|  |
| --- |
| Human computer interaction |
| Semitag – External Specifications Documents |
| Master M1 MOSIG, Grenoble Universities |
| TCHECHMEDJIEV Andon |
| TCHOUGOURIAN Tigran |
|  |
| **05/04/2011** |



|  |
| --- |
| **Supervisors :**  Gaëlle CALVARY  François BERARD |

|  |
| --- |
| The goal of this project is to put in practice the User Centered Design method for designing interactive computer applications. |

Contents

[1. Design 2](#_Toc289778587)

[1.1. User Model 2](#_Toc289778588)

[1.2. Environment Model 2](#_Toc289778589)

[1.3. Platform Model 2](#_Toc289778590)

[1.4. Task Model 2](#_Toc289778591)

[2. Detailed scenarios 2](#_Toc289778592)

[3. Interface descriptions 2](#_Toc289778593)

[3.1. General description 2](#_Toc289778594)

[3.2. The input form 3](#_Toc289778595)

[3.3. Search bar algorithm 5](#_Toc289778596)

[3.4. Results 5](#_Toc289778597)

[3.4.1. The Summary view 6](#_Toc289778598)

[3.4.2. The Details view 6](#_Toc289778599)

# Design

## User Model

The application is designed for novice user. Familiar with paper based forms.

## Environment Model

The application is provided by the semitag web site. It has to be accessible from any computer which has an internet connection and a web browser (Internet Explorer 9.0/Firefox 4.0/Chrome 10.0 from Windows, Firefox 4.0 for Linux, and Safari 5.0 for Mac)

## Platform Model

The application runs on Personal Computer web browser. It uses the mouse and the keyboard to navigate and get input.

## Task Model

# Detailed scenarios

# Interface descriptions

When the user arrive on semitag search patch functionality it should be able to browse the answers in less than 2 click (submit click)

## General description

The page function is divided in two frames as shown on the Figure 3-3. The A frame manage the input of the user, whereas the B frame manage the results.

Frames can be toggled up and down (Figure 3-1 and Figure 3-2) to not pollute the user view. When the user starts a new search, the frame B is completely hidden. Once the request submitted e.g. [3.2 The input form](#_The_input_form), the frame A is closed (only the title of the frame is visible) whereas the frame B is visible and the results appear.

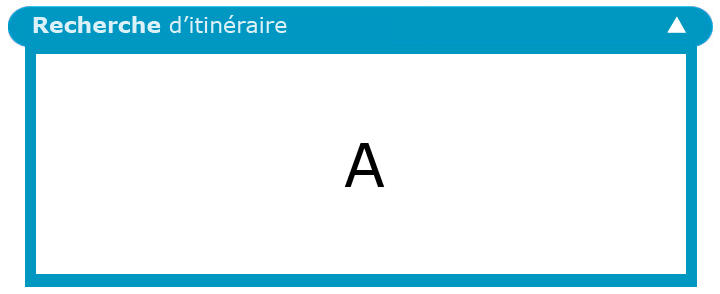


Figure 3‑1: Open Frame

The user can always reopen the frame A to refine his path. That means that the frame A should remember the input values to help the user. The frame B has to be open during the refining phase, and reloaded with the new results after the user submit.

C:\Users\NargiT\Université\MoSIG M1\S8 - Human Computer Interaction\Projet\img\closeFrame.png

Figure 3‑2: Close Frame

[3.2 The input form](#_The_input_form) instantiate the frame A, whereas [3.4 Results](#_Results) instantiate the frame B.

