

# CityBike Vienna

Franz Brandl, Valon Lushaj and Artan Toplanaj

University of Vienna, Vienna, Austria

**Abstract.** This paper is about CityBike Vienna, a visualization project developed at University of Vienna, during the Visualization class of summer 2015. Its main purpose is visualizing the data city bikes in Vienna, so a city bike employee can have a better look of the most or least frequented stations of city bikes. . .

**Keywords:** Visualization, city bike, graphical tool, Schneidermann's Mantra

## 1 Motivation

CityBike Vienna has already 120 stations. This means, that there is lots of work to do. This includes many problems, like bike transportation from one station to another, empty stations (no bike to borrow anymore), filled stations (no place to park my bike), broken bikes that need to be fixed and thousands of clients, that need to be kept satisfied with the service of CityBike Vienna.

We concluded that the biggest problem of all is the station's traffic. This sums up the other three problems mentioned before, empty stations, filled stations and unhappy clients. A CityBike client wouldn't be happy if he would go down to the station and doesn't find a bike, especially if this happens almost everyday. Another bad scenario would be, if a client arrives to the station, which is near to his faculty building or his workplace, and can't park his bike. A better overview of the stations is needed!

A good visualization of data provides a better overview of each station. A good visualization tool would provide the real time traffic change of each stations over the day, or even hour!

## 2 Approach

Our goal was to develop a tool, that would visualize the CityBike data, to help the CityBike employees to better maintain the traffic. After the goal was set, the next step that followed was sorting and filtering the data. There was lots of data, that we didn't really need for our tool, e.g. the data about broken bikes, since we didn't want to visualize that.

We first sorted the data from one big file, to many other smaller files. We would save in one file informations about only one stations, like borrowed bikes, time when the bike was borrowed, the number of returned bikes and the time when

the bike was returned back. For us was clear that we wanted to draw circles on the map of Vienna, each circle representing a bike station. For data sorting and filter has been used ruby on rails [1].

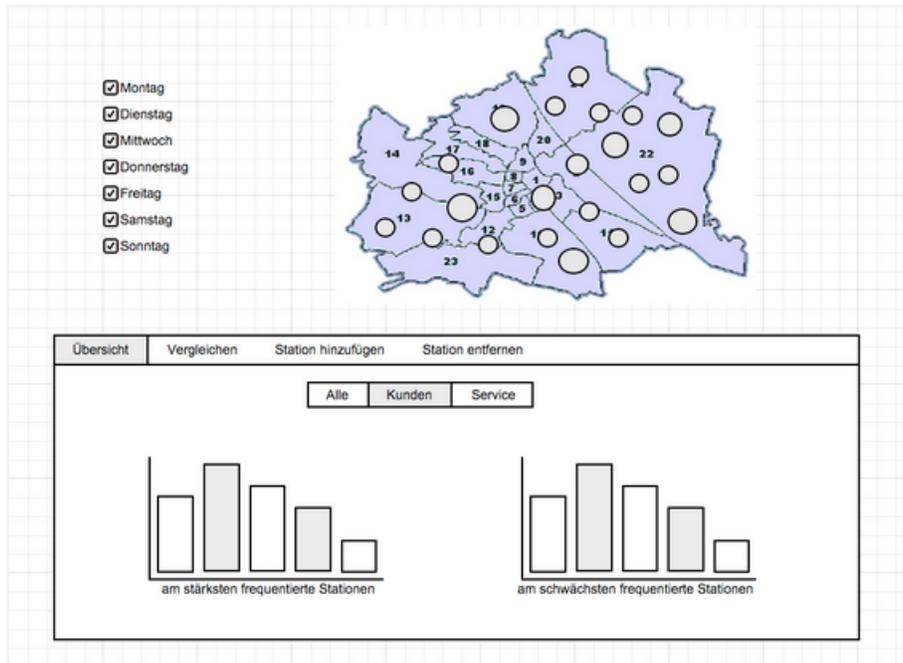
The next step was to think about our design, how it would look like, and where should we put the map and the charts. That's why we drew three different mockups.

