WP8 – Use Case II:
Enhancing the Driving Experience

D8.2: Use Case Implementation Report (Use Case II)

Deliverable Lead: WORLD

Contributing Partners: ASC, CRF, TALK, WORLD

Delivery Date: 11/2015

Dissemination Level: Public

Version Revision v1.0

This deliverable reports the work carried out within task T8.2 for the implementation of the Use Case topics defined in WP8 that have been used for the real-world validation process within SIMPLI-CITY.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 318201

Document Status	
Deliverable Lead	Toni Paradell, WORLD
Internal Reviewer 1	Philipp Hoenisch, TUV
Internal Reviewer 2	Michaela Kargi, FGM
Type	Deliverable
Work Package	WP8: Use Case II: Enhancing the Driving Experience
ID	D8.2: Use Case Implementation Report (Use Case II)
Due Date	31.07.2015
Delivery Date	13.11.2015
Status	For Approval

Document History	
Contributions	V0.1, WORLD, Proposal of Table of Contents, 29.06.2015 V0.2, WORLD makes a proposal on contents, 21.07.2015 V0.3, CRF provides first material for the document, 23.07.2015 V0.4, TUV provides some additional comments, 04.08.2015 V0.5, CRF provides new contents based on Use Case II.1, which also integrates ASC contribution, 04.08.2015 V0.6, CRF provides complete description for Use Case II.1, 07.08.2015 V0.7, WORLD prepares the first version of the Use Case II.2 contribution, 18.09.2015 V0.8, WORLD adds additional contents to Use Case II.2, 16.10.2015 V0.9, WORLD provides complete description for Use Case II.2, 04.11.2015 V0.10, ASC, text updates and comment resolving for the EcoContest V0.11, WORLD provides a new version after FGM review, 13.11.2015 V1.0, WORLD reviews format and final version ready for approval, 13.11.2015
Final Version	November 13th, 2015

Note

This deliverable is subject to final acceptance by the European Commission.

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

Project Partners

Vienna University of Technology (Coordinator), Austria

Ascora GmbH, Germany

TIE Nederland B.V., The Netherlands

Technische Universität Darmstadt, Germany

IBM Research – Ireland

Smarter Cities Technology Centre

Forschungsgesellschaft Mobilität, Austria

Talkamatic AB, Sweden

Atos Worldline, Spain

Centro Ricerche FIAT, Italy

SRM – Reti e Mobilità, Italy

Executive Summary

The SIMPLI-CITY project has included two Use Case scenarios, which aim at testing the theoretical structure and software prototypes in a real-world environment. Indeed the work package WP8, analogously to WP7, is aimed to demonstrate the effectiveness of SIMPLI-CITY to provide reliable mobility-related services and apps to be delivered to road users by means of the Personal Mobility Assistant.

In WP8, the deliverable D8.1.2 established precise and in-depth definition of the Use Case scenarios. The deliverable report at hand presents the final implementation of these scenarios through apps developed as part of T8.2.

The Use Case II is further divided into two topics: “Environmental Awareness Rising” (Use Case topic II.1, defined by CRF and located in Turin) and “Rising the Diver’s Comfort” (Use Case topic II.2, defined by WORLD and located in Dublin). These two Use Case topics are strictly connected to each other; they indeed cover two different aspects that the driver is interested in: The sustainability and reduction of consumption from one side and the comfort and peace of mind from the other.

Table of Contents

1 Introduction

SIMPLI-CITY – The Road User Information System of the Future – is a project funded by the Seventh Framework Programme of the European Commission under Grant Agreement No. 318201. It provides the technological foundation for bringing the “App Revolution” to road users by facilitating data integration, service development, and end user interaction.

Within this document, an in-depth description of the implementation of Use Case defined in WP8 (Use Case II) is provided.

1.1 SIMPLI-CITY Project Overview

Analogously to the “App Revolution”, SIMPLI-CITY adds a “software layer” to the hardware-driven “product” mobility. SIMPLI-CITY takes advantage of the great success of mobile apps that are currently being provided for systems such as Android, iOS, or Windows Phone. These apps have created new opportunities and even business models by making it possible for developers to produce new apps on top of the mobile device infrastructure. Many of the most advanced and innovative apps have been developed by players formerly not involved in the mobile software market. Hence, SIMPLI-CITY supports third party developers to efficiently realise and sell their mobility-related service and App ideas by a range of methods and tools, including the Mobility Services and App Marketplaces.

In order to foster the wide usage of those services, a holistic framework is needed which structures and bundles potential services that could deliver data from various sources to road user information systems. SIMPLI-CITY provides such a framework by facilitating the following main project results:

- Mobility Services Framework: A next-generation European Wide Service Platform (EWSP) allowing the creation of mobility-related services as well as the creation of corresponding apps. This enables third party providers to produce a wide range of interoperable, value-added services, and apps for drivers and other road users.
- Mobility-related Data as a Service: The integration of various, heterogeneous data sources like sensors, cooperative systems, telematics, open data repositories, people-centric sensing, and media data streams, which can be modelled, accessed, and integrated in a unified way.
- Personal Mobility Assistant: An end user assistant that allows road users to make use of the information provided by apps and to interact with them in a non-distracting way – based on a speech recognition approach. New apps can be integrated into the Personal Mobility Assistant in order to extend its functionalities for individual needs.

To achieve its goals, SIMPLI-CITY conducted original research and applied technologies from the fields of Ubiquitous Computing, Big Data, Media Streaming, the Semantic Web, the Internet of Things, the Internet of Services, and Human-Computer Interaction. For more information, please refer to the project website at http://www.simpli-city.eu.

1.2 Document Purpose, Scope and Context

The overall objective of WP8 is to demonstrate (together with WP7) the feasibility of SIMPLI-CITY’s approach to build services and innovative mobility-related end user

applications by integrating different data sources and to present them to the end user by means of the Personal Mobility Assistant. In the context of WP8, the purpose of this document is to show how the solutions defined for SIMPLI-CITY have been converted into real apps that are applicable in a real-world environment.

Therefore, this deliverable D8.2 is focused on the application of the Use Case Specification (Use Case II) defined in D8.1.2, and the applications that were developed to demonstrate the Use Case.

1.3 Document Status and Target Audience

This document is listed as “public”, as it contains general, supplementary information helpful for the understanding of the SIMPLI-CITY deliverables. Therefore, it is useful for all readers of SIMPLI-CITY project deliverables.

1.4 Abbreviations and Glossary

A definition of common terms and roles related to the realization of SIMPLI-CITY as well as a list of abbreviations is available in the supplementary document “Supplement: Abbreviations and Glossary”, which is provided in addition to this deliverable.

Further information can be found at http://www.simpli-city.eu.

1.5 Document Structure

This document is broken down into the following sections:

- Section 1 provides an introduction for this deliverable, including a general overview of the project, and outlines the purpose, scope, context, status, and target audience of this deliverable.
- Section 2 provides a description of the frame of this deliverable, taking into account the relation with the SIMPLI-CITY platform.
- Section 3 describes the implementation of Use Case Topic II.1: “Environmental Awareness Rising”.
- Section 4 describes the implementation of Use Case Topic II.2: “Rising the Driver’s Comfort”.
- Section 5, finally, contains the conclusions.

2 Use Case II: Enhancing the Driving Experience

2.1 Introduction

The present deliverable describes the two applications that have been implemented during T8.2 to fulfil the Use Case scenarios defined in T8.1, that together with those already provided within WP7 allowing a more extensive validation of the SIMPLI-CITY platform and components delivered by WP4-6.

In particular, this deliverable provides implementation of the Use Case II “Enhancing the driving experience” with regard to the two defined topics: ‘Environmental Awareness Rising” and “Rising the Driver’s Comfort” (see Section 3 and 4).

The two Use Case topics indeed cover two different aspects that the driver is interested in: The sustainability and reduction of fuel consumption from one side and the comfort and peace of mind from the other.

They demonstrate that SIMPLI-CITY is able to offer a set of services and applications that cover the whole trip from the preparation to the arrival to the destination, enhancing the driving experience.