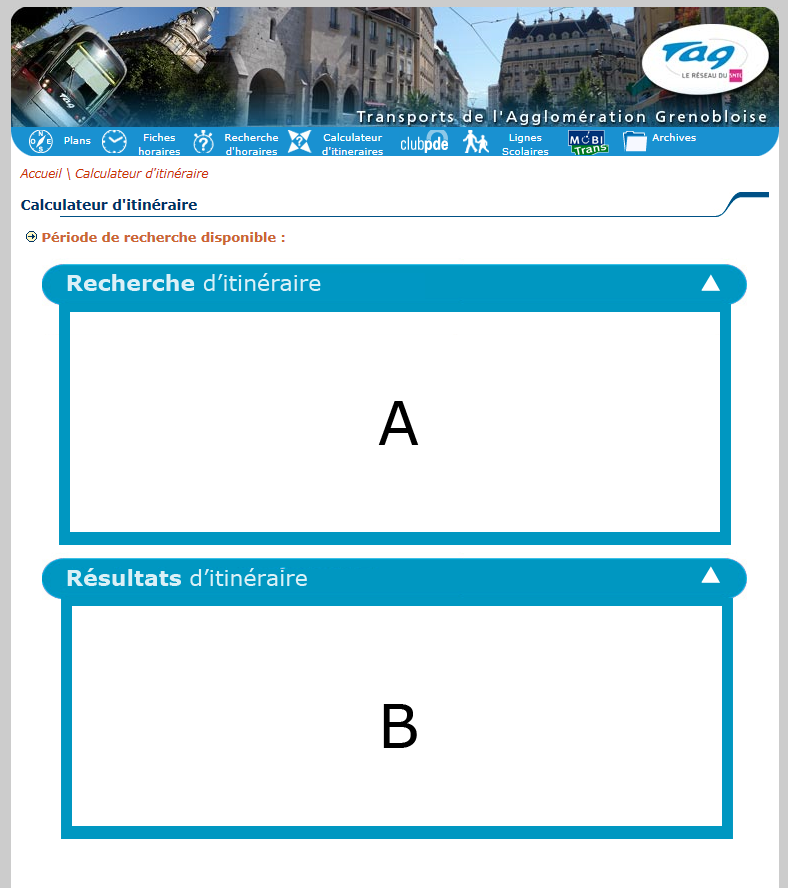


Figure 3‑3: General view of the function

## The input form

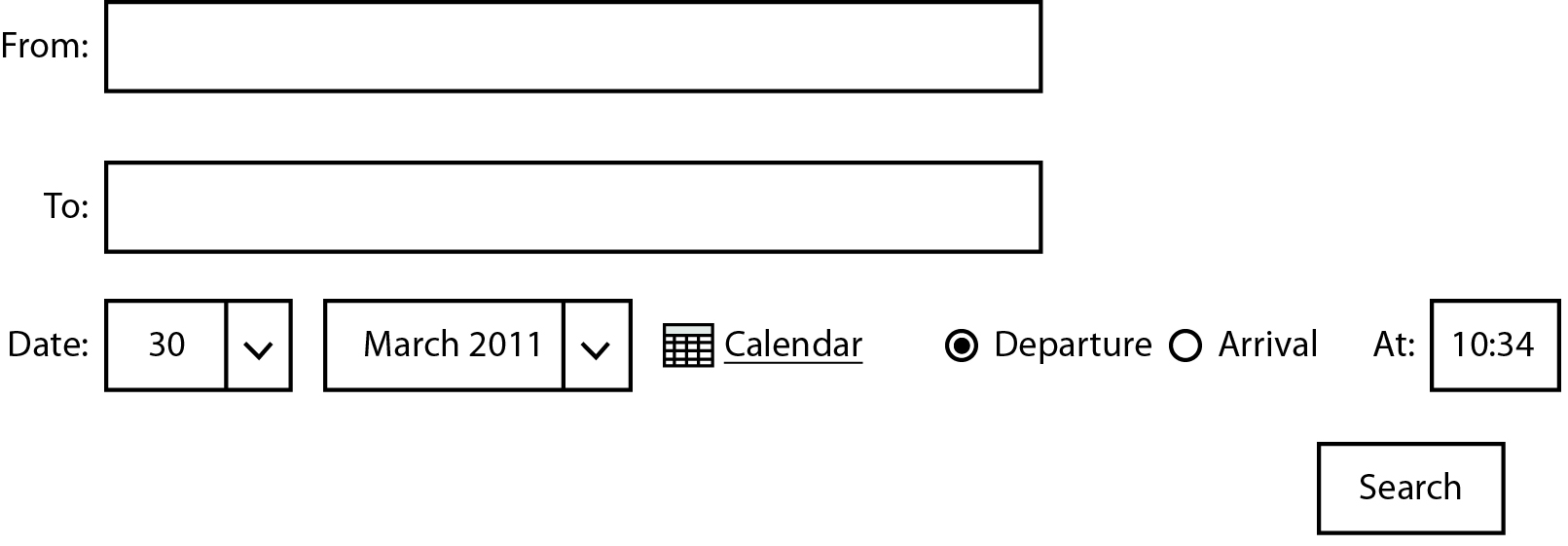


Figure 3‑4: Unique form to submit before to find an answer

The input form is divided in 4 different information parts:

* The “From” input field, which indicates the source of the itinerary. The user set its departure. e.g. [Search Bar algorithm](#_Search_bar_algorithm).
* Then the input field “To”, which indicates the destination of the itinerary. The user set its arrival. e.g. [Search Bar algorithm](#_Search_bar_algorithm).
* Date of usage. By default the date should be the current date. The user has two ways to set the date: With the drop down list or with a calendar script.
  + The drop list should respect the number of day per month.
  + The calendar by default is set to the current day.
* Which sort of search. By default “Departure” is selected and “at” field is set to the current time + 5 min. If the user clicks with the mouse to “Arrival”, it will automatically erase the content of “At” field and set the cursor on that field. The “At” field should be enough smart to let the user only input numerical values and the “:” char. The “Arrival” option represents the time that the user wants to arrive at the destination point.

Example:

|  |  |  |
| --- | --- | --- |
| Digits | Allowed | Transformed |
| 1 | 1 | 01:00 |
| 2 | 01 | 01:00 |
| 2 | 24 | 00:00 |
| 2 with colon | 01: | 01:00 |
| 2 with colon | 1:1 | 01:10 |
| 2 with colon | :10 | 00:10 |
| 3 | 100 | 10:00 |