## 2.1 Design choices

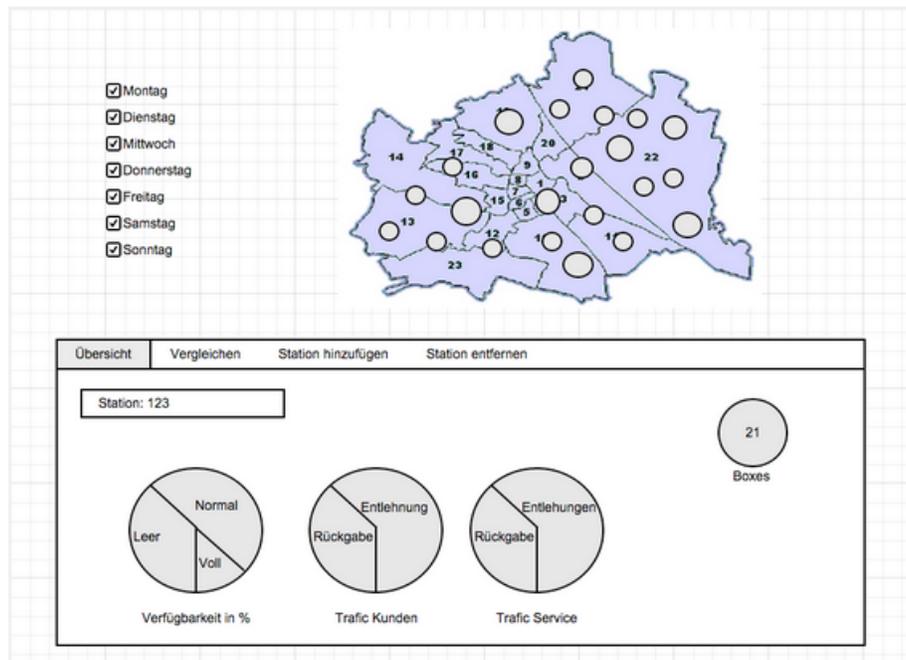
We chose, for our opinion, the best mockup. We made that decision, because the mockup was a good combination of the Vienna map with the stations drew on it and some other bar and pie charts. This combination provides a better overview from the whole traffic, because it has different functions built in it, like comparing the stations, it displays the most and least frequented stations, station traffic during the day, week, weekends, whole month or the whole year. It also corresponds Schneidermann's mantra design guideline of Overview First, Zoom and Filter, Details on Demand.

In the Start (First) view) all items in the dataset are being shown simultaneously, without any need for navigation. We could zoom in the map and filter out some stations, or only the stations in the first district. This is Zoom and Filter from Schneidermann's mantra. Under the Vienna map there are different options. By clicking any options, the tool corresponds with different charts or data, e.g. comparing the stations. That's Details on Demand from Schneidermann's mantra.

The following screenshots are taken from our lo-fi prototype:



**Fig. 1.** First view of our tool. The bar chart display the most (left) and least (right) frequented stations.



**Fig. 2.** Hier is some statistics about each station. The small pie chart on the right displays the number of bikes in the particular station. The other pie charts display aivailability of the station, borrowed bikes from the clients respectively employees and returned bikes from clients respectively employees.

