QROWD - Because Big Data Integration is Humanly Possible

Innovation action

D2.4 – i-Log

<table>
<thead>
<tr>
<th>Author/s</th>
<th>Fausto Giunchiglia, Mattia Zeni, Enrico Bignotti, Alessio Zamboni, Gabor Bella, Andrea Passerini, Ivano Bison, Mattia Fumagalli</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due date</td>
<td>31.11.2018</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
</tbody>
</table>

Project co-funded under Horizon 2020 Research and Innovation EU programme, grant agreement no. 732194
# Table of Contents

Table of contents 2

Abstract 3

Executive Summary 4

1. Introduction 5

2. Description of i-Log 5

3. i-Log architecture 6

4. Data collection module 6
   4.1 Adaptability to different smartphone models / operating systems 7
   4.2 Battery efficiency 8
   4.3 Temporary data storage 9
   4.4 Data collection remote configurability 10

5. User’s contributions module 10

6. Communication module 13
   6.1 To the phone 13
   6.2 To the server 14

7 User interaction module 14
   7.1 Installation procedure 14
   7.1.1 Experiment selection 15
   7.1.2 Login with her account 15
   7.1.3 Informed consent form (ICF) 16
   7.1.4 Grant permissions 16
   7.1.5 Fill in a questionnaire to profile the user 17
   7.2 User interface 17

8. Similar applications 19

9. Demonstrator description 20

10. Conclusions 22

References 23

Annex 1 - List of available sensors 25
**ABSTRACT**

Travel surveys are steadily being moved from paper-based to sensor-based, leveraging the power of devices already owned like smartphones and wearables, while at the same time balancing the level of intrusiveness, to ensure a high rate of response and not hurting the relationship between citizen and public administrations. This deliverable illustrates the i-Log mobile application, which collects streams of big personal data from the user in an unobtrusive, privacy compliant and efficient way. As such, it will be used as basis to test the system and to run the business case evaluation phase. Within the QROWD project, i-Log is deployed in the Business Cases involving citizens to obtain information about their modal split by collecting their mobile data and asking questions about these data to improve the overall accuracy, allowing integrated real-time information on traffic and multimodal transport. i-Log is also the interface for harvesting crowdsourced data, which means that the citizens participate in challenges to help to improve the understanding of the mobility infrastructure in the municipality of Trento.
EXECUTIVE SUMMARY

The deliverable D2.4 presents a general overview of the i-Log mobile application and its deployment basis to test the system and to run the business case evaluation phase. i-Log is an application which is able to collect user's personal information, generate streams of data from smartphone’s integrated sensors, and also collect feedback from the user in a variety of input formats. Its purpose within QROWD is to provide both an interface for harvesting crowdsourced data and integrated real-time information on traffic and multimodal transport.

The audience for this deliverable is constituted by the QROWD members, especially the ones involved in the business cases of the Municipality of Trento since i-Log is to be installed on the citizens' smartphones and will allow them not only to have their data collected but also to participate in crowdsourcing campaigns.

D2.4 is related to the deliverables that describe the work carried out with respect to the pilots, namely D1.4, D2.2, D3.2, D7.3, and D8.3, and in general provides the technical information for WPs of the business cases (WP1, WP2, and WP3), in addition to the architecture-based WPs (WP7 and WP8).

The main output of the deliverable D2.4 is the technical specifications of the features to be used in QROWD for running the pilots within the business cases of the Municipality of Trento.
1. INTRODUCTION

This deliverable provides the technical specifications about the deployment of i-Log mobile application, which is utilized for the collection of both sensor and user-based inputs in crowdsourcing campaigns. Indeed, smartphones allow for the collection of user-based inputs which can take a wide variety of formats, from text to visual. Such features make smartphones the ideal tool for crowdsourcing experiments that require to tap into the wisdom of the crowd and exploiting their knowledge. Once it is installed on the user's smartphone, it unobtrusively collects the sensor data and at the same time interact with user depending on the crowdsourcing strategy, e.g., answering a survey or taking a photo of their surroundings. With respect to the QROWD project, i-Log is a pivotal tool to obtain data from citizens of the Municipality of Trento about real time information on traffic and multimodal transport and involve them to improve its mobility by enabling data-driven policy making by providing the users with an interface for harvesting crowdsourced data.

The remainder of the document is structured as follows. Section 2. Description of i-Log introduces i-Log, while Section 3. i-Log architecture provides an overview of the architecture. We dedicate a section for each module, where Section 4. Data collection module details data collection module, Section 5. User's contributions module the user contribution module and Section 6. Communication module presents applications similar to i-Log and makes a comparisons. Then, Section 8. Similar applications. Section 9. Demonstrator description explains the demonstrator and Section 10. Conclusions concludes the deliverable.

2. DESCRIPTION OF I-LOG

As described in [17], i-Log is a mobile application system that collects information from two main data sources, i.e., humans and sensor:

- **Sensors:** i-Log is designed to collect data from up to 30 different sensors simultaneously, both hardware (e.g., GPS, accelerometer, gyroscope, among others) and software (e.g., in/out calls, applications running on the device). The full list of the available sensors in i-Log with the corresponding sensing frequency is presented in Annex 1 - List of available sensors. A dedicated backend infrastructure manages the tasks of synchronizing and storing the streams of data from the smartphones.

