deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

António Fernandes [92880], David Morais [93147], Mariana Ladeiro [92964]
v2021-06-23

1 Introduction

1.1 Overview of the project

The project assignment was proposed in the scope of the TQS course as the final project and objectives like implementing a viable software product and a software quality assurance (SQA) strategy, to be followed throughout all of the process, should be met. The project should implement a digital marketplace for last minute deliveries by developing two sub-projects: the deliveries platform, engine, and a specific application to make use of the engine.

Engine is a delivery platform made to manage services as well as riders that will deliver the orders. Essentially, it works as a middleman by accepting services into the system and taking care of the dispatch of riders to deliver the orders coming from those services.

Book2Door is a book delivery application, allowing users to order books from their favorite stores at any time and receiving them in a matter of minutes. Book2Door is the service that leverages on Engine.

1.2 Limitations

The limitations of our project are the following:

- Search on Engine as an admin is only permitted through username
- User's can't edit their accounts in both systems
- The delivery address is currently being hard coded, since we needed to pass a certain address to coordinates (lat, long)
- Have a live tracking of where a rider delivering an order is
- Performance tests

2 Product concept

2.1 Vision statement

Book2Door is an online bookstore platform that allows its users to be connected to a large variety of bookstores as well as local bookstores, to make orders and get any type of book at any time. Book2Door was developed with the decrease of book purchases that has been visible in the past years in mind, providing a platform to attract everyone, but especially young people, into this hobby. Book delivery services are still very much in exploration and there aren't a lot of platforms that offer this type of delivery.

Engine is a general delivery system meant to deal with the dispatch of riders, as well as matchmaking between orders and riders, and offer to its users a dashboard containing a few statistics related to the certain type of user. Engine can be used with any type of delivery platform - from food to healthcare - and is supposed to make the delivery process simpler for the service that uses it.

2.2 Personas

Persona 1
Matilde Silva is a 18 year old freshman at university, studying Literature in Lisbon. Matilde loves going to the cinema, karaoke nights and getting out of the country to travel, but her favorite thing in the world are books. She used to go to the mall in her hometown to buy books every other week, but, since moving to Lisbon, and starting college, she finds it hard to find time and near places to buy books. Because of this, she is looking for a platform that will allow her to search for books and get them quickly delivered to her dorm.

Persona 2
Amélia Alves is a 48 year old woman living in Aveiro. Amélia has two sons, and loves spending time with them. She is also a natural leader, always starting small businesses to be able to have a more flexible schedule as well as manage her life the way she wants. 10 years ago, Amélia opened a bookstore, Livraria da Amélia, which was pretty successful until Covid-19 happened and Amélia's bookstore has had a big fall in sales. Because of this new reality, she is looking for a way for her store to gain customers again. Her two sons recommended joining a platform in which her books can be delivered directly to the customer's house and Amélia is now trying to find the best option for her.

Persona 3

Pedro Antunes is a 32 year old man who has just moved out to Aveiro. Pedro used to be a maths teacher, but three years ago decided to leave his job to travel around the world. Pedro is a very happy, carefree and athletic person, one of his hobbies being riding his bike every single morning to the beach. After returning from his journey, Pedro started tutoring high schoolers as well as college students, his home being his workplace. Unfortunately, because of Covid-19 Pedro had to update his work, which left him with less students than before. Because of this, he is looking for a part time job to make some extra money. Given his athletic background, Pedro wants to find a job in the delivery world.

Persona 4

Vicente Montenegro is a 38 year old entrepreneur and creator of WellWhishes. WellWhishes is a delivery platform that sends gifts to people hospitalized. Vicente just recently became a father and is having a hard time to keep up with his company as well as family life. To fix these issues, he wants to find a platform that will take care of the delivery part, making his job easier to manage.

2.3 Main scenarios

Scenario 1: Matilde orders a book from Book2Door

Matilde has a break in between zoom classes and misses reading new books so she goes to her computer and opens Book2Door to search for a new book to start reading right away. She starts by allowing the website to get her current location in which the book about to be ordered can be delivered at. A list of stores in her location range as well as some books from those stores and displayed. She decides to search for a specific author she loves, getting a list of books available by that author. She realizes she has read all of those books, so she searches for a specific store her friend told her about. She picks a book from that store and adds the book to her cart, proceeding to the checkout. After paying, she is redirected to a page where details about the tracking of her order are displayed.

