Final Report Group 08

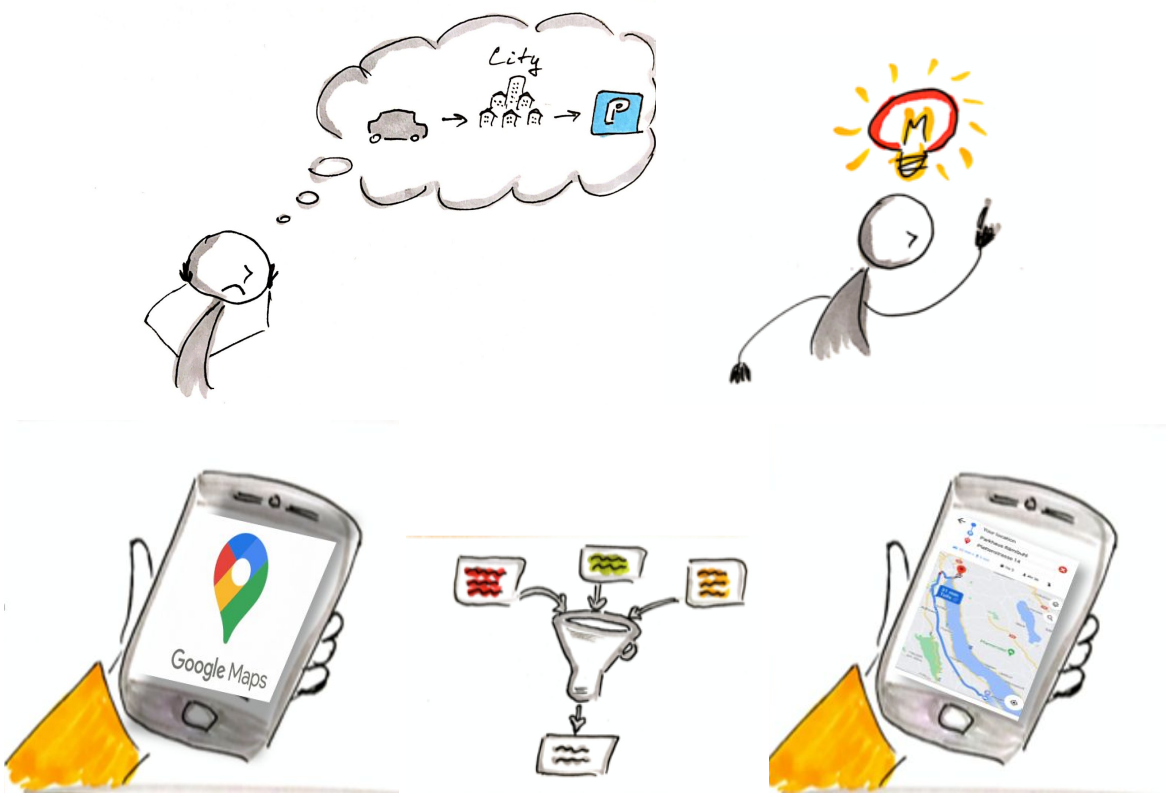
Olga Bespalova, Zoia Katashinskaia, Iuliia Mozhina, Alexander Theus, Natalia Werner, Debora Beuret

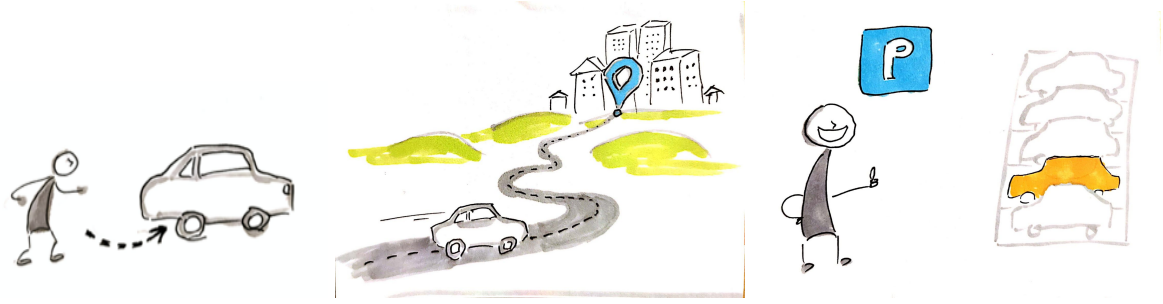
Design Problem

Our user group is car drivers in Switzerland who use Google Maps. We discovered that popular navigation devices have limited functionality when it comes to searching for a parking slot and Google Maps is no exception. In order to find a parking space the user has to either use the navigation twice (first by looking for the destination, then manually searching for parking next to destination) or add an intermediate stop towards the destination, which is difficult to find. Both ways are cumbersome and the displayed results are far from perfect: they don't show parking next to the destination, but display generic results of parking slots in the much larger area, they are incomplete, and last but not least, they don't offer any filtering options. Our design aims to provide a solution to these limitations by supporting the activity of finding a parking spot near the destination that best fits the user's criteria and to add it to the route before the destination.

Usage Scenario

The following user scenario is representative of the typical scenario that our application is designed for. Our main focus lies on the parking search, providing information on parking and filtering options. This differs from the user scenario presented in the video, which additionally features the "reserve parking" functionality, which was added to our design at the later stage and wasn't tested by any of the test users.





