FP7-SMARTCITIES-2013

STREETLIFE

Steering towards Green and Perceptive Mobility of the Future

WP5 – END-USER APPLICATIONS

D5.2.2 – End-user applications techniques and tools (intermediary)

Version 0.0

Due date: 30.09.2015 **Delivery Date**: 02.10.2015

Author(s): Antti Nurminen (AALTO), Konsta Sirvio (AALTO), Stefan Schaffer (DFKI), Annapaola Marconi (FBK), Giuseppe Valetto (FBK)

Partner(s): AALTO, DFKI, FBK **Editor**: Konsta Sirvio (AALTO)

Lead Beneficiary of Deliverable: AALTO

Dissemination level: Public **Nature of the Deliverable:** Report

Internal Reviewers:
Monika Pepik (DFKI), Mika Vuorio (LOGICA)

EXECUTIVE SUMMARY

Work Package 5 of STREETLIFE project delivers end-user applications that are used in the mobility context in order to promote use of sustainable transport modes. The applications developed during the project are tested in real life by the end users. The applications belong to different Technology Readiness Levels. Some of the applications have already been in commercial use, but their functionality has been improved during the project. Some applications, on the other hand, are on the research or development phase.

The Work Package 5 is divided into five tasks: 1. Intermodal Personalised Travel Assistance and Routing, 2. Virtual Mobility, 3. Citizen Participation and Gamification, 4. Advanced Graphical Interfaces and 5. Mobile App Development. Task 5 integrates selected techniques, algorithms and components from the four other Tasks, and delivers them as user-facing sustainable mobility mobile Apps.

We report in this deliverables on the advances that have been made in Year 2 of the STREETLIFE project, with respect to all the tasks, and on the applications that are ready to be used in the pilot cities and small-scale research experiments during the upcoming second iteration of in-the-field experimentation and evaluation. Multimodal routing services integrate personalisation features in order to better serve users of journey planners. A gamification component has been integrated in all the city pilots and more elaborated incentive concepts were developed. A small-scale Mixed-Reality application was created for Berlin to be used to test how mixed reality can help navigation in the mobility context. Improvements on Augmented Reality and support for full Berlin 3D map are underway to allow a more definitive assessment of the usefulness of the interfaces. New device platforms have been introduced including smart watches for travel assistance and immersive head-mounted displays for Virtual Mobility.

The future work will concentrate on refining the end-user applications so that they can be successfully utilised in the city pilots and small-scale research experimentation. The feedback from the pilots will be incorporated in the application development to increase their technology readiness level. In terms of mixed reality, field tests will be designed and tested in Berlin and Tampere.

Disclaimer: This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 608991.

D5.2.2 – End-user applications techniques and tools (intermediary)

TABLE OF CONTENTS

ABBREVIATIONS

2D	Two dimensional
3D	Three dimensional
API	Application Programme Interface
AR	Augmented Reality
AV	Augmented Virtuality
BER	Berlin
CO_2	Carbon dioxide
CSV	Comma Separated Value
DK2	Development Kit 2
GE	Gamification Engine
GPS	Global Positioning System
GTFS	General Transit Feed Specification
GUI	Graphical User Interface
ICT	Information and Communication Technology
IJP	Intermodal Journey Planner
IMU	Inertial Measurement Unit
IoT	Internet of Things
IT	Information Technology
JNI	Java Native Interface
MMIR	Mobile Multimodal Interaction and Rendering
MR	Mixed Reality
OTP	OpenTripPlanner
PA	Public Administration
PBF	Protocolbuffer Binary Format
PT	Public Transport
PnP	Perspective-n-Point
ROV	Rovereto
RTK	Real-Time Kinematic
SaaS	Software as a Service
SDK	Software Development Kit
SfM	Structure-from-Motion
SIRI	Service Interface for Real Time Information
SLAM	Simultaneous Localisation and Mapping
TRE	Tampere
VRS	Virtual Reference Station
Y1	Year 1
Y2	Year 2

EXPLANATIONS FOR FRONTPAGE

Author(s): Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

Partner(s): Name of the partner(s) whose employee(s) the author(s) are. List them alphabetically.

Editor: Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author – if he has generated the Foreground - as well as an Editor – if he also edits the report on its own Foreground.

Lead Beneficiary of Deliverable: Only one. Identifies name of the partner that is responsible for the Deliverable according to the STREETLIFE DOW. The lead beneficiary partner should be listed on the frontpage as Authors and Partner. If not, that would require an explanation.

Internal Reviewers: These should be a minimum of two persons. They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically this review includes: Identifying typos, Identifying syntax & other grammatical errors, Altering content, Adding or deleting content.

PARTNER

Fraunhofer	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
FBK	Fondazione Bruno Kessler
SIEMENS	Siemens AG
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
AALTO	Aalto University
DLR	Deutsches Zentrum für Luft- und Raumfahrt
CAIRE	Cooperativa Architetti e Ingegneri - Urbanistica
Rovereto	Comune di Rovereto
TSB	Berlin Partner for Business and Technology
Tampere	City of Tampere
LOGICA	LOGICA Suomi Oy
VMZ	VMZ Berlin Betreibergesellschaft mbH

LIST OF FIGURES

List of Tables

1. Introduction

Work Package 5 focuses in development of STREETLIFE end-user applications in a research-oriented and user-centred manner. Our end users are mainly individuals, ranging from local people, travelling daily between their home and work, to casual visitors, experiencing a new environment. We assume that our travellers have a variety of travelling means available to them, and require our applications to provide multimodal alternatives.

STREETLIFE end-user applications aid end users in mobility planning, pre-experiencing routes and act as online travel assistants with real time features. The design and development of these applications are driven by the key motivation of encouraging end-users participation and energy-efficient/carbon-low behaviours. To this extent, the work package develops participation and gaming techniques that are exploited within the mobile applications for engaging end-users in creating and sharing their mobility information and experiences, and to set-up and manage green mobility incentives and rewards.

This deliverable presents the work toward supporting our *Use Cases* in deliverable D8.1.2 by presenting the current status of each *Task*. We focus on the highlights and the effort put forth by each partner since the last deliverable D5.2.1.

Work Package 5 is divided into five Tasks, which set up the main technologies for the foreseen application features:

1. *Intermodal Personalised Travel Assistance and Routing* sets up a common toolset for personalised travel assistance and routing algorithms. This toolset is the basis for advanced context-aware real time assisting features supported with multimodal routing that can proactively aid the end user before and during a trip.
2. *Virtual Mobility* seeks to lower the threshold of using alternative and greener means of travel via visual pre-experiencing of the suggested routes.
3. *Citizen Participation and Gamification* provides development of engagement techniques that will a) facilitate end-users with tools for creating and sharing their mobility information and experiences, and b) increase enjoyment of greener means of travel via games, implemented with a customisable gamification engine.
4. *Advanced Graphical Interfaces* advances the visualisation of traffic, routes and virtual mobility via mixed reality techniques, namely a) 3D virtual environments and b) augmented reality.
5. *Mobile App Development* integrates selected techniques, algorithms and engines from the four previous Tasks, as well as from other STREETLIFE WPs, into actual applications, with advanced user interfaces and personalisation features.

Tasks 1-4 form the basis for the actual integrated application as fundamental *features* i.e. functionalities that the application performs. These features depend heavily on available static, real time and crowdsourced data (T3.1, T3.2, T3.3). This data dependency also leads to site dependency, for any localised data. For generally available data, such as OpenStreetMap, a routing solution can be a general feature. However, even in this case, data quality may vary, and lead to site-specific differences in quality of service.

The last task (5), *Mobile App Development*, is the locus where all of the innovations developed in WP5 are integrated and delivered, by means of implementation as user-facing mobile Apps to support and enhance sustainable urban mobility.

STREETLIFE end-user applications are customized to the three pilot sites: **Berlin**, **Tampere** and **Rovereto**. Each city has its own city-specific requirements features and available data, which affect the resulting end-user applications. Table 1 summarises the applications per pilot site and the innovations that have been pushed to those Apps.

Table 1: STREETLIFE End-User Applications

Application name	**Pilot site**	**Innovations delivered**
STREETLIFE Berlin Mobility App	Berlin	IJP, travel assistance, user personalisation, gamification
TrackMe App	Berlin	User tracking
Mixed Reality Berlin App	Berlin	3D map of Berlin, AR of PoIs, Interest Management, crowdsourcing and crowdsensing
Mixed Reality Tampere App	Tampere	3D map of Berlin, AR of PoIs, Interest Management, crowdsourcing and crowdsensing, IJP
LOGICA IJP	Tampere	IJP, availability of parking, gamification
ViaggiaRovereto	Rovereto	IJP, travel assistance, user tracking, mode detection gamification, user personalisation, integration of mobility policies
ContaParcheggi	Rovereto	Crowdsourcing of parking availability
Bike sharing App	Rovereto	Crowdsourcing of bike availability and status
Car-pooling App	Rovereto	Car pooling trip planning and execution, gamification

1.1. Outline of the deliverable

The deliverable is organised according to the Work Package tasks. Chapter 2 introduces the developments of intermodal personalised travel assistance and routing. The virtual mobility concepts and research are presented in Chapter 3. The gamification concept and engine is outlined in Chapter 4. This chapter also introduces the work we have carried out in a study on incentives and engagement models, although the bulk of that work is reported in Annex I. Advanced graphical interfaces, namely augmented reality and 3D maps with their current improvements are explained in Chapter 5, while Chapter 6 sheds light to the current status of the mobile applications in each of the pilot city. The last chapter concludes the deliverable.

2. Intermodal personalised travel assistance and routing (T5.1)

Intermodal personalised travel assistance and routing is a core functionality needed for the end-user applications developed in STREETLIFE. In D5.2.1 a general concept for personalised travel assistance and routing has been defined. Based on these ideas we further develop the existing pilot solutions for personalised routing and travel assistance with new or refined features.

Section 2.1 summarises the general concept for personalised routing and travel assistance. In Section 2.2 extensions and new developments in the filed of personalisation, conceptualised and implemented for the STREETLIFE pilots during Y2, are depicted. Developments in the filed of travel assistance, including tracking as a basic functionality, are described in Section 2.3. In STREETLIFE the general concept for personalised routing and travel assistance is instantiated by four routing services, fulfilling specific characteristics and motivations of the different pilots. The basic approaches were discussed and can be looked up in D5.2.1. The last two sections put a focus on how the concept for personalised routing and travel assistance can be deployed as a back-end service based (Section 2.4), or a client based system (Section 2.5).

2.1. General concept for personalised routing and travel assistance

This section briefly summarises the general concept for personalised routing and travel assistance. For a complete description, please refer to D5.2.1. The proposed concept consists of 3 fundamental mechanisms: an intermodal journey planner (IJP), a personalisation plan, and a travel assistance plan.

At the European context each municipality has its own solution as the IJP, which is the very reason that STREETLIFE project includes four implementations. It is shown in the project that different IJP implementations can be integrated in the overall integrated application and IJP results can be personalised according to user preferences with the help of pre- and post-processing techniques and develop advanced Travel Assistance solutions. Technical implementations and usability beyond the original scope differ from router to router. LOGICA Tampere and VMZ Berlin router solutions are running totally at the server side and commercially acquired by municipalities and therefore commercially exploitable. VMZ Berlin offers multi-modal routing services for public transportation, car, bicycle and walking. For this purpose external routing services from the regional public transport association (VBB), TomTom, Google and bike are already in place and used. The modal routers are integrated in one VMZ router, to generate multi-modal trip planning results. AALTO solution is Open Source, lightweight and implemented mainly on the client side utilising OpenStreetMap data. This is transferrable to other municipalities as well as other Open Source solutions such as OpenTripPlanner (OTP), which is used in Rovereto.

An *IJP* can provide travellers with itineraries involving two or more means of transportation whereas unimodal suggestions can also occur. The approach allows both the integration of 3rd party services or own journey planner modules. Using 3rd party software can minimise implementation expenses and allow for other priorities within the routing concept. The BER and the ROV pilots utilise a 3rd party IJP and LOGICA's TRE development utilize LOGICA's IJP. Own implementations allow e.g. for modifying the underlying algorithm independently and client side routing. The TRE development by AALTO implements an own IJP.

The *personalisation plan* specifies how personal *mobility recommendations* are made out of *routing proposals*. Routing proposals represent the output of the IJP. Mobility recommendations can e.g. be generated by filtering and reordering of routing proposals. First, we distinguish between two personalisation approaches, namely *pre-* and *post-personalisation*, which can be used individually, in combination or partly integrated in the routing process. In principle pre-and post-personalisation can be utilised with 3rd party routing and with own implementations. Own implementations further offer the possibility of directly affecting the weights of the graph during routing by personalisation options. We call this way of personalisation *integrated personalisation*. Figure 1 depicts the possibilities of personalisation within the general routing process.

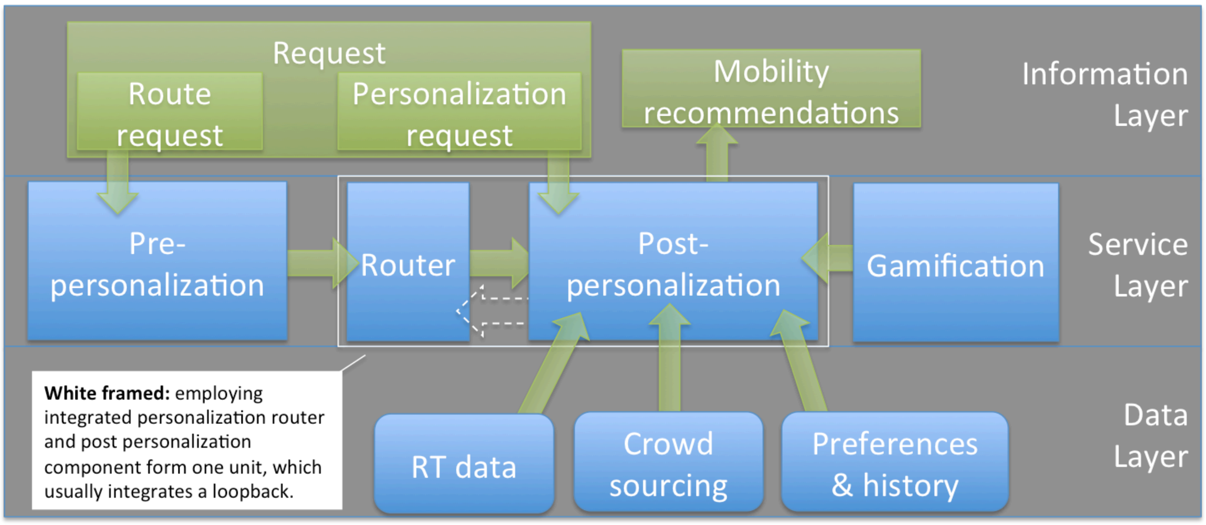


Figure 1: Personalisation within the general routing process

The *travel assistance plan* specifies how to define, control and adapt currently taken routes. In this sense the plan concerns the entire routing process, ranging from the specification of travel preferences to the arrival at the final destination. Travel assistance encompasses the employment of an IJP and a personalisation plan for *defining* an appropriate itinerary. Additionally it specifies a *monitoring* process to control the actual route and an *adaption* process to re-calculate the itinerary if an impairment occurs. Regarding the adaption of routes two concepts, namely *confirmed rerouting* and *automatic rerouting* are proposed.

2.2. Extensions and developments in the field of Personalisation

2.2.1. Incorporation of policies in journey planner

As part of the concept and framework for route planning and travel assistance in STREETLIFE, we have conceived an extension of the framework in Figure *1*, which aims at modifying and augmenting mobility recommendations not only responding to the preferences and profile of the single user (i.e. personalisation), but also taking into account the sustainable mobility policies of the Smart City and its administration.

For that reason, FBK has developed and experimented with - in the context for now of the IJP for ROV - an additional component that can be programmed to represent preferential types of itineraries, due to city policies, and which can be used in the post-processing of the mobility recommendations output by the router component. This module is called a “programmable policy recommender”, and it is shown in Figure 2.

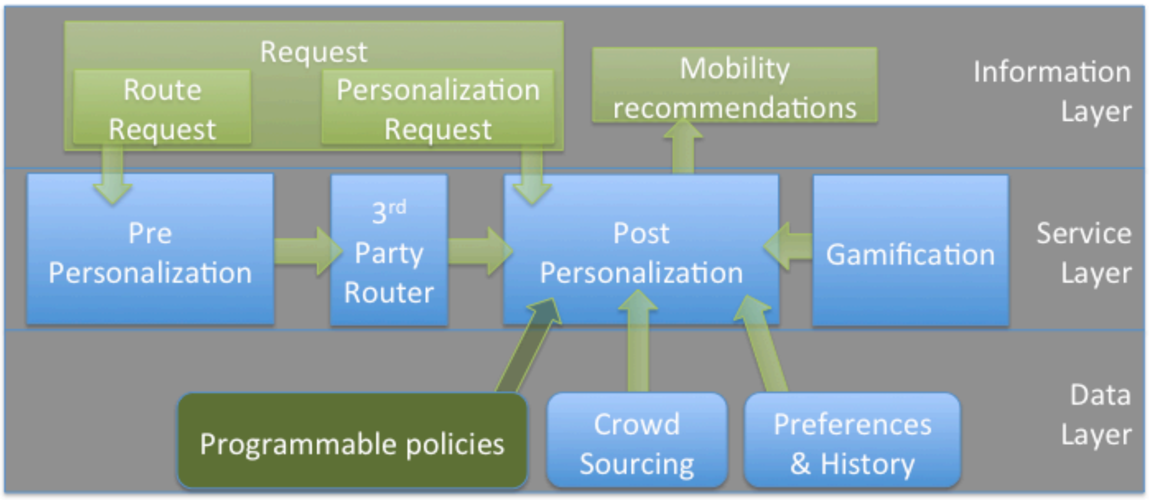


Figure 2: Modifying route recommendation based on programmable policies recommender

The module assumes that itineraries output by the routing component, including a third-party router like OTP, can be evaluated and annotated according to one or more dimensions, such as:

- A comprehensive estimate of cost, which includes a variety of factors, including parking costs, public transport fees, fuel consumption, etc.;
- A comprehensive estimate of time, which includes besides travel time and layover time also other factors, such time overhead for parking search, etc.)
- Environmental impact, i.e., carbon emissions estimate;
- Health factors (e.g. steps taken, calories consumed, etc.)

Information on the dimensions above can, on the one hand, be passed directly to the final user, to enable her to make a more informed personal decision on the itinerary she wants to choose. On the other hand, however, they can be passed to the programmable policy recommender, which matches the annotated itineraries according to policy specifications that have been configured into it. The results of this matching process are a short list of itineraries that best fit the priorities set by the policies; that short list can be highlighted or otherwise presented in preferential ways when itinerary recommendations are returned to the end user. The classic example, which was already tested in the first iteration of the ROV pilot, in which an early prototype of the programmable policy recommender was validated, is the highlighting of the “greenest” itineraries available between origin and destination on the screen of the user’s routing App.

However, the usage of programmable policies is not limited to that, and it does not even necessarily rely strictly on the availability of an estimation of quantitative factors related to an itinerary, like in the examples above. Preferential rules that can be injected in the programmable policy recommender can include also other factors: for example, the city may want to promote itineraries that suggest the use of a specific mobility service, or that encourage using a public transport route, as opposed to another, to reach the same destination at certain critical times during the day, etc.

The net effect of being able to program city-specific policies in the process of route planning is that this provides an ability to induce a high degree of engagement in the citizens who use the smart and sustainable mobility information system of STREETLIFE, with respect to citywide objectives.

It is noticeable how there is interplay to be considered between the programmable policy recommender and the personalisation component. They both work as pre- or post-processing facilities on top of a routing component; however, the concerns according to which they process, filter and rank itineraries may be very different and even inconsistent. As a result, it is important to avoid the kind of inconsistency where itineraries chosen and highlighted because policy-based recommendations are in direct opposition to the preferences stated in the profile of a user, which instead lead to totally different recommendations, and hence remain unacceptable. We have found, therefore, that is important to *modulate the policy-based recommendation according to the user's personal preferences*. That means that the itineraries that are ultimately chosen for policy-based highlighting must also remain acceptable to a degree according to the user's preferences.

For example, a user that never wants to use bikes (perhaps because of physical fatigue reasons, or own safety concerns) should not be presented with a policy recommended itinerary that requires biking even if it represents the greenest solution as required by the city policy; alternative green itineraries should be selected and highlighted instead, which do not involve bike riding. Conversely, there are some personal preferences that can be stretched; for instance, even if the only criterion that a user has set in her profile is to find the quickest possible itineraries, we have found that is appropriate for the programming policy recommender to suggest and highlight itineraries that take some time longer, but respond to policy concerns, such as involving more sustainable transport modes.

2.2.2. Interaction mode based personalisation

We conceptualised an extension of the user interface of mobility apps by automatic speech recognition enabling speech-based routing requests of the user, and by speech synthesis for the generation of speech-based system feedback. As part of the pre-personalisation concept, the extension, developed by DFKI, aims improving the efficiency of the user interface by personalised speech-based route requests. Interaction mode based personalisation is implemented into the BER mobility app. Using speech input the users routing requests can be pre-personalised in the following ways:

- The departure time is automatically set to now, making it unnecessary to make any indications regarding the travel time. This increases the overall efficiency of the routing request task.
- Shortcuts like "take me home" are possible. This further increases task efficiency and reduces the probability of automatic speech recognition errors.

Speech technology more and more enters our everyday life and more and more users intend to use these forms of interaction in modern Apps. Services like "Ok Google" [1], Apple's Siri [2], or the Android speech recognition API[1] are already well known and make it easy to render formerly graphical user interfaces (GUI) multimodal. For many applications speech input is advantageous in terms of efficiency, as e.g. search requests can easily be performed by spoken utterances [3] and spoken dialog often results in benefits regarding interaction time or task steps [4].

For the implementation of the STREETLIFE BER App we used the Mobile Multimodal Interaction and Rendering (MMIR)[2] framework, which is an Open Source tool for building lightweight multimodal dialog systems [5]. MMIR enables a stepwise integration of single interaction modalities. This allowed us to create a purely touchscreen-based system integrating basic functionalities of mobility apps during the first project year, and to extend the system with speech input during the second year. Before the speech technology integration, we conducted a user study gathering feedback about the suitability of speech input and expected system feedback, including speech. The study revealed that speech interaction is mostly requested for the routing request task.

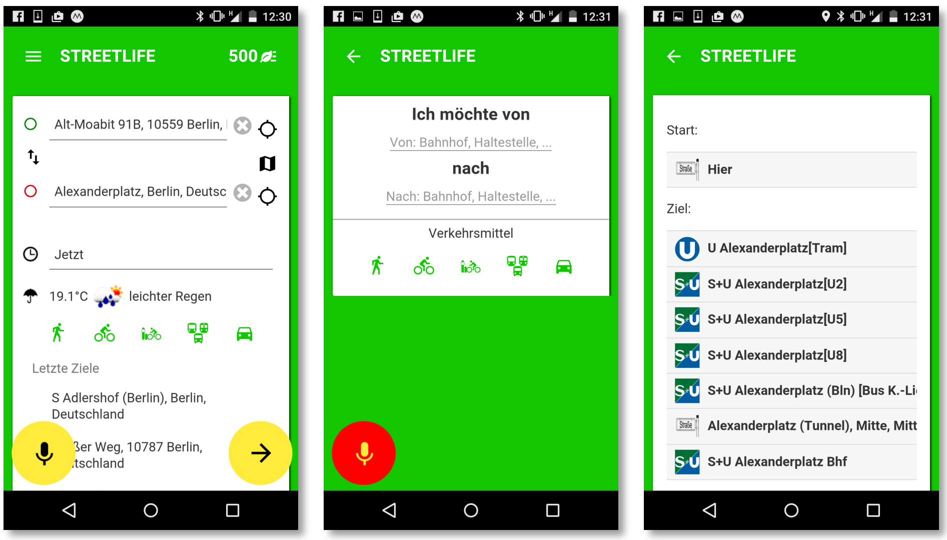

12:30
STREETLIFE
500
Alt-Moabit 91B, 10559 Berlin,
Alexanderplatz, Berlin, Deutsc
Jetzt
19.1°C
leichter Regen
Letzte Ziele
S Adlershof (Berlin), Berlin, Deutschland
ßer Weg, 10787 Berlin, tschland
12:31
STREETLIFE
Ich möchte von
Von: Bahnhof, Haltestelle, ...
nach
Nach: Bahnhof, Haltestelle, ...
Verkehrsmittel
12:31
STREETLIFE
Start:
Hier
Ziel:
U Alexanderplatz[Tram]
S+U Alexanderplatz[U2]
S+U Alexanderplatz[U5]
S+U Alexanderplatz[U8]
S+U Alexanderplatz (Bln) [Bus K.-Li
Alexanderplatz (Tunnel), Mitte, Mitt
S+U Alexanderplatz Bhf

Figure 3: Interaction mode based personalisation. Left: conventional routing interface with microphone button. Middle: "conversational interface" suggesting an utterance for speech input. Right: proposed route.

Due to the fact that the grammar includes around 10,000 entries of locations in the city of Berlin, all of which proper names (i.e. names of streets and stops), the speech input language is German only. Figure 3 (middle) shows a "conversational interface" that was designed to indicate a valid input utterance to the user. It is shown if the microphone button in Figure 3

[1] http://developer.android.com
[2] http://github.com/mmig/mmir

(left) is touched. The GUI labels "Ich möchte... von... nach" (engl.: "I want... from... to") reflect the utterance that can be optimally recognised by the grammar. Parallel to the speech input, optional information regarding desired traffic modes and preferences for the routing request can be made. Routing information from speech input and touchscreen are fused into one routing request. Figure 3 (right) shows an additional view for ambiguity resolution. It is presented if more then one location was found in the used database. The example shows the case where multiple entries for the destination were found.

2.2.3. Device based personalisation

The eventual aim of STREETLIFE is to shift the mobility behaviour of the users towards greener mobility behaviour. The BER pilot mobility app e.g. should motivate citizens to an increasing bicycle usage. In order to simplify the usability of mobility services for bicyclists, DFKI conceptualised a user interface extension for the BER mobility app by a wearable device, namely a smart watch. Similar to the interaction mode-based personalisation, the extension aims at improving the efficiency of the user interface. As part of the pre-personalisation concept, highly personalised mobility recommendations can be requested directly via the smart watch without the necessity of putting the smartphone out of the pocket. Due to the limited screen size of the smart watch device the requests are not generated by touch-screen based interaction, instead the user can request routes via speech input, which is also the interaction mode proposed by Android wear[3]. Additionally to the pre-personalisation options depicted in the previous section, the routing requests can be further personalised in the following ways:

- The traffic type is automatically set to “bicycle” only. Consequently only one itinerary for cycling is returned. Due to the limited screen size the smart watch is hardly suitable for displaying a greater number of different itineraries. However, the smart watch device is well suited to visualise a single personalised mobility recommendation.
- The origin is set to the actual location.

Figure 4 shows three screenshots of the smart watch extension. The left screenshot displays the routing interface. A microphone button and the textual label “Wohin möchtest Du?” (Engl.: Where to go?) convey that a destination can now directly be specified by speech input. The user can e.g. say “Ich möchte zum Alexanderplatz!” (engl.: I want to go to Alexanderplatz). The destination is determined from the spoken utterance and combined with the personalised route information. In this way a request formed that is sent to the IJP. The screenshot in the middle illustrates how an overview of the routing result is displayed on the smart watch. The actual position is illustrated by a blue dot on the map. After pressing the navigation button (right) the display changes to the navigation view shown in the right screenshot. The actual position is centred. In the screenshot the actual position is visualised close to the origin. In the navigation view the route is displayed in black. Further a textual label shows the actual distance to the destination.

[3] https://developer.android.com/wear

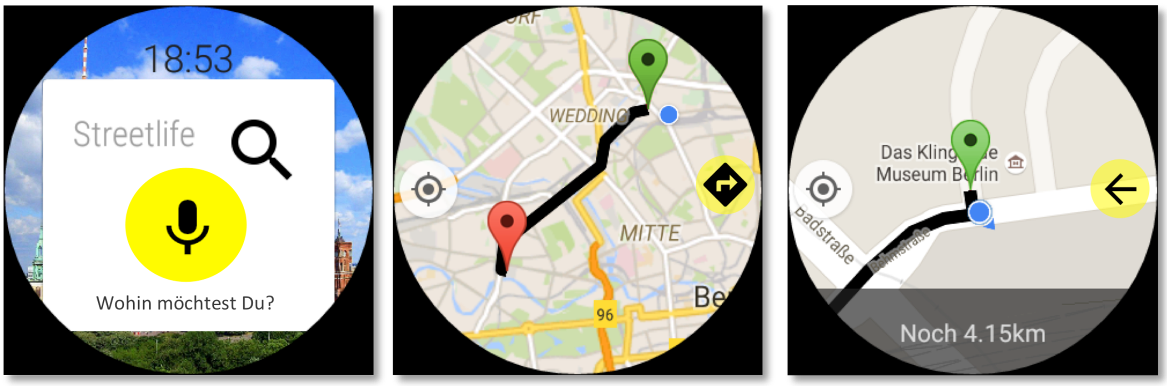


Figure 4: Smart-watch based routing

In summary the smart watch extension allows for highly per-personalised routing requests, as the user only has to state a destination. Other request parameters like origin, travel time, or traffic mode are automatically derived from the context of usage. In has to be noted that the smart watch interface directly utilises already implemented app functionalities for grammar parsing and the interface to the routing service.

