Delivelo - Specifications

Groupe 4114
-
Annie Abhay
Brandon Da Silva Alves
Sébastien Goll
Louis Hasenfratz
Sophanna Ngov
Jade Prévôt

August 30, 2023

Contents

List of Figures

Chapter 1

Introduction

Delivelo is an application for optimising delivery tours in cities, done with bicycles. It allows loading a city map, loading the addresses of pickup and delivery, and establishing a ride of delivery.

Chapter 2

Glossary

City map : a map composed by lists of intersections and roads.

Client : a person who collect or send a package.

Coordinate : a position represented by a latitude and a longitude.

Delivery address : an intersection where the user must drop off a package.

Depot address : an intersection where starts the delivery man.

Distribution : a set of requests.

Duration : the time in seconds.

Intersection : a coordinate in the map labeled by a number (id).

Map view : a part of the UI that displays graphical information about the Tour, the City map and the Distribution.

Path : an ordered list of roads connecting two points of interest.

Pickup address : an intersection where the user must pickup a package.

Point of interest : a pickup address or delivery address or depot address.

Request : a request is composed by a pickup address, the duration of the pickup, a delivery address, the duration of the delivery.

Road : a segment defined by an origin intersection, a destination intersection, a label and a length in meters.

Roadmap : an ordered list of instructions that describes the tour.

Roadmap view : a part of the UI that displays information about the roadmap.

Tour : an ordered list of points of interest and the paths connecting them. A tour begins and ends with the depot address.

Tour duration : the total traveling time which is the duration of the trip to visit every points of interest of the tour plus the duration of every pickup and delivery addresses.

Travel time : duration of travel between two points of interest.

User : the person who uses the application.

Chapter 3

Use Cases

We listed 5 main functionalities :

- Load a city map;
- Load a distribution;
- Compute a tour;
- Modify the tour;
- Validate the roadmap.

Those use cases and their interactions with the actor are represented in a use case diagram figure 3.1.

