

### End of Studies

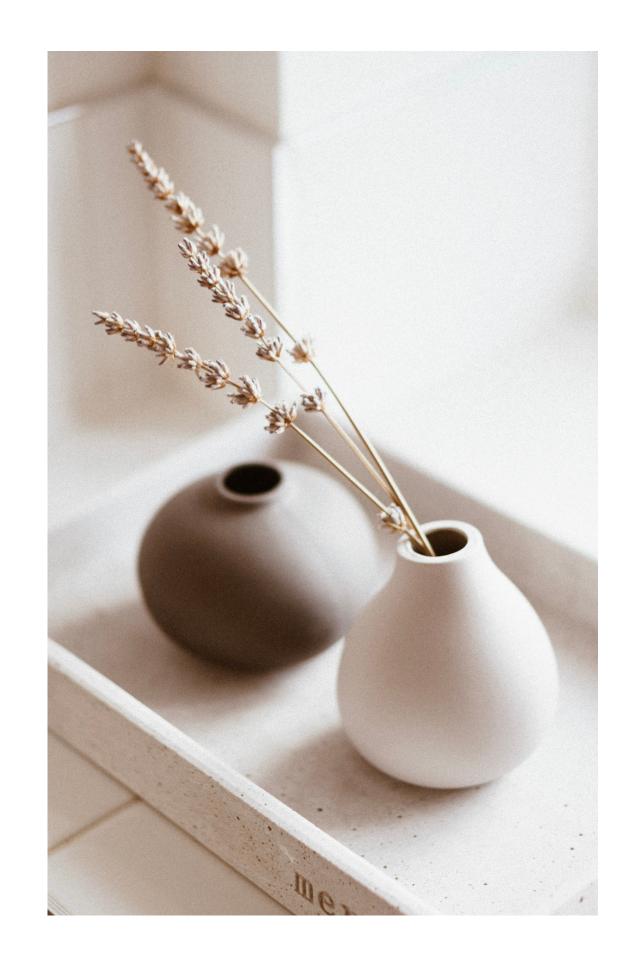
Design & Development of a Fullstack Web-Based platform for resource management to handle online appointments and reservations within different facilities

Selim HORRI



### TABLE OF CONTENTS

- O7 Introduction
- O2 Context model
- O3 Product Backlog
- 04 Design
- 05 Implementation Stack



#### Introduction



Most of service providers need to manage their resources and reservations within time slots in order to fulfill their customer needs.

In this context, we are planning to develop a resource management platform that can be used in various fields, including medical offices, cosmetic saloons etc...

This platform will be accessible via a Web application. (for now)



### Context model

#### **Actors**



Customer [Visitor]

••



Worker

• • •



Manager

•



**Owner** 





Dive into

# User Authorities & Usecases

#### Visitor service

Visitor is <u>authorized</u> to consult:

O1 Branches

O2 Facilities/Saloons



#### **Customer service**

After being authenticated, a customer should be <u>authorized</u> to manage :

O1 Favourites

O2 Reservations

O3 Ratings



#### Owner service

After being authenticated, an owner should be <u>authorized</u> to manage:



Facilities



#### Manager service

After being authenticated, a manager should be <u>authorized</u> to manage:

- O1 Categories
- O2 Service details
- O3 Workers
- O4 Appointments



#### Worker service

After being authenticated, a worker should be <u>authorized</u> to manage:

O1 Timetable

O2 Tasks/Todos



### Product Backlog

### Backlog & Releases

App Configurations

#### Sprint 1:

- Setup Backend & Frontend apps
   Sprint 2:
- Infrastructure as a Service (laaS)
- Configs (env vars, secrets...)
- CI / CD pipeline conf
- Deploy Poc(s) as Images