Our user Bob is about to start his journey to the university cafeteria at Plattenstrasse 14 in Zürich where he will go by car and he wants to search in advance for a parking option at the destination. He wishes to park his car in the indoor parking for 3 hours. Additionally, he wants to pay cashless. Since being a typical representative of our user group, Bob uses Google Maps. He enters the address and decides to use the new feature of Google Maps that allows setting the parking options at the destination as an intermediate step between "Your location" and the final destination of "Plattenstrasse 14". (For the screenshots to each screen walkthrough please see the Appendix A)

Screen 1 (first screen)
Bob presses the "Add parking near destination" button.
Screen 2 (search results)
Bob sees that the proposed parking is called "Parkplatz Schönleinstr" and he looks down at the parking profile icon with the same parking name displayed at the bottom of the screen. He wants to see more information about the proposed parking and clicks on that icon.
Screen 3 (Parkplatz Schönleinstr)
From the photo, he immediately sees that it is outdoor parking ("Blue Zone"), which is also confirmed down in the description. Moreover, Bob looks further down and sees that the maximal duration of stay is only 1 hour. He wants to go back to the previous screen and sees the arrow in the top left corner and clicks on it.
Screen 4 (back to search results)
Now Bob wants to use filters to find indoor parking. He finds an icon for that in the top left corner and clicks on it.
Screen 5 (filters)
Bob finds the filtering by "Type" option and chooses "Indoor". He also wants to specify that their car is of non-standard size and clicks on "Maximum height of car"
Screen 6 (filters - car height)
Bob clicks on the black arrow for the drop-down list and chooses height "2.10 m". Then Bob clicks on the "Done" button.
Screen 7 (back to filters)
Bob checks the filters he set before and clicks on the "Apply" button.
Screen 8 (search results after filters)
Bob sees that there is just one result "Parkhaus Rämibühl" with an indoor parking sign next to the name in the parking profile. Bob is also interested in the price identifier and clicks on the parking profile icon down the screen for more information.
Screen 9 ("Parkhaus Rämibühl")
In the "Tarif" section Bob clicks on the black arrow next to it to see more details about the price. He is satisfied with the price and clicks again on the same arrow next to "Tarif" to wrap down the information. Bob sees the "ABOUT" tab and decides to go there by clicking on its name.

Screen 10 (“About”)
Bob goes through the given information and sees the desired Payment section. He wants to go back to the previous screen and clicks on the arrow in the left above corner to wrap down the “About” section
Screen 11 (back to “Parkhaus Rämibühl” main)
Bob is satisfied with the parking option he found and clicks on “ADD AS PARKING”
Screen 12 (modified start screen)
Bob sees that the intermediate step is added to the route, and the app will navigate him to the parking place first and then to the final destination.

Design Highlights

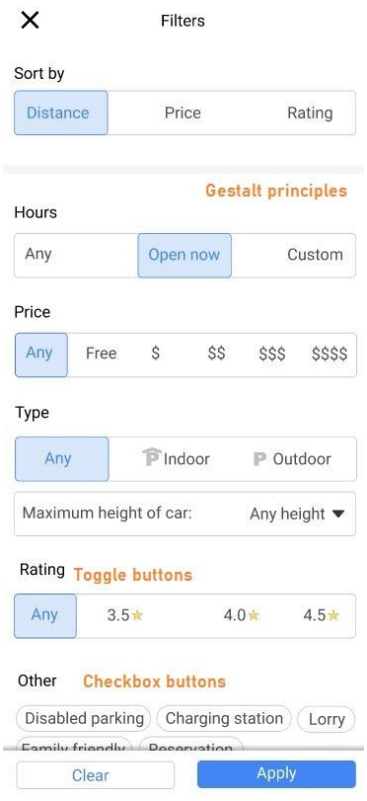
Since our design is an extension of Google Maps, we largely kept Google’s own design principles. One such example is the filter screen: the rectangular buttons are *toggle buttons* (only one can be clicked) and the round buttons are *checkboxes* (in the “other” section, multiple options such as “disabled parking” and “family friendly” can be clicked at once). *Visual feedback* indicates the selected options (blue vs white). But we also introduced some changes to Google’s design, which can also be seen on the filter screen. Firstly, we added icons where possible to provide quick *visual cues*. Secondly, in order to *minimize interactions* with the users, most of the buttons are toggle buttons and by default set to “any”. This means that the user has to interact only if he or she is after particular results, for example if they want to see parkings only with the highest ratings. Thirdly, the expert user, with whom we were testing our prototype mentioned that it can be confusing for users to put sort and filter results on the same screen. That’s why we decided to use *Gestalt principles* in order to differentiate between two groups of objects on this screen, namely between sorting and filtering features. Having the knowledge that perceptual grouping helps chunking visual information, we’ve used the principle of *proximity:* left some place between sorting and filtering options, so that filtering buttons are located near each other. We also used the principle of *closure* by putting a line between sorting and filtering features. This line is barely visible to avoid *visual cluttering* - see screenshot 1.

Another consideration was the scenario where users will use our product. Many users will probably be driving or be in a hurry, that is why we had to think about *minimizing the extraneous information (noise*) in the parking profile screen to *increase the signal- to- noise ratio*. In order to do that we kept only the most important information: i.e. walking distance to destination, maximum parking time and price, and took away photos.

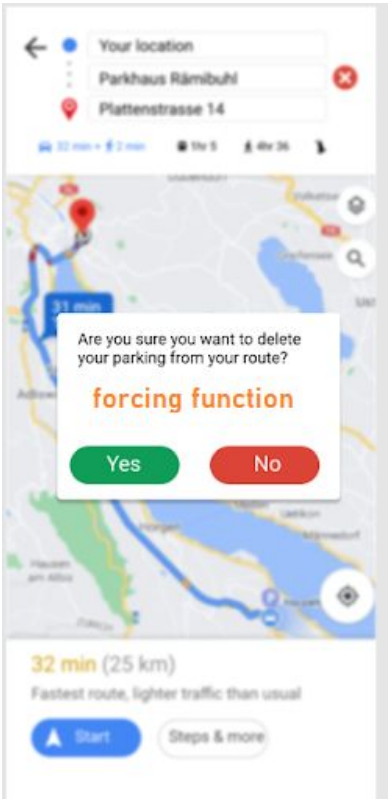
To prevent a user from accidentally deleting a parking slot near their destination we implemented a *forcing function*, namely the *lock-out* that asks, if the user is sure that he/she wants to delete parking from his/her route. -see screenshot 2

