GSBLapp: Tailored Chemical Substance Information for Arbitrary On-Site Usage on Mobile Devices

Stefan Barthel†, Tristian Pfofe†, Christian Rößler†, Marius Bozem†
†Federal Environment Agency of Germany
Unit IV 2.1, 06844 Dessau, Germany
Email: stefan.barthel@uba.de
‡Otto-von-Guericke-University, 39106 Magdeburg, Germany
{tristan.pfofe, christian.roessler, marius.bozem}@st.ovgu.de

Abstract—Technical assistance for the provision of substance information for firefighters, specialist consultants and rescue services replaces more and more conventional, paper-based solutions. Recently, the Federal Environment Agency of Germany developed a mobile application for smart phones and tablets specialized to the needs of firefighters. This so called GSAapp allows users to efficiently and instantly search offline for information on 16000 substances. We realized soon that there are additional possibilities of use for chemical substance data and relevant chemicals-related legislation on mobile devices. However, we noted that required techniques and functionality are similar for such applications. Therefore, we developed one mobile application that is customizable for different scenarios. We present the approach and our results in this paper.

I. INTRODUCTION

The Joint Substance Data Pool of the German Federal Government and Federal States (GSBL) [3] is a continuously maintained, uniform collection of validated information on chemical substances, held in a central national database. The mission of the cooperation is to provide public authorities and the general public with reliable, topical chemicals information in support of all activities to prevent and avert danger and to protect humans and the environment. Records contain information on physical and chemical properties, on ecotoxicological and toxicological parameters and on environmental fate. They provide descriptions of environmental hazards, health dangers, fire and other technical hazards and give information on relevant chemicals-related legislation. Responsibility for operating the system lays with the German Federal Ministry for the Environment, Nature Conservation and Nuclear Safety and the Environmental Ministries of the Federal States (Länder). The cooperation was established through an administrative agreement in 1994.

The Hazardous Substances Rapid Information System (GSA) is a component of the GSBL, which contains relevant data specifically for rescue services. In 2014, a view for displaying on PCs was developed especially for firefighters, specialist consultants and rescue services. Due to this successful project, the Federal Environment Agency of Germany (UBA), additionally, developed a mobile version specially tailored to the needs of firefighters, called GSAapp [4]. This mobile version allows different emergency forces on the spot and in case of crisis or disaster to efficiently search offline for information based on 16000 substances. GSAapp can be installed on smart phones and tablets with the operating systems Android and Apple’s iOS.

While developing GSAapp, we realized soon that there are additional possibilities of use for chemical substance data and relevant chemicals-related legislation for on-site application on mobile devices such as consumers and health protections, food control and occupational safety. Since, we assume some more use cases and we do not want to develop one mobile application for every scenario as well as practicability should be tested before publishing, one customizable mobile application (GSBLapp) was developed and tested as a prototype.

The process of realization is pictured in the research presented here.

II. REQUIREMENTS

Since, we already have experience in developing an application for mobile devices and our expectations for the GSBLapp prototype are known, deduction of requirements and identification of potential issues for a similar mobile application are easier to accomplish. Moreover, we already have an intuitive usable mobile application (GSAapp) and existing users got used to it. We do not want them to relearn proper handling.

1) Usability and surface feel should be similar to the GSAapp and
2) both screen modes (portrait and landscape) have to be supported.

Even though the common interface standard for GSBL is determined to the proprietary, self-invented SSF format (Schnittstellenformat),

3) data should be loaded in XML format,
4) one data package can be loaded simultaneously and
5) data package are replaceable.

Currently, the major part of GSBL’s substance data is only accessible to public authorities due to publication rights. Hence, GSAapp as well as the GSBLapp are not made freely available.

6) GSBLapp and GSBL’s data packages need to be online available via authentication.
Since, we want to create a prototype where cost burden should be reduced to a minimum and open standards are favored, we determine to use

7) Android as development platform and
8) allow usage on tablets as well as smart phones.

Due to the fact that we want to develop only one mobile application which can be used for all application scenarios based on GSBL’s chemical substance data, GSBLapp needs a

9) freely customizable search mask and
10) user-group-specific result presentation as well as
11) storage capabilities for any substance datasets (schema-free).

