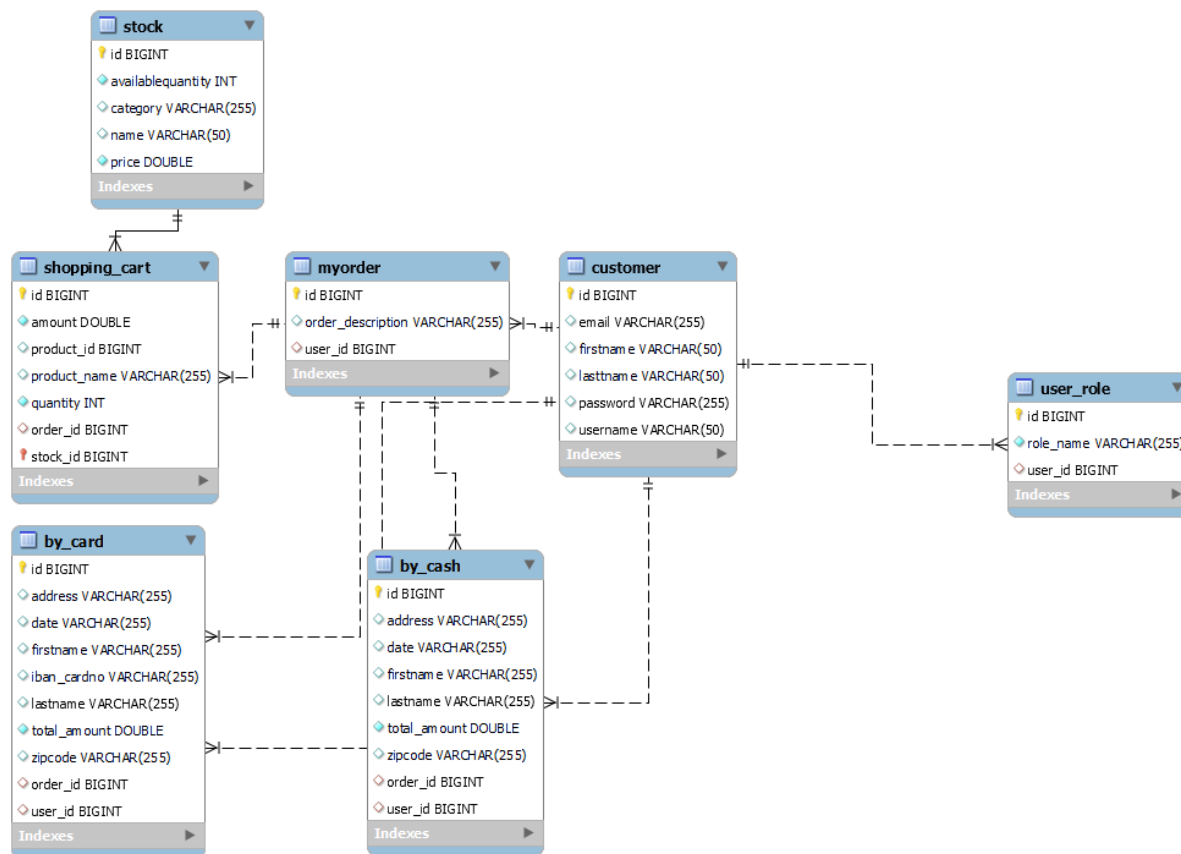
Before the development of this project, I had to decide what technologies I will be using to make the project a possibility. For the frontend I will be working with the React framework which uses JavaScript, Router and Axios, as well as CSS for designing. The reason why I chose Reacts NodeJS, is because it runs on JavaScript and typescript which are versatile, also because the angular frame possesses packages which results in more efficient development. For backend part I will be working with spring boot to create an API which will communicate with the MySQL database to access its data to be used in the frontend of the website. My reason for choosing spring boot is because it works well to support the SOLID principles enabling me to create interfaces for implement inheritance to exhibiting the open close principle.

Frontend Framework

Before I finally decided on what frontend framework to use, I carried out a bit of research to see which framework best suite my project. Below you can see the frameworks which I researched on and howl judged them.

Database

I have set up a MySQL database where I will store all the data for this project, the data will range from the users email, password and the user's type, to the information about an order which has been placed. Below is the entity relationship diagram of the database I plan to set up.



A basic description to the EDR above is that a user is able to have zero or as many orders as they like, an order consist of one or many products, and a product can belong to one or multiple orders, and a product can be long to one or many categories. This EDR will probably be updated as my project progresses.