The present deliverable D8.2 shows how these two applications follow the two topics in the Use Case functional definition and details the architecture of both applications in the framework of SIMPLI-CITY environment. In addition, both applications are described in detail.

2.2 Relation with the SIMPLI-CITY Platform

The specified apps in this deliverable have been developed and implemented in T8.2. They build up onto the SIMPLI-CITY platform developed in the RTD WPs 4-6. The apps are structured into a backend service component and a frontend app for the SIMPLI-CITY PMA.

2.2.1 Usage of the SIMPLI-CITY Platform by the Use-Case Apps

Both apps developed for this Use Case (topic II.1 and II.2) are using the SIMPLI-CITY platform as the base framework. Thus, we demonstrate how the SIMPLI-CITY platform help to create fully functional applications simplifying the use of services and car sensors by hiding implementation of these components and offering a common interface for app development.

2.2.1.1 Service Component

The service component of each topic app makes use of services hosted by the Service Runtime Environment as a runtime container. It enables the Use-Case functionality to be reusable in different SIMPLI-CITY apps, due to the module separation. In this context, the implemented service component also makes use of the already developed service as data sources, e.g., it accesses end user and device data, derived from different other sources without taking care “where” they come from. Crucial data for both topic apps are the user GPS position for rerouting, data from the used car to provide a calculated Eco-score and a set of data to aggregate a personal trip history.

2.2.1.2 App Component

Use Case app components act as a middleware between the Multimodal Dialog interface and the respective backend services. For example, a request of the end user to take part in an Eco-Race is forwarded to the app component that hands it over to the responsible service parts. Also the other way round, the service component of the app can notify the app component about, e.g., a change in the Eco-Race high score, and provides some hints about how the personal Eco-score can be improved to achieve a better result.

2.2.2 Advantages of Using the SIMPLI-CITY Platform

SIMPLI-CITY has developed and implemented many sensors, such as GPS or those collected from OBU (see Section 3.2), and data sources (see Sections 3.3.1 and 4.3.1). Building on top of these artefacts, the Use-Case apps access a wide basis of data related to the end user, the car and the respective environment. This enables apps to aggregate or process these data to produce completely new information (see Sections 3.3.3 and 4.3.3).

The apps for WP8 show just a small subset of the possibilities, and serve as prove of concept how to bring the RTD outcome to the “real world”.

2.3 Required Equipment

Figure 1: Use Case II Logical Architecture

The Environmental Awareness Rising Use Case is designed for offering to the driver a set of services aimed to improve her/his driving behaviour, suggesting the better manoeuvres for driving in eco way, saving fuel consumption, etc.

In Figure 1 the Use Case II Logical Architecture is illustrated.

The vehicle that has been used for the provision, the demonstration and the validation of the Use Case is a FIAT 500 L Trekking equipped with an on board unit with the following features:

- Display 5.0” Color Touchscreen WQVGA (Wide Quarter Video Graphics Array with a resolution of 400 x 240)
- AM/FM (Amplitude Modulation, Frequency Modulation)
- Dual tuner (only for Europe market)
- Aux/USB/iPod Integration

- Integrated CD/MP3
- Steering wheel controls
- Back up camera input
- Bluetooth Connectivity
- Hands Free Phone
- Streaming Audio
- SMS Reader
- GPS, Compass

In order to complete the test bed, two further devices are required for running the PMA:

- A smartphone running Android 4.4 is required for the apps installation and the connectivity with the backend services.
- A server/virtual machine, running Linux (Ubuntu 12.04) with Python 2.7.2 and pip installed, is needed for the MMDI and SRE (including backend services) that are used internally during the implementation of the Use Case.

3 Use Case Topic II.1: “Environmental Awareness Rising”

3.1 Use Case Overview

The main objective of this Use Case topic is to offer a set of services able to improve the eco-driving during the normal use of the vehicle extending the as-is application already available in the last vehicles models. The application eco:Drive, is a software system that calculates and visualizes on the On Board Unit display an instantaneous index that evaluates latest driving manoeuvres, and the mission index that averages the whole mission from the key-on.

In more detail, the following functionalities have been implemented within this scenario:

- Current trip information supply: a set of information about next planned trip that includes complete address for destination, expected arrival time, and distance from destination.
- More ecological alternative route suggestion: An alternative route that can be travelled obtaining a higher eco-score, where the eco-score is a way to measure the driving style from an environmentally point of view.
- Eco-driving monitoring: Continuous calculation of achieved eco score on the basis of several driving parameters (speed, gear, acceleration, and deceleration) and suggestion for increasing it. Comparison with score previously achieved when travelling the same route in order to stimulate a continuous self-improvement.
- Offline vehicle information supply: Access to historical information about travelled trips and driving style.
- Eco-driving Contest management: Drivers can compete with each other to achieve the best eco score in the “ecodrivers” community.

While the first four functionalities have been integrated in a single application called EcoAssistant, the eco-driving contest management has been developed by partner Ascora as separated application.

According to the implementation plan presented in D8.1.2 the EcoAssistant application has been implemented in two stages:

- The initial prototype demonstrated during the review held in November at Brussels covered the first functionality
- The final prototype enriches the initial trip assistant offering a full set of eco services that includes the suggestion of an alternative route, the eco race challenge and the possibilities to browse trip history.

Moreover, in order to properly engage Italian native speaking end users, an Italian version of the EcoAssistant application has been developed.

In order to motivate people to drive more eco-friendly, the project has also implemented an EcoContest. This contest allows drivers to compare themselves with others. For this purpose, the EcoContest measures the eco-friendliness of the current trip and creates a highscore-like view displaying the own rating in competition to those of others.

3.2 Use Case Architecture

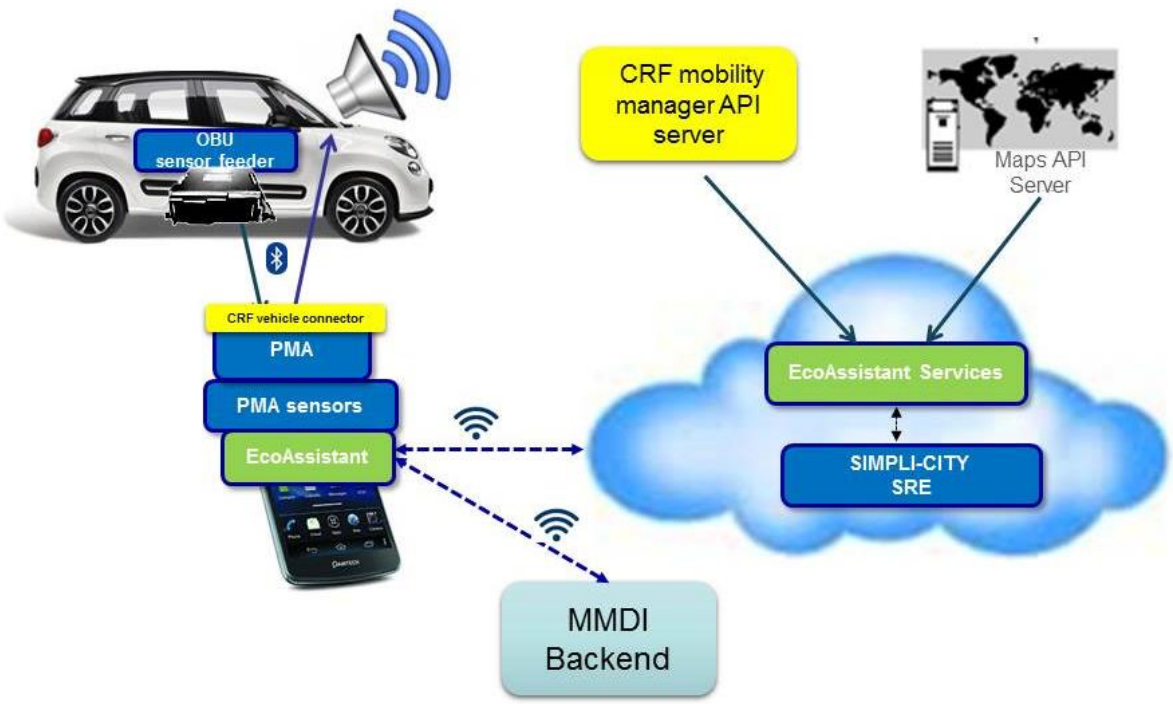

Figure 2: Use Case II.1 Logical Architecture

The Environmental Awareness Rising Use Case is designed for offering the drivers a set of services aimed to improve her/his driving behaviour, suggesting the better manoeuvres for driving in an eco-friendly way, saving fuel consumption and money..

