Taxi Checker: A Mobile Application for Real-Time Taxi Fare Analysis

Sebastian Osswald  
Technische Universität München  
Boltzmannstr. 15  
Garching b. München, Germany  
osswald@ftm.mw.tum.de

Niklas Brueckel  
Technische Universität München  
Boltzmannstr. 15  
Garching b. München, Germany  
n.brueckel@tum.de

Carsten Brickwedde  
Technische Universität München  
Boltzmannstr. 15  
Garching b. München, Germany  
carsten.brickwedde@tum.de

Markus Lienkamp  
Technische Universität München  
Boltzmannstr. 15  
Garching b. München, Germany  
lienkamp@ftm.mw.tum.de

Martin Schoell  
Technische Universität München  
Boltzmannstr. 15  
Garching b. München, Germany  
m.schoell@tum.de

Abstract
Taxi customers tend to use their smartphone to call and manage their taxi and even store details about their favorite taxi driver. Nevertheless, they often do not have the possibility to comprehend how the taxi fare is calculated and if it is correct. In our approach, we developed a smartphone application to track the taxi ride and calculate the fare in real time. At the end of the ride, mean cost calculations for three different sources are presented to the user and the tracked data contributes for community-based fare calculation. The application was tested by experts and will be evaluated in a user study to improve the fare calculation pattern.

Author Keywords
Taxi, Smartphones, Android, User Study, Automotive

ACM Classification Keywords
H.5.2 User Interfaces: Prototyping; H3.5 Online Information Services: Web-based services

Introduction
The idea of the taxi checker application was derived from the requirement that taxi customers have the basic desire to control the taxi driver if the fare price is acceptable. This could either be reasoned by the need to be sure if the fare price is calculated correctly or to
be sure about that the duration and distance of the taxi ride is acceptable.

A taxi ride is influenced by a broad range of impact factors e.g. it plays a major role if it is rush hour, the customers chooses a busy road to the airport and if the trip requires mainly the fast track line. Further, the price varies depending on road tolls, fees for ordering the cab, traffic jams and the weather. The perceived correctness of the final trip fare is thus heavily depending on factors that cannot be controlled. Taking the controllable factors in account, it is possible to combine the information about the chosen route, the fees and the price calculation to determine the correctness of the taxi fare.

As taxi companies operate mainly in city/urban areas the geographical area of application is limited to urban driving conditions. The characteristics of this areas are mainly an lower average driving speed and higher standing times compared to rural areas. To use this two impact factors as basis for the fare calculation, it seems fruitful to generate a database for each larger city. We believe that a comparison of the average taxi fare per kilometer will show a significant difference between cities.

In our approach, we are using a smartphone application to realize a real-time and post hoc analysis of the taxi fare of an example city. In the following, we present the state of art, the application and the implementation details.

**Background**

Enabled by the integration of various sensors into modern smartphones, the interest of users in mobile tracking applications increases rapidly [3]. While most of the tracking applications emerge from a trend commonly described as the quantified-self [4], there are also considerable possibilities emerging in the field of automotive and transportation applications. Smartphone applications are being used e.g. to analyze and influence the driver behavior [1]. Applications are further designed to support the driver and the passenger by providing different levels of access to the vehicle information system [2]. Out application is targeted to support the passenger and built up a community database that improve the application through user input. This will lead to an enhanced price fare calculation pattern for taxi fares and to an improved user experience, as the results of the calculation will get more accurate and thus more useful for taxi customers.

The Google play store for Android applications lists easy Taxi¹ and mytaxi² as the most requested taxi applications. The business models of both applications are focused on replacing taxi offices and on offering a convenient booking service for taxi rides. Besides the booking process, these applications are presenting various extra features, including the live tracking of an ordered taxi until arrival and in-app payment. In addition, mytaxi is offering the service of an upfront price calculation, based on google maps and local taxi fares.

