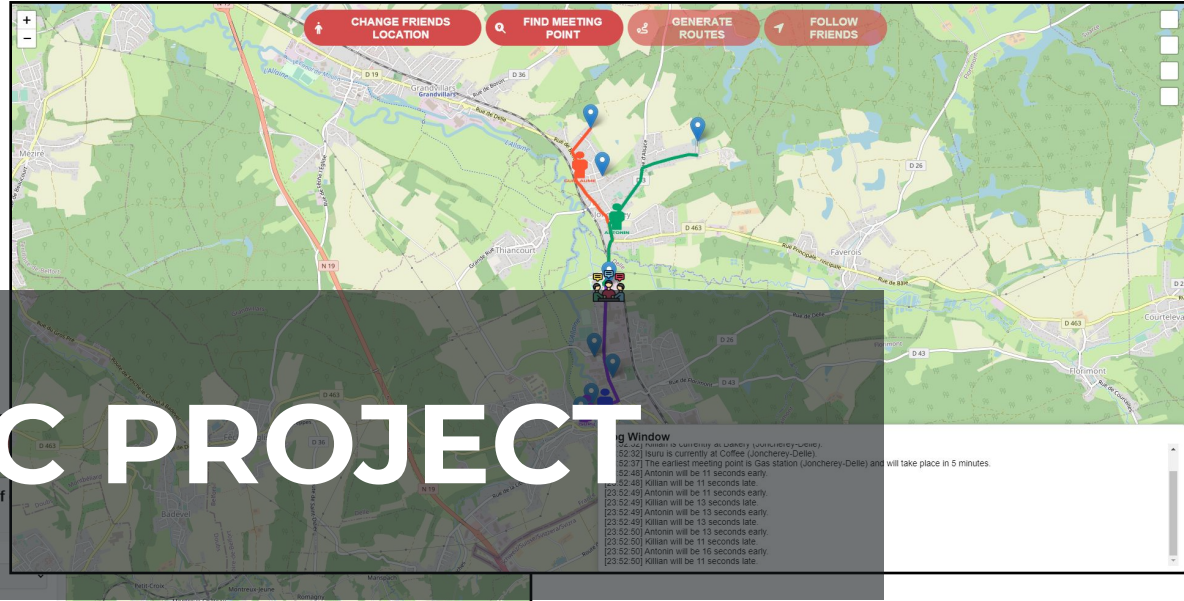
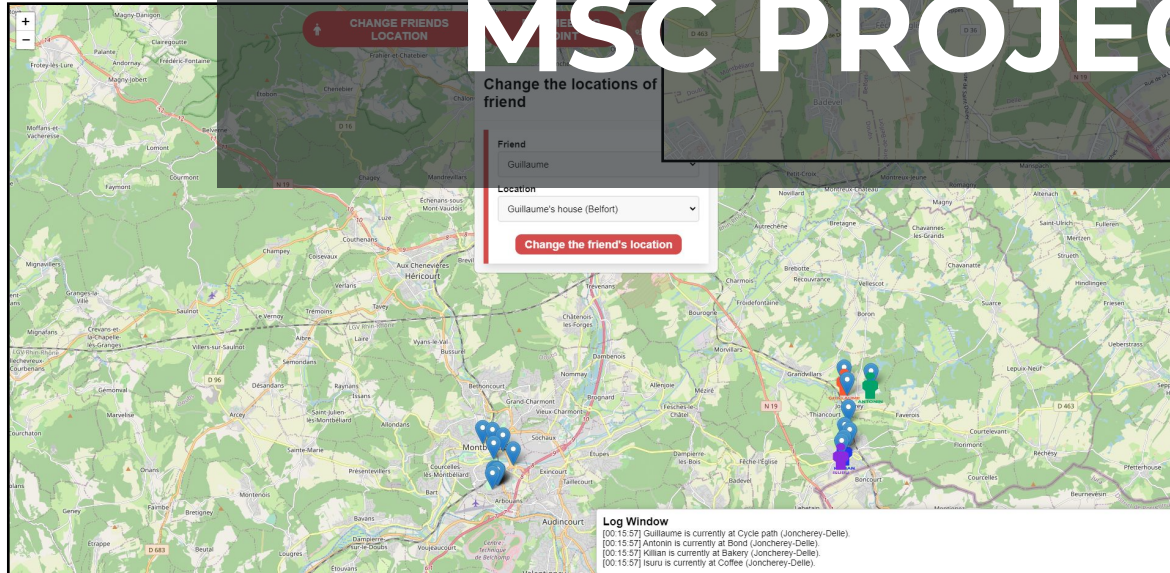


