Saving time, money and the environment – *vHike* a dynamic ride-sharing service for mobile devices

Christoph Stach
University of Stuttgart
Institute for Parallel and Distributed Systems
Universitätsstraße 38
70569 Stuttgart, Germany
Email: Christoph.Stach@ipvs.uni-stuttgart.de

***Abstract*—In times of increasing traffic-related problems, such as air-pollution or traffic jams, ride-sharing is one of the most environmentally friendly and pleasantest ways to travel. The many benefits are offset by a multitude of prejudices and fears, including security concerns and a heavy scheduling and coordinating burden. For this reason this paper introduces *vHike* an easy-to-use management system for dynamic ride-sharing running on modern Smartphones. By the use of techniques well-known from Web 2.0 social networks the threats and social discomfort emanated by ride-sharing is mitigated. With *vHike* we want to show that a proper designed social dynamic ride-sharing system can be feasible and viable.**

***Index Terms*—ride-sharing; trust; security; location-based.**

1. Introduction

In 2009 there were nearly 50 Million cars with a rising trend owned by private hand alone in Germany. These cars generate a lot of running costs and cause increased traffic congestion, not to mention that they are also accountable for a bulk of air pollution. While they are driven over 900 Billion Passenger-km every year with just about an average of 1.36 passengers in each vehicle a majority of these rides could be economized if the drivers would engage in ride-sharing programs. Unfortunately, there are many prejudices and fears towards those programs. The most frequently mentioned reasons why people do not want to participate in a carpool are a lack of flexibility, a high organizational effort and security issues in general.

In order to counteract these reasons, there exists some software solutions to manage carpooling. The best-known and accepted products in this sector, such as MitfahrZentrale.com [1], are capable to arrange long-term planned carpooling. Therefore, they provide a platform, that allows registered user to place or search for carpooling offers. Even for the management of frequently repeating journeys, such as the formation of a commuter communities, solutions are provided by portals such as Pendlernetz [2].

Nevertheless, these solutions facilitate the organization of a trip only to a small part as the actual agreement has to be made separately and without the support of the platform. For issues of security, all user can rate each other and report suspicious incidents that are related to the trip. For real security, this is however far from optimum. Furthermore, these platforms are extremely inflexible.

However, these portals exhaust the potential of current trends in technology not sufficient by far. The widespread distribution of modern mobile phones equipped with high performance CPUs and additional features such as Bluetooth or GPS, makes it possible to assume that a user is always trackable and has a permanent connectivity and reachability. Also servers have sufficient computing power to enable a dynamic driver / rider matching service. Finally, rating technologies in social networks have evolved since the first static carpooling services have emerged. They are now capable to maintain trust among strangers in the Internet.

For this reason innovative solutions using these technologies are required to enable a relatively short-term planning phase, so that they are also promising. With *vHike* we will introduce such a solution. The rest of the paper is organized as follows. In Section 2 we will present some major approaches in this area. Section 3 deals with the requirements of such a system. Our implementation of *vHike* is presented in Section 4. Desirable but missing functions are the topic of Section 5. Finally, Section 6 gives a short conclusion of our work.

2. Related Work

While there are obvious some more or less good implementations for traditional carpooling solutions for the so-called *dynamic ride-sharing* exist only as theoretical approaches. We will give a brief overview of the most important representatives in the following.

Hartwig et al. [3] determine some key factors that influence the success of a dynamic ride-sharing service significantly. Most people who reject hitchhiking in general do so because of the high security risks. Therefore, it is important for a platform organizing dynamic ride-sharings to supply the user with detailed and reliable information about the other participants. A management service has to be very flexible and easy to use in order to become accepted by a broad user base, also. If such a system does not become totally ubiquitous and cannot get integrated seamless into the everyday life it will not be viable.

Resnick [4] presents a research agenda on how to implement such a system. To increase the flexibility he suggests to drop the concept of a few standardized pickup and drop off locations and replace them with dynamically chosen collecting points regarding the current position of the rider and driver. Furthermore, the estimated time-to-pickup will be calculated automatically in real-time and the rider will be informed about it. Moreover, he determined that a user interface has to be convenient as well as reduced to the most necessary. If the handling is too awkward especially the driver who would like to offer a voluntary service will reject to use it. The minimalism is required if it should be operable while driving. For security matters Resnick relies on a login / logout mechanism where all involved parties have to check out after the arrival or else an emergency call will be made. Without question, this approach satisfies all above mentioned requirements, but a concrete implementation is missing.