- **Humans:** i-Log can obtain information from the user in a variety of ways, depending on the crowdsourcing design needs. At the time of writing, i-Log can collect the following inputs from a user: text (be it free or closed), photos, and videos.

i-Log has been designed to be modular and adapt to each smartphone model,
especially in terms of sensing strategies for both smartphones and their internal sensors (which can greatly vary among different models), ii) to be robust to Android version changes, iii) to consume as little battery as possible, by devising sensor-dedicated energy consumption strategies and delegating all computation server-side, and iv) to ensure users’ privacy from data collection to its analysis.

3. i-Log Architecture
A simplified version of i-Log’s architecture is presented in Figure. 1 - i-Log architecture. The system is composed of a set of modular components that have their roles inside the application. Based on how the system has been designed, each component is logically isolated from the others so that, if necessary, they can be modified or replaced without affecting the others as soon as the input/output specifications are respected. This architecture gives i-Log a significant advantage in terms of adaptability and extendibility of its features. In the next sections, we will describe in detail the three main components of the i-Log application, namely the data collection module, the user’s contribution module, and the communication module.

4. Data Collection Module
This is the main module of the application, which is responsible for the data collection from the smartphone’s internal sensors. We describe it in this section by presenting its main features and characteristics.

4.1 Adaptability to different smartphone models / operating systems
Android devices come in a variety of brands, shapes, sizes, and capabilities. They can differ from one to another in terms of operating system-level features, pre-installed apps, and innovative interface experiments. Unlike Apple's iOS, the Android Open Source Project (AOSP) gives device manufacturers a lot of leeway in creating unique features that will draw customers. This can be an advantage with
i-Log

respect to a closed source system, where the owner has full control over the hardware and platform. However, at the same time can create issues that have to be solved, at the application level for example, like in the case of i-Log. Before going into the details of our implementation, we will explain how a branded Android version works, so that to clarify and put into context the work done.

When a manufacturer produces a new device or when it has to update an already sold one, it starts by taking the last available version of the code of the AOSP project. Among the first things they add, there are the drivers required by the device to operate special hardware embedded in it, like dedicated components or sensors. Additionally, they would want to customize the visual elements, like menus, widgets and pre-installed applications.

In the context of the i-Log application, the most critical aspects of the customization process are two:

1. specific battery-saving solutions
2. the drivers of the sensors

Since these two elements are brand or even device-dependent, it is always critical to deal with them because in the worst case scenario you have to define a specific solution for each of them. The Android project makes standard APIs available, specifically for the sensor part, but, due to the many differences, they are always problematic and buggy. Consequently, a considerable effort was devoted to the debugging of the data collection process, and we can obtain stable results on most of the devices on the market at the time of writing:

- Samsung
- Motorola
- LG
- HTC
- Huawei and Xiaomi (partially, it requires additional procedure at the user side)

Similarly, each brand has different personalizations at the operating system level to enable specific strategies to save battery on the device. There are three main strategies to do so:

1. **Stock Android (AOSP)**: Samsung, among others, share the same approach with the stock Android (AOSP), which facilitates our work because we can use the standard APIs to deal with them, while others use specific solutions that require custom code.

2. **Whitelisting**: Huawei and Xiaomi have a whitelist of applications that are allowed to run without restrictions. By default, an application is not inside this list, and the user has to add it manually: for this situation, we modified i-Log to help the user in this process by, for example, opening the menu window where they can grant the permission to whitelist the application. Unfortunately, this works only on stock roms.

3. **Brand-specific strategies**: These are the worst case scenarios situations where we can do little to modify the app. For example, OnePlus has a modified operating system that simply kills the applications, without any advance notice and without allowing the application to perform any activity before closing. This behavior is actually considered a bug by most developers, and there is little that can be done on the developer side until
OnePlus releases an official answer and/or solution. To conclude, except in some rare situations, we believe the massive amount of work did in this regard allows i-Log to run on almost any smartphone without compromising the user experience.

4.2 Battery efficiency

The evolution in the smartphone industry is so disruptive that every year important breakthroughs are made in regard to features and device characteristics. Unfortunately, there is still a significant limit in today’s devices — the battery. Mitigation to the phenomenon (but not a solution) is given by the increase in the size of the device which results in an increase in the battery size, allowing for more energy storage capacity. Additionally, we start seeing fast charging solution that allows charging the phone in less than one hour.

Due to this critical limitation a smartphone can rarely last more than a whole day with normal usage patterns. This phenomenon is even worse when the phone is used for tasks other than messaging, calling and browsing, e.g., for collecting data from the internal sensors, when the battery can barely last for few hours. In fact, as several works in the research community pointed out [2,12,15], there are sensors inside the smartphone that consume a considerable amount of energy. The battery life problem in modern smartphones is today considered their major limitation with apparently no short-term commercial widespread solutions. This aspect is strictly related with the user experience and in the end it becomes an obtrusive factor. Having a device that needs to be charged every 5 hours makes impossible to use it to collect realistic data. The user will most likely uninstall the application after a few days because the application will prevent her from using the smartphone normally.

In i-Log, we addressed this issue by finding a good solution that balances the needs of the data scientist (in terms of data quality/quantity) and the user. We analyzed each sensor consumption rate under different conditions, the collection frequency and the combination of multiple sensors together. We discovered that the collection rate affects the battery consumption (as well as the data quality as explained above) and we found a balance between the two needs. The sensors that consume more energy on modern devices are the radio ones, namely Wi-Fi, Bluetooth, and GPS. The GPS, in particular, can consume up to 3138.99 times the energy consumed by the phone in idle state. In a smart city scenario like QROWD, the data collected by the GPS sensor are the most important one and allow generating the most useful insights.