2.2.4. User preferences based personalisation

With respect to an increasing number of personalisation options, resulting from available data sources and intermodal travelling, it is recommend to focus on specific user groups or app purpose [35]. For STREETLIFE the consideration of such design principles is highly relevant in order to ensure the usability of the emerging applications. With regard to the bicyclist focus of the BER pilot the IJP (provided by the BER partner VMZ) is updated with new routing constraints for bicycle routes. Users will be able to state their preference for safe, fast, and comfortable bicycle routes. Corresponding UI elements are integrated in the BER mobility app developed by DFKI. Figure *5* shows a mock-up illustrating how the integration of features for personalised bicyclist routing. The features will be realised as pre-personalisation options of the IJP.

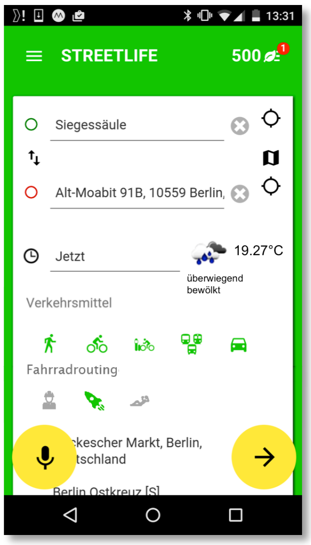


Figure 5: Integration of preferences for bicyclists

2.3. Developments in travel assistance

2.3.1. Tracking as a basic functionality for travel assistance features

As depicted in Section 2.1 travel assistance is an iterating process of defining, monitoring, and adapting routes. The definition of a route is accomplished by utilising the IJP in order to obtain a number of suggested itineraries, and by selecting one of these itineraries. In order to provide further travel assistance features while the user is on the route it is necessary to track the actual route. Active routes can be correlated with a number of data sources including real time and crowdsourcing data, see STREETLIFE Deliverable D3.2.2, as well as data regarding the selected itinerary like transport changes or delays. Tracking data can also be utilised in the context of Gamification in order to validate game relevant information and to issue the correct amount of points or other incentives.

In order to optimally support eventual requirements of all pilots we conceptualised a tracking component that makes use of as many smartphone sensors as possible. The component is implemented as a part of the BER pilot mobility app, developed by DFKI, and can be delivered to other pilots as Android code. The proposed solution covers the following data:

- The activity of the user classified by the Android activity recognition[4]. Possible values are: IN_VEHICLE, ON_BYCICLE, ON_FOOT, RUNNING, STILL, TILTING, UNKNOWN, WALKING. These values can e.g. be used as a first approximation for mode detection or for the validation of bicycle trips.
- The position of the user including: altitude, direction, latitude, longitude and speed.
- Radio mast data including: Cell-ID, location area code, primary scrambling code
- Accelerometer, including: minimum, maximum, and average acceleration as well as standard deviation of the average acceleration at fixed intervals

The tracking can be triggered by any event of an app integrating the tracking component. In the case of the BER pilot app the tracking is started when the users activates the companion mode for a specific itinerary. The tracking ends if the user manually stops the companion mode or if the end-of-route detection mechanism depicted in the next section automatically detects that the destination has been reached. The tracking data is stored in a CSV file on the user's device. At the end of a route, the stored data is sent to a server component.

2.3.2. End-of-route detection

One application of tracking is end of route detection. The endpoints of itineraries should be automatically detected for two reasons: (1) the tracking has to be turned off as it consumes the battery of the users device, and the user might forget to quit the companion mode manually; (2) at the end of the route the user gets notified, as feedback about specific characteristics of the route will be asked in the context of crowdsourcing. In order to detect the endpoint the current position of the user can be compared to the target position of the route by calculating

[4] https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognition

the distance between the two points. If this distance is below a certain value (e.g. 20 meters), the arrival at the destination is recognised and the user gets a notification. The notification supports both the smart phone and the smart watch. Figure 6 illustrates a smart watch notification for end-of-route detection implemented by DFKI for the BER pilot mobility app. After pressing the check mark button the user is asked to switch to the smart phone in order to give feedback to the crowdsourcing questions.

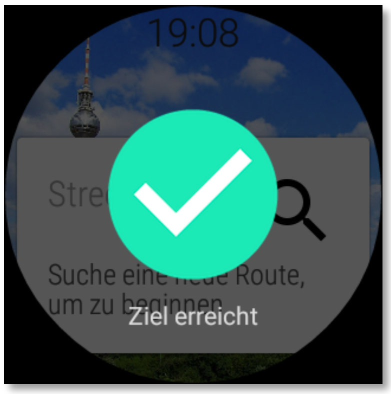


Figure 6: End-of-route notification on the smart watch

2.3.3. Route deviation recognition

In order to implement the last step in the sequence of defining, monitoring, and adapting routes we conceptualised the route deviation recognition. The aim of the route deviation recognition, which is a further application of the tracking mechanism, is to warn the users that the route is being left. The consequence of a route deviation can be that tracking is switched off and that a re-routing might be triggered if desired by the user.

In order to recognise a deviation the current position of the user is compared with the positions of the waypoints of the selected itinerary by calculating distances for all waypoints. For the shortest distance it is checked whether this it is below a certain threshold (e.g., 200 meters). If the value exceeds the threshold, the user receives a notification. Figure 7 illustrates a smart watch notification for route deviation recognition implemented by DFKI for the BER pilot mobility app. After pressing the checkmark button the user can trigger a re-routing.

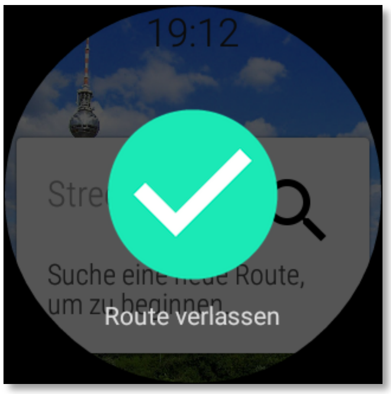


Figure 7: Integration of preferences for bicyclists

2.4. Example of a back-end service based instantiation of the concept for Travel assistance and multimodal routing

2.4.1. Background

At the time of the launch of STREETLIFE project, Tampere was offering rather modern, and amongst Finnish cities, a competitive set of Internet based travel assistance services, including a multimodal routing service. In some areas, such as real time tracking and traffic light priority of buses, Tampere was and still is a technology leader in Finland. However, utilisation of the new technology was not yet best possible. Their two systems and parking information were not integrated and therefore there were possibilities to give better and more accurate information to travellers. Also, the existing routing service user interface was not meeting today's needs and expectations for high quality mobile services.

From this background, a natural goal for the STREETLIFE Tampere pilot, initiated to boost and ensure development towards efficient public transit services leading to carbon reduction through user behaviour change, was easy to set. In STREETLIFE Tampere pilot integration of real-time data sources where used to provide state of art real-time routing service, which would utilise the existing IT infrastructure and provide a mobile-friendly end-user application. The multimodal routing service will be able to route all transit model taking into account real time events regarding bus arrival estimates and delays or cancellations. In addition real-time router will check in park and ride mode the availability of parking places in park and ride locations.

To launch the first pilot period in October 2014 required data source integrations, testing, interoperability testing and configuring a new mobile STREETLIFE Tampere pilot application. However, a mobile real-time multimodal journey planner is new to users; expectations were that feedback from test users, surveys and other findings under real production use would lead to further developments.

2.4.2. Data sources

The integration of existing IT infrastructure and data sources has been successful. Travel assistance and multimodal routing service uses pre-processing method where real-time information is taken into account in real-time when user searches the route. This way important information is included automatically in routing results, for example what is the availability trend of park and ride location's parking places. The CO_2 footprint calculation uses post-processing method. Today, STREETLIFE Tampere pilot uses data from the following sources:

- Road and street network
- National road network
- Maps of Finland
- Tampere street addresses
- Bicycling network
- Public transport timetables and routes (Buses and trains)
- Route planning API

- Tampere public transport SIRI
- Parking halls
- Parking hall status
- User preferences
- CO_2 emission information per vehicle type
- User location tracking

2.4.3. Performance and reliability

The backend service architecture is designed for high reliability for real-time systems. So far, there has been no downtime at all, and the service is running smoothly. Instantaneous service response ensures positive user experience and helps to make the service popular.

Solution is redundant. If real-time data sources have had service breaks, it does not cripple the STREETLIFE Tampere pilot service, because it uses static data in absence of up-to-date real-time data.

Most feedback from test users has been related to the front-end application. It has become obvious, that today's users demand high quality; an application must be intuitive to use and work smoothly; otherwise it might get rejected during the very first trial session. The development group has reacted to the feedback and the application can now be considered as mature. Especially presentation of real-time information has developed a lot, so that the traveller now understands what is real-time and what is not.

2.4.4. Usage

The STREETLIFE Tampere multimodal routing service is targeted mainly for early adopters and selected test user groups. These have been sufficient for collecting the user feedback. The new service is not yet promoted as main service, neither presented as the primary routing service for Tampere. This policy of the Tampere city may change in the near future after STREETLIFE Tampere pilot, by the maturation of the new service. Tampere Transport authorities see the potential of this new improved service.

2.4.5. New development

One new major feature, gamification of the front-end app, is currently under development. The application will include a game called 'Zonehunter', where travellers can collect badges and trophies by visiting as many postal zones in the traffic region as possible. The goal is to motivate citizens to use public transport in many various trip types, not only for some regular routine travelling such as the daily work trips.

The game feature will be published first for a limited test user group at the start of the second pilot period, November 2015.

2.4.6. Future prospects

The STREETLIFE Tampere pilot appears to be a natural successor for the traditional public transit routing services, thus providing a modern service development and maintenance platform for the city. The new capabilities of the service, such as traffic flow control by stop goodness values, will be tested during the second pilot iteration, and may become options for the actual production services in the future after STRETLIFE Tampere pilot.

2.5. Example of a client based instantiation of the concept for Travel assistance and multimodal routing

AALTO's TRE Open Source routing service is designed for lightweight mobile use, with support for client-side routing and off-line situations. It integrates directly with the research on real-time data management (D3.2.2, Section 5) via the shared OpenStreetMap based World Graph data model and allows wider crowd-sourced development by Open Source licensing. For example, one can create alternate optimisation targets within the engine by modifying the weights of the underlying graph, i.e. toward greener behaviour.

AALTO's routing implementation depends on open data formats, built on OpenStreetMap and GTFS. Targeting multiple platforms with small memory footprint, pre-processing the raw data to an efficient storage and transmission format is the main stage. This stage utilises the full pipeline of *OSMSqueeze*, described in D5.2.1. To support multiple platforms, including web browsers, during Y2 we have produced an implementation with JavaScript.

LocalRoute.js (lr.js) calculates public transport routes on web browsers and mobile devices, online and offline. Map, route and timetable information from open data sources is pre-processed and compressed into a custom, *squeezed* format. These comparatively tiny files are delivered to the client and decompressed before use. Figure 8 presents the data pipeline.

LocalRoute.js guarantees routing availability and zero unexpected costs also abroad, and maintains the user's privacy.

- Supported timetable formats: GTFS, Kalkati.
- Input map data format: PBF-compressed OpenStreetMap.

On Android, lr.js is evoked via Java Native Interface, JNI, from Native side.

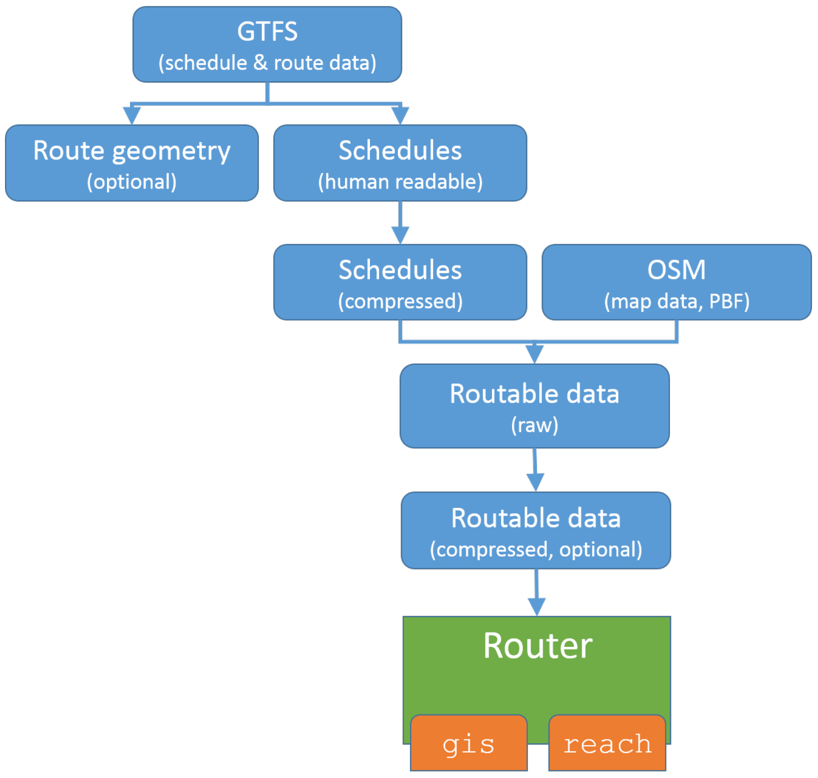


Figure 8: Map and schedule data pipeline for routing

Step 1: Pre-process schedule data.

```
node lr.js --date 2013-12-02 --in-gtfs tampere/gtfs.zip \
           --out-tempt tampere/readable.txt --out-gtfs-geom
tampere/geom.txt
```

The interval of dates is 30 days starting from and including the given date parameter.

This pre-processes GTFS data in `tampere/gtfs.zip`, stores (somewhat) human-readable transit schedules in `tampere/readable.txt` and (if the last, optional argument is passed) compresses original route polyline coordinates into `tampere/geom.txt`.

Step 2: Compress schedule data.

```
node lr.js --in-tempt tampere/readable.txt --out-trans tampere/trans.txt
```

Step 3: Pre-process map data.

```
node lr.js --in-tempt tampere/readable.txt --in-pbf tampere/osm.pbf \
           --out-map tampere/map-big.txt
```

This reads the previously stored transit data to get the coordinates of transit stops used to guess the area relevant to routing. Then it extracts the map data in PBF format, applies basic compression and stores it in `tampere-map.txt`.

Step 4: (Optionally) compress map data.

```
node lr.js --in-map tampere/map-big.txt --out-map tampere/map.txt \
           --map-round 5 --compress-map
```

Note that the parameter `--compress-map` needs to be last.

Step 5: Calculate routes! (Debug output for now)

```
node lr.js -M tampere/map.txt -T tampere/trans.txt \
           -f 60.1688,24.9412 -t 60.3093,24.5141 \
           -D 2013-12-03 -d 08:00
```

2.5.1. Library structure

API uses two global objects, `gis` and `reach`. `gis` has some general-purpose functions, while `reach` is related to public transit. The main class to access data is `reach.trans.TransSet`. It has members `stopSet`, `lineSet` and `tripSet` for all public transit stops, lines and departures.

In `localroute.js` terminology a *trip* represents a single time that a vehicle goes from the first to the last stop of its route in one direction. Therefore a trip contains a single arrival for all stops along the route. All information the public uses to identify the route is connected to every trip.

A *line* is the list of stops passed along a vehicle's route, and in the future also its geometry. It has no other information.

2.5.2. Structure of compressed data

- List of strings (names and codes), LZ77 compressed.
- Stop IDs, names (references to list of strings) and coordinates, delta encoded.
- Lines, encoded so that stops previously seen following the same predecessor are stored as indices into the predecessor's follower list.
- Keys: line names, codes and head signs stored as references to list of strings.
- Trips, grouped by valid day mask, delta encoded, LZ77 compressed.

2.5.3. Next Steps

`LocalRoute.js` utilizes a given map data for routing (the `reach` object) and other GIS operations such as mapping to closest street segment (the `gis` object). As the compressed data is very lightweight, running this routing on web browsers, or in our case, in an Android tablet, does not consume much resources. However, for AALTO's dynamic entity management using a World Graph, we need to implement one part of the software in C++ for World Graph related functionalities.

3. Virtual Mobility (T5.2)

3.1. Introduction

The idea of Virtual Mobility is to experience the travelled routes virtually instead of physically. The objective of Virtual Mobility is to remove any mental obstacles that are often associated with sustainable transport modes. Major mental obstacles are caused by multimodal public transport chains when the route and perhaps the city is unknown to the user as locating the right transport and direction can prevent people choosing public transport. Biking, on the other hand may be felt unsafe if the cycling routes are not available or the user does not know the biking route, which would also be difficult to verify once the journey has begun.

For above-mentioned reasons new ICT tools and techniques are used to create the pre-experienced feeling of the route. Users can pre-experience the trips already by now using 2D maps with visualisation of the multimodal routes, but implementation in Virtual Reality should be more effective and remove possible mental obstacles more efficiently.

Since the realistic Virtual Reality equipment are not yet readily available for consumers field tests with a limited number of users will be conducted instead of large-scale user testing. The main goals of the field tests are to test several hypotheses including the following:

- Route pre-experiencing is better with Virtual Reality approach
- User is more confident that (s)he can take the route an IJP proposes
- User is more prone to choose sustainable transport modes with pre-experiencing the route with Virtual Mobility due to spatial knowledge transfer

Field tests are run separately from the pilots evaluation period since the participants are different, the participants cannot use their own devices and field test design is done in interaction with the actual state of the technical capabilities of the system. The main goal is to get the technical capabilities to the maximum achievable during STREETLIFE before the actual field tests and therefore their timing is left to the end of the project, which differs slightly from STREETLIFE pilot time periods.

3.2. Oculus Rift Head-mounted display

As part of virtual mobility and AALTO's implementation plan, we have implemented Oculus Rift support for 3D city visualisation on desktops. This work stems from our hypothesis of immersive views to facilitate better or more comprehensive pre-experiencing of a travel plan than abstract representations. Immersion on Oculus is created by feeding rendered output to two separate parts of the display, each representing one eye, with pupillary distances matching those of human viewers. So, in the rendering engine, two separate renderings of the same scene are provided, with two virtual cameras. Figure 9 presents output suited for Oculus. The main tracking component is an Inertial Measurement Unit (IMU), which registers head orientation and reports that to a rendering engine via Oculus SDK APIs.

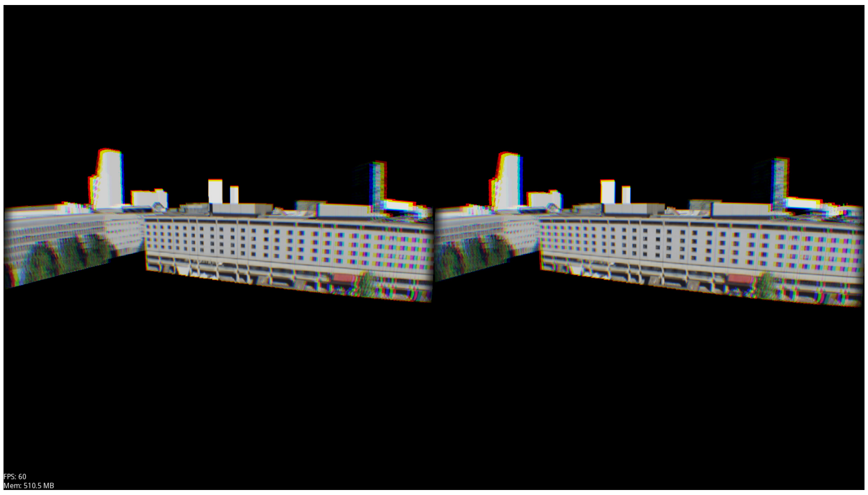


Figure 9: Oculus Rift's display constitutes two parts, one for each eye, for stereoscopic viewing (from Berlin 3D model)

While Oculus is not yet a commercial product, the second version, DK2, is greatly improved from DK1. Table 2 demonstrates the main differences between the two. As Oculus renders separate images for both eyes using a single display, the actual resolution observed by users is half of the screen resolution – 640x400 and 960x540, for DK1 and DK2, respectively. As a new feature in DK2, it allows translational tracking via infrared sensors. Even with the small motions of head in a desktop set-up, this allows more realistic feel to the presence. The most critical feature, however, is probably latency. The most common annoyance in virtual, immersive environments is *cybersickness*, which stems from the difference between sensed head motion and observed visual input. The main cause for this is latency between external sensory systems and rendering engine, although slow rendering will also increase latency. Improvement of both sensory latency and display refresh rate were most welcome for DK2.

Our implementation relies on OpenGL for rasterisation, and Oculus SDK APIs, with C++ as the programming language. Our development environment is Linux. Unfortunately, Oculus has paused support for it (and Apple's OS X) "*in order to focus on delivering a high quality consumer-level VR experience at launch across hardware, software, and content on Windows*" [6]. We have to rely on the last version of the SDK that had Linux support, v0.4.3. This is effectively stalling our DK2 dependent parts of our development. Basically, we cannot implement translational tracking and cannot enjoy the better quality of the device.

Table 2: Differences between Oculus Rift DK1 and DK2

Feature	**DK1**	**DK2**
Screen Resolution	1280 x 800	1920 x 1080
OLED	NO	YES
Screen Size	7"	5.7"
Screen Manufacturer and model	Innolux HJ070IA-02D 7″ LCD	Samsung Galaxy Note 3

Latency	50ms – 60ms	20-40ms
Low Persistence	NO	YES
Refresh Rate	60Hz	75Hz
Orientation Tracking	YES	YES
Positional Tracking	NO	YES
Gyroscope, Accelerometer, Magnetometer	YES	YES
Field-of-view	110	100

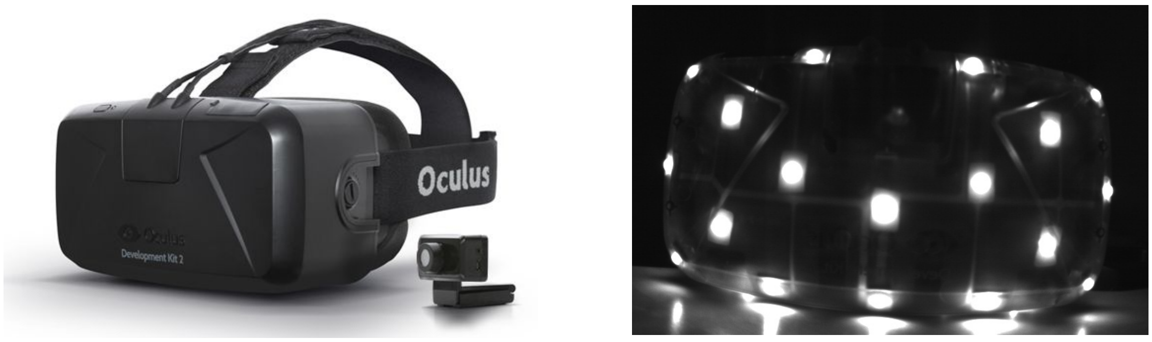


Figure 10: The Oculus Rift DK2 head-mounted display with the infrared sensor (left) and the front of DK2 with infrared LENs active (right)

3.3. Support for spherical images

Oculus Rift requires a realistic 3D model for providing immersive views with a strong feeling of presence. However, such artificially constructed and computer-generated imagery may suffer from various issues in quality, which would lessen the experience. An alternate way of easily producing credible environments is 360º (4π) photography, which registers the full view of actual environment in a single image. Although moving from such "bubble" to another will evidently be less than realistic, the feeling of presence could be strong for these individual points.

We have earlier acquired about 50 spherical photos from Tampere as our test case using the Ricoh THETA camera in Figure 11. We have now developed support for viewing them both on Oculus and Android tablets.

In Oculus, which relies on stereoscopic rendering, these single images cannot portray depth. Therefore, much of Oculus's potential is lost. For users, the spherical image is exactly that: although it is experienced in an immersive manner, the surface is flat and far away. For any objects or structures that were close to camera at the moment of photography, the flattening effect is emphasised. However, if no objects were close, the effect is quite acceptable. Naturally, translation tracking of DK2 would not be able reproduce any parallax motion.

Implementation for Oculus involved the creation of a spherical surface around the viewpoint and applying THETA images to that as texture. We also needed to extract the orientation data from the EXIF metadata.

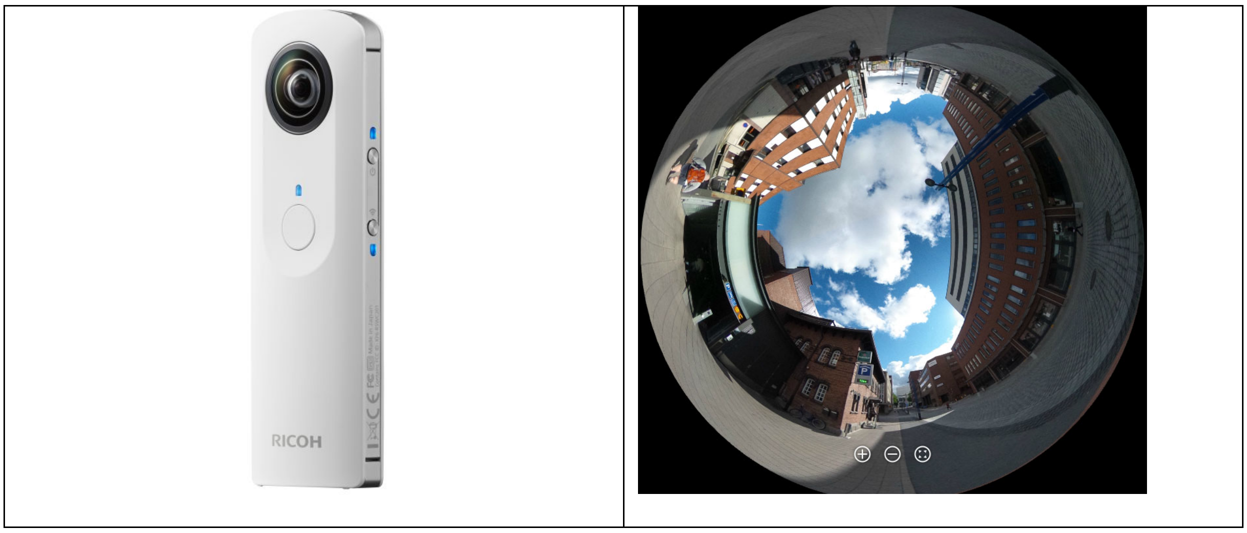


Figure 11: The Ricoh THETA spherical camera and a scene from Tampere

For Android, we adapted the natural way of viewing our surroundings, allowing the tablet's orientation to be tracked with its internal IMU unit, and rendering the spherical image accordingly. Again, EXIF metadata provides the original orientation. The field of view for rendering the image can be altered for a zoom-like effect. However, if the field of view differs significantly from the natural field of view of the device's screen in a normal viewing position, the natural "window" effect is diminished. A user enters the spherical image in the 3D map by simply moving the viewpoint

Implementation for Android included spherical surface and texturing generation at Android's native side with OpenGL, and use of Android's IMU API for tracking.

Automated addition of points-of-interests to spherical images depends highly on the accuracy of positioning and orientation of the device. In Chapter 5 we discuss issues related to quality and robustness.

3.4. Next steps

For the Oculus Rift, AALTO will continue the development using the latest Linux compatible SDK in hopes of Oculus to return to Linux support soon after their product release. We will aim at conducting a spatial knowledge transfer experiment during the last Pilot phase.

As of September 2015, a new version of the THETA has been released. The *Theta S* is a "high-spec" model, improving on all aspects of the original version, with direct support for

Google Street View app[5]. Although the advertised specifications do not address the accuracy of the orientation, we will acquire one Theta S in hopes of better IMU implementation.

Figure 12: The updated Ricoh THETA S spherical camera

4. Citizen Participation and Gamification (T5.3)

4.1. Gamification Engine: current release

In Y2 of the project, we have continued the development of a Gamification Engine (GE in the remainder). An early prototype of the GE, and its general design, were already discussed in D5.2.1 [7]. That prototype was deployed and validated within the Rovereto pilot during its first iteration. The validation process and its results are reported in D6.2.1 [8], and some of the highlights have also been published in [9].

Based on our Y1 experience, we have continued to enhance the GE. Below, we report on the latest advancements, including its current design, its main features, and the deployment options that are available at this stage for its integration, either in the context of the existing pilots, or to external parties that want to incorporate gamification features in an instance of the STREETLIFE sustainable mobility system, or -- more generally speaking -- in any Smart Cities or Smart Mobility System.

[5] https://theta360.com/en/about/theta/s.html, last retrieved September 22nd 2015.

4.1.1. Current GE Design

The main design principle of the GE remains the same as had been discussed in D5.2.1. The design principle is based on a service-based approach that brings about the following benefits:

- The extensibility of the set of game concepts that can be used in specific gamification applications;
- The integration with open and heterogeneous and multiple-ownership systems (including legacy) that are expected to be operating in a Smart City, and which need to interact within the gamification applications;
- The standardisation of interactions that occur between GE and players through Apps and other frontends.

With respect to the prototypical design reported in D5.2.1, the design of the GE has evolved and expanded. It now includes a larger number of software components and service-based functional interfaces among them as well as support for external systems and user Apps displayed in Figure *13*. The GE also includes some design-time facilities for configuring and deploying games, in addition to runtime facilities for the instantiation and execution of deployed games.

