Data collection using a smartphone app at Utrecht University

Peter Lugtig
Department of Methods and Statistics, Utrecht University
22 May 2019
p.lugtig@uu.nl

An infrastructure for collecting sensitive data through mobile phones
This report documents the main results of a project that followed a grant by UU research IT awarded in 2018. The project centered on developing an infrastructure at UU for collecting mobile phone app data. Mobile phones are useful for collecting data in diverse fields: e.g., health sciences, geosciences and social sciences. Apps are the main tool for this; they are small pieces of software that record data from sensors and send the raw or aggregated data to an external database.

Modern smartphones contain about 20 different sensors (e.g., measuring location, movement, the environment and use of the phone). Many ad-hoc apps, built for specific research projects, are already available. The problem with such apps is that they are tailored to a specific project and cannot be easily used in other research projects. Commercial apps are often more generic and easily adaptable, but expensive. More problematic is that commercial apps store data on a remote server (often outside the Netherlands), and are outside the researchers' control. New privacy regulations (GPDR) guide to a large degree how data can be stored. Although it is technically possible to store mobile phone app data in any cloud environment, the goal of the project was to develop an infrastructure so that smartphone app data could be stored locally at Utrecht University.

Key results
The project ran for a 14 month-period and ended in February 2019. During this time, a back-end infrastructure was set up at Utrecht University, and tested using an open-source smartphone app that was developed by researchers from Utrecht University and Statistics Netherlands. The back-end infrastructure is set up in such a way, that it can easily deal with different types of data collected through different apps. Although not part of the original research plan, we also set up a portal so that access to data can be organized safely and efficiently. Moreover, although again not part of the original project plan, the infrastructure is set up to be scalable. This report will show some results of a small-scale test that was part of the original project plan. We also did scalability tests to see if the infrastructure would be able to handle large data volumes. Although it would be important to perform these tests again if future projects want to use the infrastructure, the tests showed that the infrastructure can easily deal with large volumes of text-data. The infrastructure was not tested for large volumes of picture and video data, however.

The remainder of this document describes in more detail the infrastructure, how it can be used in future by researchers at Utrecht University. First, we describe in more detail the app we used for this project.

Smartphone apps
Smartphone are becoming an increasingly popular tool for data collection. Most people carry their smartphone around wherever they go, and use the smartphone at least several times a day for all kinds of different tasks. Smartphones allow for different methods of communication (phone, messaging, bluetooth, Internet), and nowadays contain more than 20 different sensors (e.g. a microphone, camera,

temperature, light, gravity, movement, location sensors) which could potentially be used to collect both information on the people using these smartphones and their environment.

One of the most popular pieces of smartphone software are apps. Apps are (usually small) pieces of software built for a specific task. Apps can be used to answer questions, but can also communicate with the sensors on a phone and allow for notifications. Within the social and health sciences, smartphones are increasingly popular as a research tool.

The example app used in this project was the TABI app. This app was developed open-source by researchers of Utrecht University and Statistics Netherlands. The primary goal of the app which was tested was to document travel behavior by passively recording GPS data without negatively effecting battery life. Apart from the passive data, the app also provides the opportunity to ask questions in the app.

The pictures below show an example. The left panel shows the travel behavior of one person on a day. The "diary" of stops and trips is automatically produced by processing the GPS, WiFi and cell phone data on the phone and use of an algorithm to detect stops and trips. By clicking on one of the orange pencils, users are directed to a new screen showing the location visited (middle panel) or trip made (right panel). At each screen questions can be asked. The app also provides the opportunity to send push notifications.

