Back-end developer test

Please upload the solutions to a public or private GIT repository when you finish the test. If you choose a private one, please send invitations for the following colleagues of ours:

- zoltan.gergo.cari@avinty.com
- david.kun@avinty.com
- balazs.fodor-pap@avinty.com
- kevin.kovacs@avinty.com
- peter.szilagyi@avinty.com
- adam.schlichter@avinty.com
- gyozo.zsok@avinty.com

Through working on the test, please register the time you spent on the tasks. Upload the report next to your solution.

You have one week, started with sending this mail, to send us the link to the GIT repository you used.

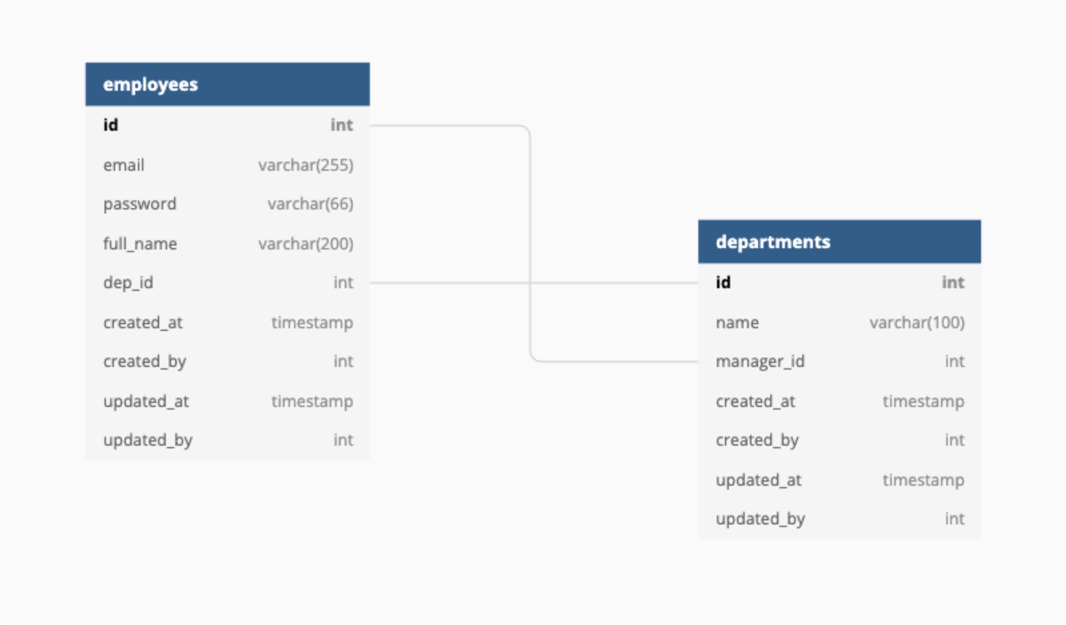
Please do not forget to send the mail to all our colleagues on the list above! If you have any questions about the tasks, feel free to contact persons from the same list!

Please use English through the work!

Create the following skeleton project, using Spring initializer:

- **Group:** com.avinty
- **Artifact:** hr
- **Name:** hr
- **Packaging:** Jar
- **Java version:** 11
- **Use the following dependencies:**
  - Spring Boot DevTools
  - Spring Web
  - Rest Repositories
  - Spring Data JPA
  - Lombok
  - JUNIT
  - Liquibase
  - Any other with reason
- **Use H2 in memory DB**

Create the HR schema for the application. Where it's needed create columns with NOT NULL.



employees	
id	**int**
email	varchar(255)
password	varchar(66)
full_name	varchar(200)
dep_id	int
created_at	timestamp
created_by	int
updated_at	timestamp
updated_by	int

departments	
id	**int**
name	varchar(100)
manager_id	int
created_at	timestamp
created_by	int
updated_at	timestamp
updated_by	int

The API has to have for the following list:

- **(GET) localhost:8085/API/V1/employees**
  - Gives back all of the records of the table.
- **(GET) localhost:8085/API/V1/dep-emp**
  - Gives back every department with the connected employees.
- **(GET) localhost:8085/API/V1/department?name=?**
  - Filter departments based on name. The data is used in a drop-down, so the whole entity is not required.
- **(POST) localhost:8085/API/V1/employees**
  - Creates an employee.
- **(DELETE) localhosts:8085/API/V1/department/:id**
  - Deletes a department based on ID. Set Null to every connected employee's DEP_ID field.

We are working with Rest API, so every communication must be done in JSON structure between the API and the client.

Bonus tasks for Juniors (**Mandatory for Mediors and Seniors**):

- Set the API to be able to handle calls from *localhost:5000* (CORS).
- **Extend** the existing EMPLOYEES table, using Liquibase, with a Boolean, not null column. The column name is: IS_ACTIVE.

Bonus tasks for Mediors (**Mandatory for Seniors)**:

- Implementing Custom Exception handling.
- Configure the Spring Security. Method level security and different ROLEs are needed (ROLE_ADMIN, ROLE_USER) Use JWT communication between the API and the client, because the API is stateless. For this task, use the EMPLOYEES table.