Design-time elements in the current GE release include:

- The **GameService**, a service, which contains the specification of all games, developed and ready to be instantiated. Each game is specified as a package that includes:
    - A set of general game meta-data: each game is characterised by: a unique game **id**; a mnemonic game **name**; a game **owner**, responsible for the maintenance and management of the game; and, most importantly, a **game model**, which specifies the **gamification concepts** used in the given game, which compose the game state (for instance, different kinds of points, different badge collections, different kinds of leader boards, etc.); each concept has its own specific service-based interface that enables the game rules that predicate upon that concept to modify that part of the game state, in compliance with its game semantics and according to concept-specific logic;
    - A set of ECA **rules** associated to the game, which codify the game logic, and operate with service-based operations on the game state;
    - A set of **tasks** associated to the game; tasks are periodic or one-shot operations that automatically take action on the state of the game; for example, initialisation tasks for players and game state, or tabulation task for the construction and maintenance of leader boards and other views over the game state, etc.

- The **game console**, which provides the game owner with a facility to package, deploy and install the game specifications for a given game onto the GameService; in the current release of the GE, the game console is an interactive Web Application;
- The **game API**, which provides a service-based interface for the GameService, with operations that are used by the game console for orchestrating the deployment and installation of games

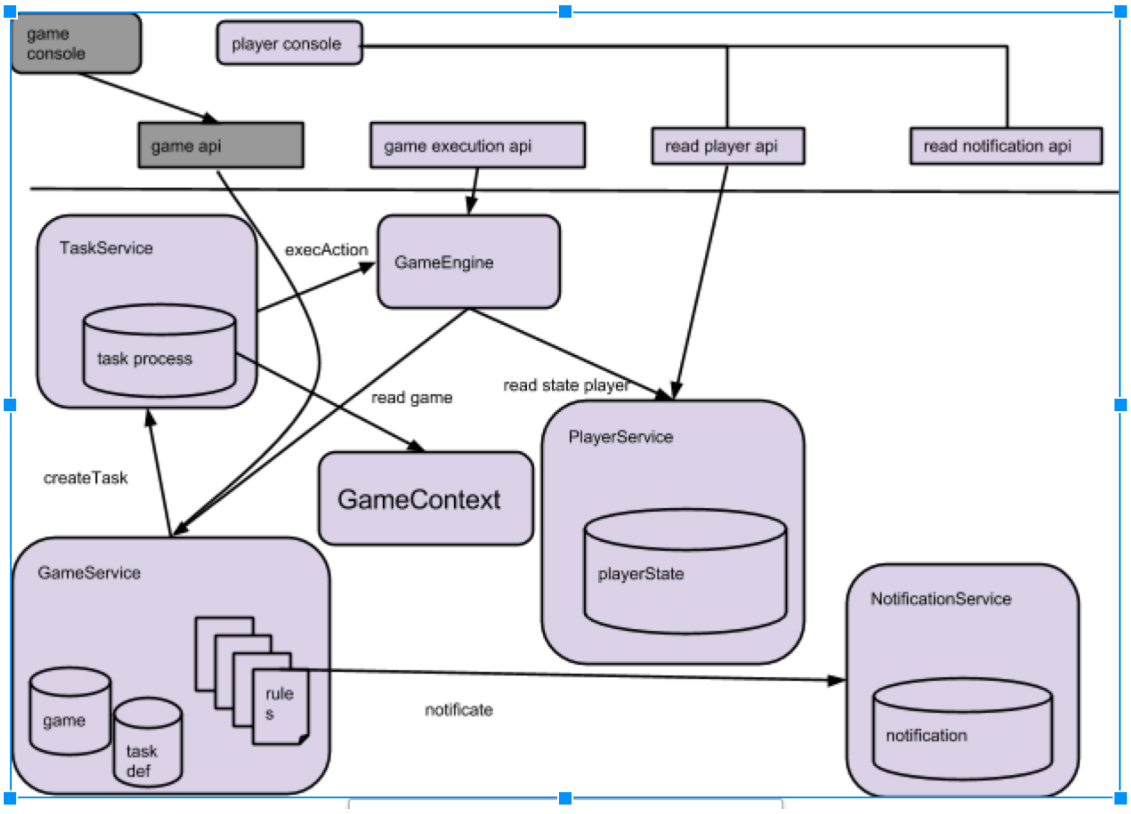


Figure 13: Design of GE release

Notice that at the current stage, the design-time part of the GE does not include high-level support tools for writing the game specifications based on the game model. For example, the coding of game logic as rules and tasks is carried out with regular software development editors - directly in the programming (rule-based) language adopted by the GE runtime – as opposed to dedicated development environments or domain-specific languages. That kind of higher-level development tools is something that we plan to provide in a future release of the GE, but it is not essential to the GE role in STREETLIFE, since it does not change the GE impact on sustainable mobility games.

Runtime elements in the current GE release include:

- The **GameEngine**, which allows to load game specifications from the GameService, instantiate one or multiple games, and execute the game logic described in the game rules; the execution model of the GameEngine is *reactive*, that is, the GameEngine fires rules included in the rule set of a game, based on "triggers" defined in the left-hand side of the rules; triggers predicate on incoming events that originate from a variety of sources: external applications, systems or sensors operating in the Smart City environment, player actions communicated via Apps, specified game tasks, or specific modifications in the game state itself; the execution of triggered rules has side effects, which can either modify the game state, or issue notifications directed to players or third-party systems. In the current release of the GE, the GameEngine component wraps state-of-the-art rule engine technology, i.e., the Open Source Drools rule engine[6]; as part of the game, the GameEngine also instantiates the gamification concepts included in the game specifications, through the corresponding service-based software modules (a.k.a. *gamification plugins*);

- The **game execution API**, which provides a service-based interface for the GameEngine; it is used by external systems to register "game actions", which are events that can constitute rule triggers for the GameEngine, and to communicate those events to the GameEngine during the lifespan of the game;
- The GameContext, which is an abstraction and a mechanism used to isolate different instances of games from one another, and make sure they execute and manipulate game state independently;
- The **TaskService**, which reads the specifications of tasks included in the deployed games, and schedules and executes those tasks as needed;
- The **PlayerService**, which is a service holding the profiles and state of all players taking part in each of the deployed games, and provides a persistence service for the GameEngine, ensuring persistence of all necessary player data and game state;
- The **NotificationService**, which is a service holding all notifications produced as part of the execution of the game logic, and which needs to be communicated to external systems and/or individual players; the technology and means for delivering these notifications remain application-specific, so it is not part *per se* of the GE;
- The **player console**, which represents the player-facing frontend of any gamified application running in the GE; the player console provides to each player the presentation of her own state in the game, and game results in general; it also provides a way to present the player with game notifications of relevance at any given time; in the current release of the GE, the player console supports Web-based and mobile Web modes of presentation;
- The **read player API**, which provides a service-based interface for the PlayerService, with operations that are used by the player console for retrieving game state and player information to be presented;
- The **read notification API**, which provides a service-based interface for the NotificationService, with operations that are used by the player console for retrieving pending notifications to the player.

[6] Available at: http://www.drools.org/

4.1.2. GE deployment options

The GE has gone through two release cycles so far.

The current GE release, besides implementing the design and features described in Section 4.1.1, it has been developed and packaged in such a way that it is available for deployment according to two different modalities, *stand-alone* and *hosted* (a.k.a. SaaS).

One principal reason for providing these two different deployment options is to offer flexibility of integration within specific instances of the STREETLIFE platform, and to respond to a diversity of integration requirements. We do not go into any further detail here about either deployment and integration modality, since they are discussed in Deliverable D2.2.2 [10].

4.1.3. Current feature set

The design of the current GE largely defines the functionality. The major blocks that we have described in Section 4.1.1 provides the main coarse-grained functional elements of the GE, that is: game definition and configuration, game logic execution, integration with Smart City systems, game and player state persistence, game and player state presentation, game notifications for players and third-party systems.

To understand the array of gamification applications that the GE can support, however, the most important thing is to look at the set of *gamification concepts* that it currently covers. Gamification concepts are an architectural abstraction native to our GE: each concept models a typical game element (e.g. *Points*), and implements the semantics typical of that game element. That implementation is reflected in a twofold way in the GE:

1. Each concept can have multiple instantiations in the same game (e.g. multiple types of points, such as Green Leaves, Health Points, Participation Points, etc.); each instantiation becomes at runtime part of the game and player state, upon which the game rules predicate, and which can be persisted;
2. Each concept is tightly associated to a corresponding service (or *plugin*), that is, a unit of runtime functionality, which is pluggable onto the GameEngine component. Each service has an interface, which dictates how game rules can interact and affect the corresponding gamification concept. The purpose of the interface is to ensure the proper semantics for that specific concept. For instance, the interface for generic points is rather simple, since points can be only added or detracted as the effect of a game action.

In the current release of the GE, the repertoire of gamification concepts that have been defined and can be employed for game definition is still limited. They are:

- **Points**: points can be accumulated by the individual player, as effect of the execution of gamifiable actions occurring via App and services in the real world, and which are mapped to specific game rules (e.g., *"Earn 1 Health Point per Km. travelled by bike"*, or as the effect of rules and tasks within the game logic (e.g., bonus or special offer rules, etc.).
- **Badges**: a badge is a marker that corresponds to a particular player achievement, either within the game or signalled by a gamifiable action (e.g., *"Earn the Green Commander*

badge when reaching 1000 Green Leaves points", or *"Earn the Bike Explorer badge the first time the player uses the bike sharing service in the city"*).

- **Badge Collections**: a badge collection is a set of logically connected badges; the main purpose is to create a stimulus for the player to complete the badge collection once it has earned one or more of the badges in it, and hence provide the player with an additional incentive to reach the corresponding achievements.
- **Leader Boards**: a leader board is a ranking and filtering of players' status in the game based on a given instantiated point type. In the GE, the leader board gamification concept does not need its own plugin, since it is an ancillary concept with respect to points. Leader boards are configured instead as tasks that the GE executes when needed to collect or refresh the relevant points state information, and organise it for presentation.

The gamification concepts provide the game features proper, around which a game can be defined, developed and executed.

The repertoire of gamification concepts available in the current GE is limited, but can be modularly extended. First of all, each concept can be specialised (e.g. a game designer can define *Reedemable Points,* or *Karma points,* as specialisations of the generic Points concept [11]). Any changes to the semantics of such specialised game concepts must be reflected in the corresponding plugin its own service interface, and of course the implementation of the operations listed in the interface. Moreover, new concepts can be added by specifying the data structure holding the related game state, and the corresponding service interface and plugin.

4.1.4. Planned features

The GE technology will be further extended and augmented during the rest of the STREETLIFE project, with the objective to increase reach of the software, and turn it into a mature solution for providing gamification solutions within Smart City environments [12].

In part, the direction of future developments will be established on the basis of the experience gained through the validation taking place in the STREETLIFE pilots during the second iteration; we expect that – as it already happened in the first iteration - the pilot experiments that include gamification will teach us a number of new lessons on how to design increasingly effective games for sustainable urban mobility. Another important source for further innovation is the study the STREETLIFE project has recently carried out on engagement and participation (see Annex); lessons learned and recommendations from the study will be incrementally incorporated in the development of – and experimentation with - the GE throughout Y3 of the project, which will culminate with the a final release of the GE.

Finally, there are also a number of features that have already been identified for inclusion in the next GE release, and are planned or in progress. Those features are related to two aspects of the GE:

1. Extension of the GE system infrastructure:
   - Development of a fully contained "sandbox", to be used in the context of hosted (SaaS) deployments of the GE; the main purpose of the sandbox facility is to let game developers experiment with game logic alternatives and evolutions in an isolated development and testing environment, which does not impact or perturb any game versions that are already deployed in the SaaS modality. This full-fledged sandbox

evolves from a prototype of a ruletesting environment, which is already present as a simplistic development support tool in the current GE release.
   - Development of a set of "widgets" for the GE frontend; the main purpose of a widget system is to provide a compositional approach to designing and developing the player-facing frontend of game, making frontend programming fully modular and customisable. A widget system and a set of widgets can extend the GE ability to present game and player state (plus any other relevant game information) towards a diversity of modes, media, and devices.
2. Extension of the GE expressive capabilities for the specification of more sophisticated games:
   - Development of support for collaborative and team games; currently the GE is able to represent and execute only individual games, where players have no real interaction with one another; we are planning to support personal social interactions between players, as well the ability to define in-game interactions, such as collaborative game actions, team-forming and team-based collaboration and competition.
   - Inclusion of additional gamification concepts for the GE, and development of the corresponding services and service interfaces for the GE runtime; some of them are directly related to the support of collaboration within the GE: for that, we are planning a **Team** concept. We are also planning a **Mission** general concept, to capture quest or challenges to be assigned to players within a game, with specific specialisations of the Mission concept that support collaborative games, i.e., **Collaborative Missions**, and **Team Missions**.
   - Extension of game logic; in particular, we will introduce support for temporal reasoning in the game rule base; temporal rules are going to be particularly useful – among other things – for game logic that predicates on the concept of Missions.

Some of the features above are going to be deployed in experimental form already within the games planned for the second pilots iteration; other are not necessary for the pilots and will be completed later.

4.2. A framework for Smart Cities gamification

As part of our research on citizen participation and gamification prompted by the STREETLIFE, we have now devised an innovation agenda, which uses the GE as a technological asset upon which to build a wider more comprehensive approach for enabling the gamification of complex and open socio-technical systems and applications, like those that characterise Smart Cities, and in particular its Smart Mobility sub-domain.

We have recognised that gamification for this kind of systems looks quite different from traditional approaches, which typically deal with adding game-like features to one single, well-known, closed-boundary application, for instance an information system, and for a single objective (or a set thereof) that is decided a priori. Therefore, one major objective of our current line of research is to come up with a gamification approach that supports and promotes *behavioural changes among Smart City citizens* that are in line with city-defined sustainability policies and KPIs. Those policies are dynamic, i.e., they may change often, or be manifested in varying ways; this is one significant difference from traditional gamification. Another important difference is that many subsystems of a Smart City are equally dynamic, and also open-boundary. The Smart Mobility subsystem is a perfect example: it is made of a collection of loosely integrated or fully autonomous ICT systems, information sources, (mobile) applications, sensors etc., which are owned and controlled by multiple parties and which *collectively* provide the mobility services sought by citizens. A Smart Mobility game must thus promote, supervise and incentivise the use of some of some features across those multiple systems, in a uniform way and must be in line with the aforementioned sustainability policies.

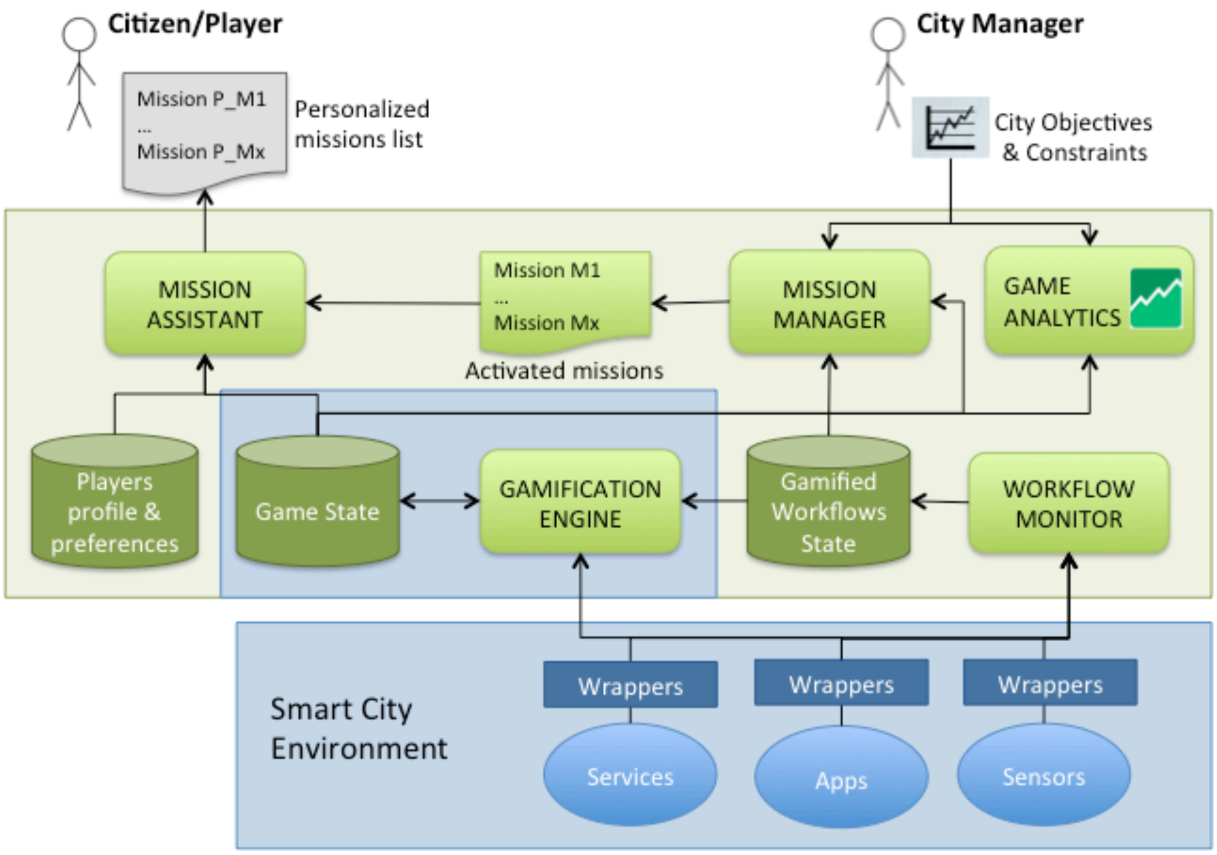


Figure 14: Gamification framework design

To address this multi-fold complexity we are designing an innovative gamification framework. An informal depiction of its design is shown in Figure *14* in which the elements already covered by the current GE are framed in light blue. The design of our gamification framework as a whole is based on the following concepts:

- As discussed already in D5.2.1, and evident from our GE design discussed in Section 4.1, we adopt a *service-oriented approach* as the foundation of the framework. This enables simple and standard interfacing and interactions between the functionality of Smart City systems and the runtime elements of our gamification solution, and for loosely coupled integration between those elements, since the service-oriented paradigm is often used as a common denominator between heterogeneous ICT technologies, which may range from mobile Apps, to IoT, to Cloud-hosted applications, to legacy information systems of enterprises and PAs. Service-orientation is a unifying principle for both to services existing in the Smart City and being used by citizens/players, and the runtime components of the gamification framework.

- All gamification solutions are built around a repertoire of Smart City services. Those services are decorated with suitable annotations, which, besides specifying the syntax and basic mechanics of their invocation and operation (as customary in Service Oriented Computing), describe their provisions related to the behaviours that the gamification approach wants to promote; for example, some Smart City services could be annotated with tags for the type of service, such as the MOBILITY tag, and among them, some could be tagged as PUBLIC_TRANSPORT, other as VEICHLE_SHARING, other as VEHICLE_RENTING, etc. Smart City services interact uniformly with the gamification framework through a layer of *wrappers*, which expose as events interesting interactions between citizens and services, and add the suitable annotations.
- Operations with Smart City services must be *monitorable*, which allows the gamification framework to be automatically informed about when and how its players exercise the corresponding service features [13, 14]. Atomic monitors are provided by wrappers as is. However, monitors may be also composed according to some given *game logic*, in order to garner game-relevant information that is signified by complex combinations of service invocations and workflows of actions, as opposed to the simple invocation of a single service operation.
- Monitored executions of Smart Cities interactions by citizens/players are represented as *"gamified workflows"* of varying complexity within the Smart City. The purpose of gamified workflows and their state is to internalise in the gamification framework the information about how citizens/players have interacted / are interacting with the Smart City in any way that may be relevant to the application domain being gamified.
- Changes and progress in gamified workflows, as well as direct interactions of citizens with Smart City Services also trigger interaction with *game services*, whose main purpose is the modification of the *game state* i.e. tracking the progress of the players with respect to given game objectives (e.g. increase of points, achievement of badges or awards, fulfilment of missions, etc.)
- One useful way to organise the aforementioned game logic is around the concept of *missions* in the Smart City. A mission is a logical unit of playable game content, which is assigned to a player (or sometimes to a group of players) and activated. A mission example for a sustainable mobility game is: "try the new Bike Sharing service of the city", or "commute using Park&Ride tomorrow". Once a mission is active (i.e. accepted by a player), it has its own state, which is part of the overall game state, and a satisfaction condition, which predicates about some portion of the state of some gamified workflow, and which needs to be itself monitorable. We envision a *Mission Manager* component in our gamification framework, in charge of tracking the progress of instantiated missions, and check whether the satisfaction conditions of active missions are reached. When that happens, the mission is fulfilled, and the player earns benefits, which are reflected as progress in her game state. The concept of simple missions can be extended, for instance, by *quests*, i.e., a series of logically or causally concatenated simple missions, or *challenges*, i.e., long-term objectives that are reached by the incremental accumulation of state in either the domain and game state (for instance, "ride Bike Sharing vehicles for a total of at least 200 Km during work days this month").
- Complementary to the Mission Monitor component, we envision for our gamification framework also a *Mission Assistant* component. The Mission Assistant is a personal software agent, which is responsible to issue recommendations to the player on what

actions she may take (i.e., what Smart City services to interact with) in order to progress on outstanding missions and complete them. The Mission Assistant builds upon the meta-data decorating Smart City services, as well as knowledge about the missions' state, and the game state of its player. A future version of the Mission Assistant could also issue recommendations about what missions the player should take up next, in order to improve its game standing the most, which requires the Mission Assistant to also be able to reason also about element of the game logic – pre-eminently the logic regulating incentives and the acquisition of player standing and benefits.

- Missions are created in support to a set of *behavioural objectives* of the Smart City, either strategic and long-term, or tactical. For example, in the Smart Mobility sub-domain, a strategic objective can be to increase the average usage ratio of existing means of public transportation in the city by a given percentage; a tactical objective is to promote commuting via bike sharing and car sharing in a given day, because of an impending public transport strike in the city. Behavioural objectives are decided by city managers and other administrators (for example, by the city Mobility Manager on the basis of the reporting and simulation functionality offered by the STREETLIFE mobility control panel). Those objectives must then be turned into a set of relevant and viable missions; this can be done manually by an expert - a gamification designer - who also decides the corresponding in-game benefits and incentives for the various missions. In the future, we envision that mission could generated (semi-) automatically, by means of a the Mission Manager component, which can incorporate a *Mission Generator* functionality in the gamification framework, which can use some form of *Procedural Content Generation*, a set of AI-based techniques adopted at design and run time in contemporary computer games [15, 16]. Missions – as they are assigned to individual players – could also be *personalised*, i.e., customised according to the player's profile, preferences, and her current game state.
- We also envision a *game analytics* component, which is in charge of enabling game designers as well as Smart City administrators to gain insight on the impact of certain game mechanisms (for instance, the launch of a new mission) have with respect to the achievement of the aforementioned behavioural objectives. This component would allow running statistical analyses that correlate intended behavioural effects to the delivery in-game incentives and the corresponding progress of players within the game. That would allow, on the one hand, to collect evidence about the efficacy of the gamification mechanisms put in play, and, on the other hand, to adjust those mechanisms in order to increase that efficacy.

A gamification framework encompassing the elements described above is designed to augment the ability of designing citizen engagement games in open-boundary, heterogeneous and multi-party service environments like Smart Cites, and to align them with policy objectives of the city. We envision several major advantages that such a framework will bring about, including a high degree of automation for the support the definition and generation of the logic around which games for Smart Cities revolve; a high degree of automation for the support the specification and evolution of the integration facilities between the game and the Smart Cities ICT infrastructure, which is necessary for the execution of a given game; a high degree of support for the customisation, contextualisation and personalisation of each game to the specific Smart City environment it is built for; and a high degree of support for management and evolution of gamification applications that keeps the participation incentives and rewards closely in sync with the relevant policy decisions and priorities of the Smart City where they are deployed.

4.3. Incentives and Citizen Participation

Annex I contains information on how users of STREETLIFE application can be incentivised not only by gamification, but by other means including information, rewards, dynamic planning and social networking. The Annex introduces the first prototype of incentivisation applied to STREETLIFE application.

5. ADVANCED GRAPHICAL INTERFACES (T5.4)

Our advanced graphical interfaces support both Augmented Reality (AR) and Augmented Virtuality (AV), or 3D Maps, together providing both ends of the *Mixed Reality* virtuality continuum. D5.2.1 describes the basic features of AR and 3D maps. They are both aimed at exactness and intuitivity in contrast to more traditional 2D interfaces. . The main goal of this research is to assess if these interfaces would support users in travel related tasks better than traditional means.

In STREETLIFE, we identify use cases where they would provide additional benefits for people on the move. The initial set of use cases (See also D5.1 use cases AGI-3DMAP and AGI-AR for TRE) include the following:

1. User is not familiar with the bus routes proposed by the IJP. (S)he uses the MR application to identify the exact location of the bus stop, where (s)he has to go.
2. User is getting to the bus, but (s)he is still tens metres away and sees buses coming. However (s)he doesn't see the bus numbers because of the angle of the bus or buildings preventing visibility. With the MR application (s)he can visualise the correct bus on the application screen with estimation if (s)he has time to get to the bus.

These use cases are tested both in Tampere and Berlin in field tests, where a limited test group (3-10 people) is selected, they are given a mobility-related task, which they perform and the experience is transferred not only by interviews, but also by device sensors.

One of the key issues in assessing the usefulness of 3D maps and AR is the robustness of implementation. If our current implementation lacks sufficient quality, we are not in position to provide definite answers regarding the interfaces itself; for example, if a 3D model is of insufficient quality or scale, we cannot properly validate 3D maps as an interface technology. Similarly, if our AR registration is insufficient, we cannot validate AR as an interface technology. Our efforts for technical improvements are described in Sections 5.1 and 5.2.

Development of a useful mixed reality interface is far more demanding than using traditional means. Both methods (3D maps and AR) are technically challenging, and depend on available hardware, programming interfaces, data and algorithms, which must yield a robust system for just achieving the fundamental requirements. Therefore, a wide range of commercial applications does not exist in the mobility context.

However, advances made in sensor technology, data transfer and computation will eventually bring the advanced graphical interfaces to commercial use. The objective is to apply the advanced graphical interfaces in STREETLIFE to gain understanding how they can help users in the mobility context and how more sustainable transport modes can be promoted with the help of advanced graphical interfaces. Once tested in Tampere and Berlin the same methods and principles can be used in any other municipalities if the prerequisites are fulfilled. The prerequisites include accurate 3D model of good quality of the municipality, 3D models of transport modes and real-time mobility data sources that can be visualised by the 3D models of the transport modes on the 3D city model. For AR, a critical prerequisite is accurate registration with the real world.

The MR requires utilisation of the mobile device sensor data and computations to show the correct view on the device. The engine architecture relies on standard data interfaces to integrate STREETLIFE data sources. The MR environment will also be integrated with the IJP in order to show the routes on the MR user interface. The MR is also integrated with the crowdsourcing component to get user feedback of bus routes in TRE, and with the ActivityRecognition component for real time crowd-sensing data feeds.

5.1. Improvements on 3D Maps: Work towards scalable 3D worlds

As part of Horizontal Integration (See D2.2.2 for STREETLIFE Integration Concept), we needed to provide a version of the TRE Mixed Reality App for BER site. Given a compatible process pipeline, such work would normally require only integration of data sets. Of two available models, the one with compatible data model and larger coverage was selected. However, it is immense in size. The model covers the entire greater Berlin with over 50.000 buildings (in CityGML format) and 200.000 textures (in JPEG format), with a total of 28GB compressed data. In the following, we present our current pre-process scheme and the updates that are necessary.

5.1.1. Static geometry: scalable processing of large models

Our 3D pre-process consists of the following pipeline:

1) File parsing, data validation and storage in internal representation
2) Texture LOD generation and dominant colour selection
3) Visibility pre-determination for individual 3D grid cells
4) Visibility data compression
5) Packaging to binary cache files for transmission and storage

Figure 15 presents the idea of our approximate visibility pre-determination: if the viewpoint is within the blue cell, only a part of the city is visible and needs to be resident in memory for rendering. The entire world is divided to such cells, and each cell is associated with a list of potentially visible facades or other objects. As these lists tend to become very large in complex environments, they need to be compressed. In a global compression scheme, we typically achieve over 95% compression rates, for example using Huffman encoding (we have implemented multiple compression schemes for comparison purposes). However, our implementation of the off-line visibility process has – until now – relied on sufficient memory to hold the entire 3D model and the visibility data resident in memory (in the on-line situation, a 3D client only needs to hold the visible parts!).