As illustrated in Figure 2, the Use Case is designed to be integrated directly in the vehicle.

The used vehicle is a FIAT 500 L Trekking which is equipped with a prototype version of an on board unit called UCONNECT.

A component (OBU sensor feeder in the picture) aimed to sends car signals collected from the CAN bus in broadcast to the CRF Car Sensor that is part of the PMA sensors is installed on the OBU.

The OBU sensor feeder and the CRF Car sensor have been developed by CRF as part of the Car Sensor that in turn is included in PMA Sensors; a detailed description is provided in D.4.3.2.

The smartphone, a Nexus 5 with Android 4.4.42, is connected to the vehicle via Bluetooth for receiving car data thought the OBU and for reproducing audio outputs of the App thought car loudspeaker.

On the smartphone the PMA with SIMPLI-CITY Apps (that include the EcoAssistant App) and the PMA sensors are installed.

The smartphone is connected via Wi-Fi/3G to the backend services deployed on the SIMPLI-CITY SRE and to the MMDI backend for processing the multimodal interaction.

The EcoAssistant backend service uses a third-party service for its navigation functionality. In addition, the backend service can retrieve information stored in the trip history database

using the services implemented by CRF (CRF mobility manager API). The trip history database and the services for accessing historical data are pre-existent to SIMPLI-CITY.Similarly to this, the EcoContest is communicating to backend services for transferring the current eco driving behavior of a driver to the server side. The services on the server then compare the values collected during a trip to those of other SIMPLI-CITY users and returns a comparison list. Finally, the EcoContest displays the rating of the own car to the users and also shows the rating of other drivers from the same car model.

3.3 Use Case Implementation Review

3.3.1 Data Sources

According with the implementation plan the data needed for the EcoLiveAssistant application have been provided and exposed according with the SIMPLI-CITY format. In particular, as described in D8.1.2 the two data sources which are needed are: (i) Car signals data coming from the FIAT show car and (ii) data user calendar.

The EcoContest uses data from the car itself querying the values from the underlying OBD II adapter. This includes the current speed and the distance of the trip. The App combines this information with information from the car model using a static databased licensed by partner Ascora. Finally, local sensors from the smartphone are used to consider the GPS position and the accelerometer.

3.3.2 Backend Services

The TripAssistant backend service implemented in the first stage exposed initially only one method *doTripCalculation* that takes as input parameters:

- The current user position, i.e., latitude and longitude of the vehicle collected through the PMA sensors
- the destination address coming from the user agenda

and gives as output several information about the next trip:

- including a map of possible routes to reach the destination
- the estimated arrival time
- and the route's length

This service has been extended in the second stage with two further methods to cover the other functionalities requested.

In particular *getMapForDestination* is the methods that takes as input

- the coordinates of destination

and provides as outputs

- all information needed to support the Eco Race and the Alternative Route;

the last method *getListOfRouteNames* takes as input

- the vehicle id

and returns as outputs

- the list of routes and related information for a given vehicle stored in the trip history database.

The EcoContest App uses services to match the own driving behavior to those of other drivers. Though this process does not seem complex, it should be considered that the backend services needs to perform the comparison by considering several context information including the car model but also the configuration of the car model such as the motor data, the fuel type and others.

3.3.3 EcoAssistant App

The Figure 3 shows the navigation flow of the EcoAssistant App as it has been implemented.

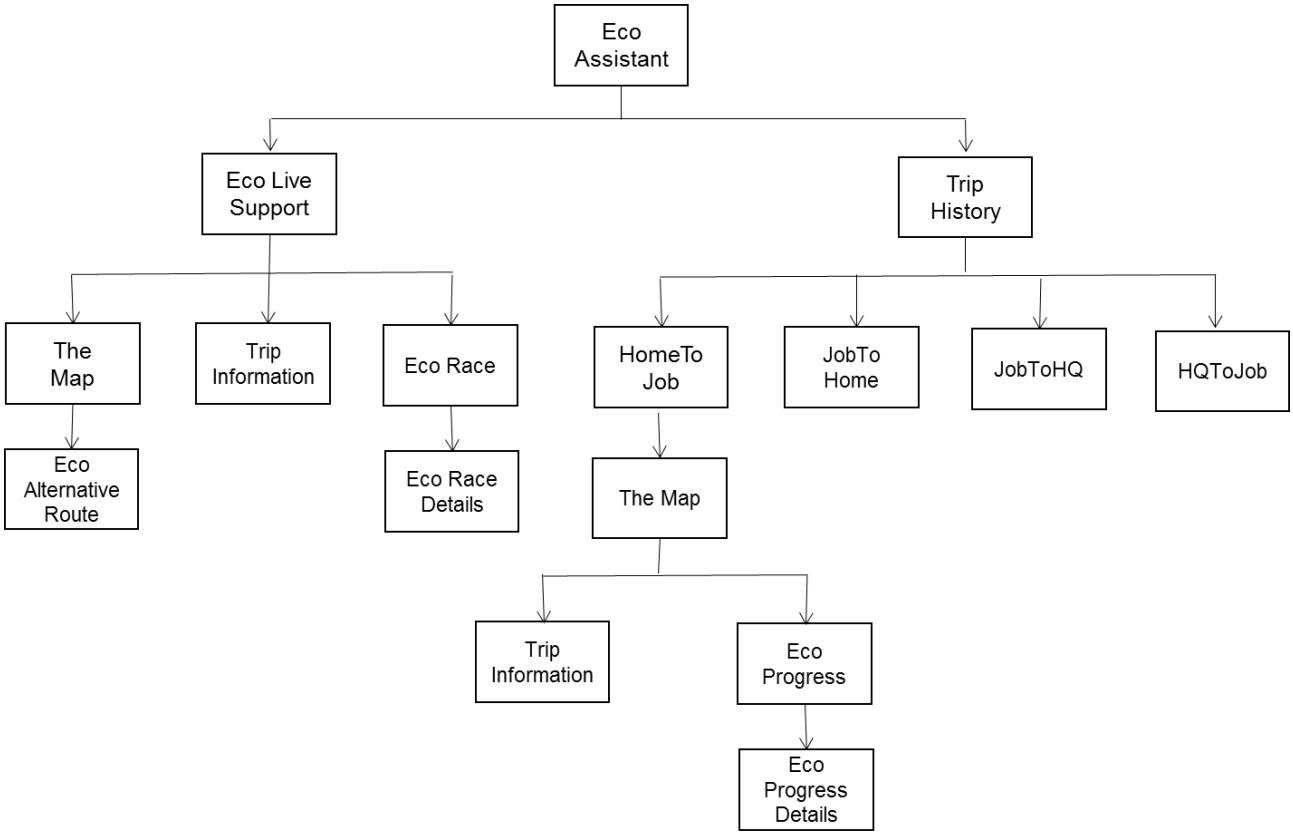

Figure 3: EcoAssistant App Navigation Diagram Flow

After starting the application the main menu in the Figure 4 is shown.

The system reproduces vocally what appears on the screen saying “Do you want to launch eco live support or see trip history?“

The user can navigate through the app using the menus or voice commands as provided by the PMA’s Multimodal User Interface.