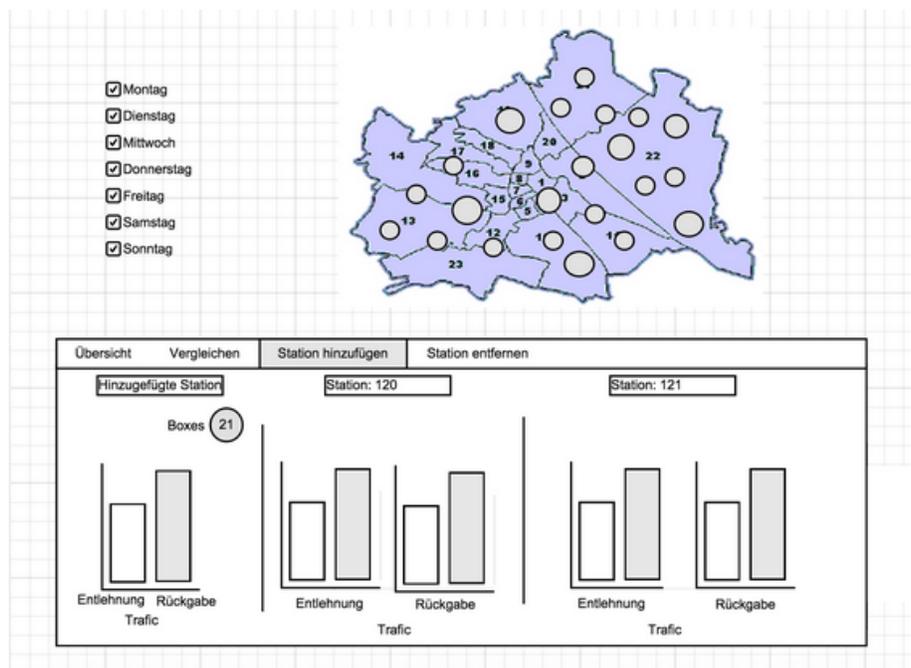


Fig. 3. Adding a new station.

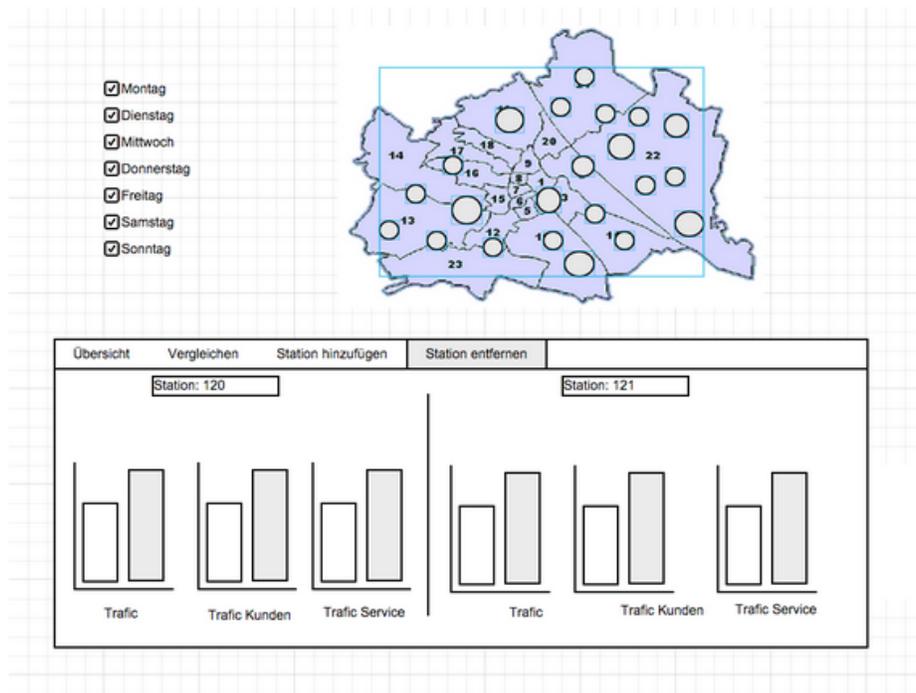


Fig. 4. Deleting a new station.