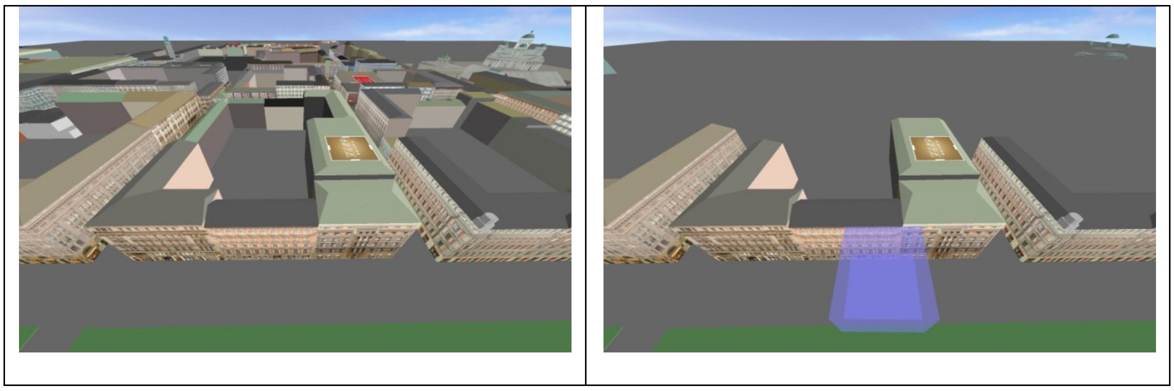


Figure 15: The use of pre-computed potentially visible sets (PVS) to select a model subset

A piecewise solution

For the case of 3D Berlin, we realised that our current pipeline is insufficient to handle such vast models. We are currently in the process of creating a scalable pre-process phase, which loads in only parts of the city model and does also visibility compression in multiple phases. Additional care has been paid in maintaining unique identifiers for each texture and piece of geometry in the piecewise process (the process includes an assignment of unique identifiers to each mesh and texture; in a piecewise process, these need to remain consistent over the entire model). Currently we have run the first visibility determination tests with the city model, but have not yet tested a piecewise compression scheme. As the resulting visibility data sets are very large, we are worried about raw I/O performance and our computational resources.

5.1.2. Dynamic Entities: Visibility Based Interest Management

Along with updating the static visibility pipeline, we similarly need to suit our dynamic entity management to large-scale environments. While we manage any moving entity in our world via the World Graph, we also use the same underlying topology for visibility based interest management. Figure 30 illustrates the concept. The World Graph is based on street topology, and we map every tracked entity to the graph. Now, if a viewing client is *interested* in only visible entities, we make a simple approximation: if a street segment is potentially visible from given view position, so are all entities that are currently *occupying* it.

During pre-process, we create *virtual hulls* over every street segment, with sizes that matches the largest possible entity, such as a truck or tram, and width of the road or pavement. For areas with large visibility, hulls can consist of multiple segments (Figure 16, right). We then pre-determine visibilities separately from building geometry to these virtual hulls. As part of dead reckoning optimisations, we assert a *validity period* for each hull, during which we expect our entity to remain in the given hull. A large entity such as a tram may occupy more than one segment at a time.

In Figure 16 (right), tram A has just reached a new segment, and sends a state update, with its new position, speed and occupied street segments. Tram B is still in the crossing, where a cross-shaped hull covers four segments, occupying that hull. Its validity period has not yet expired, and a dead reckoning algorithm keeps updating its position.

Due to the large-scale of Berlin, we are now re-writing our World Graph management code and visibility pre-process of these virtual hulls, simultaneously with the new pipeline for managing static geometry. This case is especially demanding as a server may receive state updates from any part of the city. A single server cannot manage the entire planet Earth, and we face another scalability challenge. However, at the moment we will attempt to manage Berlin in one machine by writing tight code with optimized memory usage, setting up temporary caches for the Interest Management look-up tables.

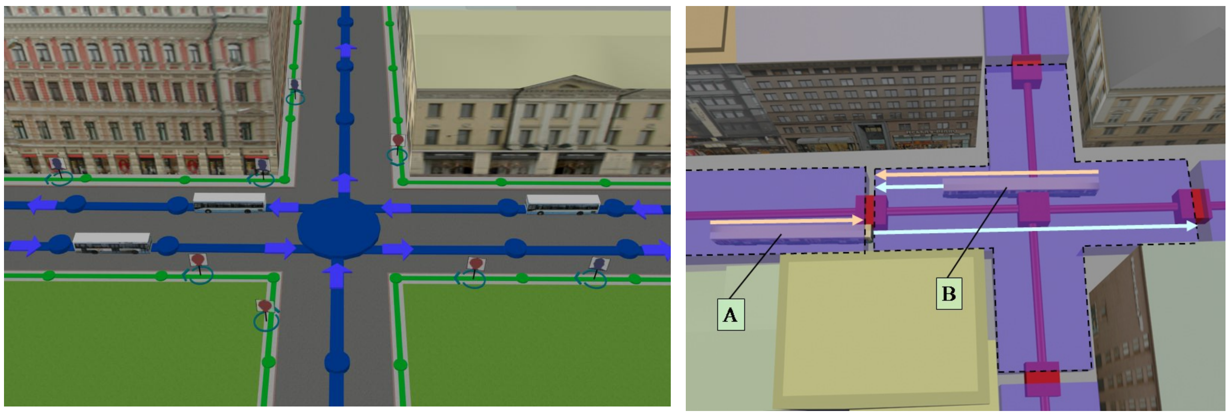


Figure 16: Creating virtual hulls over street topology for visibility management

5.2. Improvements on Augmented Reality

5.2.1. Challenges for Augmented Reality

For AR, the main challenge is *registration*, which has received considerable research effort for some time already and is almost a science of its own [17]. The fundamental challenge in registration is matching the real world and the virtual world, where all our digital data resides. Misalignments in AR are very serious, as the conflicts are of the type visual-visual for human senses, and therefore directly observable. Figure 17 presents a simple example of a misaligned augmentation. The arrow is supposed to point at a bus stop (Fig. 17, upper) but due to sensory inaccuracies, it is slightly off target (Fig. 17, lower).

For immersive virtual environments such as Oculus Rift or even non-immersive 3D maps (such as those portrayed on a tablet) these errors are not so critical: for Oculus Rift, conflicts are visual-kinaesthetic and visual-proprioceptive, which would require specific attention and testing to be revealed[7]. For a 3D map run in a tablet, while the type of error is visual-visual (tablet's display and the actual world), the mismatches are not in the same frame of reference: our augmenting content is registered in the same coordinate system as the 3D model, and in a tablet's display, there is no conflict. However, the pose of the entire 3D map may not match with the real world in an egocentric view (the view may not be the same as if the tablet would

[7] Visual-kinesthetic and visual-proprioceptive mismatches are the cause of cybersickness; although this is easy to feel, the exact amount of mismatch isn't easy to estimate.

be transparent); on the other hand, a 3D model is not limited to an egocentric view, but our user may interactively choose whichever viewpoint she prefers.

During our Y1 testing for bus stops and buses for augmented targets on AR view, it was evident that our accuracy was insufficient.

A second rather critical challenge for AR is related to *occlusions*. If we simply project content on top of the camera view, what if the content is actually behind a building? During Y1, we implemented a method to render occluded content (such as 3D bus models) in the 3D view using red colour. On AR, we mainly used arrows and bus numbers as textual labels. However, almost any view with bus labels showed multiple occluded labels without any occlusion cue, causing visual confusion.

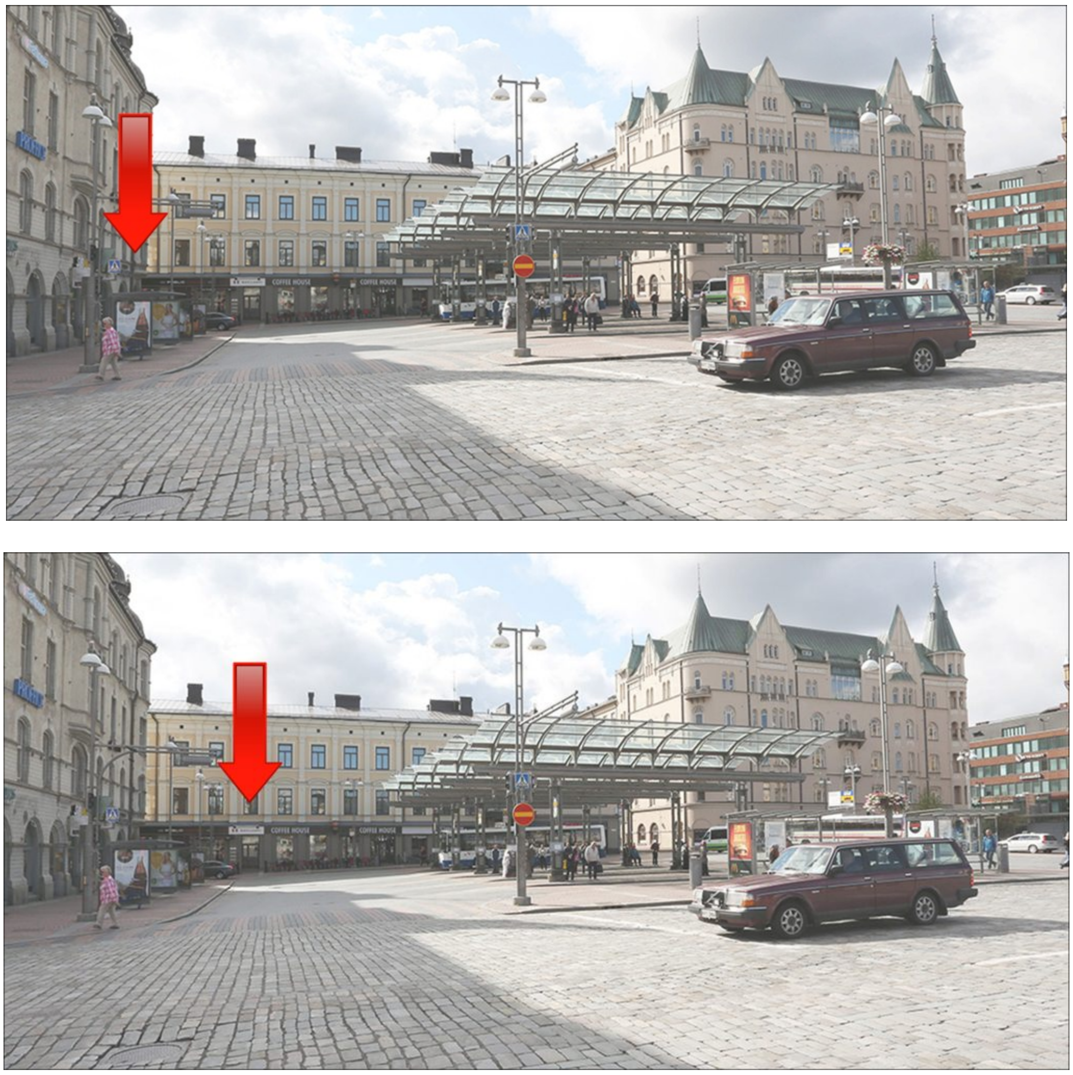


Figure 17: Pointing out a bus stop in augmented reality (top). Even a small error in placement easily confuses users (bottom)

As described in D5.2.1, there are two families of registration for augmented reality, namely sensor-based and computer-vision based. The former relies on external sensors to provide the pose (position and orientation) of our input, namely our tablet's camera. The latter attempts to deduce the pose directly from the camera view by analysing pixel data, or at least would provide tracking based on it.

5.2.2. Improvements on Augmented Reality

In order to improve our implementation of AR, we analysed our options and capabilities. A "full" registration system would provide perfect 6 degrees-of-freedom tracking and occlusion management. We limited our options to three cases:

1) Computer vision based localisation (position and orientation)
   a) Replace both GPS position and IMU based orientation tracking
2) Computer vision based tracking (view direction following)
   a) Replace IMU based orientation tracking
   b) External method for positioning
3) 3D Model based occlusion determination
   a) This would provide either a) graphical or b) analytical occlusion management

1a) Replacing GPS position and IMU based orientation tracking with computer vision based localisation. Computer vision based localisation methods depend on 1) a 3D world data set and 2) matching of an input image to the data set. For recognition purposes, a common approach is to create a sparse 3D point cloud, where each point represents a certain computationally defined *feature*, a description that allows re-identifying the surrounding area in previously unseen images. Such 3D point clouds are typically created from a set of highly overlapping images in a *Structure-from-Motion* (SfM) pipeline of operations.

State-of-the-art Structure-from-Motion (SfM) pipelines are developed to work with highly redundant image datasets in a batch-based manner. Hence, the first important step of an SfM pipeline is the image acquisition, where the quality and completeness of a reconstruction largely depends on the provided image sets. The properties of an image dataset to obtain an accurate, fully connected and complete reconstruction are versatile. The main influencing parameters are the degree of redundancy, the spatial distribution of viewpoints over the area of interest, the relative orientation between viewpoints and the texture of the scene. A non-expert user in 3D reconstruction is often not able to acquire a suitable image set because he has no knowledge about the important parameters.

A typical SfM pipeline consists of the following steps:

1) Extracting interest points from images (i.e. SIFT [18] or SURF [19])
2) Calculating descriptors for interest points
3) Matching descriptors
4) Geometrically verifying matches using a robust hypothesise-and-verify framework
5) Triangulating 3D world points
6) Bundle Adjustment

Once the sparse 3D point cloud model has been generated, it can be used for *Image-Based Localisation*. Image-based localisation deals with the problem of how to precisely recover the 3D camera pose from a query image (the current viewpoint on the 3D world) with respect to a

3D point cloud obtained by any Structure-from-Motion (SfM) pipeline. In general this problem is solved by four subsequent steps:

1) Detect interest-points in the query image and describe local appearance in terms of descriptors (SIFT, SURF…).
2) Find correspondences between interest points detected in the query image and 3D points (so-called matching step).
3) Verify correspondences in a geometric manner using a perspective-n-point (PnP) absolute pose algorithm in a hypothesise-and-verify framework.
4) Use verified correspondences to calculate the 6-degree-of-freedom camera pose in terms of a rotation matrix and a translation vector.

Although image-based localisation would essentially provide a very robust AR solution, developing a full SfM pipeline and related localisation scheme from scratch is out of scope of STREETLIFE, and suspect to multiple other issues such as efficiency of implementation and computational resources of available devices. Furthermore, capturing sufficient image datasets of our case areas would require substantial effort and specialised equipment. We therefore had to look at other possibilities.

2a) Replacing IMU based tracking with computer vision. Computer vision tracking methods can be subdivided to two categories: 1) fiducial marker based tracking and 2) markerless tracking. Fiducial markers are specially constructed graphical icons, which possess high contrast and are uniquely identifiable. In the case of AR, these icons are typically non-symmetrical, which allows resolving and visual tracking of their orientation. In our real world case, we only consider markerless tracking.

Tracking differs from a true 3D registration by the less demanding requirement of real-time matching the orientation of an initial frame of reference with an input image flow. Tracking does not provide localisation (the position component), and can be independent of any pre-processed feature data sets. In this case, a tracking algorithm can be based on the concept of *Simultaneous Localisation And Mapping* (SLAM): for each video frame, the camera is first registered based on features in the map; in a second step, the map is then extended with new features from viewing directions that have not been observed before. [20]

Due to our collaboration with Technical University of Graz on the field of mixed reality, we were able to acquire their implementation of a panoramic SLAM tracker for augmented reality. It was provided as a library with a simple API for feeding the tracker with video frames, and receiving status responses and a view matrix.

Integration of TUGraz’s tracker required us to re-program our video stream management on Android tablets. Our sensor-based AR solution did not require any analysis stage for the video stream, as we were able to simply direct the stream to the display background. TUGraz’s method required us to

1) Extract video frames to memory
2) Scale video frames down (1:2 … 1:4)
3) Provide individual frames to their OpenCV based implementation
4) Simultaneously create textures of each video frame at full resolution (non-scaled)

5) Render each video frame texture onto a quad on the frame buffer
6) Replace our view matrix with the one received from the tracker and re-order the coordinate system to match our 3D model

We can currently configure our mixed reality platform to use either our previous sensor-only version of AR, or the integrated TUGraz's tracking version. We can also choose a suitable video mode and define downscaling factors of the video stream for tracker and rendering separately.

TUGraz's method is computationally very demanding, with the additional burden of creating textures and rendering textured quads for each video frame. On ASUS Tf700T and ASUS Tf701T tablets, this was too much, even for a 1:4 downscaled video streams. On an NVidia SHIELD, however, the system runs smoothly.

The current tracking method is based on computer vision only, and is suspect to movement in front of the camera, fast or sudden motions of the device itself, and environmental conditions. Some surfaces are very repetitive and suffer from visual artefacts that do not facilitate tracking, i.e. a façades with identical windows. In practice, natural interaction is not yet completely achieved, but we expect a better implementation to be available soon with fusion of gyroscopic sensing for very short term cases where tracking is lost.

2b-1) External positioning methods: GPS based systems. In order to improve *positioning*, we could not find a feasible outdoor alternative to GPS based methods. Real-Time Kinematic (RTK) techniques appeared among the most promising outdoor positioning methods. RTK is based on GPS signals, but instead of simply interpolating received timestamp differences against known satellite positions, RTK attempts to lock in the phase of the signal's carrier wave.

RTK solutions for interactive use are not common, and we expected to perhaps be able to use it for one or two test systems at most.

The difficulty in making an RTK system is properly aligning the signals. The navigation signals (timestamps) are deliberately encoded in order to allow them to be aligned easily, whereas every cycle of the carrier is similar to every other. This makes it extremely difficult to know if you have properly aligned the signals or if they are "off by one" and are thus introducing an error of 20 cm, or a larger multiple of 20 cm. This **integer ambiguity** problem can be addressed to some degree with sophisticated statistical methods that compare the measurements from the C/A signals and by comparing the resulting ranges between multiple satellites. However, none of these methods can reduce this error to zero. RTK techniques depend on available reference stations, which can be physical or virtual. A virtual reference station (VRS) can be created by triangulating the measurements of three or more physical stations to a given location.

We engaged a decently affordable *Piksi* RTK system in an urban environment, Espoo, Finland. After some issues with the development environment and waiting for a decent firmware, we were able to test it, but failed to achieve a phase lock, which is essential for the technology. Other developers have observed similar difficulties:

"*By getting fixed RTK and a high precision path, we have made proof of concept that the Piksi does offer RTK positioning for a low cost, but does not have reliable results in varying conditions. Until the Piksi modules and firmware become more accurate and reliable, we do not recommend using this system*" [21]

Bluetooth GPS Test. As we could not resolve our positioning problems with Piksi, we made an outdoor test with a set of external Bluetooth GPS devices and two internal GPS devices:

1) Samsung S6 Android phone internal GPS
2) NVidia SHIELD Android tablet internal GPS
3) Dual XGPS 150A Bluetooth GPS
4) GNS 2000 Bluetooth GPS
5) Garmin GLO Bluetooth GPS
6) Canmore GT 750 FL Bluetooth GPS

The goal of the test was to determine the device that would yield the best possible positioning. The internal devices embedded with tablets and phones would be the default ones, but with previously observed inaccuracies, we were hoping to achieve better results with external Bluetooth devices.

The test was performed at the parking lot on the North side of Open Innovation House in Otaniemi, Espoo, Finland (See Fig. 18). The building is 3 stories tall and the white markings on the parking lot allowed rather accurate replication of the test path.

The experiment was performed using AALTO's modality detection software *ActivityRecognition Logger*. The Logger uses Google Play Services API for position, which automatically includes available other sources such as known WiFi access points. Bluetooth GPS devices were enabled by using the *Bluetooth GPS* app together with setting the Android tablet to *developer* mode and allowing "mock locations" and Android Developer settings.

For each device, the experimenter first waited for a good signal quality using Google Maps approximately 25m away from the building. Once a good signal quality was reached, he moved to the green marker on Figure 18 and started the logger. Holding test devices in his hands an arm's length from his body, he walked along the path marked with white on Figure 18, circling the parking lot three times, effectively creating parallel paths on distances 19m, 14m, 4m and 3x0.5m. The test path was walked separately for each device, but during the same afternoon, without breaks.

We expected largest errors nearest to the building. The circling route gave us a three-time repetition on the 0.5m distance, allowing us to verify the consistency of each device at that challenging range.

The results were rather varying. Figure 19 presents the recorded data. As expected, the accuracy near the building was the worst. In addition, there were anomalies and huge extrapolation errors. Samsung S6, GNS2000 and XGPS 150A all managed to extrapolate the tester's path over 100m multiple times. In these cases, the devices failed to notice that the experimenter had moved closer to the building from the North-East corner, and at the loss of signal strength near the building, these devices simply expected the experimenter to continue his walk forward. XGPS 150A exhibited most variation and error. Most devices suffered from single jumps. Of the devices, the GT 750 FL performed most adequately, with only 1 anomaly and otherwise having less than 4m error at all times. At the distance of 19m, the error was less than 1m, which would be sufficient.

Based on the test, we chose GT 750FL as the positioning device for further experimentation with Mixed Reality, where accuracy is one of the critical requirements. Although the results of GT 750FL were quite promising, we have later observed variation in accuracy depending on environmental conditions and location. Our plan is next to identify potential areas in TRE city centre, where most reliable results can be expected for subsequent experiments.

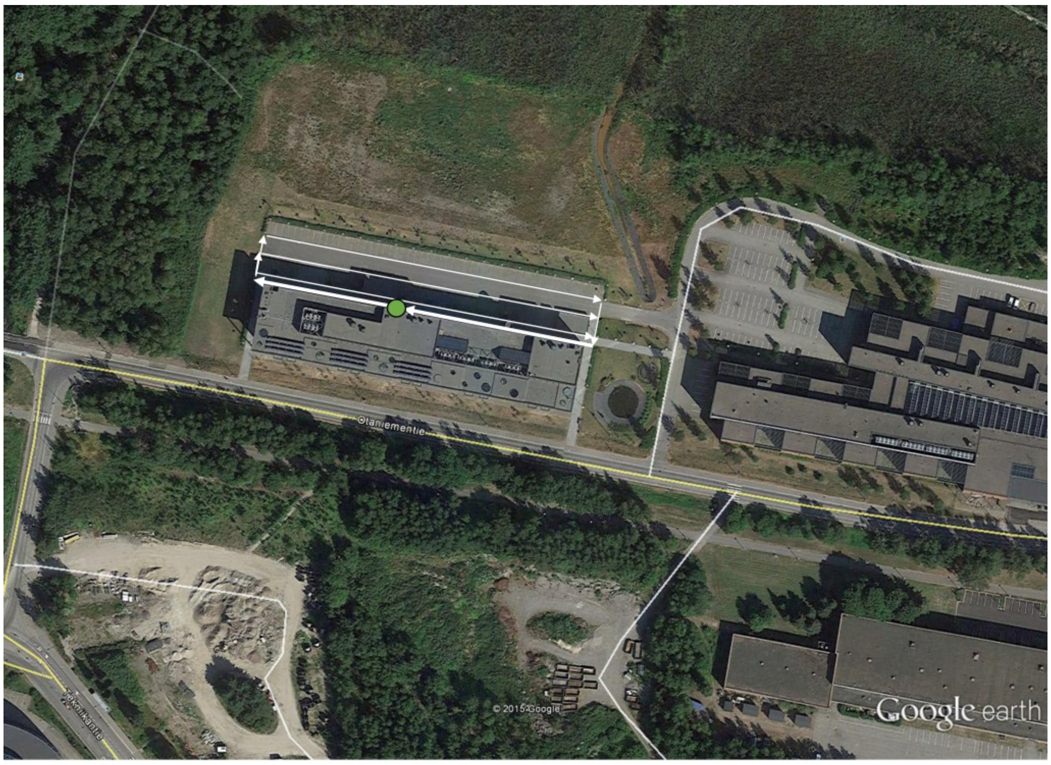
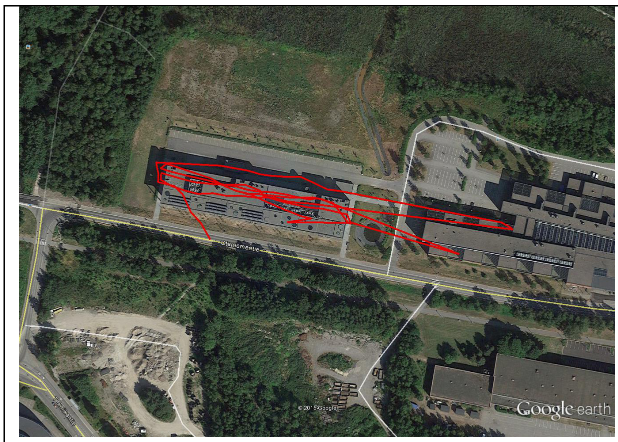
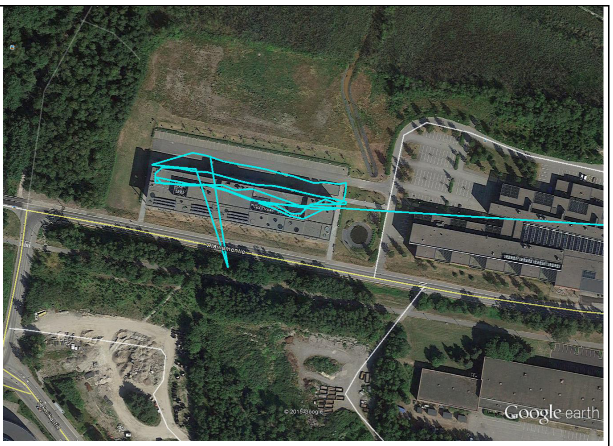


Figure 18: A reference path for a GPS test just North of a three-storey building

	
Samsung S6	NVidia SHIELD

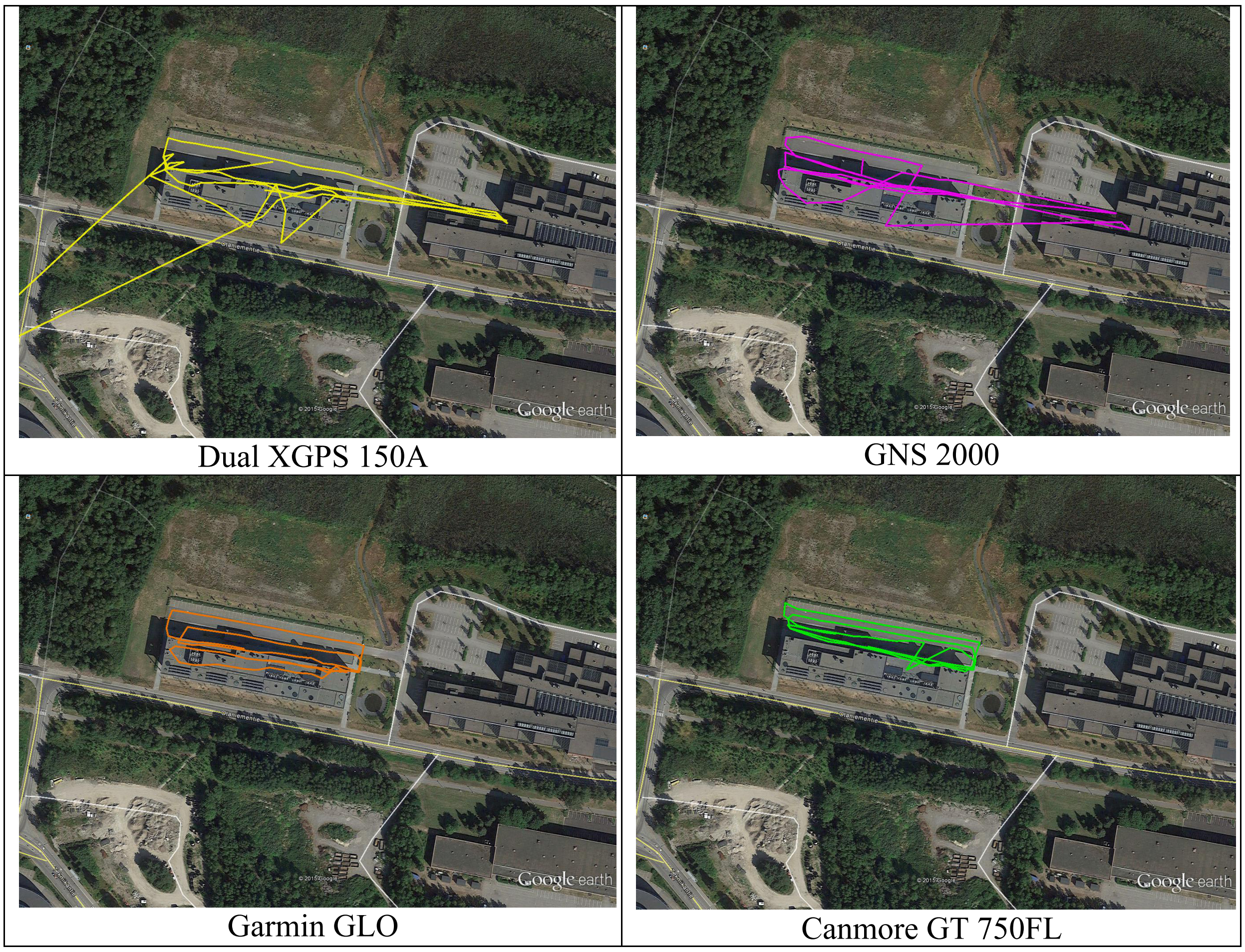


Figure 19: GPS logs for two internal and 4 external GPS devices

2b-2) External positioning methods: Wizard-of-Oz. For cases where external GPS devices are not reliable, we have the last resort in a Wizard-of-Oz method. Here, we use artificial means for positioning. We currently support two methods:

1) Pre-determined unique positions
2) A secondary map interface where an expert person positions our subjects

Our pre-determined positions can currently be set up as a configuration file for a field experiment, together with a bit of heuristics for determining task randomisation methods and other parameters. In addition to positions, these parameters can include mode (AR, 2D, 3D) and view position and orientation (2D, 3D). To accommodate for consistent local anomalies on magnetic fields, such as underground metal structures, our pre-configured parameters also include orientation offsets.

We have also created a simple pedestrian entity simulator with a map interface. We describe this *Map Tapper* in Section 6.

3) 3D Model based occlusion determination. With an accurate 3D model available, it is possible to use it directly for occlusion determination for augmented reality, given an accurate pose. We already had a graphical representation of occluded content for 3D models in a 3D city map. However, we hypothesised that an analytical result of occluded content would allow us to use more advanced ways of representing data in the AR view.