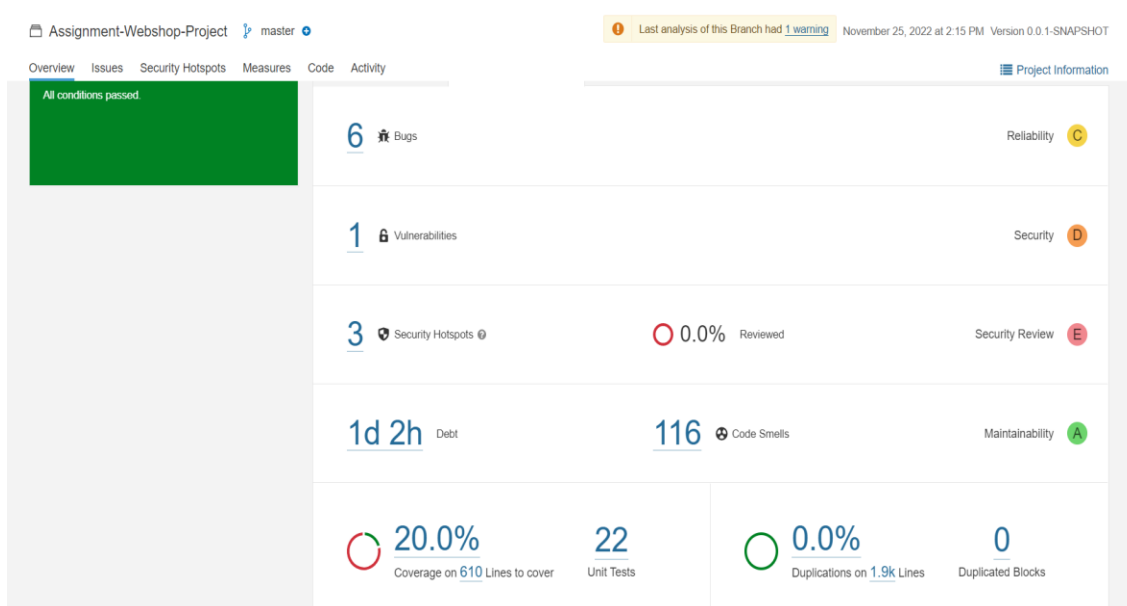
Architecture

Here I will be showing the architecture of this project and displaying all the layers with regards to the C4 model. I will be covering the first 3 C's, this is because, due to the dynamic nature of my project, I cannot provide a steady 4th stage of my C4 model.