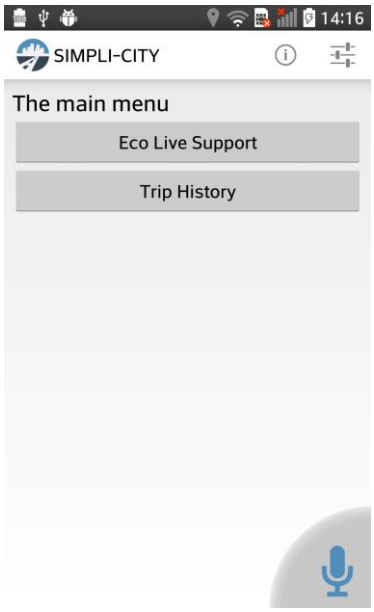

Figure 4: Eco Assistant Main Menu

3.3.3.1 Eco Live Support

Choosing the first option “Eco Live Support” by voice or by touching, the next destination is retrieved from the user’s agenda and the next screen is shown, see first image in Figure 5. The system reproduces vocally what appears on the screen and asks for further interactions, e.g. in this case “Your next destination is Corso Luigi Settembrini, Do you want to go to the map, know information about your trip or start Eco Race?“.

Choosing the first option “The Map”, the map, showing the route from the current car position to the next destination is visualized as shown in the second screen below, while the system says “Your next destination is Corso Luigi Settembrini, Do you want to find the more ecological route?“

Choosing “Eco Alternative Route”, the system shows on the screen (see the third screen in Figure 5) a map visualizes both, the alternative and the “normal” route to reach the given destination.

From this screen the only possibility is going back to previous screens along the navigation flow; while this screen is visualized, the system says ”do you want to go back”.

Figure 5: Eco Live Support – Map for Usual and Alternative Route

Choosing the option “Trip Information”, information about the next planned trip is visualized on the screen and vocally reproduced as shown in the second screen of Figure 6.

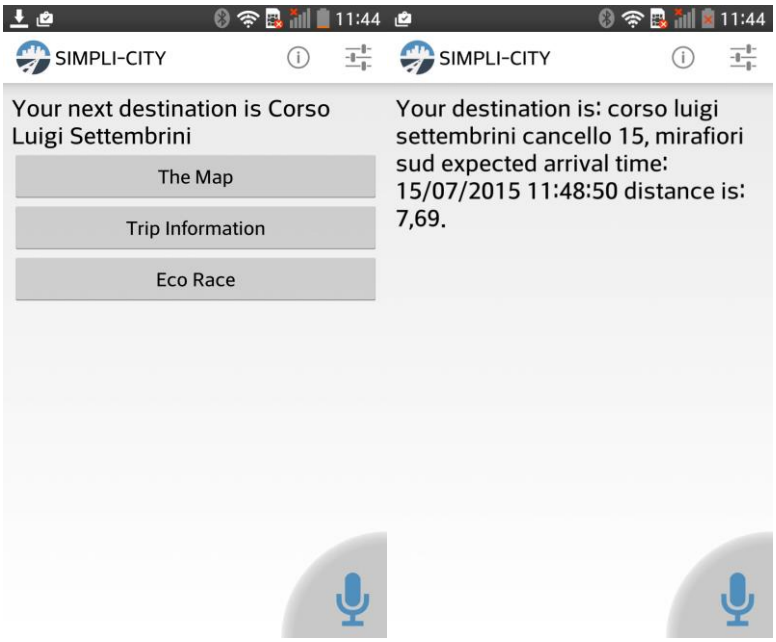

Figure 6: Eco Live Support – Trip Information for Current Trip

Finally, choosing the last option “Eco Race” from the “Eco Live Support” menu, a gauge is visualized that highlights the difference between the ecoscore just achieved for the last portion of the route and the average ecoscore achieved in the past for the same route portion (see the second screen in the Figure 7). While this screen is shown, a comment is reproduced by the system for stimulating the user to improve his/her own score. There are three possible situation that can occurs:

- The current ecoscore is lower than one previously achieved
- The current ecoscore is close to the one previously achieved
- The current ecoscore is higher than one previously achieved

The visualized gauge shows an arrow on the left, in the middle or on the right in the three respective situations. The comments reproduced for the three situations changes as follows:

- Your ecoscore is lower than usual: attention!
- Your ecoscore is steady: try to improve it!
- Your ecoscore is better than usual: very well!

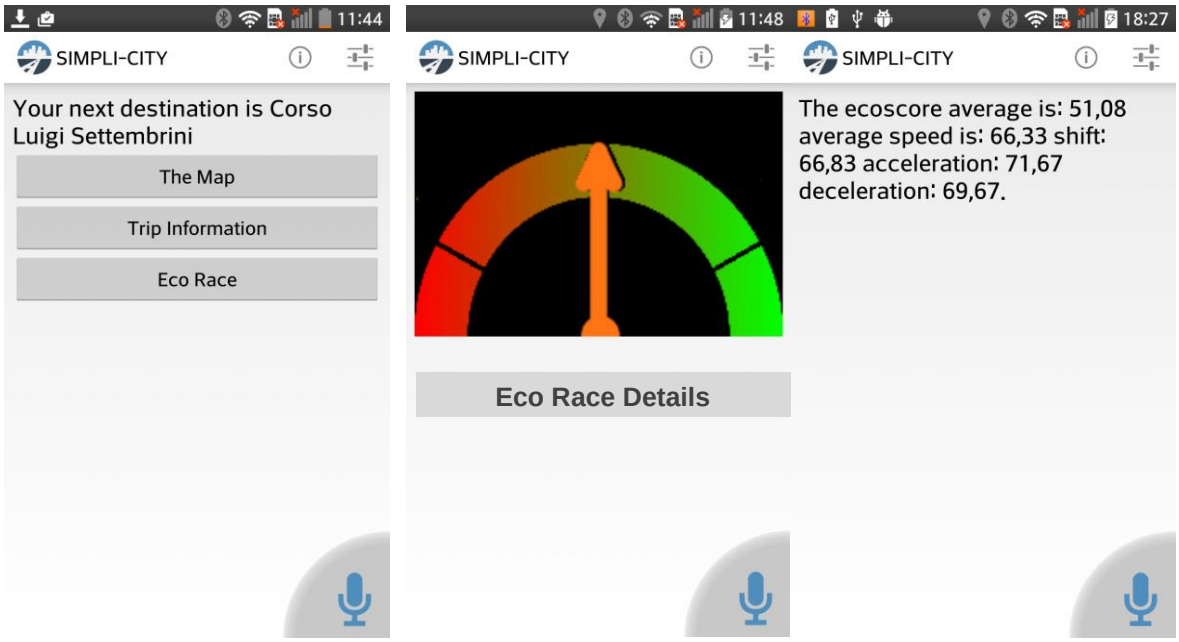

Figure 7: Eco Live Support – Eco Race and Related Details

3.3.3.2 Trip History

Choosing the second option “Trip History” from the main menu, the list of past trips stored in the history trip repository is visualized as showed in the first screen in Figure 8.

After selecting one of the historical trips from the list by voice or by touch, a map showing the corresponding route is visualized (see the second screen in Figure 8). The route portions are colored according with the average ecoscore achieved so far. Two options are available from this screen and reproduced by voice “Do you want to know information about your trip or go to the Eco Progress”.

Choosing the option “Trip Information” several information about the selected trip is visualized on the screen and vocally reproduced as showed in the third screen of Figure 8.

Figure 8: Trip History – Maps and Information of Selected Usual Trip

Choosing the second option “Eco Progress”, a gauge is visualized. This gauge evaluates the average ecoscore achieved for the selected trip (see the second screen in the Figure 7). While this screen is shown, a comment is reproduced by the system for stimulating the user to make own score better and better.

As prior, in this case there are three possible situations depending on the level of the ecoscore that can be low, sufficient or good. The gauge shows the arrow in different position according with the ecoscore level (on the left for low, in the middle for sufficient and on the right for good) while the reproduced comment changes respectively as follows:

- Average ecoscore is low: try to achieve your best that is ..
- Average ecoscore is sufficient: try to achieve your best that is
- Average ecoscore is good: try to achieve your best that is ..

