



Report – Time Use App “STAT_time”

Grant Agreement Number: 836897-2018-AT-INNOV

Statistics Austria

Vienna, December 2019

For further information please contact:

Project Manager and Time Use Survey Austria:

Sonja Ghassemi-Bönisch
Guglgasse 13
1110 Vienna
Tel.: +43 1 711 28 - 7103
sonja.ghassemi-boenisch@statistik.gv.at

Software development:

Leonhard Fürst
Guglgasse 13
1110 Vienna
Tel.: +43 1 711 28 - 7871
leonhard.fuerst@statistik.gv.at

Friedrich Csaicsich
Guglgasse 13
1110 Vienna
Tel.: +43 1 711 28 - 7182
friedrich.csaicsich@statistik.gv.at

Table of Contents

- 1 EXECUTIVE SUMMARY 6**
- 2 INTRODUCTION 7**
- 3 MEASURING TIME USE VIA APP 7**
 - 3.1 Description of the awarded Grant 7
 - 3.2 Timetable of the Grant 9
 - 3.3 Results and their use 9
- 4 INNOVATION IN THIS GRANT 10**
 - 4.1 Open Source 10
 - 4.2 Authentication..... 10
 - 4.2.1 User credentials..... 10
 - 4.2.2 Authentication mechanism 10
 - 4.3 Internationalization and Localization..... 11
 - 4.4 Multi timeslot filling method 11
 - 4.5 CSPA service catalogue 11
 - 4.6 User Guide 12
 - 4.7 Setup Guide 12
- 5 THE APP – TECHNICAL INFORMATION..... 13**
 - 5.1 Installation 13
 - 5.1.1 Google Play Store / Android Phone 13
 - 5.2 Usage..... 14
 - 5.2.1 Starting the App..... 14
 - 5.2.2 Login 14
 - 5.2.3 Start time 15
 - 5.2.4 Initial list of time slots 16
 - 5.2.5 Details view..... 17
 - 5.2.6 Subjective Well-being questions 21
 - 5.2.7 Completed list of time slots 22
 - 5.2.8 Finish the days diary 23
 - 5.2.9 Repeat steps 5.2.2 to 5.2.7 to finish the second day..... 24
 - 5.3 Finish diary 25

6	SOFTWARE ARCHITECTURE	26
6.1	Requirements	26
6.1.1	Offline operation	26
6.1.2	Online operation.....	26
6.1.3	Local notifications.....	26
6.1.4	Sufficient number of devices	26
6.1.5	Open source.....	26
6.1.6	Authentication standard.....	27
6.1.7	Internationalization and Localization	27
6.2	Target devices.....	27
6.2.1	Operating system versions	27
6.2.2	Market share of supported devices.....	29
6.3	Development platform	29
6.3.1	Native development	29
6.3.2	Web apps.....	29
6.3.3	Hybrid apps.....	29
6.4	Structure.....	30
6.5	Distribution	31
6.5.1	Google Play Store	31
6.5.2	Apple App Store.....	31
7	IMPLEMENTATION DETAILS.....	32
7.1	Technology stack	32
7.2	Activity Diagram	32
7.2.1	Before using the App	32
7.2.2	Using the app.....	33
7.2.3	Filling a time slot.....	33
7.2.4	App server state activity	34
7.2.5	Backend server database activity	34
7.2.6	User activity	35
7.3	Communication Diagram	36
7.4	Implementation details	37
7.4.1	Basic authentication	37
7.4.2	Hashing	37
7.4.3	Internationalization and Localization	37
8	GENERAL CONCLUSIONS AND FUTURE STEPS.....	39
9	LIST OF REFERENCES.....	41

1 Executive Summary

The aim of this project was to internationalise an existing prototype of a Time Use Survey App based on browser-technology that works on mobile devices.

The app-prototype STAT_time was developed 2017 in accordance with the HETUS-Guidelines from 2008 and has two important new features: look-up tables (i.e. testing verbatim mode of data collection with predefined list of activities that will appear after typing some initial word letters) and the possibility for copying information across time slots. There was also the possibility for measuring subjective well-being integrated in STAT_time.

For the TUS-app, the decision has been made to implement a hybrid app using the Cordova framework. Cordova provides a container for each target platform that contains the almost platform-independent web app providing the functionality to the user. Plugins allow access to system functionality like storage and local notifications. The user interface has been developed using the Ionic framework which allows adaptation of the app to the look and feel of native apps thus providing a uniform overall user experience. The app is available for the operating systems Android and iOS. There were some improvements of the App STAT_time done. A two-day-login was created and the App is built with Open Source Software and internationalised.

The structured and CSPA-compliant documentation of this tool should help Eurostat and other countries to compare the function of the tools. The App STAT_time might be used or tested by other interested countries.

2 Introduction

The last time use survey in Austria was conducted in 2008/09. The next European wave of Time Use Surveys will start in 2020 and therefore it will be necessary to enhance the design of the survey to increase data quality and response rates.

2017 with the Grant of Eurostat it was possible to develop a mobile app for time use surveys and introduce a new technology in this survey in Austria. So respondents will have the possibility of filling out the diary in a more convenient and contemporary way.

2019 we adopted this prototype for open-source-requirements and translated it in English so that other NSIs have the possibility to test this app STAT_time.

It is not clear at the moment if Austria can attend the next wave. It depends on the funding of the next time use survey by the ministries. With this app-prototype it is possible to go one step towards the next wave of time use surveys.

3 Measuring Time Use via App

The following chapter gives a description of the awarded Grant with the time table of this project.

3.1 Description of the awarded Grant

Eurostat started in 2017 a project to modernise the data collection in EU/EFTA Member States for completing the Harmonised European Time Use Survey (HETUS). With time use data it is possible to measure unpaid work, household production, including volunteering, accurate working time, gender balance in daily activities, care for children and the elderly, transport time and use, leisure and non-work activities, use of ICT, loneliness and isolation and several elements of quality of life including subjective perceptions.

Like Eurostat states the burden on respondents (e.g. completing diaries) and NSI's (data processing) and the subsequent low and/or decreasing response rates stress the need to improve ways of collecting the data. This includes exploring new tools and sources. Other new technological developments may be expected in the near future, like detection of the activity with sensors (sleeping time, etc.).

The aim of the action is to document an existing tool – a prototype App for measuring time use via smartphone. This TUS-Diary includes all essential features for users to fill out the diary and follows the HETUS-Guidelines. Those are for example the 10-minutes time slot, main and secondary activity, where was the activity done and with whom. Also a wellbeing –function was elaborated and a list with suggested activities after typing it in.

This action should be a contribution to improve the quality and the base for other interested countries for testing. The tool is described in a standardised way that makes it easier for a Member

States to compare functionalities of the tool and to be aware what kind of solutions to be used for modernising the national data collection in a Member State of Eurostat.

Some requirements of the App will be programmed like authentication, internationalization and localization but also we will program the possibility for a 2-day-login integrated in the App and a function to choose more time-slots at one time.

The structured documentation of this tool should help Eurostat and other countries to compare the function of the tools. The developed prototype might be used or tested by other countries.

3.2 Timetable of the Grant

The action started in January 2019. After a detailed conceptual phase the main steps of the action are to modify and punctually improve an existing prototype for measuring time use via smartphone. The conceptual phase aims to find optimal solutions for the upcoming tasks. An Induction report was submitted two months after the start of the work.

Project Task – Description	Target	Status
Detailed conceptual phase Induction Report	M + 2	January- February 2019
Start of programming of the App– modifications: - Integrated 2-day-login - Language configuration, preset in English - Function to fill out more than one time-slot - Programming of authentication-function and of backend-data-structure	M + 9	March – October 2019
TUS App STAT_time	M + 10	October 2019
Documentation Report on the Action, User Guide, SetupGuide, CSPA	M + 11	November 2019

3.3 Results and their use

Following results were allocated from this awarded Grant:

- Induction report: An Induction report that describes the planned work of the next levels of the grant. The report was submitted 2 months after the start of the work.
- Report on Action: The report on the action is submitted by the closing date of the action.
- Including CSPA-Documentation of the tool according to the CSPA-templates provided by Eurostat
- User-Guide in English for other National Statistics Institute, where use and the deployment of the App is described
- Time-use-App-tool, including authentication and backend-data-structure- ready for testing preset in English language with possibility to set it up in other languages

The Documentation (Report, CSPA and User Guide) and the tool will be a good base and fundament for other countries for testing this prototype STAT_time.

4 Innovation in this grant

4.1 Open Source

As other NSI will be able to install, build and run the app on premise, the technology stack of the app has to contain open source libraries/packages and dependencies only. So for developing, building, installing and providing the app on their own, no costs arise for the NSI doing so.