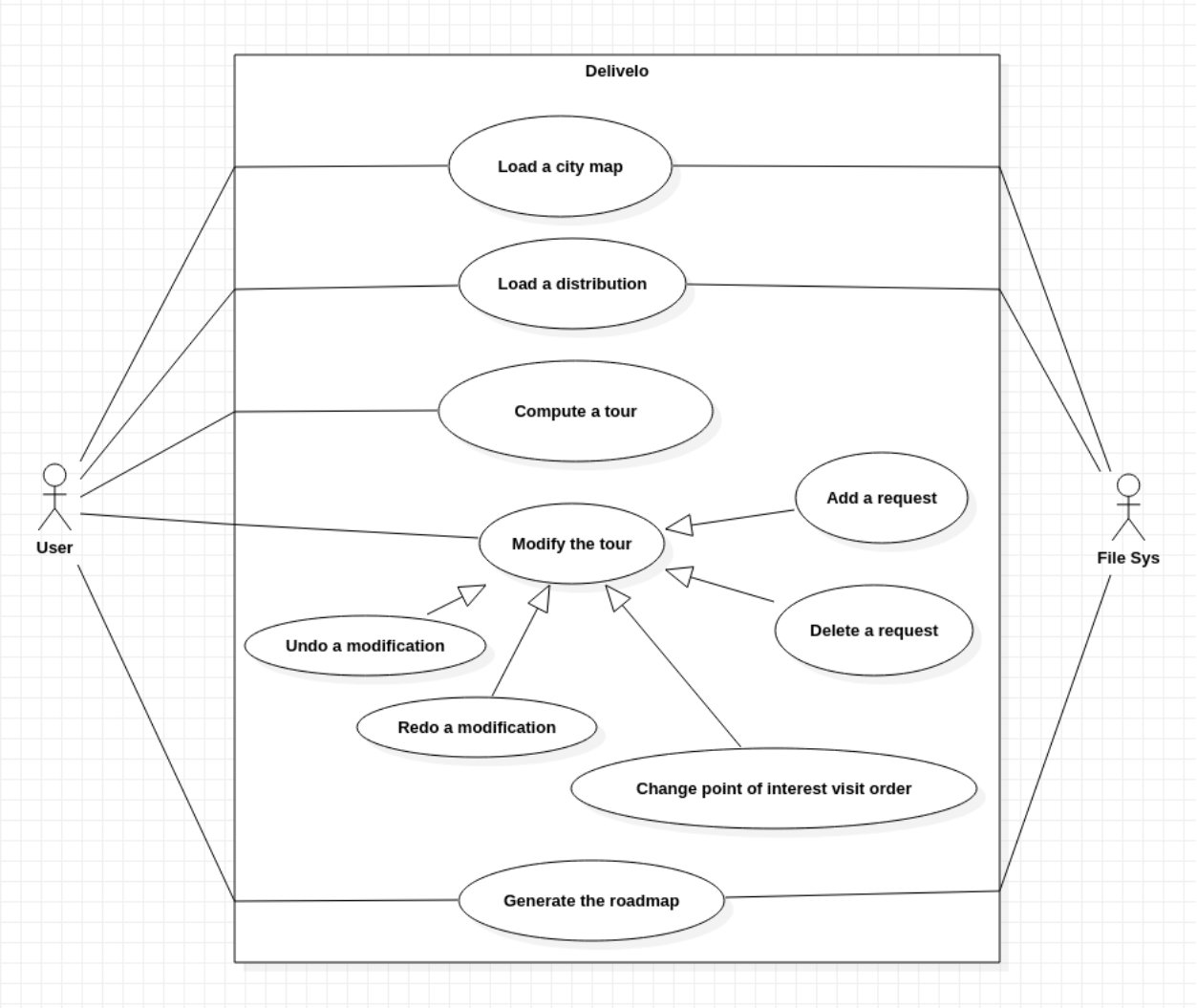

Figure 3.1: Use Case Diagram

A description of each of them is given below.

3.1 Load a city map

Brief format

The user clicks on the "Load city map" button. The user selects the city map file and confirms this choice. The system then displays the corresponding city map.

Structured description

Precondition

None

Nominal scenario

1. The user tells the system to load a city map.
2. The system asks to choose the city map file.
3. The user selects the XML city map file and confirms this choice.
4. The system opens the file, parses it and displays the corresponding city map.

Extensions

4a. The file contains unusable information (wrong XML structure)

- The system notifies the user that the file is unusable and goes back to step 2.

2-4a. The user tells the system he wants to cancel the action

- The system cancels the action.

3.2 Load a distribution

Brief format

The user clicks on the "Load a distribution" button. He then selects the distribution file and confirms this choice. The system then displays the corresponding points of interest on the map.

Structured description

Precondition

A map is loaded.

Nominal scenario

1. The user tells the system to load a distribution.
2. The system asks to choose the distribution file.
3. The user selects the XML distribution file and confirm this choice.
4. The system opens the file, parses it and display the points of interest on the map.

Extensions

4a. The file contains unusable information (wrong XML structure/blank file)

- The system notifies the user that the file is unusable and goes back to step 2.

4b. The data contained in the file does not correspond to the loaded map

- The system notifies the user that the file cannot be used and goes back to step 2.

2-4a. The user tells the system he wants to cancel the action

- The system cancels the action.

3.3 Compute a tour

Brief format

The user clicks on "Compute the tour". The system computes the tour and displays it on the map.

Structured description

Precondition

A distribution and a map are loaded.

Nominal scenario

1. The user tells the system to compute the tour.
2. The system computes the tour and displays it on the city map and displays the roadmap next to it.

Extensions

2a. The computation time exceeds the time limit

- The system displays the best tour found so far and notifies the user that this path is not optimal.

3.4 Modify the tour

Brief format

The user tells the system that he wants to modify the tour and clicks a first time on the map to add a pickup point then clicks a second time on the map to define the delivery point. The system modifies the tour accordingly.

Structured description

Precondition

A tour has been computed.

Nominal scenario

1. The user tells the system that he wants to modify the tour.
2. The system enters the modification state.
3. The user clicks on a road on the map to add a pickup point.
4. The system displays the pickup point and focuses on the roadmap view.
5. The user indicates between which step to insert the pickup point. (option only possible between the to depot points)
6. The system displays tour modification and focuses on the map view.
7. The user clicks on a road on the map to add a delivery point.
8. The system displays the delivery point and focuses on the roadmap view.

9. The user indicates between which step to insert the delivery point. (option only possible between the to pickup point and the last depot point)
10. The system updates the tour, displays it and returns in multi-purpose modification state (same as step 2).

Extensions

1-2a. The user tells the system to cancel the modifications

- The system reloads the tour before modification and exits modification state.

3-10a. The user tells the system to cancel the step addition

- The system removes the points already placed and stays in modification state.

3.5 Validate the roadmap

Brief format

The user tells the system that he wants to generate the tour roadmap. The system displays the tour roadmap.

Precondition

A tour has been computed.

Nominal scenario

1. The user tells the system to validate the tour roadmap.
2. The system generates the roadmap file that can be given to the delivery man.

Chapter 4

Domain model

During the design phase, we established a domain model, shown in figure 4.1, representing the relevant concepts to be modeled in our application.