MSC PROJECT



Antonin WINTERSTEIN
Killian MAXEL

Teacher: Mr. Philippe CANALDA

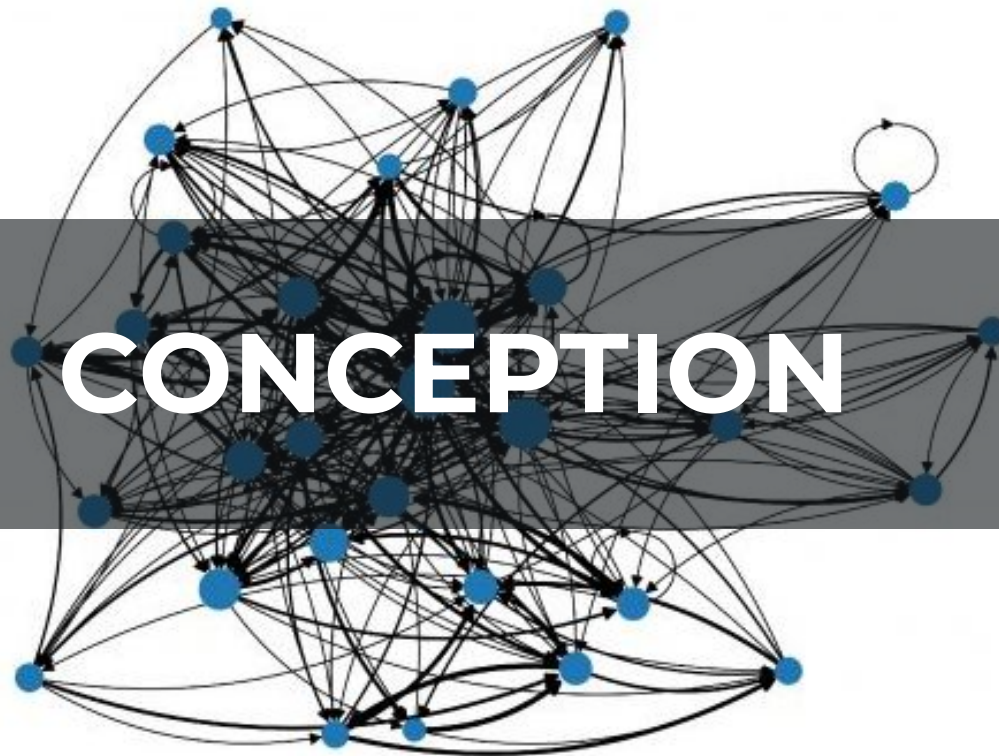
Optimizing Student Meetups with Location-Based App

GOAL

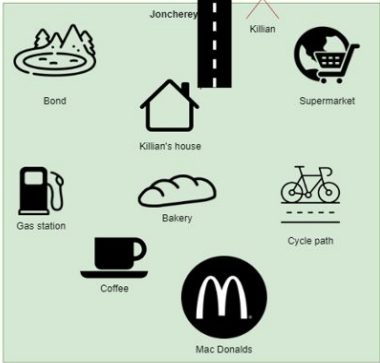
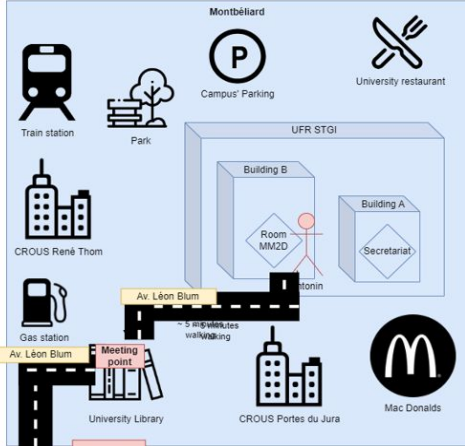
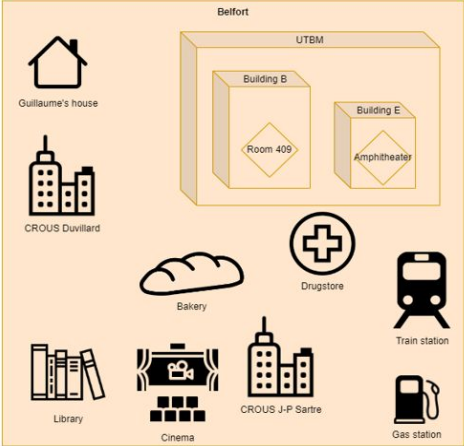
Creating a Mobile/Web App mapping friend's locations (home, university, parking...) using an Origin-Destination (OD) matrix for efficient group meetups

- ❖ OD matrix with 30 locations: distances, time estimates, routes
- ❖ Considering Belfort-Montbéliard environment
- ❖ Registering friends for accurate positioning
- ❖ Include roads for custom route generation and tracking