| 3 | 230 | 23:00 |
| 3 with colon | 23:0 | 23:00 |
| 3 with colon | 1:10 | 01:10 |
| 4 | 1000 | 10:00 |
| 4 with colon | 10:00 | 10:00 |

Figure 3‑5: Input allowed and transformation

|  |  |
| --- | --- |
| Digits | Not allowed |
| 2 | 25 |
| 2 | 99 |
| 2 with colon | 25: |
| 2 with colon | 0:6 |
| 2 with colon | :60 |
| 3 | 250 |
| 3 | 106 |
| 3 with colon | 23:6 |
| 3 with colon | 1:60 |
| 4 | 2460 |
| 4 with colon | 10:60 |

Figure 3‑6: Corrupted input

The “search” button will submit the form if all the values are filled. If there is missing information or wrong data, a red rectangle has to surround the wrong input and the user has to redefine the new values.



Figure 3‑7: A missing field

In case of the “From” and “To” field a drop down list will be available with all the possible matches.

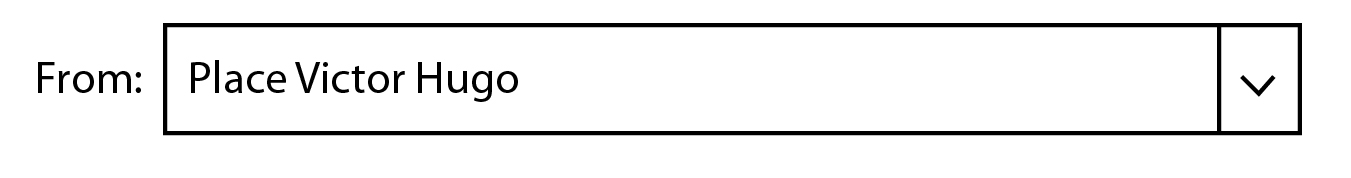


Figure 3‑8: Possible Matches

## Search bar algorithm

The tool uses a form to get information about the source and destination address/station. The input field “From” and “To” are not simple input field because they uses an algorithm that find for the user the exact name of the address or station that they use. The algorithm autocompletes under the input field the best matches. It is an incremental search, which means that to avoid requesting the database to often it should starts to search after 3 char. At most 7 values are returned. Each row is identifying with an icon on the right. A house represents an address, a bus represents a bus station, and a tram represents a tramway station.