02

#### **Auth Security process**

#### Sprint 1:

Authentication & Authorization

#### Sprint 2:

• Registration

03

#### **Customer service**

#### Sprint 1:

- Consult Facilities & Locations
- Consult Topics & tags

#### Sprint 2:

Manage Favourites & Reservations

#### Sprint 3:

Rate a worker

04

#### Worker service

#### Sprint 1:

- Checkout timetable
- Manage Todos

05

#### Manager service

#### Sprint 1:

Manage facility categories & services

#### Sprint 2:

Manage Workers & Reservations

06

#### **Owner service**

#### Sprint 1:

Manage facilities

### Design

#### Use case:

https://github.com/SelimHorri/atic-backend-app/blob/master/docs/doc/cita\_usecase\_diag.drawio.pdf

#### **Entity Relationship (ERD):**

https://github.com/SelimHorri/atic-backend-app/blob/master/docs/doc/cita\_erd\_diag.drawio.pdf

#### **App Architecture**:

https://github.com/SelimHorri/atic-backend-app/blob/master/docs/doc/cita\_component\_complexity.drawio.pdf

### Implementation Stack

### Datastores







### Backend develop



### Frontend develop



# Code Quality Measurement



### Version Control

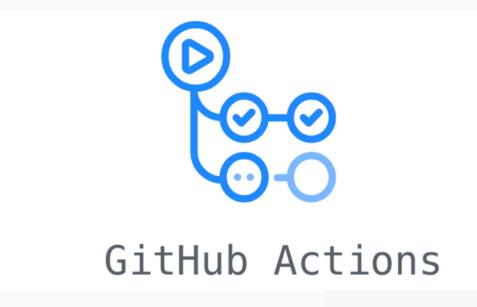




Azure DevOps



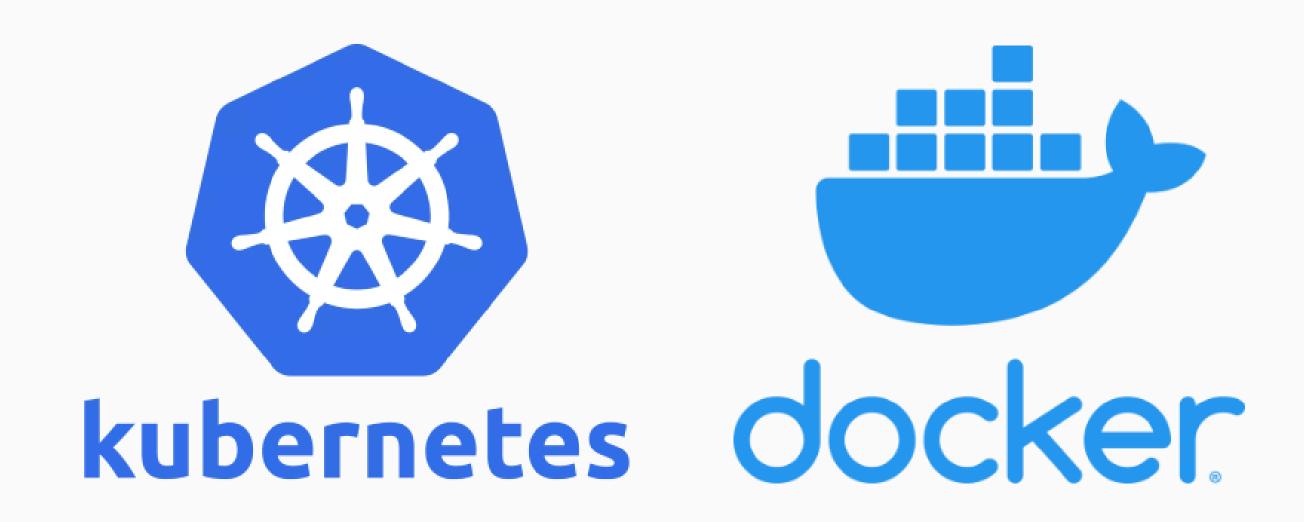
## CI/CD pipelines Continuous Integration - Continuous Delivery







### Container Runtime



### Cloud Provider





### Monitoring



#### 0000

# THANK YOU

•••





github.com/SelimHorri



hub.docker.com/u/selimhorri



selim.horri@ooredoo.tn



dev.azure.com/SelimHorri