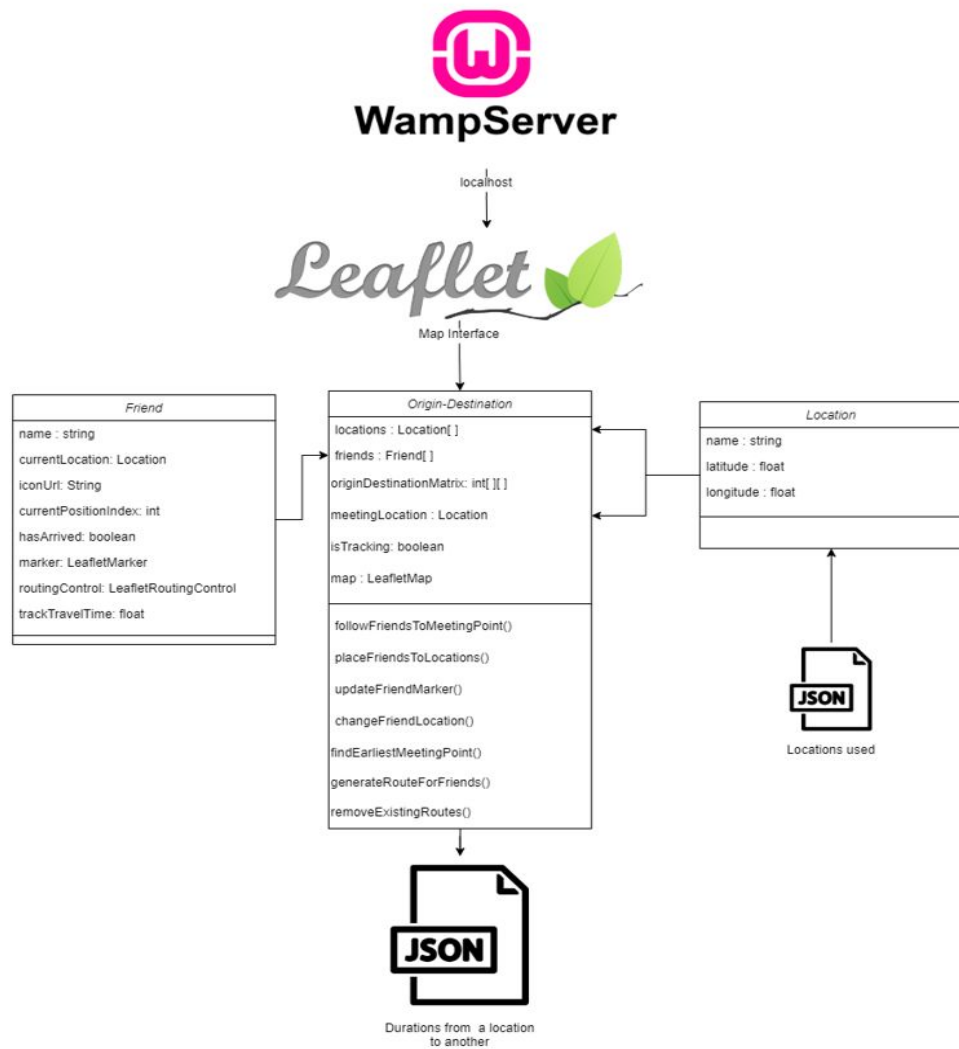
- ❖ Optimal Meetup Planning
 - Determine ideal meeting place and time window
 - Accounts for individual schedules and locations
- ❖ Individual Route Generation
 - Create personalized routes for each friend
- ❖ Real-Time Tracking
 - Monitor friends' progress to meetup location
 - Provide alerts for delays or advancements



