

Master of science in Engineering in Computer Science

a.y. 2018/2019 | 2nd semester

Human Computer interaction

Submitted to:

Tiziana Catarci
Valeria Mirabella

Presented by:

Leonardo Balzoni
Diego Giancane
Federico Giulianelli
Marawan Hassan



Index

1. Abstract	2
2. Introduction.....	3
2.1 Overview.....	3
2.2 Competitor analysis.....	3
2.3 User profile, Personas and Scenarios	4
2.4 Requirement analysis	6
3. Work plan	14
4. Design and analysis.....	14
4.1 Definitions	14
4.2 Search route (HTA and STN)	15
4.3 Point of interest (HTA and STN)	17
5. Prototype 1	18
5.1 Prototype description.....	18
5.3 Prototype 1 Mock-ups	19
5.4 Heuristic evaluation.....	26
6. Prototype 2	30
6.1 Prototype 2 description.....	30
6.2 Think aloud	32
6.3 Anova evaluation.....	32
7. Final product	35
8. Conclusion & future works	39

1. Abstract

Hermes is an application which aims to provide its users the best way to move from their location to a chosen one helping the user to have in a single application different kind of transportation service without having to switch between multiple app in order to make a search and have multiple results all together also to make easy the process of comparing each other.

After searching for directions, the app offers different possible paths with different transportation services basing itself on the average cost and time of that option. After choosing an option the user can either make the reservation or see the path on google maps. Another section of the application will show the user's current reservations also reminding with push notifications when some event is taking place.

One last important feature will be to find points of interest around a selected location. The different points will be divided based on their category (museums, restaurants, pub and night club).

This application is developed considering the User Centered Design approach, focusing on students as the main stakeholders.

2. Introduction

2.1 Overview

In this document we will present a brief timeline of development of the whole project starting with collecting the user requirements, moving through the implementation of the mock-up and using the heuristic evaluation to establish the second prototype. Lastly, we used the results of user evaluation as Anova and think aloud to build the final product.

2.2 Competitor analysis

Before analysing personas and scenarios we look at the market, analysing possible competitors focusing on their strengths, weaknesses and major features they provide:

	Moovit	Car2go	Uber	Kayak	CityMapper
Core feature	Show all nearby transport services.	Find and book a car near the user.	Allows booking of taxi like services.	Allows to find the best prices when travelling.	Compare different transportation systems.
Functional strengths	Group up many different transport services. Suggests the best path.	Easy to use.	Works in lots of countries. Allows for reservation of predefined trips.	The comparison it's done between lots of different options.	Lots of different options. Allows to share current progress.
Functional weaknesses	Inaccurate, sometimes leads to false information.	Inaccurate location system. High costs in peak hours.	More expensive than its competitors.	Offered prices are often different from the real ones.	Weak GPS localization.
Customer base	50 millions users	1 millions users	50 millions users	10 millions users	5 million users
Satisfaction score	4,4	4,1	4,3	4,1	4,5
Suggested path	V	X	X	X	V
Different transportation services	V	X	X	X	V
Book ticket	X	X	V	V	V
Call the service	X	X	V	V	V
Push notification	V	V	V	V	V

Looking at this table we can see many major features in an app that are missing on the other and overall we noticed that on the market is missing something that provides services for all kind of transportation, so our goal is to group up any major feature provided for all major transportation service.

2.3 User profile, Personas and Scenarios

User profile:

Age	16-50 (avg. 25)
Gender	50% Male, 50% Female
Job Titles	Any
Location	Italy
Technology	Some basic app knowledge
Disabilities	Not Applied

Personas:

Marawan Hassaan is a 23 years old computer science student who decided to go for an exchange program in Rome.

His main objective is to get his masters degree and at the same time enjoy the beautiful city of Rome. To do so, he will have to move between lots of different locations.

He has a smartphone that allows him to search for transport logistics on multiple apps, both online and offline.

Eleonora Rossi is a 46 years old journalist who doesn't own a car. She works at the press agency «ANSA» and lives in the Monteverde neighborhood.

Each morning she needs to get from home to her workplace. Her issue is that often the public transportation does not work properly, so she would like to have real time updates on which is the best way to reach «ANSA». She uses her smartphone to check the metro/bus updates.

Scenarios:

It is a Saturday evening and Marawan would like to do something different with his friends. Since he just came to Rome he still does not know nice restaurants and clubs to where he could go to spend the night.

His need is to find a restaurant not too far from the place where he lives, and to follow it with a nice bar or club where to go and have fun.

Since he too doesn't own a car he would like to know which bus/metro would bring him to the restaurant or how much it would cost him to get there by using a taxi service.

It is Friday morning and ATAC system is having a strike, Eleonora needs to get to work and she would like to know which buses are passing by and how much it would cost if she were to take a UBER.

She would like to compare the time needed for the free bus option against the cost of getting an UBER. This information can be used to decide when she should leave or how much she would have to spend.

At the moment, her only option is to download a different app for each service and manually compare the results, this not only wastes her time but because of complexity, it might not direct her to the best option.

2.4 Requirement analysis

Questionnaire:

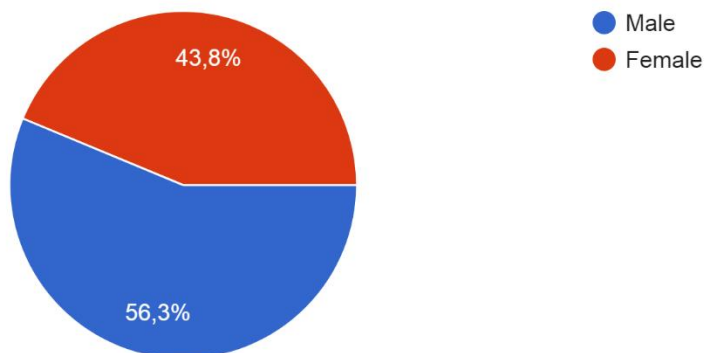
We selected few questions to be asked to the public, we posted it on google form on 2nd April

User background:

1.

What is your gender?

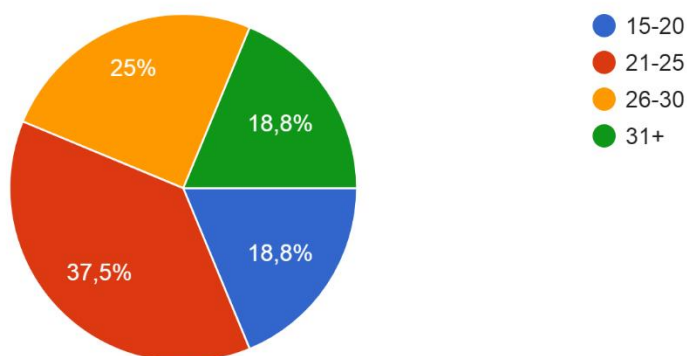
16 risposte



2.

What is your age?

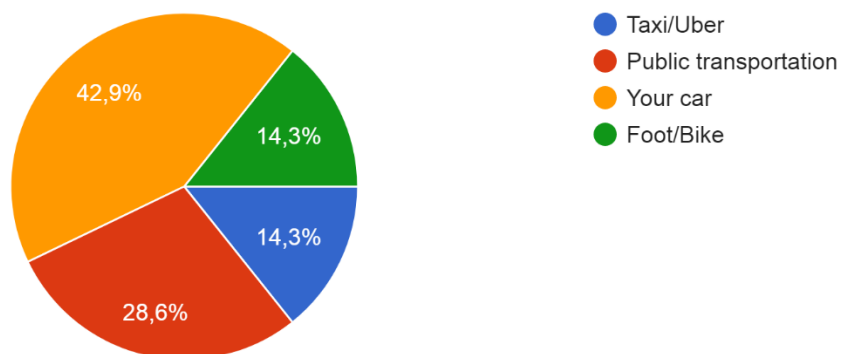
16 risposte



3.

Which of these transportation services you use the most?

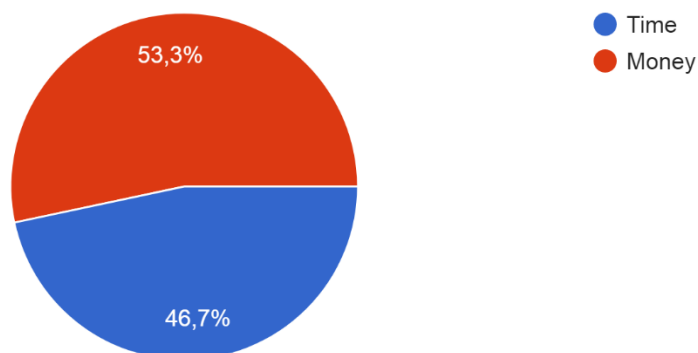
14 risposte



4.

Are you more interested in saving time or money when you commute?

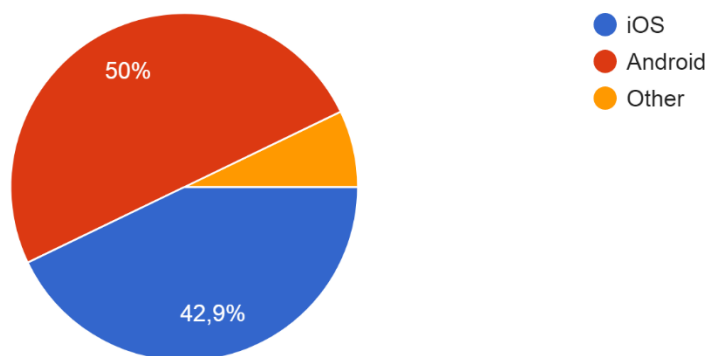
15 risposte



5.