For this reason, we had to design the data collection process carefully and find solutions so that to make the process energy efficient. Apart from general optimization tweaks, most of the work focused on using the GPS only when it is strictly needed. This translates in the following actions:

- We reduce the collection frequency of the GPS sensor to one value per minute (configurable for every experiment). To select this value we leveraged on the assumption that in one minute, in an urban environment with a speed limit of 35km/h (but with a lower average speed due to traffic congestion) a car travels for up to ~500m, which allows us to keep track of the path it performed.
i-Log

- We reduced the accuracy of the GPS measurement so that the position could be collected faster. In fact, when the GPS starts collecting the location, the very first readings are very inaccurate (in the order of hundred meters) and progressively improve with a final accuracy in the order of some meters. To save battery, we decided to reduce the timing the GPS is turned on for every measure (every 1 minute) at the expense of the final accuracy. We selected a threshold (configurable for every experiment) of 15m in the final accuracy so that when the first measure with an accuracy value equal or lower than the threshold is collected, we stop the GPS sensor and wait for a minute.
- We defined a threshold to the time required by the GPS sensor to generate an accurate measure. If after 20 seconds (configurable for every experiment) the GPS was not able to collect a position accurate enough according to the threshold defined at the previous step, we stop the GPS sensor, and we wait for the next one-minute interval.
- We defined a rule according to which the GPS is turned off if the smartphone is connected to a Wi-Fi network, since we are able to locate the person with an accuracy of ~20m through the network. Using the GPS in these situations will provide little benefit and a considerable battery drain.

All these counter measurements with the GPS translated in a ~2/3 of energy savings, in the worst case. As of now, on a modern smartphone (e.g., OnePlus 3) i-Log can run continuously from 7AM to 9PM with a single charge.

4.3 Temporary data storage

The amount of data generated by i-Log varies, depending on the configuration of each experiment. In the worst case scenario (from a storage capacity point of view) with all the sensors collecting data at the highest frequency, it can quickly generate ~1.5GB of data per day per user. As per the requirements of the business cases, the data have to be sent on a daily basis to the back-end infrastructure to be analyzed and contribute to the business case. To do so, we have to synchronize them through a Wi-Fi connection (more details will be provided in Section 6.2 To the server). Since it may be the case that a Wi-Fi connection is not always available, we designed i-Log to store the data locally temporarily. On modern smartphones, storage is no longer a problem, and it is also quite cheap compared to other features/components. A study tells us that by the end of 2018 the average storage capacity of a smartphone will be of ~60GB, more than enough to store all the user’s data and the data generated by i-Log even with some days without Wi-Fi connectivity.
On i-Log, the data are kept in-memory till they reach a specific (configurable) size and then they are flushed on the smartphone internal storage as a zipped log file, as shown in Figure. 2 - Flow diagram of data storage. The algorithm chose to compress them is Bzip2 because it proved to be the one with the highest compression rate still keeping a good processing time to be executed on a smartphone. Each log file is a csv where each row is a sensor reading characterized by a label defining the sensor, the collected value(s) and a timestamp expressed using the UNIX timestamp in milliseconds:

```
ACC,9.81,0.89,0.01,1540824238000
```

The logs file are saved inside the application’s own package (called sandbox) that prevents anyone (except the application itself) to access it. This means that no one will be able to access the collected personal big data apart from i-Log, in the way we define it. This means that also the user cannot extract her data from the application, but instead, she has to login to the backend and access the data from there. This was done on purpose to forbid unwanted people to have access to the data, e.g. in case the phone is lost or stolen. The way we saved the data also facilitates its deletion: in fact, it is enough for the user to uninstall the application also to delete the logs temporarily stored on the device, something that would not be possible if the files were stored somewhere else on the device.

### 4.4 Data collection remote configurability

One of the requirements of the business cases was to allow i-Log to be used simultaneously in multiple data collection sessions. To enable this, we moved the experiment definition and configuration to the backend server and let the user on i-Log select which experiment to participate in by using a code to be typed in the application. This allowed us to re-design the data collection module and make it configurable depending on the characteristics of each pilot. More in details, there are two main parameters that we can configure depending on the experiment requirements in terms of balancing data quality and energy efficiency:
i-Log

1. which sensors to collect data from
2. frequency for receiving data from them.

For example, [11] states that with data generated from an accelerometer at only 20Hz and 10Hz they are able to obtain an accuracy of up to 99.27% and 95.42%, respectively.

To make a tangible example, let’s assume we run a pilot to study the modal split in the city of Trento on a sample of the citizens. In terms of parameters, we collect data from i) the accelerometer, ii) the gyroscope, and iii) GPS, each with a general frequency. At the same time, if we were to study how the citizens use their car in the city, we select the same sensors as before but with a different configuration. In this case, we can choose a different collection frequency to the GPS and increase it due to the different requirements.

5. USER’S CONTRIBUTIONS MODULE

In crowdsourcing, human knowledge is often provided as annotations or labeling of data collected via sensors [8]. This type of human contribution is at the heart of the "human in the loop paradigm"\(^1\), which leverages both human and machine intelligence to create hybrid machine learning models by involving humans in training, tuning and testing data for a particular machine learning algorithm. The final goal is to use humans to improve the quality of the results of the machine.