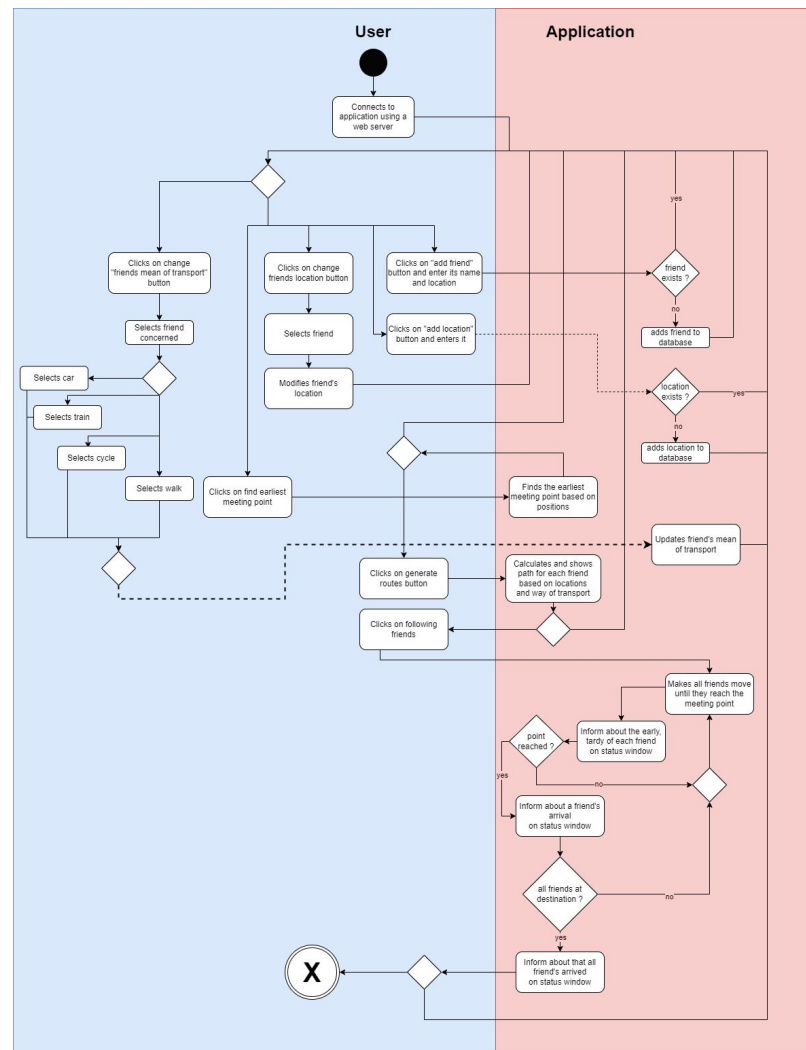
CONCEPTION DIAGRAM



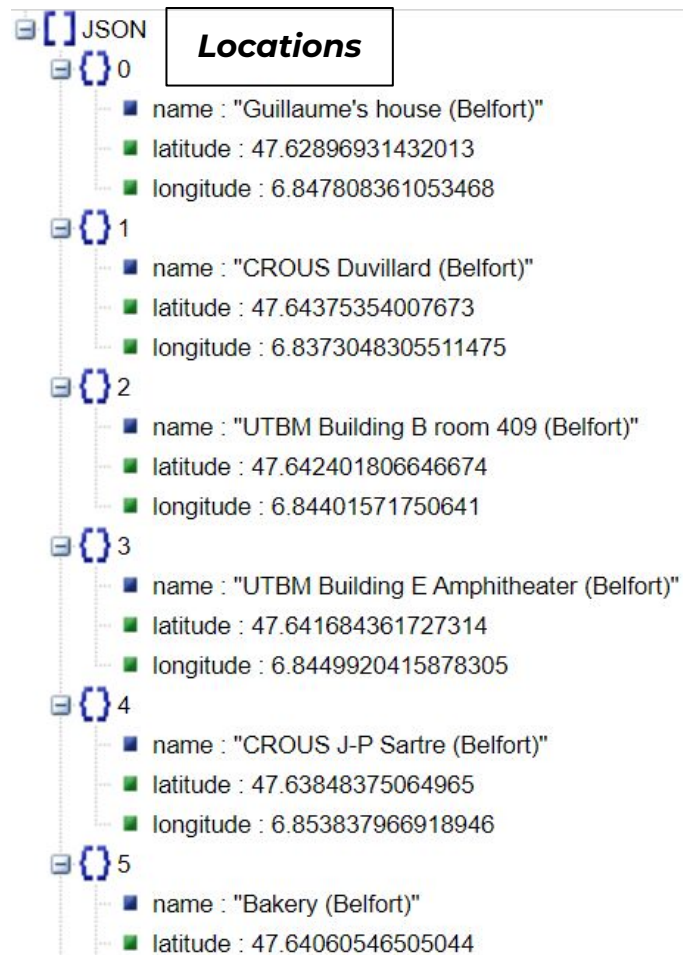
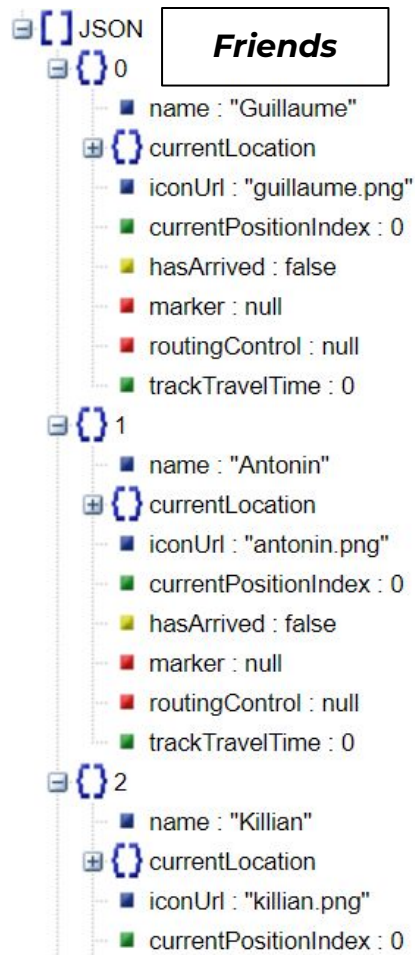
CLASS DIAGRAM