4.2 Authentication

In order to participate in the time-use survey using a mobile device, respondents have to authenticate against the service in order to identify themselves as legitimate participants at the time-use survey. Authenticated respondents are authorized to read and write their time use diary subject to business rules.

Authentication has to be designed for data protection regarding confidentiality and data integrity. Confidentiality means, that sensitive data is not exposed to unauthorized persons.

For data integrity, the service needs to ensure that unauthorized persons cannot corrupt data. These security requirements have to be balanced with usability requirements, especially in the context of mobile device input methods. The authentication process needs to specify requirements regarding user management and the authentication protocol.

4.2.1 User credentials

Credentials allow the service to identify a respondent in order to expose their time-use diary data while denying access to other respondent's time-use diaries. User credentials first need to be issued by the NSI, and then communicated to the respondent using a communication channel external to the service. Finally, a respondent needs to input the provided credentials in order to use the TUS app.

The current solution provides a simple login form where a respondent enters a username and a password which has been mailed to them before together with general instructions regarding the time-use survey. The usability test results suggest that alternate input methods, e.g. using a QR code may improve usability.

While the service needs access to the NSI's database of user credentials, management of those credentials is not in scope of the service.

4.2.2 Authentication mechanism

An authentication mechanism specifies the technical implementation of the authentication process. The authentication process needs to provide for the secure transport of the authentication data from the user's device over the Internet to the backend service, which is responsible for validation. After validation, the authentication protocol has to provide a means to allow re-identification of already authenticated respondents so that they do not have to enter authentication data a second time.

The current solution tightly integrates with the authentication infrastructure of Statistics Austria. Thus, an authentication process has to be designed and implemented which is decoupled from the NSI's infrastructure and fulfils the requirements stated above regarding security and usability. In order to meet security best practices, it is recommended to select and implement a solution based on open standards which is widely in use for mobile apps.

4.3 Internationalization and Localization

Internationalization and localization refers to the ability of an app to adapt to the preferred language and input scheme of the respondent. In order to allow for internationalization, the program structure has to be reorganized in order to separate text data and program logic.

Localization then refers to the actual translation of text data into the destination language. The result of internationalization and localization is a user interface that is presented either in the preferred language specified in the mobile device's system configuration or in a language explicitly selected by the user within the app.

In addition to the language, the application needs to adapt to other data representation conventions at the respondent's site, including the date and time format (e.g. "7.8.2018, 13.00" and "08/07/18, 1:00 p.m." refer to the same date and time using different representations). While localization refers to the way data is presented to the user, backend data processing needs to be location-independent. For example, the date and time mentioned before is processed as "2018-08-07T13:00" within the service.

Generally, internationalization also has to provide for different time zones. For the purpose of the time-use survey, time zones may be disregarded though, since the times specified in time-use diaries refer to the local time zone of the respondent.

The current solution provides a user interface in German and uses the date and time format which is common in Austria and Germany. Thus, internationalization and localization has to be implemented as described above. While implementation includes providing support for multiple languages and for selection of a language, the actual translated texts need to be provided by the respective NSIs, though.

4.4 Multi timeslot filling method

As the first draft of the app only provided the functionality to copy activity information from the previous timeslot, in this grant we implemented a method to fill multiple timeslots at once. So the user can log very time consuming activities easily. All the timeslots contained by the time span will be filled with the same information and but kept as separate timeslots. Single timeslots within the provided timespan or sub time spans can be edited easily afterwards. The last alteration is the one kept by the app. After all the total number of 144 timeslots per day stays the same.

4.5 CSPA service catalogue

Statistic Austria will produce a full system documentation in English and using the Common Statistical Production Architecture (CSPA) templates like discussed in the Task Forces for TUS innovation.

Since the app and the backend shall be reused by multiple NSIs, it is a candidate for inclusion in the CSPA service catalogue. The service consists of a frontend app available for Android and iOS smartphones, and a backend REST web service. The service covers the collect phase of the Generic Statistical Business Process Model (GSBPM).

In terms of the Generic Statistical Information Model (GSIM), the service collects time-use data using an electronic questionnaire presented on a mobile app as exchange channel, which produces an information set consisting of the collected time-use data and metadata.

This will be submitted no later than seven months after the start of the action.

4.6 User Guide

Furthermore we will produce a User Guide for other National Statistic Institutes (NSIs) where question like “How to use the tool?” and questions regarding like the deployment “How to install the tool and make it operational” will be documented in English.

4.7 Setup Guide

Additionally we will produce a Setup Guide for other National Statistic Institutes (NSIs) where question like “How to set up the development environment?”, “How do I customize the app?” and “How do I run the app and server setup locally?” is documented in English.

5 The App – technical information

The following chapter describes the technical structure of the App. There will be an overview of the technical features the app, how the installation worked and what software was used.

5.1 Installation

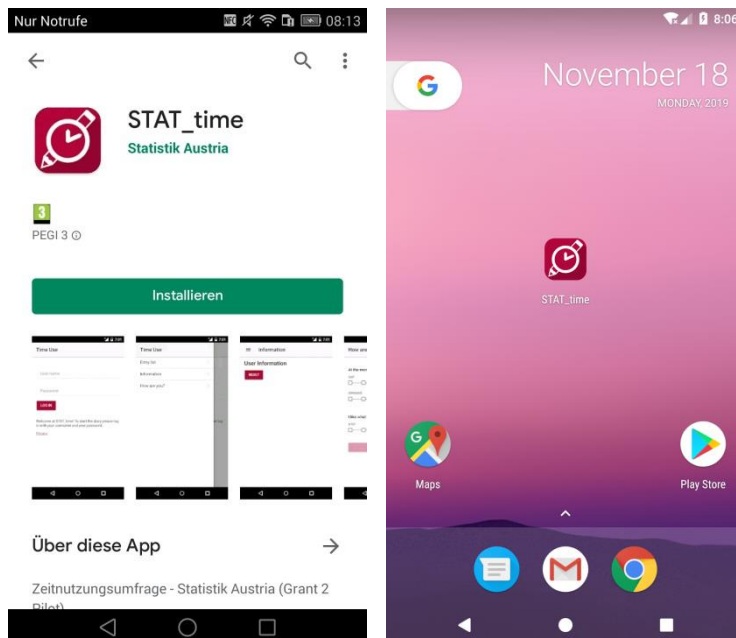
The TUS app is available for two different types of smartphones: Apple’s iOS (iPhone) and Google’s Android, which is the mobile operating system running on most smartphones from other vendors than Apple. Each of those platforms provides a store, which is a method for distributing new apps: App Store on iOS and Play Store on Android.

In order to install the app, the respondent searches for the term “STAT_time” and taps the search result entry that matches the icon and description of the STAT_time app. Then, installation follows the standard process of installing an app on the smartphone, to which users of the device are accustomed. If the app has already been installed before, the store redirects the user to the installed version.

The Following instructions will contain the screenshots and instructions for

5.1.1 Google Play Store / Android Phone

The following pictures show the overview page and details page of the app descriptor on Google Play Store. If the user taps on the green “Installieren” (install) button, the app will be installed.

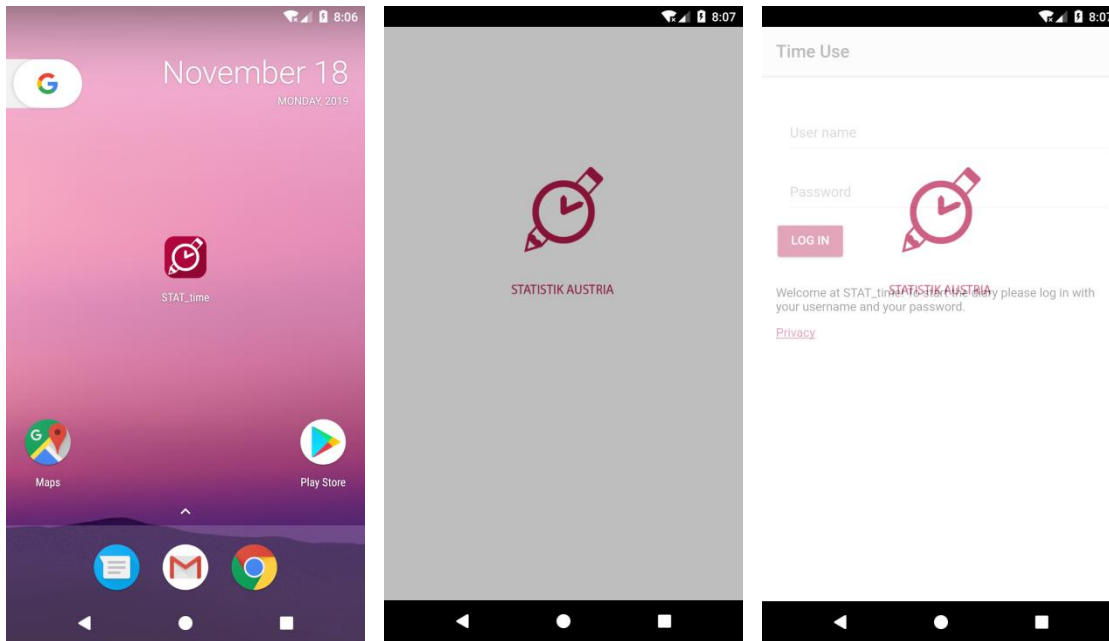


5.2 Usage

This section describes how to use the app on the mobile device, if it has already been installed following the instructions in the previous section. The following sections show the different views (screens) of the app and describe their functionality.