Given the importance of the user contribution, i-Log has a dedicated module for managing the interactions with the user to collect her input in terms of answers to different types of questions. In i-Log we define 4 different types of user’s feedback:

- **Tasks**: the user is required to reply to questions of a different type, both text questions but also richer content such as maps or multimedia elements. With respect to time diaries, tasks do not require the user feedback at fixed time interval but instead are triggered from the server when necessary;
- **Messages**: to inform the user about events related to the pilots or to provide him feedback or additional information. He is not requested to perform any action;
- **Challenges**: the user can decide to contribute (with pictures and/or text answers) to a challenge defined by the municipality. A challenge has the characteristic to allow the user to perform it whenever she wants to and is not triggered remotely like time diaries or tasks;
- **Time diaries**\(^2\): the user is required to reply to some short and fast questions at fixed time intervals. These labels will be used as annotations for the machine learning algorithms that have to learn the user behaviour and routines.

The difference between the types of user’s feedback above is only in the way they are proposed to the user, i.e., at fixed time intervals, on events, among others. On the other hand, the content is the same for all of them (tasks, challenges, time diaries). The user is presented with a set of views that require him to perform an action, e.g., provide a text answer, take a picture, pinpoint a position on a map. What

---

\(^1\) [https://www.figure-eight.com/resources/human-in-the-loop/](https://www.figure-eight.com/resources/human-in-the-loop/)

\(^2\) After careful consideration, the consortium decided to not leverage on this functionality in QROWD
i-Log follows is a list of all the views and actions currently available in i-Log and used in
the hybrid workflows of the business cases:
- **Static message with text and images (No action):** presents a series of
  messages and/or images ([Figure. 3 - i-Log screenshots](#), first row first column).
  Among other uses, an example is to explain to the user what the
  task/challenge is about and present her with the rules;
- **Text question with a map with a pinpoint (Multiple choice answer):**
  presents a text question integrated with a map view and a pinpoint on it.
  Usually the question asks the user what she was doing in that specific
  location or if the user has ever been there;
- **Text question with a map with a path (Multiple choice answer):** similar to
  the previous question, but in this case instead of a point on the map we
  present an highlighted path;
- **Text question with an image (Multiple choice answer):** the user is required
  to provide additional information in terms of a multiple choice answer about
  the image displayed to her;
- **Text question (Multiple choice answer):** the user is required to choose one
  answer among the available ones to answer the text question presented;
- **Map (Place a pinpoint on a map):** the user is asked to place a pinpoint on
  the map to indicate a specific location as specified in the question;
- **Camera (Take a picture):** the user is allowed to take a picture of the object
  specified in the description of the challenge/task. With the data about the
  image we also collect additional metadata such as the location where the
  image was taken;
- **Map (Confirm the position):** we ask the user to confirm his position. The
  process is automatic and does not require any human intervention except a
  confirmation.
These are the building blocks for the content of the user’s feedback, i.e., tasks and challenges. The hybrid workflows that describe how human can collaborate with the machine to reach the goal defined by the business cases are presented in D3.2 and D3.3. The workflows define challenges and tasks that use the building blocks described above.

Since the human factor is critical in the QROWD project, we defined a way to monitor the user behavior in replying to these feedback requests [7]. In fact, research in Social Sciences shows that user’s answers may be affected by biases, i.e., a lack of congruence between subjects’ answers and their real value [14]. Among several biases, on two specific types. The first one is cognitive bias, i.e., the inadequate recall of respondents when reporting their time use [5], while the second one is carelessness, i.e., of behaviors leading to hurriedness when answering, which may be caused by, e.g., boredom or annoyance [16]. The use of smartphones allows us to parameterize them. The two behavioral parameters are defined as:

1. $\Delta_{QA}$, formalizing the cognitive bias, and defined as the time interval (in minutes) from when the question is presented to respondents to when the answer is given. The lower the value, the better in terms of reliability;

2. $\Delta_{A(X,Y)}$ formalizing carelessness, and defined as the time interval (in seconds) elapsed from when the user starts answering one question of the time diary $X$ and answers another question $Y$, where $X \geq 1$, $Y \leq Z$ and $Z$ is the total number of questions and $X < Y$. The higher the value, the better in terms of reliability.

Additionally, we also designed a way to understand how the user interacts with the maps components of the views presented above. We do so by collecting quantitative metrics about the interaction process, such as swipes, zoom in/out, pinches and the intensity of those gestures. This helps in understanding if the user understood what she is visualizing and consequently if the information shown on the map were correct and understandable.
6. COMMUNICATION MODULE

This module is responsible for connecting i-Log with its own backend and allows the communication in both directions. The data flow diagram of the communication module is illustrated in Figure 4 and explained in more details in the following subsections.

![Flow diagram of the communication module](image)

**Figure. 4 - Flow diagram of the communication module**

6.1 To the phone

The network components (Wi-Fi and/or 3G/4G chips) on modern smartphones are among those components, together with the GPS and the screen, that consume more energy overall. To not compromise the usage of the phone and make the whole experience with i-Log as transparent to the user as possible, we had to design the communication from the server to the devices carefully. In fact, the most commonly adopted strategy in modern smartphones to save battery is to put the phone in a sleep state (also called Doze mode in Android) when the user is not using it. If we compromise this by keeping sending messages to the smartphone, we will also compromise the energy consumed and consequently the user experience. That’s why the communication server-to-device is the most critical one.