Scenario 2: Amélia registers her store in Book2Door and adds books to her store catalog

Amélia opens Book2Door to make an application for her store. She fills in the needed information and waits for her application to be accepted. After a while, Amélia is able to login into her account and can now add books to her catalog which will be automatically shown to the clients. Amélia can also see the orders that were placed as well as the money made.

Scenario 3: Pedro submits his application to Engine, starts his shift and begins delivering orders

Pedro decides to submit his application to Engine. After getting accepted, Pedro starts working by selecting Start Shift in his dashboard page. Pedro receives an order and he is able to see the delivery address as well as the store address in which he will pick up the order. After picking it up, Pedro selects the Picked Up option in the Track field so that the client can get updates and after delivering it, Pedro selects the Delivered option. The order is now complete and Pedro waits for the next one. While waiting for another order, Pedro takes a look at the money made as well as the history of orders delivered.

Scenario 4: Vicente adds his service to Engine and checks the history of orders as well as other information

Vicente heard about Engine through a business partner and decided to try it out. He sends an application to it, filling in the information requested. After a while, WellWhishes is now registered as a service in the Engine and Vicente can have a better look at statistics like orders delivered. VIcente can also see the amount of order's delivered on the current day.

2.4 Project epics and priorities

In the first iteration, besides deciding the service to construct, the main user stories to be developed were chosen and prioritized in order to organize the implementation of the project. Since the engine was a priority when comparing to the service, the user stories regarding the engine were prioritized and the frontend for those started to be developed.

With the help of GitHubs projects, functionalities were added throughout the process:

Iteration 1:

- Define the product concept
- Setup github repository, shared drive and github backlog

Iteration 2:

- Define the system's architecture
- Develop prototypes of the interface in the selected technologies
- Define user stories to be implemented
- Define the SQE strategy

Iteration 3:

- Implement the following user stories (by priority):
    - As a service I want to submit my application to Engine
    - As a rider I want to submit my application to Engine
    - As a service I want to login into Engine
    - As a rider I want to login into Engine
    - As a client I want to create an account in Book2Door
    - As a store owner I want to submit my application to Book2Door
    - As a client I want to search for books or stores in Book2Door
    - As the Engine admin I want to search for riders or services
- Implement the CI pipeline and merge requests policy
- Prepare QA manual and Specification report

Iteration 4:

- Implement the following user stories (by priority):
    - As the Engine admin I want to accept or deny riders and services applications
    - As the Engine admin I want to see a list of services and riders requests as well as already registered riders and services
    - As the Book2Door admin I want to accept or deny stores applications
    - As a store owner I want to able to add books to my catalog
    - As a client in Book2Door I want to place an order and get feedback on the order status
    - As a rider in Engine I want to be able to start my shift and end my shift
    - As a rider in Engine I want to see the current order I have to deliver

- Implement CD pipeline
- Prepare QA manual and Specification report

Iteration 5:

- Finish Specification report
- Enhancements on both systems
- Implement rider and store reviews in Book2Door

3 Domain model

The Engine is composed by a generic user, that is the parent of both the Rider and Contrib that represent, correspondingly, a rider on the Engine's workforce and a contributor service. This role is also reflected on the role attribute of the User model, where 0 should represent an admin, 1 represent a rider and 2 the contributor.

There is also an order, which has a Rider associated, which represents the pickup rider for that order, as well as Contrib, that represents the store that originates the order. Furthermore, each order must also have two Locations, one representative of the delivery location where the order should be delivered and another representative of the service location, where the order should be picked up from. Lastly, there is also an Enumerator OrderStatus, that describes the status of which the order is, those being waiting for being dispatched, assigned to a rider and waiting to be picked up, being delivered by a rider and that the order was delivered to the client.

The image below represents the domain model diagram, and was automatically generated by InteliJ's IDE.