Which mobile operative system you are currently using

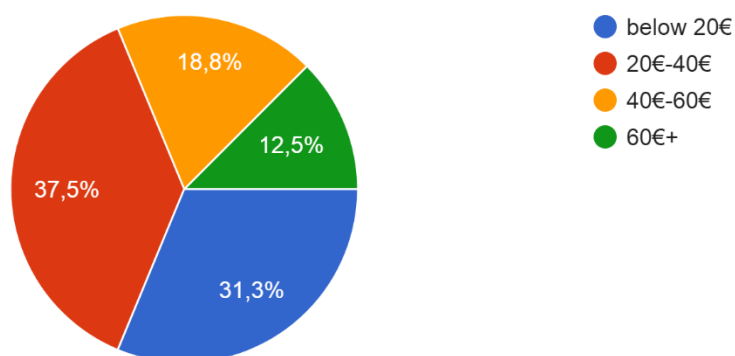
14 risposte



6.

How much do you spend per month in transport services? (avarage)

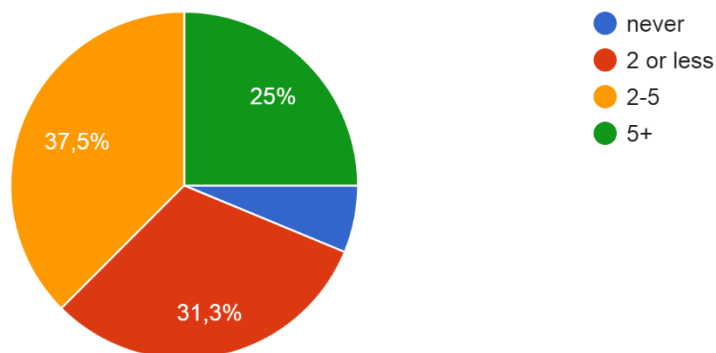
16 risposte



7.

How often do you use transportation services per week?

16 risposte



Conclusions:

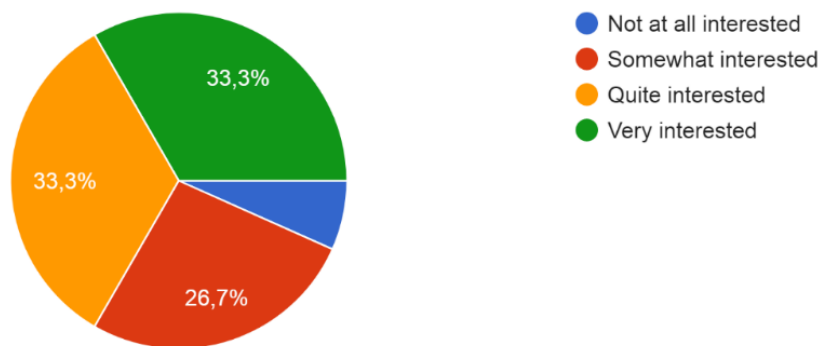
1. 56.3% of the participant that answered this questionnaire are male and 43.8% are female.
2. Over 60% of the students have an average age between 21 and 30 years old.
3. Most of the students who did the research use the car as transportation service.
4. 53,3% of the students prefer to save money rather than time.
5. Although Android users constitute half of the percentage of the result, we cannot neglect iOS percentage which is also significant.
6. Almost 70% of the students spend less than 40€ on transport services.
7. The majority use transportation service at least one time per week.

Product background:

1.

After reading this product description, how interested are you in the product itself?

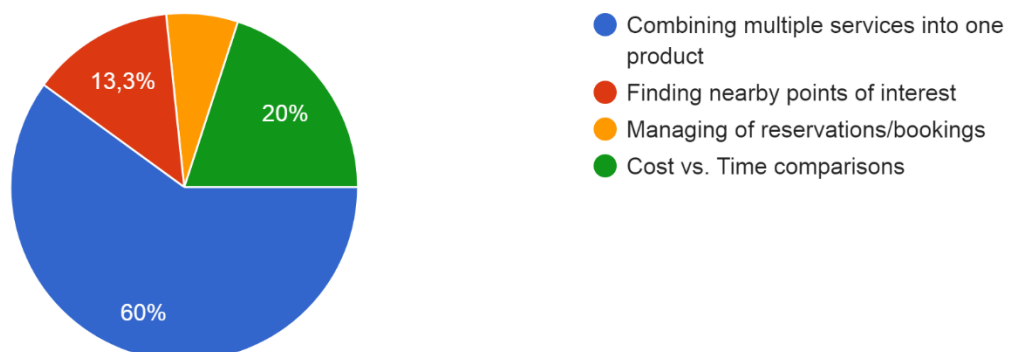
15 risposte



2.

Which of the following features interests you the most?

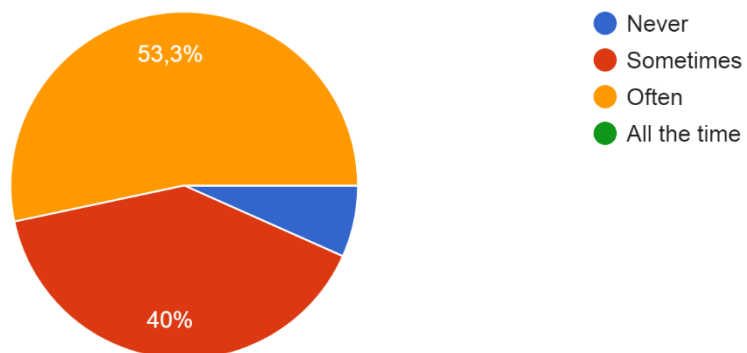
15 risposte



3.

How often do you think you would use this app?

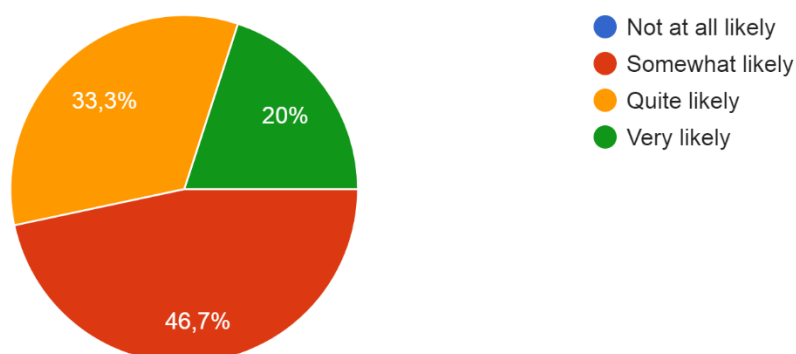
15 risposte



4.

How likely is it that you would recommend this product to a friend or colleague?

15 risposte



Conclusions:

1. The majority of the students are interested in the product
2. In particular, they are interested in combining together multiple services (60%)
3. Furthermore, they will frequently use this app
4. Students would recommend this product to others

Interview:

We focus on students aged from 21 to 30 years old, they use everyday transportation services, especially to reach university or work. The interviewees said that they want to reach a place as fast as possible. Also, they would like to be informed on possible public strikes and, in that case, evaluate multiple alternatives in order to choose the best option. In addition, students said they love to go to restaurant, museum or pub, chilling out and having fun with friends.

Conclusion: after collecting the requirements and analyzing the data, we were able to start our first prototype. We conclude that:

- Since Android and IOS are used equally, we will implement our app using ionic platform which is a cross platform framework giving its users the ability to implement or design application supporting both systems. It is built on Angular platform for building mobile and desktop web applications.
- We will show results from searching route function ordered by time.
- We will show information about transportation services.
- We will add a point of interest page in which a user can choose between different categories of entertainment places.

3. Work plan

After collecting the requirements and analyzing the data, we started designing the first prototype by making the HTAs and STNs. Then, we will present our mockups. Also, we will apply on the mockups the expert-based evaluation techniques, including heuristic and ones. Finally, in the prototype 2 we will apply other evaluations, including user-based evaluation (think aloud) and controlled evaluation (Anova), and obtain the final product.