For this reason, we decided to use the Firebase Cloud Messaging service made available by Google. With this service, a developer can notify his client applications simply using a REST API: then, the delivery of the message is up to Google, which delivers it as soon as it can but always within a certain amount of time defined by the developer. The reason why this is the most efficient way to send data to the smartphone is that it leverages on the Google services natively installed in all the Android devices. This service, among other characteristics, has the advantage of delivering the message at the right moment, when the smartphone is not in a standby state or when other messages from other services/applications have to be delivered, optimizing the whole delivery process.
i-Log

The service allows sending the messages to different audiences: single devices/users but also to groups of users. This second feature is managed using so-called topics. A topic is a string which is used to group together multiple users: in QROWD, for example, a topic is the data collection pilot. At the same time, we can leverage on the socio-demographic collected from the participants and create topics to target a particular subset of the participants. For example, we would want to send a communication to all the women enrolled in the experiment, or all the students or all the men older than 40 years old.

This component, like all the others in the i-Log application, is logically separated from the others and can be substituted easily by fulfilling the input/output specifications. This gives i-Log a significant advantage in terms of adaptability and extendibility of its features. For example, due to geographical restrictions, Google is not available in all countries around the world and can be that we have to substitute this module leveraging on other services, like Amazon SNS, Baidu Cloud Push, among others.

6.2 To the server

The communication phone-to-server is less critical from a battery consumption point of view since it occurs only when the device is not in a sleep state. On the other hand, it needs to synchronize a massive amount of data, and for this reason, some considerations must be done. In fact, we cannot just transfer the data as soon as they are created, because this will quickly drain the users' data plans. For this reason, i-Log has been designed to synchronize the logs using a Wi-Fi connection while the user feedbacks can also use a 3g/4g network since. This was a design choice in QROWD, but as other components in i-Log, this can be configured based on the needs of the experiment. The main reason that drove us to distinguish between the two was that concerning the user's feedback we had more restrictive constraints in terms of real-time synchronization, while for the sensor data, collecting them once per day was excellent. To allow i-Log to send data to the backend database, we created our own API system as described in Deliverable D7.3.

7 User interaction module

This component is the one that manages the interaction of the user with the application. i-Log has a minimal user interface due to the fact that its main function is to collect personal big data. Then, to make the process as less obtrusive as possible, we put a lot of effort in simplifying its user interface to show only essential information, as presented in Section 7.2 User interface. A very important aspect concerning the user interacting with i-Log is during the installation phase, were she has to provide an initial set of information and permissions so that i-Log can run smoothly in the future, as presented in Section 7.1 Installation procedure.

7.1 Installation procedure

i-Log\(^3\) can be downloaded from the Google Play Store (Figure. 5 - i-Log on the Google Play Store) of 142 countries worldwide and is available in two languages as

---

\(^3\) https://play.google.com/store/apps/details?id=it.unitn.disi.witmee.sensorlog
of now, Italian and English. As of now, considering multiple experiments in different projects, i-Log has been downloaded by ~1000 unique users in four countries. Since it deals with a lot of data about the person, we have to make sure she is well informed about what the application does, which kind of data it collects and for which reason. Moreover, we need to ask the user to explicitly grant i-Log the permissions to collect data from specific (critical) sensors of the smartphones, in compliance with the latest regulations. All these elements are managed by an installation procedure we designed and implemented. What follows is the description of each step of the procedure.

Figure. 5 - i-Log on the Google Play Store

7.1.1 Experiment selection
This is a mandatory step that needs to be performed immediately after the app has been downloaded or just in case an experiment is finished and the user wants to enroll again in another one. In the current version of the application, the user is required to insert a 4 to 6 digit code that identifies the experiment. The code is provided to the user during the enrollment phase managed by the Municipality of Trento. Once the code is inserted, the application connects to the server where all the experiments are stored and downloads the information of the selected one, such as the text of the informed consent, the list of required permissions, the questions of
i-Log

the profile and the configuration file. Having all the experiment info stored in the backend allows us to have full control and flexibility: for example, a user with an older version of the application can still participate in new experiments without updating.

7.1.2 Login with the user account
The data are stored anonymously on the server in compliance with GDPR using a unique identifier. However, in some limited situations, e.g., assigning prizes or contacting users for feedback, we need to be able to connect with them. For this reason, we ask the user to login into i-Log using her account on the smartphone. With this approach, the user registers with her email address that is saved in the database. The tuple \(<email, unique\ identifier>\) is stored in a dedicated disambiguation table that can be accessed only by the data controllers (and not the data processors that instead can only access the anonymized data). The process has been simplified as much as possible, and the user is prompted with a window showing the accounts logged in on the phone and the user just has to select the ones preferred.

7.1.3 Informed consent form (ICF)
The informed consent is a very important document that provides the user with information about its rights as a data subject, and it is mandatory under current GDPR regulation. In i-Log, every project has its own informed consent form because of the different characteristics of each pilot. Generally speaking, the ICF provides the following information:

- A general description of the project and its objectives
- The type and amount of data collected
- The parties involved in the data treatment and their role, e.g., controllers and processors
- How the data will be collected and processed
- Where the data will be stored and for how long
- How the data will be disseminated

The ICF comes with a dedicated box to be checked to ensure that the user has understood and authorizes the data processing for the experiment.