Our implementation of "analytical" occlusion detection is rather straightforward:

1) Use an off-screen frame buffer object to render our 3D city model with depth information only
2) Render representative graphical icons for any content or entity that needs occlusion to be resolved on their 3D location with Z test on; use colour mapping by assigning a unique colour to each entity
3) Read in the frame buffer object and enumerate each colour that is present; reverse map the colours back to identifiers
4) Return the list of identifiers

We implemented the occlusion detection by utilising our existing code for façade visibility determination, which we use at 3D model pre-process stage, porting it to Android with the frame buffer object management. We created an API to the Java side of Android to launch the occlusion determination, and receive the list of visible entities.

Our next step is to manage our AR content according to its visibility; we can, for example, render red 3D bus models in the AR view for any buses that are occluded by the city model in a similar manner as we do in the 3D map mode.

The drawback of the scheme is in the occlusions caused by dynamic objects such as people, cars and buses, urban furniture and vegetation. If we had very accurate tracking, we could use 3D bus models for occlusion determination.

6. Mobile App Development (T5.5)

6.1. Tampere apps

6.1.1. LOGICA's Tampere Pilot app

6.1.1.1. UI design principles

According to surveys conducted under the STREETLIFE Tampere pilot, over 70 % of travellers use the routing service from a mobile device. Consequently, it is obvious that the multimodal routing application should be designed for mobile use. In practice, this involves that:

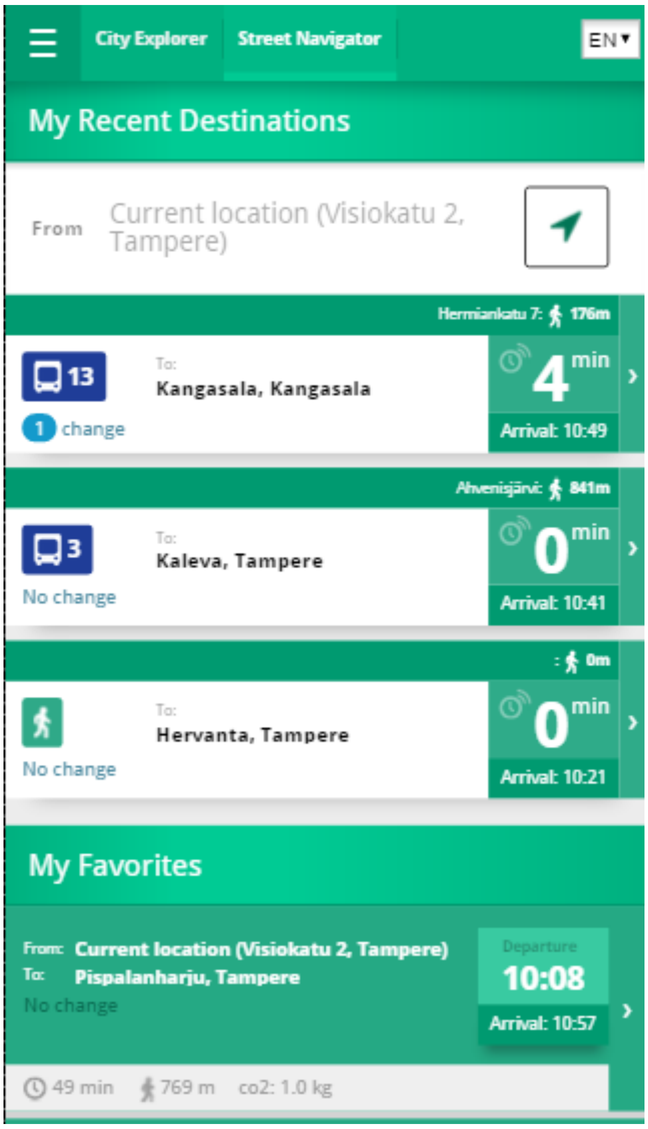
- User interface is responsive and scales/adapts to all kind of device resolutions
- The application works in all kind of modern HTML5 -capable devices and browsers
- Navigation and usage is easy, fast and intuitive. Users are not willing to perform complex actions on a small display while moving.

HTML5-requirement naturally limits usage a bit, but also enables development using modern frameworks. It is a wise trade-off, considering how quickly mobile device technology proceeds nowadays.

Multimodal routing application user interface was configured to match better mobile use. Thanks to the highly optimised backend architecture, the routing application is straightforward and easy to use. The key features contributing to this aspect are:

- Route search is launched automatically as soon as the start and the end point are defined, or whenever any search option, such as departure time, changes.
- The application remembers recent searches, and also maintains a list of favourite locations. Itineraries into these locations from the current GPS-tracked location are computed automatically as soon as the application starts.
- Users no longer have to enter various search criteria through dedicated query forms, to find their favourite route types. Variations are made automatically inside the routing service, and the application can therefore suggest a versatile collection of itineraries (soonest, fastest, least walking etc.)

In the optimal case, the user does not have to click or enter anything to get a suitable itinerary – starting the application is enough. Such an automatic multi-target routing approach naturally causes a workload for backend services tens of times higher than with traditional ‘click the search button’ interfaces.



My Recent Destinations

From Current location (Visiokatu 2, Tampere)

Hermiankatu 7: 176m
13 To: Kangasala, Kangasala
1 change
4 min
Arrival: 10:49

Ahvenisjärvi: 841m
3 To: Kaleva, Tampere
No change
0 min
Arrival: 10:41

: 0m
To: Hervanta, Tampere
No change
0 min
Arrival: 10:21

My Favorites

From: Current location (Visiokatu 2, Tampere)
To: Pispalanharju, Tampere
No change
Departure 10:08
Arrival: 10:57

49 min 769 m co2: 1.0 kg

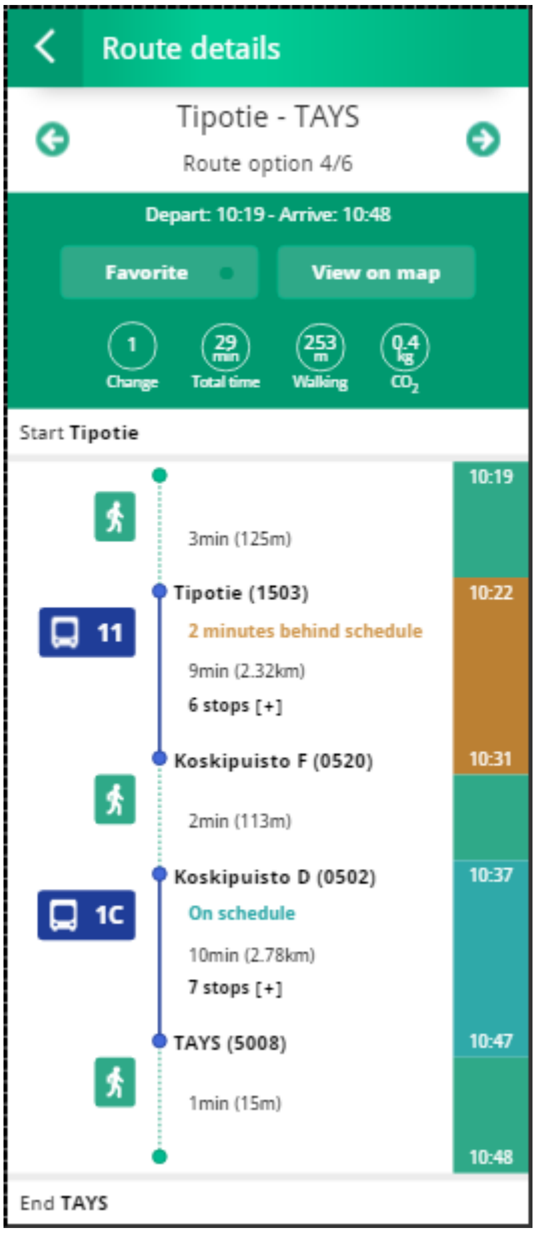
Figure 20: Automatic routing to recent destinations and favourites

6.1.1.2. Real-time routing

The routing itineraries, which the STREETLIFE Tampere pilot application fetches from the backend service automatically, include real-time corrections obtained from the bus tracking system. The closer to the current moment an itinerary is, the more accurately it will be realised.

Presenting a real-time dependent itinerary to a traveller who is already moving involves some non-obvious challenges. Real-time schedules continue changing after the user has got the itinerary, and a straightforward re-routing to update the itinerary can lead to a situation where the traveller loses the itinerary she is already following. The STREETLIFE Tampere pilot application solves this problem by sticking into a selected itinerary, and by updating the presented times. Possible delays or ahead of time departures are indicated clearly to the user;

it is her conscious decision to ask rerouting from the new reached location, if the current itinerary seems to fail.


Route details

Tipotie - TAYS

Route option 4/6

Depart: 10:19 - Arrive: 10:48

Favorite

View on map

1 Change | 29 min Total time | 253 m Walking | 0.4 kg CO_2

Start Tipotie

10:19

3min (125m)

Tipotie (1503) 10:22

11

2 minutes behind schedule

9min (2.32km)

6 stops [+]

Koskipuisto F (0520) 10:31

2min (113m)

Koskipuisto D (0502) 10:37

1C

On schedule

10min (2.78km)

7 stops [+]

TAYS (5008) 10:47

1min (15m)

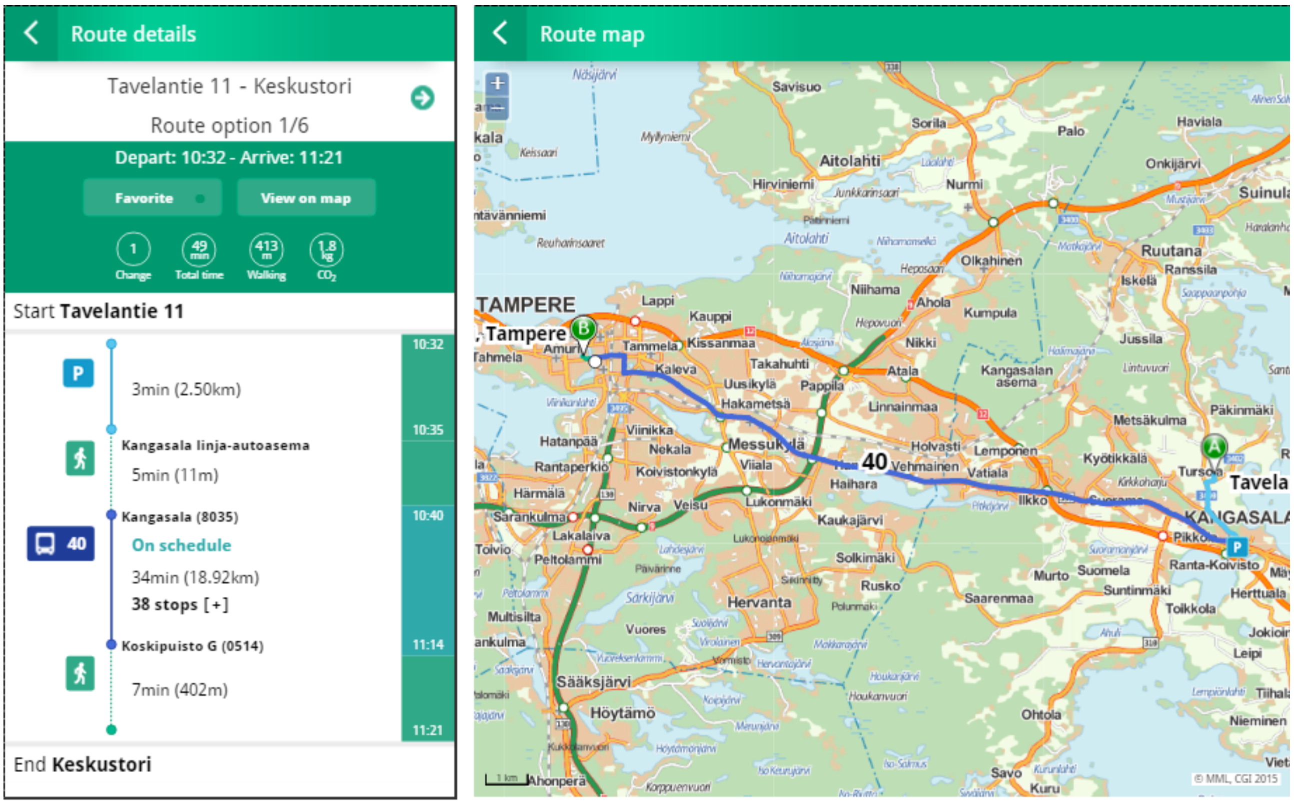
10:48

End TAYS

Figure 21: Real-time routing itinerary

6.1.1.3. Park & Ride

The STREETLIFE Tampere pilot application supports also park and ride type routing. This feature extends multi-modality to include route sections by a private car. The data loading process automatically queries real time the current set of parking halls from the local provider, and some additional free sites defined by regional traffic controllers. Congestion in the city centre reduces, when travellers know in advance where to leave their car (no driving around to find a free parking place) and also because the service favours public transit instead of driving a private car.



Route details

Tavelantie 11 - Keskustori

Route option 1/6

Depart: 10:32 - Arrive: 11:21

Favorite | View on map

1 Change | 49 min Total time | 413 m Walking | 1.8 kg CO₂

Start **Tavelantie 11**

10:32 P 3min (2.50km)

10:35 Kangasala linja-autoasema 5min (11m)

10:40 Kangasala (8035) 40 On schedule 34min (18.92km) 38 stops [+]

11:14 Koskipuisto G (0514) 7min (402m)

11:21

End **Keskustori**

Figure 22: Park&ride itinerary and the route map

The trip uses mainly public transit connections. Kangasala has a Park & ride location. The multimodal journey planner checks in real-time the availability of parking places in park and ride locations. It is important we do not guide people to the park and ride locations, which are getting full or are already full.

6.1.2. AALTO's Map Tapper for Wizard-of-Oz positioning

To accommodate accurate positioning during field experimentation, AALTO created a small helper app, *Map Tapper*, which allows an external person to re-locate our subject to a new virtual position using a map interface. Map Tapper reads touch events on a map screen and *publishes* these entity position updates to a server using our dynamic entity schema. The server transmits the updates to a *subscriber;* in this case our subject's tablet. The mechanism reproduces our entity data management fully using our networking scheme, but with a map interface. The position updates move the virtual position of the subject accordingly. Simple interpolation scheme avoids unnecessary jumping. We use Google Maps as the map interface (Figure 23).

The drawback of the Map Tapper is the requirement of a human operator to assess the positions on a map, which needs practice and sufficient resolution from the map itself. The Map Tapper can be run on any touch screen enabled Android device. In practice, best granularity is achieved with tablets with a large screen and high resolution, such as ASUS TF701T.

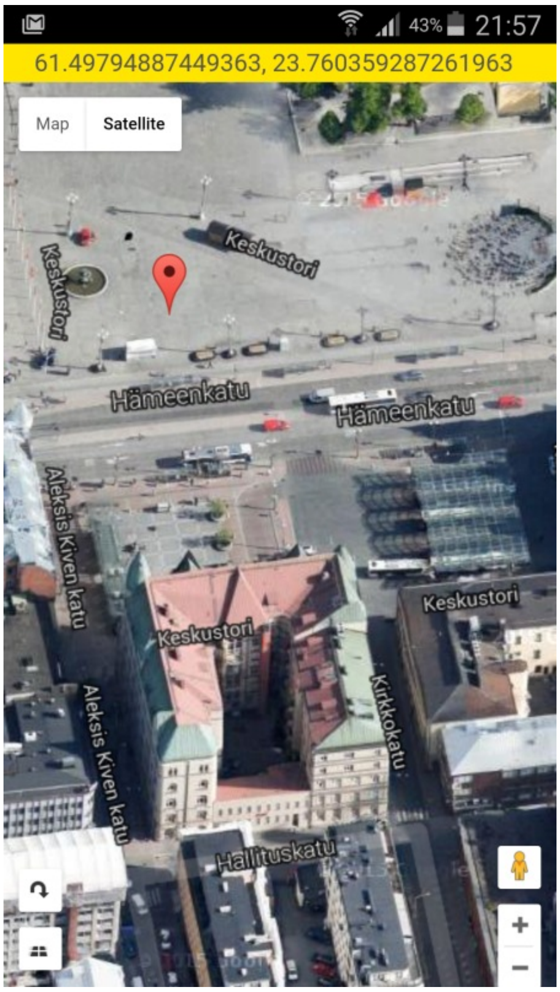


Figure 23: The Map Tapper app

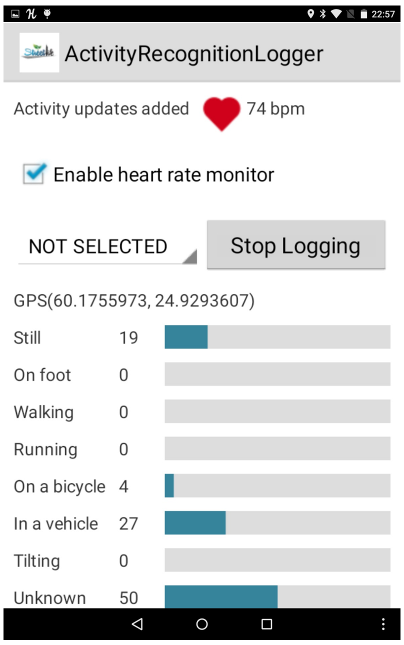
Built on top of Android's web view and Google Maps, Map Tapper allows interactive positioning of the view of a subject by the experimenter during a field experiment.

6.1.3. Standalone Activity Recognition App

AALTO has worked toward passive crowdsensing and modality recognition by creating a standalone app as part of work in T3.3 (Crowdsourcing) and T5.5 (Mobile app development). The *Activity Recognition Logger* (Figure 24) logs and transmits a rich user state using APIs that are available on Android devices:

- Position (internal or external GPS)
- Orientation (internal IMU)
- Heart rate (external heart rate device)
- Illumination (internal sensor)
- Ambient noise (microphone)

- Ambient and device temperature (internal sensors)
- Device tilting (Google Play services)
- Approximate modality as a set of probabilities (Google Play services)
- User-selectable modality (in addition to Google Play services estimate; optional)



Activity updates added 74 bpm

☑ Enable heart rate monitor

GPS(60.1755973, 24.9293607)

Still	19
On foot	0
Walking	0
Running	0
On a bicycle	4
In a vehicle	27
Tilting	0
Unknown	50

Figure 24: A standalone app for crowdsensing and modality detection

6.1.4. Crowd-sourcing features of the Mixed Reality App

A feedback mechanism was added to the TRE Mixed Reality app, to allow users to share their opinions on bus routes and bus stations. Figure 25 presents the feedback screen. The data is transmitted to AALTO's server holding a local database, and a Datex II service provides an interface to CGI or any other external organisation for the data. The data includes bus line number and bus station name with related usability feedback, timestamp, application id and location as latitude/longitude pair.

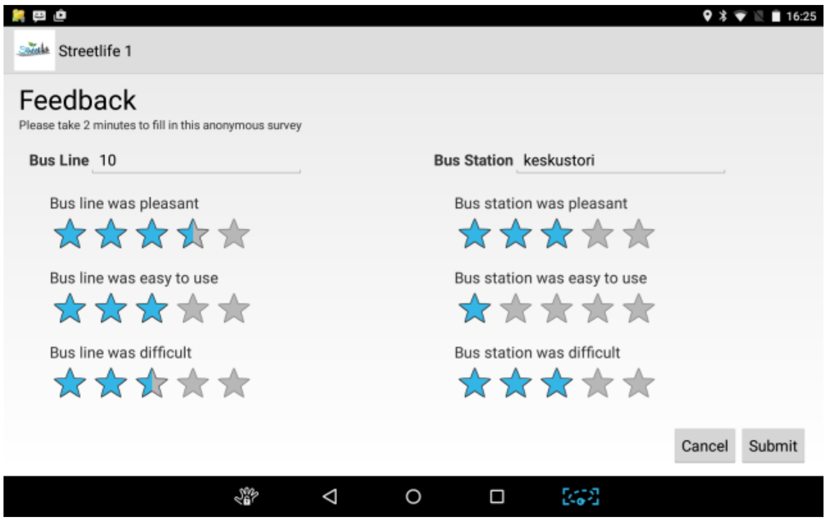
Data queries for user feedbacks from the Datex II service are defined by start and end date:

```
data.xml?start=2015-09-23&end=2015-09-24
```

A full request example:

http://streetlife.logbot.org/9a6f79cab3bf5af89f940922a4470b6f/data.xml?start=2015-09-23&end=2015-09-24

The Datex II service was implemented on *Nginx* web server, with Node.js web application built on the Express framework, using Sqlite3 database for storage. Data retrieval causes the data to be queried from the database and rendered as Datex II compatible XML.



Feedback

Please take 2 minutes to fill in this anonymous survey

Bus Line 10

Bus Station keskustori

Bus line was pleasant

Bus station was pleasant

Bus line was easy to use

Bus station was easy to use

Bus line was difficult

Bus station was difficult

Figure 25: TRE Mixed Reality app user feedback form

6.2. Berlin apps

6.2.1. The STREETLIFE Berlin mobility app

The STREETLIFE Berlin mobility app is the main App that interfaces users in Berlin with the route planning and travel assistance subsystem of STREETLIFE. The app itself is a multi-modal trip-planner promoting sustainable as well as energy efficient mobility. It was already introduced in D5.2.1. Here we discuss the most important updates to the app that we have been developing within the last year.

The app was newly developed during the 1st iteration of STREETLIFE. During the 2nd iteration we further integrated novel features like speech input and smart watch interaction. Therefore the app is still available as a prototype only for members of the STREETLIFE consortium and for participants of the evaluations. In the following, besides changes resulting from the results of the 1st iteration, we list a number of other advancements that have been developed (or are under development at the moment of writing), which will be leveraged and validated within the 2nd iteration of the BER pilot.

- We integrated smart watch based interaction. Additionally to the mere smart watch interaction depicted in Section 2.3.1.2 that allows only to request personalized routes for bicycling, we implemented a synchronisation of the routing information between the smart phone and the smart watch. Itineraries that are requested by means of the smart phone app are automatically forwarded to the smart watch. Figure 24 shows how the information is displayed on both devices. Using this feature the smart watch can also be utilized for non-bicycle routes.

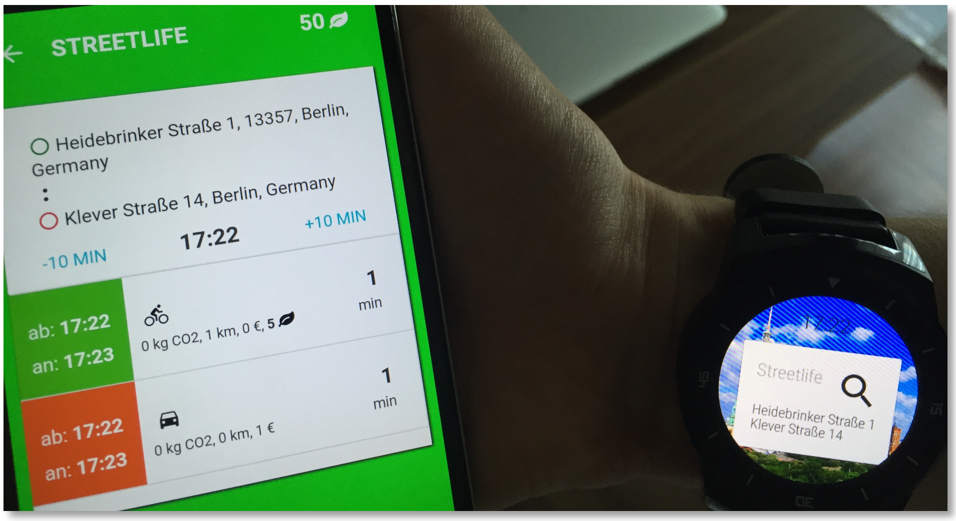
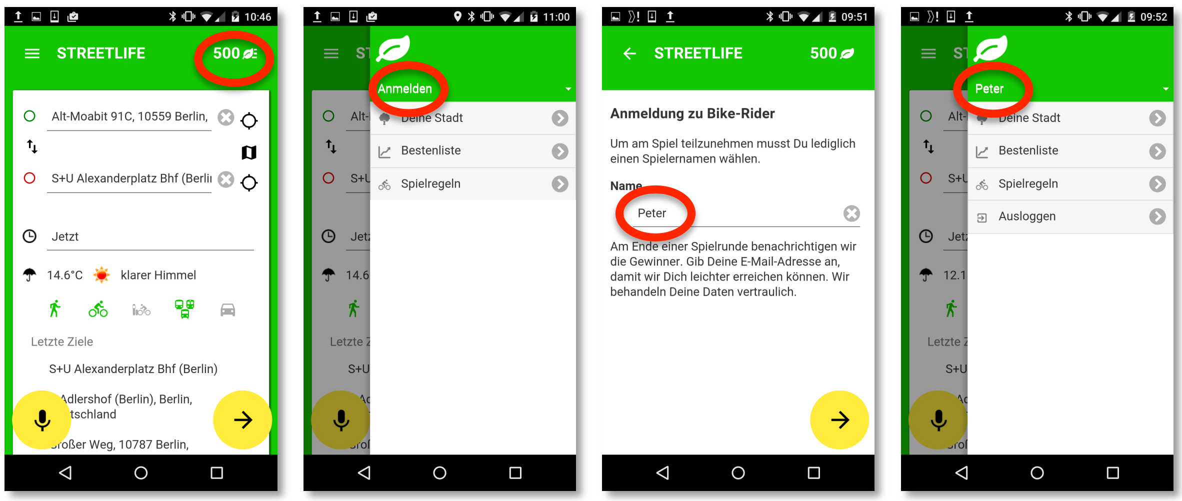


Figure 26: Synchronisation or route information between smart phone and smart watch

- We implemented end-of-route detection and route deviation recognition as depicted in Sections 2.3.2.2 and 2.3.2.3.
- We integrated speech input for entering route information as already depicted in Section 2.3.1.1.
- We removed the login, which was necessary to start the companion mode of the app. During the companion mode the actual itinerary is monitored. Further log data about the actual route is only stored in the back-end if the companion mode is activated. In order to enabled analysis of data in the back-end it is necessary to simplify the usage of the system as far as possible.
- In the course of usability and the bicyclist focus of the BER pilot we removed the individual profile configuration and the manual adjustment of cost functions of the routing request. The 1st iteration experiments revealed that the users forget to re-adjust their profile if they made special settings. As a result the subsequent routing proposals can be impractical as they also relate to these special settings. The focus on bicyclists further revealed new requirements, which is considered by the adaption of the user preferences depicted in Section 2.1.
- Gamification and crowdsourcing: Together with BER partners we developed a Gamification concept, which was integrated in the BER mobility app. The crowdsourcing feedback screens are embedded into routing and Gamification. Here we list a set app sequences in order to illustrate the developments:
    - Subscription: in order to subscribe for the game it is just necessary to choose an individual user name. From left to right Figure 27 illustrates the subscription sequence. In left screen the newly integrated Gamification menu can be viewed. The red circle illustrates the steps and changes that are implemented by the sequence.

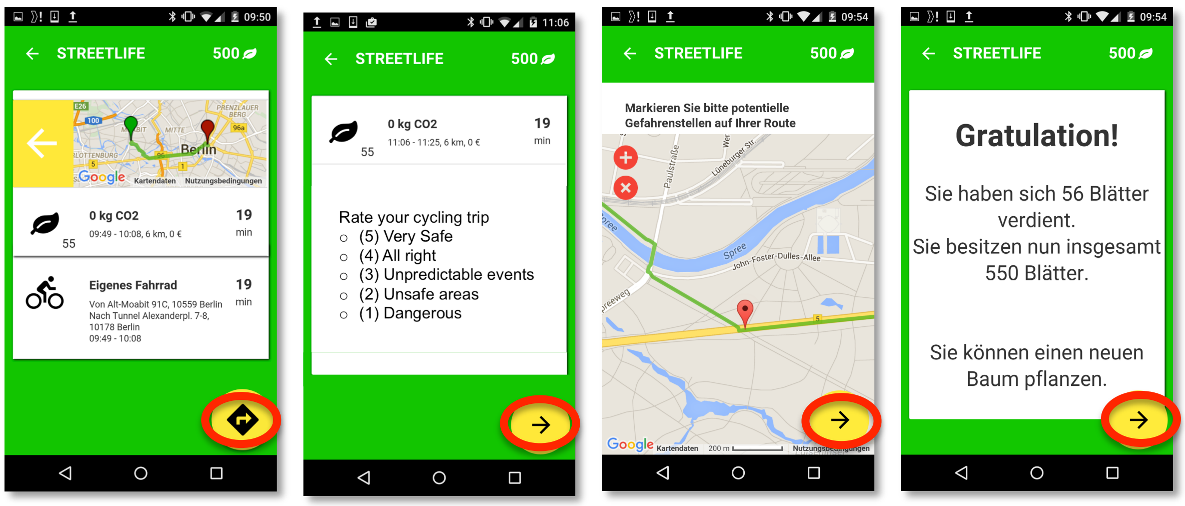

STREETLIFE
500
Alt-Moabit 91C, 10559 Berlin,
S+U Alexanderplatz Bhf (Berlin
Jetzt
14.6°C
klarer Himmel
Letzte Ziele
S+U Alexanderplatz Bhf (Berlin)
Adlershof (Berlin), Berlin, Deutschland
Großer Weg, 10787 Berlin,
Anmelden
Deine Stadt
Bestenliste
Spielregeln
Anmeldung zu Bike-Rider
Name
Peter
Ausloggen

Um am Spiel teilzunehmen musst Du lediglich einen Spielernamen wählen.