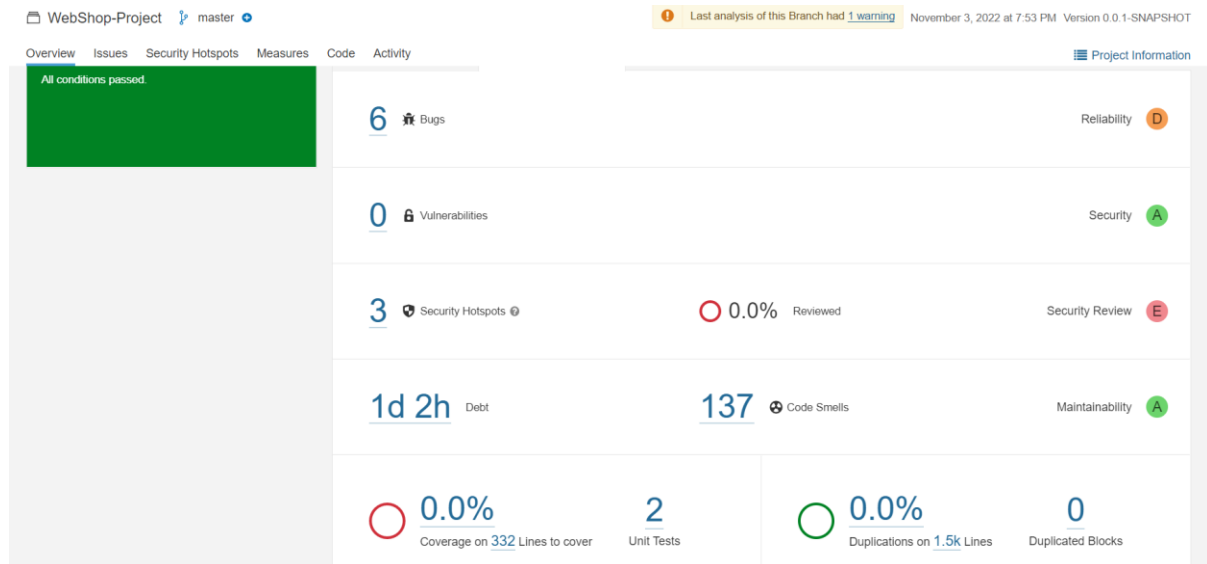
SonarQube Code Quality

One of the requirements of this project is to write my code in a SOLID way, I have explained earlier in this document what that means, and in order for me to check my code quality as I build my project, I will be using SonarQube which is an open-source software that carries our code quality checks of a project.

I have inserted a screenshot of the first code quality check using SonarQube on my project.



After seeing the initial analysis, I wanted to reduce the amount of code smells, and also the general quality of my code which received an improved analysis.



Test Plan

User Acceptance Test

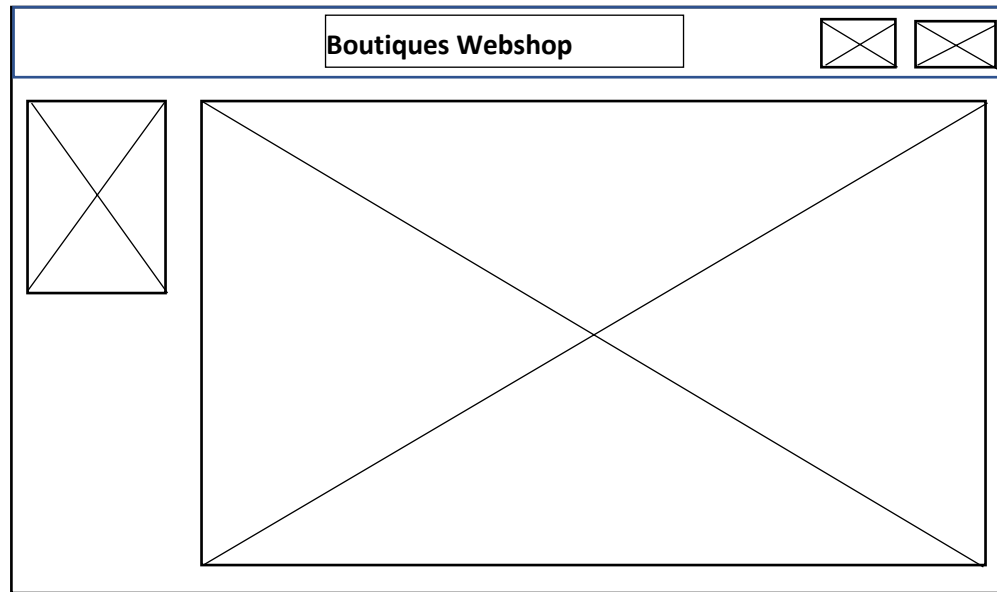
The user acceptance test are one of the methods which will be used in testing the functionality of the application. It will consist of a user attempting to complete a set of tasks to prove functionality of the agreed features of this application.

Test Strategy

Another way which I ensure functionality of this application is by carrying out two other tests on the application, unit tests and integration tests namely. Unit tests will be carried out inside the software of the application, where functions will be made to test the functionality of the code which has been written and to ensure it works as intended. While the integration test will be carried out during the integration part of the software of this application, this will be carried out using GitLab's CI/CD pipeline where I am able to test these applications in different stages.