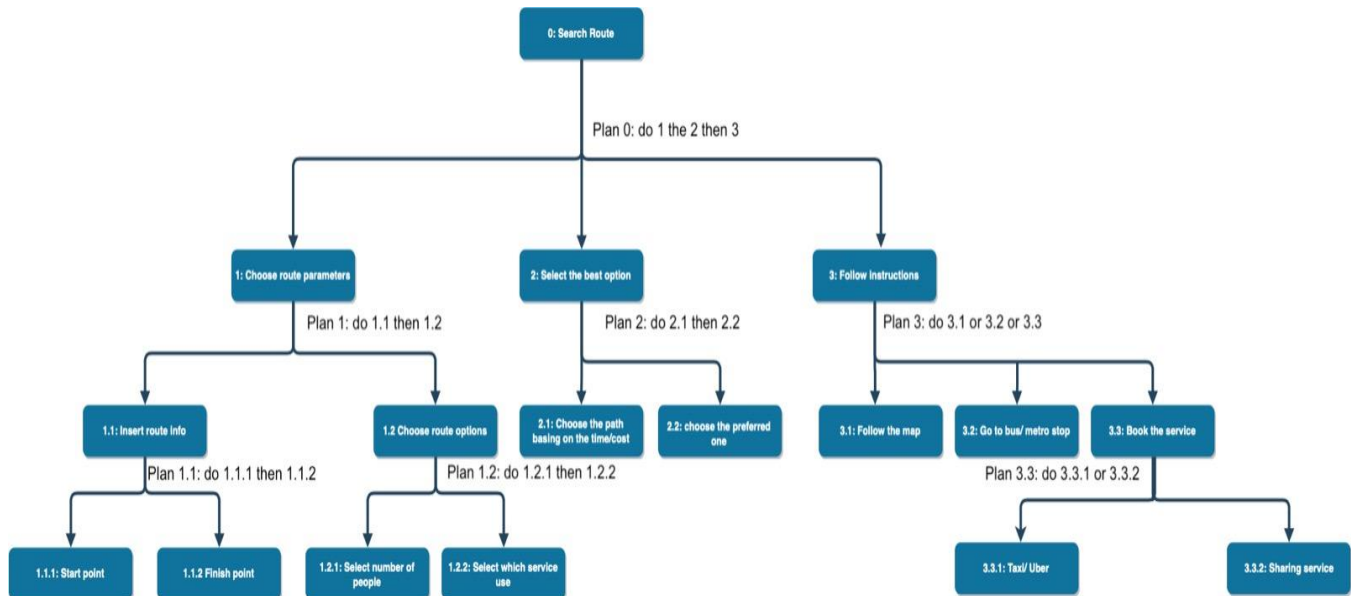
4. Design and analysis

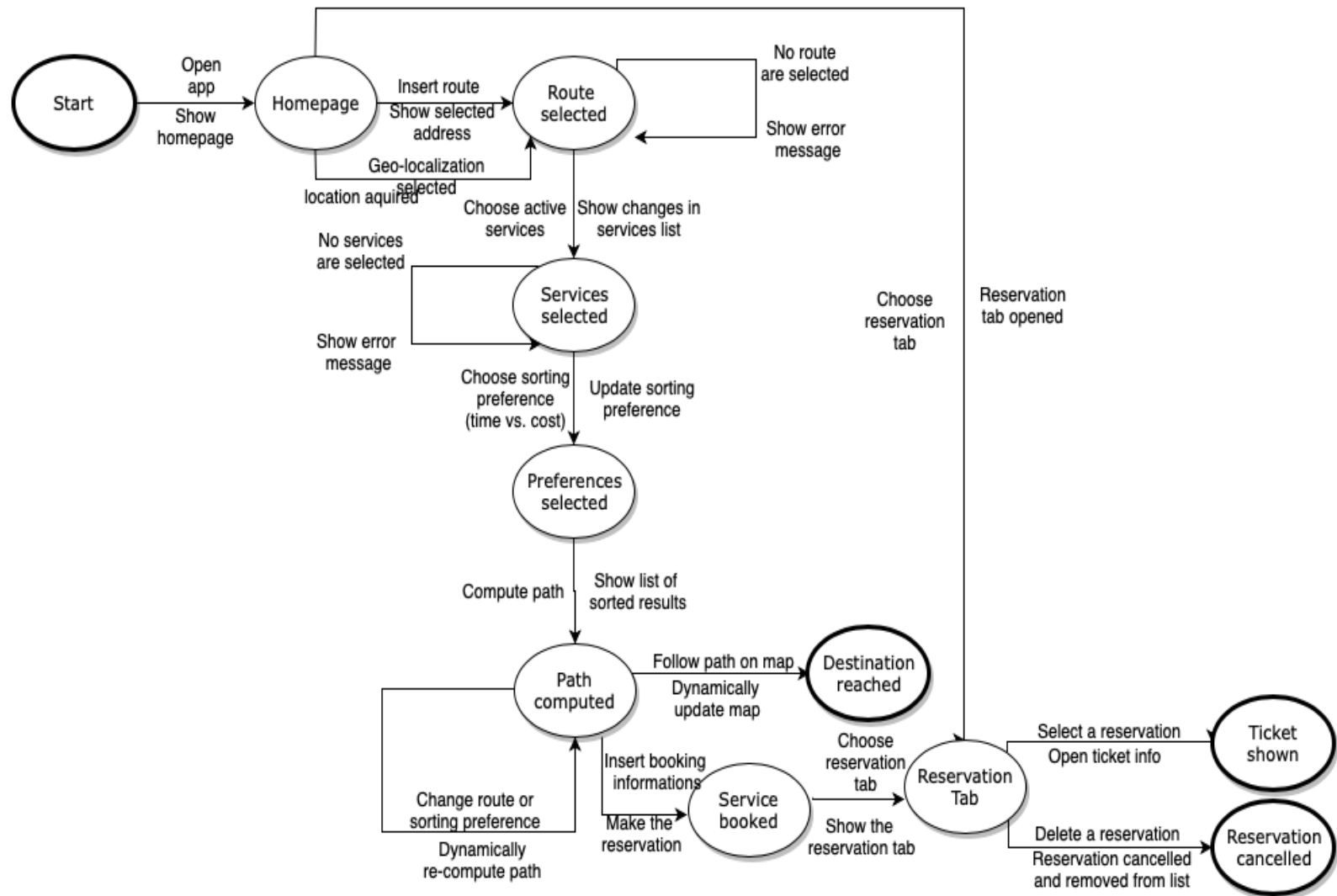
4.1 Definitions

Hierarchical Task Analysis (HTA) is a task description method, a variant of task analysis and is a necessary precursor for other analysis techniques. HTA is used to produce an exhaustive description of tasks in a hierarchical structure of goals, sub-goals, operations and plans. In HTA, tasks are broken down into progressively smaller units.

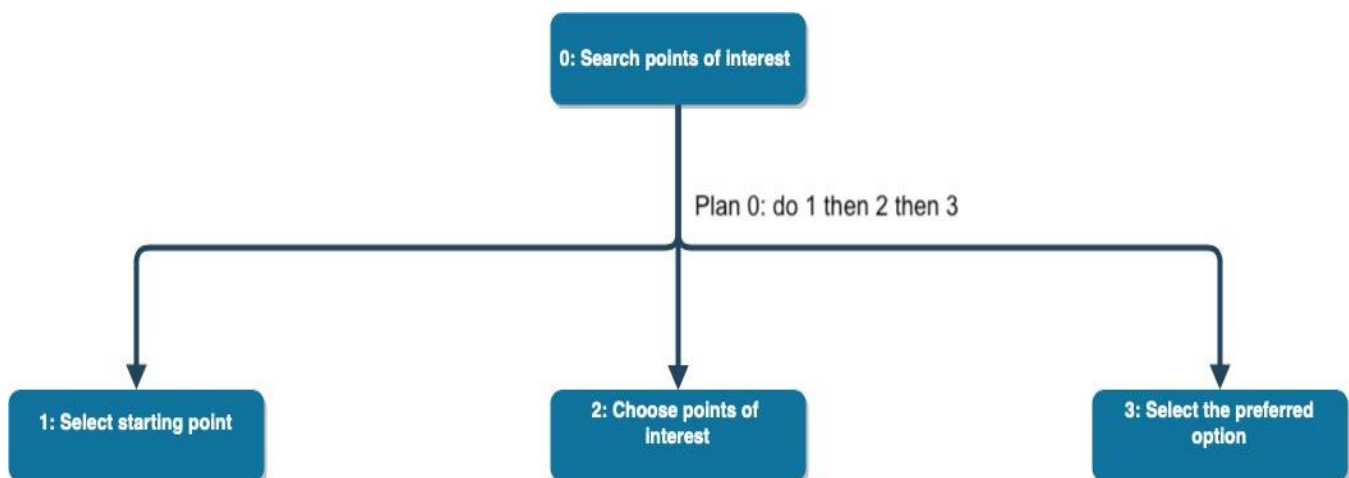
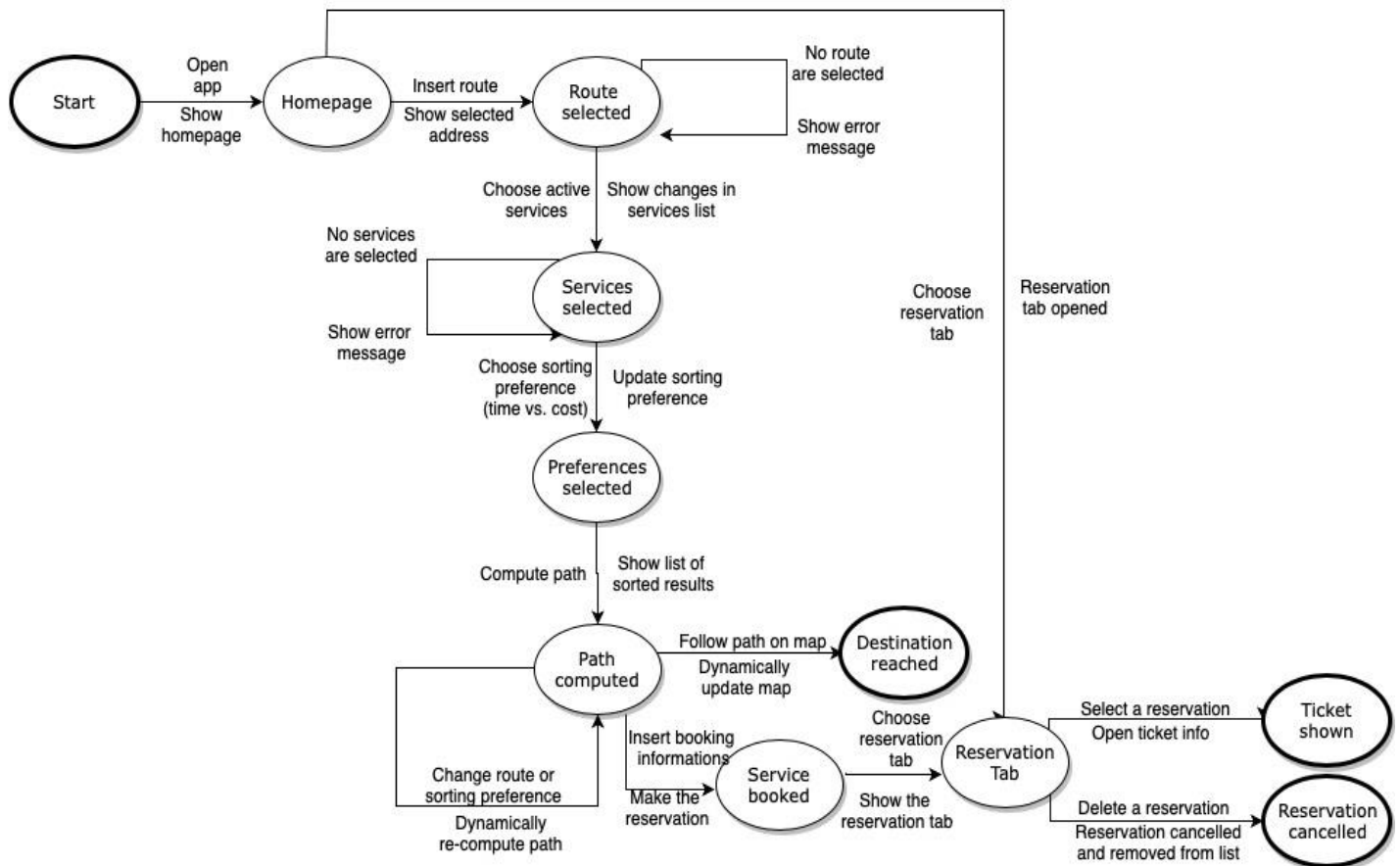
A state transition network (STN) is a diagram that describe a process as a sequence of non-concurrent actions. Nodes are state of the program and arcs are action performed with a related feedback. The transition begins from an initial state and follows a sequence of actions reaching a final state.