One of the most important elements of our design is the button “Add parking near destination”. Its main purpose is to decrease the amount of interactions that the user has with Google Maps to achieve the goal of finding parking. In our prototype the user can click on *Add Parking to Destination → Add Parking* and the parking will be added to a route, whereas in Google Maps the user has to take many additional steps to achieve it. We first made this button very prominent and placed it at the bottom of the screen, right next to Google's “Start” button. However, even though this design greatly enhanced *discoverability*

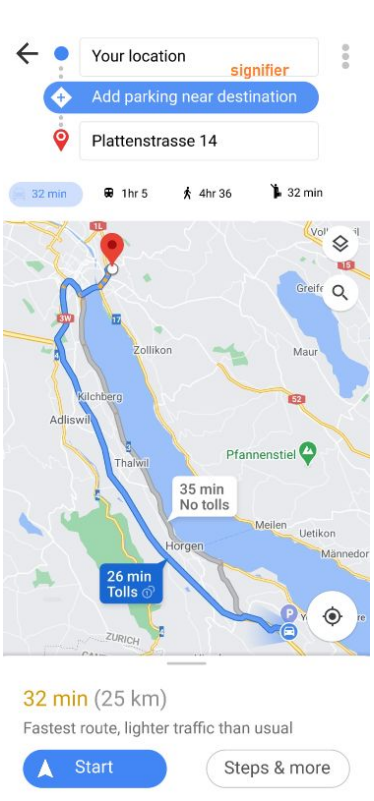
(all tested users had no problem finding and understanding it) and followed the *Fitt's Law by* minimizing the distance to the "Start" button, after much discussion, we settled on a different design. We decided that it shouldn't be that prominent, because not all drivers are in search of parking and we didn't want to disturb the *visual hierarchy* of Google Maps. We feel that moving it to the top of the screen still allows for good discoverability and keeping in with Google's design and making it blue turns it into a good *signifier.* It also follows the logical order of steps, which helps the user with building the correct *mental model*: the user has to leave the location, park the car and reach their destination. -see screenshot 3



Screenshot 1



Screenshot 2



Screenshot 3

Prototypes and Evaluation

Iteration 1

Prototyping method

The very first low-fidelity iteration of our prototype was carried out in Google diagram software drawio. It allowed us to transfer high-level design concepts into a digital form and think through the interaction flow of a new application feature. We created a coarse interpretation of our prototype in order to have a basis to discuss and to develop into a more detailed version. Application functionality and basic interaction logic were more important at this stage than the visual appearance of the product. The first iteration of our prototype can be accessed using the following link or in the appendix.

Prototype

This first iteration was used mainly within our team: we used it to discuss the basic ideas, to decide which we wanted to develop further and to see if we all agree on their implementation. It also allowed us to roughly estimate the workload for each step of the design process in later iterations. Once the first iteration was done, we also showed it to our TAs during a coaching session and got feedback from them.

Learnings

At this stage it quickly became clear to us that the functionality of searching for a parking slot and implementing a filter was complex enough in itself and allows enough depth to use it as a main functionality of our project. Therefore we dropped the functionality of navigation in multi storey parking houses, ‘find your parked car’ feature and paying for parking.

Iteration 2

Prototyping method

For the second iteration of our prototype we decided to switch to another prototyping software: Figma. There we could create a visually rich prototype with complex interaction that simulated real-life product usage. The main focus of this iteration was the breadth and depth of our application as well as visual design. The second iteration of our prototype can be accessed using the following link or in the appendix.

Prototype usage

With this prototype at hand we could conduct a series of user- and expert-based tests. To make the usability testing, we made a choice in favour of a summative test as our goal was to evaluate the effectiveness of specific design choices. We wanted to be sure that the main functionalities of our prototype (Add parking as a destination and using filters) are easily discoverable for users that see our application for the first time. In order to do that we conducted a think-aloud test with 2 users. After we made some improvements in our prototype following the user feedback, we also tested it with one expert user (a product designer) using a cognitive walkthrough method. On top of that, this iteration was discussed with Prof. Dr. Wacharamanotham, who gave us a rich feedback concerning our design choices.

Learnings

A few important take-aways could be gathered which would flow into our subsequent prototype. For one, we learned that certain buttons were not conclusive and left the user wondering what the functionality behind them was. One example was the “reserve” button which was at first placed quite high in the hierarchy. Similarly, we changed the hierarchy of the button “Add parking”, which was initially placed to the right of the “start” button in the

Google Map navigation. Following the feedback from our expert-user, who pointed out that placing the “Add parking” button in such a prominent position for a mobile device could interfere with the main Google Maps functionality, we moved the button to the top of the screen. That was implemented in the next iteration.
Furthermore, we observed that certain information that was shown in the parking profile on the map was not relevant for all users, namely the picture of the parking spot. This aspect was also pinpointed by Prof. Dr. Wacharamanotham and he suggested to reconsider the structure of parking information. Subsequently, we decided to redesign the parking profile that is shown on the map for further iterations.
On top of that we decided to proceed with implementing an experimental feature of the reservation of a parking slot for the following iteration.

Iteration 3

Prototyping method

We continued to work in *Figma* because it allowed us to continuously work with the existing prototype as well as improve it’s interactivity and polish the existing design. In this iteration we included or otherwise addressed the feedback that we got during usability testing of the second iteration. The third iteration of our prototype can be accessed using the following link or in the appendix.

Prototype usage

We decided not to test the third iteration of the prototype with users and discussed it within our group only. At this step we wanted to be sure that we considered all design and psychology principles learned in this lecture and that the feedback of our users was properly implemented.