From this screen is possible to select the option “Eco Progress Details”. After choosing this option, details about the eco progress for the selected trip are visualized on the screen and vocally reproduced as shown in the third screen in Figure 7.

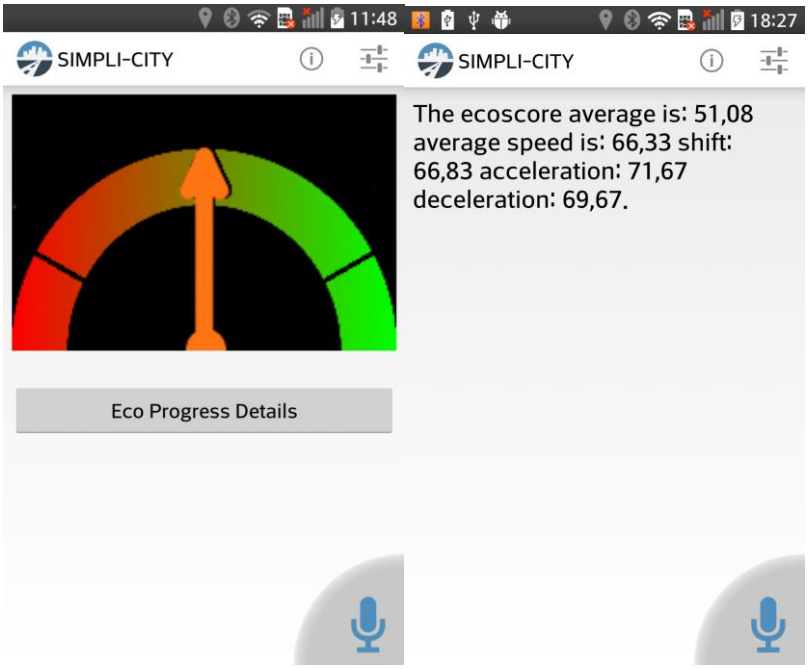

Figure 9: Trip History – Eco Progress and Related Details of Selected Usual Trip

3.3.4 EcoContest App

The EcoContest is a separate app based on the SIMPLI-CITY components and makes use of its underlying concepts. It has been implemented in two variants. Firstly, a set of libraries have been developed to measure, monitor and calculate the base data for the EcoContest. This includes the monitoring of eco related information such as the speed, the acceleration, the car model, the location and others. This information is anonymously sent to the SIMPLI-CITY server side, where values are compared to those of other users using the EcoContest. This allows the EcoContest to rate the user and its driving behavior and to compare it to other users. This part of the EcoContest may be used in different apps based on SIMPLI-CITY.

Secondly, the EcoContest UI has been developed. This UI provides a stand-alone view shown in Figure 10. A different view has been chosen as the EcoContest is likely to be provided - and used - as an independent app that is running alongside with either other SIMPLI-CITY components or even as an own app in the app store hence lowering the entrance barrier for first time users significantly.

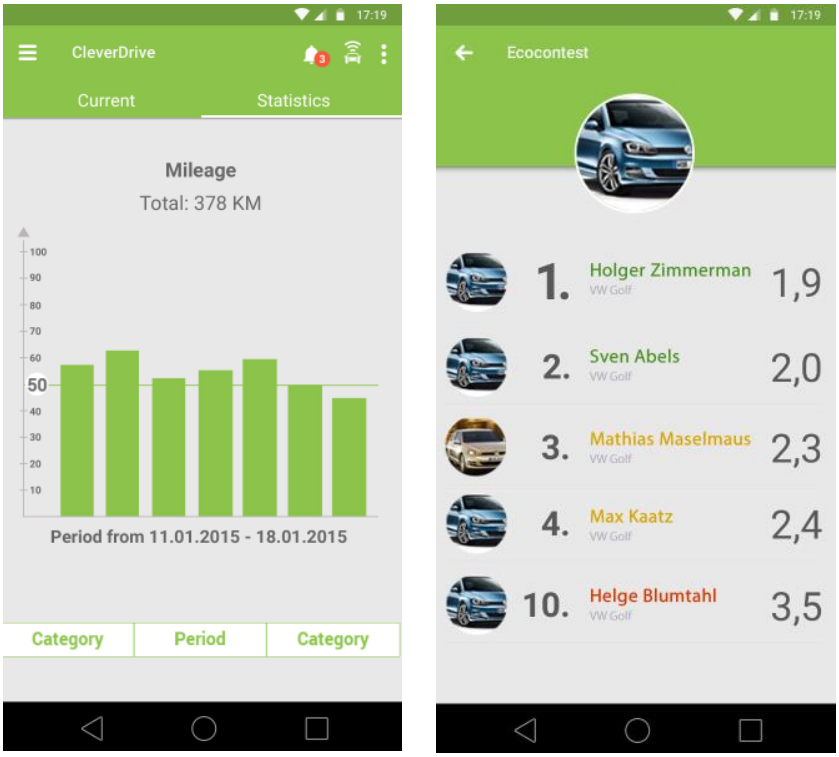

Figure 10: EcoContest – Statistics and Comparison to other Drivers

Figure 10 shows two views of the EcoContest to the driver. The left side shows the long-term protocol of the app while the right side shows the comparison of a driver to other drivers. The app provides a web based view, which is normally used to track and analyse the own driving behaviour. Figure 11 shows the equivalent web based view of the driving comparison.