4.2 Search route (HTA and STN)





4.3 Point of interest (HTA and STN)



5. Prototype 1

After we finished the design part with modelling the possible tasks, we implemented our first prototype.

5.1 Prototype description

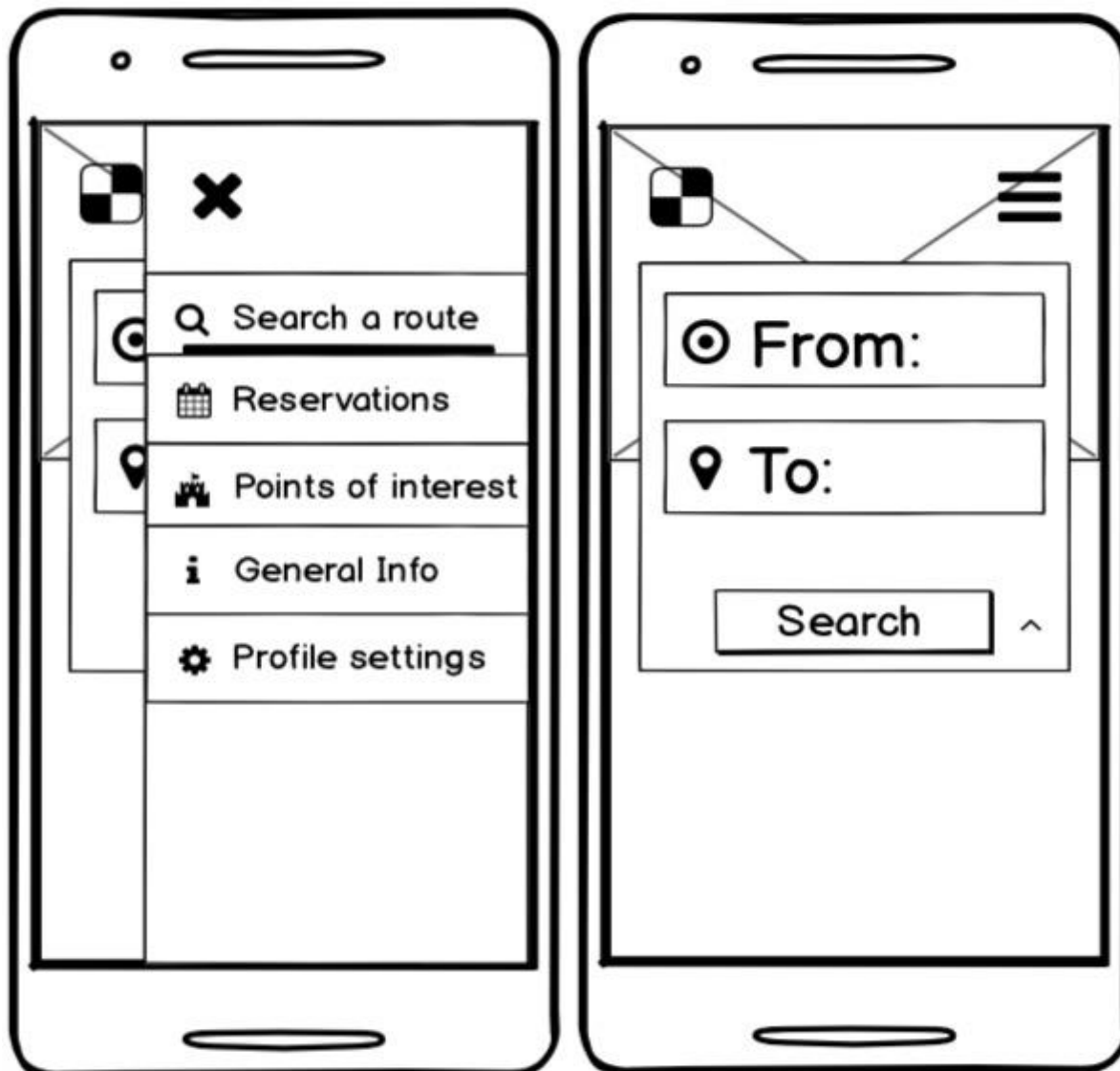
Taking into consideration STN and HTA we built the mock-ups.

We implemented the search route function where you must insert the start point, the destination and choose which services you are searching for. After that you click search you are redirected to a page where you can compare the results based on the service. This can be done by choosing a tab from the top selection.

We also implemented the Point of interest function. This allows the user to select a location and to find interesting places nearby. The user can even filter those results by selecting - for example - restaurant, pub, night club or museum. After you click on search, various cards are shown. Each card contains information regarding some place, different kinds of places are grouped into different tabs.

We finally have a last page where you have further more information regarding every service we implemented.

5.3 Prototype 1 Mock-ups

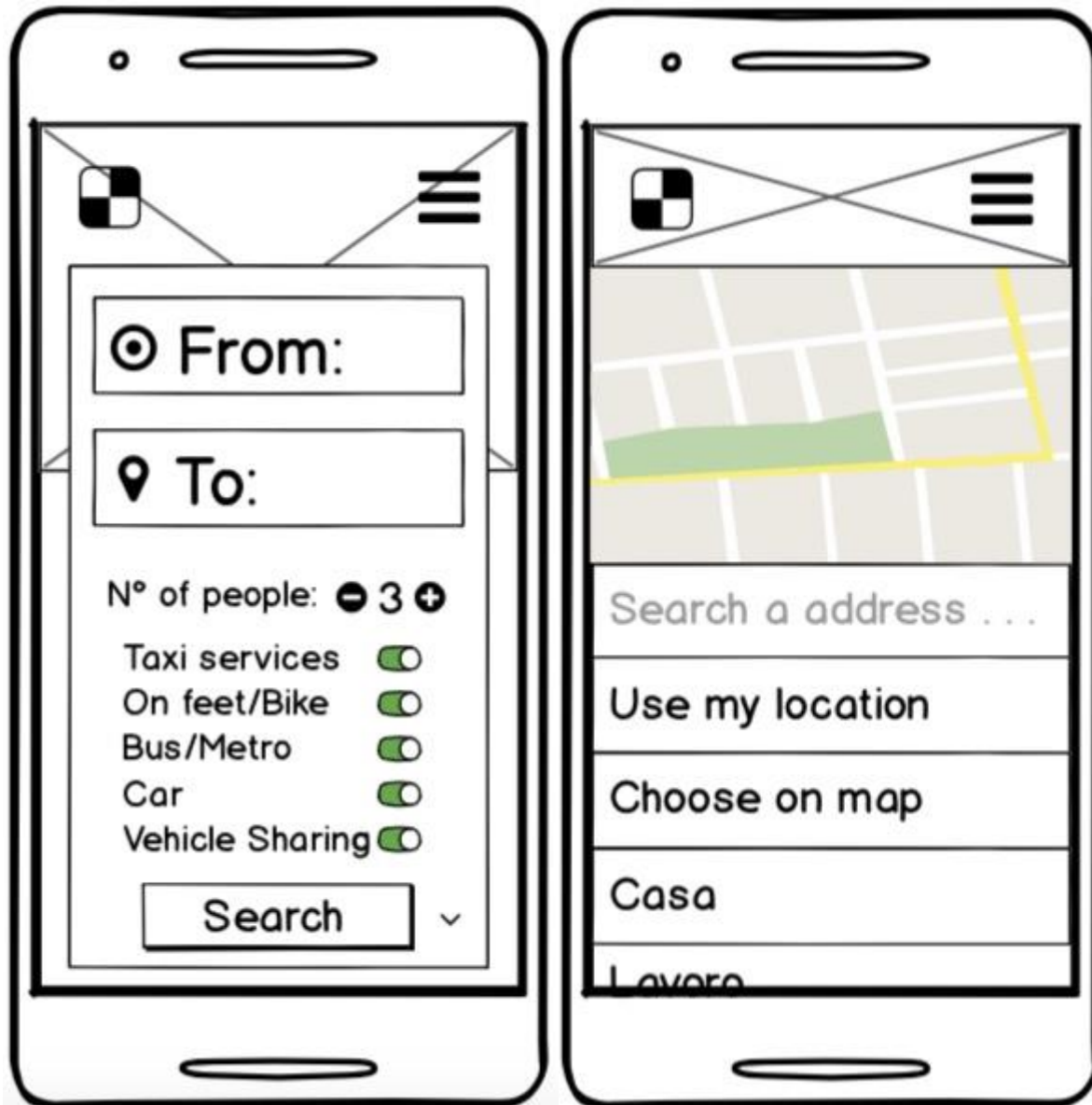


Interface 1:

From the side menu we select
search route

Interface 2:

We select a starting location
and a destination



Interface 3:

If we click on the arrow at the right side of the search button, we can filter the services that are going to appear in the results page

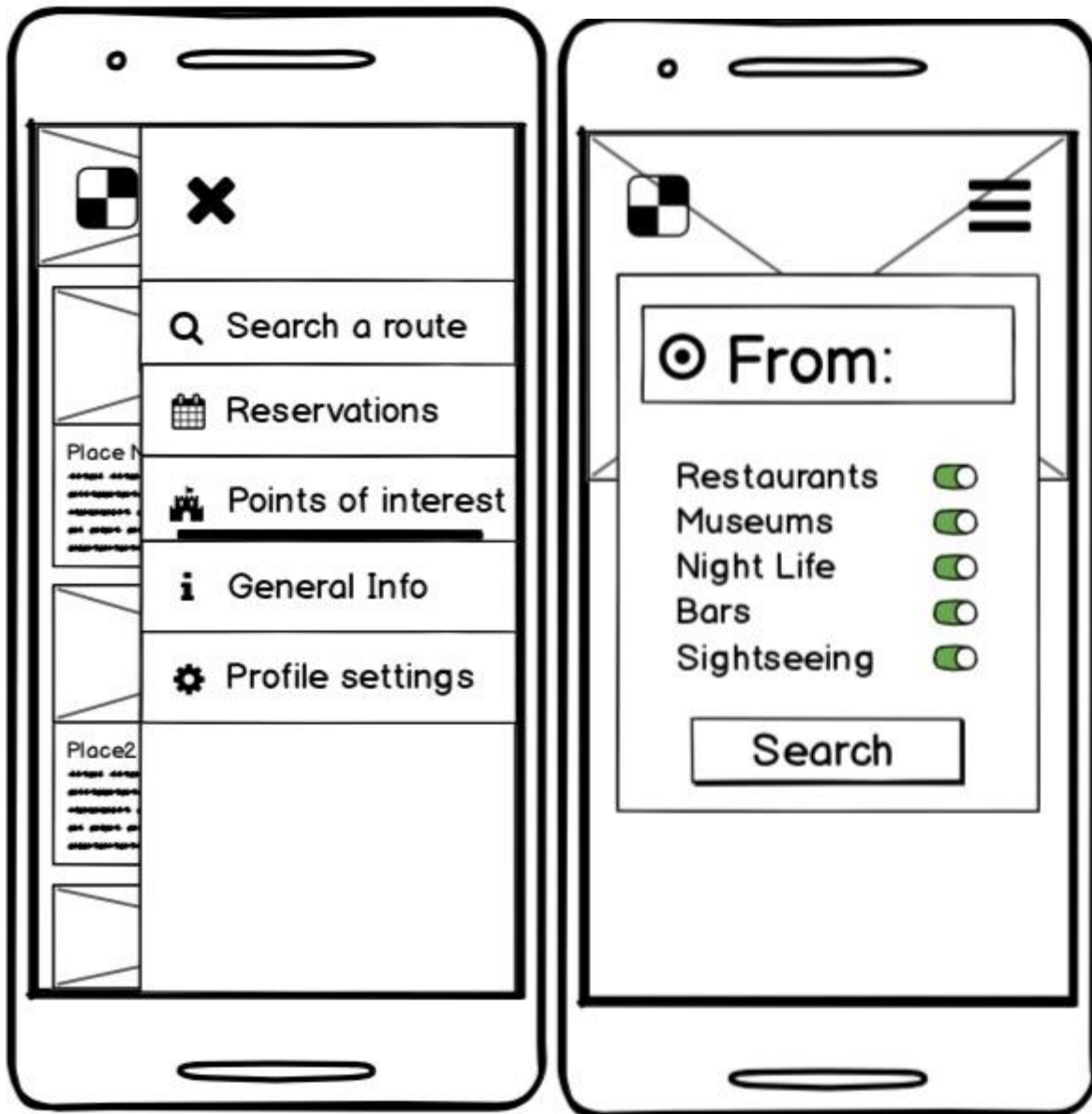
Interface 4:

Here we select the location with the possibility of typing it also you can choose it on the map or use my location button



Interface 5:

Here we have the results
organized into tabs



Interface 6:

From the side menu we select points of interest

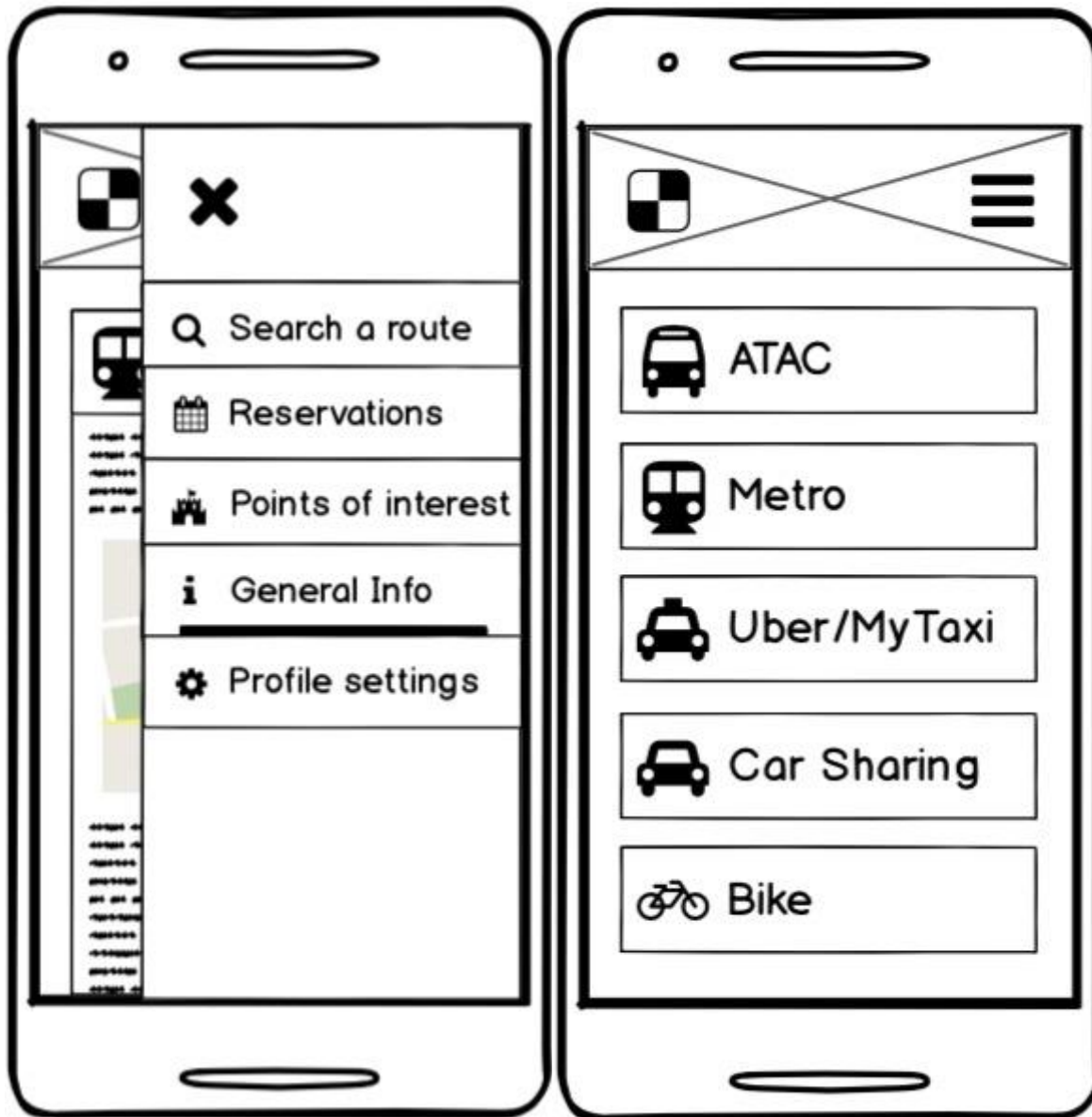
Interface 7:

Here we can choose a location starting from which we can search for a place we are interested in



Interface 8:

Here we have the result
organized also into tabs



Interface 9:

From the side menu we select
General info

Interface 10:

Here we can find any further
information regarding
transportation services



Interface 11:

An example

5.4 Heuristic evaluation

Definition:

A heuristic evaluation is a usability inspection method for computer software that helps to identify usability problems in the user interface design. It specifically involves evaluators examining the interface and judging its compliance with recognized usability principles (the "heuristics"). These evaluation methods are now widely taught and practiced in the new media sector, where UIs are often designed in a short space of time on a budget that may restrict the amount of money available to provide for other types of interface testing. The main goal of heuristic evaluations is to identify any problems associated with the design of user interfaces. Usability consultant Jakob Nielsen developed this method on the basis of several years of experience in teaching and consulting about usability engineering. The heuristics as published in Nielsen's book *Usability Engineering* are as follows:

1. **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. **Match between system and the real world:** The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6 Recognition rather than recall: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7 Flexibility and efficiency of use: Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8 Aesthetic and minimalist design: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9 Help users recognize, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. 10 Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

10 Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Heuristic evaluation:

Evaluator: Valeria Mirabella

#	Frame	Heuristic violated	Severity	Description / Comment
1	Interface page 2	Recognition rather than recall	2	Minimize the user's memory load by adding in the first page some information about the app
2	Interface page 2	Recognition rather than recall	3	How to use the "advances search" is not visible enough
3	Interface page 5	Aesthetic and minimalistic design	3	Is the map useful? It seems the have to "open map" to see the specific route
4	All	User control and freedom	3	Allow user to recover from error if he chooses by mistake the wrong action putting a back button. Moreover, it should be visible the option to be back un home page from every page.
5	Interface page 7	Match between the system and the real world	4	It seems you can indicate the number of persons only in case of search a route. In case of reservation or museum how to do it? In case you must select a route before every other kink of search it will be a problem of flexibility of use
6	All	Help and documentation	2	Provide help and documentation

Suggestions:**Suggestion 1:**

We added a description on top of search page to guide the user through the operation of searching the route.