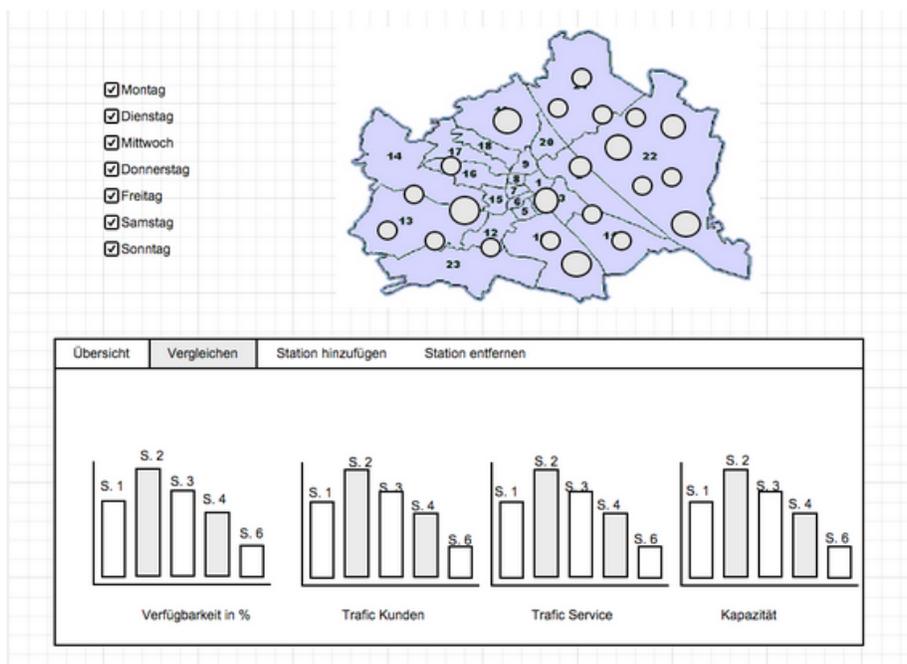


Fig. 5. Comparing the stations.

All that we decided to program in D3. But since we already had shown our lo-fi prototype to the class, we recognized that the most of them were confused and pretty sceptic about our functions to add respectively delete a new station. When we started implementing the tool, we saw that, the function is not possible to implement, since we weren't able to predict the changes in the CityBike traffic.

### 3 Implementation

Our visualization tool was done as a web application. The visualization tool is implemented with Data Driven Documents - D3 [2]. The implementation of the tool was the hardest part, since nobody from our team had experience with D3 and that's also why the project was very time consuming, so that we didn't achieve our goals from lo-fi prototype.

### 4 Results

As said before, because of our implementation problems, our tool couldn't achieve the predefined goals or functions. It is rather a minimalist version of our lo-fi prototype. There is only one major scenario of use:

1. A CityBike employee wants to know the traffic about any station, this means the number of borrowed and returned bikes. So he opens our tool, where he can see all the stations (circles) on the map of Vienna (the circles at the beginning have the same size) and on the dropdown menu he chooses between borrowed bikes (entlehnungen) and returned bikes (rueckgaben). Under the map is a line plot that displays the traffic for the whole year. On the end the employee can click on a certain station and read the statistics about it.

The following screenshots are taken from our tool:

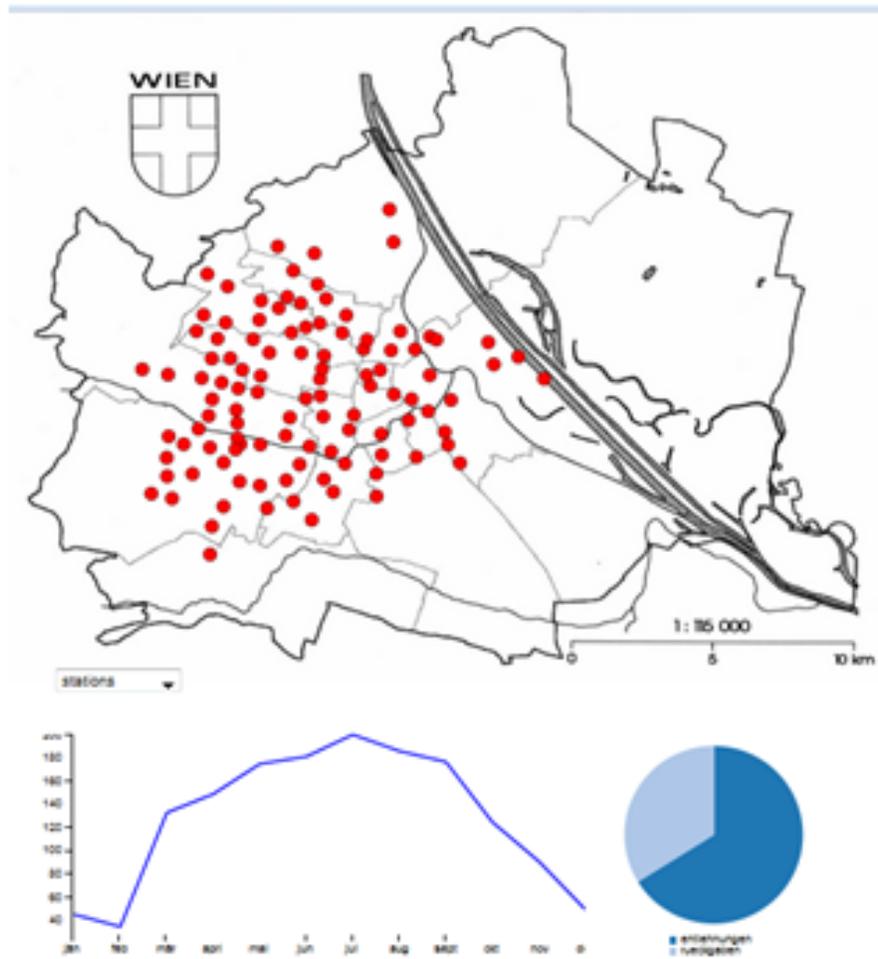


Fig. 6. First view of our tool

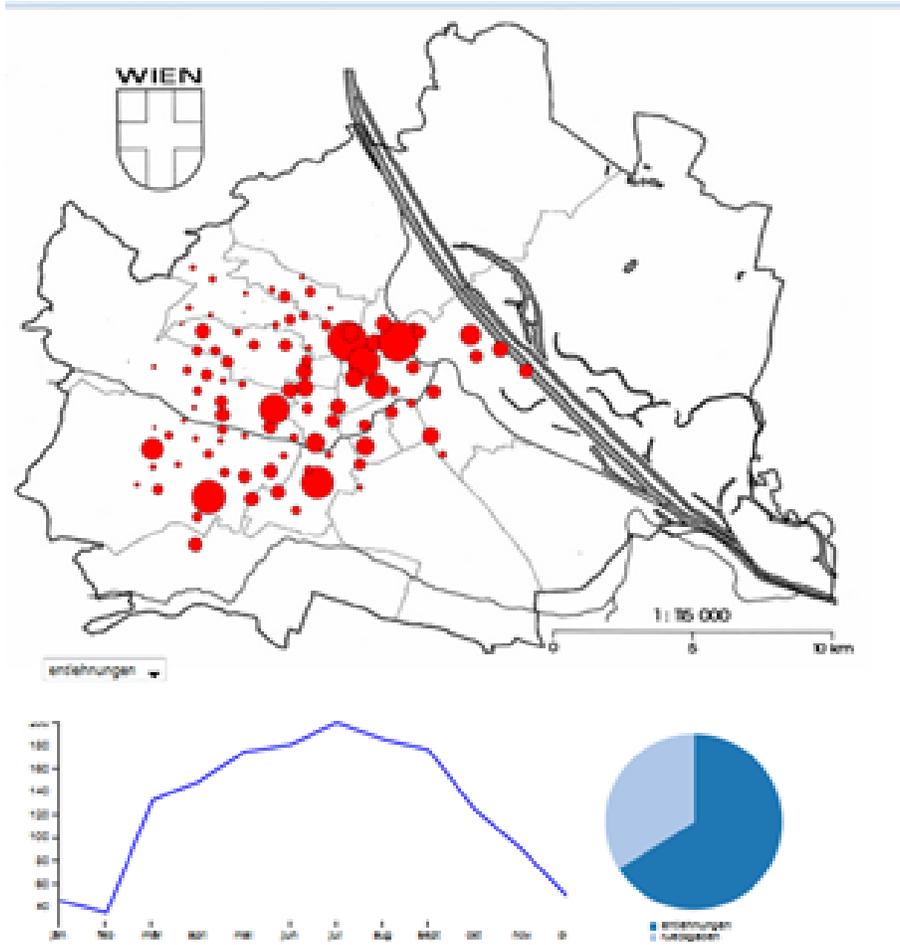


Fig. 7. The number of borrowed bike for all stations

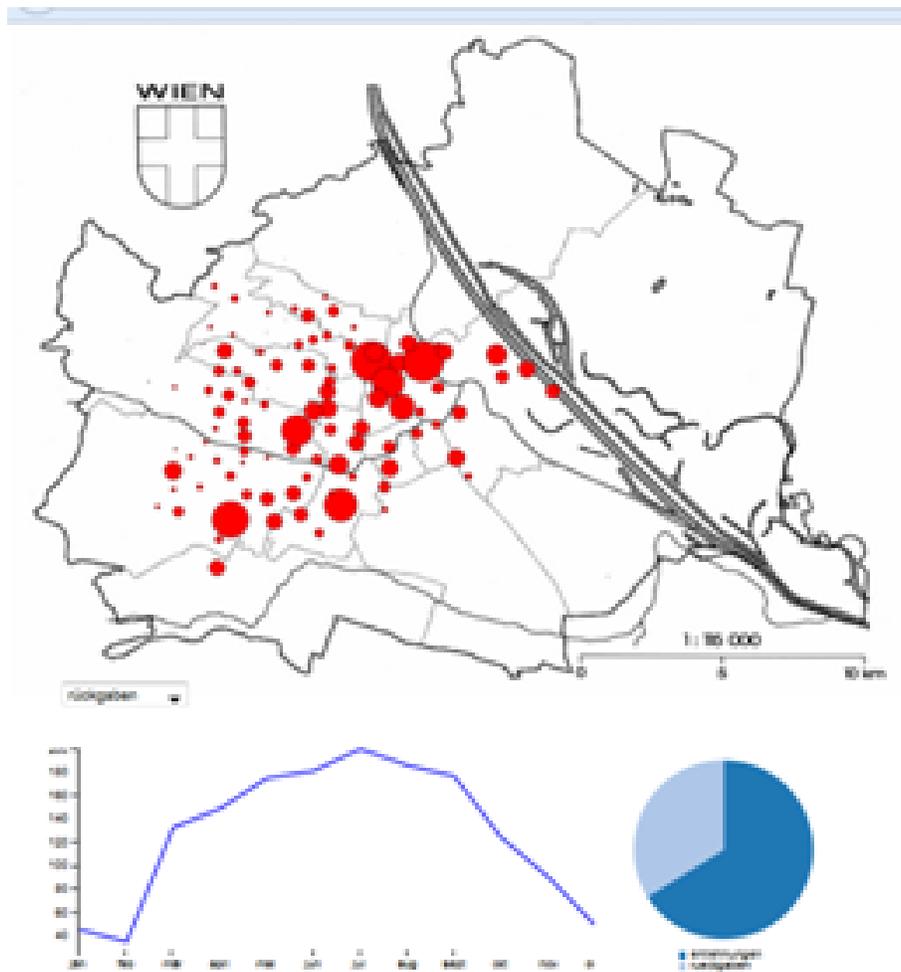


Fig. 8. The number of returned bike for all stations

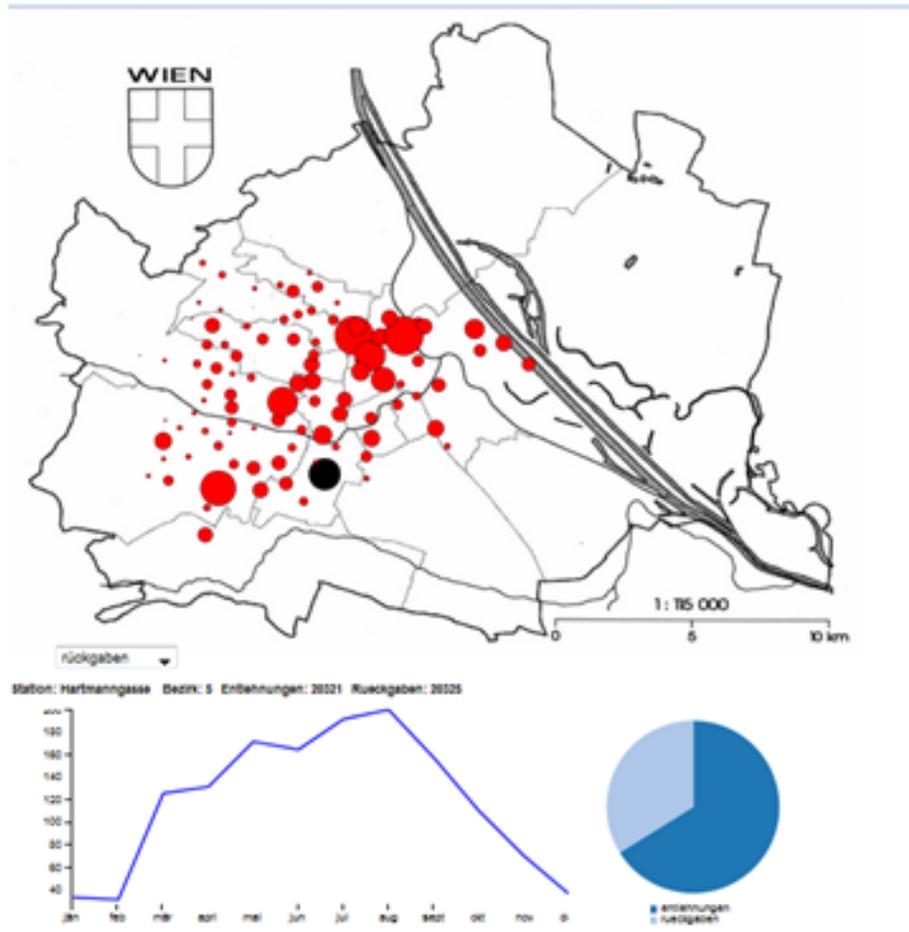


Fig. 9. Statistics about the selected stations (black circle).

## 5 Discussion

### 5.1 Strengths

We concentrated on a map-view, which we believe makes the tool very intuitiv and therefore easy to explore the data. For People who are familiar with the city, the map with the tips makes it very easy to find a station. The dropdown-menu allows the user to easily filter the stations with the highest/lowest traffic and then get details about them. It also allows you to see areas with a lot or very little traffic very easily. This is the main use case and we believe the tool handles this very well.

Something that was very usefull is the D3 function tip. We implemented it so that if you hover over a station with your mouse the name of the station is displayed. This makes it even easier to find specific stations.

D3 also allowed us to implement transitions which makes it easier to see that all the graphs are connected with each other. When you click on a station the linediagram and piechart change so the user immediatly knows that these graphs are from the selected station.

### 5.2 Weaknesses

Its hard to compare the sizes of the stations in the map and, if for example you want to know what the station with the highest trafic is you would need to click through a few stations and remember the exact numbers.

There is no way to compare stations to each other, which takes alot away from the linediagram, because most stations behave quit similar over the month.

There is no way to see how people use the bikes other than directly at the station, so no way to see where people are going and where they came from.

## References

1. Ruby Documnetation: <http://ruby-doc.org/>
2. D3 Documentation: <http://d3js.org/>
3. Bootstrap Documentation: <http://getbootstrap.com/>
4. Munzner, T.: Visualization Analysis, December 1, 2014