Am Ende einer Spielrunde benachrichtigen wir die Gewinner. Gib Deine E-Mail-Adresse an, damit wir Dich leichter erreichen können. Wir behandeln Deine Daten vertraulich.

Figure 27: Subscription sequence

- Routing, Crowdsourcing, and Gamification: for "green" routes the users receive green leaves. From left to right Figure 28 illustrates the routing and gaming sequence. In the left screen the companion mode must be activated. After the end of the route has been automatically detected next the crowdsourcing screens are shown, while the first allows to give feedback about a specific question about the rout and the second allows to mark dangerous points on a map. The last screen shows the new Gamification status.


STREETLIFE
500
Berlin
0 kg CO2
55
09:49 - 10:08, 6 km, 0 €
19 min
Eigenes Fahrrad
Von Alt-Moabit 91C, 10559 Berlin
Nach Tunnel Alexanderpl. 7-8, 10178 Berlin
09:49 - 10:08
11:06 - 11:25, 6 km, 0 €
Markieren Sie bitte potentielle Gefahrenstellen auf Ihrer Route
Gratulation!

Rate your cycling trip
○ (5) Very Safe
○ (4) All right
○ (3) Unpredictable events
○ (2) Unsafe areas
○ (1) Dangerous

Sie haben sich 56 Blätter verdient.
Sie besitzen nun insgesamt 550 Blätter.

Sie können einen neuen Baum pflanzen.

Figure 28: Routing and gaming sequence

- Planting virtual trees: every time 500 new green leaves have been collected the users can plant a virtual tree. From left to right Figure 29 illustrates how virtual trees can be planted. In the left screen a notification is shown beside the Gamification menu. In the menu the map can be accessed under "Deine Stadt". Trees of other users are displayed on the map and the ownership of the trees can be viewed by tapping on a tree. By selecting a shape and colour the new "own tree" can be individualized.

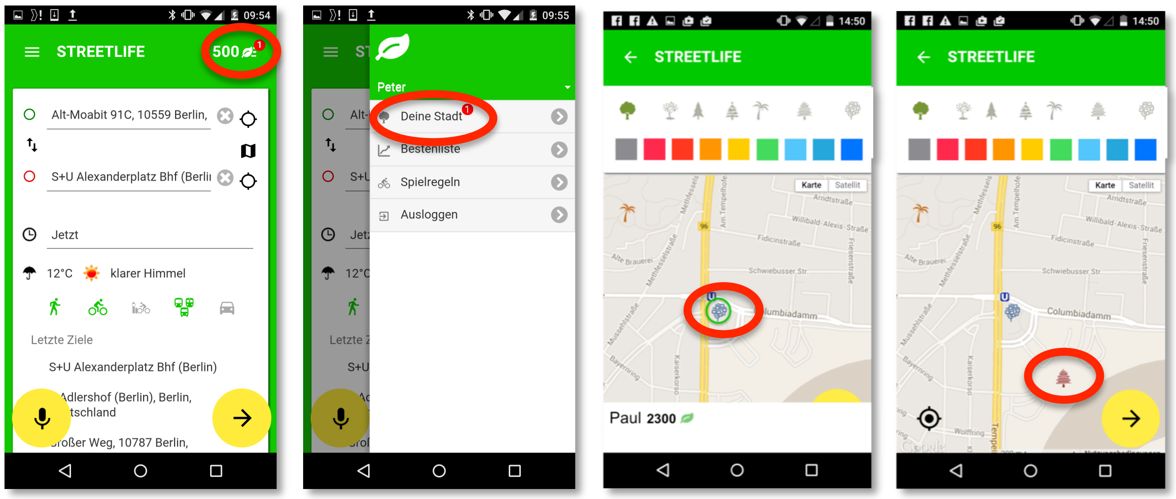
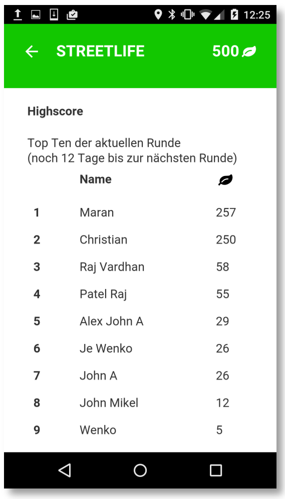

STREETLIFE
500
Alt-Moabit 91C, 10559 Berlin,
S+U Alexanderplatz Bhf (Berlin
Jetzt
12°C
klarer Himmel
Letzte Ziele
S+U Alexanderplatz Bhf (Berlin)
Adlershof (Berlin), Berlin, Deutschland
Großer Weg, 10787 Berlin,
Peter
Deine Stadt
Bestenliste
Spielregeln
Ausloggen
Karte
Satellit
Paul 2300

Figure 29: Planting virtual trees

- Leader board: a leader board showing the top ten gamers and the own position can be accessed over the Gamification menu (Figure 30).


STREETLIFE 500

Highscore

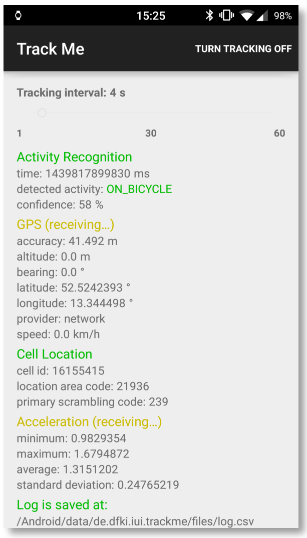
Top Ten der aktuellen Runde
(noch 12 Tage bis zur nächsten Runde)

	Name	
1	Maran	257
2	Christian	250
3	Raj Vardhan	58
4	Patel Raj	55
5	Alex John A	29
6	Je Wenko	26
7	John A	26
8	John Mikel	12
9	Wenko	5

Figure 30: Leader board

6.2.2. The TrackMe app

The TrackMe app is a standalone implementation of the tracking component that has been developed as part of the BER mobility app. The aim of having a standalone version is twofold: (1) other pilots can inspect the functionalities in order to determine if the integration of the tracking component into their own pilot developments might be valuable. (2) The BER partners developing components that make use of tracking data can utilise the TrackMe app to produce data sets needed for the development of their components before the tracking functionality is fully available within the BER pilot mobility app.


15:25 98%

Track Me TURN TRACKING OFF

Tracking interval: 4 s

1 30 60

Activity Recognition
time: 1439817899830 ms
detected activity: ON_BICYCLE
confidence: 58 %

GPS (receiving...)
accuracy: 41.492 m
altitude: 0.0 m
bearing: 0.0 °
latitude: 52.5242393 °
longitude: 13.344498 °
provider: network
speed: 0.0 km/h

Cell Location
cell id: 16155415
location area code: 21936
primary scrambling code: 239

Acceleration (receiving...)
minimum: 0.9829354
maximum: 1.6794872
average: 1.3151202
standard deviation: 0.24765219

Log is saved at:
/Android/data/de.dfki.iui.trackme/files/log.csv

Figure 31: TrackMe app

6.3. Mixed reality app for Berlin

The main hypothesis for AALTO's Mixed Reality app are related to providing a direct visualisation of travel related content, which should lessen the confusion of users in finding their key targets, i.e. bus stops. The goal of this research is in finding whether 3D maps or Augmented Reality would succeed in this manner – are these interfaces more natural and comprehensive than traditional abstract maps. While TRE provides the main case area, it was felt that the Mixed Reality demonstrator could be useful for other cities as well. As a Horizontal Integration effort (See D2.2.2 for STREETLIFE Integration Concept), the MR app was decided to support another site as well.

MR app's 3D map is dependent on the availability of a 3D city model. The AR side, without computer vision based localisation, is independent of existing AR specific data sets: if an initial pose can be established, sensor or computer vision based tracking are available for interactive use. Both 3D maps and AR require relevant point-of-interest data sets.

Two 3D models of Berlin were available, of highly different modelling schemes. The first, limited in size, was based on very large texture atlases and city blocks, which did not suit our optimisation paradigm well. The second contained the entire Berlin with over 1 million pieces of geometry and related textures. We decided to provide support for this very large model.

Unfortunately, our current pipeline for model processing could not handle such large environments in memory. Section 5.1 describes our improvements towards scalable model processing.

6.3.1. Small-scale Berlin Mixed Reality

While we work on the large-scale Berlin 3D model pre-process, we also selected a smaller piece of the model and used our existing pipeline for creating a simple demonstrator of the Berlin Mixed Reality app. The demonstrator consists of an area approximately between the Zoologischer Garten railway station and the Europa Centre, with nearby surroundings. With an area of 1.65x1.3km, 727 buildings and 24789 textures, we used our existing process pipeline for the small-scale Berlin Mixed Reality app.

For content, we received a data set on U-Bahn entrances, U-Bahn and S-Bahn platforms, other underground locations and bus stops. As usual with digital content, we needed to clean the data to limit the data to just U-Bahn ground level entrances and bus stops. The coordinate system of the data was matched with the 3D model with a simple coordinate transformation, allowing us to do local testing with GPS in this limited area (outside the area a scaling error will become significant). Figure 32 presents the U-Bahn entrances and bus stops on Google Earth.

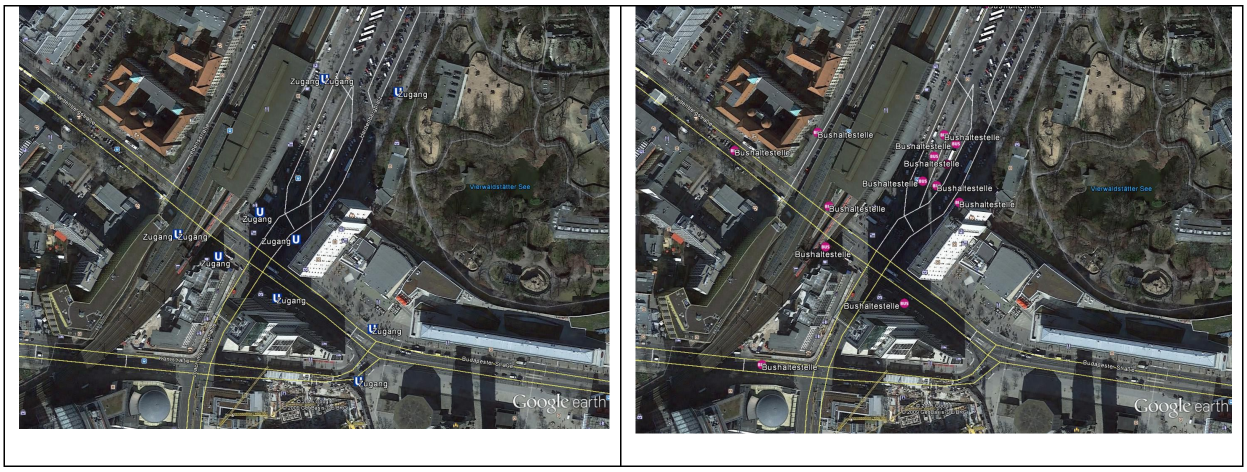


Figure 32: A data set of U-Bahn entrances and bus stops on the Zoologische Garten railway station area in Berlin

Figure 33 presents screenshots from the small-scale Berlin Mixed Reality app with U-Bahn entrances (upper row), bus stations (lower left) and both combined (lower right).

The issue of model quality comes into play with the Berlin model. The 3D model includes buildings with varying quality of textures, but no ground model nor ground textures. In a 3D environment, lack of these features may lessen the usefulness of the model in assistance of entrance or bus stop spotting. However, given an accurate augmented reality registration, these icons might become quite useful.

The main idea for the Berlin demonstrator is to find use cases for the advanced graphical interfaces that would ease traveling in Berlin.

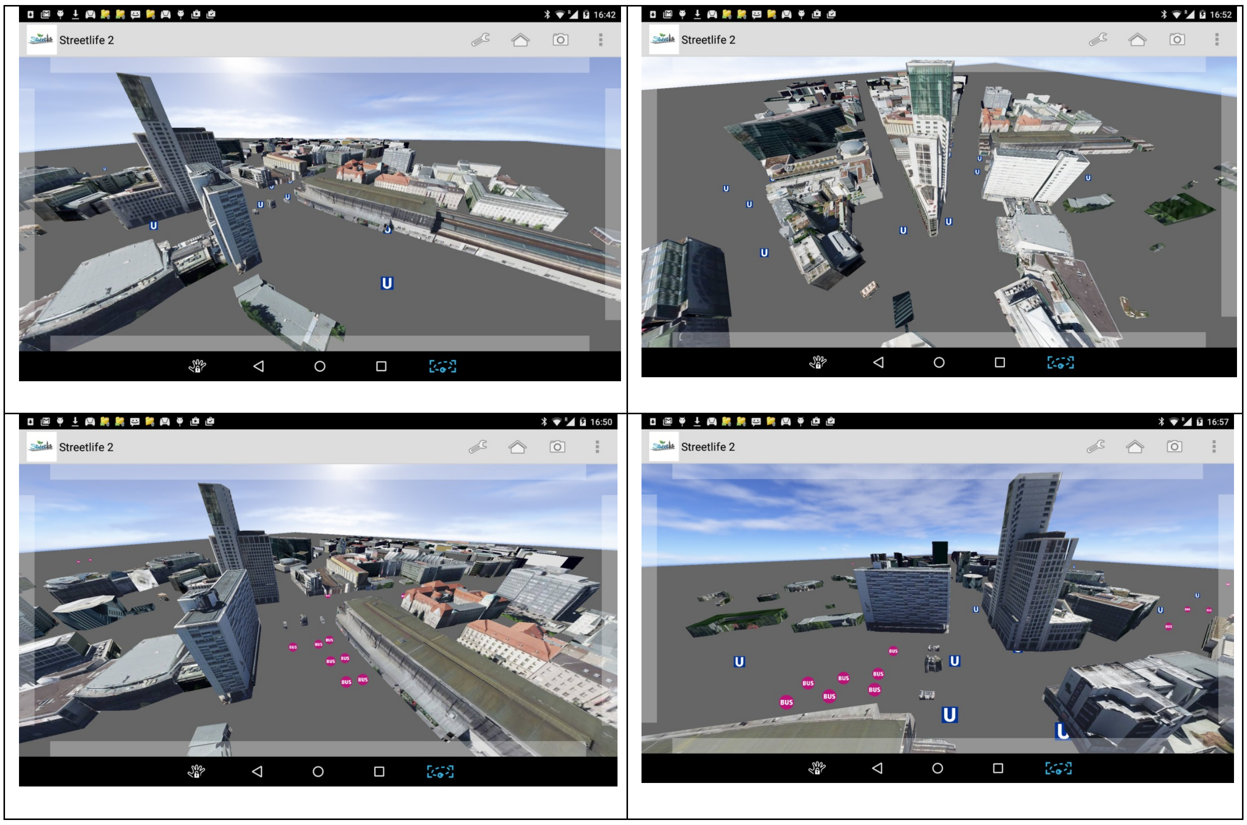


Figure 33: U-Bahn entrances and bus stops on the Zoologische Garten railway station area in Berlin in the Berlin Mixed Reality app. Top: U-Bahn entrances; Lower left: Bus stops; Lower right: combined data set

6.4. Rovereto applications (FBK)

6.4.1. ViaggiaRovereto updates and new release

ViaggiaRovereto is the main App that interfaces users in Rovereto with the route planning and travel assistance subsystem of STREETLIFE. As the App itself was already introduced in D5.2.1, we discuss here the most important updates to ViaggiaRovereto that we have been developing within the context of the project.

First of all, it is important to remark that, for the experiments of the first iteration of the ROV pilot, we had introduced initially an experimental version of ViaggiaRovereto, that is, a fork with respect to the App as publicly available on the Google Play App Store. Following the positive results of the STREETLIFE pilot, the features included in that experimental version have been integrated with the main version of the App, and are available to the whole user base of ViaggiaRovereto. The latest public release of ViaggiaRovereto that is available on Google Play thus incorporates several STREETLIFE features, such as promotion of "greenest" itinerary options and itineraries pushed because of city policies, information on parking availabilities and on the time required to search for an on-street parking, possibility to save itineraries and receive notifications about potential issues affecting the trip, integration with Bike Sharing app.

Besides the integration of STREETLIFE results from Y1 and the pilot in the mainstream ViaggiaRovereto release, we list below a number of other advancements that have been developed (or are under development at the moment of writing), which, once again, will be leveraged and validated within the ROV pilot (2nd iteration).

- Development of a browser-based Web interface for ViaggiaRovereto, besides the native Android mobile App GUI. This was prompted by observations of, and feedback by, tourists and other occasional visitors to the city of Rovereto during the first iteration of the ROV pilot: a browser-based GUI lowers the entry barrier into the system for such occasional users, who may not necessarily want to download an additional App on their mobile terminal; consequently, it has potential to increase penetration and usage rate of the App.
- We are in the process of delivering a new *cross-platform* release of the ViaggiaRovereto App, which will thus become available also on the iOS (iPhone and iPad devices) mobile operating system, in addition to the Android mobile operating system; again, this will favour penetration and usage rate with a wider public of citizens and visitors.
- We have integrated in the itinerary options that have at least a car leg information about hourly parking costs, that, together with the estimated additional time due to searching for a parking spot, suggests the "real value" of the trip, and makes more sustainable itineraries more attractive; this information originates from the integration of parking data in the backend of the STREETLIFE sustainable mobility information system, including crowdsourced parking availability and cost data from the *ContaParcheggi* App;
- We have integrated in the itinerary options that have one or more Public Transportation legs the cost of PT fares related to those legs;
- We are integrating an estimate of Carbon emissions for all the itinerary options presented by the App;
- We are integrating ViaggiaRovereto with the new ROV CarPooling App (see below); the idea is that those ViaggiaRovereto users who also participate in car pooling will also be able to include car pooling in their planning, and will hence presented with their car pooling options for a trip, in addition to all the other multi-modal itinerary options and recommendations currently retrieved by the ViaggiaRovereto App.

6.4.2. ContaParcheggi official release

This mobile app allows collecting crowdsensed data about parking availability for on-street parking and parking lots. The key functionalities supported by the app are i) an interactive map that provides an overview of parking lots and on-street parking segment, ii) the possibility to select a certain parking lot/segment (either from the map or through free text search) and to specify the number of free parking spaces (differentiated by parking type: free parking, parking in parking meter, disc parking) and unusable parking spaces (e.g. due to road works), iii) for each parking lot/street segment, inspect the list of previous notifications.

The collected data is aggregated through the Data Management Rovereto sub-system (D3.2.2), and exploited by ViaggiaRovereto mobile app to show real- time parking lots occupancy and estimated time to find an on-street parking.

The app has been evaluated through an on-the-field experiment involving Rovereto traffic aids and has been revised, both in terms of offered functionalities and of usability, according to the received feedback. The app has been officially released and is currently adopted by Rovereto traffic aids in their daily activities.

The director of AMR (Rovereto parking company) gave a very positive feedback, considers this package marketable and is willing to test and promote it among other parking companies and municipalities.

6.4.3. CarPooling App

Car-pooling refers to a mode of transportation in which individual travellers share a private vehicle for a trip with others that have similar itineraries and time schedules. Although **car-pooling presents several benefits** both from a **personal perspective** (i.e., reduced travel cost for fuel, tolls and parking, reduced stress for driving) and from an **environmental and mobility perspective** (i.e., reduced carbon emissions, traffic and parking occupancy), so far it has obtained limited success [22, 23]

We started by reviewing the state-of-the-art and state-of-the-practice of car-pooling systems and initiatives in order to understand the factors that might have restricted a widespread adoption and to derive the attributes of a car-pooling concept that has the potential to improve its attractiveness and resilience.

Two main problems came out to be the **key limiting factors for car pooling**: i) the psychological barriers associated with **riding with strangers** and ii) **poor schedule flexibility**, particularly in handling schedule/route variations and aggravated by the dependence and impact on other persons [23-26]. Not surprisingly, car-pooling success cases can be divided in two main categories:

- Sharing of long-distance trips (e.g., BlaBlaCar), where flexibility is not a critical aspect and personal advantages, in terms of reduced cost and driving stress, become significant;
- Car pooling for commuters belonging to the same employment centre [27-31], where the work organisation establishes a base trust level between participants and customised incentive programs are set up by employers to encourage employees to carpool (e.g., reduced cost or free parking, more flexible work schedule, or specific reward programs).

Long distance trips, although effective from an acceptance perspective, have a limited impact in terms of carbon emissions and traffic reduction. On the other hand, dynamic form of instant car pooling, that could lead to great reductions of single occupant vehicle trips, have been tested before but have proved to be very ineffective when applied independently [23, 32, 33]

To overcome these limitations and fully exploit the advantages of car pooling, **we have defined a car pooling model that is based on two key features**: i) it is based on the concept of **community**, establishing a base trust level among participants, and ii) supports **dynamic ride matching** either within the community or in an alternative group, when the pool member has a trip schedule different from the usual one.

From a psychological perspective, the community structure provides a common basis for establishing a relationship of the kind needed to form car pooling teams: "creating a group whose members have similar interests will tend to make the group more attractive, and emphasising to group members their unique skills or knowledge will tend to make them believe their effort matter" [34].

Examples of communities could be groups of employees sharing similar journey needs (in terms of route, work schedule), people attending the same sport club, parents chauffeuring their children to the same school, but also groups of friends going to a specific event.

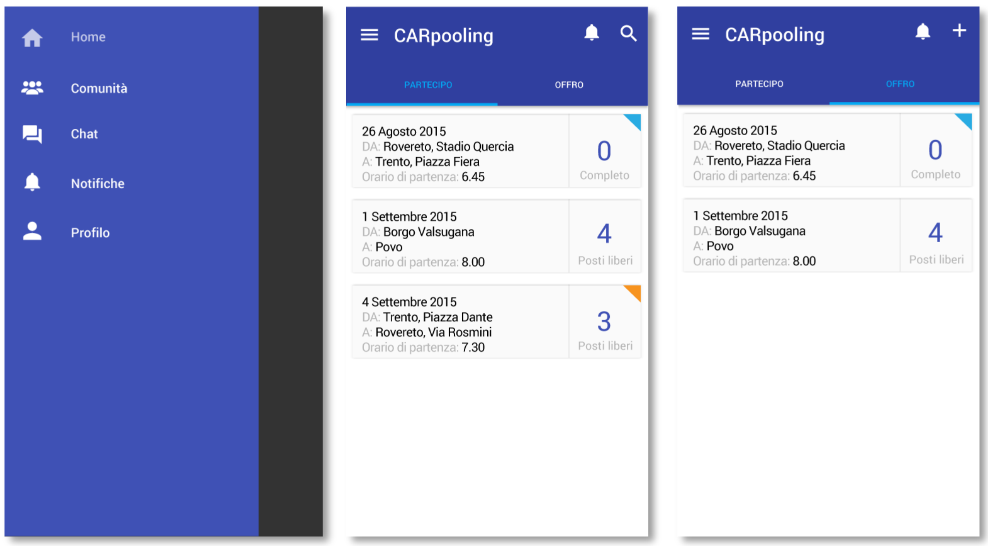
The key functionalities offered by the app are: offering rides, either within a community or to all users; searching for rides, through dynamic ride matching, either within a specific community or from all users; confirming and validating shared rides; providing feedback on drivers/passengers; joining or leaving a community; interacting with the other car poolers via chat.

Concerning incentive programmes, the Car Pooling app allows to validate rides (as well as saving user participation and usage data) and will exploit the features offered by the STREETLIFE Gamification Framework to define car pooling-specific games and incentives.

The STREETLIFE Car Pooling app at the moment does not support any payment procedure between the involved parties: how to share the costs of gas, toll, and parking, and how they handle the transaction.

In the rest of the Section we present in details the key functionalities offered by the app.

6.4.3.1. Homepage and Navigation drawer



26 Agosto 2015
DA: Rovereto, Stadio Quercia
A: Trento, Piazza Fiera
Orario di partenza: 6.45
0 Completo

1 Settembre 2015
DA: Borgo Valsugana
A: Povo
Orario di partenza: 8.00
4 Posti liberi

4 Settembre 2015
DA: Trento, Piazza Dante
A: Rovereto, Via Rosmini
Orario di partenza: 7.30
3 Posti liberi

26 Agosto 2015
DA: Rovereto, Stadio Quercia
A: Trento, Piazza Fiera
Orario di partenza: 6.45
0 Completo

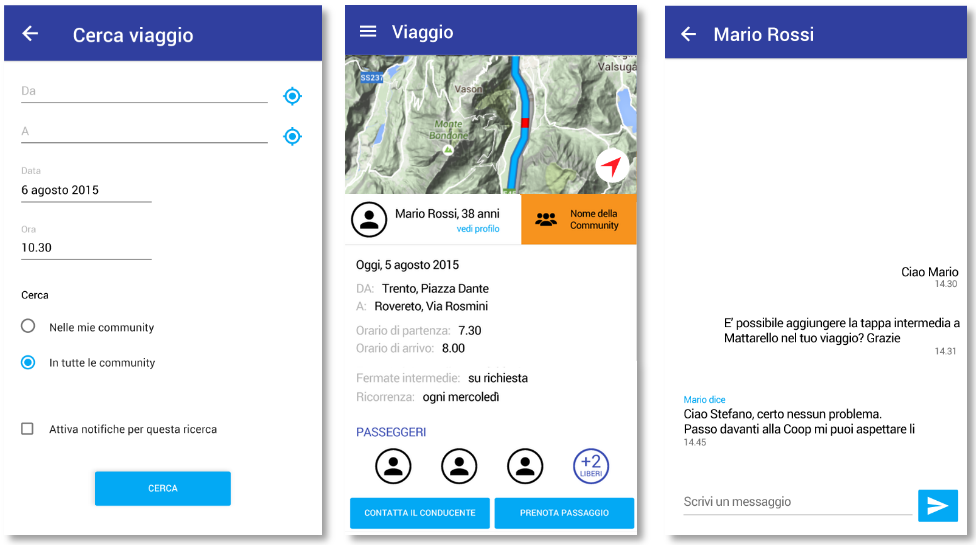
1 Settembre 2015
DA: Borgo Valsugana
A: Povo
Orario di partenza: 8.00
4 Posti liberi

Figure 34: Navigation drawer

Within the Homepage the user has an overview of the car-pooling rides he booked (tab "PARTECIPO") and offered (tab "OFFRO"). For each ride, the app shows the scheduled date, departure and destination location, time of departure, as well as the community the rides belong to (identified through a colour characterising the community) and the number of free places. From "PARTECIPO" action bar the user can search for a new ride (magnifier icon), or access the list of notifications (bell icon). From "OFFRO" action bar the user can offer a new ride (plus icon) and access the notifications.

The navigation drawer provides access to the other functionalities offered by the app: communities ("Comunità"), online chat ("Chat"), notification list ("Notifiche"), and personal profile ("Profilo").

6.4.3.2. Searching and booking a ride


Cerca viaggio

Da

A

Data
6 agosto 2015

Ora
10.30

Cerca
Nelle mie community
[✓ In tutte le community]

Attiva notifiche per questa ricerca

CERCA

Viaggio

Mario Rossi, 38 anni
vedi profilo
Nome della Community

Oggi, 5 agosto 2015
DA: Trento, Piazza Dante
A: Rovereto, Via Rosmini
Orario di partenza: 7.30
Orario di arrivo: 8.00
Fermate intermedie: su richiesta
Ricorrenza: ogni mercoledì

PASSEGGERI
+2 LIBERI

CONTATTA IL CONDUCENTE
PRENOTA PASSAGGIO

Mario Rossi

Ciao Mario
14.30

E' possibile aggiungere la tappa intermedia a Mattarello nel tuo viaggio? Grazie
14.31

Mario dice
Ciao Stefano, certo nessun problema.
Passo davanti alla Coop mi puoi aspettare li
14.45

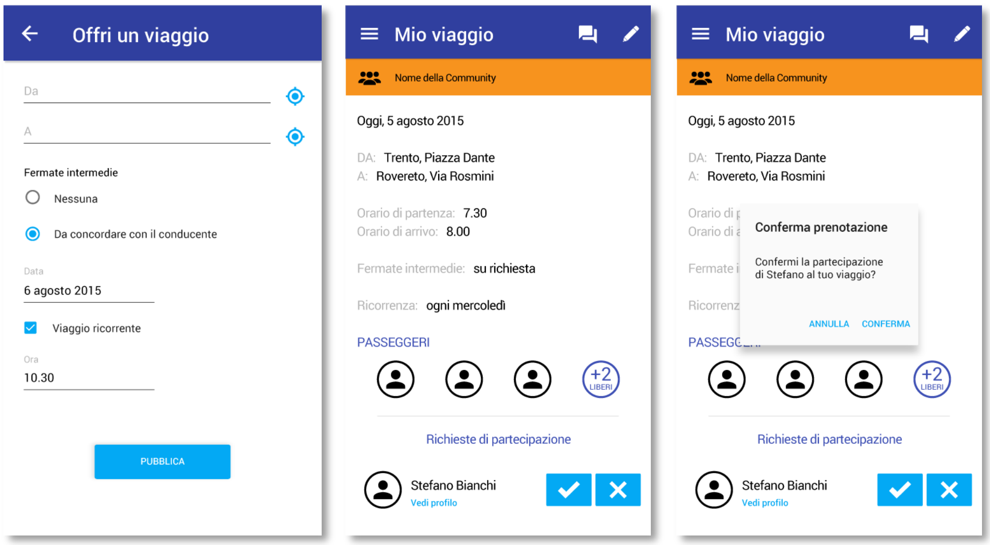
Scrivi un messaggio

Figure 35: Searching and booking

When searching for a ride, the user specifies origin, destination, date and time of the trip and the application presents a list of possible matching rides. The user has the possibility of searching within his own communities or among rides offered to all users. In the case there are no rides satisfying his needs, the user can activate notifications for his request: as soon as there are new matching rides, he will be notified.