Learnings

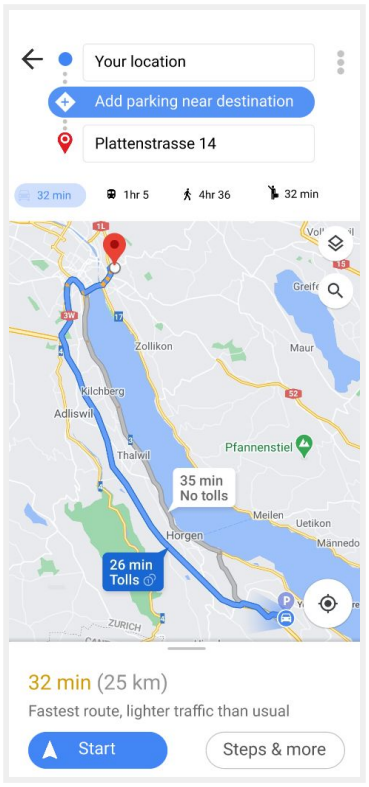
We’ve recalled all the suggestions and comments after testing the prototype with users during the second iteration and made some changes in it. The most crucial one was changing the design of the button “Add parking near destination” (after testing it with an expert-user) and putting it on the top of the screen. Furthermore the parking profile shown on the map was completely redesigned and thought over to only show the most important information to the user. Generally, at this stage of design we learned how to understand users’ needs better and to effectively implement them in the prototype.

Limitations

We had to limit our design in two fundamental ways. Firstly, we had to decide in which environment the activity we intend to support shall be implemented. For one, we could have created a separate parking application, that would aid the finding of an ideal parking spot. When the interaction would be finished, that is, when the user has found an ideal parking spot, the app would link to Google Maps and configure a route to the chosen parking. However, this option would require users to use an additional application which our users were not so keen on. Furthermore, this option would not have allowed to meaningfully integrate the parking spot into a route, but would have just prompted Google Maps to start a route with the parking location as the destination. Another option would be to integrate it within a navigation system to allow for a uniform and continuous experience. This option was favoured by our users and thus we decided to implement our solution as an extension to Google Maps. Due to this decision car drivers who do not use Google Maps would be excluded.

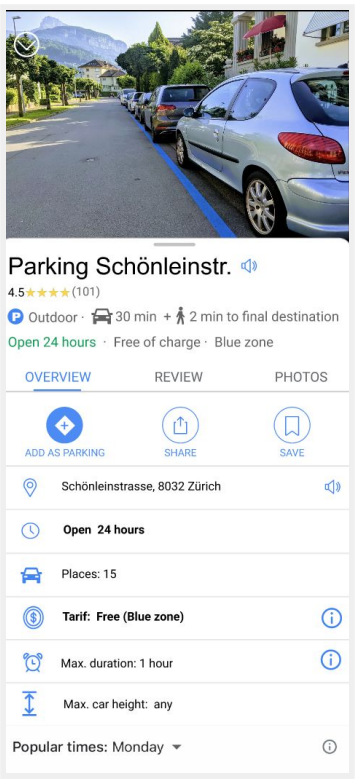
Another fundamental way in which we had to limit our design is in how users would approach the parking feature. For one, finding a parking spot can be done in an asynchronous or synchronous way. If the user were to approach it asynchronously, they would search for a parking spot before wanting to navigate to the final destination; in the synchronous approach they would look for a parking spot shortly before or at the time of wanting to start the route. These two options are not mutually exclusive, and technically could both be implemented in Google Maps. However, as for the time being, Google Maps for car drivers is intended to be used in a synchronous manner and is designed as such. Allowing for an asynchronous user interaction would require major changes in Google Maps. The synchronous approach, however, is more in line with how Google Maps is supposed to be used right now, which is why we intended to implement it as such.

Appendix A: Screenshots to Usage scenario

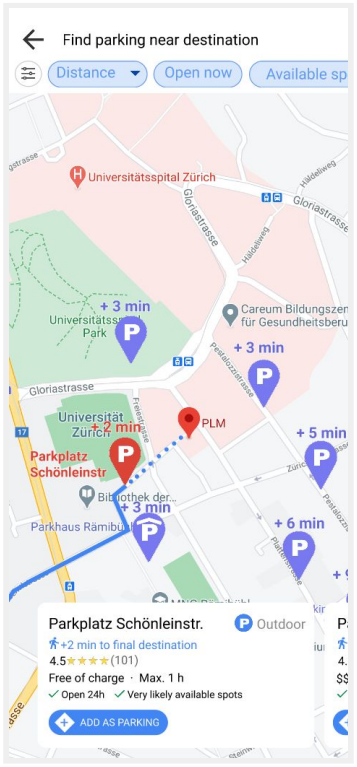


Screen 1 (first screen)

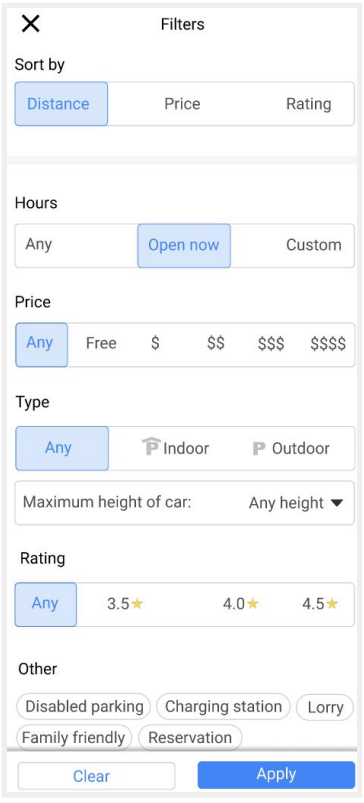
Screen 2 (search results)