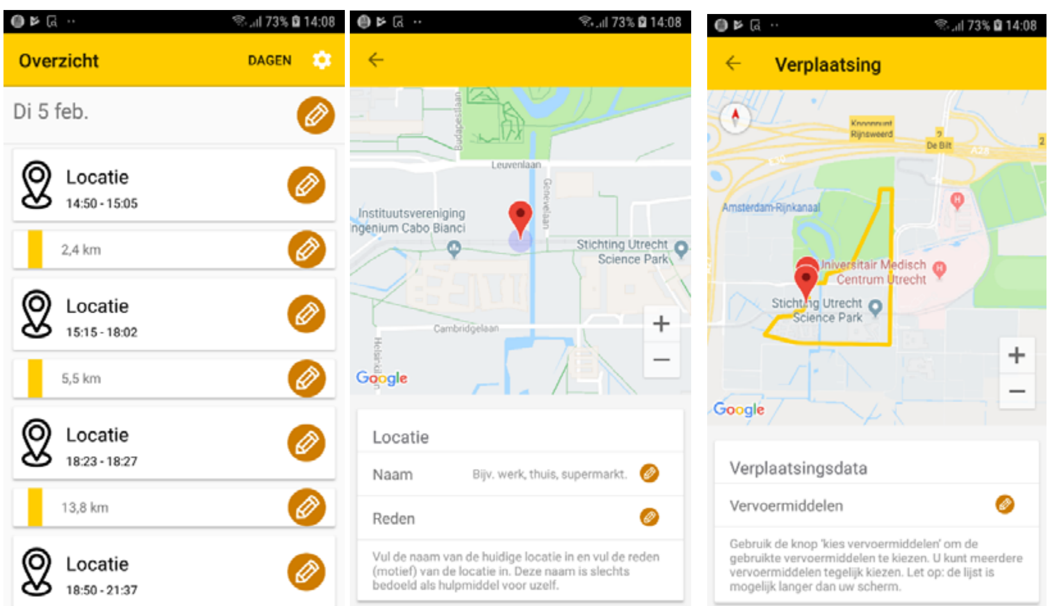

Figure 1: examples from the TABI app that was used to test the infrastructure

Front-end and back-end infrastructure

The Travel App System is comprised of a front-end and a back-end. The front end consists of the TABI Travel App, which collects the location data, resolves stops and tracks, and exposes these to the user for annotation purposes. Both the raw location data and the resolved data is stored locally in an SQLite database on the mobile device. The backend consists of an API written in GO that performs the data ingestion and transformation and a Postgres database that ultimately receives and stores the data. See Figure 2 for a schematic overview of the back-end infrastructure. The app relies on an API to process the raw sensor data, and an API to transfer data from the mobile device to a database. Depending on the purpose of the app, it is relatively easy to either add other types of sensor data (i.e. gravitational, light, movement), make changes to the kind of questions asked in the app, or change the way data are transferred. In 2019 the infrastructure will be extended to comprise a module that can be used to send data from the database back to the device. In practice, this will enable sending dynamic messages (push notifications based on data received), or the possibility to process data in a database and send processed data back for further annotations or follow-up questions.

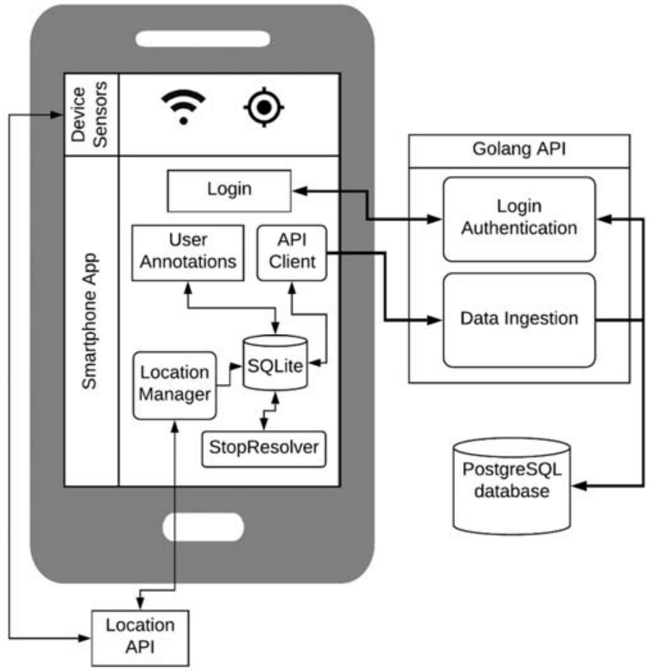

Figure 2: Back-end infrastructure for app data collection.

The front-end infrastructure

In order to be able to deploy equivalent algorithms to both Android and iOS versions, the app used in the test was developed in C# using the Xamarin framework. This framework provides compilation to Intermediate Language which is Just-in-Time compiled to native assembly on Android devices, and Ahead-of-Time compilation into native ARM assembly code for the iOS build. The application was developed Open Source and hosted on a publicly accessible collaborative code repository in order to facilitate distribution and address potential privacy concerns. See for the current, and future updates: https://gitlab.com/tabi/tabi-documentation. Layout, logos etc. can all easily be amended by copying and adjusting the source code.

The back-end infrastructure at Utrecht University

It is relatively straightforward to link a different app, other than the TABI app to the backend of the infrastructure. The infrastructure is hosted at the servers of the department of ICT and Media, part of the faculty of Humanities. They have organized licenses for all the software that is necessary for the backend infrastructure, and offer scalable hardware that was tested for large volumes of data as part of this project. The infrastructure is cloud-based, meaning that maintenance on the server and data-access is easy. Security is an issue for all cloud-bases services. However, because data are stored at servers of the UU, it means that sensor data are stored safely using the security protocols of the general UU infrastructure.