7.1.4 Grant permissions
On modern smartphones, the vendors require the user to manually grant to single applications the permissions to collect a certain type of data. This also depends on the version of the operating system, in fact, the trend is always to put more constraints and put the user in a better position where he has to grant as many permissions as possible. i-Log is an application developed for Android smartphones and since operating system version 6.0 (October 2015), it also requires to provide these so-called runtime permissions\(^4\). For this reason, we created a dedicated procedure that visually drives the user through the different permissions, explaining which data we collect and for which reason. Depending on the experiment and consequently on the type of sensors used in it, the number of permissions prompted to the user can vary. As of now (Android 8.1, October 2018), if we want to collect

\(^4\) https://developer.android.com/guide/topics/permissions/overview#permission-groups
i-Log
data from all the available sensors, we need to ask ten different permissions to the user, which are as follows:

- permissions.put(battery), i.e., permission to whitelist the app. This only works on stock Android roms.
- permissions.put(fineLocation); i.e., permission to collect GPS
- permissions.put(wifiPermission); i.e., permission to upload data through WiFi
- permissions.put(notification), i.e., permission to access notifications from apps
- permissions.put(audio), i.e., permission to record audio
- permissions.put(usageStats); i.e., permission to collect analytics from the phone usage
- permissions.put(drawOnTop), i.e., permission to record the user interactions with her screen
- permissions.put(contacts), i.e., permission to access the user's contacts
- permissions.put(phoneCalls), i.e., permission to access the calls made and received
- permissions.put(sms), i.e., permission to access sms sent and received

The procedure works as follows: the application shows the user one popup window that describes what the permission we are requiring is, why we are asking it and other information; immediately after the actual window to grant the permission is presented. After all the permissions have been granted, a final summary is presented to describe how many permissions the user granted. Additionally, not all the permissions are the same: we can define if permission is mandatory or optional. In establishing the procedure, the University of Southampton helped with their expertise in the field.

7.1.5 Fill in a questionnaire to profile the user
One requirement from the business cases was to collect socio-demographic information about the users. The main reason for this, specifically in business case two, is to understand where the data comes from and consequently to be able to produce results that are both reproducible and also comparable with others. Specifically, in BC #2 we have one use case that is about computing the modal split in the city of Trento. Previous measurements in the past were done with paper-based surveys on a sample of the population. Asking the citizens directly in i-Log to spontaneously provide some socio-demographic information helps in comparing the results to the previous study and repeating the measurement in the future. To collect the socio-demographic, the user is required to answer multiple choice questions defined by the experiment.

7.2 User interface
i-Log was initially designed as a tool for computer scientists and researcher used to collect personal big data in terms of sensor streams and user feedback [17]. Since the audience in QROWD is very different, i.e., someone that is not an expert but rather ordinary citizens of different ages, we put a lot of effort into completely redesigning the UI/UX. We had to implement many different functionalities as explained in the other sections of this document, and consequently adapt the user interface. After an evaluation of a population of students at the University of Trento,
we detected that the application was too complex to be used and was providing too much unwanted information. This last element was causing the user to be concerned while using i-Log because she perceived the app to collect much more information. For this reason, we went through a process that led to the removal of the whole user interface for the data collection module of the application. However, since we had to inform the user about the fact that i-Log was collecting data about her, we decided to create an always visible notification as shown in Figure 6. i-Log notifications. This last element was causing the user to be concerned while using i-Log because she perceived the app to collect much more information. For this reason, we went through a process that led to the removal of the whole user interface for the data collection module of the application. However, since we had to inform the user about the fact that i-Log was collecting data about her, we decided to create an always visible notification as shown in Figure 6. i-Log notifications. In this way, we do not affect the way the user uses her phone (for example drawing on top of other applications, among others), but at the same time, we let her be informed of the data collection process with a straightforward and clear UI element. This aspect is critical especially since the adoption of GDPR in May 2018 and its stress on the importance of transparency and users' right to be informed about their data processing.

Another important aspect related to GDPR that we had to tackle is about the user being always in control of her data and of the data collection process. For this reason and to facilitate the user as much as possible, we decided to put all the controls of the application directly in the notification as buttons, as shown in Figure 6. i-Log notifications. From those, the user can stop the collection process whenever she prefers to. Additionally, she can enter the ‘Settings’ and ‘Contributions’ menus as shown in Figure 7. i-Log menus. The former allows the user to manage the account, manually synchronize the data, manage the permissions and reset the application to enroll in another experiment. The menu has been kept as simple as all the other elements of the UI. The ‘Contributions’ instead allows the user to manage his contributions, as explained in Section 5. User’s contributions module. In this regard, we created separated notifications shown to the user once each of them is sent to the device and require the user to act.
**Figure 6. i-Log notifications**

**Figure 7. i-Log menus**
8. Similar Applications

Exploiting smartphones for understanding user behavior has recently gained traction thanks to their pervasiveness, ease of use and sensing capabilities. Such features make smartphones the ideal tool for travel surveys and fields such as ubiquitous computing [4] and participatory sensing [8] have been developing several tools for annotating sensor data collected via smartphones or other ubiquitous devices. The first example is MyExperience [6], which is an open source mobile data collection tool developed for Windows Mobile devices (including PDAs and mobile phones). It combines sensing and self-reports to collect both quantitative and qualitative data on human behaviors and activities in the field. MyExperience is based on a three-tier architecture of sensors, triggers and actions; triggers use sensor event data to launch actions conditionally. With respect to i-Log, it is only designed to work on Windows phone, and it is used primarily in more psychological experiments. More recently, [9] proposed an annotation system using the multi-sensory stream for daily activity. It segments each day only in a small set of meaningful events which the user has to annotate with multiple tags categorized by activity, place and people, e.g., eating in a restaurant with friends. Unlike i-Log, the system has only been evaluated by one volunteer. [10] proposed the idea of "mission", i.e., a sequence of selecting activity class and device position as well as performing the activity via a smartphone application. Over 35,000 activity data were gathered from more than 200 users over 13 months. However, only one type of sensor data was collected, i.e., the accelerometer of the participants’ smartphones.