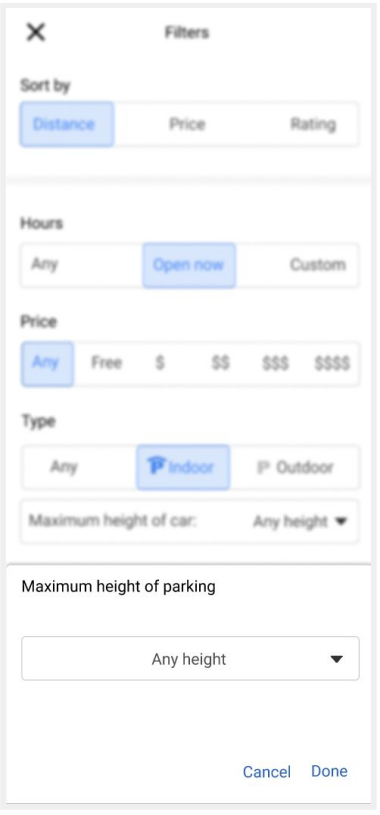
Screen 3 (Parkplatz Schönleinstr)



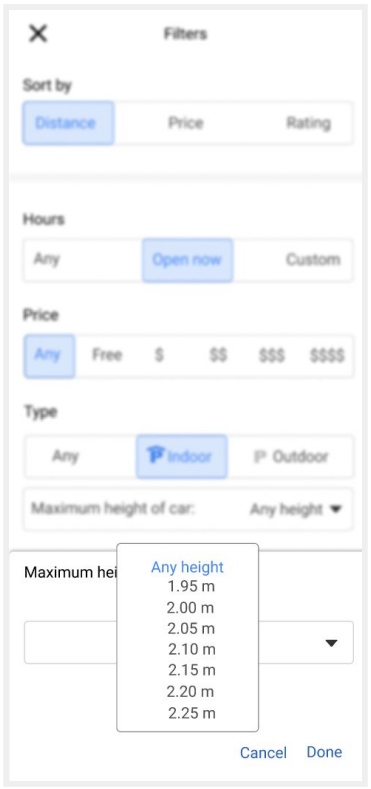
Screen 4 (back to search results)



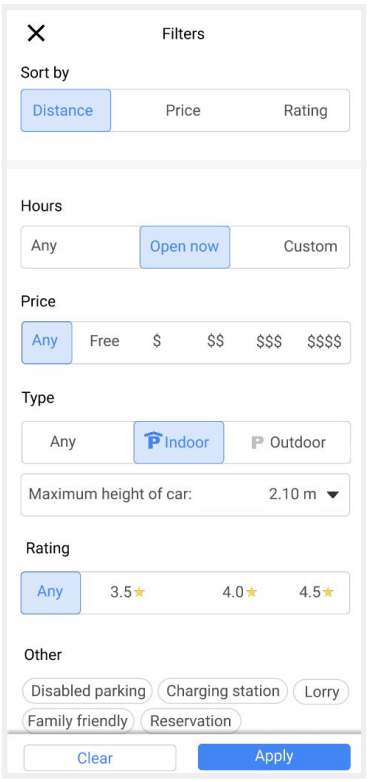
Screen 5 (filters)



Screen 6 (filters - car height) Part 1

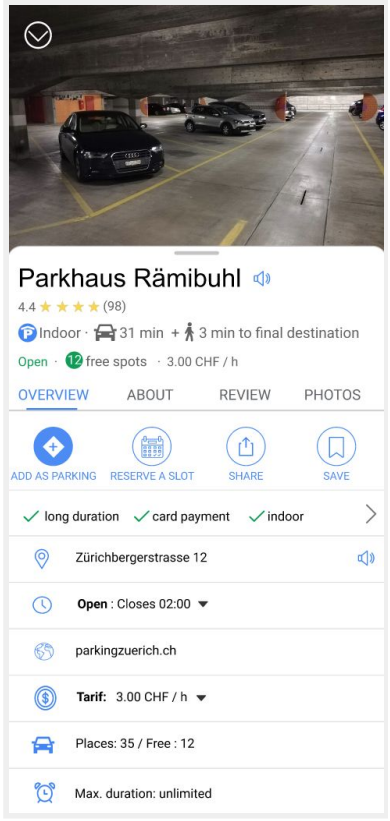


Screen 6 (filters - car height) Part 2

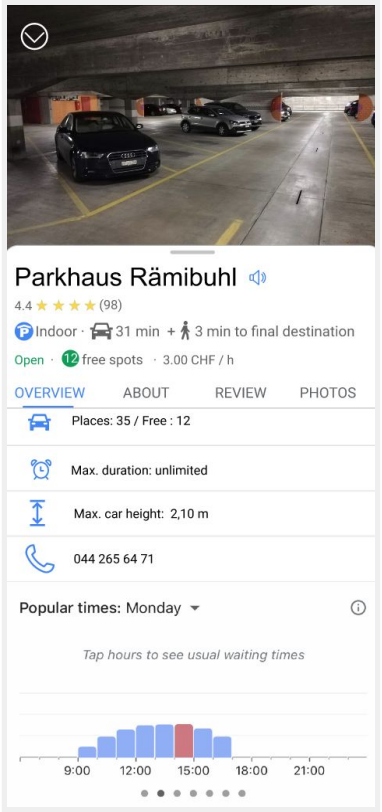


Screen 7 (back to filters)

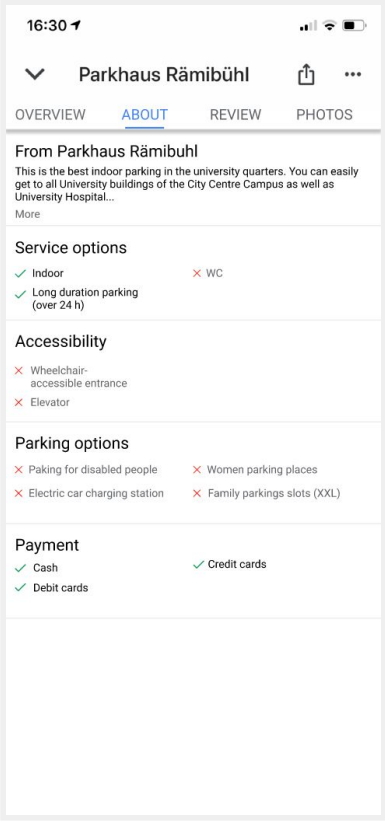
Screen 8 (search results after filters)