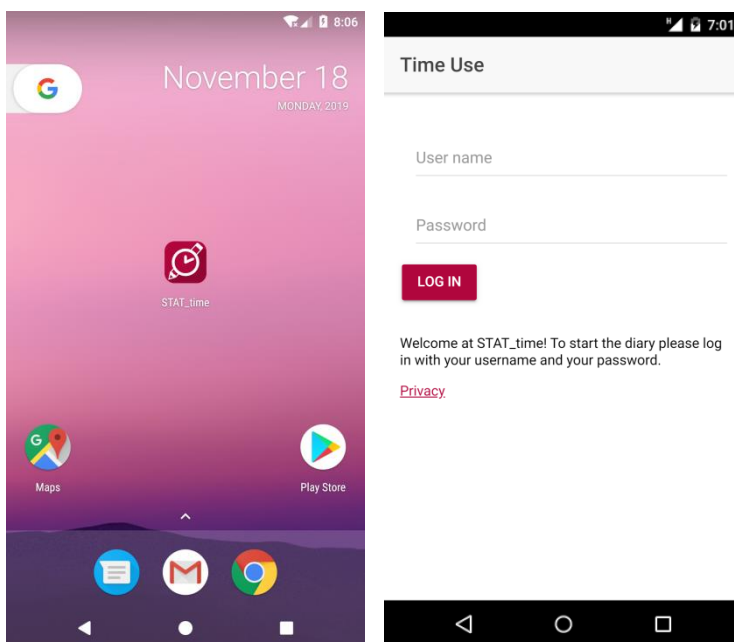
5.2.1 Starting the App

Start the app via clicking the symbol on the screen.



5.2.2 Login

Initially, the app shows the login screen. If the respondent has been provided with a username and a password by the NSI, they shall be entered here. Otherwise, the user will not be able to use the app.



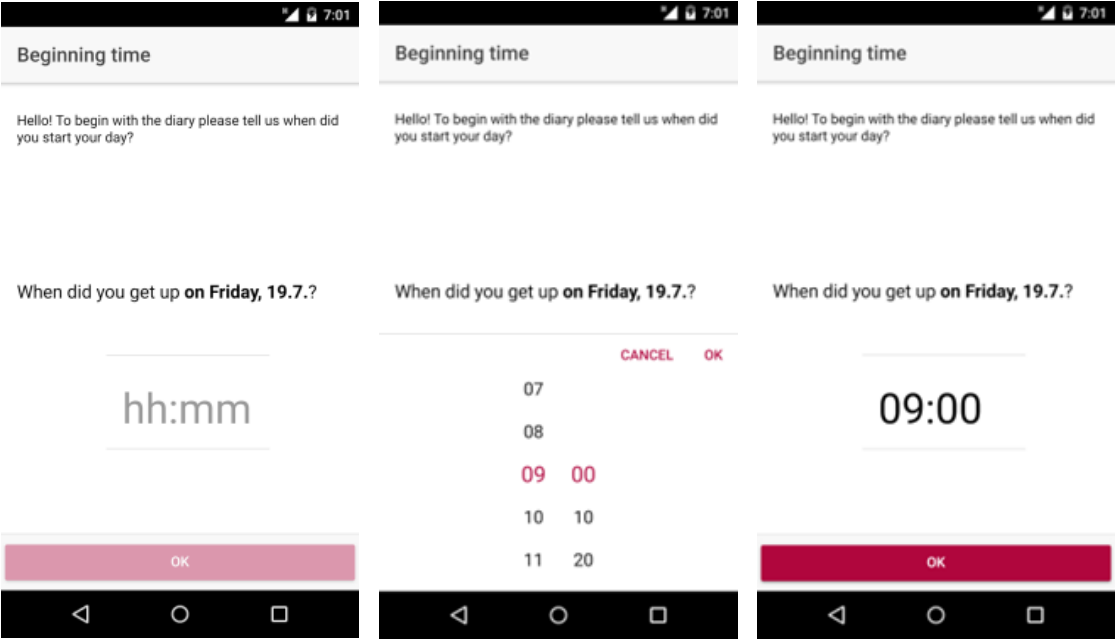
Authorization data is stored locally on the phone using SHA256, allowing the respondent to close and reopen the app without having to enter the user id and password again. Only after finalizing the

diary, clearing the cache data or uninstalling the app, authorization data is cleared from local storage and has to be entered again.

5.2.3 Start time

After login, the respondent selects the start time of the specified day. This is supposed to mean the time of getting up from bed in the morning for most respondents. The survey period starts with the selected time and ends ten minutes before the selected time on the following day.

The respondent selects the start time by tapping on the grey “hh:mm” text. Then, the operating system specific time controls are displayed, which allow the respondent to enter the time value.

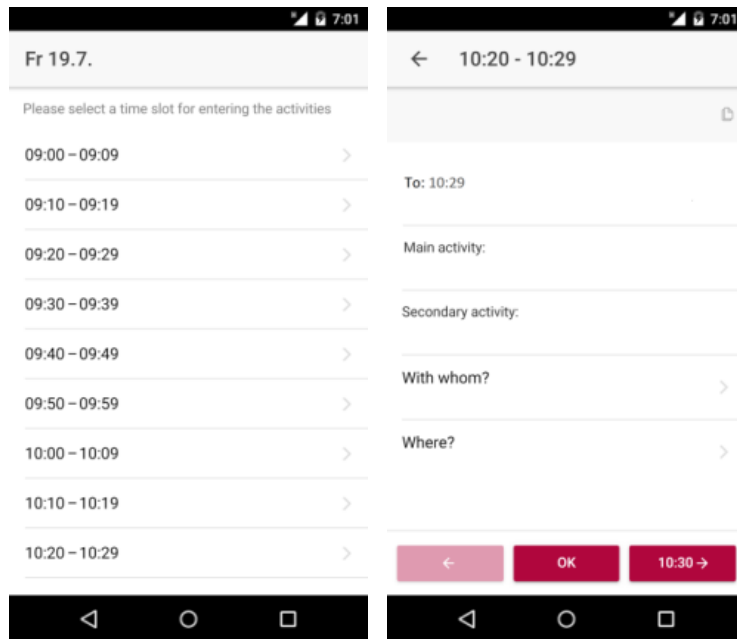


5.2.4 Initial list of time slots

After the respondent has logged in and selected the start time as described in the previous sections, the list of time slots is displayed. This is the central point of the app that allows the respondent to enter activities by tapping on the appropriate time slot. If the respondent swipes up or down, the time slots currently not visible will be shown.

Note that the 23:50 time slot is followed by the 00:00 time slot for the next calendar day – unless the respondent has selected 00:00 as start time, in which case the 23:50 time slot is the last one.

The following pictures show the list of time slots of an empty diary.