Suggestion 2:

We deleted the arrow and make the option of filtering the services shown by default.

Suggestion3:

Map is only shown after we select “the view on map” button

Suggestion 4:

We replaced the side menu with a tab toolbar that is always shown to the user allowing him to navigate through the app, making every page reachable by just one click. Additionally, every time we are navigating through an operation, we have the back button to recover from error.

Suggestion 5:

We deleted the number of people option from search route and reservation since we thought that if we are not going to have a reservation and we are just finding either a way to reach a place or interesting places near a position, it's useless to specify the number of people. So, we are moving the number of people option to the reservation of a taxi or uber but it is also not needed here since there is not a payment into the reservation. Consequently we will remove this option.

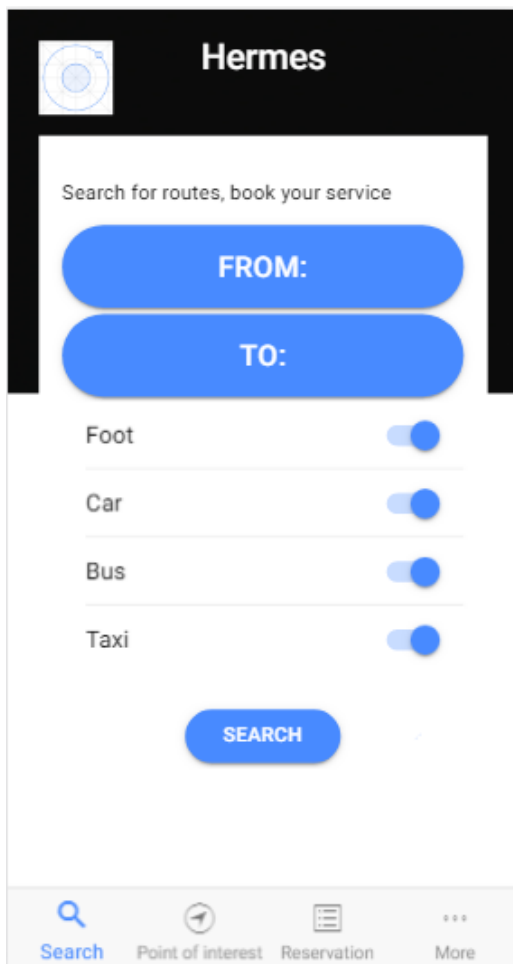
Suggestion 6:

We decided to show the user a guided slides when using app for the first time that explain every tabs and its function and there is also a button on the “more tab” that redirect the user to these slides.

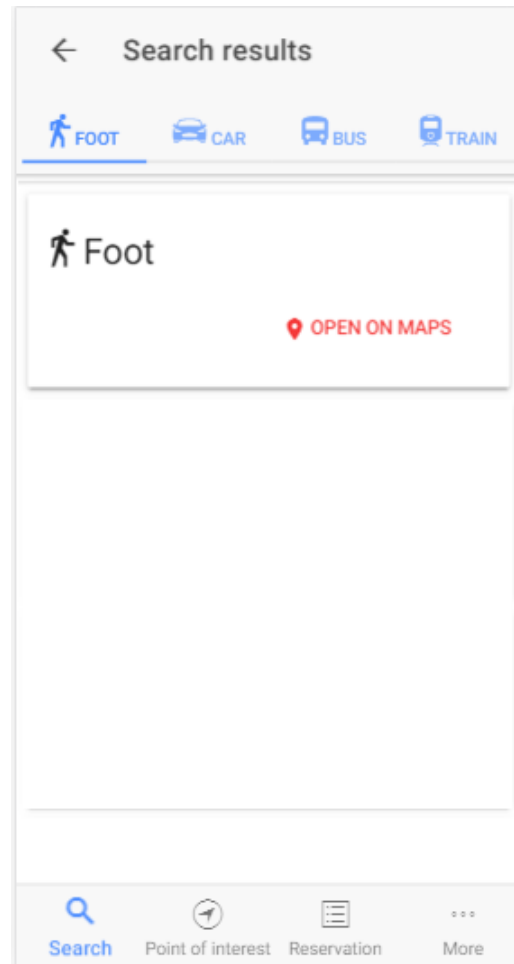
6. Prototype 2

Based on the heuristic evaluation we built the prototype 2. These are the modification we made:

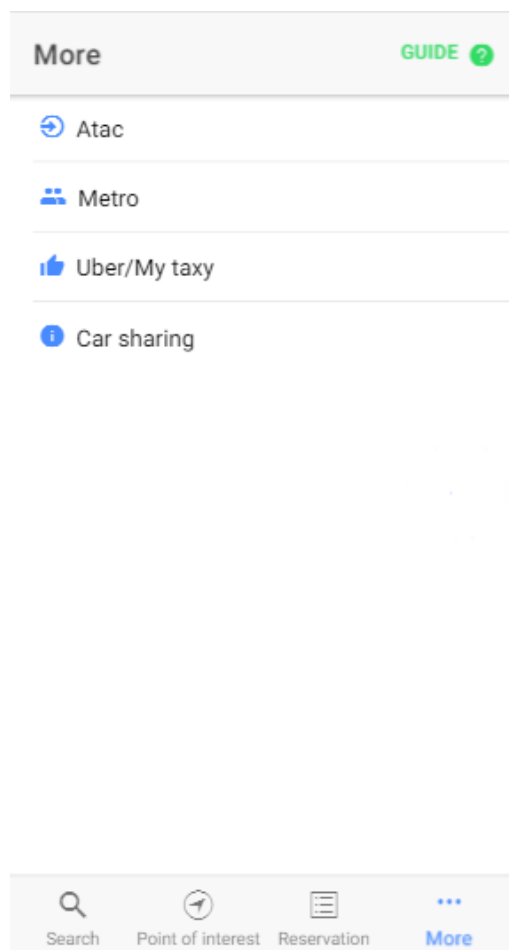
6.1 Prototype 2 description



Here we added the description of each tab to explain its function



Here it is shown that we have added the tab to improve the navigation through the app and a back button for every step to help the user recover from error



Here we added the guide button that will show an explicative slides In the final product

6.2 Think aloud

User is asked to describe what she/he is doing and why, what she/he thinks is happening, etc. After applying this technique to our prototype, we add the following features to our project.

We asked to some students to use our application and provide some feedback. These are the conclusion:

- Add geo-localization to the search page to automatically select our position to the starting location of the route
- It's better to have the results sorted by time or cost
- Add more information to the point of interest results like price range, opening and closing hours and contacts.

6.3 Anova evaluation

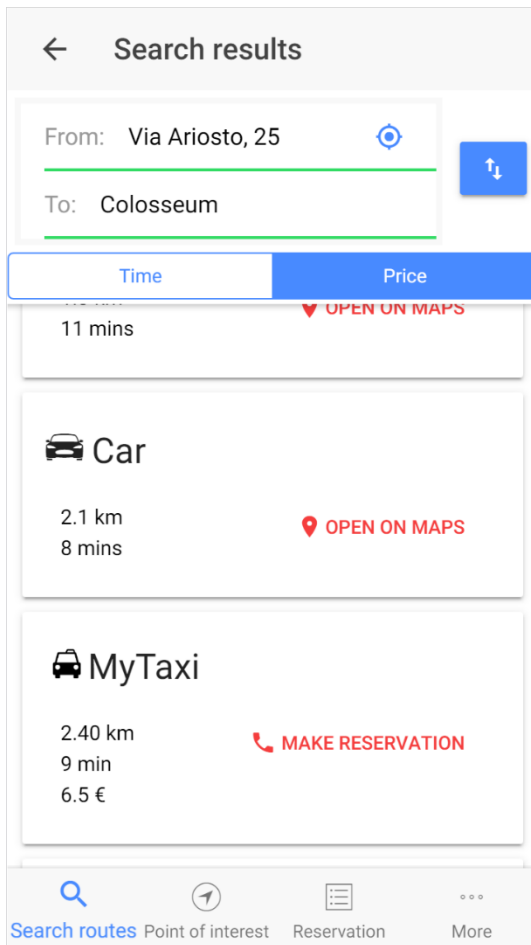
Anova is a controlled experiment in which everything is held constant except for one variable. Usually a set of data is taken for a control group, which is commonly the normal or usual state, and one or more other groups are examined, where all conditions are identical to the control group and each other except this one variable. Sometimes it's necessary to change more than one variable, but all of the experimental conditions will be controlled so that only the variables being examined change and the amount or way they change is measured. Controlled experiments are considered to be the most rigorous of empirical methods capable of providing empirical evidence to support a particular claim or hypothesis.