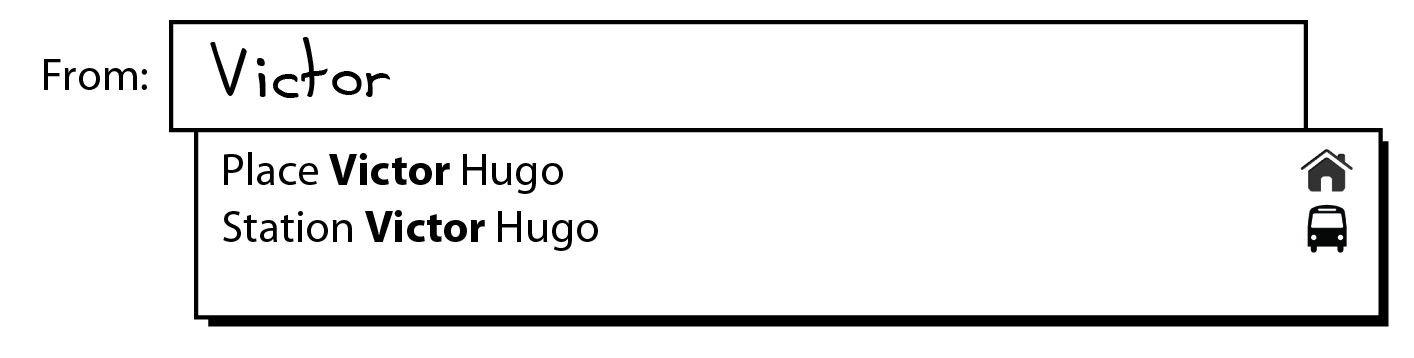


Figure 3‑9: A user tries to find a station or address with "Victor" inside

To help the user, the matching letters should be bolded. Only beginning of word should match not letter inside a word.

Example:

A user search for “André”, he starts to write “Andr”. The algorithm shouldn’t return Alex**andr**a

Signature of the function:

Precondition: The string should at least of length 3

Find (String address): List<Complete address, Image>

Postcondition: a list of 7 items at most, the item is composed of the full name of street/station and corresponding image (bus, house, tramway)

## Results

Once the input form finished and the form is submitted, a new window is toggled down and the previous one is toggled up. The results should be presented in the current form e.g. Figure 3-7. At most 3 results with different paths. They should be sorted by the departure time, and the first one should be selected.