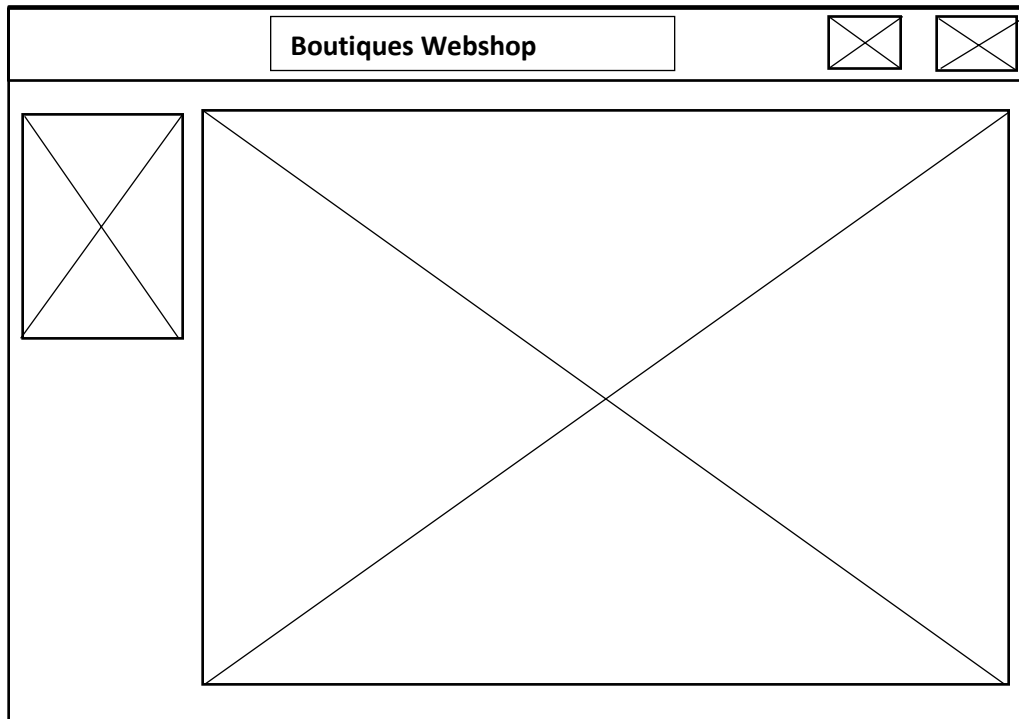
Wireframes

Home Page





The home page displays the websites routing options on the left and at the top right corner are the shopping cart icon and the log in icon.

Category Page



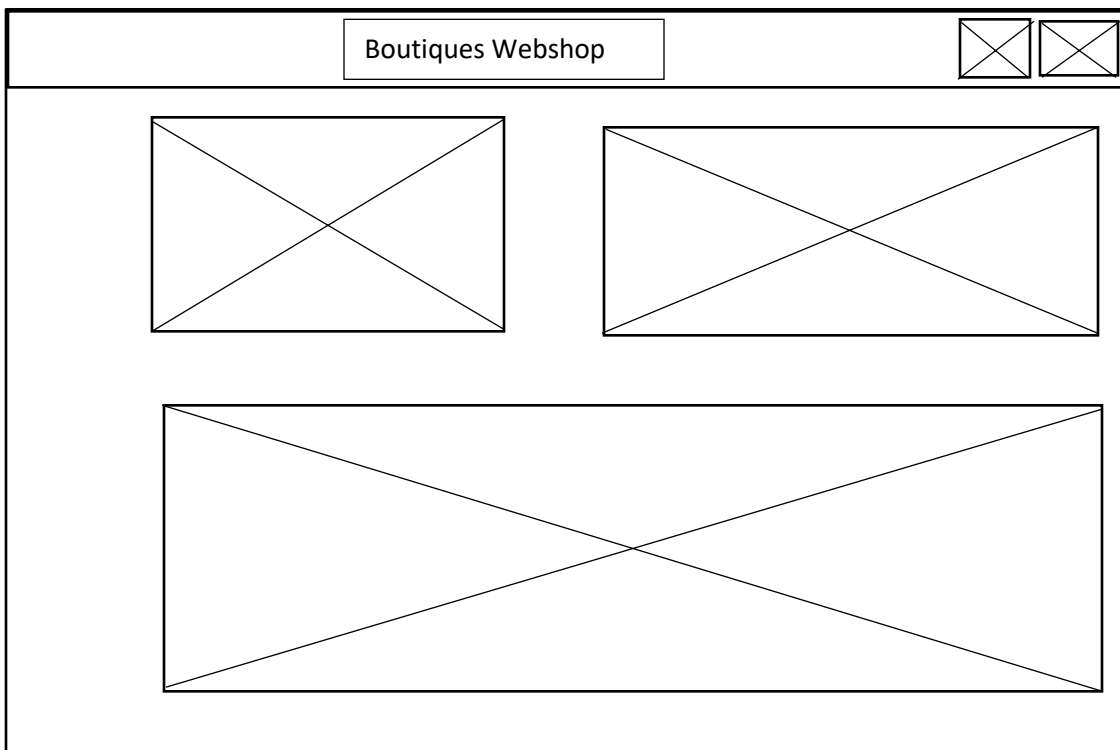
The shopping category page contains different categories of sneakers, and to the left the user is able to apply filters to select for a specific group of sneakers.