Neither mytaxi, nor other less popular applications available in the Google play store or in Apple’s App Store, offer the service of tracking the actual taxi ride

---

¹ http://www.easytaxi.com/
² http://us.mytaxi.com/
and analyzing the fare in real time and ex post. The unique selling proposition of the proposed Taxi Checker application addresses this shortcoming. The application gives users the possibility to monitor the correct calibration of the taximeter during the ride in real time as well as an extensive ex post analysis, based on tracked GPS data and multiple price points to address the impact of the driven path.

**The Taxi Checker Application**

To ensure that the functionality of the targeted application enables the taxi user to start the real time calculation of the taxi fare during a taxi ride and to analyze the actually paid fare at the end, several data sources are combined:

- GPS data for tracking the taxi ride
- fares from the taxi company to calculate the costs
- the fare from the taxi ride itself as a reference
- Google maps data to calculate the best route in comparison to the tracked data
- Community data from previous rides to calculate a reference fare

The start screen of the applications provides a slick design and focuses on the only function available: the start button. After starting the tracking of the taxi ride, the second screen provides information about the trip duration the covered distance and the current taxi fare in real-time (upper screen half). At the bottom of the screen, the additional fees for e.g. extra luggage and booking can be checked and the standard fees are added to the current taxi fare. After pressing the end ride button when the taxi arrived at the destination, a prompt asks for the final fare the taximeter in the taxi displays.

**Figure 1**: Interface design and menu structure of the taxi checker application. From left to right: start screen – ride screen – result screen.

The following price check screen consists of a chart diagram displaying the actual fare as a yellow line across the three price range bars. These first price range bars displays the fare based on the tracked GPS data, the second displays the price range based on Google Maps data and the third displays an average community-based price.

For the fare calculation, the price patterns of the Munich taxi company are used:
<table>
<thead>
<tr>
<th>Minimum fare</th>
<th>3,50 €</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fare per kilometer</td>
<td></td>
</tr>
<tr>
<td>&gt;= 5 km</td>
<td>1,80 €</td>
</tr>
<tr>
<td>5 - 10 km</td>
<td>1,60 €</td>
</tr>
<tr>
<td>+ 10 km</td>
<td>1,50 €</td>
</tr>
<tr>
<td>Waiting time p.h.</td>
<td></td>
</tr>
<tr>
<td>(includes traffic)</td>
<td>6,00 €</td>
</tr>
<tr>
<td>Add-ons</td>
<td></td>
</tr>
<tr>
<td>suitcase</td>
<td>0,60 €</td>
</tr>
<tr>
<td>animal and cage</td>
<td>0,60 €</td>
</tr>
<tr>
<td>order fee</td>
<td>1,20 €</td>
</tr>
<tr>
<td>6 passengers +</td>
<td>5,00 €</td>
</tr>
</tbody>
</table>

Table 1: Fares for a taxi in Munich, based on the fare table available from the Munich Taxi Trade³.

Implementation and Evaluation Approach

The implementation of the taxi checker application focuses on the fare calculation for the taxi trip, the calculated reference route via Google Maps and the community-based calculation.

Fare Calculation

The main input data for the fare calculation are the covered distance and the standing time. These values will be filtered from the tracked GPS data and further used for the Google navigation reference calculation of the ideal trip. Additional data like the fare values (see Table 1) are needed. The starting price is given and extra fees for baggage, a minivan taxi or the order fee can be included by activating the checkboxes on the ride screen.

³ http://www.taxi-muenchen.com/privatkunden/tarif-und-preise.html

The taxicab ride needs to be divided in the distance driven (dd) and waiting time (wt) according to the fare for driving (fd) and waiting (fw) in Table 1. The calculation 

$$((dd*fd)+(wt*wf)+fe) = fare\ total$$

is used to calculate the real time fare as well as the reference values for the fare analysis. The extra fares are reflected in the value fe.

For the community fare, the application creates a new taxameter class using the average paid fees per kilometer and per hour. The values are fetched from our database, where all trip fares are stored. As the fares underlies influence factors that cannot be controlled (traffic jam), the application displays price ranges instead of exact calculated values.