Kelley [5] introduces a system which reports about available opportunities to ride fully automatically. It takes the available seats, the travel time and travel route as well as the current context of the rider into consideration. The needed information is retrieved by special RFID tags. Thereby, the agreement will be simplified, because no user interaction is needed. When a driver is close to a rider both of them will be informed and can decide whether or not sharing a ride. In order to build trust towards the random travel mate a registration and rating mechanism is integrated into the system. However, this rating system is mainly based on the quantity of accomplished rides, without taking their quality into consideration.

Smart Jitney [6] is a slightly different system design approach. Murphy suggests to equip every vehicle with identifiable and trackable hardware. Just as a cab, this device can be contacted to reserve a seat. The driver has to acquire a special license. In addition to the licensing a mechanism to monitor the ride in real-time should guarantee the security of all involved persons. Nevertheless, an emergency button in each car informs the closest law enforcement center as soon as it is pushed by any participant.

Gidófalvi et al. [7] draw their attention towards algorithms that allow a completely automatic driver / rider allocation. Therefore they mainly take two factors into account: minimization of detours according to the current location and maximization of social matching which is solved by grouping all users into homogeneous groups based on a social network. Mobile phones with GPS sensors are used for tracking and communication.

SafeRide [8] is a draft of a dynamic ride-sharing system for modern Smartphones. Morris proposes to use existing Web technologies such as Google Transit and combine them in order to facilitate user interaction. To ensure security matters the author implies to use standard mechanisms of social networks, including registered users and comprehensive profile pages, as well as additional vetting in the form of licenses and driving records. Further, he recommends the riders to rate the driver (and vice versa) in an also included eBay-like reputation system. In the case that there still is an incident during the ride, any participant can call for police assistance via their Smartphones and all accrued date will be stored by SafeRide as pieces of evidence. Unfortunately, there is no testable implementation of the system.

With *Ride Now!* Kirshner [9] introduces a testable computer- and telephone-based dynamic ride-sharing system. In his proposal registered users can contact the system whenever they need a ride. Then the system finds a feasible driver for this request and informs the rider about the drivers contact details. The agreement shall be made without support of the system. Kirshner implemented Ride Now! under the above-mentioned aspects and showed in several demo sessions variable acceptance within the users [10]. Nevertheless, all of his ambitious pilot programs were discontinued because of economic failure. Therefore, such a project has to rely on volunteers and should not include a monetary reward for the drivers.

3. Requirements

Learning from the experiences of the systems introduced in Section 2 we were able to ascertain multiple strongly needed technical as well as non-functional requirements towards a dynamic ride-sharing platform.

Web-based: Whenever an application needs to manage plenty of user data, the best way to achieve this is to use a database. A system for dynamic ride-sharing should be implemented with a classical three tier architecture, therefore. The mobile device has to act as a thin client mainly submitting any context data accruing on the device to a central main server and providing the user with query results from it.

Community: As any user of a dynamic ride-sharing platform has to put a lot of trust into the other service providers these services have to be made available to a closed community of registered members exclusively. Furthermore, each user should be enabled to form an private *crowd* - a small subdivision within the community consisting of special friends and associates. These crowds can be used to prefer certain providers or to exclude others already at a very early stage.

Privacy: In order to identify a user properly he or she has to lodge his or her private master data as well as his or her current context data, e.g. the position or the travel intention, in the *vHike* System. This data has to be evaluated by *vHike* and several parts of the data are transferred to other selected user. However, this may not happen arbitrarily. The owner of the data has always to be informed who wants to access which kind of data and why this is important for the work flow. A user has to agree these transfers in advance of a data exchange.

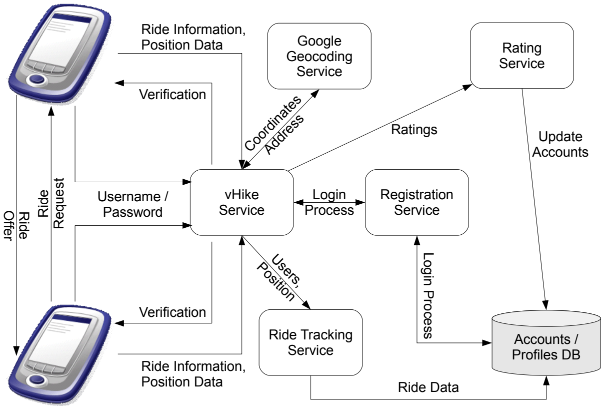


Figure 1. Services and System Components of *vHike*

Trust: After each successfully completed ride the rider as well as the driver is encouraged to rate each other. This is not only a confirmation that the trip has been successful, but also a way to build trust towards the other members. Similar to systems such as eBay thus can identify especially recommended drivers, respectively, riders or exclude others very fast. Extremely inappropriate participants can also be excluded completely from *vHike*.