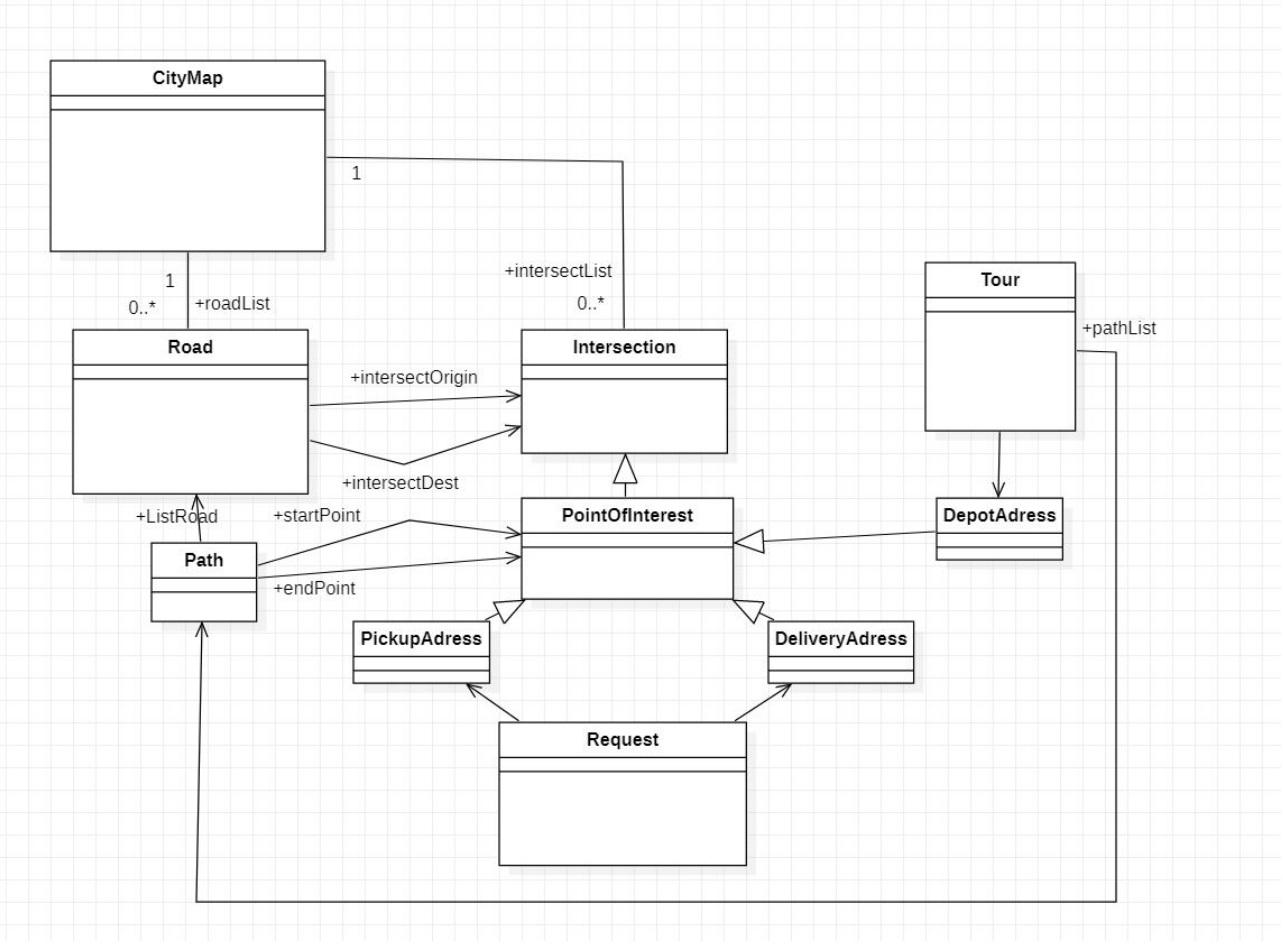

Figure 4.1: Domain Model

Chapter 5

Class and package diagrams

In figure 5.1 we can see a package diagram. In the other figures there is a class diagram of all packages :

- Model : figure 5.3;
- View : figure 5.4;
- Controller : figure 5.2;
- Delivelo : figure 5.4;
- TSP : figure 5.5;
- XML : figure 5.5;
- Observer : figure 5.4.