5.2.5 Details view

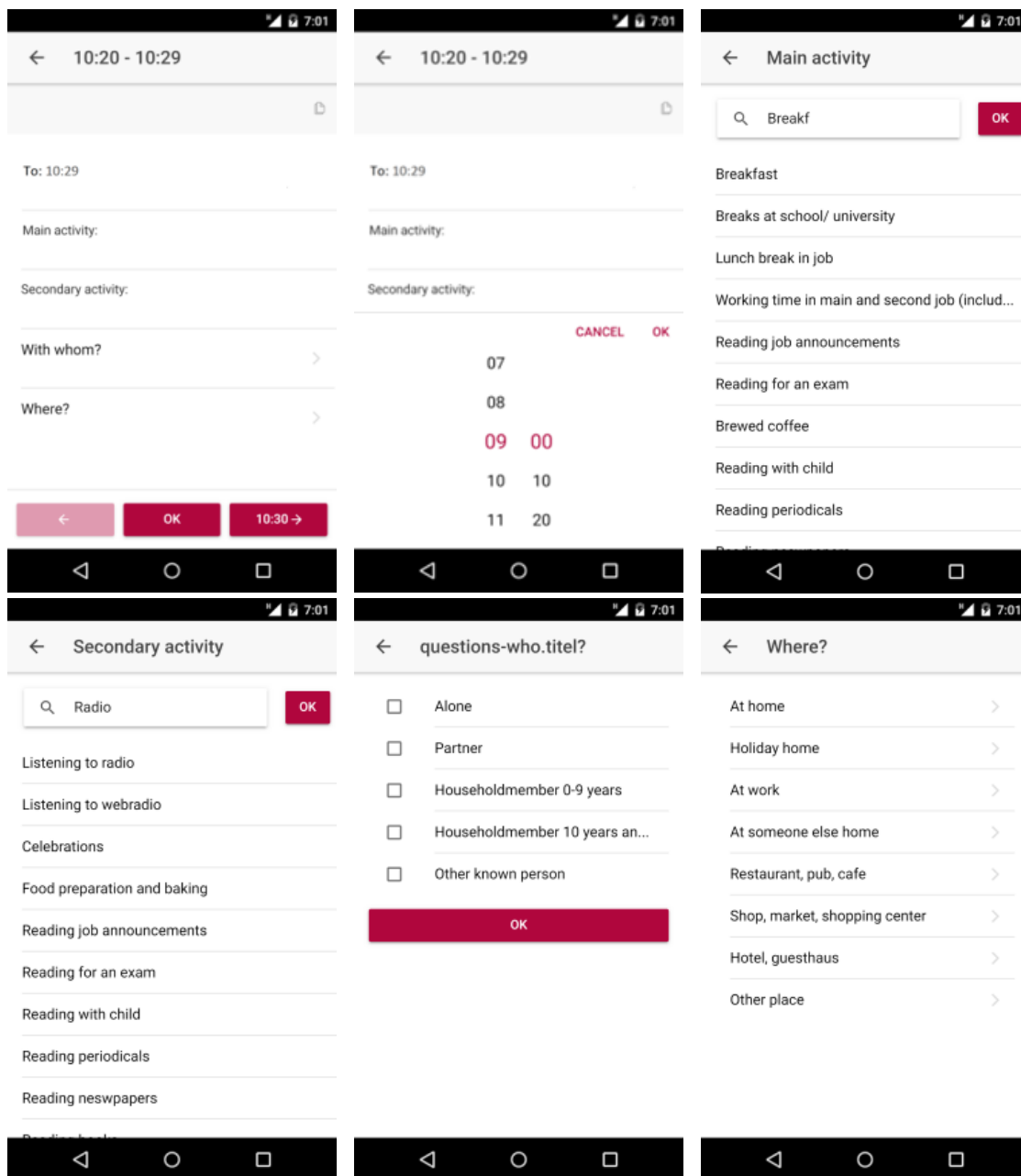
If the respondent taps one of the time slots, the details view is displayed and allows the user to fill in the activity.

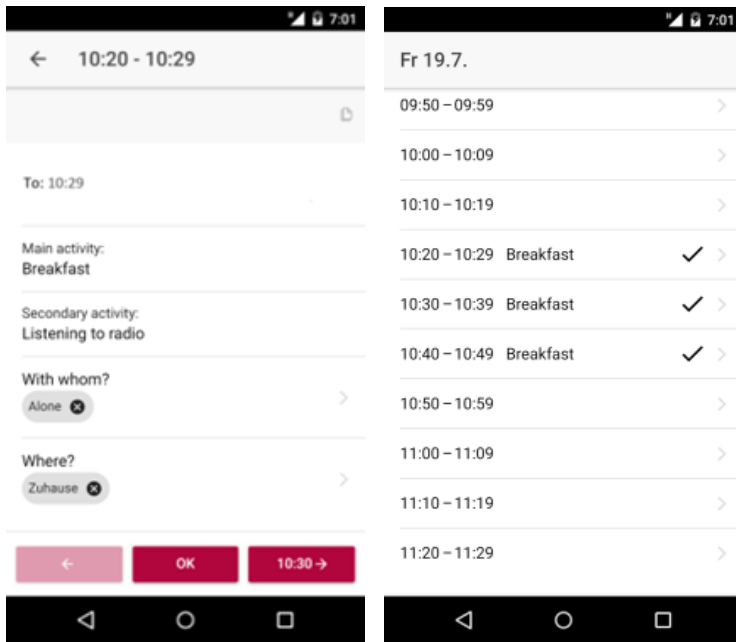
The first row is to select a timespan which will be filled identical with the rest of the provided data.

The second and third items contain the primary and secondary activities which are described further below.

Tapping on the fourth row (“With whom?”) switches to a view that allows specifying the accompanying persons.

Tapping the bottom row (“Where?”) switches to another view that allows selection of the location.





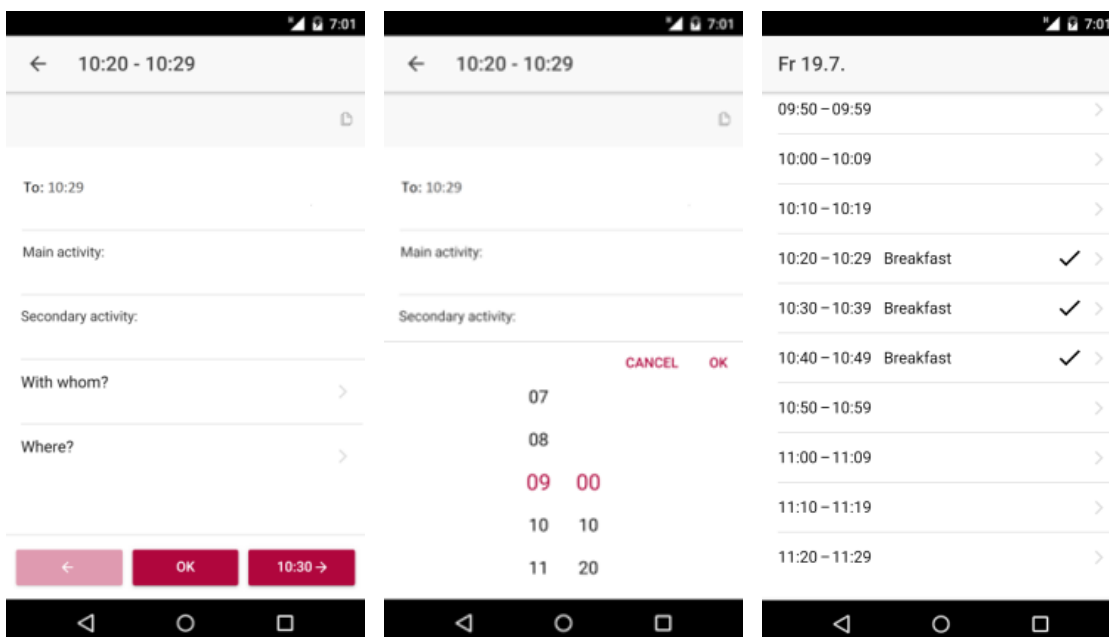
5.2.5.1 Navigating

The three red buttons on the bottom row allow the respondent to directly navigate to the previous time slot (left button), the next time slot (right button) or to the list of time slots (middle button). Swiping left or right is an alternate method for navigation to the previous or next time slot.

5.2.5.2 Copying entries, filling multiple time slots at once

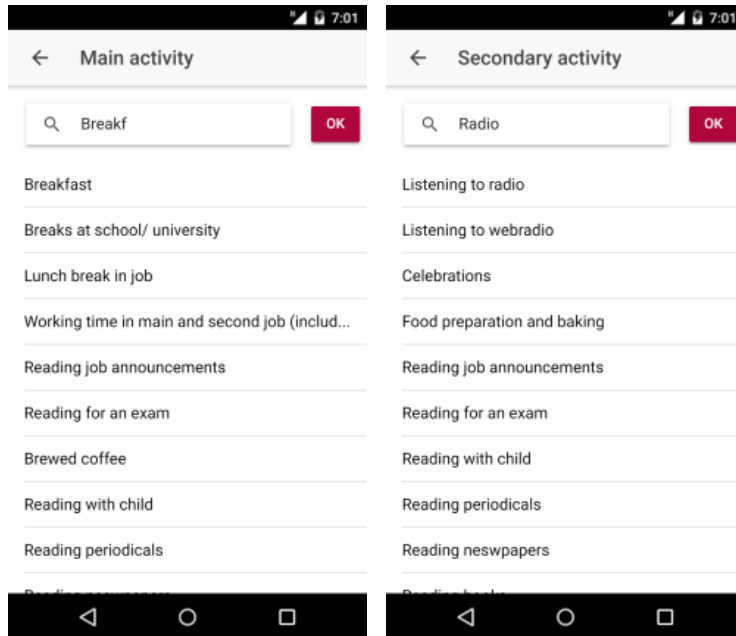
In order to facilitate entry of activities that last longer than ten minutes, a copy button is provided in the top right corner of the activity view, provided that a previous entry exists and is filled out correctly. If the respondent taps the copy button, the information contained in the previous time slot is copied into the current one.