For each offered ride, the user can access the "Viaggio" screen and inspect the following details: standard information about the trip (origin, destination, date, departure and arrival time) as well as the profile of the driver, the community, current passengers and available places, whether the driver accepts intermediate stops, and whether the ride is a single trip or a recurrent one (e.g., every Wednesday morning). From "Viaggio" screen the user can book the ride ("Prenota passaggio") or start a chat session with the driver ("Contatta conducente").

6.4.3.3. Offering a ride and handling passenger's booking



Fermate intermedie
Nessuna
[✓ Da concordare con il conducente]
Data
6 agosto 2015
[✓ Viaggio ricorrente]
Ora
10.30

Oggi, 5 agosto 2015
DA: Trento, Piazza Dante
A: Rovereto, Via Rosmini
Orario di partenza: 7.30
Orario di arrivo: 8.00
Fermate intermedie: su richiesta
Ricorrenza: ogni mercoledì
Stefano Bianchi

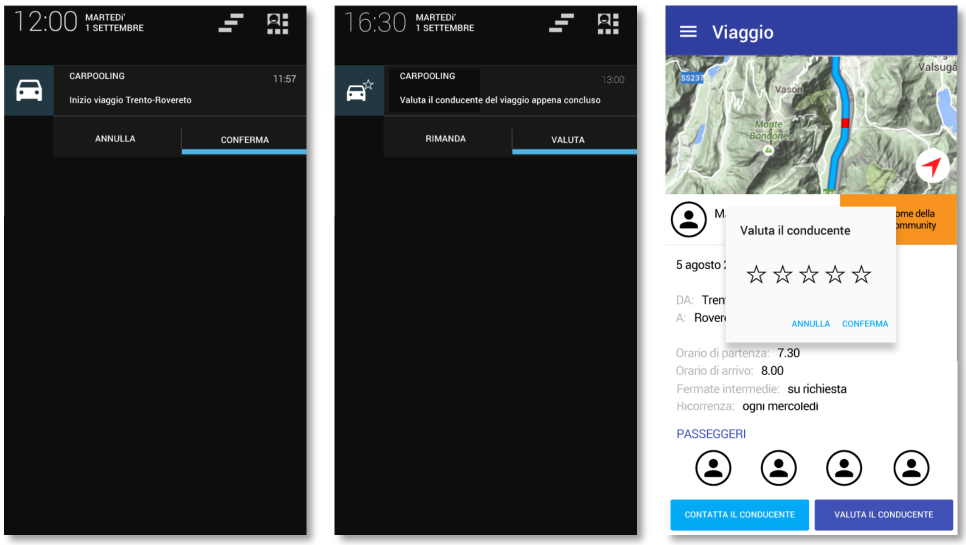
Conferma prenotazione
Confermi la partecipazione di Stefano al tuo viaggio?

Figure 36: Booking handling

Offering a ride ("Offri un viaggio" screen) requires to specify the itinerary details (departure and arrival location, date and time), the community of users to share the ride with, the possibility of having intermediate stops and whether it is a recurrent ride or not. Once the ride is published ("Pubblica" button) it appears in the offered rides of the driver (Homepage) and can be searched/booked by passengers.

When a passenger books a ride, the driver is notified and can access "Mio viaggio" screen to confirm or refuse the booking. From this screen he can also see the details of the ride, edit it (through the pencil icon in the action bar) and access the profiles of current passengers.

6.4.3.4. Confirming and rating rides


12:00 MARTEDI' 1 SETTEMBRE
CARPOOLING 11:57
Inizio viaggio Trento-Rovereto
ANNULLA CONFERMA

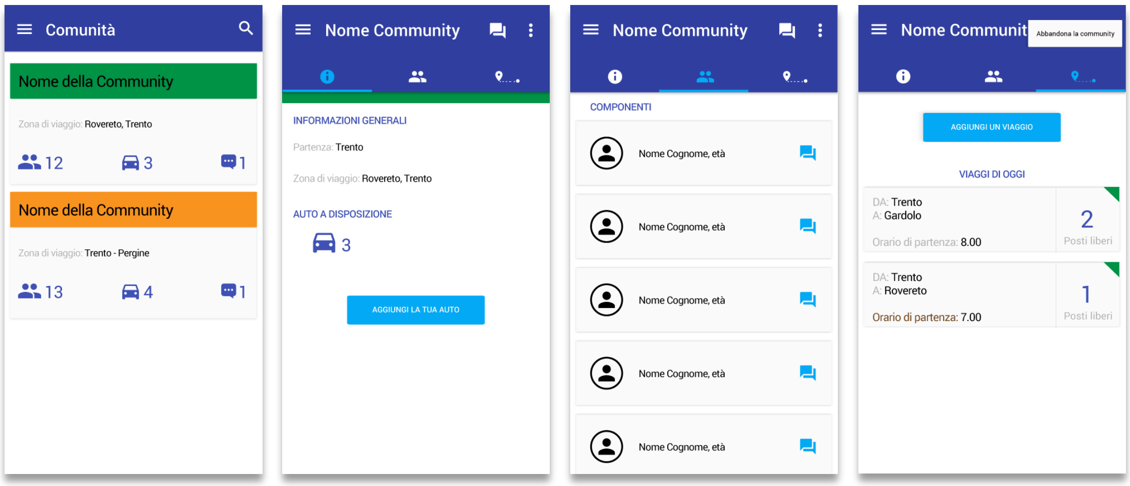
16:30 MARTEDI' 1 SETTEMBRE
CARPOOLING 13:00
Valuta il conducente del viaggio appena concluso
RIMANDA VALUTA

Viaggio
Valuta il conducente
☆ ☆ ☆ ☆ ☆
ANNULLA CONFERMA
5 agosto
DA: Tren
A: Rover
Orario di partenza: 7.30
Orario di arrivo: 8.00
Fermate intermedie: su richiesta
Ricorrenza: ogni mercoledì
PASSEGGERI
CONTATTA IL CONDUCENTE
VALUTA IL CONDUCENTE

Figure 37: Ride rating

The app supports the confirmation of a ride through notifications sent to the passengers: whenever a ride is supposed to start, the passengers receive a notification and are asked to confirm that the driver has picked them up. Similarly, both passengers and driver can express their feedback on a ride by rating the driver/passengers.

6.4.3.5. Communities


Comunità
Nome della Community
Zona di viaggio: Rovereto, Trento
12 3 1
Nome della Community
Zona di viaggio: Trento - Pergine
13 4 1

Nome Community
INFORMAZIONI GENERALI
Partenza: Trento
Zona di viaggio: Rovereto, Trento
AUTO A DISPOSIZIONE
3
AGGIUNGI LA TUA AUTO

Nome Community
COMPONENTI
Nome Cognome, età
Nome Cognome, età
Nome Cognome, età
Nome Cognome, età
Nome Cognome, età

Nome Communit
Abbandona la community
AGGIUNGI UN VIAGGIO
VIAGGI DI OGGI
DA: Trento
A: Gardolo
Orario di partenza: 8.00
2 Posti liberi
DA: Trento
A: Rovereto
Orario di partenza: 7.00
1 Posti liberi

Figure 38: Communities

From the navigation sidebar, the user can access the “Comunità” page and see the list of communities he belongs to. For each community, the app shows the travel area covered by the community, the number of users, the number of cars (drivers) and active notifications.

Each community has a dedicated screen organised in tabs. The first tab presents all the information for the community (e.g., area of travel, number of cars) and allows adding the users car (thus becoming a potential driver that offers rides). The second tab presents the list of users belonging to the community and allows accessing user profiles or starting a chat. The last tab presents the list of rides offered today within the community. From the action bar, the user can access the notifications and request to leave a community.

6.4.3.6. User profile

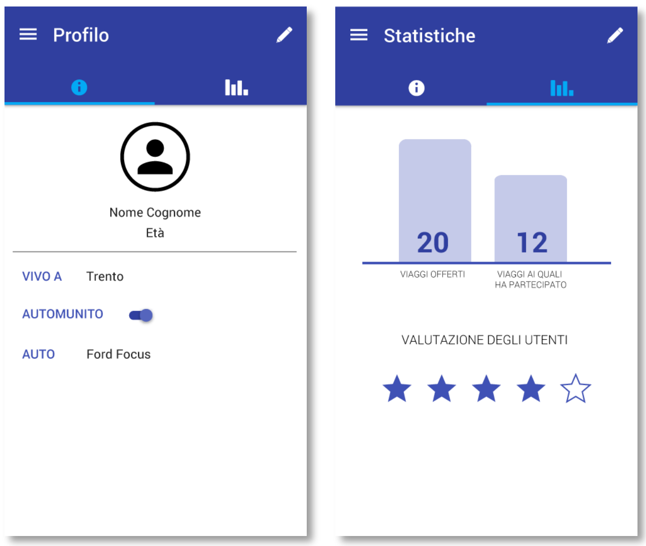


Figure 39: User profile

The “Profilo” screen presents the information of a user profile. The first tab presents basic information: name, age, area of travel, whether he is sharing a car and, in this case, type of car. The second tab shows user statistics (e.g., number of offered rides, number of taken rides) as well as the rating from the other users.

7. Conclusion and next steps

Important technical foundation has been built for the integrated STREETLIFE mobility applications. Although each pilot site has separate end-user applications of variable level of commercialisation and technology readiness level they all share the same goals of incentivisation of greener and more convenient transport and transport planning. All the applications also share the most important technical functionalities, which are essential to any mobility application. The groundwork has been done for multimodal routing solution that incorporates policy function and personal preferences, incentivisation of users for more sustainable transport through gamification, tracking, mode and mode change detection and more research-oriented work on advanced, not yet commercial mixed-reality applications.

The essential technical functionalities will be further improved prior to the next round of actual pilot studies, where STREETLIFE users provide usage data and feedback for further improvements towards the final versions of the end-user applications.

In task T5.1 work is done towards injection of programmable city policies in the route planning and travel assistance component, including some support techniques and tools for city administrators and mobility managers to easily specify and modify those policies, which are then automatically injected. Advanced personalisation and travel assistance features will be further improved including post-personalisation by Gamification and real-time data (like weather), as well as improving the existing travel assistance features.

In Task T5.2 the engine from T5.4 will be integrated to virtual mobility. The problem is annotation of the pictures on the 360-degree images. This is the main challenge, which includes solving the problems with orientation accuracy of the mobile device such as Oculus Rift. The hypothesis that virtual mobility increases confidence (knowledge transfer) of the users to find public transportation will be tested and use it with the help of pre-experiencing the route and modal change locations. The field tests will be properly designed before the actual tests.

In task T5.3 the vision is developed for the comprehensive gamification framework for Smart Cities outlines in Section 4.2. Improvements are anticipated in incorporation of incentivisation in the gamification engine integrated in the applications.

In task T5.4 Berlin 3D model will be further developed by increasing the ground model and air photos as well as some rail tracks, tunnels and some graphical logos. Since there are some quality problems with the model it will later decided whether the current model can be used for mobility related use cases such as spotting bus stops and correct metro entrances with the help of MR. It will also be explored how the quality problems can be tackled and the model further improved. The target is to perform two field tests with focused test groups with the hypotheses that navigation is improved with MR. Field studies of the Mixed Reality application both in Tampere and Berlin in traffic context. There are several hypotheses including how intuitive and helpful Mixed Reality is in locating moving public transport vehicles, bus stops and parking places. In augmented reality a general interest is to know if visual tracking – i.e. spotting computer graphics on the right location on the screen using computer vision – helps users in the mobility context and whether the future commercial graphical user interfaces should include elements from augmented reality. In case of 3D maps the advantages are more straightforward, but the actual graphical presentation of moving objects by real-time information is a research matter that will be further explored and explained in Y3. The field tests are properly designed before the actual tests.

In task T5.5 car-pooling will be pushed as another ICT-backed modality available to the journey planning in ROV. Car-pooling options in the app will be integrated with the ViaggiaRovereto for personal planning. Outcomes of T5.1 into the BER mobility app will be implemented applications are further refined considering the results of the 2nd iteration experiments.

Overall, an important collaboration will be done in the work package to find possible joint WP5 publications such as forming a taxonomy of routing services using the general routing concept or testing of mode detection accuracies in different cities using the developed method.

REFERENCES

[1] Franz, Alexander Mark and Henzinger, Monika H and Brin, Sergey and Milch, Brian Christopher (2006) Voice interface for a search engine. US Patent 7,027,987, year 2016.

[2] Aron, Jacob (2011). How innovative is Apple's new voice assistant, Siri? *New Scientist* (212), p 24.

[3] Wechsung, Ina and Schaffer, Stefan and Schleicher, Robert and Naumann, Anja and Möller, Sebastian (2010). The Influence of Expertise and Efficiency on Modality Selection Strategies and Perceived Mental Effort. *Proceedings of the 11th Annual Conference of the ISCA (Interspeech 2010)*, pp. 1930-1933, 2010.

[4] Schaffer, Stefan and Jöckel, B. and Wechsung, Ina and Schleicher, Robert and Möller, Sebastian (2011). Modality Selection and Perceived Mental Effort in a Mobile Application. *Proc. 12th Ann. Conf. of the Int. Speech Communication Assoc. (Interspeech 2011)*, pp. 2253-2256. Florence, Italy: International Speech Communication Association (ISCA).

[5] Ruß, Aaron (2013). MMIR Framework: Multimodal Mobile Interaction and Rendering. GI-Jahrestagung, pp. 2702-2713.

[6] Binstock, Atman. May 15 2015. "Powering the Rift", https://www.oculus.com/en-us/blog/powering-the-rift/

[7] A. Nurminen, S. Schaffer, A. Marconi, G. Valetto, "D5.2.1 - End-user applications techniques and tools (initial).", STREETLIFE, 9-Dec-2014.

[8] F. Avesani, D. Frigeri, M. Garzoglio, A. Haikola, T. Kärkäs , R. Kelpin, M. Kulmala, A. Merigo, V. Meskanen, T. Pezzato, N. Perri, S. Ruotsalainen, T. Schilling, S. Schaffer, G. Valetto, M. Vuorio, "D6.2.1 - City pilots planning and evaluation results (initial)." STREETLIFE, 14-Apr-2015.

[9] R. Kazhamiakin, A. Marconi, M. Perillo, M. Pistore, G. Valetto, L. Piras, F. Avesani, and N. Perri, "Using Gamification to Incentivize Sustainable Urban Mobility", in Proceedings of the IEEE International Smart Cities Conference, Guadalajara, Mexico, October 2015.

[10] Y. Nagappa, S. Cuno, F. Thiemer, M. Vuorio, G. Valetto, "D2.2.2 – Blueprint Architecture, Security Architecture and Pilot Specific Architectures (intermediate).", STREETLIFE, 30-Sep-2015.

[11] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From Game Design Elements to Gamefulness: Defining ``Gamification". ", in Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '11, pp. 9-15, 2011.

[12] M. Sakamoto, T. Nakajima, and S. Akioka, "A Methodology for Gamifying Smart Cities: Navigating Human Behavior and Attitude", in Distributed, Ambient and Pervasive Interactions, Springer, 2014.

[13] F. Barbon, P. Traverso, M. Pistore, and M. Trainotti, "Run-Time Monitoring of Instances and Classes of Web Service Compositions", in Proceedings of the International Conference on Web Services, ICW^ '06, September 2006.

[14] F. Dalpiaz, P. Giorgini, J. Mylopolous, "Adaptive socio-technical systems: a requirements-based approach", Requirements Engineering, 18(1):1-24, September 2011.

[15] J. Togelius, G.N. Yannakakis, K.O. Stanley, and C. Browne, " Search-Based Procedural Content Generation: A Taxonomy and Survey", IEEE Transactions on Computational Intelligence and AI in Games, 3(3):172-186, September 2011.

[16] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, " Procedural Content Generation for Games: A Survey.", 9(1):1-22, Frebruary 2013.

[17] Azuma, Ronald T (1997). A survey of Augmented Reality. Presence: Teleoperators and Virtual Environments 6(4), 355-385.

[18] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

[19] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer vision and image understanding, 110(3), 346-359.

[20] Reitmayr, G., & Drummond, T. W. (2006, October). Going out: robust model-based tracking for outdoor augmented reality. In Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on (pp. 109-118). IEEE.

[21] Hessaenauer, S., Kral, R. and Adelman, C. Radio Collar Tracker Project. Retrieved Sep 2015 from http://kastner.ucsd.edu/ryan/wp-content/uploads/sites/5/2014/03/admin/radio-collar-tracker-final-report.pdf

[22] Hartwig, S., 2007. *Empty Seats Traveling – Next Generation Ridesharing and Its Potential to Mitigate Traffic and Emission Problems in the 21st Century.* NOKIA – Research Center.

[23] Gonçalo Correia and José Manuel Viegas. *Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a Stated Preference web survey in Lisbon, Portugal*. Transportation Research Part A 45 (2011) 81–90

[24] Concas, S., Winters, P.L., 2007. *The impact of carpooling on trip chaining behavior and emission reductions.* In: Proceedings of the Transportation Research Board 86th Annual Meeting, Washington, DC.

[25] Duecker, K.J., Bair, B.O., Levin, I.P., 1977. *Ride sharing: psychological factors.* Transportation Engineering Journal of ASCE 103 (6), 685–692.

[26] Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, Sven Koenig. *Ridesharing: The state-of-the-art and future directions.* Transportation Research Part B: Methodological, Volume 57, November 2013, Pages 28–46.

[27] *Carpool Incentive Programs: Implementing Commuter Benefits as One of the Nation's Best Workplaces for Commuters*. United States Environmental Protection Agency, Office of Air and Radiation November, 2005

[28] Willson, R.W., Shoup, D.C., 1990. *Parking subsidies and travel choices: assessing the evidence.* Transportation 17, 141–157.

[29] Bianco, M.J., 2000. *Effective transportation demand management combining parking pricing, transit incentives, and transportation management in a Commercial District of Portland, Oregon*. Transportation Research Record 1711, 48–54.

[30] https://www.zimride.com/virginia/

[31] http://www.jojob.it

[32] Dailey, D.J., Lose, D., Meyers, D., 1999. *Seattle smart traveler: dynamic ridematching on the World Wide Web*. Transportation Research Part C: Emerging Technologies 7 (1), 17–32.

[33] Smith, V., Beroldo, S., 2002. *Tracking the duration of new commute modes following service from a ridesharing agency*. Transportation Research Record 1781, 26–31.

[34] Terveen, L., McDonald, D.W., 2005. *Social matching: a framework and research agenda*. ACM Transactions on Computer–Human Interaction 12 (3), 401–434.

[35] Schaffer, S., Reithinger, R. 2014. *Intermodal personalized Travel Assistance and Routing Interface*. In: Butz, A., Koch, M., Schlichter, J. (eds.): Mensch & Computer 2014, 343-346, München, Germany, De Gruyter Oldenbourg, 2014.

8. ANNEX I

8.1. Introduction

8.1.1. Project Purpose & Initial Questions

The purpose of this annex in STRRETLIFE is to figure out what the user of a mobile App need to change the behaviour from using a car to using more sustainable travel modalities. For this reason we like to know which incentives are supporting the change and how can we support this within an App.

8.2. Methodology

For getting a concept and visual design of a mobile App, which supports the user in changing their behaviour, a Design Thinking Workshop was conducted. In this workshop the participants worked out what they need for changing the travel modes as well as ideas for incentives and gamification approaches.

8.2.1. Design Thinking Workshop

The workshop is a user-centred approach for the conception of new products and focuses on people and their needs. The main task for this workshop within the STREETLIFE project was to get to know, which incentives do real users need for exchanging the car into more sustainable travel modalities.

The workshop had five steps:

1. Understand: Identification of relevant stakeholders, incl. users as well as external decision makers but also internal decision makers. In this workshop we focused on the user.

   Outcome: Identify topics for further research (see Chapter 8.3.3)

2. Observe: Collection of qualitative field data through conduction of on-site stakeholder/customer observations and interviews. In the STREETLIFE workshop we collected data from the participants, as they will be the future users of this App.

   Outcome: Stories and data from observations and interviews with potential users (see Chapter 8.3.4)

3. Synthesis: Synthesizing the existing data and information to identify common patterns, possible connections that lie beneath the obvious.

   Outcome: Insights based on findings in the field and opportunity areas for idea/solution exploration (see Chapter 8.3.4 an 8.3.5)

4. Ideate: Generating a broad range of ideas and concepts. Clustering of ideas that fit well together in order to develop them further into more complex and meaningful concepts.

   Outcome: A wide range of ideas and concepts (see Chapter 8.3.4 an 8.3.5)

5. Prototype: First prototypes will be executed in a rough and rapid manner that is simple, quick, cheap and effective.

   Outcome: Visualisation of a number of combined ideas articulated as various concepts or use case scenarios for testing (see Chapter 8.4)

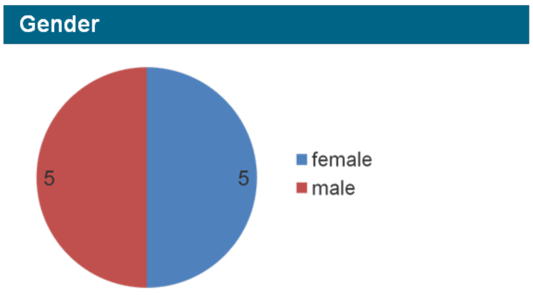
8.2.2. Qualitative data

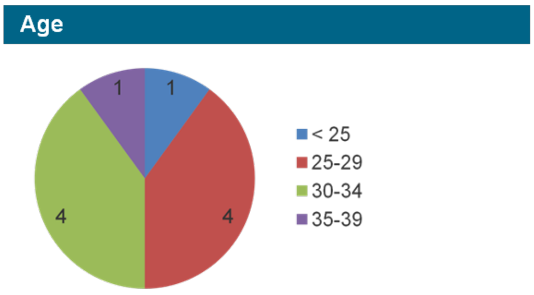
In addition the participants were asked for a self-assessment. They had to evaluate what their current travel situation is and whether they are willing to change this into a more sustainable way of travelling *(see Chapter 8.3.1)*.

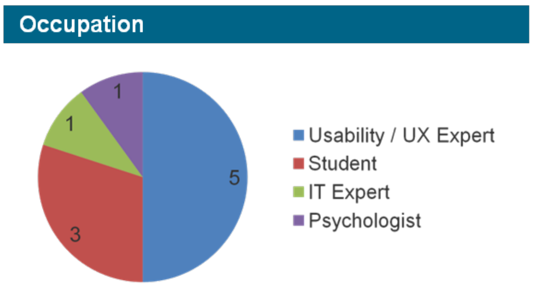
Furthermore we did a ranking questionnaire about gamification types to figure out what kind of App the user need *(see Chapter 8.3.2)*. In general there are 5 types: Achiever, Free Spirit, Philanthropist, Player and Socialiser [2] [3].. According to these types the participants had to rank 5 statements. For detailed description please refer to.

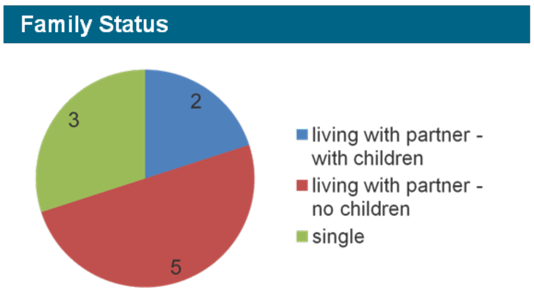
8.2.3. Participants of the Workshop

Socio-demographic data of the participants:

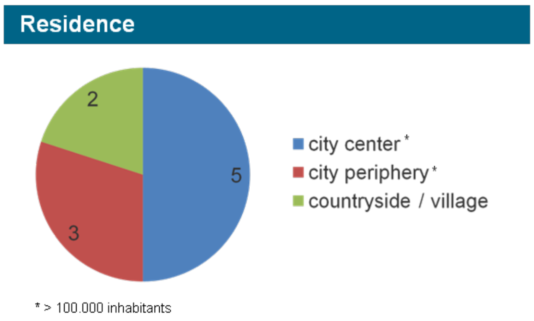


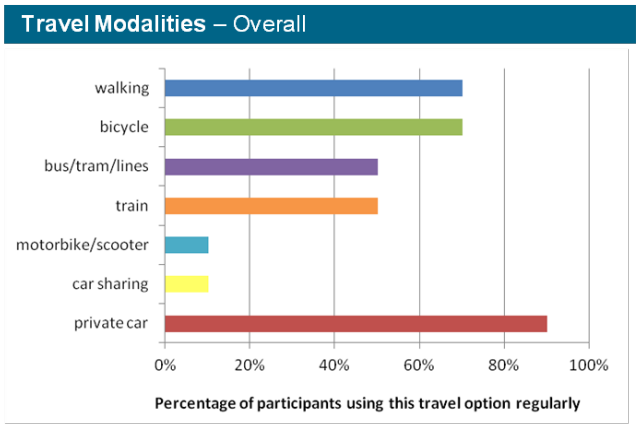


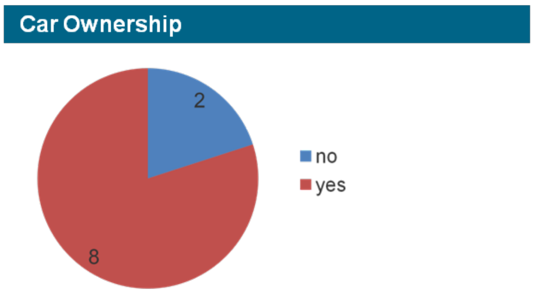


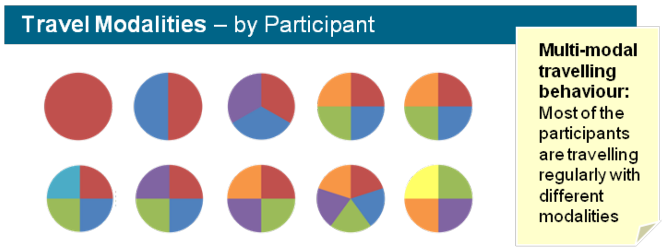


Current travel behaviour of the participants:









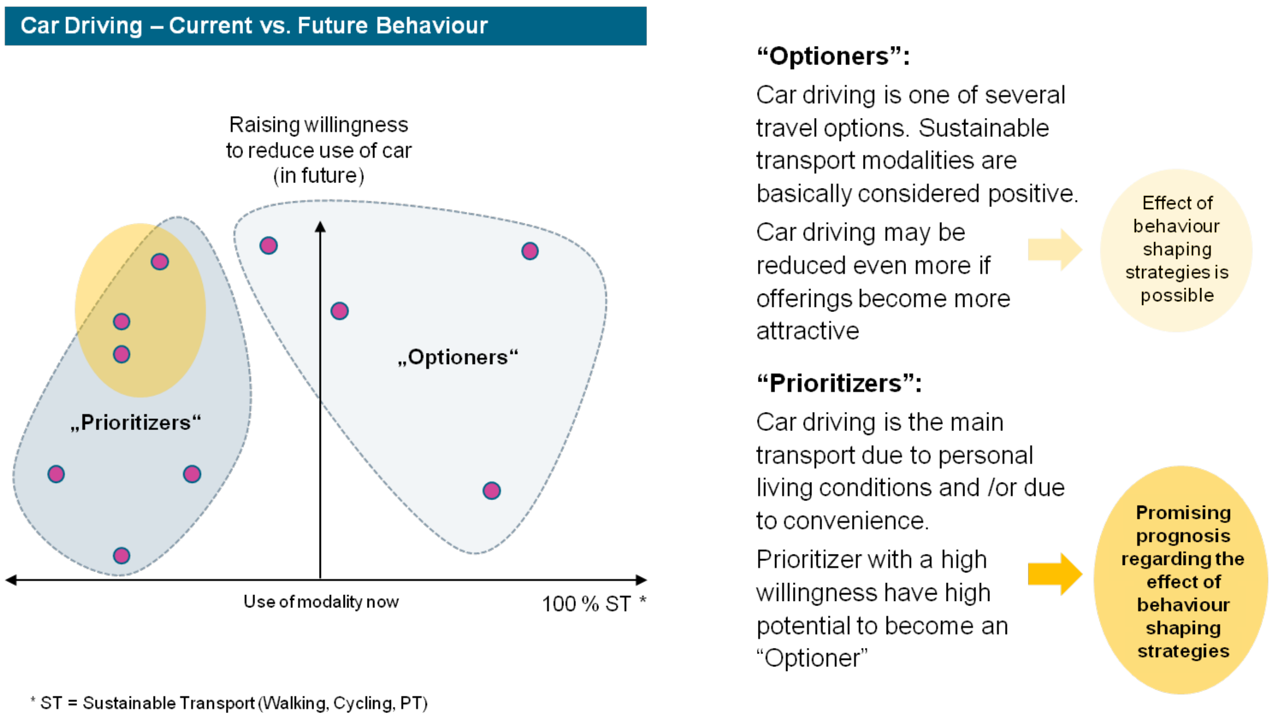
8.3. Results

8.3.1. Potential for Shaping Travel Behaviour [1]

The self-assessment of the **10** participants shows that

- **6** have a high potential to shape the future travel behaviour
- **3** use the car currently quite often, but want to change it
- **3** use the car currently quite often, but don't want to change it

→ The participants of the workshop show a broad distribution of different preconditions.