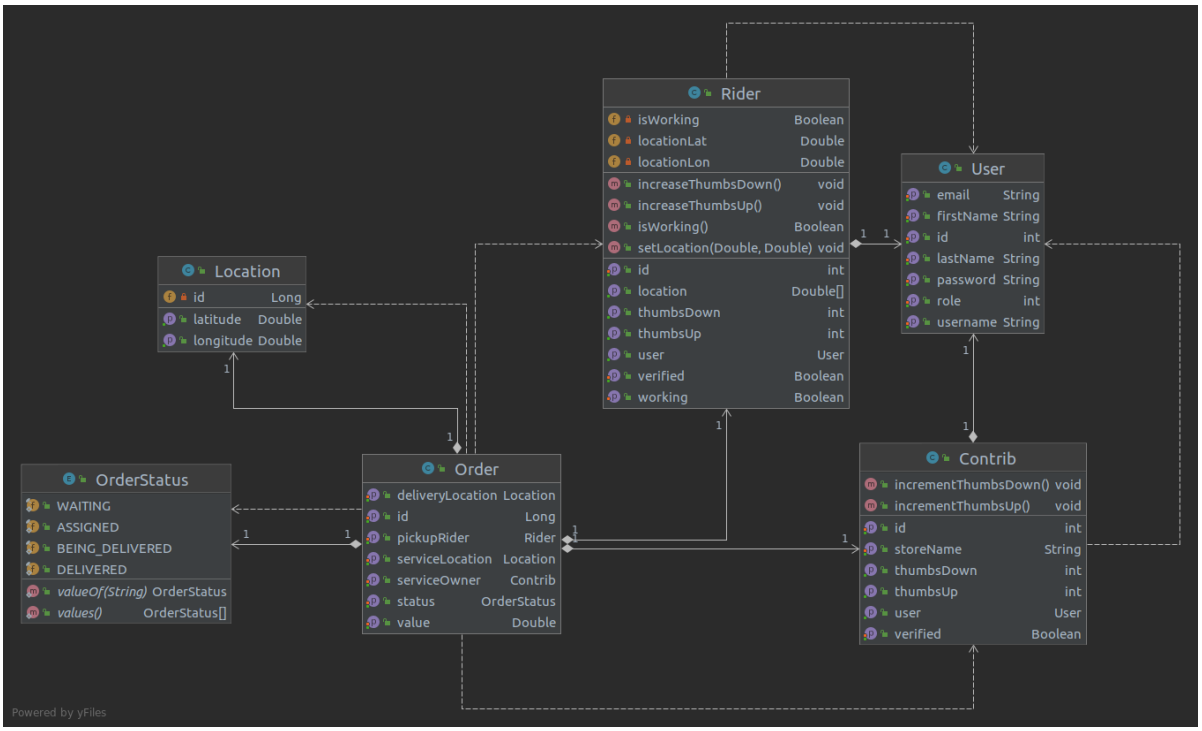

Figure 1: Engine domain model

The service is composed of six different entities: client, book, store, admin, BookOrder and JwtUser. The JwtUser is an entity used for authentication purposes, abstracting the type of user being logged in. The client uses the service in order to buy books and the store uses it to sell books. Meanwhile the

admin makes use of the platform to accept or deny stores applications to use the service. Finally the BookOrder is an entity to save the details of a purchase made by a client.