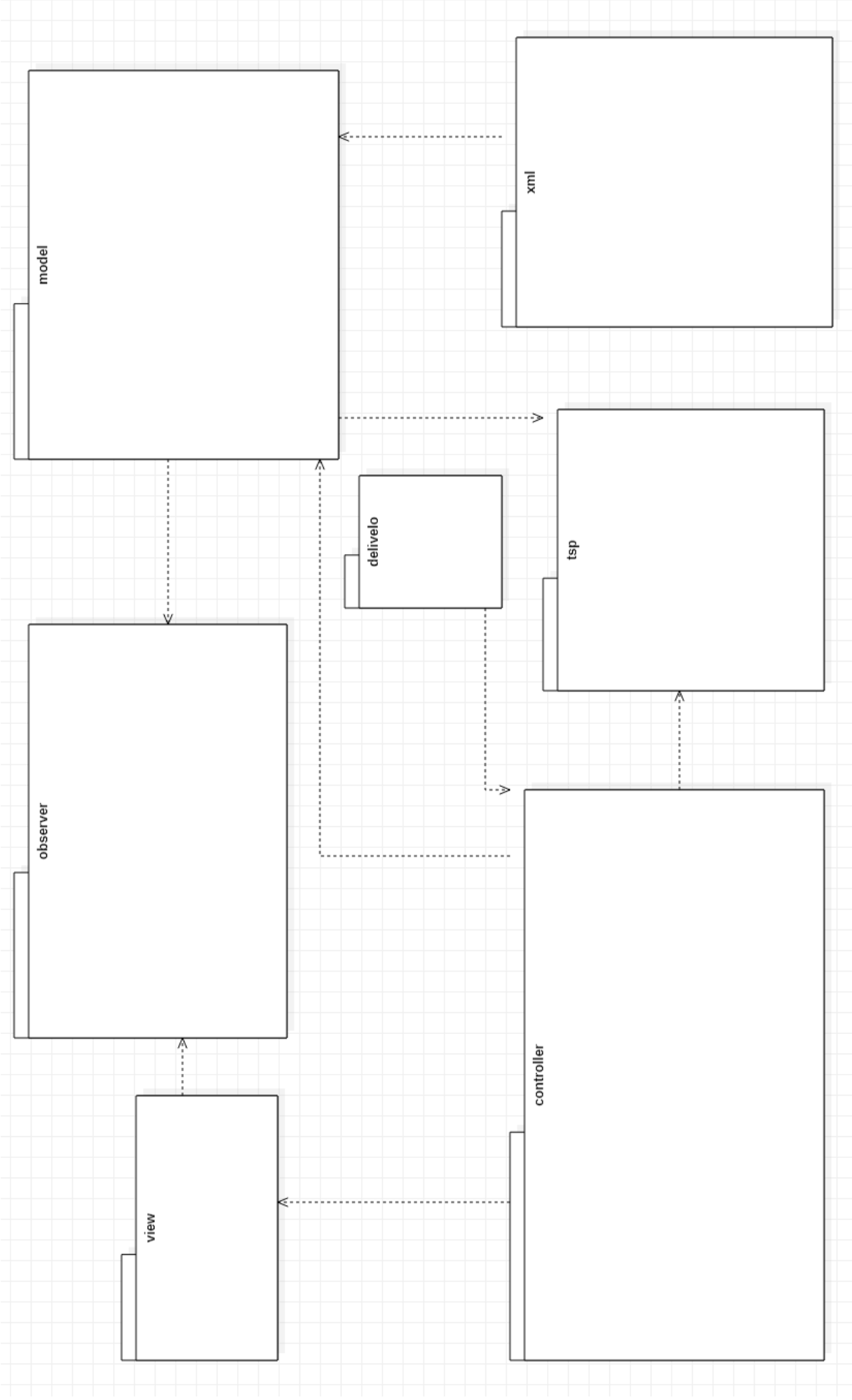

Figure 5.1: Package diagram

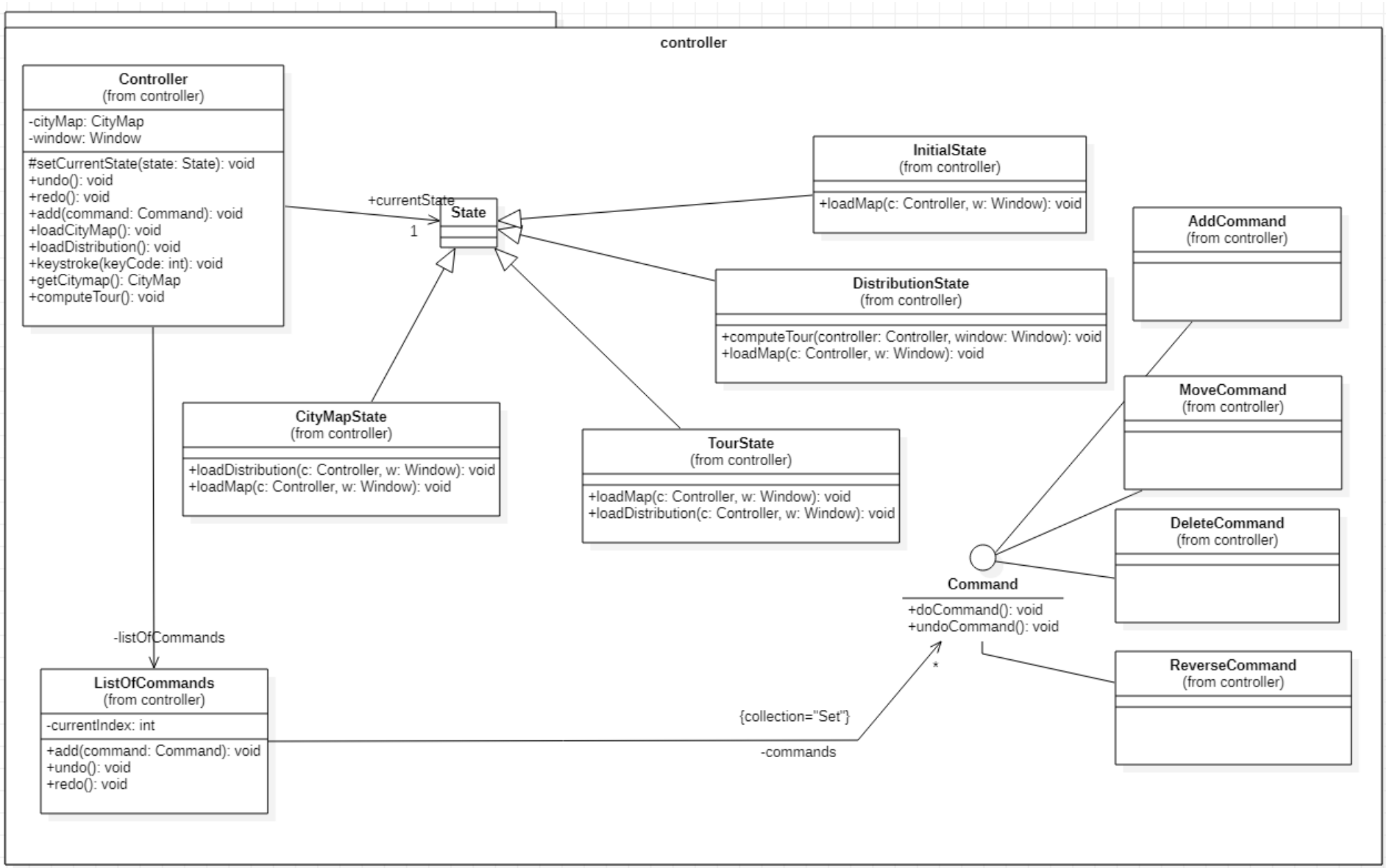

Figure 5.2: Class diagram of Controller package

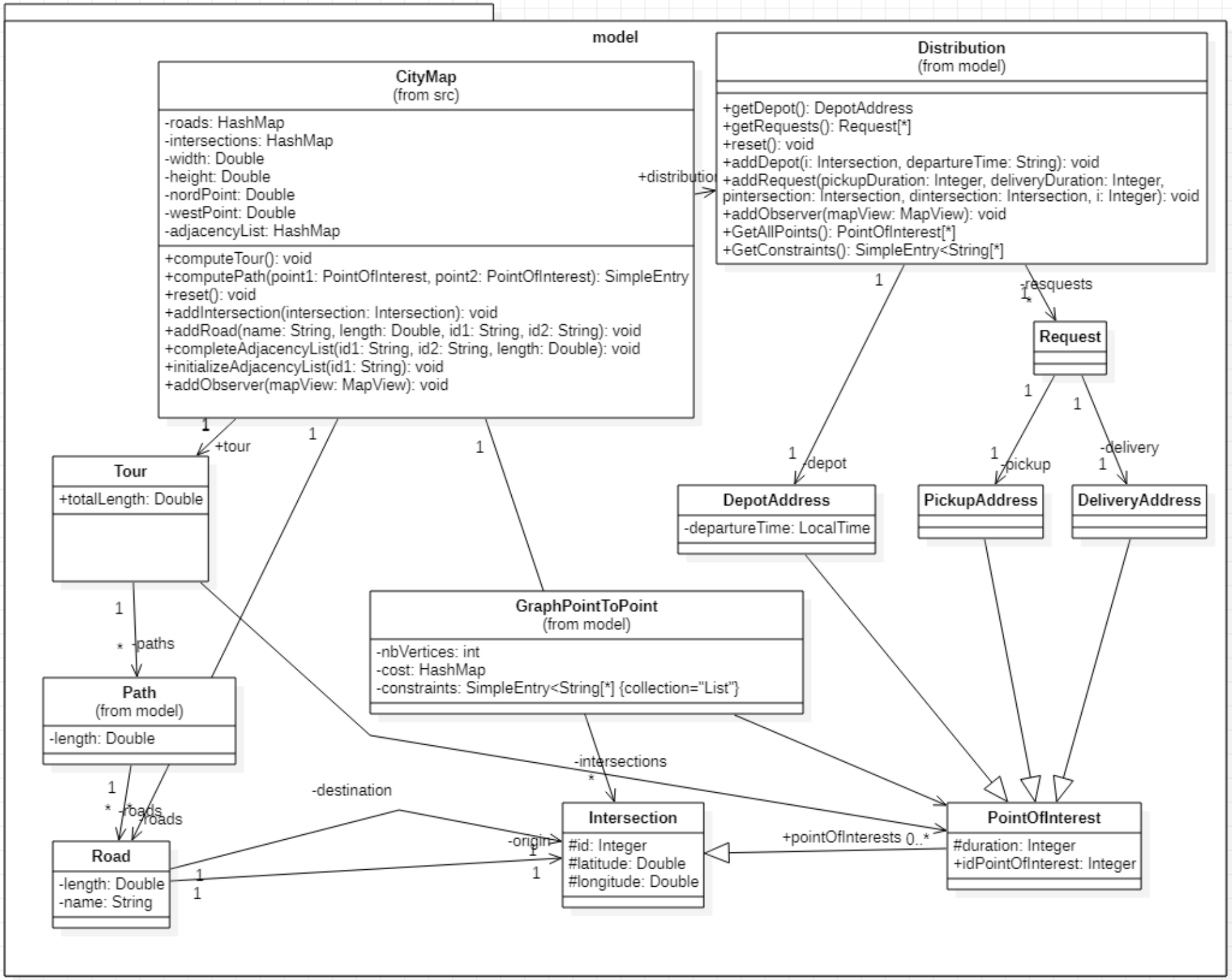

Figure 5.3: Class diagram of Model package

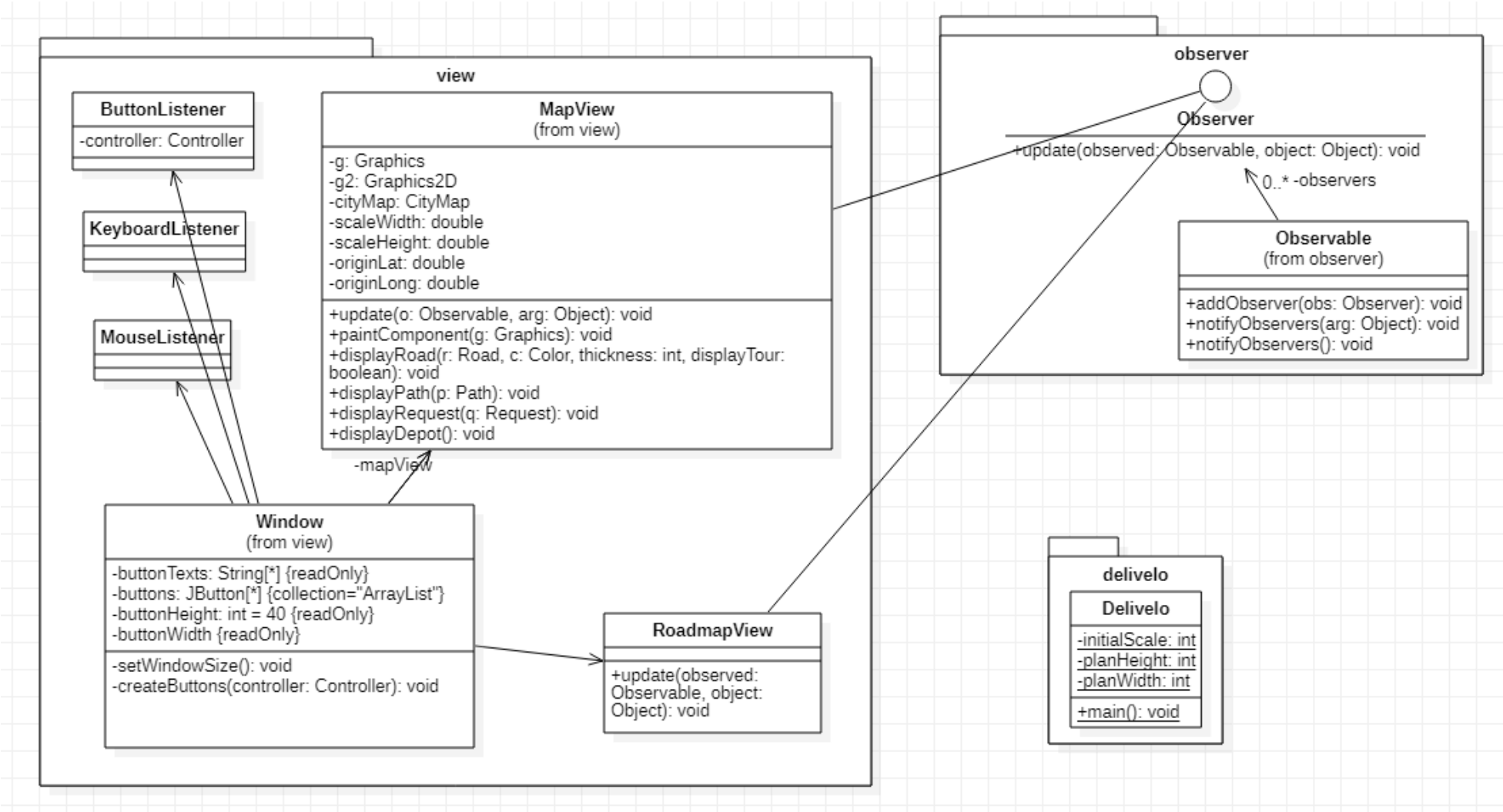

Figure 5.4: Class diagram of View, Observe and Delivelo packages

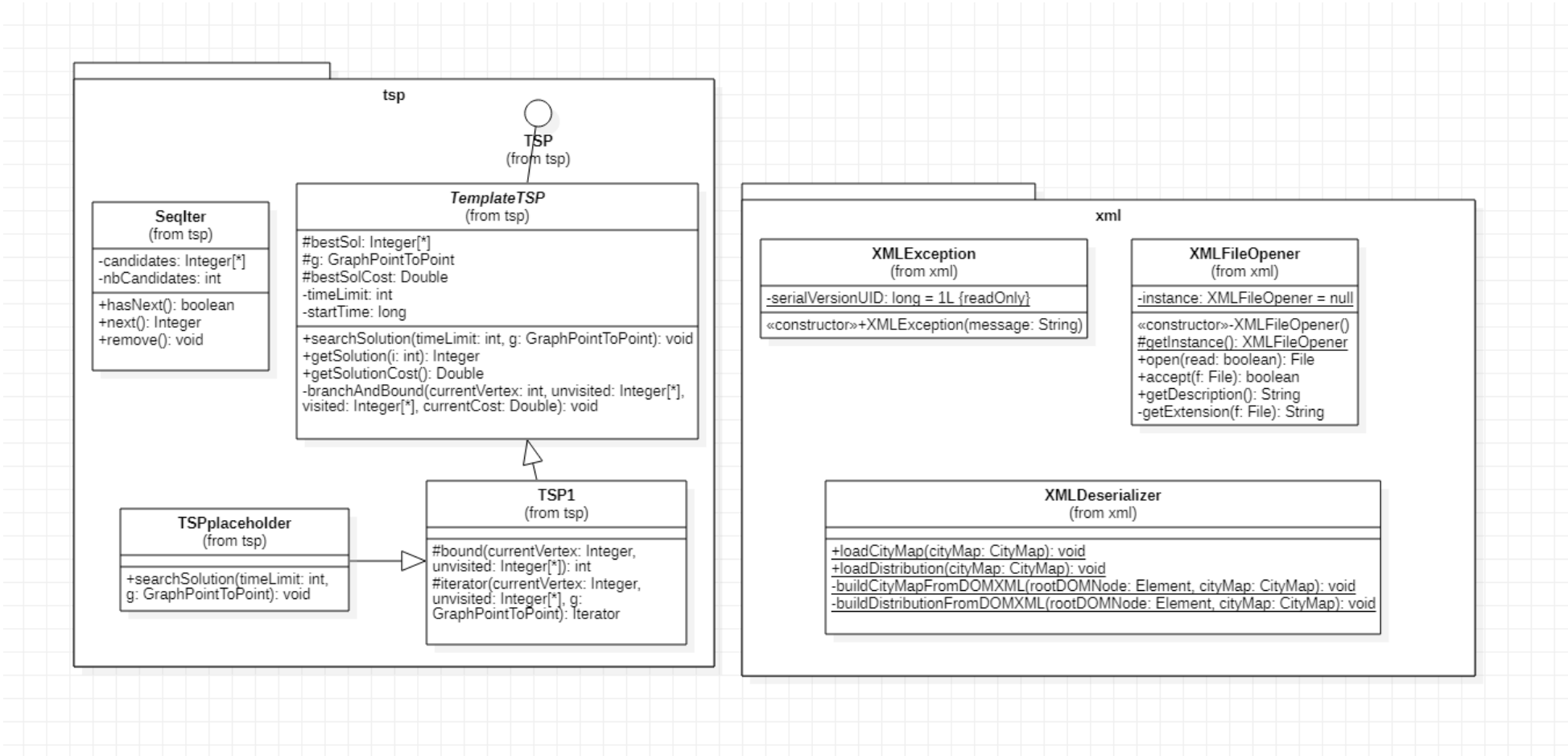

Figure 5.5: Class diagram of XML and TSP packages

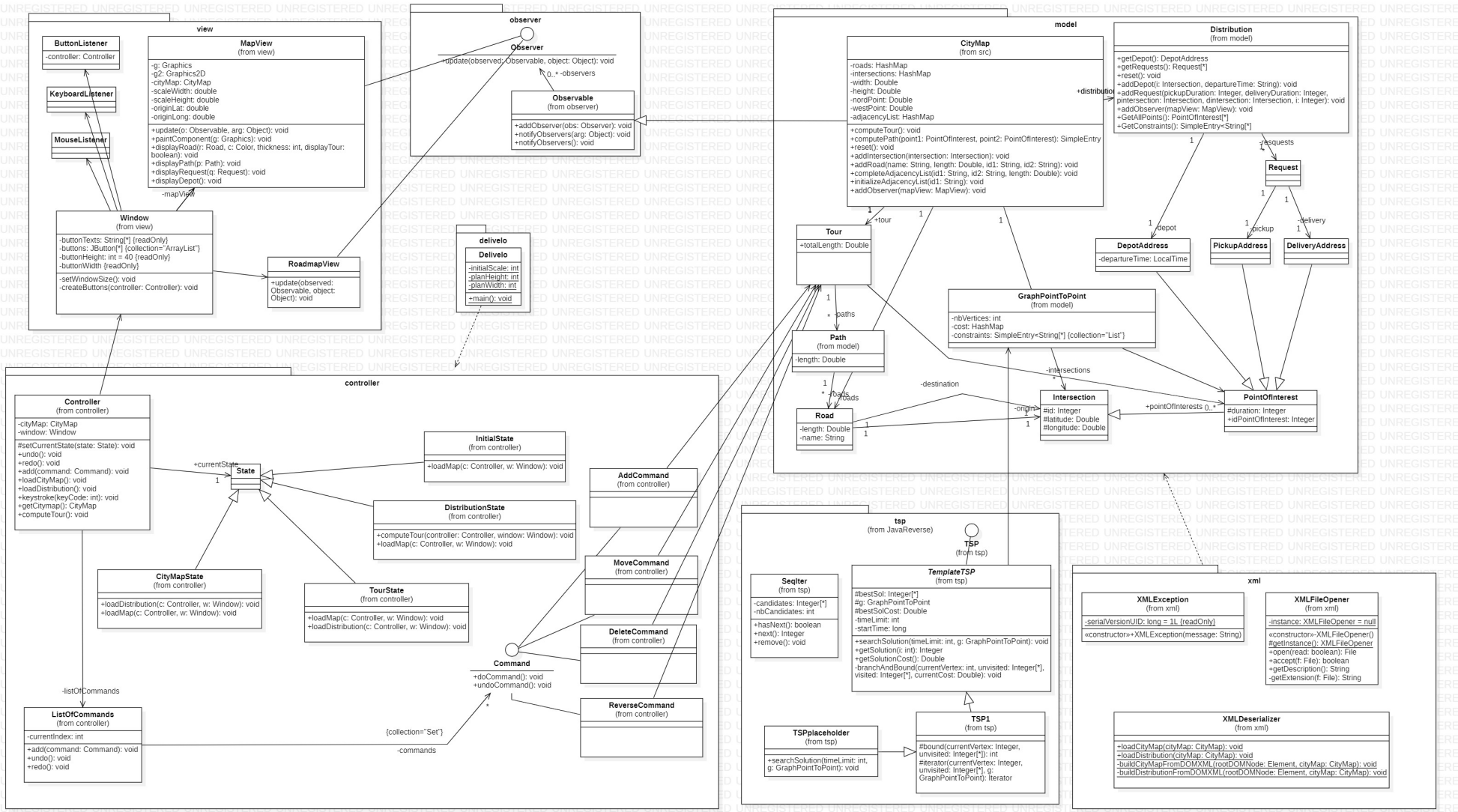

Figure 5.6: Class diagram

Chapter 6

Provisional Schedule