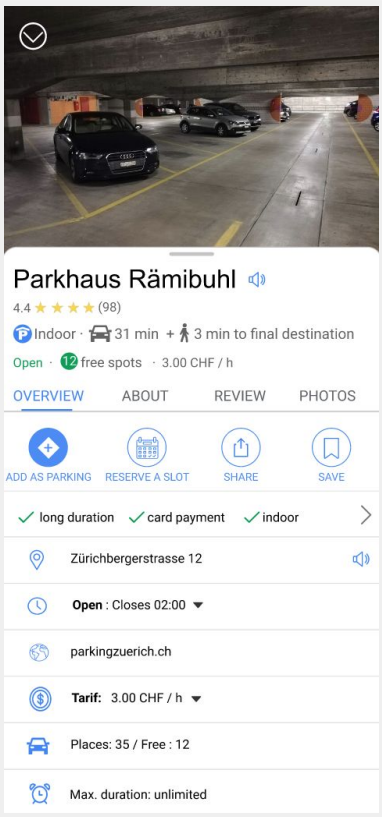
Screen 9 (“Parkhaus Rämibühl”) Part 1



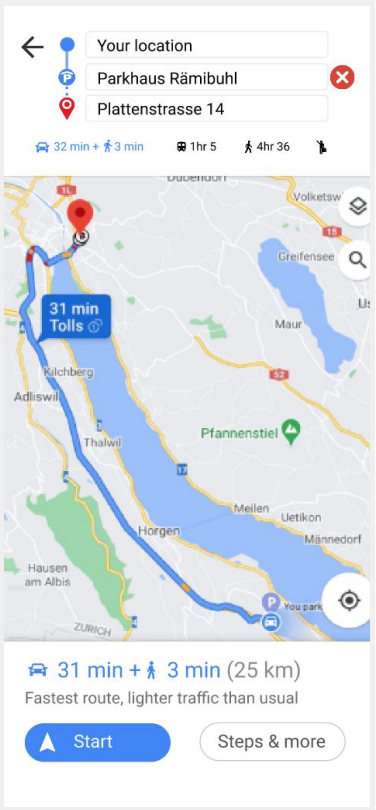
Screen 9 (“Parkhaus Rämibühl”) Part 2



Screen 10 (“About”)



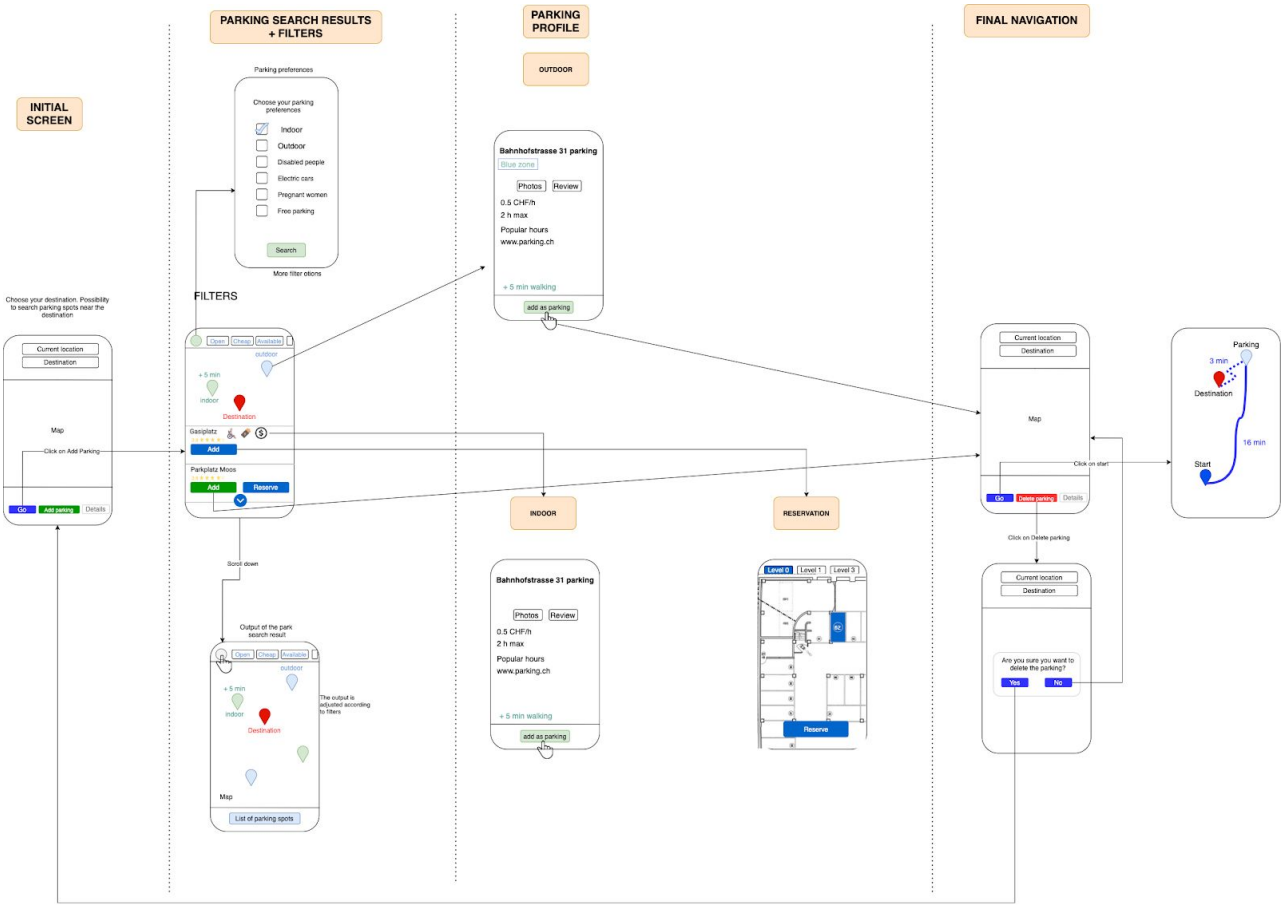
Screen 11 (back to "Parkhaus Rämibühl" main)