A more related work in terms of travel surveys is the Future Mobility Survey (FMS) [3], which is a smartphone-based prompted-recall travel survey that aims to support data collection initiatives for transport modeling purposes. It is currently now being deployed as part of Singapore’s latest household travel survey, with an expected sample of 1000+ users. With respect to i-Log, this study points some potential issues that i-Log addresses by design such as the wide variety of sensor capabilities in smartphone models and privacy pitfalls.

In addition to research done in the field of travel survey, there has been an increase in commercially available tools specifically for doing travel surveys, such as MEILI, Smartmo [1], Peacox, SenseDAT, TRavelVU, and Modalyzer. Although these tools were developed in the span of roughly ten years, they are fundamentally very similar. In fact, they all provide the functionalities of trip segmentation, and they infer travel modes, purposes, and destinations or allow users to manually correct their trips and related information, e.g., means of transportation used.

As [13] notes, these systems have been developed internally or as cooperation with industrial partners, but ultimately without revealing any implementation details. The lack of transparency on the implementation details as well as the high cost of implementing a system based on a published architecture of a prototype represents a significant obstacle. Instead, i-Log, moves in a radically different direction, as proven by this deliverable, and also aims at being as adaptable as possible with respect to the needs of the application scenario, e.g., by enabling challenges or in complex crowdsourcing tasks as detailed in D3.2.
9. **Demonstrator description**

This section describes the workflows that we are showing as demonstrators in the videos attached to the deliverable:

1. **Installation procedure (Video Demo: D2.4-installation.mp4):**
   1.1. The user accesses the Google Play Store and searches for i-Log
   1.2. Once the user finds it, downloads it on her phone
   1.3. The user then can open i-Log and begin the login procedure
   1.4. To begin, the user must input the code to access the experiment
   1.5. The user is greeted by a welcome message that presents the experiment the user is participating to
   1.6. The first step is the login: the user can choose any of her Gmail accounts
   1.7. The second step consists in reading the ICF (Informed Consent Form) and authorizing the data processing by toggling the "Autorizzo" (I authorize) button
   1.8. The third step consists in choosing the permission to be given to i-Log: the permissions are as follows:
      - *Synchronization*: it allows i-Log to synchronize data collected over WiFi
      - *Battery*: it allows to remove any battery restriction and work in the background (a guide to do that is included in the permission request)
      - *Position*: it allows i-Log to collect the position of the user
   After this step, the permission selection is complete
   1.9. The fourth and final step is the profile creation, which collects the following information about the user:
      - Whether the user is a commuter or resident
      - Her occupation: worker, student, or other
      - Her gender: male or female
      - How many cohabitants she has
      - Whether the user has a driving license
      - How many vehicles are available in the household
      - The preferred mode of transportation
      - Whether the user has WiFi available
      - Whether the user suffers from any disability
   1.10. After completing this step, the i-Log setup is done

2. **i-Log usage (start/stop the application) (Video Demo: D2.4-usage.mp4)**
   2.1. To start i-Log, the user must find it among the applications and click the i-Log icon
   2.2. A notification appears, telling the user that i-Log is running. If the user clicks the drop-down menu, three buttons appear:
      - *Stop*, to stop the application
      - *Impostazioni* (Settings), to access i-Log settings
2.3. To access the settings, the user clicks on Settings
2.4. The settings menu is divided into three sections:
   - **Login**: Allows the user to login/logout
   - **Data**: Shows the user if there are any data to be synchronized with the server
   - **Applicazione (Application)**: Allows the user to manage settings concerning the application, such as
     - **Permessi (Permissions)**: allows the user to manage permissions
     - **Riavvio dell’applicazione (Reset)**: allows the user to reset the application
     - **Versione Applicazione (Version)**: it shows the version of the application
2.5. To turn off i-Log, the user simply need to press Stop in the drop-down menu

3. **Hybrid workflow: modal split (Video Demo: D2.4-hybridworkflow-modalsplit.mp4)**
3.1. From the notification, a new task is shown to be available to the user, from 2 to 3
3.2. The user scrolls down to **Rilevazione/Domande (Tasks)** and clicks on **Complete (To be completed)**
3.3. The user is presented with three possible tasks
3.4. By clicking on the last, the user is shown a map with a point where the user was detected to be at a certain time and that is taken to be the beginning of a trip. The user can confirm whether this is the case and in she chooses *Ho effettuato un viaggio, ma non ha avuto inizio in quel luogo* (I started a trip, but not from this point)
3.5. The user is then asked to pinpoint the correct place on the map, which she can do by moving around the map and then clicking on the map
3.6. The system has also detected that the user stopped somewhere after beginning the trip, which the user can confirm. The user does by clicking *Sì, ho raggiunto la fine del mio viaggio* (Yes, I reached the end of my trip)
3.7. The user is then asked to specify the means of transportation used during the trip. In this case, she chooses *A piedi* (By foot).
3.8. A message thanks the user, who can then exit the app.

