Use Case Description

Louis LABORY, Jason LAVAL, william MICHAUD, Lou REINA--KUNTZIGER, Andy GONZALEZ, Elise BACHET

IF-4, hexanome 3

Contents

1. Introduction

This document describes all the Use Cases present in the Use Case Diagram available on our GitHub repository.

2. Use Case Descriptions

2.1. Load and display City Map (XML)

- **Primary actor:** Dispatcher
- **Goal:** Import the city map from an XML file and display it.
- **Trigger:** The Dispatcher selects a map file (XML).
- **Preconditions:**
  - The XML file is accessible (valid path).
  - The file matches the expected format.
- **Postconditions:**
  - The contents of the XML file are parsed and loaded into memory (nodes, segments).
  - The map is displayed in the graphical interface.

2.2. Load Delivery Request (XML)

- **Primary actor:** Dispatcher
- **Goal:** Import delivery requests from an XML file and complete the required attributes.
- **Trigger:** The Dispatcher selects a requests XML file.
- **Preconditions:**
  - The map is loaded.
  - The requests XML file is accessible and valid.
- **Postconditions:**
  - The requests are created in the system.
  - For each request: a courier is assigned, pickup/delivery locations are mapped to nodes of the map, and service times are defined.

2.3. Select Courier

- **Primary actor:** Dispatcher
- **Goal:** Assign a request to a courier.
- **Trigger:** The Dispatcher selects a courier for the request (or an automatic assignment rule is applied).
- **Preconditions:**
  - At least one courier exists in the system.
- **Postconditions:**
  - The request is linked to a courier.

2.4. Select Pickup and Delivery Location

- **Primary actor:** Dispatcher
- **Goal:** Set the locations for collecting and delivering the parcel (only in the case of a manual addition).
- **Trigger:** The Dispatcher chooses/validates the pickup and delivery locations for the request.
- **Preconditions:**
  - The map is loaded.
  - A delivery request is being edited.
- **Postconditions:**
  - The nodes for picking up and delivering the parcel reference valid intersections of the graph.

2.5. Set Service Times

- **Primary actor:** Dispatcher
- **Goal:** Define pickup and delivery service durations.
- **Trigger:** The Dispatcher enters/validates the pickup and delivery durations for the request.
- **Preconditions:**
  - The target request is identified.
- **Postconditions:**
  - Pickup and delivery durations are recorded with values greater than or equal to 0.

2.6. Set Number of Couriers

- **Primary actor:** Dispatcher
- **Goal:** Define the total number of couriers employed.
- **Trigger:** The Dispatcher changes the value N of the number of couriers using the appropriate buttons.
- **Preconditions:**
  - The system is initialized.
- **Postconditions:**
  - The system has $N \geq 1$ operational couriers (created/reinitialized if necessary).
  - Reducing N may invalidate existing assignments (to be indicated in the interface).

2.7. Compute Best Tour

- **Primary actor:** Dispatcher
- **Goal:** Compute optimal tours that satisfy the constraints.
- **Trigger:** The Dispatcher starts tour computation.
- **Preconditions:**
  - The map is loaded.
  - At least one request is available.
  - At least one courier is available.
- **Postconditions:**
  - For each courier, an ordered tour is produced, starting at the depot at 08:00, respecting the order *pickup then delivery*, with estimated times and distances.
  - The following validity criterion is satisfied: the return time does not exceed the latest allowed time.

2.8. Display Tour on Map

- **Primary actors:** Courier and Dispatcher
- **Goal:** Visualize the tour on the map with operational details.
- **Trigger:**

  - A tour has been computed by the Dispatcher or a tour is restored.
  - The courier selects their tour to visualize the route to be completed.
- **Preconditions:**
  - Tours have been computed or restored.
- **Postconditions:**
  - The selected courier's tour is displayed (polyline, stops, schedules, service durations).

2.9. Save Current Tours to File

- **Primary actor:** Dispatcher
- **Goal:** Save the current state of the tours to a file.
- **Trigger:** The Dispatcher chooses the "Save" action and selects a file location.
- **Preconditions:**
  - Tours exist.
- **Postconditions:**
  - A file is created at the chosen path.
  - The file contains consistency metadata, the number of couriers, and the tours (visit order, schedules).
  - A success confirmation is displayed.

2.10. Restore Tours from File

- **Primary actor:** Dispatcher
- **Goal:** Restore a previously saved state.
- **Trigger:** The Dispatcher selects a backup file to restore.
- **Preconditions:**
  - The backup file is accessible.
  - The map is loaded and consistent.
- **Postconditions:**
  - Couriers, requests, and tours are reloaded and displayed.

2.11. Add a delivery request manually

- **Primary actor:** Dispatcher
- **Goal:** Manually create a new delivery request and enter all required information.
- **Trigger:** The Dispatcher selects the "Add a delivery request manually" action in the interface.
- **Preconditions:**
  - The map is loaded.
  - At least one courier exists in the system.
- **Postconditions:**
  - A new delivery request is created in the system.
  - A courier is assigned to the request.
  - The pickup and delivery locations are selected and mapped to valid nodes of the map.
  - Pickup and delivery service durations are defined.
  - The request is ready for route computation.