

# JAVA PROJECT

## Context

The aim of this mini project is to contribute to the development of a goods delivery application. The application is intended for customers, schedulers, and drivers.

## Description of the project

When the application is launched, a window appears asking the user to either register or authenticate.

The registration window allows the user to enter his/her e-mail address, password, cell phone number, and role (customer, scheduler, or driver). If the user is a driver, he/she must also enter his truck's registration number and capacity of the truck in Kg.

Once authenticated, the user can modify the following information: e-mail address, password, cell phone number and, if applicable, truck registration number and capacity truck.

### **As a customer,**

The user can select from a list of products (hard stored in the database), the goods he/she needs and the quantity in Kg. The user then chooses the delivery date and address.

### **As a scheduler,**

The user has access to the list of all deliveries to be made sorted by date. The user can assign a route to a driver (a route is a set of destination points). The user can generate a Word file with all assignments for a specific day (date chosen by the user). The generated document must contain justified title "[THE CHOSEN DAY]" in a bold text. The body of the document contains the list of missions. Each mission is made up of the driver's first and last name, and the route. A mission starts with the warehouse address, followed by the delivery addresses, and ends with the warehouse address.

*Optional question:* Use the Google OR-Tools library to plan daily journeys automatically.

### **As a driver,**

The user can view the list of missions assigned to him/her.

The user can view the list of missions he/she has already carried out.

## Constraints and instructions

The database management system used should be MySQL.

The programming language used must be Java.

At the end of each session, you **must** push your code to GitHub.

# Documentations

Use menu with Java:

<https://docs.oracle.com/javase/tutorial/uiswing/components/menu.html>

Generate word file with Java:

<https://poi.apache.org/>

[http://www.tutorialspoint.com/apache\\_poi\\_word/](http://www.tutorialspoint.com/apache_poi_word/)

Use the **Google OR-Tools** with Eclipse:

Download: [https://github.com/or-tools/java\\_or-tools/archive/v9.5.zip](https://github.com/or-tools/java_or-tools/archive/v9.5.zip)

In Eclipse, select File > Import > Maven > Existing Maven Projects then click on Next button.

Select in the Root Directory the folder where you saved the downloaded file, then click on Finish button.

In Eclipse default view, select the created project and right click on it and choose Run As > Maven Install.

For more details: [https://developers.google.com/optimization/install/java/pkg\\_windows?hl=en](https://developers.google.com/optimization/install/java/pkg_windows?hl=en)

JAVA - How To Move JTable Selected Row UP and DOWN:

[https://www.youtube.com/watch?v=-2\\_iFI0RQvY](https://www.youtube.com/watch?v=-2_iFI0RQvY)

Use GitHub:

To create a new repository on GitHub and clone it using Eclipse, follow these steps:

1. **GitHub Repository Creation:**
  - a. Go to [GitHub](#) and sign in or create a new account if you don't have one.
  - b. Once logged in, click on the "+" sign in the top right corner and select "New repository."
  - c. Fill in the necessary information for your new repository, including the repository name, description, and other settings.
  - d. Optionally, initialize this repository with a README, add a .gitignore, or choose a license.
  - e. Click on the "Create repository" button.
2. **Cloning the Repository in Eclipse:**
  - a. Open Eclipse and go to "Window > Show View > Other..."
  - b. In the "Show View" dialog, select "Git > Git Repositories" and click on the "Open" button.
  - c. In the "Git Repositories" tab, click on "Clone a Git repository."
  - d. Choose "Clone URI" and click "Next."
  - e. Back on GitHub, copy the repository URL by clicking on the "Code" button and selecting the HTTPS URL.
  - f. In Eclipse, paste the copied URI into the "URI" field.
  - g. Provide your GitHub credentials if prompted.
  - h. Choose the desired branch to clone (usually "main" or "master").
  - i. Choose a local directory for your repository.
  - j. Click "Next" and then "Finish."

Now, you have successfully created a new repository on GitHub and cloned it using Eclipse. You can start working on your project locally and push changes to GitHub **at the end of each session**.

Please note that the steps might vary slightly based on the versions of GitHub and Eclipse you are using. Always refer to the official documentation for the most accurate and up-to-date instructions.