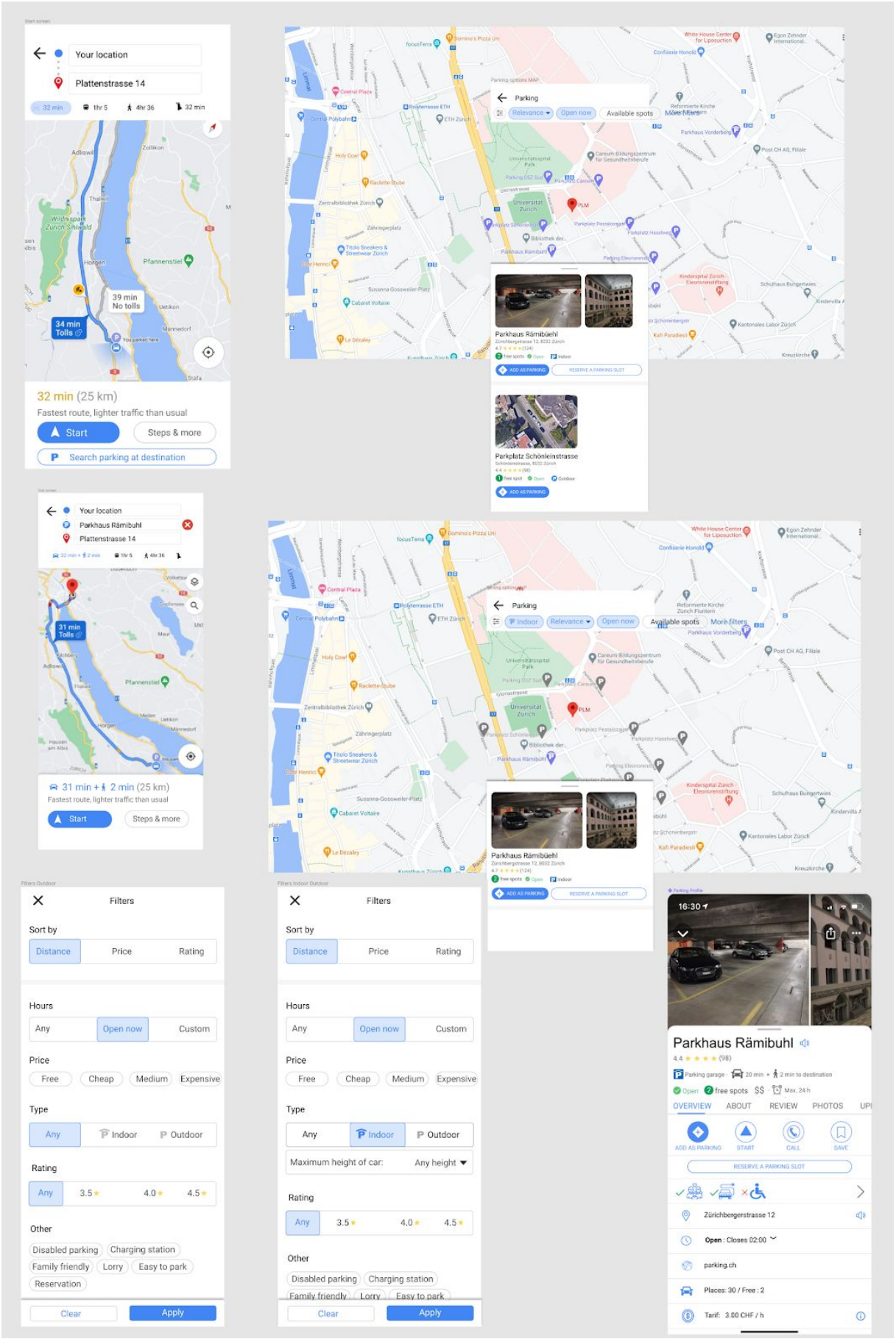
Screen 12 (modified start screen)