ACTIVITY DIAGRAM



JSON FILES



JSON FILES - ORIGIN DESTINATION MATRIX

```
[
  [
    0, 5, 6, 6, 5, 5, 5, 3, 7, 6, 8, 3, 18, 21, 17, 17, 18, 18, 19, 19, 19, 18,
    16, 23, 26, 24, 23, 22, 24, 24
  ],
  [
    5, 0, 3, 3, 6, 2, 6, 7, 9, 11, 6, 21, 24, 20, 20, 21, 21, 22, 22, 22, 21,
    19, 26, 29, 27, 28, 27, 26, 26, 27
  ],
  [
    6, 3, 0, 1, 4, 2, 5, 6, 8, 10, 6, 21, 24, 20, 20, 21, 21, 22, 22, 22, 21,
    19, 26, 29, 27, 28, 27, 26, 26, 27
  ],
  [
    6, 3, 1, 0, 4, 2, 5, 6, 8, 10, 6, 21, 24, 20, 20, 21, 21, 22, 22, 22, 21,
    19, 26, 29, 27, 28, 27, 26, 26, 27
  ],
  [
    5, 6, 4, 4, 0, 4, 2, 1, 5, 6, 2, 20, 22, 19, 18, 19, 19, 20, 20, 20, 20, 17,
    24, 27, 26, 26, 25, 24, 24, 25
  ],
  [

```

JSON

0

1

- 0 : 5
- 1 : 0
- 2 : 3
- 3 : 3
- 4 : 6
- 5 : 2
- 6 : 6
- 7 : 7
- 8 : 9
- 9 : 11
- 10 : 6
- 11 : 21
- 12 : 24
- 13 : 20
- 14 : 20
- 15 : 21
- 16 : 21
- 17 : 22
- 18 : 22
- 19 : 22
- 20 : 21

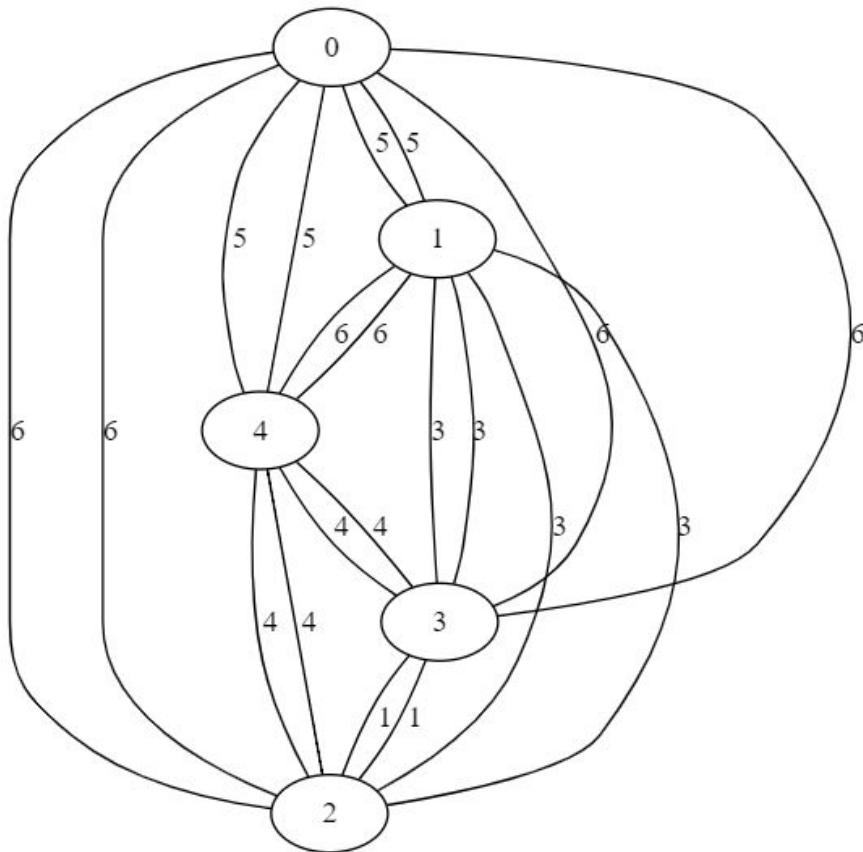
*Each value is the duration to go from point A to point B
(if A=B then 0) (same value from point B to point A)
matrix symmetry, reflexivity and transitivity are checked at launch*