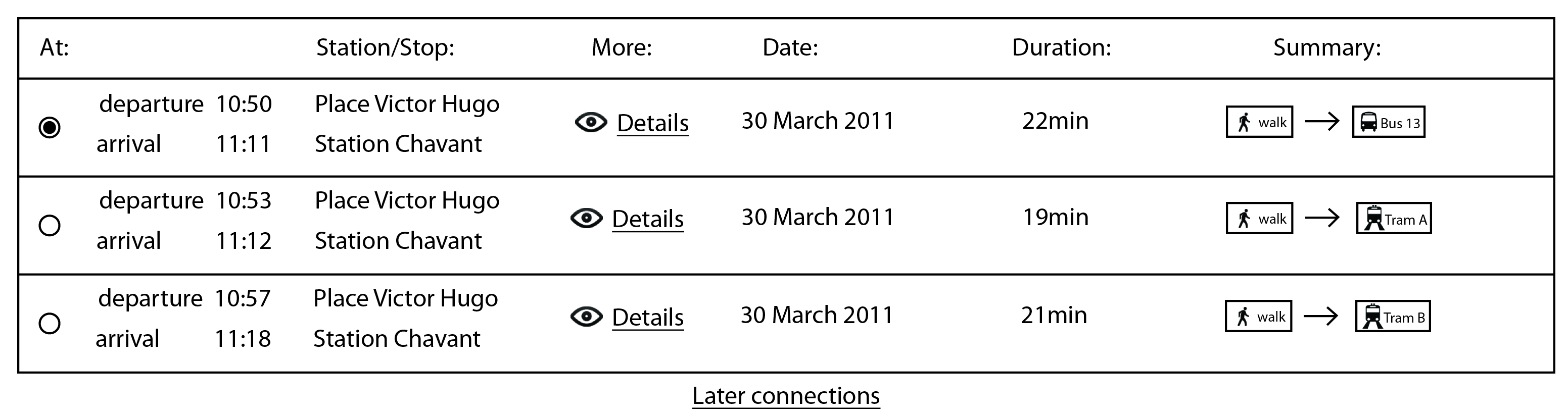


Figure 3‑10: The results

### The Summary view

C:\Users\NargiT\Downloads\1301477217_user_walk.pngThe “Summary” represents a fast way for the user to see what he/she should do during the trip.

A walking person is shown if one of the following actions has to be done by the user.

* The user has to go from one address to another one. If multiple address changes are performed, then the image is shown only once.
* The user has to go from one address to a station (bus or tramway).
* The user has to go from one station (bus or tramway) to another station (bus or tramway).
* The user has to go from one station (bus or tramway) to an address.



A bus with the identifier of the line is shown if one of the following actions has to be done by the user.

* The user has to take the bus

C:\Users\NargiT\Desktop\picto_train.png

A tramway with the identifier of the line is shown if one of the following actions has to be done by the user.

* The user has to take the tramway

The arrow represents the next step to be performed.

### The Details view

Each line can be opened to see in details the steps describe in the summary.

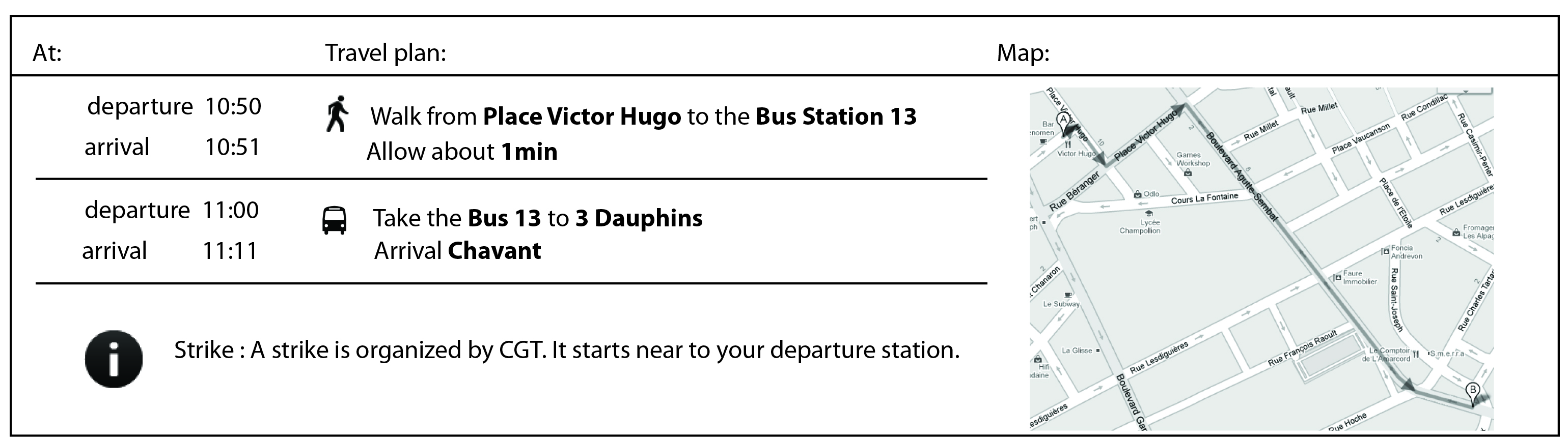


Figure 3‑11: Detailed view of one path