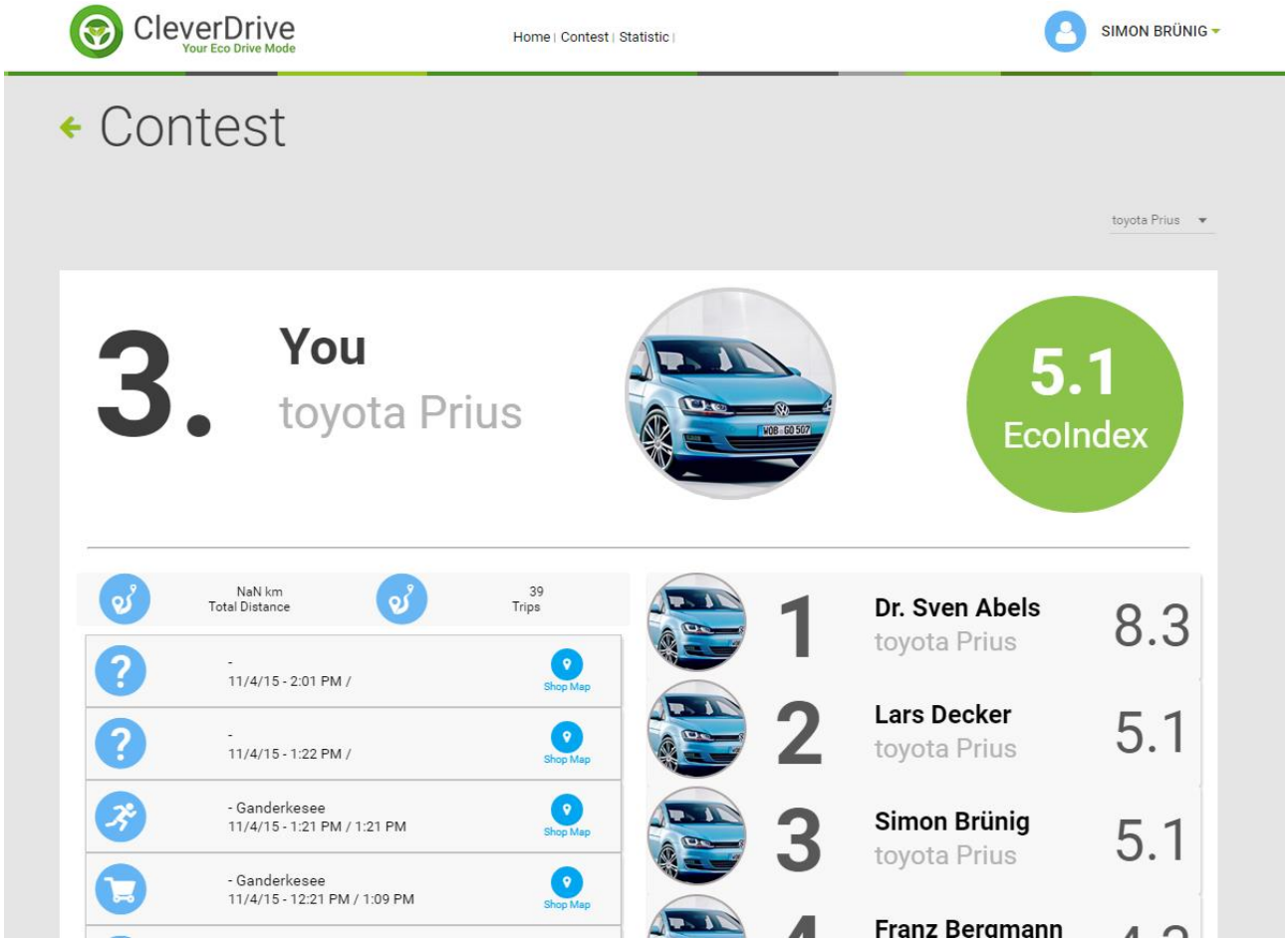

Figure 11: EcoContest – Web View

3.4 List of Changes from the Topic Definition and Justification

- Simplified Menus to provide an easier navigation: during the implementation of the App several options have been simplified in order to make the conversation between user and device more fluently. For example excessive nesting of menus have been avoided.
- Italian version of EcoAssistant App: an Italian version of the EcoAssistant app has been implemented in order to engage the Italian natives in the evaluation according with the esplicit request dne by the reviewer.
- Separated app for EcoContest: In order to speed up the development and in order to create an additional exploitation possibility, the EcoContest has been realized as a stand-alone app on top of the SIMPLI-CITY components. This allows people an easy start for SIMPLI-CITY without the need to download the full stack and it allows a more focused exploitation towards green driving, which is also reflected by the green color of the app interface.

4 Use Case Topic II.2: “Rising the Driver’s Comfort”

4.1 Topic Overview

The main objective of this Use Case topic is to offer a set of services able to improve the road safety and entertainment during the normal use of the vehicle, extending the existing embedded software that could be already available in the car. In more detail, the following functionalities have been implemented within this topic:

- Route planning and navigation. The application is able to propose routes to the user for the navigation according to several aspects, like daytime, type of transport, etc.
- Accident prevention warning. The application has the ability to show accident black spots in the route and even warn the user when approaching a black spot during the driving.
- Notification of proximity of points of interest. The application is able to notify the user the proximity of a point of interest (POIs) of different kinds, like gas stations, touristic attractions or hotels.
- Social network integration. Users can share the POIs they find during their travel through social networks.

The functionalities have been integrated in a single application called DrivingExperience. For a more detailed description of this second Topic on this Use Case, please see deliverable D8.1.2.

4.2 Topic Architecture

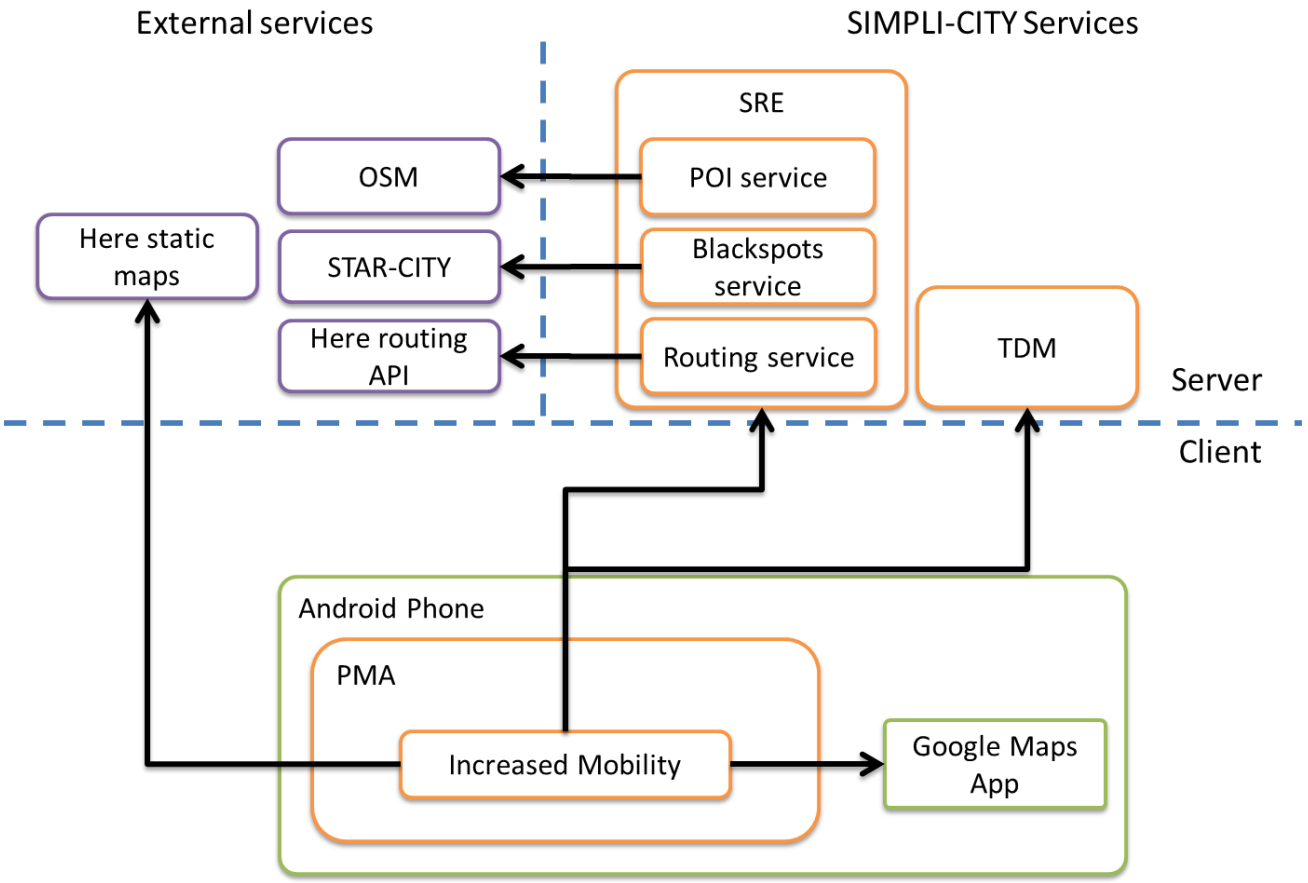

Figure 12: App Architecture

The “Rising the Driver’s Comfort” application is composed of several modules both in client and server application:

- Increased Mobility app. This is the main component created for this Topic. This application executes on top of the PMA and is responsible for communicating with all the server components to get information, show this information of the user and retrieve answers to questions. It is also responsible distributing this information to the TDM when needed. See section 4.3.3 for a full description of the app.
- SRE is the set of services offered as part of the SIMPLI-CITY platform. For this Topic, three of them are used. More information on Section 4.3.2
- TDM. This component is responsible for handling the conversation with the user, giving appropriate answers and options to the user, and requesting any missing information. For this Topic, a new Dialog Domain Descriptor has been created to contain all needed app navigation and conversation flow.
- External Services. In order to integrate all Data Sources needed for the application, several external services, both from SIMPLI-CITY partners or third-party companies are used. Find a description of the Data Sources in Section 4.3.1

4.3 Use Case Implementation Review

4.3.1 Data Sources

In order to have a rich set of data to show to user, several Data Sources have been used. Some of these services or Data Sources are the result of the work of SIMPLI-CITY partners, and some are from third-party companies. This is the list of Data Sources:

- Open Street Maps (OSM). OSM is an open project intended to create a geographical worldwide database through collaborative work. OSM contains not only vector-based and rasterized maps of the whole world, but also information about elements in the map, like buildings, services, touristic attractions, street elements, gas stations, etc. In our Topic implementation, it is used to extract information about POIs.
- STAR-CITY. This is an information system developed by IBM and already used for the Use Case presented on WP7. It extracts information from several sources and uses it to create a model of the current and future status of the driving conditions. For this Topic, the information extracted is the list of Black Spots in the route.
- Here Routing API. This is a service offered by Here Maps that generates a route based not only on starting and destination points, but also allowing the use of waypoints, transportation modes, avoid areas, etc. This is the main source for the routing.
- Here Static Maps. This service provides static images based on route information (origin, destination, waypoints, etc). This is mainly used to present visual information to the user while managing routes.