Participants:

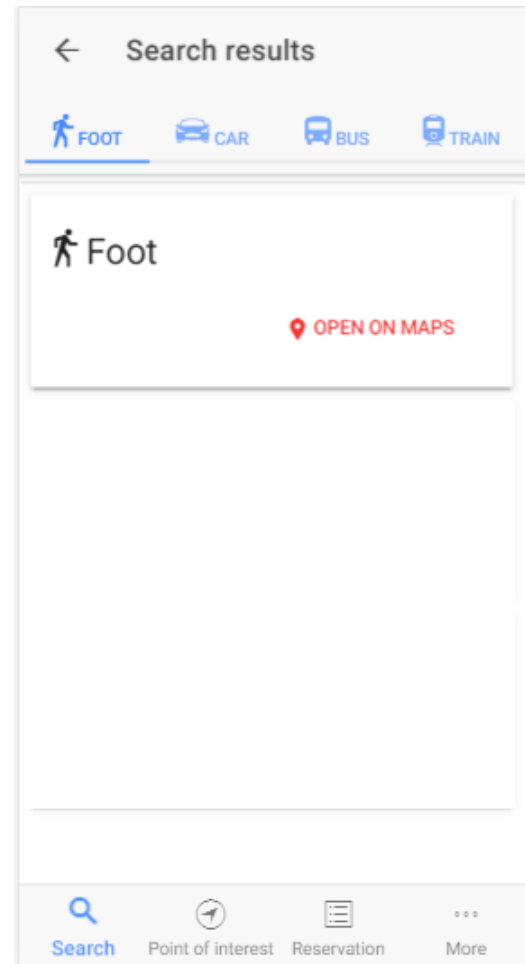
Sample of students following the defined "User Profile".

Variables:

- Independent variables: interface style 1, interface style 2
- Dependent variables: time to execute a task (in seconds)



Interface style 1



Interface style 2

Hypothesis:

A prediction of the outcome of the experiment.

- Our hypothesis: users will take less time to perform the task using application with interface style 1 than interface style 2
- Null hypothesis: there will be no difference between using application with interface style 1 and interface style 2

The task that user must perform is the following: “choose a path”. This task is based on these assumptions:

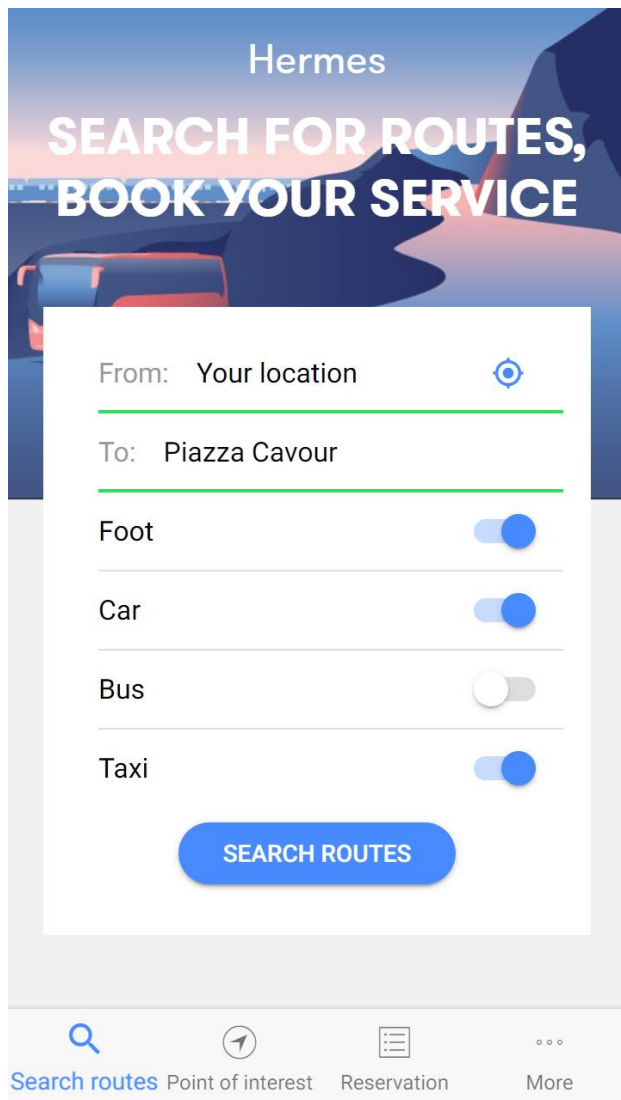
- The user has already selected starting point and destination
- The user has already selected to sort the result by time

	A	B	C	D	E	F	G
1	interface style 1	interface style 2					
2	12	30					
3	14	21					
4	10	20					
5	15	16					
6	17	23					
7	9	25					
8	13	22					
9							
10							
11							
12	Anova: Single Factor						
13							
14	SUMMARY						
15	Groups	Count	Sum	Average	Variance		
16	interface style 1	7	90	12,85714286	7,80952381		
17	interface style 2	7	157	22,42857143	18,95238095		
18							
19	Source of Variation	SS	df	MS	F	P-value	F crit
20	Between Groups	320,6428571	1	320,6428571	23,96263345	0,000368944	4,74722522
21	Within Groups	160,5714286	12	13,38095238			
22	Total	481,2142857	13				
23							

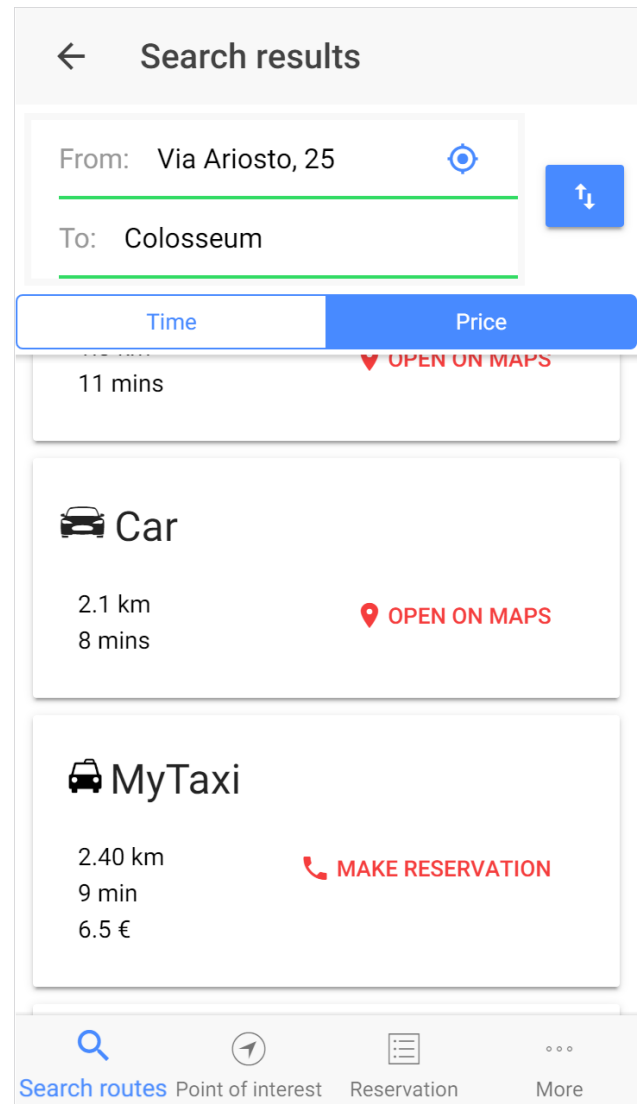
Analysis of data:

If $F > F_{critic}$ we reject the null hypothesis. In this case $23.96 > 4.74$, so we reject the null hypothesis and we proved that interface style 1 is better than interface style 2.

