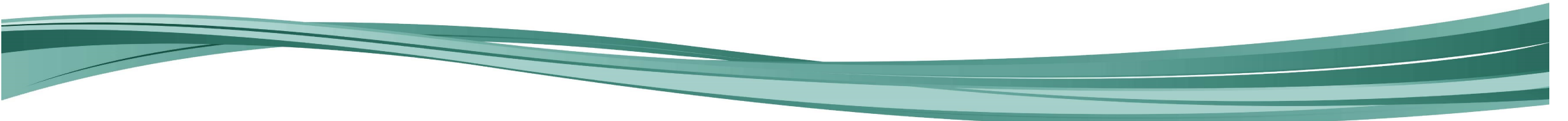




# Tasks for candidates

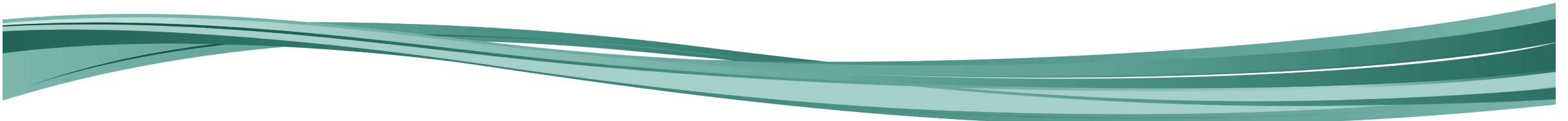
V 2.0





# Preferable technologies

- Java 8+, Gradle (preferable) or Maven
- Spring Boot, Spring Data JPA (use Postgres), Liquibase
- Proxy/API Gateway
- Message Brokers
- Swagger (detailed API description for resources)
- Unit tests written on Spock Framework (preferable) or JUnit
- Provide detailed deployment instructions for getting your solution up and running
- Create an opportunity to start services using **docker compose** or **minikube**
- Feel free to add any additional features and practices so your submission in order to demonstrate your expertise in writing production grade software



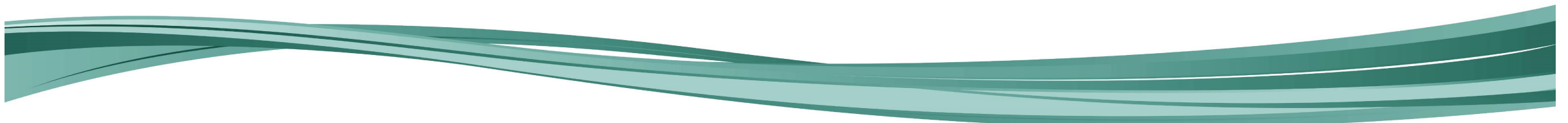


# Parcel Delivery APP

You'll be expected to build the back- end of the web app, while the front-end will be built by another developer in the team.

To do this, you are expected to implement a solution that enables the user stories provided. This solution must be implemented as a **micro-service architecture** that is deployable using **container technology**.

- Design a solution architecture diagram for Parcel Delivery APP based on micro-service technologies;
- Implement the micro-services required for Parcel Delivery APP to bring the user stories provided to reality;
- Containerize your micro-service solution to allow for it to be easily tested on other developer computers;



# User stories

User	Admin	Courier
Can create an user account and log in*;	Can change the status of a parcel delivery order;	Can log in;
Can create a parcel delivery order**;	Can view all parcel delivery orders;	Can view all parcel delivery orders that assigned to him;
Can change the destination of a parcel delivery order***;	Can assign parcel delivery order to courier;	Can change the status of a parcel delivery order;
Can cancel a parcel delivery order****;	Can log in and create a courier account*;	Can see the details of a delivery order;
Can see the details of a delivery;	Can track the delivery order by coordinates;	
Can see all parcel delivery orders that he/she created;	Can see list of couriers with their statuses;	

\* - with JWT;

\*\* - define fields yourself;

\*\*\* - define when user can change the destination yourself;

\*\*\*\* - define when user can cancel parcel delivery order yourself;

\* - define courier types yourself;