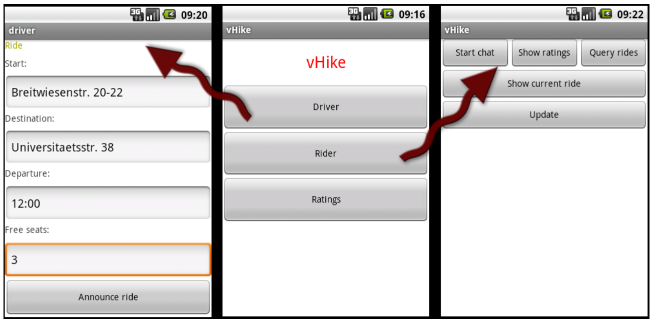
Insurance: In order to improve the safeness within *vHike* even further, details about each ride are recorded by the system. In this way, it is not only possible to resolve any emerging issues in connection to a ride, but also track the course of a ride in real-time is feasible. These data could be shown on a map and some selected persons could keep overview of the whereabouts of a user at any time. This may cause unusual deviations from the expected route to be established soon and in case of major violations against guaranteed agreements this information can be handed over to security authorities.

4. *vHike*

Based on these considerations and demands towards such a dynamic ride-sharing system *vHike* has been implemented as depicted in Figure 1. In the following Section we will discuss the operating mode in detail.

After a user has successfully generated an account using any given web browser he or she may connect to the system with a mobile device. Currently Android-based devices are supported, exclusively. Once logged in, the user may choose a role according to his or her needs.

In Figure 2 screen captures of the two main menus and the start menu are shown. Following the left arrow the main GUI for drivers can be seen. A driver can create a new offer for a ride, defining the start time and the destination. This information is transmitted to server on the one hand and also broadcasted to all activated *vHike* applications in the surrounding via Bluetooth. In the case of *vHike* we do not see the relatively short range of Bluetooth as a disadvantage but as a benefit. Thereby, only user who are close enough to the driver are informed. Evaluations taken from some of the related work has shown, that most of the drivers are not willing to make a detour, just to pick up a rider. So there is no need for a notification to a wider audience. When entering the tour a user is supported by *vHike* in two ways. Our system tries to predict both the starting point as well as the destination in order to fasten up the input process. For the former the current user location is determined as precise as possible - we cannot relay on a sufficient GPS signal in the sample use cases we had in mind for *vHike*, so tracing by Cell-ID has to be good enough. This location is passed to the Google Geocoding Service in order to receive an address. Most likely, this will be the intended starting point. The latter will be the home address of the driver in many cases. For this reason the address can optionally be stored within a user profile. In addition to the starting point, the destination and the start time the driver indicates how many free seats he or she is willing to offer. After announcing his or her offer the driver enters a standby state waiting for interested riders until he or she starts the ride.



Start:
Breitwiesenstr. 20-22
Destination:
Universitaetsstr. 38
Departure:
12:00
Free seats:
3

Figure 2. *vHike*'s GUIs

A rider on the other hand is not allowed to initiate a request for a ride. The rider menu, shown in Figure 2 on the right, features a list of offerings available. If one of them interests the rider he or she can take a look at all available rating of the corresponding driver to get a first impression. The rider can either start a chat via Bluetooth with the driver to clarify final open points or apply straight for a free seat if the ratings seem acceptable. After applying a request the rider may retrieve detailed information about the ride again at any time. Finally, when no suiting offer - or no offer at all - is shown in the riders menu a possibility to update the list manually is given, also.

Whenever a rider applies for a free seat the offering driver is informed about it. Then the driver can either obtain additional information about the rider or accept or decline the application. Of course he or she can enter a chat with the interested rider on his part, also.

In our understanding, an agreement, worthy to be monitored further by *vHike*, is only reached once the driver has accepted a rider. The driver / rider(s) combination is forwarded to the system server and stored in the record created for this ride. To begin a ride the driver collects all riders as it has been agreed. *vHike* then displays for both sides a huge emergency button which immediately triggers an alert to the responsible authority. In addition to the call the current location of the person feeling threatened is recorded in our database.