Login Page

Boutiques Webshop		 	
<div><div><div>Username</div><input type="text"/></div><div>Password</div><div><input type="password"/></div></div> <div><div>Login</div><div>Sign Up</div></div>			

On the log in page the user is able to log in to their account but could also have the option to sign up for an account.

Dashboard (Admin)



The dashboard is for the admin user, after the user is authorized after login in they are able to see the sales that have been made, a section to add, delete or edit stock.

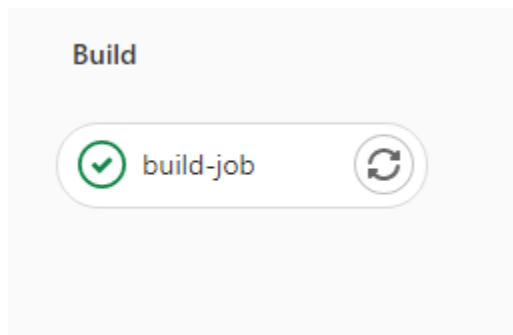
CI/CD Pipeline

Here I show the first setup of my CI/CD pipeline in my GitLab repository (<https://git.fhict.nl/1426309/mywebshop-project>). The pipeline carries out continuous integration for this project, it builds it and putting it through unit tests, quality metrics(SonarQube), and finally verifying it for deployment.

I will be showing the multiple stages/versions of the pipeline as the course progresses.

Project build implemented

This stage handles the building of the application, it helps confirm that the application is able to run successfully.

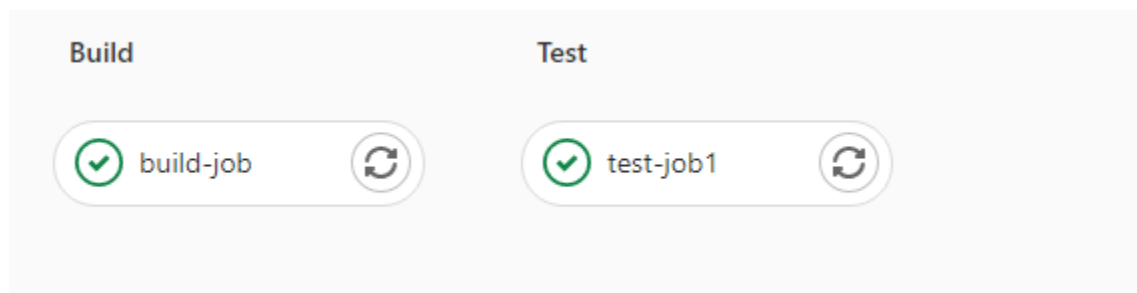


The image above is a screenshot taken for the CI/CD pipeline in my Gitlab. It shows that the build was successful.

```
✓ variables:  
  GRADLE_OPTS: "-Dorg.gradle.daemon=false"  
  
✓ stages:  
  - build  
  - test  
  - sonarqube-check  
  
✓ build:  
  stage: build  
  script:  
    - cd C:\Users\vera2\IdeaProjects\ONLINE_WEBSHOP-PROJECT\Assignment-Webshop-Project  
    - ./gradlew assemble
```

This is the script which is in the '.gitlab-ci.yml' file, in order to carry out the build test.

Project unit tests implemented



The image above is a screenshot taken for the CI/CD pipeline in my Gitlab. It shows that the tests(unit tests) were successful.

```
java:
  stage: test
  script:
    - cd C:\Users\vera2\IdeaProjects\ONLINE_WEBSHOP-PROJECT\Assignment-Webshop-Project
    - ./gradlew test
  artifacts:
    when: always
    reports:
      junit: build/test-results/test/**/TEST-*.xml
```

This is the script which is in the '.gitlab-ci.yml' file, in order to carry out the unit tests which I have written.

Project SonarQube implemented



The image above is a screenshot taken for the CI/CD pipeline in my Gitlab. It shows that the SonarQube analysis was successful.

```
sonarqube-check:
  stage: sonarqube-check
  image: gradle:jre11-slim
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar" # Defines the location of the analysis task cache
    GIT_DEPTH: "0" # Tells git to fetch all the branches of the project, required by the analysis task
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  script:
    - cd C:\Users\vera2\IdeaProjects\ONLINE_WEBSHOP-PROJECT\Assignment-webshop-Project
    - gradle sonarqube

  only:
    - main # or the name of your main branch
```

This is the script which is in the '.gitlab-ci.yml' file, in order to carry out the SonarQube analysis which I have written.