Appendix B: Prototype

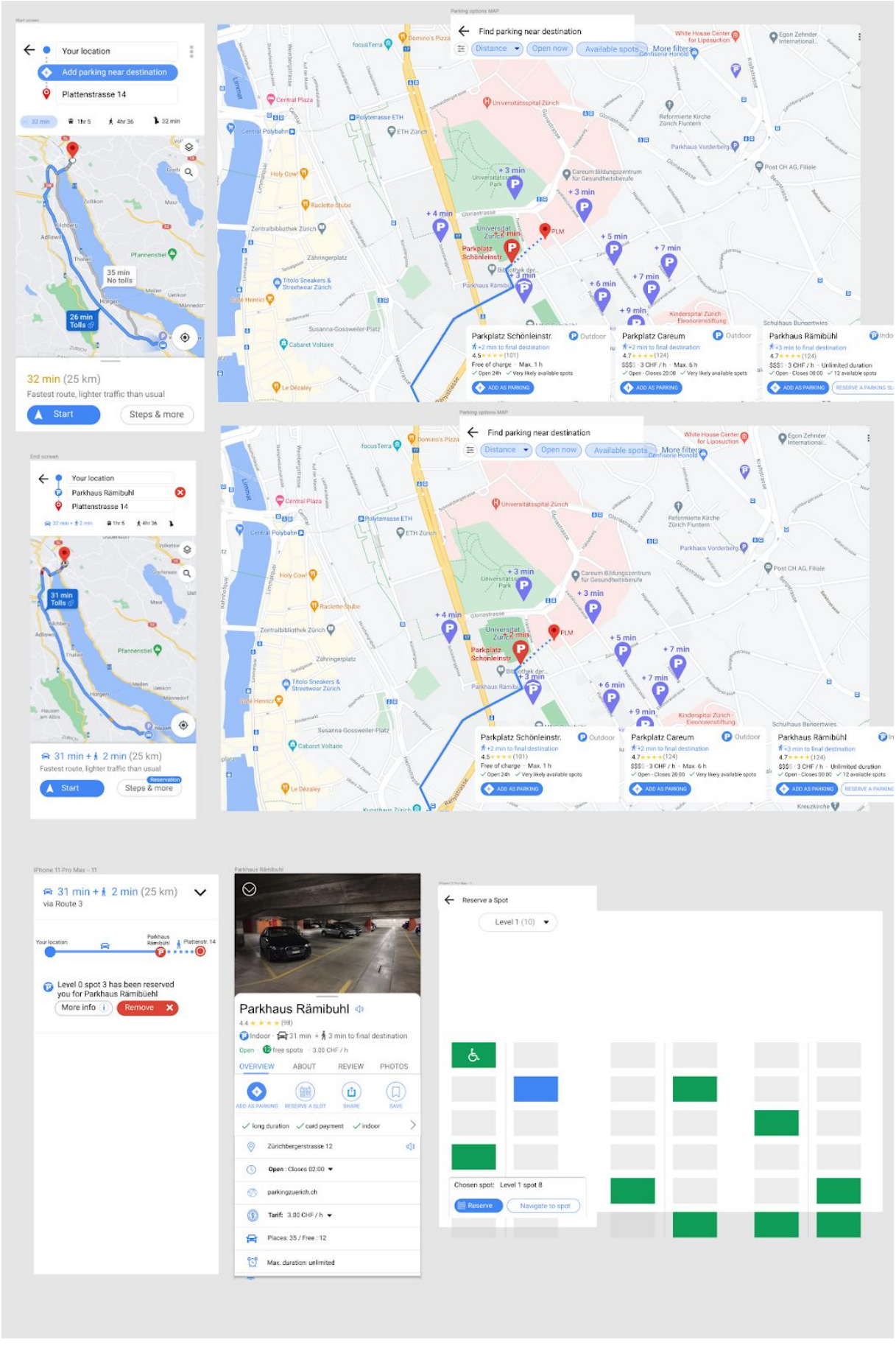
Iteration 1



Iteration 2


Your location
Plattenstrasse 14
32 min (25 km)
Fastest route, lighter traffic than usual
Start
Steps & more
Search parking at destination
39 min
No tolls
34 min
Tolls
Parking
Relevance
Open now
Available spots
More filters
Parkhaus Rämibühl
Parkplatz Schönleinstrasse
Your location
Parkhaus Rämibühl
Plattenstrasse 14
31 min
Tolls
31 min + 2 min (25 km)
Fastest route, lighter traffic than usual
Start
Steps & more
Parking
Indoor
Relevance
Open now
Available spots
More filters
Parkhaus Rämibühl
Filters
Sort by
Distance
Price
Rating
Hours
Any
Open now
Custom
Price
Free
Cheap
Medium
Expensive
Type
Any
Indoor
Outdoor
Rating
Any
3.5
4.0
4.5
Other
Disabled parking
Charging station
Family friendly
Lorry
Easy to park
Reservation
Clear
Apply
Filters
Sort by
Distance
Price
Rating
Hours
Any
Open now
Custom
Price
Free
Cheap
Medium
Expensive
Type
Any
Indoor
Outdoor
Maximum height of car:
Any height
Rating
Any
3.5
4.0
4.5
Other
Disabled parking
Charging station
Family friendly
Lorry
Easy to park
Clear
Apply
16:30
Parkhaus Rämibuhl
4.4 (98)
Parking garage · 20 min · 2 min to destination
Open · 2 free spots · $$ · Max. 24 h
OVERVIEW
ABOUT
REVIEW
PHOTOS
ADD AS PARKING
START
CALL
SAVE
RESERVE A PARKING SLOT
Zürichbergerstrasse 12
Open : Closes 02:00
parking.ch
Places: 30 / Free : 2
Tarif: 3.00 CHF / h

Iteration 3


Your location
Add parking near destination
Plattenstrasse 14
32 min (25 km)
Fastest route, lighter traffic than usual
Start
Steps & more
35 min
No tolls
26 min
Tolls
Find parking near destination
Distance
Open now
Available spots
More filters
Parkplatz Schönleinstr.
Parkplatz Careum
Parkhaus Rämibühl
ADD AS PARKING
Your location
Parkhaus Rämibuhl
Plattenstrasse 14
31 min + 2 min (25 km)
Fastest route, lighter traffic than usual
Start
Steps & more
31 min
Tolls
Find parking near destination
Parkplatz Schönleinstr.
Parkplatz Careum
Parkhaus Rämibühl
ADD AS PARKING
31 min + 2 min (25 km)
via Route 3
Level 0 spot 3 has been reserved you for Parkhaus Rämibüehl
More info
Remove
Parkhaus Rämibuhl
OVERVIEW
ABOUT
REVIEW
PHOTOS
ADD AS PARKING
RESERVE A SLOT
SHARE
SAVE
long duration
card payment
indoor
Zürichbergerstrasse 12
Open : Closes 02:00
parkingzuerich.ch
Tarif: 3.00 CHF / h
Places: 35 / Free : 12
Max. duration: unlimited
Reserve a Spot
Level 1 (10)
Chosen spot: Level 1 spot 8
Reserve
Navigate to spot