Additionally to the copy function you can select a “to” time for the timeslot. All the single timeslots in this span will be filled with the exactly same data. After doing such an mass manipulation the user is free to edit the single timeslots individually.



5.2.5.3 Activity selection

If the respondent taps on the labels main activity or secondary activity, the activity view is displayed. Here, the respondent must provide a textual description of the main activity and may provide a textual description of the secondary activity. The textual description of the activity may be typed in free form or selected from a dictionary list described in the next section.



5.2.5.4 Dictionary search

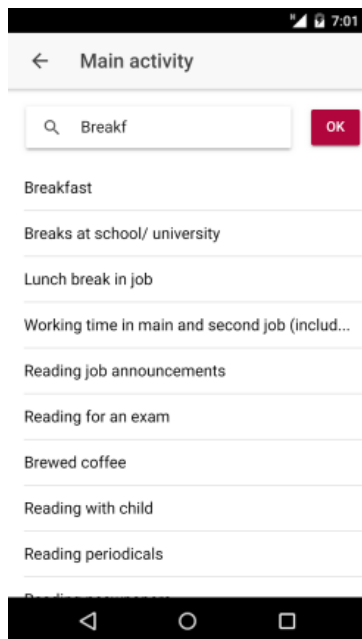
In order to help the respondent to fill out the activity description, after typing the third letter the app displays a list of dictionary entries that match the yet incomplete description. The initial dictionary is stored centrally on the server and contains terms derived from responses from the previous time-use survey. If the intended activity description is displayed, the respondent may complete the entry by tapping on that description. If the list does not contain an appropriate activity description, the respondent may ignore the list and finish typing. In addition to the dictionary list, the respondent may use the dictionary feature provided by the mobile device, if available.

In order to allow the app to handle typographical errors or alternate spellings, a fuzzy search algorithm is employed to find matching entries. The app incorporates the “fuse.js” programming library¹ for fuzzy search, which implements the Levenshtein distance algorithm² in order to find matching entries and sort them by similarity to the respondent’s input.

¹ <http://fusejs.io/>

² <https://xlinux.nist.gov/dads/HTML/Levenshtein.html>

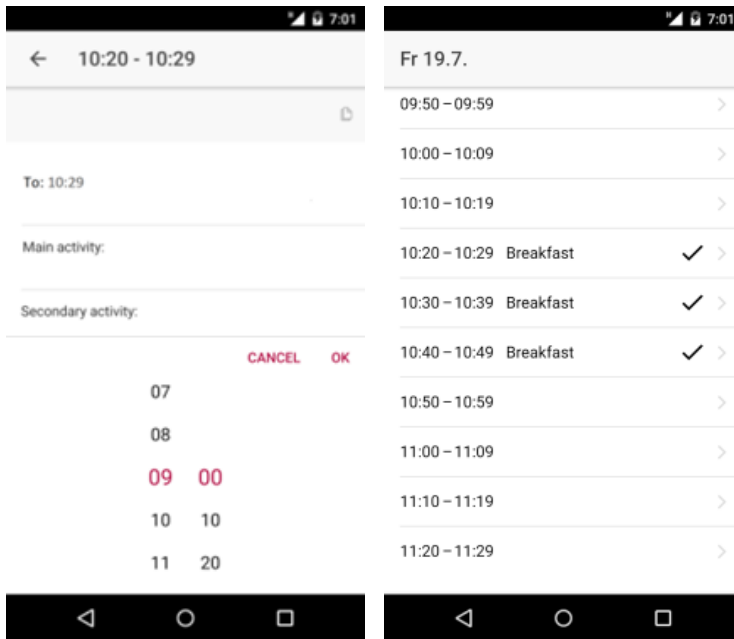
The following picture shows a list of suggestions provided to the respondent after starting typing. Note that the respondent typed “Breakf” which results in the app showing a list of terms which start or contain parts of “Breakf” due to the fuzzy search algorithm described above:



Future versions of the app may learn from the respondent’s previous answers by adding them to the locally stored copy of the dictionary, while leaving the central dictionary unchanged.

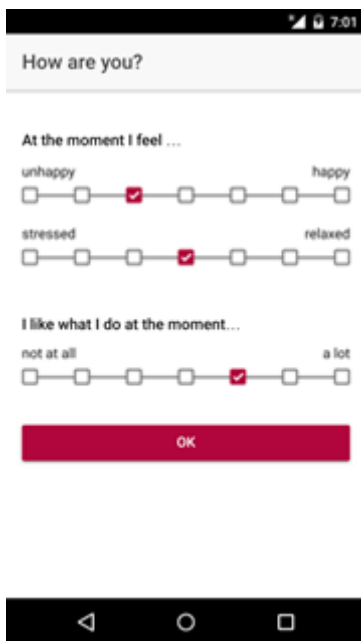
5.2.5.5 Long-lasting activities

In order to prevent the respondent from having to copy or even repeatedly type long-lasting activities into several consecutive ten-minute-slots, certain dictionary entries may be designated for allowing variable duration. This feature is especially targeted at the activity of sleeping which concludes the day for most respondents. So, if the respondent selects or types sleeping as main activity, the app asks for the activity’s end time and automatically fills out all of the corresponding consecutive time slots.



5.2.6 Subjective Well-being questions

In addition to the activities, the app collects the respondent's answer to well-being questions. The app tries to ensure that well-being questions are filled out just in time in order to gather accurate results. So, at certain times predefined in the respondent's user profile, the app is interrupted and the well-being page is displayed:



In order to provide the visual appearance of a slider while avoiding suggesting a default value, the input control of each well-being question consists of several checkboxes connected with a line. Initially, all of the checkboxes are empty. If the user taps on a checkbox, it is activated while all other checkboxes in the same row are deactivated. Only if all of the subjective well-being questions are answered, the Ok button is active.

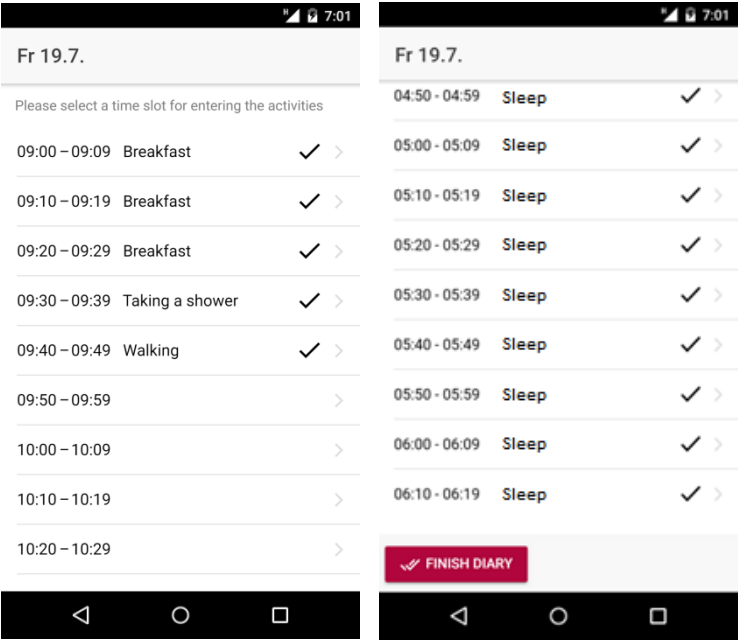
If the app is not in the foreground (i.e. active) at the specified time, the device issues a notification which is displayed at the typical screen location, which may be the top margin, a popup window or

the lock screen depending on the mobile device type. If the user taps on the notification within ten minutes, the subjective well-being question is displayed. After ten minutes, the notification is cancelled automatically. Unfortunately, the usability tests have shown that the cancelling mechanism does not work under all circumstances, so the final survey may contain well-being questions answered out of time. Those answers may be easily identified by comparing answer time to question time, though.

5.2.7 Completed list of time slots

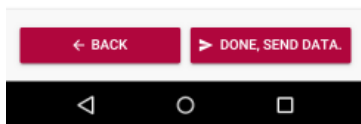
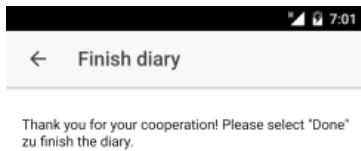
If the respondent has scrolled to the bottom of the list (by repeatedly swiping in the down-up-direction), there is a button that allows to finish the diary. The following picture shows the completed survey with the button to finish the diary for the day (“Finish diary”) at the bottom of the list. Note that the checkmarks in the rightmost column state that the corresponding entry has been filled out completely. On incomplete entries, a red exclamation mark will be displayed instead.