Finally, we have to emphasize that all mentioned requirements are results arose by general needs and none of these issues is key user specific. Since, their individual requirements on data presentation shall be regarded by the configuration designing process within the GSBLapp.

III. BASIC CONCEPTION

Derived from requirements mentioned above, objectives for the development of the GSBLapp are clearly framed. We want to develop one mobile application which can be used for all application scenarios based on GSBL’s chemical substance data. GSBLapp also needs to be configurable as well as provide information for search requests. This means in particular that we will have two kinds of user roles, respectively, distinguishable user intensions.

- **Administrator role** configures search mask and result presentation according to field of application and key user interests.
- **Key user role** utilizes predefined configuration to search and to obtain results for a search request.

Since, not every key user has technical skills, both role may be taken by one person, but do not have to. Therefore, we decided not to mix up both intensions and hide administrative functions as far as possible in order to encourage non-technical people (key users). For further use, we call the predefined configuration to search and to obtain results for a search request as **frontend** and the configuration interface as **backend**.

IV. IMPLEMENTATION PROCESS

Because Android was preselected as development platform, we only needed to choose an appropriate development environment. Finally, we decided to use Eclipse [1] with Android Development Tool (ADT) [2] plugin, since, it is free to use and we already had some experience with Eclipse. Moreover, ADT extends the capabilities of Eclipse in a way that we can easily create an application user interface, add packages based on the Android Framework API and export .apk files for demanded distribution purpose.

Most challenging was to identify an appropriate database solution, because commonly used and by default included file-based database SQLite (sqlite3) [5] has a substantial shortcoming: it is relational. But the majority of our substance datasets does not have a rigid data layout. Most substance records are schema-free and can be arbitrary shaped. Only after storing all the data, the user (in administrator role) has to decide which attributes should be searchable and visible on the detailed substance presentation view. Therefore, we identified as an suitable alternative CouchBase Lite [7]. It is a fully functional, lightweight, native, embedded NoSQL database (JSON [6] based) that can work standalone and lives...
locally (offline) on mobile devices. In addition, CouchBase Lite is well supported and has a detailed installation manual including Couchbase Lite project instructions for Eclipse, which reduces set-up time significantly.

As demanded, we split the GSBLapp into two functional distinct operating modes: frontend and backend. After the application starts, GSBLapp always loads the frontend and present a predefined search mask to the user (see Fig. 2). At this point, search requests can be submitted and data is afterwards displayed in a preconfigured result presentation. In case no configuration exists a message is prompted with the instruction to set up a configuration before work may continue. Backend can be easily reached by first clicking menu-button on the respective device. Subsequently, option "settings" will be displayed which is the entry point for the backend. By simply quitting any configuration process at every time the user is taken back to the frontend (for key users) by default. All changes are automatically stored and applied to current session as well as for further use. One configuration always consists of at least on tab that holds again a minimum of one substance attribute. Names of attributes are included in the current loaded data package whereas configurations and tabs can be named individually. Figure 3 depicts an example of a configuration "fire" which has three tabs called "Dangers", "Protections" and "Other". Each tab itself contains of several substance attributes. To be able to search efficiently, database search indexes are set immediately after determination of search attributes.

V. CONCLUSION AND FUTURE WORK

So far, our developers did amazing work and the prototype is fully functional, even though, there is still potential for improvement. However, differentiated by subject area and evaluated by experts (chemists, biologists, firefighters, etc.) all enforced requirements have been fulfilled and the usability is intuitional. Despite that we faced several issues while implementation, development time was hardly exceeded. Nevertheless, we recognized afterwards some improvable circumstances. Beside performance optimizations, we have to implement a more optimal search algorithm, since at present, we only do a simple text comparisons without any wildcard options. Furthermore, loading of data packages is currently not encrypted. Another point of criticism is related to the storage configuration. Prospective, configuration should be exchange-able between different devices and, therefore, a configuration handling and storing on the mobile device’s file system has to be implemented. However, given the fact that a prototype was demanded, we are very satisfied with the outcome.

ACKNOWLEDGMENT

The first author would like to thank Otto-von-Guericke-University for support and developers Tristan, Christian and Marius for the truly amazing work they have done in such a short time.

REFERENCES