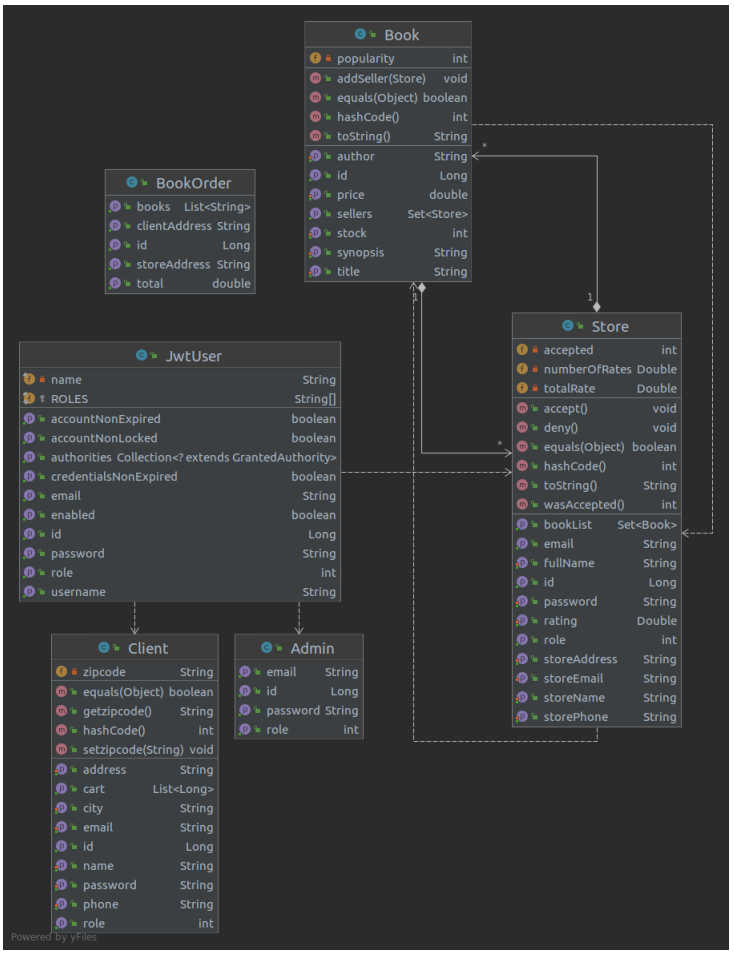

Figure 2: Book2Door domain model

4 Architecture notebook

4.1 Key requirements and constraints

- Both systems - Engine and Book2Door - must be accessible through PCs or other devices with internet connection;
- Both systems must be protected with authentication and authorization to ensure some level of security and each type of user only has access to its corresponding information;
- The engine should be capable of high performance to assure all orders get treated;
- Both systems should secure the user's data.

4.2 Architectural view

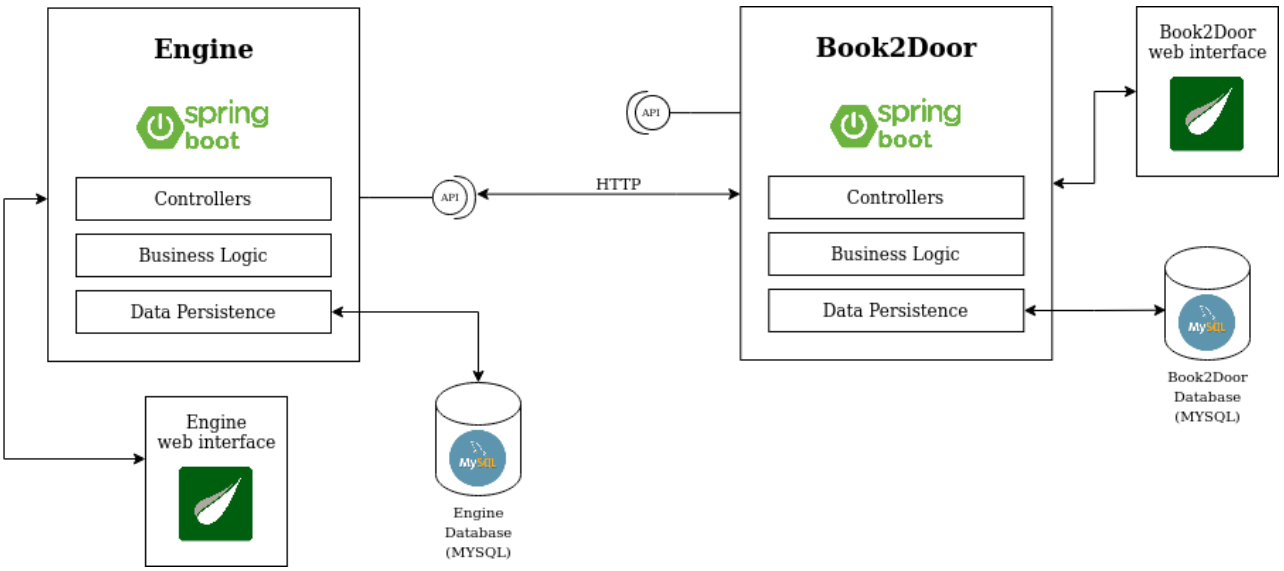

Figure 3: Architecture Model

The architecture, shown in the image x, details the 2 main components of the project: **Engine** and **Book2Door**. Both systems were developed with SpringBoot, using Thymeleaf for the frontend. Each system has its own database which was implemented with MySQL. Both Engine and Book2Door are 'divided' into three parts: the controllers, business logic and data access/ persistence. Book2Door connects to the Engine by sending data and fetching from it through the Engine's REST API, using http requests.

A Book2Door client places an order, by interacting with Book2Door web interface, which is then sent to the Engine's REST API where a rider receives this information and makes the corresponding delivery. The rider gives updates such as "Picked up" and "Delivered" which is fetched by the Book2Door web interface to show updates to the client.