During the first project phase, we were organized according to the following schedule, on figure 6.1.

PROVISIONAL SCHEDULE	Workload (h)					
Tasks	Annie	Brandon	Jade	Louis	Sebastien	Sophanna
Glossary		1.5	1.5			
Use-Case diagram				1.5	1.5	
Meetings with client	2					1
Domain diagram	2					2
Class diagram & updates	1.75	2				
Planning & organisational documents	4	0.5	4.5	1	0.5	0.5
Brief specifications				1	1	
Detailed specifications				2	2	
Architecture & Design Pattern choice	2	1.5	1.5	1.5	1.5	2
Project Shell					1	
Hand test design					1	
Summary of naming convention						0.8
Appropriation of Java FX vs Swing						2
Stand up team meetings	1	1	1	1	1	1
Design Pattern State		1				
State-Transition diagram		2.5	2.5			
Appropriation of XML parsing	1			1		
Reading sequence diagram	0.75			0.75		
Software configuration	0.5	0.5	0.5	0.5	0.5	0.5
Creation of test XML (city map)					0.3	
Parsing LOAD MAP implementation	2.75			2.75		
Parsing LOAD LIST implementation				2		
Compute tour sequence diagram		1.5				
IHM thinking			2			2
IHM elaboration			1.5			
Creation of the main window on Swing						2
Creation of JUnit files (city map)					2	
Design & Creation of XML files					0.5	
Controller implementation start						
IHM Tree		2.75	2.75			3
Buttons adding on ButtonListener		2				
Reading & Comprehension of TSP file				1.5		
DSS Computing				1.25		
Observable / Observer implementation	0.5					
Creation of equality methods					1	
Drafting of the report	3.35	3.25	1			
Equality method test					1	
Creation of the adjacency list in the deserializer				1		
Compute of the adjacency list to get graph				1.5		
Creation of the class containing the PtoP graph				0.75		
Implementation of TSP related classes				2		
Creation of the tour in CityMap				2.25		
Dijkstra algorithm					2.5	
Map + Distribution display		1.5	3			4
Tour display				0.5		0.5
Debug ComputePath				1	1	
Code review	1	1	0.5	0.75	0.5	0.5
Report formatting		0.6	1.5			
IHM review				1		1
Exception handling		1				
Make a diagram package (zoom)	0.2					
Final presentation of the first iteration with the client	0.25	0.25	0.25	0.25	0.25	0.25
List the features and preparation for future iterations	1.25	1.25	1.25	1.25	1.25	1.25

Figure 6.1: Provisional planning first iteration