4.3.2 Backend Services

The Routing service provides the method getDirections. This methods takes as input parameters:

- Current latitude and longitude of the vehicle (collected through the PMA sensors)
- Destination address (selected by user)

And, returns the different data needed to create a route, this data is processed and shown to the user in an appropriate way.

The Black Spots service returns information about nearby traffic anomalies. In Road Safety Management, a Black Spot is a place where road traffic accidents have historically been concentrated. It takes as input the user location and it returns information about nearby traffic Black Spots information, in particular, it gives helpful information: a description of the incident, the location and the distance from the user to the black spot.

The POI service returns information on Tourist Attractions or Gas stations around the specified area. The method getPOI uses as input:

- Current latitude and longitude of the vehicle (collected through the PMA sensors)
- Radius around the location in meters
- Type of element like e.g. Gas stations

The output of the service is information of the requested element, giving the location, address and a description.

4.3.3 DrivingExperience App

The DrivingExperience application has this menu navigation:

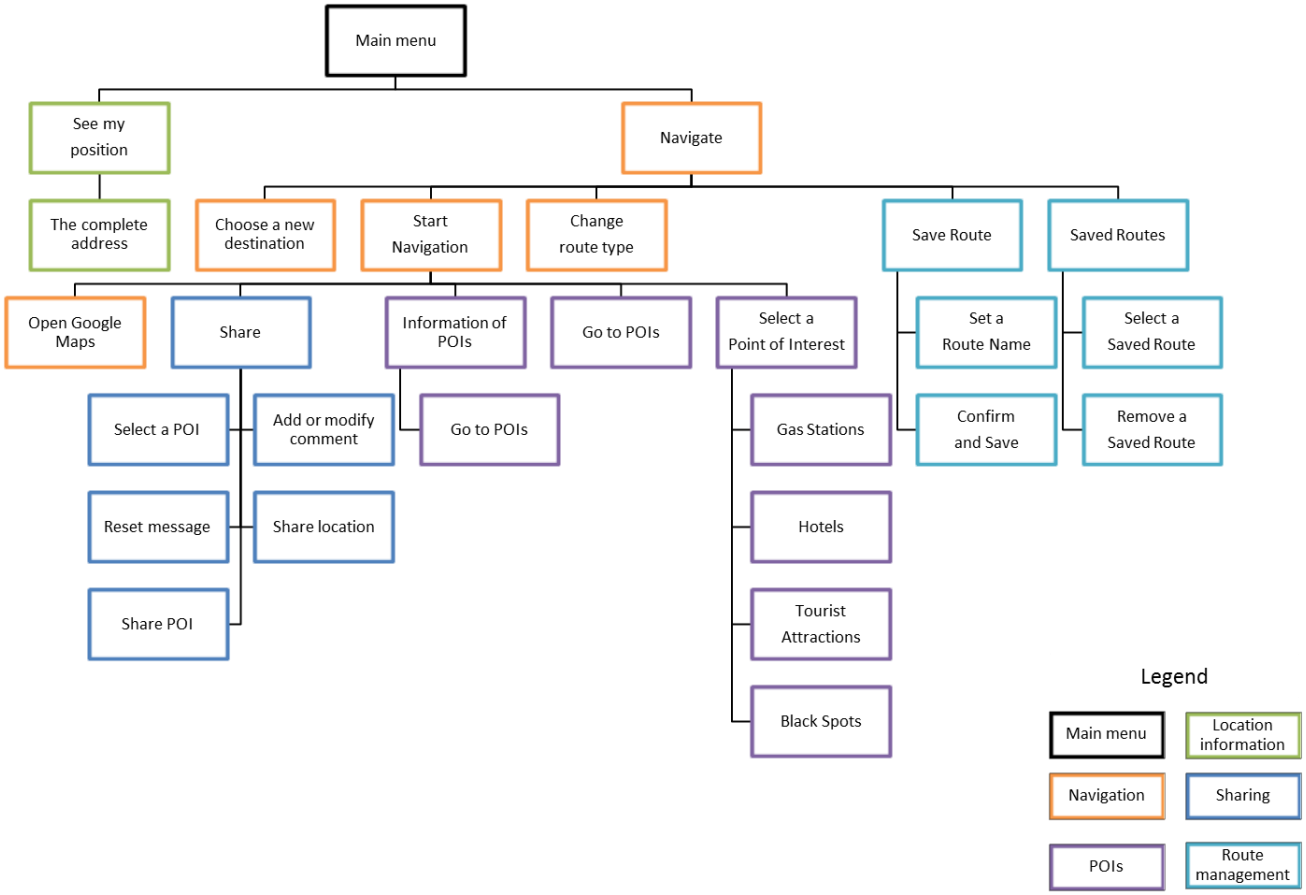

Figure 13: App Structure

The main menu of the application contains two options as depicted in Figure 14

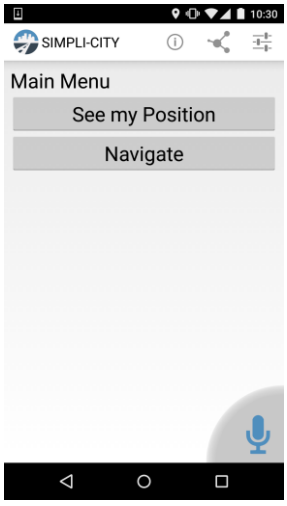

Figure 14: Main Menu

This is a standard screen of the application. User is presented with several options to choose. These options are presented in two ways, first they are shown in screen as buttons, and second, they are read aloud by dialog engine. The user has two ways of

selecting an option. First, tapping on one of the buttons, and second, pressing the mic button in the lower right corner and reading the option. In some cases, when the amount of options to choose is too large, only voice is allowed as user input. For example, when asked for an address, user can only answer by voice.

4.3.3.1 See my Position

If the user selects “See my Position”, a map displaying the user location and a description below is shown allowing the user to know where he/she is located (see Figure 15). Additionally the user can choose to see address information in more detail by selecting the “Complete Address” option (see Figure 16).

Figure 15: See my Position Menu

Figure 16: Complete Address Screen

4.3.3.2 Navigation

After selecting the option “Navigate” in the main menu, the user is requested to enter a destination (if it was not previously set). After selecting the destination address, the Navigation section is shown.

The user has then the option to change the destination, change the route type, or manage saved routes.

When the destination and route type are set, user can select “Start Navigation” to show all options related to navigation as seen in Figure 17.