4. **Hybrid workflow: mobility infrastructure**
As described in D3.2, there are two types of hybrid workflows: freeroam and validation. The former consists in finding a bike rack, take a photo and define its type (wheel- or chassis-block or other), while the second consists in providing information about bike racks, with the difference that the users must confirm whether a bike rack is still in the place from the map. The "D2.4-hybridworkflow-freeroam" video shows the general procedure to access the contributions, which is as follows
4.1. From the dropdown menu, the user clicks on Contributions
4.2. She click on **Disponibili (Available)** to check whether a challenge is
available, which is then notified by a message "Ci sono 1 sfide disponibili" (There is one challenge available)

4.3. She clicks again on Disponibili, which shows one freeroam type challenge

4.4. Once she clicks on the challenge, a new page appears with a map of the Trentino region and a small description of the challenge

4.5. The user goes back, and checks the ongoing challenges, which are two: freeroam and validation

4.6. Once the user chooses the freeroam Trova le rastrelliere per biciclette in città, she is presented with a page with the time limit for the challenge

4.7. After clicking, the user is presented with a welcome message

4.8. The user can then take a photo of his surroundings

4.9. The user is then asked to define which of the three possible types of bike racks is the current one the total number of spots, and the available number of spots

4.10. The "D2.4-hybridworkflow-validation" video shows that the procedure is the same for the freeroam

4.11. Once the user begins the challenge, a message explaining how to help the project is provided to the user

4.12. After the message, i-Log looks for the user's position and confirms it

4.13. A message appears asking the user whether the bike rack is still there

4.14. The user then has to take three consecutive photos, and may have to grant i-Log permission to do so

4.15. The user is then presented with the three possible types of bike racks as in the free roam, the total number of spots, and the available number of spots

10. Conclusions

In this deliverable, we presented i-Log, a mobile application that collects both sensor and user-based inputs in crowdsourcing campaigns. It has been designed to be adaptable, transparent, unobtrusive and enables smart sensing strategies to preserve battery life. All these features make i-Log the ideal tool for QROWD as it is deployed in the business cases of the Municipality of Trento to involve the citizens in crowdsourcing campaigns to improve the mobility of the city of Trento.

We provided the specifications of i-Log, detailing not only its general architecture but also its role within the workflows for challenges in mobility and the understanding of the modal split. We also compared i-Log with other applications for travel surveys, underlying how their main pitfall is the fact that they are commercial endeavors, which limits their adaptability and reusability. Finally, we also provided a description of the demonstrator to help the reader understand it.
### ANNEX 1 - LIST OF AVAILABLE SENSORS

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Linear Acceleration</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Gravity</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Rotation Vector</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Magnetic Field</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Orientation</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Temperature</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Atmospheric Pressure</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Humidity</td>
<td>20 times per second</td>
</tr>
<tr>
<td>Proximity</td>
<td>On change</td>
</tr>
<tr>
<td>Position</td>
<td>Once every minute</td>
</tr>
<tr>
<td>WIFI Network Connected to</td>
<td>On change</td>
</tr>
<tr>
<td>WIFI Networks Available</td>
<td>Once every minute</td>
</tr>
<tr>
<td>Bluetooth Device Available</td>
<td>Once every minute</td>
</tr>
<tr>
<td>Bluetooth Low Energy Device Available</td>
<td>Once every minute</td>
</tr>
<tr>
<td>Detect Incoming Calls (No audio)</td>
<td>On change</td>
</tr>
<tr>
<td>Detect Outgoing Calls (No audio)</td>
<td>On change</td>
</tr>
<tr>
<td>Detect Incoming Sms (No text)</td>
<td>On change</td>
</tr>
<tr>
<td>Detect Outgoing Sms (No text)</td>
<td>On change</td>
</tr>
<tr>
<td>Running Application</td>
<td>Once every 5 seconds</td>
</tr>
<tr>
<td>Screen Status [ON/OFF]</td>
<td>On change</td>
</tr>
<tr>
<td>Feature</td>
<td>Update Frequency</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Flight Mode [ON/OFF]</td>
<td>On change</td>
</tr>
<tr>
<td>Battery Charge [ON/OFF]</td>
<td>On change</td>
</tr>
<tr>
<td>Battery Level</td>
<td>On change</td>
</tr>
<tr>
<td>Doze Modality [ON/OFF]</td>
<td>On change</td>
</tr>
<tr>
<td>Headset plugged in [ON/OFF]</td>
<td>On change</td>
</tr>
<tr>
<td>Audio mode [Silent/Normal]</td>
<td>On change</td>
</tr>
<tr>
<td>Music Playback (no track information) [ON/OFF]</td>
<td>On change</td>
</tr>
<tr>
<td>Audio from the internal microphone</td>
<td>10 seconds per minute</td>
</tr>
<tr>
<td>Notifications received</td>
<td>On change</td>
</tr>
<tr>
<td>Touch event</td>
<td>On change</td>
</tr>
<tr>
<td>Cellular network info</td>
<td>Once every minute</td>
</tr>
</tbody>
</table>
REFERENCES