4.3 Deployment architecture

The project was deployed on the Virtual Machine provided by the university that we had access to. To do so a Continuous Delivery Pipeline using Github Actions, in which a Self-Hosted Runner was configured, and it will execute the deployment whenever a commit is made to the main branch.

Thus, the project will be deployed after the pull request is made to the main branch and two team members review the pull request and verify the release.

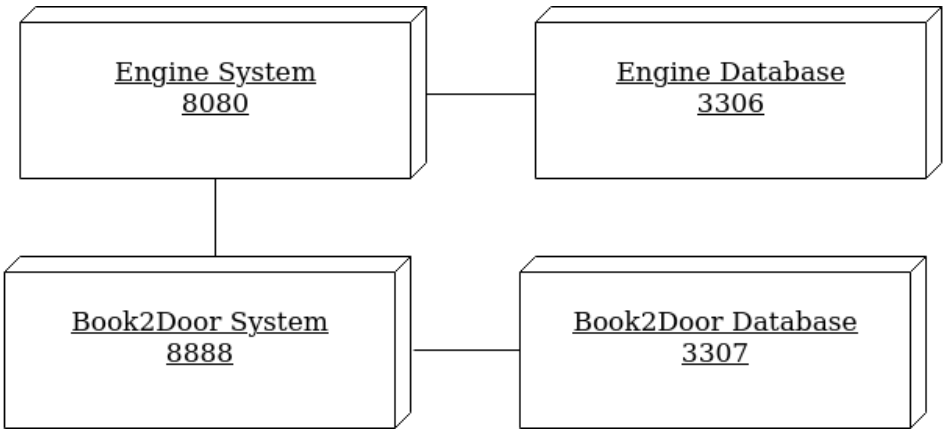

Figure 4: Deployment Stucture

5 API for developers

The following image shows the Engine's API regarding the Authentication endpoints and what each one of the endpoints perform.

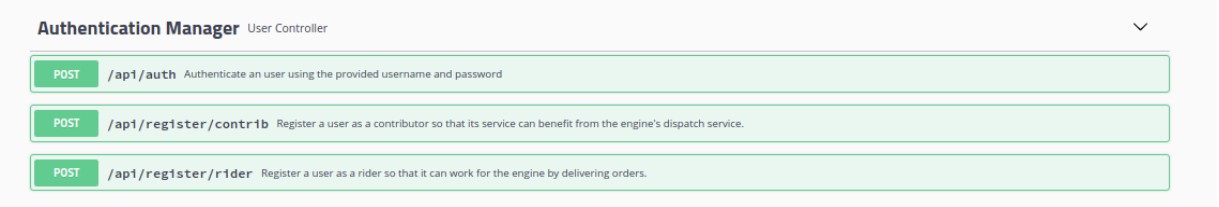

Figure 5: Swagger2 UserController's endpoint documentation

The following image shows the Engine's API regarding the endpoints related to Contributor's/Service's and what each one of them performs.

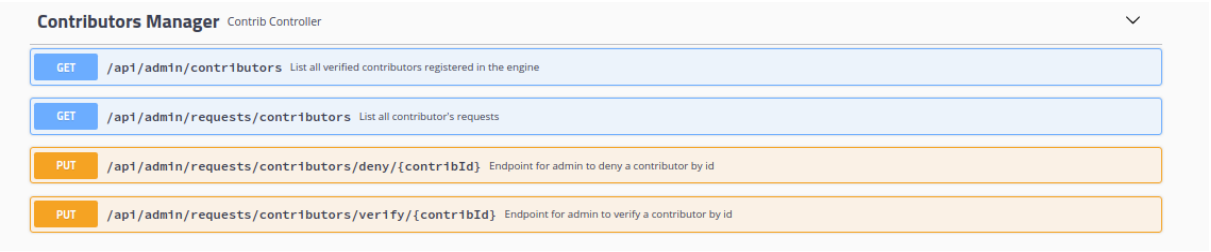

Figure 6: Swagger2 ContribController's endpoints documentation.

The following image shows the Engine's API regarding the endpoints related to Order's and what each one of them performs.