If the main activity of all time slots is filled out, the app will close the diary and send all the data to the server. Note that incomplete entries (i.e., with empty location or unspecified accompanying persons) do not prevent the diary from being marked as finished, while totally empty entries lead the app to display an error message.



5.2.8 Finish the days diary

After the respondent has tapped the “finalize” button described in the previous section, the final page is shown which by default just confirms that the respondent intends to finalize the diary and send data to the server. After finalizing the diary, data may not be changed anymore. In the current version of the app, this is ensured by logging out the user and preventing further login with the same id. Future versions of the app may provide a read-only view of the data after completion.



5.2.8.1 Retrospective subjective well-being question

If the respondent has not answered any well-being question before finalizing the diary, the well-being page described further above is merged into the final page in order to allow answering a single subjective well-being question retrospectively. The user profile designates the timeslot for the retrospective question:

← Finish diary

Thank you for your cooperation! How did you feel at this moment?:

12:00 - 12:09 Uhr

From: 12:00 until 12:09 o'clock I felt...

unhappy happy

stressed relaxed

What I did from 12:00 until 12:09 I liked...

not at all a lot

← BACK DONE, SEND DATA.

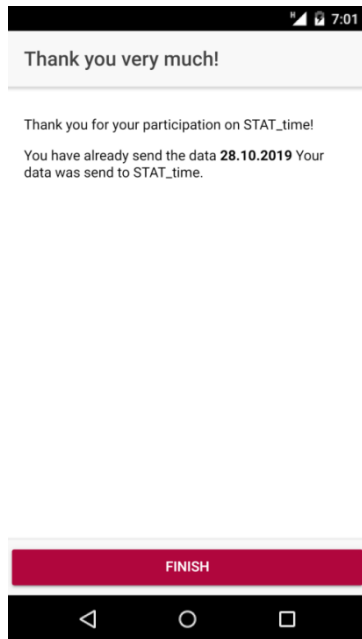
Note that unlike in the case of the well-being questions issued on time, answering the retrospective well-being questions is optional.

This chapter has described the installation and use of the TUS app. The next chapter describes the architecture of the app and provides rationales for the design decisions that have led to the current appearance and functionality of the app.

5.2.9 Repeat steps 5.2.2 to 5.2.7 to finish the second day

5.3 Finish diary

After you finished sending the data for day 2 you will be provided with a final page, showing that the data collection is complete, and no further interaction with the app is necessary.



6 Software Architecture

Software architecture describes the fundamental structure of an application. The architecture of the STAT_time app has been developed based on several requirements, which are explained in the following sections: the application shall operate without Internet access (offline), shall frequently send data to the server if online, shall issue well-being notifications even if the application is currently closed and shall run on most of the mobile devices (smartphones) currently in use.

6.1 Requirements

Functional requirements describe the features of an application, that is, what the application should do. Non-functional requirements describe environmental or performance characteristics of the application, that is, how the application should accomplish the functional requirements.

6.1.1 Offline operation

A goal of the application is to allow the respondent to fill in the current activity just in time in order to gather accurate data. Since respondents are possibly situated in a location with poor Internet access or no Internet access at all, the application must allow the respondent to update their diary without being online. Consequently, the application must provide a means to store entered data locally in order to prevent loss of information not yet transmitted to the server due to power outage (e.g. empty battery), device failure or the respondent deliberately powering off the mobile device.

6.1.2 Online operation

In contrast to the offline requirement, the NSI is interested in access to the most current data, albeit incomplete or subject to change. Change patterns in current data allow the NSI to detect if the respondent has difficulties in using the app. If the respondent aborts the diary, the NSI can operate on the incomplete data. So, the requirement is that the application shall frequently send data to the server, provided that Internet access is available in sufficient quality.

6.1.3 Local notifications

An important feature of the survey is to provide the respondent with subjective well-being questions. It is desired that these questions are answered on time. It is an essential feature of a smartphone to interrupt its user in order to notify them of a current event, for example a phone call or a text message. The application shall use the device's notification functionality in order to issue the well-being question. The well-being notification shall occur even if the application is currently closed.

6.1.4 Sufficient number of devices

For every application it is desired that it can be used by as many users as possible. However, supporting older devices or multiple operating systems may exceed time and budget constraints. So, a balance must be found in order to enable the majority of smartphone users to participate in the survey while ignoring devices which are too old or have a too low market share. As of 2017, the most popular types of smartphone are Google's Android and Apple's iOS (iPhone).

6.1.5 Open source

The whole technology stack of the app should be open source. This means that every NSI can install the app and its backend on premise without fearing for any costs at all.

6.1.6 Authentication standard

The app has to use a standardized method to authenticate to the backend of the app. STAT_time users the “Basic Authentication” using a username and password for implementing this. The user credentials have to be transmitted and stored in a safe way, protecting it from getting corrupted. Therefore HTTPS and SHA are used.

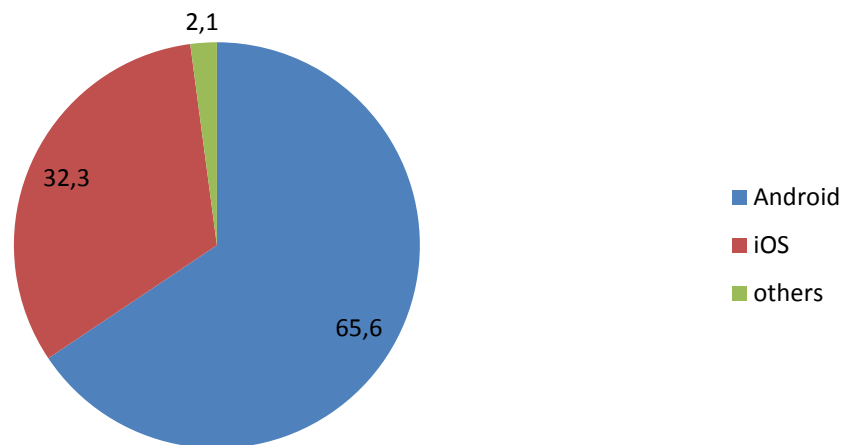
6.1.7 Internationalization and Localization

The standard for internationalization and localization of the app called “i18n” has to be fulfilled.

6.2 Target devices

This section describes the process that has been followed to decide which mobile devices shall be supported by the app. Several websites exist which allow to assess the current market share of mobile device platforms³. As of May 2017, Android has worldwide market share of 65.6 percent, iOS has 32.3 percent, and others (e.g. Windows phone, Blackberry) together have 2.1 percent.

Market share of mobile operating systems (May 2017)



The diagram clearly shows that it is reasonable to focus on supporting Android and iOS, while disregarding the other available platforms.

6.2.1 Operating system versions

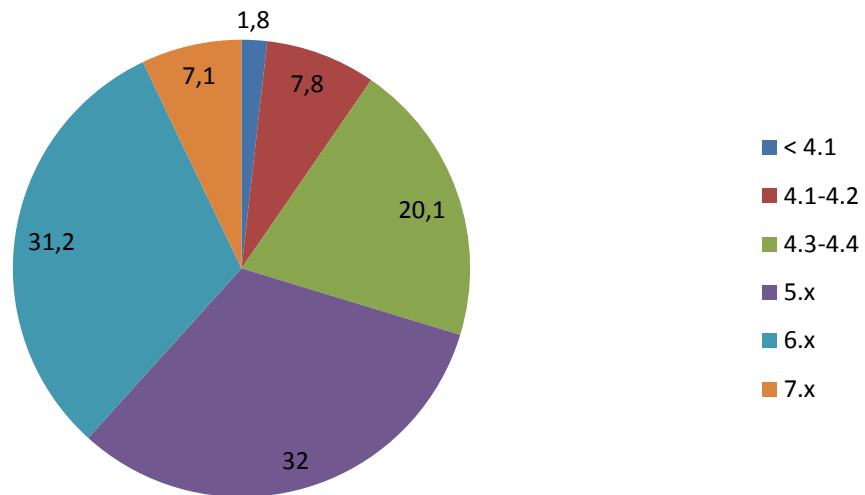
Within each operating system platform, a decision has to be made on the oldest version to support. Older operating system versions usually run on devices with less power, while providing fewer features and having more bugs than newer devices, resulting in additional effort for their support. Thus, the considerations written in the previous sections apply also to older versions. Additionally it

³ <https://david-smith.org/iosversionstats/>
<https://developer.android.com/about/dashboards/index.html>
<https://www.netmarketshare.com/operating-system-market-share.aspx>

is to be expected that an already low market share of an older device type will almost always further decrease in the future.

The following diagram shows the market share of Android versions.