Car Driving – Current vs. Future Behaviour
Raising willingness to reduce use of car (in future)
„Prioritizers"
„Optioners"
Use of modality now
100 % ST *
* ST = Sustainable Transport (Walking, Cycling, PT)
Effect of behaviour shaping strategies is possible
Promising prognosis regarding the effect of behaviour shaping strategies

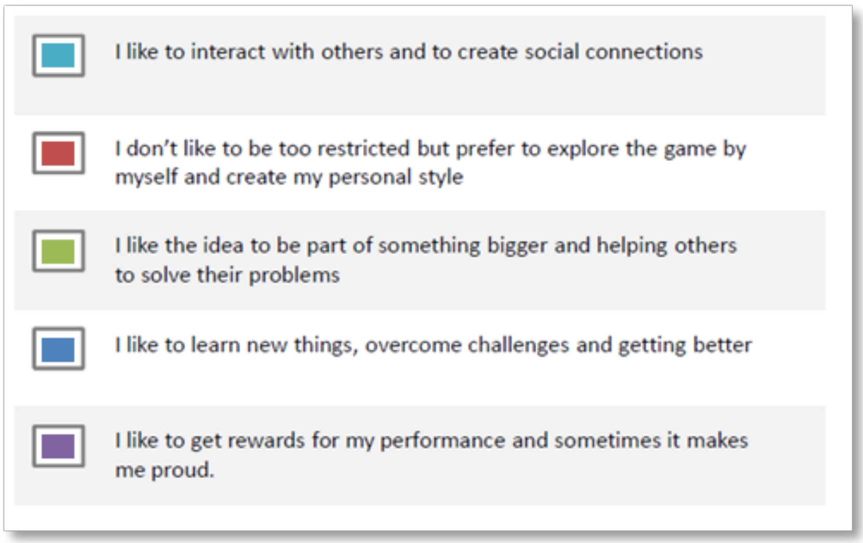
"Optioners":
Car driving is one of several travel options. Sustainable transport modalities are basically considered positive.
Car driving may be reduced even more if offerings become more attractive

"Prioritizers":
Car driving is the main transport due to personal living conditions and /or due to convenience.
Prioritizer with a high willingness have high potential to become an "Optioner"

8.3.2. Gamification Types

The analysis of the ranking questionnaire shows that most of the workshop participants can be assigned to the *Socialiser* and to the *Achiever* *.

As in the incentives discussion, participants emphasize social and competition motives like "being part of groups" and "getting better".



🟦 I like to interact with others and to create social connections

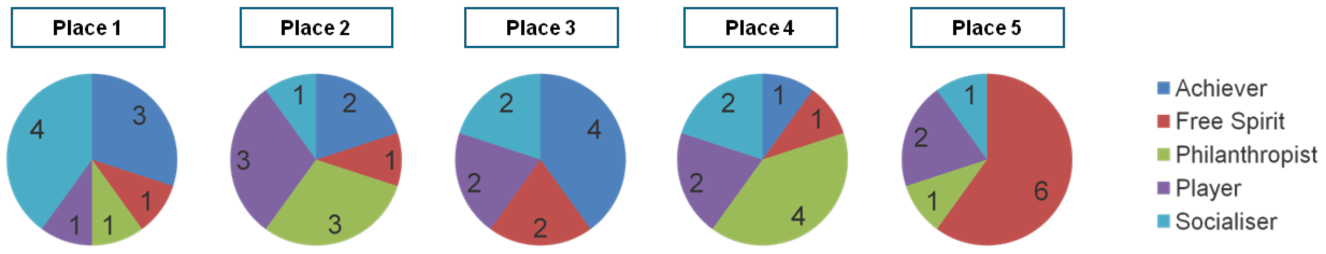
🟥 I don't like to be too restricted but prefer to explore the game by myself and create my personal style

🟩 I like the idea to be part of something bigger and helping others to solve their problems

🟦 I like to learn new things, overcome challenges and getting better

🟪 I like to get rewards for my performance and sometimes it makes me proud.

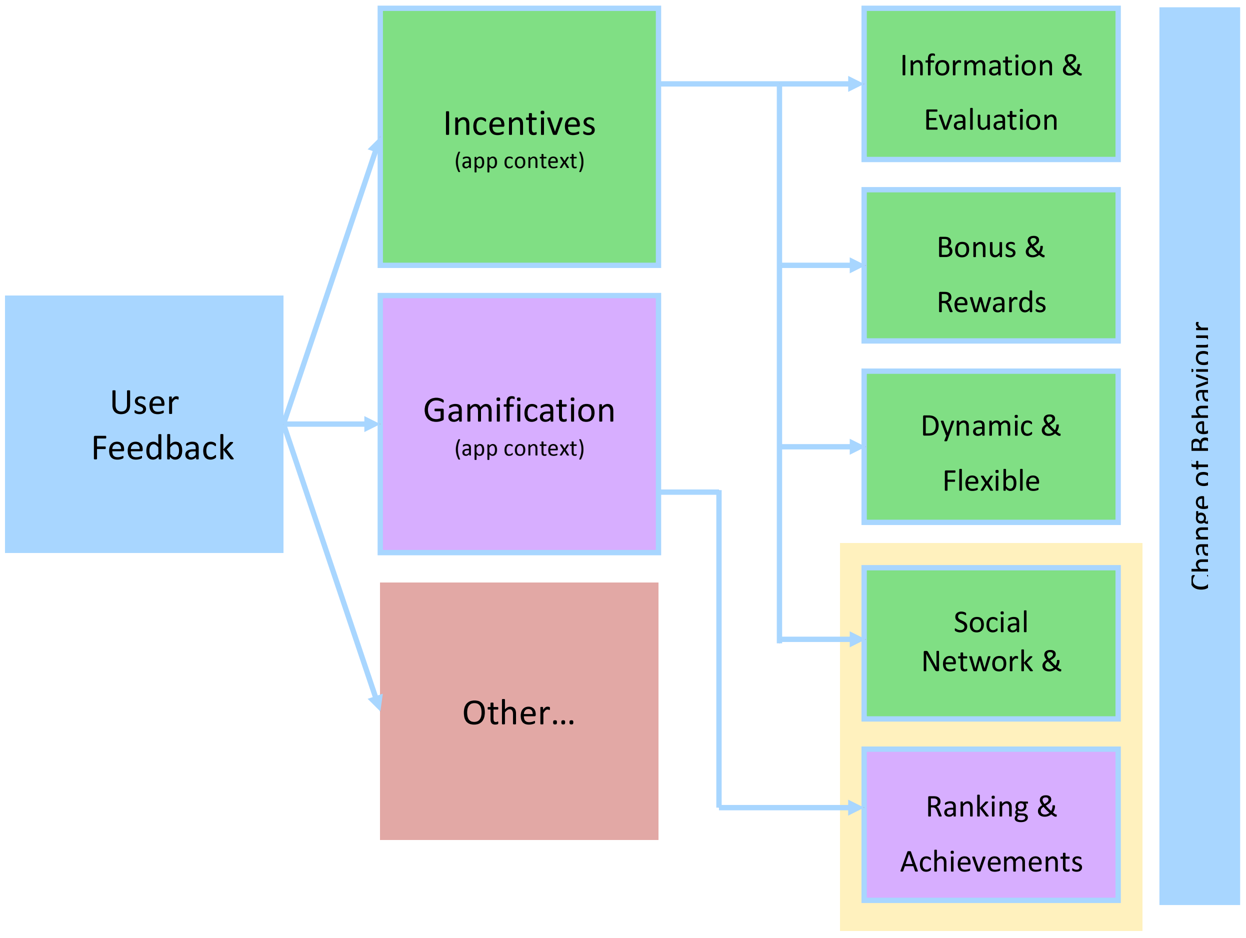
Gamification types statements



* **Regard:** the assignment to a certain gamification type should be interpreted carefully due to a) the simple method (validity of 1 item query?) and b) the fact that most of the persons have rather several game motives with different manifestations than being one type.

8.3.3. Design Thinking Workshop result overview

Within the five steps of the Design Thinking Workshop the participants generated a wide range of ideas for incentivisation and gamification concepts for the mobile App. In the picture below the result of these ideas are shown.



Incentives clustered by categories

8.3.4. Incentives

This chapter is about the incentive and gamification ideas of the workshop and shows which ideas can be implemented into an App.

1. Information and Evaluation

- Easy planning: different route and modality options at a glance with all information available
- Pros and Cons of different route and modality options are shown to enable easy comparison.
- Important criteria for route decision: time, costs, fuel, impact on environment, quality aspects
- Routes can be filtered by personal preference e.g. beautiful landscape, quiet, fast, secure, sightseeing, baggage, number of transport changes in order to have long working/relax times
- Public restrooms and drinking water dispensers are marked at routes
- Clear instructions how to walk from one interim destination to the other
- Nearest start option is shown e.g. next station, bicycle

- Concrete information about impact of car driving on environment/on health (awareness)
- Not too “green” / also information for car drivers to shape behaviour in a gentle way

2. Bonus and Rewards

- Using public transport results in e.g. free tickets, smaller prices
- For using the bicycle results in bicycle shop vouchers
- Collecting points for using sustainable alternatives results in vouchers or donations
- Subsidy/rewards from the government
- Bicycle driver lottery: “Bicycle driver no. 100.000.000 gets a new bicycle”

3. Dynamic and flexible planning

- Alarm is coming up when I use the car too often and reminds me of alternatives
- Information coming up passing points of interest e.g. sights, local news
- Flexible route planning triggered by new situations e.g. delay, weather, congestions
- Notification if the public transport is crowded right now
- Delays are shown

4. Social Network and Collaboration

- Community effect: Activity/goals/evaluation of other users is accessible (statistics, charts)
- Community effect: overall performance of groups is accessible e.g. my neighbourhood
- Network with peer group with similar attitudes regarding health and environment
- Team work: shared routes/instructions/experience
- People find together and join a private car or a car sharing ride
- Individual rides can be requested by people with similar ways
- Competition with peer group regarding performance parameter e.g. km, calories, time CO2
- Seat neighbour can be chosen by profile data e.g. interests
- Presence of friends in public transportation is shown
- Gamification

8.3.5. Gamification

Ranking and Achievements

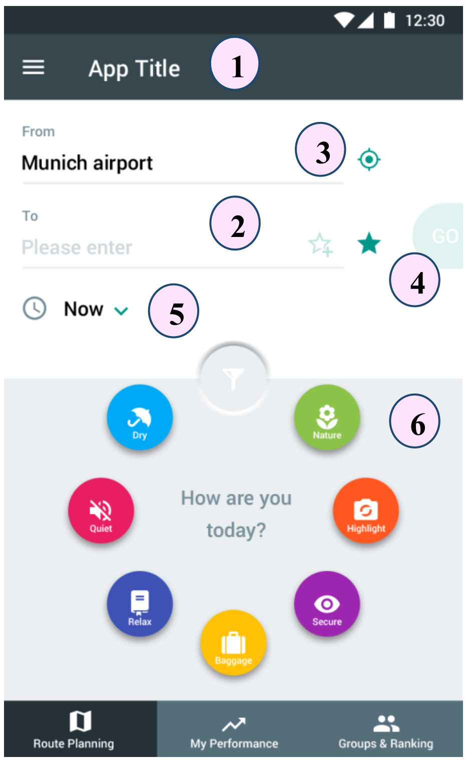
- Overview about personal performance e.g. km, calories, health, pedometer, saved money, CO2 reduction, other “green” aspects
- Competition with peer group regarding performance parameter (see above)

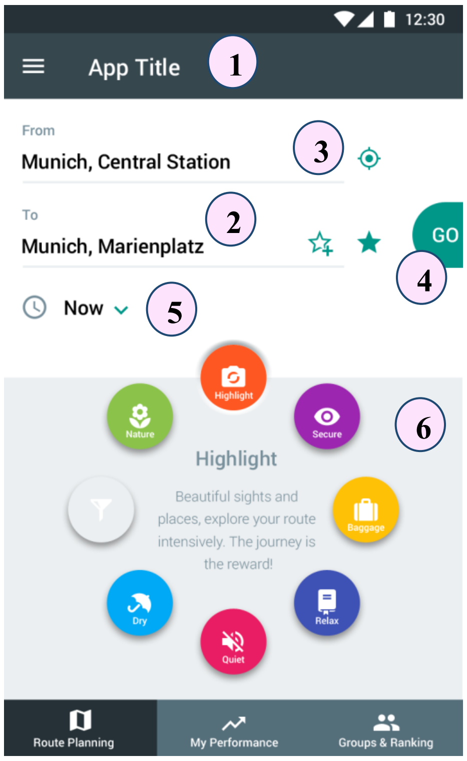
- Virtual points for sustainable behaviour / extra points for cycling or walking when weather is bad
- Game Add-On: Geocaching / surprises from other users along the way
- Game goals trigger ambition and break with routines
- Possibility to play games / compete with other users on the same route

8.4. Screen Design

After working together with potential user the results of the workshop have been consolidated and worked out into a visual design for a mobile App.

Screen 1: Route Planning_DataEntry

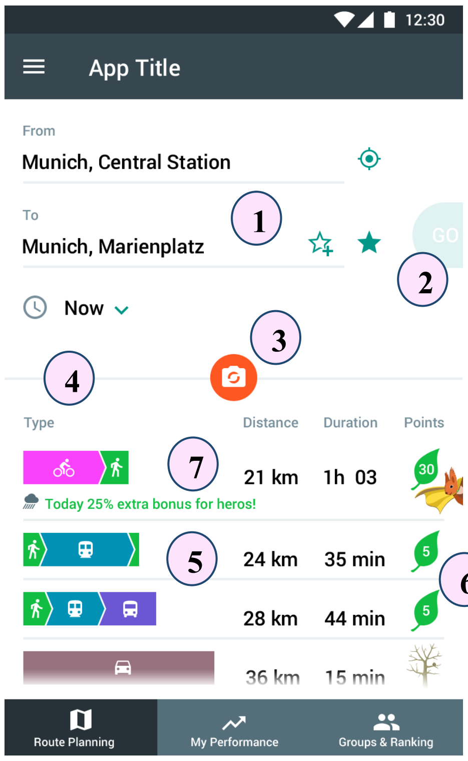
 12:30 App Title 1 From Munich airport 3 To Please enter 2 GO 4 Now 5 6 Dry Nature How are you today? Quiet Highlight Relax Secure Baggage Route Planning My Performance Groups & Ranking	**(1)** *Toolbar:* General functions e.g. *Menu* providing *Personal Profile*, *Settings* etc. are included here.
	(2) *Search:* Start location and destination of the route is entered. Suggestions expand within dropdown by typing (autocomplete).
	(3) *Shortcuts*: Also current location can be taken over by the icon next to the *Start* field. Icons next to the *Destination* field enable user to apply shortcuts / favourites or bookmark new destinations.
	(4) *Go*: Button is disabled until required data are completed, on click search is started. Screen displaying the results comes from right side.
	(5) *Timer*: The time default is *now*. On click on the element, time screen (overlay) comes up where date and time can be changed.
	(6) *Filter:* Suggested route alternatives can be restricted by special needs. The most upper item (in the dent) applies. User can either rotate/swipe the circle or click one of the filter buttons (then, circle rotates until selected item is at the top). Selected state goes together with a short explanation of the filter in the middle. Filter Options: • "*Nature*": route goes through parks, green

 12:30 App Title 1 From Munich, Central Station 3 To Munich, Marienplatz 2 GO 4 Now 5 Highlight Nature Secure 6 Highlight Beautiful sights and places, explore your route intensively. The journey is the reward! Baggage Dry Quiet Relax Route Planning My Performance Groups & Ranking	areas, along the river / creek, fields etc. • “*Highlight*”: route goes next to POI like sights, attractions, events etc. • “*Secure*”: route goes through public spaces, illuminated streets etc. to prevent criminal incidents • “*Baggage*”: larger pieces of baggage can be carried e.g. route by city train instead of crowded tram or bicycle • “*Relax*”: route doesn’t take too many changes of the transport, thus e.g. books can read without breaks • “*Quiet*”: route avoids noisy main roads, crowded places etc. • “*Dry*”: Not to get wet when it’s raining covered transports and walking paths are suggested

Integration of Incentives

Category	**Description**	**App Feature**
Information & Evaluation	Nearest start option is shown e.g. next station, bicycle	(3)
Information & Evaluation	Routes can be filtered by personal preferences e.g. beautiful landscape, quiet etc.	(6)
Information & Evaluation	Information about baggage context e.g. costs, space	(6)
Dynamic & Flexible Planning	Information coming up passing points of interest e.g. sights	(6)
Dynamic & Flexible Planning	Flexible route planning triggered by new situations e.g. weather	(6)

Screen 2: Route Planning_Results



12:30

App Title

From

Munich, Central Station

To

Munich, Marienplatz

GO

Now

Type | Distance | Duration | Points

Type	Distance	Duration	Points
	21 km	1h 03	30
	24 km	35 min	5
	28 km	44 min	5
	36 km	15 min	

Today 25% extra bonus for heros!

Route Planning · My Performance · Groups & Ranking

(1) *Search_filled:* Start and destination of route can be still changed.

(2) *Go*: Button is disabled until other data are entered at search.

(3) *Filter_active*: minimized version of the selected filter criteria is shown. On click lines goes down and the filter circle expands again displaying the normal size buttons.

(4) *Result List*: Route alternatives (max.4) are listed and indicated by key figures. Sorting criteria of the table is 1. *Points* and 2. *Distance*. Therefore the car route is always the last item. Clicking on one of the table lines, screen with respective route details and map comes next.

(5) *Transport Type*: Required travel modes of each route are indicated by images and colours. The order and proportion of the bars meets the real travel flow and distances. The longest bar takes the full column width.

(6) *Points*: That column is the guidance for travelling in a sustainable way.
(+) If user performs a "green" route he collects points for the game and cultivates his tree. (-) Selecting the car route results in harming the tree.
[Reward system, see also screen 3]

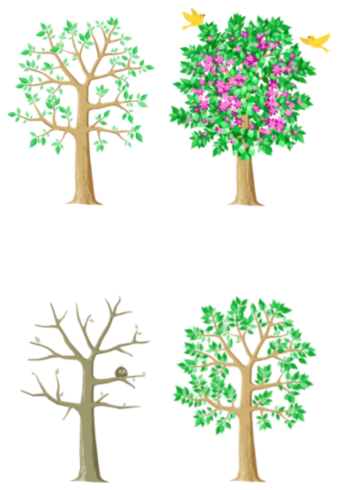
(7) *Notification:* If weather is bad (cold, raining…) the active route alternatives including walking or cycling are promoted by extra points and the *Bad-Weather-Hero*.
[Reward system, see also screen 3]

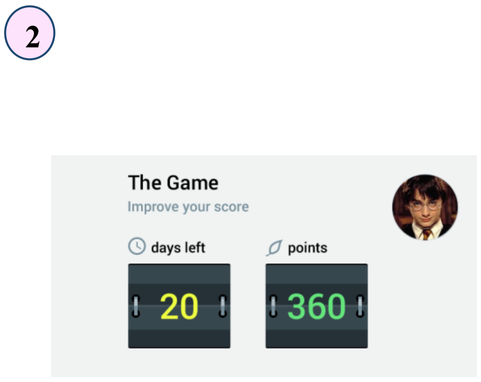
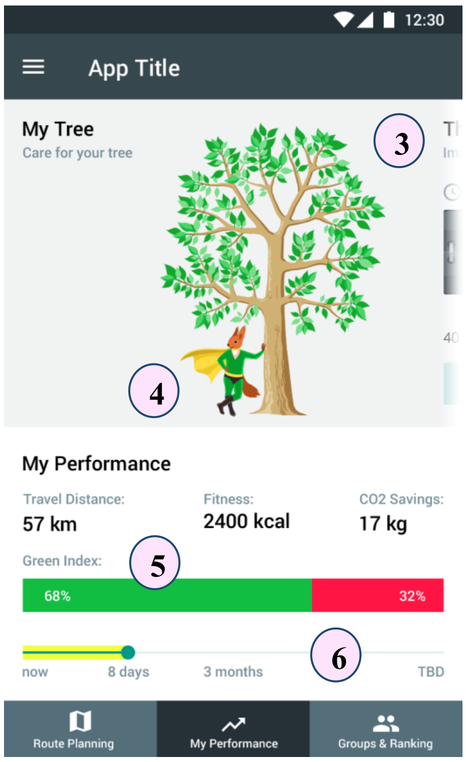
Integration of Incentives

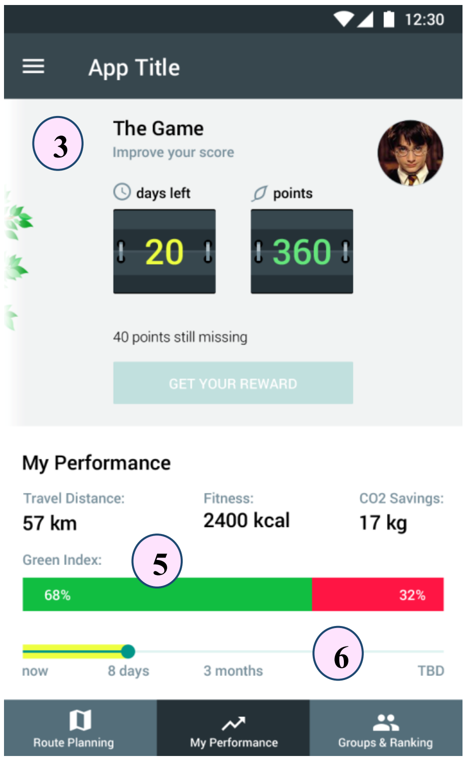
Category	**Description**	**App Feature**

Information & Evaluation	Easy planning: different route options at a glance with all information available	(4) (5)
Information & Evaluation	Pros and Cons of different route options are shown to enable easy comparison	(4) (5) (6)
Information & Evaluation	Important criteria for route decision: time, switch of transports, impact on environment etc.	(4) (5)
Information & Evaluation	No "green" behaviour by force, also information for car drivers available to shape behaviour in a gentle way [to avoid defiance]	(4)
Ranking & Achievements	Virtual points for sustainable behaviour / extra points for cycling or walking when weather is bad	(6) (7)

Screen 3: Personal Performance

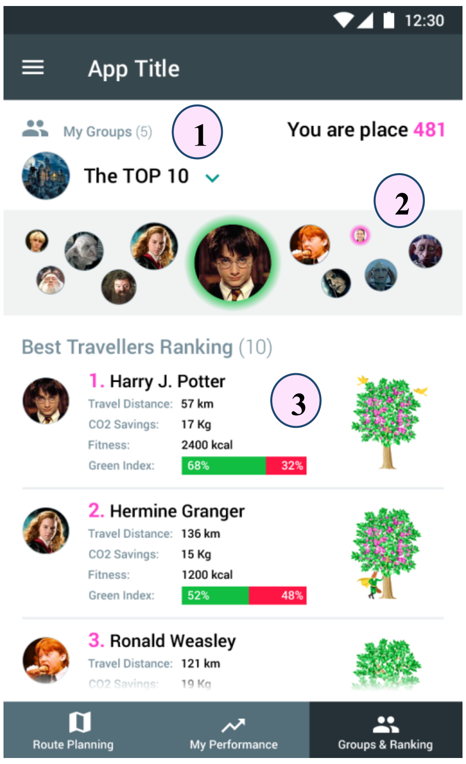
1 	**(1)** *Virtual Reward:* Every period (= 1month) user gets a new *Tree* to take care of. At the beginning the tree is young and has just some little leaves. (+) If user performs "green" (sustainable) route alternatives the tree becomes more splendid. (-) If user performs car routes, the tree looses leaves and starts to weather. At the end of every period the tree is implanted at user's garden (visible on the welcome screen, when app is started). <u>Incentive:</u> positive + negative reinforcement apply

 2 The Game Improve your score days left points 20 360	**(2)** *Real Reward:* Every period (= 1month) user has the chance to win a *Game* by collecting points. (+) If user performs "green" (sustainable) route alternatives he earns points. (-) If user performs car routes, nothing happens. If 400 points are achieved, user gets e.g. free tickets for the public transport. <u>Incentive:</u> restricted to positive reinforcement
 12:30 App Title My Tree Care for your tree 3 4 My Performance Travel Distance: 57 km Fitness: 2400 kcal CO2 Savings: 17 kg Green Index: 5 68% 32% 6 now 8 days 3 months TBD Route Planning My Performance Groups & Ranking	**(3)** *Tree & Game:* By swipe either *Tree* or *Game* can be watched (exclusive display). Accessing the tab *My Performance* or switching the views the animation starts. *Tree:* Leaves / flowers etc. start to grow or come off depending on the meanwhile route performances *Game:* Data start to refresh (like a running meter). If *400 points* are achieved, the button *Get your reward* becomes active.
	(4) *Bad-Weather-Hero:* If weather is bad (cold, raining…) user gets an award for performing an active route including walking or cycling. The *Squirrel* is shown with the tree for 3 days.
	(5) *Performance:* Personal performance parameters are shown. The *Green Index* indicates the proportion of "green" (sustainable) and "red" trips. If more than 50% of the index is red, a notification / reminder comes up e.g. "Are you sure…", when user selects a car route (at screen 2)
	(6) *Timer:* By default the performance data at *Your Performance* are shown for the current period. By moving the slider below other time frames can be displayed.

 12:30 App Title 3 The Game Improve your score days left points 20 360 40 points still missing GET YOUR REWARD My Performance Travel Distance: 57 km Fitness: 2400 kcal CO2 Savings: 17 kg Green Index: 5 68% 32% 6 now 8 days 3 months TBD Route Planning My Performance Groups & Ranking		
Integration of Incentives		
Category	**Description**	**App Feature**
Dynamic & Flexible Planning	Alarm is coming up when I use to often the car and reminds me of alternatives (cf. fitness apps)	(5)
Bonus & Rewards	Bonus points for using public transport results in e.g. free tickets, smaller prices e.g. shop vouchers e.g. helmet, light	(2) (3)
Bonus & Rewards	Collecting points for using sustainable alternatives results in vouchers	(2) (3)
Ranking & Achievements	Overview about personal performance e.g. km, calories, health, pedometer, saved money, CO2 reduction, other "green" aspects	(5) (6)
Ranking & Achievements	Virtual points for sustainable behaviour / extra points for cycling or walking when	(1) (2) (3) (4)

	weather is bad	
Ranking & Achievements	Game goals trigger ambition and break with routines	(1) (2) (3)

Screen 4: Groups & Ranking



My Groups (5)

You are place 481

The TOP 10

Best Travellers Ranking (10)

1. Harry J. Potter
Travel Distance: 57 km
CO2 Savings: 17 Kg
Fitness: 2400 kcal
Green Index: 68% 32%

2. Hermine Granger
Travel Distance: 136 km
CO2 Savings: 15 Kg
Fitness: 1200 kcal
Green Index: 52% 48%

3. Ronald Weasley
Travel Distance: 121 km
CO2 Savings: 19 Kg

(1) *My Groups:* Number indicates the count of social groups user has joined. Every user is a member of *The TOP 10* group (= all app users) at least. The ranking of that group is shown by default. By click on the current group title or image a screen with group selection comes up.

(2) *Ranking_Crowed*: That view is restricted to *max.10 (+1*) members* no matter of the group size. The ranking is visualized by the crowed of different sized profile images (place 1 is the biggest image; cf. tag cloud). Click on the images results in a jump to the respective entry at the ranking list.

* The user itself is always included even he is not one of the placements 1-10. His placement number is visible at the top.

(3) *Ranking_List*: That view shows a*ll members*, the group size is indicated at the headline. The ranking is visualized by a table including the personal performance data of the current time period and the profile images. With the *Tree* also *Bad-Weather-Heroes* a visible for others.

Integration of Incentives

Category	**Description**	**App Feature**
Social Network & Collaboration	Community Effect: Activity / evaluation of other users is accessible (statistics, charts)	(3)
Social Network & Collaboration	Network with peer group with similar attitudes regarding health & environment	(1)
Ranking & Achievements	Competition with peer group regarding performance parameter e.g. km, calories, time, CO2	(2) (3)

Implementation of more incentives by additional app features might be considered e.g. communication with other users, presence check of other users, display of other users at route

8.5. Conclusion

The workshop were conducted with participants from Germany but the results achieved can be also adapted to other countries in Europe as it is a generic approach of user needs e.g. vouchers, free tickets.

The incentives in general can be used for all of the three pilots (Berlin, Rovereto and Tampere). For a more culture specific incentivisation it is helpful to do the Design Thinking workshop with users of each country.

The visual Design of 4 screens is an example how the incentive and gamification approach could be realised. Further work has to be done for concepting a full version of the App.

References

[1] Marczewski, A. (2014) *Gamification User Types 2.0:* http://www.gamasutra.com/blogs/AndrzejMarczewski/20131129/205957/Marczewskis_Gamification_User_Types_20.php (visited: 16.07.2015).

[2] Marczewski, A. (2014) *Gamification User Types 2.0:* http://www.gamasutra.com/blogs/AndrzejMarczewski/20131129/205957/Marczewskis_Gamification_User_Types_20.php (visited: 16.07.2015).

[3] Millonig, A. and Mitgutsch, K (2014): *Playful Mobility Choices: Motivating informed mobility decision making by applying game mechanics*. In: EAI Endorsed Transactions on Ambient Systems, Volume 1, Issue 4, e3., 03-10 (2014).