UU on the move portal
As part of the project a portal was built with which researchers can access their own "projects" stored on the App-servers.

Figures 3-6 below show the interface of the portal which is called "UU on the move", available at https://tabi.acc.fss.uu.nl/ . Users need to acquire a username and password to gain access to the portal and servers. Projects are in principle assigned to one main user, but the portal allows for projects to be shared among multiple users. This implies that different researchers would be able to access the same dataset.
It is necessary to contact and collaborate with Sandor Spruit, IT-developer at the Faculty of Social and Behavorial Science, to gain access to the portal, and to link a particular app the backend and portal (for contact details: see end of this document).

Figure 3: "UU on the move " portal for access to UU app server

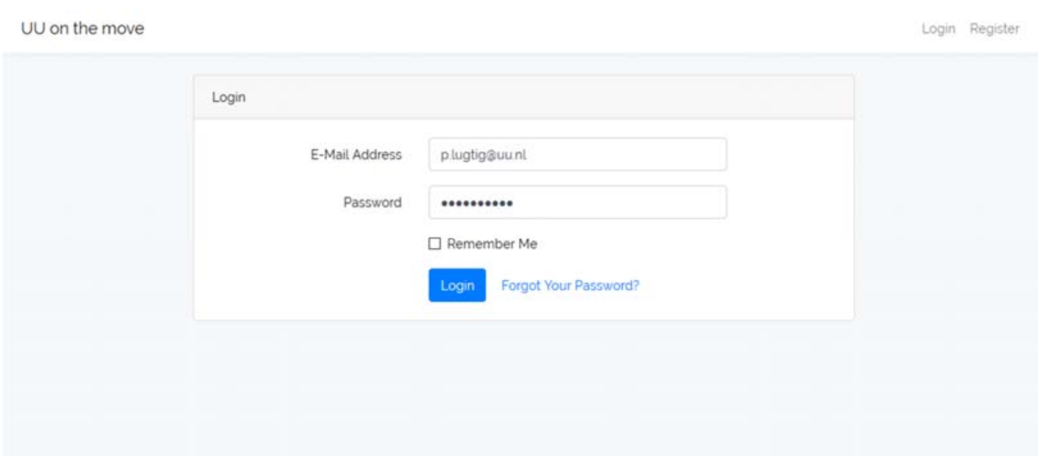

Figure 4: Login to the portal

After registering and logging in (figures 2 and 3) users are presented with an empty portal, in which they can add projects ("measurements"). Generally, users are assumed to add a new measurement for every new research project. In Figure 4 below, a user can add a project by clicking the "add measurement" button. As an example, an existing project "test" is also shown. In Figure 5, a user is asked to provide a name "description" and number of persons for every research project. Data for every research subject or participant is stored as a separate database, so it is important that a user provides a sufficiently high number of users to accommodate the expected number of participants in the research project.

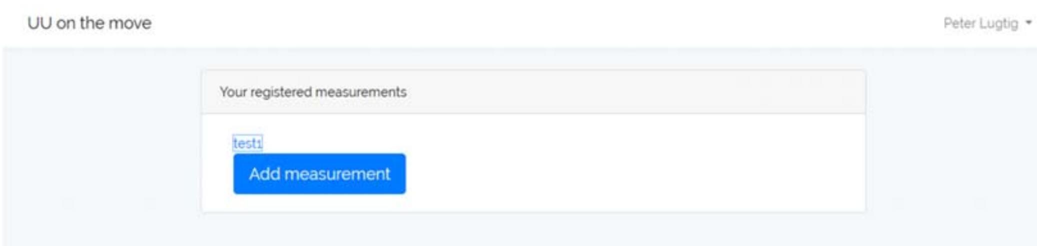

Figure 5: Overview of data collection projects for user "Peter Lugtig"