Google Maps reference route data

To calculate the best driving route, Google navigation data is used. For each trip, the end and starting point is logged and a request inside the Google Maps Android application is started. As a result, the track distance and the duration with and without traffic are calculated. All three values used to calculate the fare in the same way like with the GPS data. Unfortunately, it is not possible to get direct data access to these value as it is only possible to create an intent with given origin and destination and start the Google Maps Navigation as a new activity.

The work-around solution we implemented uses the Google Directions API. Making use of an HTTP request, you can access the majority of the static Google
navigation functionalities e.g. several modes of transportation or multi-part directions. The directions responses are returned either as a JSON or an XML file, indicated by an output flag. Within the taxi checker application the JSON format is used. The HTTP response is parsed in a JSON object and the distance and duration can be fetched. However, receiving the trip duration considering current traffic conditions is only available within maps for business customers. We used a statistic based, time and location dependent traffic factor, which is initial set and will be validated and optimized using the growing community data.

Nevertheless, the calculation of the fare for the most efficient route (distance & traffic depended) can be optimized with more accurate traffic information. If a business customer account for Google Maps is available and the data is gathered, the community data is still needed to estimate the standing time. This is reasoned by the assumption that the standing time will not be the trip duration considering current traffic conditions minus the duration without any traffic. In fact, due to the traffic, the average driving velocity will decrease and the standing time will increase. Both factors impact the actual driving duration and the community, why data is needed to estimate the standing time.

The application is designed in a modular way to add additional “taxameter” classes. This new classes represent the cost patterns for different cities. By determining the users’ area via GPS, the correct class and therefore the correct price pattern can be chosen.

Community based calculation

The community database is used to store the driven taxi rides and to compute average fares normalized on the fees per driven kilometer and per standing hour. Moreover the database can be used for the standing time estimation, as previously mentioned.

For the average fares, the actually paid values are reverse calculated to a raw price excluding the additional fees. For the average fares, paid fares are reverse calculated to a raw price excluding the additional fees. The raw fare is separated in two price ranges, one referring to the standing time and one to the covered distance. These values are stored for each ride and within the database request the overall average value is calculated. For defining the price range, we extract the variance of the fare distribution.

Figure 2: Communication between the application, the google server and the database (including GPS tracking).

The application was tested in the field by the developers to gather usability issues and iterate the
user interface design. Based on that, the price calculation was corrected and the price range for the community-based calculation was manually set. This was done to avoid outlier in the data that might significantly affect the data in the beginning.

A user study with 15 participants is planned to evaluate the accuracy of the application and the acceptance of the users. We believe that the displayed price range on the final price check page influences the acceptance of the users and will significantly affect the trust into the application. The expected results contribute to improve the price range interface for the three sources (GPS taxi meter, Google, community). This leads to an even more realistic calculation of the price range combined with an accurate representation of the taxi fares.

**Conclusion**

In this paper, we presented a concise description of a smartphone application that enables users to determine their taxi fare in real time during a taxi ride. A database is set up to store and compare taxi rides and the user receives feedback about three different ways of calculating the taxi fare. In the beginning, the community-based calculation of the average taxi fares is of low accuracy. This will improve, as soon as more users start to use the application. The future work in this project is the evaluation of our system in a user study to determine the acceptance of the users, to improve the accuracy of the fare calculation and to investigate if the usage of this application leads to potential conflicts with the taxi drivers.

We will further extend the area of application, as it is for now limited towards one city (Munich). We will integrate the price patterns of taxi companies from the major cities of Germany into the database. Subsequently, the application will be distributed in the Google Play Store with the aim to generate mobility datasets across Germany that can be used for further studies.

**Acknowledgements**

This work was conducted at the Technische Universität München, Institute of Automotive Technology. Special thanks goes to Maria Kugler, Benedikt Jäger, Johannes Betz, Stefan Grubwinkler, Moritz Hann and Matthias Ott for their support.

**References**