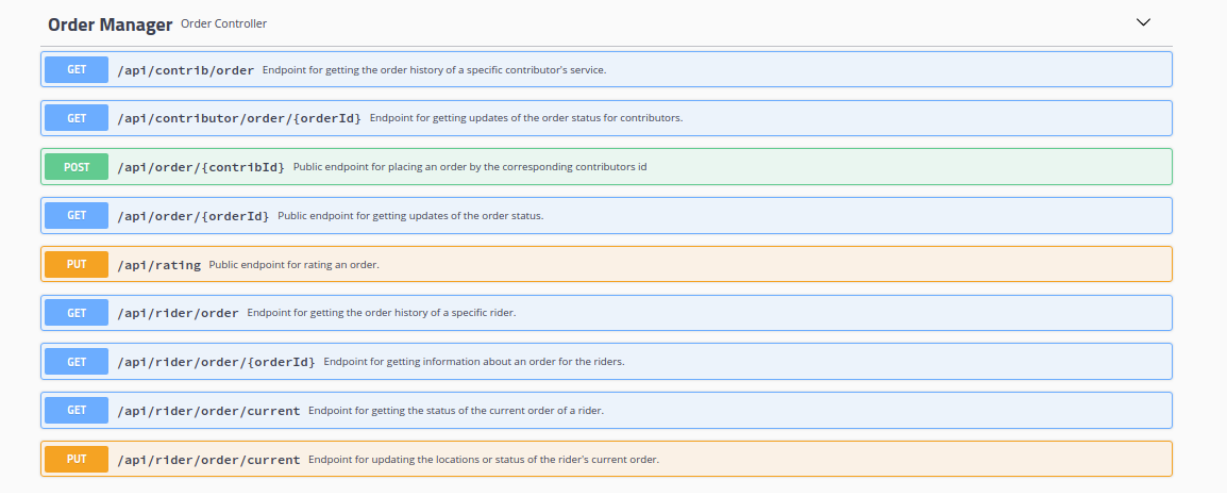

Figure 7: Swagger2 OrderController's endpoints documentation

The following image shows the Engine's API regarding the endpoints related to Rider's and what each one of them performs.

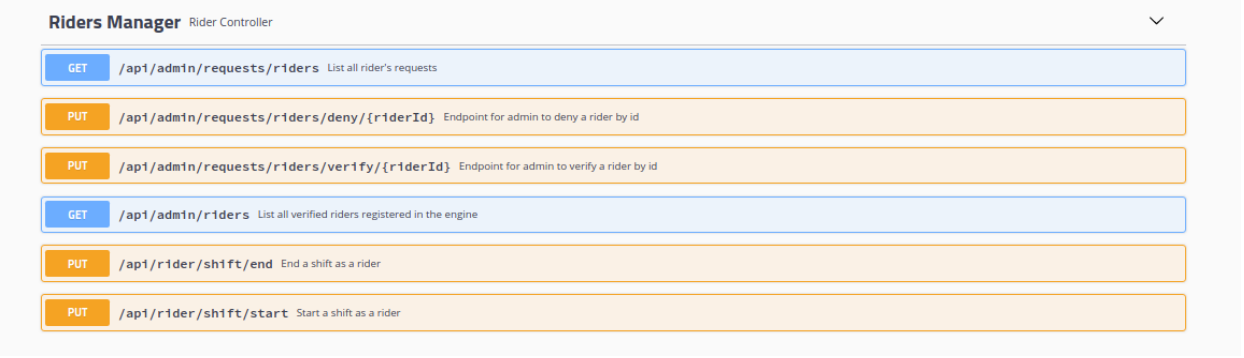

Figure 8: Swagger2 RiderController's endpoint documentation

All of the above documentation was generated using Swagger and it is available at /swagger-ui.html in addition to the detailed description of the endpoint's parameters and the models that the API utilizes.

6 References and resources

Project's Resources:

- Github Repository: https://github.com/davidgmorais/TQSProject
- Project's Backlog: https://github.com/davidgmorais/TQSProject/projects/1
- Engine API: deti-tqs-05.ua.pt:8080/api
- Engine QA Dashboard: https://sonarcloud.io/dashboard?id=engine
- Book2Door QA Dashboard: https://sonarcloud.io/dashboard?id=davidgmorais_TQSProject
- Engine Deployment: deti-tqs-05.ua.pt:8080
- Engine Documentation: deti-tqs-05.ua.pt:8080/swagger-ui.html