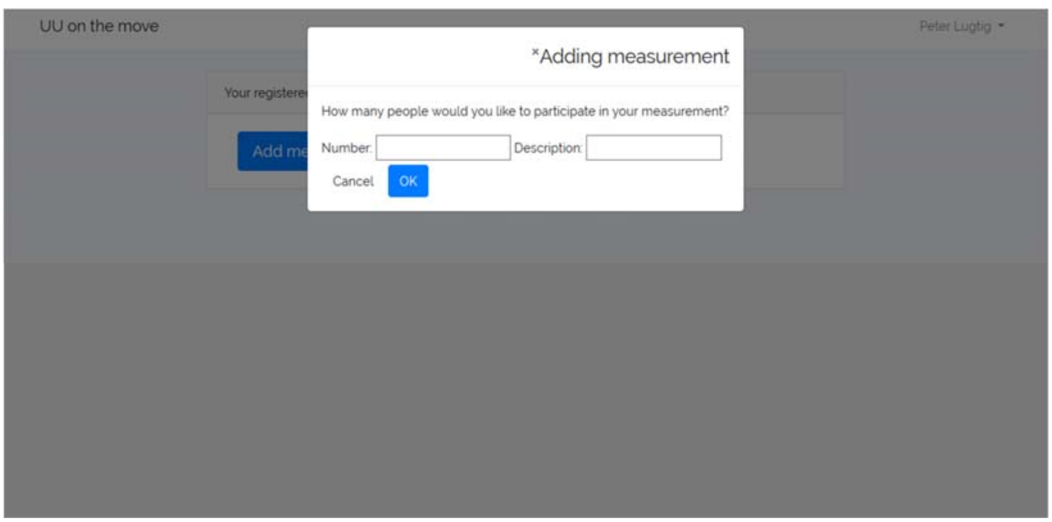

Figure 6: Adding a new research project

After a new project is established, it is necessary to link a front-end (an app) to the back-end at the UU. As long as the app uses an SQLite design to store data on the phone locally, this is a straightforward process. If another system is used, more work is required.

After the front-end and back-end are connected, data can be collected. Figure 6 shows an example of a small test project that was carried out. Here, data were collected for 10 participants. All data are stored for every respondent in a PostgreSQL database. One of the advantages of the portal and the system setup is that it is very easy to access, delete, or edit data for a particular individual should this arise. The structure of the database in which every person's data are stored is flexible. In the TABI-app which was tested in this project, GPS location data, trips, stops, and answers to survey questions were stored in separate tables within the PostgreSQL database. It is up to the user however to decide what data structure suits the project best and alter the default database structure. Respondent Ids are automatically generated.

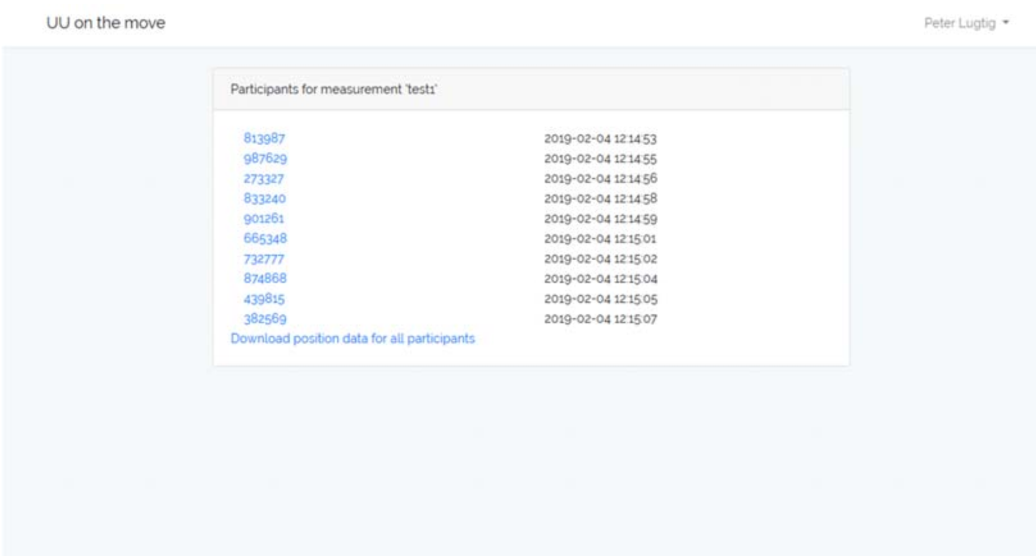

Figure 7: Overview of data for 10 research participants. Data for every individual can be accessed (and downloaded) for every participants individually, or downloaded as one big file.

Using the infrastructure

The infrastructure has been set up to be flexible. It can accommodate different types of apps, and so can easily handle apps in which different types of data are to be collected. PostgreSQL is used as the default framework for storing data. Although it is possible to change the backend-infrastructure and use a different database, this will require more work.

In any case, it will be necessary to collaborate with Sandor Spruit (a.g.l.spruit@uu.nl). He maintains access to the portal, and can ensure that the backend infrastructure at the UU is connected to every device. It is important that contact is established several months before fieldwork is planned to start.