RESULT

GRAPH REPRESENTATION

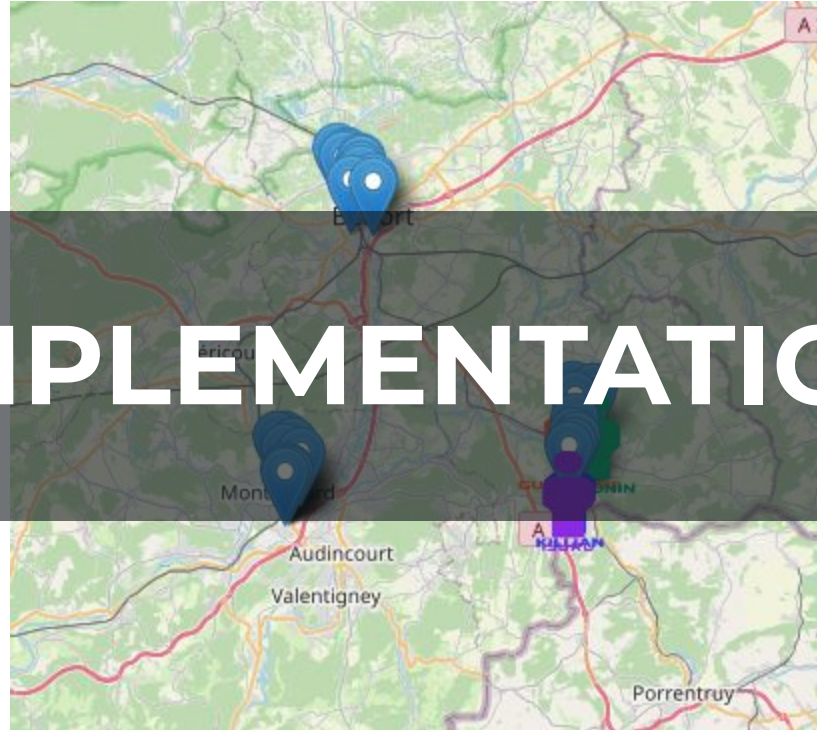
- From location 0 to location 1, duration of 5 minutes
- From location 1 to location 0, duration of 5 minutes (so **symmetry**)
- From location 0 to location 0, duration of 0 minutes (**reflexivity**)

Transitivity in our case means that if we have (i,j) and (j,k) , we should also have (i,k) . In our graph, this principle explains why $(0,3) = 6$ despite the absence of direct binary relations. that's why $(i,j) + (j,k) \neq (i,k)$ in many cases