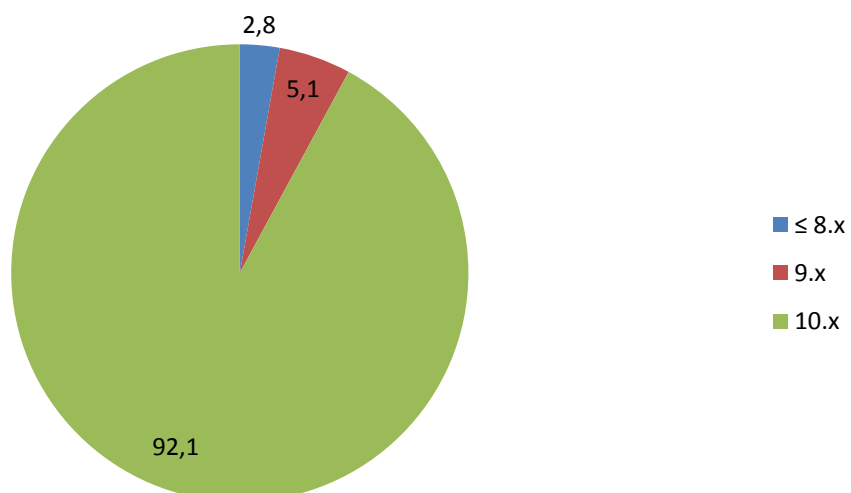
Android versions (May 2017)



The diagram would suggest that it is reasonable to support versions 4.1 and higher. The final app has been restricted to supporting versions 4.3 and higher, though, which corresponds to devices produced or updated in 2013 or later. The reasons for this restriction will be described in the section on hybrid apps below.

The following diagram shows the market share of iOS versions.

iOS versions (May 2017)



The diagram suggests that it is reasonable to support iOS version 9 and higher.

6.2.2 Market share of supported devices

Based on the market shares described in the previous sections and known technical limitations of specific older operating system versions, the decision has been made to support Android versions 4.3 and higher, and iOS versions 9 and higher. This results in a market share (in May 2017) of

$$(1 - 0.078 - 0.018) * 0.656 + (1 - 0.028) * 0.323 = 0.907$$

Thus, based on that decision, the mobile app supports about 90.7 percent of the smartphones in use worldwide.

6.3 Development platform

Currently, several approaches to develop a mobile app are in use. With every computing device, there is the possibility to develop an app targeted specifically for that device, called a native app. On the other hand, there exist tools and frameworks that allow development of an app suited to run on multiple device types. The following sections describe the main approaches to develop a mobile app.

6.3.1 Native development

Native apps, since being developed specifically for a certain operating system are able to exploit all of the features provided by that device with the best performance and consequently are superior to all of the other options. The main disadvantage is that additional effort has to be made in order to support other operating systems. The technical differences between Android and iOS do not allow any code developed for one system to be reused on the other system. In many organizations, development skills are only present for one of those platforms.

Since the decision has been made for the TUS app to support iOS and Android, native development is not feasible.

6.3.2 Web apps

Recently, there has been the trend in software technology to provide new applications as web apps. Modern web browsers meanwhile support the presentation of feature-rich user interfaces with almost no difference to traditional applications. Adaptive web design allows interfaces that change their layout in order to adapt to different screen sizes. The main drawback is that web apps are still limited in their access to basic system functionality. Plus, web apps usually are not distributed via the usual channels (stores), resulting in additional effort of directing respondents to the web site.

Specifically for the TUS app, web apps do support local notifications, but cease issuing notifications if being closed by the user. So, the requirement stated above to present well-being questions even if the app is closed cannot be met with a web app.

6.3.3 Hybrid apps

In order to allow the development of apps that run on multiple operating system platforms, while allowing access to most of the hardware functions of the underlying device, several frameworks have been developed that combine the advantages of native apps and web apps. Apps developed with those frameworks are usually called hybrid apps. Due to the layered architecture consisting of a container and an app running in that container, performance is somewhat slower than with native apps.

The following table summarizes the main advantages and disadvantages of the approaches for mobile app development.

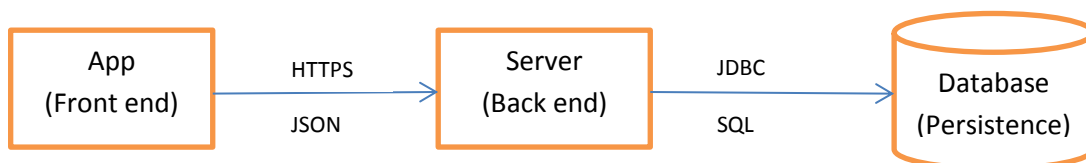
	Advantages	Disadvantages
Native	Best performance and features	Only for a single platform
Web	Runs on all platforms	Limited features
Hybrid	Many platforms, many features	Slower than native app

For the TUS app, the decision has been made to implement a hybrid app using the Cordova framework.⁴ Cordova provides a container for each target platform that contains the almost platform-independent web app providing the functionality to the user. Plugins allow access to system functionality like storage and local notifications. The user interface has been developed using the Ionic framework⁵ which allows adaptation of the app to the look and feel of native apps thus providing a uniform overall user experience.

Hybrid apps run in the web browser environment provided by the operating system. Since the browser engine that is available on Android versions 4.2 and below does not fully support the requirements of Cordova, apps that need to support older versions have to ensure installation of a new browser engine. Since this would have required additional effort and resulted in a larger app package, while targeting only about 5 percent of the mobile phone users as of May 2017, the decision has been made to support Android 4.3 and higher versions only.

6.4 Structure

While the app on the mobile device is the user visible part of the overall application, several other parts are needed to provide complete functionality. The server which is located at the NSI needs to validate authentication data, send the user profile and dictionary to the app and receive the respondent's entries from the app. In order to permanently store data, a relational database management system is used.



The diagram shows that the app sends data encoded in the JSON⁶ format to the server over an HTTPS connection, which is an encrypted connection type to ensure privacy. After receiving data, the server which is written in the Java programming language sends SQL statements that store the data into the database over a JDBC connection. This structure comprises a three-tier-architecture, which is nowadays state of the art for designing distributed computer systems.

⁴ <http://cordova.apache.org/>

⁵ <http://ionicframework.com/>

⁶ <https://www.json.org/>

6.5 Distribution

Distribution comprises packaging and publishing of the app to Google's Play Store and Apple's App Store. Before publishing an app for the first time, the software developer has to register as an app developer with both companies. The following Section describes the registration and distribution process for Play Store and App Store.

Both stores require uploading an application icon, a certain number of screen shots and textual information.

6.5.1 Google Play Store

Registration to the Google Play Developer Console requires a valid Google account (i.e. an email address ending with "@gmail.com"). This may as well be a generic (non-personal) email address, which is then redirected to an internal mailing list of the NSI consisting of the app developer team. In order to upgrade that Google account to a developer account, a one-time fee of currently US\$ 25 has to be paid by credit card and the developer distribution agreement needs to be accepted. Although this registration method is very straightforward, it requires the developer team to share the password of the Google account. After completion of the preceding steps, apps may be published.

Publishing an app consists of uploading the package to the Google Play Console web site, filling out additional information, uploading icons and screenshots and starting the publishing process which usually lasts from a few minutes to a few hours.

6.5.2 Apple App Store

In order to allow registration to the App Store, Apple has defined several roles in a developer team⁷. A special role is the team agent which must be a single person that has been granted authority in a formal business letter by an authorized signatory (i.e. a top level executive) to bind the NSI to a legal agreement with Apple. The following registration process needs to be completed using the team agent's personal Apple Id account (which may be a business account separate from their possibly existent private Apple Id account). In addition, a D-U-N-S number⁸ of the NSI may be required. Upon enrolment and acceptance of several agreements, a fee of currently US\$ 99 per year is due which has to be paid by credit card. Using the credit card information, the written authorization letter, the D-U-N-S number and the personal Apple Id, the team agent may complete the registration process. The registration process needs a few business days on Apple's behalf and includes verification of the data by calling back the team agent by phone.

After completion of the registration process, the team agent may invite the other members of the developer team (all of them being required to obtain a personal Apple Id first) to join the developer account, either as an admin or as a simple member. Additionally, in order to allow one or more persons of the developer team to publish new versions of the app, an appropriate iTunes Connect role has to be assigned to the respective Apple Id.