In *vHike* a ride can end in two ways: either successfully which means that both driver and rider reached the appointed destination inviolate or unsuccessfully which means that the ride has come to an end prematurely or at least one participant did not arrive at all. Although this may have several reasons, such a fact is logged by *vHike* for further investigation. After both sides have confirmed their unharmed arrival, they can rate each other. It is recommended to do this confirmation not until a user has reached home for security matters.

The rating process itself is comparable to other well known rating systems such as the one used in eBay. In addition to up to five stars a user can write a short description of ride-related unusual events or special features or an explanation why he or she has rated this way. After the rating process the ride is considered as completed for *vHike* and the service can be started all over again.

In addition to this dynamic ride-sharing version of *vHike*, which is the main feature of our system, we also offer a static mode comparable to systems such as MitfahrZentrale.com. In this mode user can announce their long term planed journeys in our Web-based forum in order to address to potential riders within the community. However, an agreement made in this way, is not protected by *vHike*.

5. Future Work

Future Work is planed along several directions. In upcoming releases of *vHike* we are working on minor improvements of the driver-rider-matching process to reduce the result set of possible candidates further. The user interface and the operability will be embellished and improved, too.

Our main focus is to expand the functionality of two aspects: On the one hand we would like to include a possibility to share the location of a user with his or her friends within the *vHike*-community. Thereby, they will be able to meet each other and stay in contact. Google already offers such a service called Goolge Latitude, which will have to be adapted for our application. On the other hand we will include the Google Maps service into *vHike*. This will help us to calculate directions and suggest a time- or distance-optimal route to any given destination. Additionally, we will be able to offer the full functionality of a common navigation system to the driver. But not only the driver can benefit of such a feature but also worried parents. They will be able to track the location of their children and “follow” their trace on a map in real-time via Web access.

Then we are looking forward to release a test version of *vHike* in the Android Market in order to attract a huge number of voluntary *’guinea pigs’*. With their help we will be able to evaluate how well our system runs under real conditions and we will collect measurement date to determine the success of *vHike*.

6. Conclusion

In this paper we discussed the meaning and use of a management application for dynamic ride-sharing. We identified the main reasons why people reject forming carpooling or hitchhiking. Based on these preliminary considerations we defined some technical as well as non-functional requirements.

With *vHike* we introduced an implementation of such an application for mobile devices. With our system not only the burden of scheduling and coordinating a ride but the fears against ride-sharing itself should be reduced. Therefore, we build a trustworthy community whose members can rate each other in order to identify and remove rotten apples very quickly. We relied on Bluetooth both for detecting other user nearby and for communication. In the implementation, we never lost sight of usability, safety and trust the most important aspects of an application in such an area.

References

[1] MOVECO GmbH, *Mitfahrzentrale.com*, Jun. 2008. [Online]. Available: www.mitfahrzentrale.com.

[2] ——, *Pendlernetz*, Sep. 2010. [Online]. Available: www.pendlernetz.de.

[3] S. Hartwig and M. Buchmann, “Empty Seats Traveling - Next-generation ridesharing and its potential to migrate traffic- and emission problems in the 21st century,” Nokia Research Center Bochum, Tech. Rep., Feb. 2007.

[4] P. Resnick, “SocioTechnical Support for Ride Sharing,” University of Michigan, School of Information, Tech. Rep., 2003.

[5] K. L. Kelley, “Casual Carpooling – Enhanced,” *Journal of Public Transportation*, vol. 10, no. 4, pp. 119–130, 2007.

[6] P. Murphy, “The Smart Jitney: Rapid, Realistic Transport,” *New Solutions*, vol. 12, p. 12, 2007.

[7] G. Gidófalvi, G. Herényi, and T. B. Pedersen, “Instant Social Ride-Sharing,” in *Proceedings of the 15th World Congress on Intelligent Transport Systems*, 2008.

[8] J. H. Morris, “SafeRide: Reducing Single Occupancy Vehicles,” Carnegie Mellon school of computer science, Tech. Rep., 2009.

[9] D. Kirshner, *Ride Now!* 2010. [Online]. Available: www.ridenow.org.

[10] ——, (Jan. 2007). “Pilot Tests of Dynamic Ridesharing,” [Online]. Available: http://ridenow.org/ridenow_summary.html.