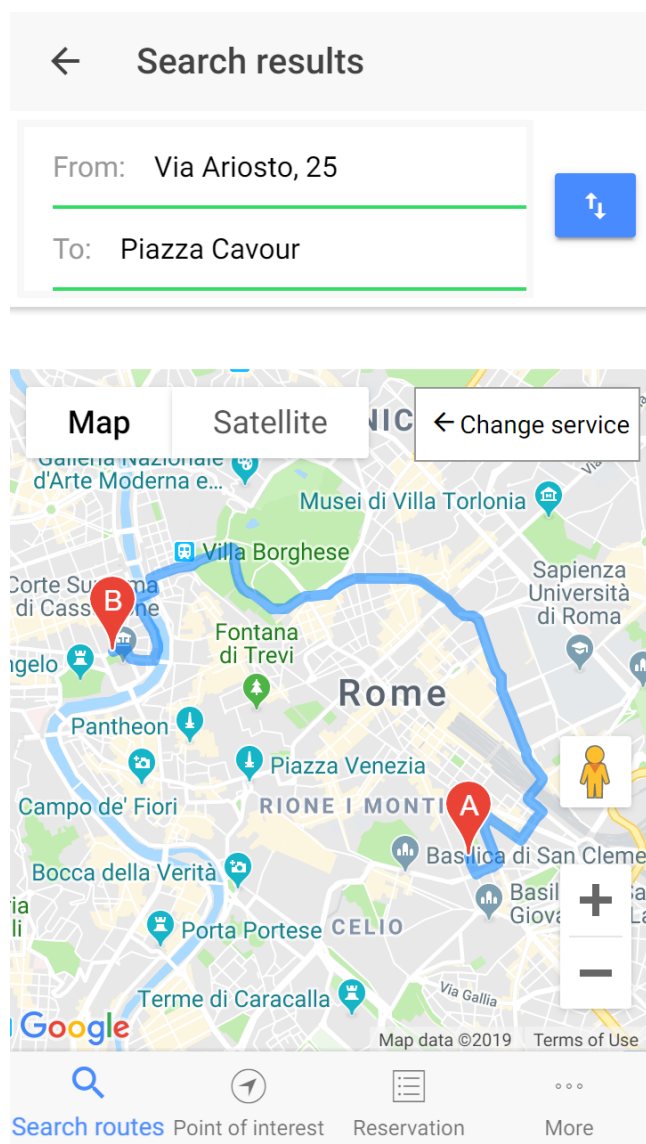
7. Final product



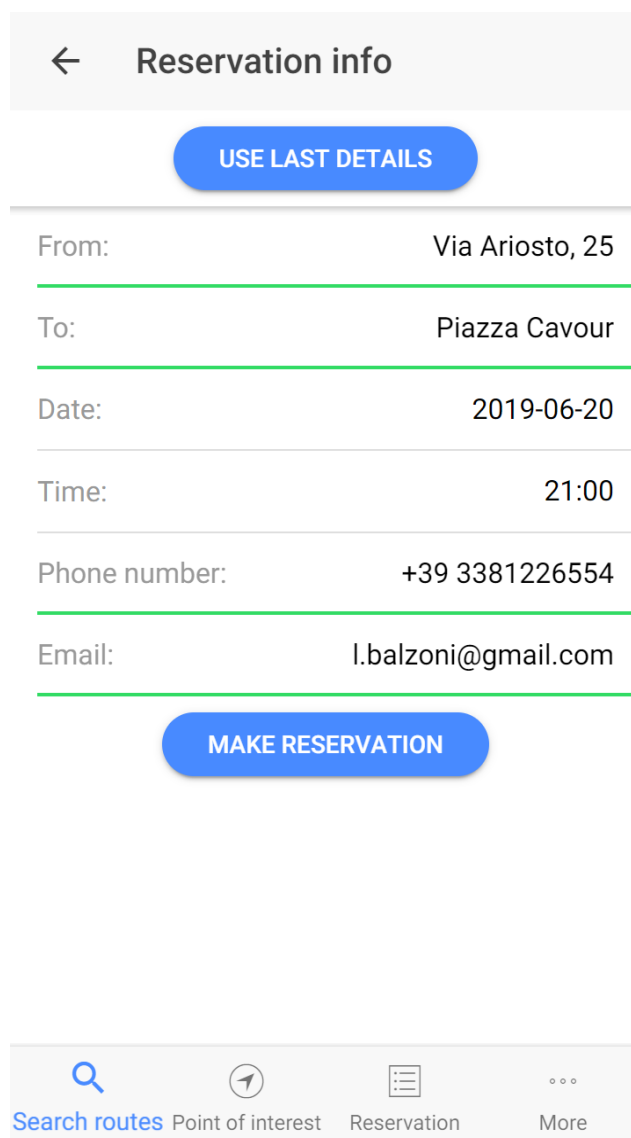
Homepage

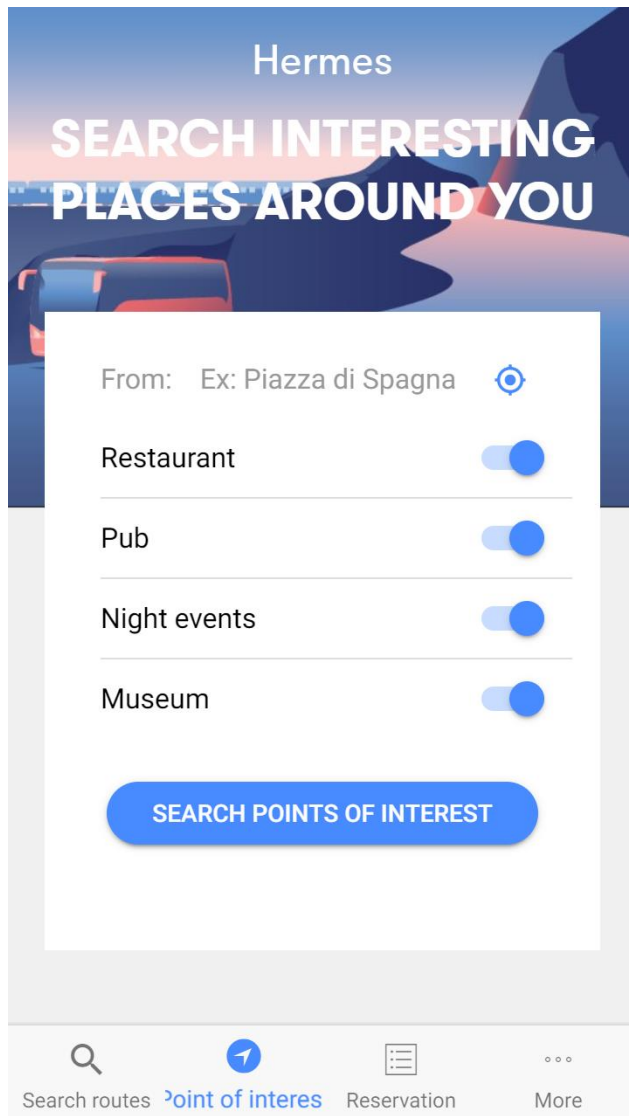


Search results

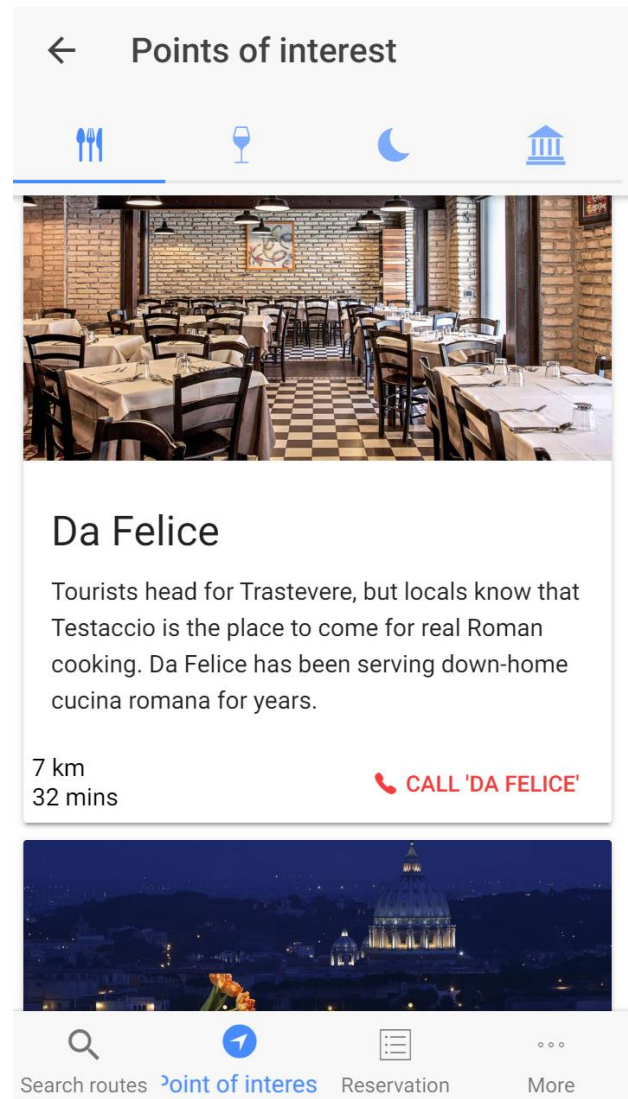


Map view

Making a
reservation




Search point of
interest



POI results


Reservations



myTaxi

From: Via Eudossiana, 18
To: Viale dei Colli Portuensi, 103
Price: €25
Date: 24 Jun 2019, 19:30





✕ DELETE RESERVATION 🌐 SHOW ON MAP



Uber


From: Via Ariosto, 25
To: Piazza Cavour
Price: €16
Date: 25 Jun 2019, 21:00


✕ DELETE RESERVATION 🌐 SHOW ON MAP


 Search routes
  Point of interest
  **Reservation**
 More


More information





Additional info GUIDE ?

 Atac

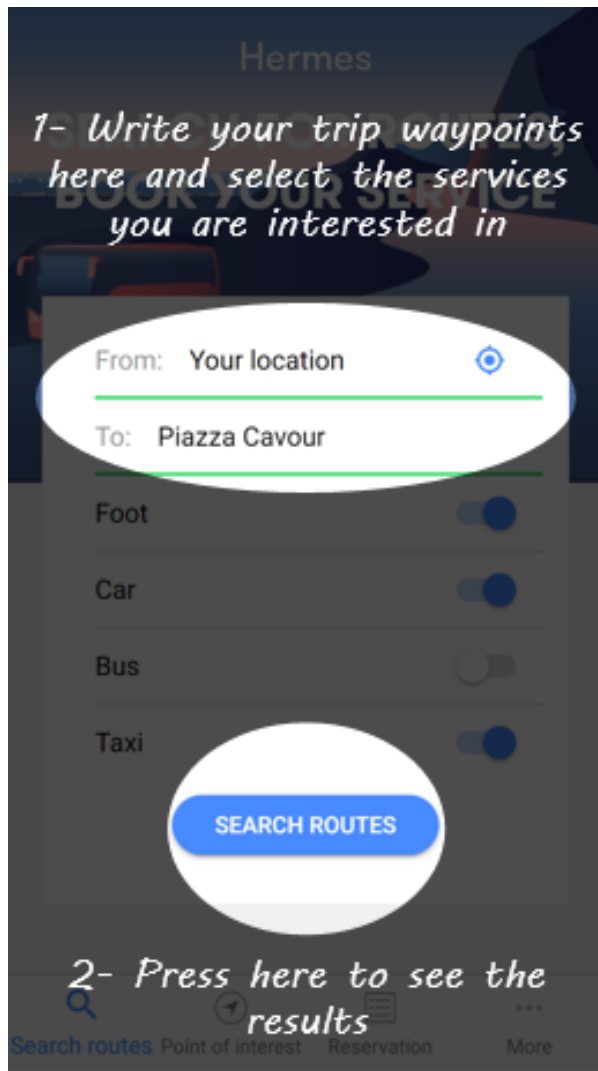
 Metro

 Uber/myTaxi

 Car sharing

 Search routes
  Point of interest
  Reservation
  **More**

Reservation list



Guided slides

8. Conclusion & future works

During this work we were able to implement all the user requirements gathered. The application is developed in such a way to enhance the user experience keeping the user always at the centre. This was done by carefully following principles and rules explained during the lectures. While completing the implementation of the application we found some new user requirements that we consider as future works:

- Saving of minor user preferences such as default sorting option, active service in searching route and different kind of places in point of interest
- Add a notification system
- Further functionality on the reservations