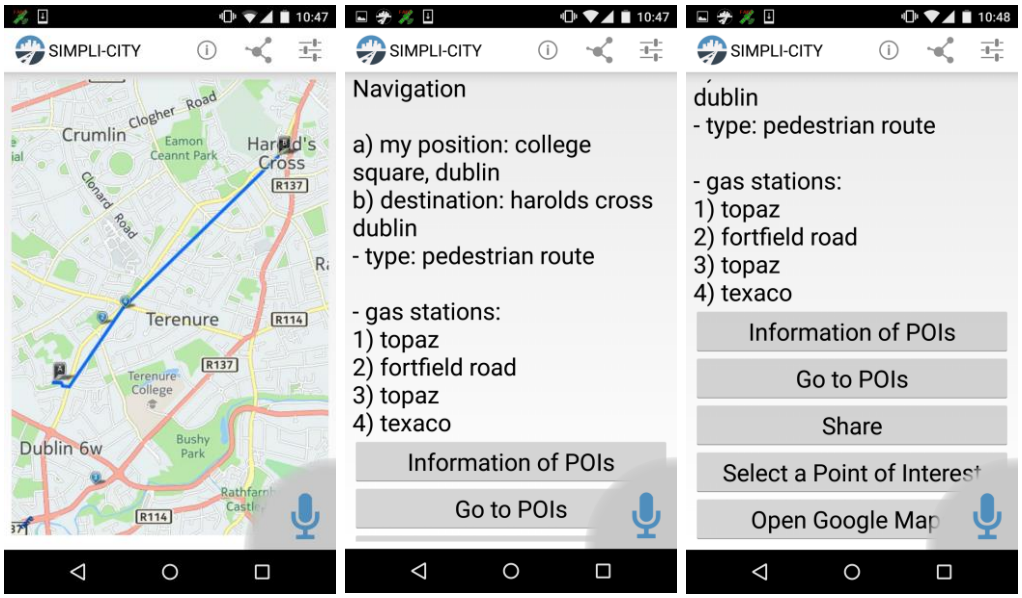

Figure 17: Start Navigation Screen

This screen shows:

- A map, with current location, destination and proposed route.
- User current location
- Destination location
- Route type
- Near POIs of selected type

From this point, user will receive notifications about nearest POIs of the type selected (gas stations, black spots, etc.) even if application is in background.

The user can share her or his experience by selecting the share option, change the destination to a nearby POI or even see more detailed information about the POI (see options in Figure 17).

4.3.3.3 Points of Interest

The POI can be found in the Start Navigation screen. Additionally, the user receives notifications about the Point of Interest that the user has chosen. Once the user opens the notification, she or he will be redirected to navigate to the POI.

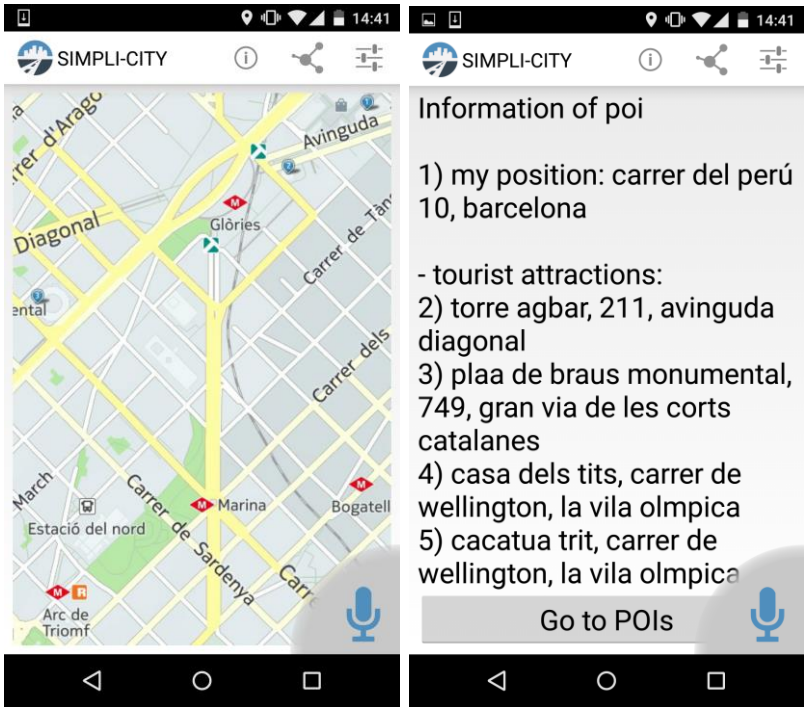

Figure 18: POI Screen

4.3.3.4 Save Route and Saved Routes

If the user selects the “Navigate” option, the user is prompted to select a location where he or she wants to go. Once the destination is entered successfully, a menu will be displayed containing options to save a route (see Figure 19) and to see the previously saved routes to retrieve them (see Figure 20).

Figure 19: Save Route Menu

Figure 20: Saved Routes Menu

4.3.3.5 Social Integration

In the “Start Navigation” screen (Figure 17), the user can share information (about a POI or about a Route) to a social network, just by selecting “Share”. The user has two options (Figure 21), first, directly sharing through Facebook (see Figure 22), or, second, to share using Android standard sharing, which allows to share information with all the applications share-enabled that the device has installed.

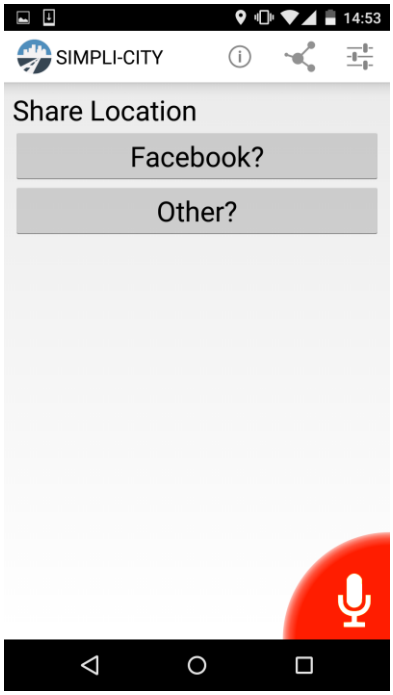

Figure 21: Share Location Menu

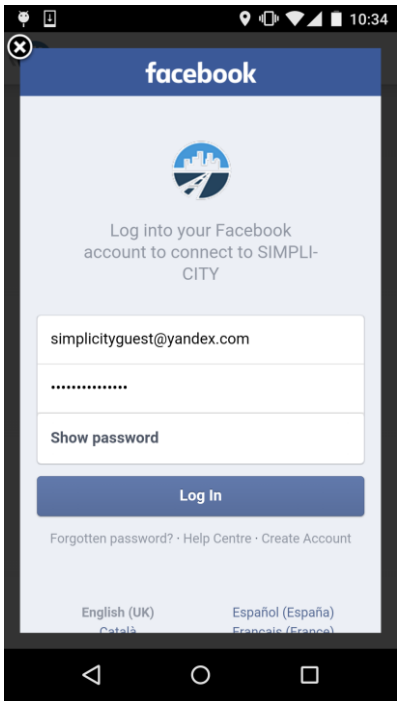

Figure 22: Facebook Login

4.4 List of Changes from the Topic Definition and Justification

- Simplified Menus to provide an easier navigation. The experience gained from WP7 and WP8 apps implementation and usability evaluation has shown us that navigation on voice-based apps cannot be the same as for non-voice-based apps. In order to make the conversation between user and device more fluent, several options have been simplified, and the phrasing for several options have been adapted to conversation and dialog, though they look not so natural as screen options.
  One example of these simplified menus is the way addresses are entered. In the original Use Case specification (D8.1.2), the option was defined as three questions, street address, city, and country. This is a very unnatural way to set an address via voice, and making the process error-prone made it even more unnatural. In the final deliverable application, the whole address is stated in a single question, allowing the use of external services to check if the whole address is correct, and making the whole dialog more fluent and natural from a human point of view.
- Share functionality is not only narrowed to a specific social network (i.e.: Facebook) but through the Android standard share system, user will have the ability to share information through the apps installed on his/her device.

5 Conclusions

The aim of this deliverable D8.2 is to present the final implementation of the Use Case II: “Enhancing the Driving Experience”, that has been used to test the SIMPLI-CITY platform and solutions and the reliability of the Personal Mobility Assistant prototype in the real world.

Two scenarios have been implemented as applications that support the whole trip from the preparation to the arrival to the destination “Enhancing the Driving Experience”.

The two topics indeed cover two different aspects in which the driver is interested: The sustainability and reduction of fuel consumption and the comfort and peace of mind from the other.

The first application, EcoAssistant, addresses the first Use Case Topic (II.1, “Environmental Awareness Rising”) as mentioned in Section 3, while the second application “Rising Driver’s comfort” solves the second Use Case (II.2 “ Rising the Driver’s Comfort”).