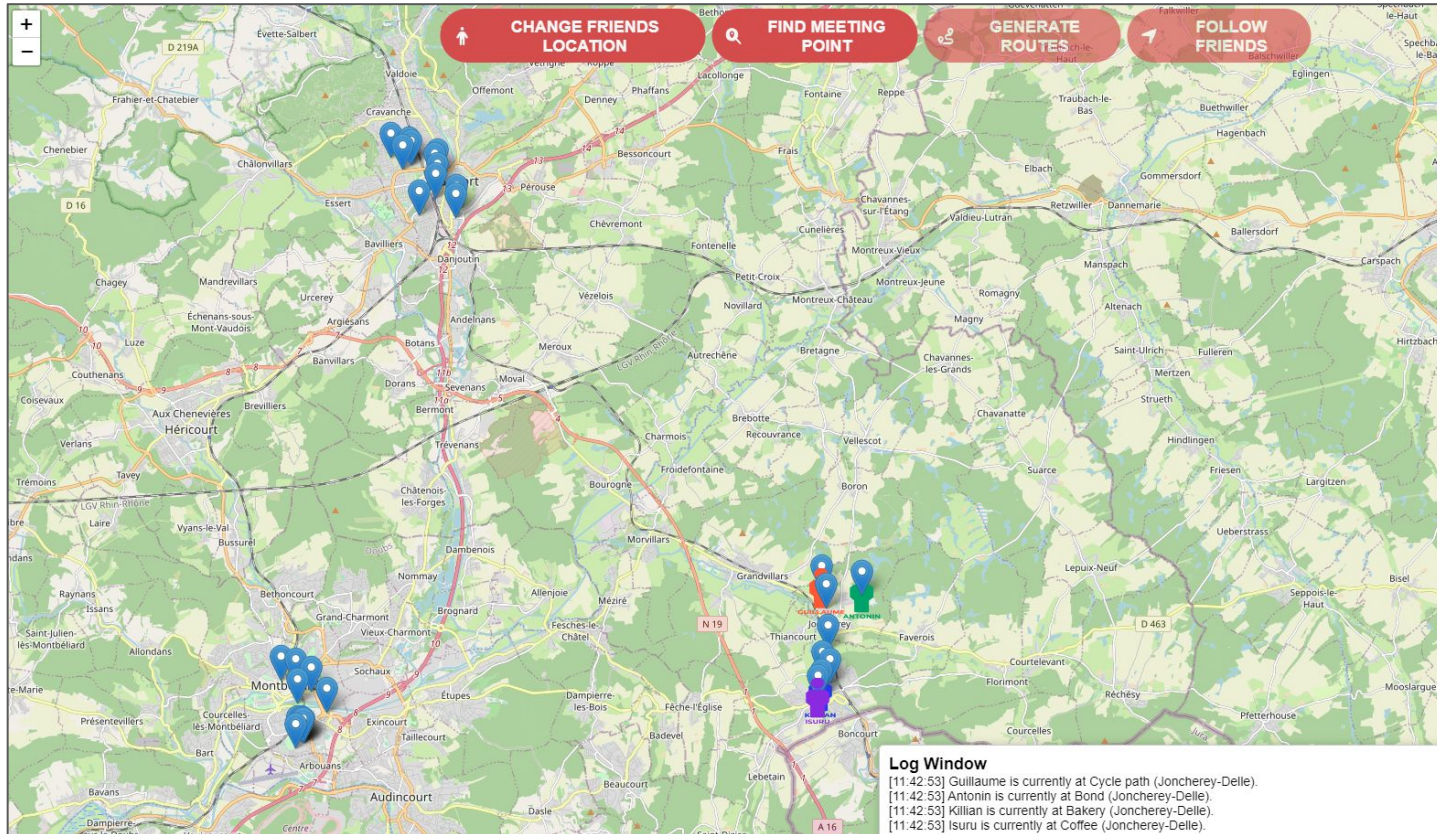


output of .dot file using our matrix (reduced
to 5 states for visibility)
([link to full graph](#))

IMPLEMENTATION

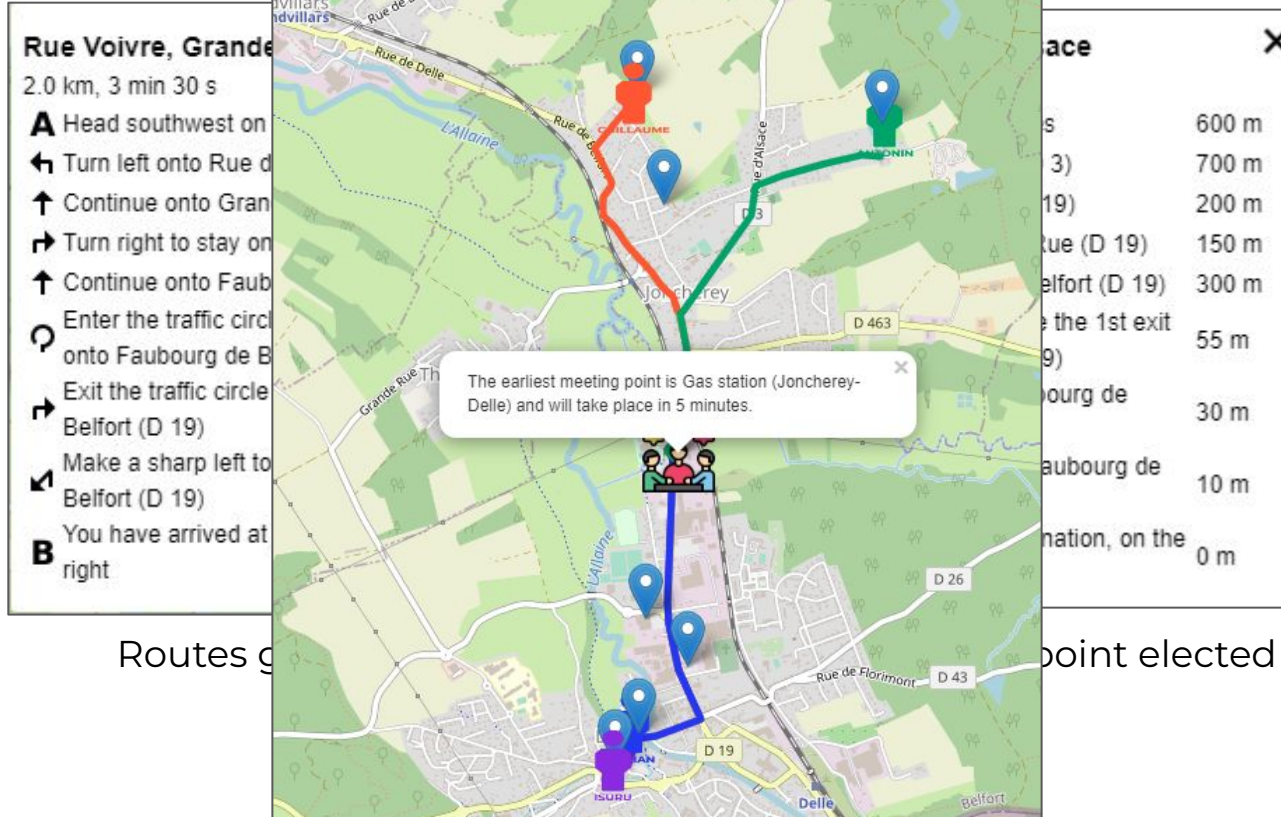


INTERFACE



RESULT

GENERATED ROUTES



Routes g

point elected

RESULT

TRACKING

Logs concerning friends (late, early, arrived, etc.)

Log Window

[14:46:12] Guillaume is currently at Cycle path (Joncherey-Delle).
[14:46:12] Antonin is currently at Bond (Joncherey-Delle).
[14:46:12] Killian is currently at Bakery (Joncherey-Delle).
[14:46:12] Isuru is currently at Coffee (Joncherey-Delle).
[14:46:15] Guillaume is currently at Guillaume's house (Belfort).
[14:46:22] Guillaume is currently at Bakery (Joncherey-Delle).
[14:46:25] The earliest meeting point is Gas station (Joncherey-Delle) and will take place in 5 minutes.

Log Window

[14:47:22] Antonin will be 11 seconds late.
[14:47:22] Isuru will be 13 seconds late.
[14:47:23] Antonin will be 13 seconds late.
[14:47:23] Isuru will be 13 seconds late.
[14:47:23] Antonin will be 11 seconds late.
[14:47:23] Killian will be 11 seconds late.
[14:47:23] Isuru will be 15 seconds late.
[14:47:24] Antonin will be 13 seconds late.
[14:47:24] Killian will be 11 seconds late.
[14:47:24] Isuru will be 17 seconds late.
[14:47:24] Antonin will be 11 seconds late.
[14:47:24] Killian will be 13 seconds late.
[14:47:24] Isuru will be 19 seconds late.

Log Window

[14:48:03] Isuru will be 23 seconds late.
[14:48:03] Isuru will be 25 seconds late.
[14:48:04] Isuru will be 27 seconds late.
[14:48:04] Isuru will be 29 seconds late.
[14:48:05] Isuru will be 31 seconds late.
[14:48:05] Isuru will be 33 seconds late.
[14:48:06] Isuru will be 33 seconds late.
[14:48:06] Isuru will be 33 seconds late.
[14:48:07] Isuru will be 31 seconds late.
[14:48:07] Isuru will be 33 seconds late.
[14:48:08] Isuru will be 31 seconds late.
[14:48:08] Isuru has arrived at the meeting point!
[14:48:08] All friends have arrived!

CONCLUSION



What's next?

Adding friends / locations / OD matrices

- ❖ Add a feature to allow the user to add or remove new friends or locations
 - Would help the user customize depending on his own need
 - Give more flexibility

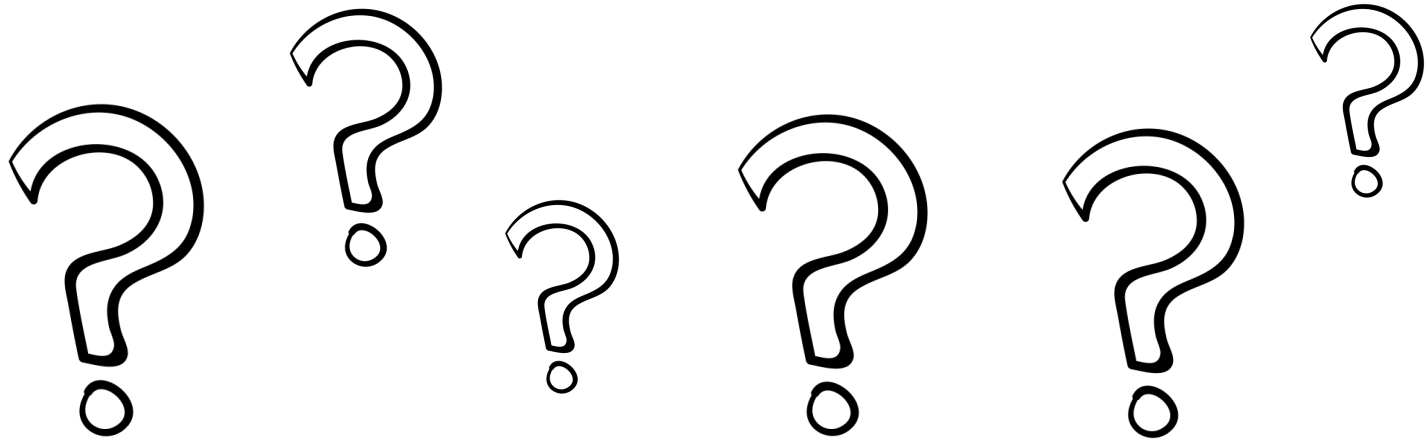
Taking into account ways of transport

- ❖ Currently only based on time travel with car
 - Could take into account other means like bus, bicycle, train, walking...
- ❖ Allow the user to select what transport mean he wants
- ❖ Adapt the travel using several means at once

Improving the tracking

- ❖ Include real-time data like traffic information, weather conditions or public transportation schedules
- ❖ Instead of relying on static data, consider integrating real-time data sources for tracking, like GPS data from mobile devices.

THANK YOU FOR YOUR ATTENTION



DO YOU HAVE ANY QUESTION?