Publishing an app to the App Store is supported only by using Apple's XCode development environment running on the macOS operating system. So, the development team needs to have access to an Apple computer system (e.g. MacBook or iMac) in order to publish an app. New versions of the app undergo a review process by Apple which usually lasts a few business days per submission

⁷ <https://developer.apple.com/support/roles/>

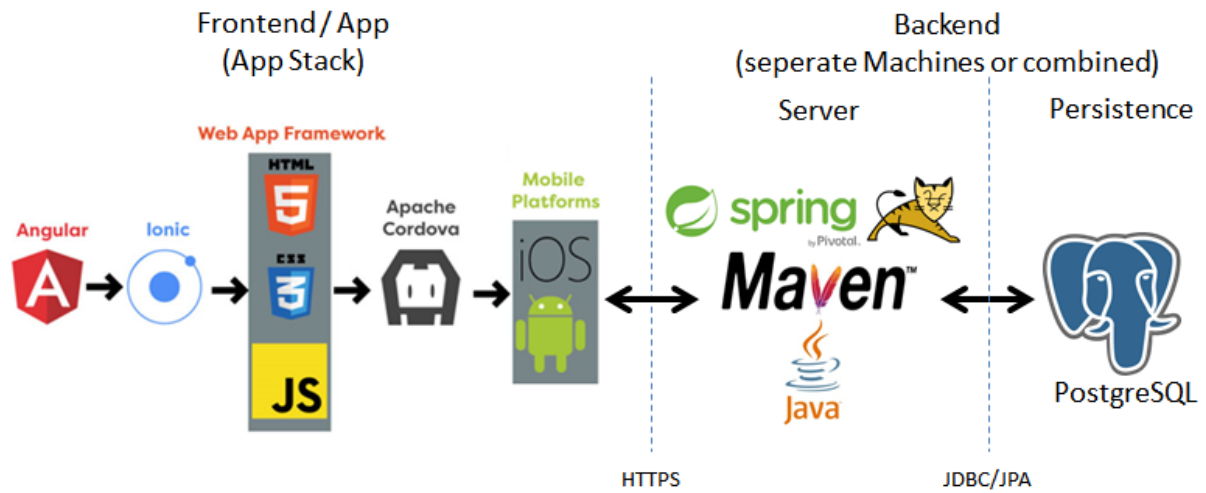
⁸ <http://www.dnb.com/duns-number.html>

and is supported by detailed guidelines. Organizations publishing their first app should plan for several iterations of the review process due to rejections.

7 Implementation details

7.1 Technology stack

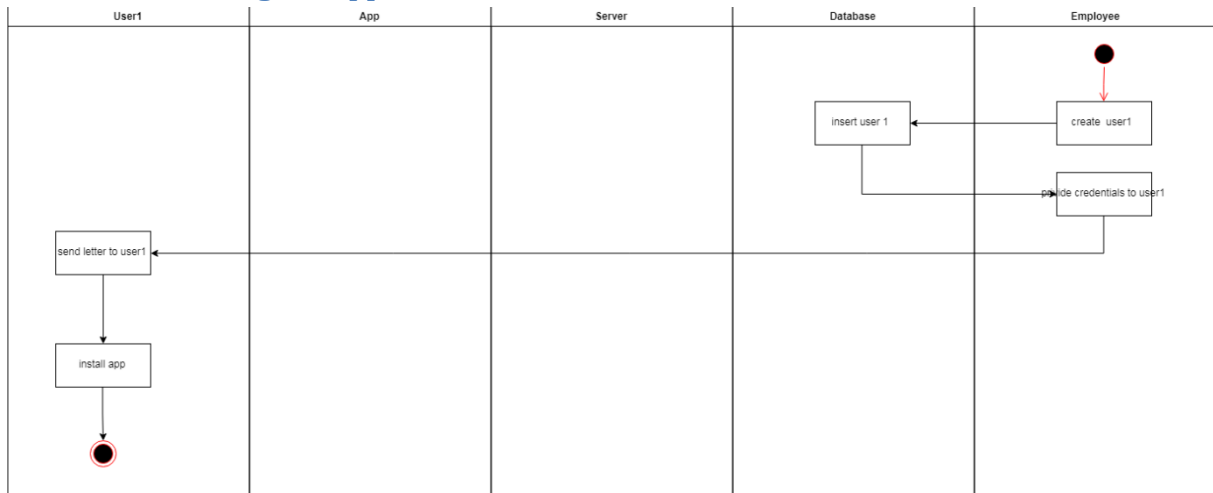
The following picture shows the full technology stack on which die app is based:



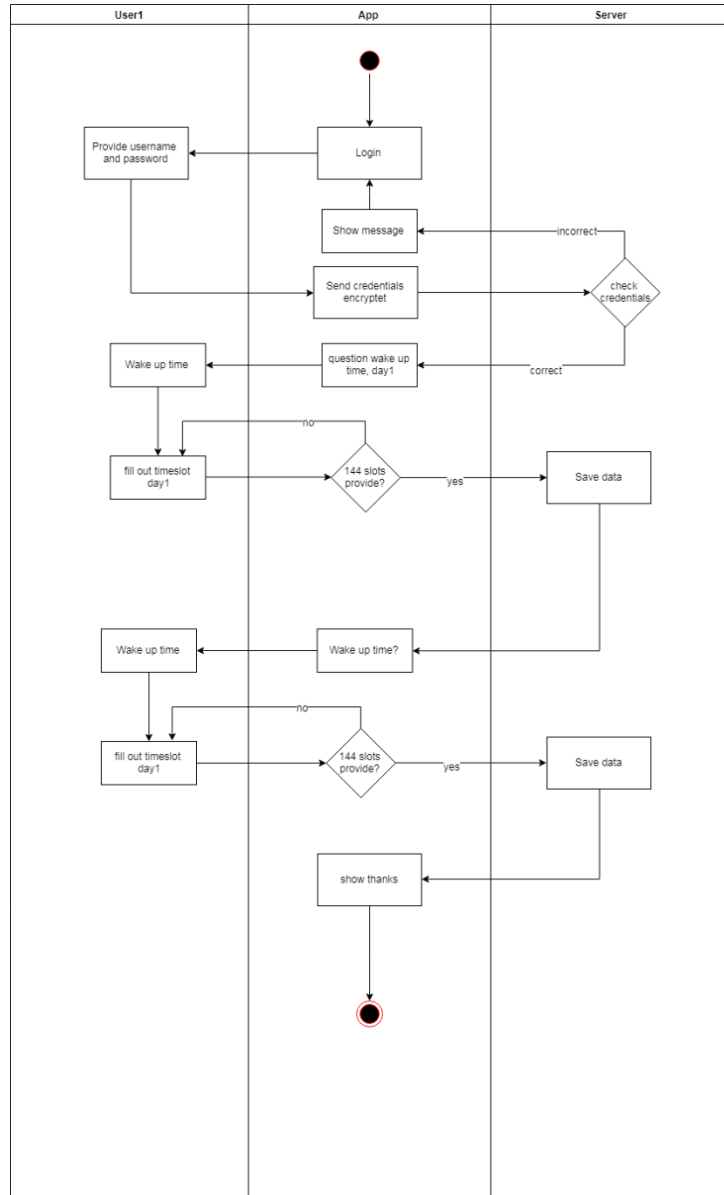
Source:
https://s14-eu3.startpage.com/cgi-bin/serveImage?url=https%3A%2F%2Fres.cloudinary.com%2Fimages%2F1_main%2C%2F_auto%2C%2F114140847%2Forig%2F176d04971c36d2b112b2e7b43336e41db8e307%2Fdevelop-ionic-cordova-apps.jpg&tr=3b94838682c337122e979d367-res.cloudinary.com%2Fimages%2F1_main%2C%2F_auto%2C%2F114140847%2Forig%2F176d04971c36d2b112b2e7b43336e41db8e307%2Fdevelop-ionic-cordova-apps.jpg&tr=3b94838682c337122e979d367
https://s14-eu3.startpage.com/cgi-bin/serveImage?url=https%3A%2F%2Fres.cloudinary.com%2Fimages%2F1_main%2C%2F_auto%2C%2F114140847%2Forig%2F176d04971c36d2b112b2e7b43336e41db8e307%2Fdevelop-ionic-cordova-apps.jpg&tr=3b94838682c337122e979d367
https://browse.startpage.com/do/show_picture.pl?deutsches=1&url=https%3A%2F%2Fmedium.com%2Fmax%2F1200%2F%2F%2F81V2q8V/DH0dDcNk_nw.png&tr=296834804711044ee01de3337&tr=default
<https://s14-eu3.startpage.com/cgi-bin/serveImage?url=https%3A%2F%2Fwww.postgresql.org%2Fmedia%2Fimg%2Fabout%2Fpress%2Frelease%2F023408303d0e2015%2F3480899e1adb3>
https://s14-eu3.startpage.com/cgi-bin/serveImage?url=https%3A%2F%2Fwww.silicon.de%2Fwp-content%2Fuploads%2F2017%2F06%2Fflem_db2.jpg&tr=ca6b3e2934083d93e83308ee93d045&tr=default
cdn.com%2Fthumbor%2F_Ao0Z2D_RV58xtVr7mUzpbk0vc%3D7%2F0%3A640w427%2F1200x800%2Ffilters%2Fcolor%2F820x0%2F84640w427%2F92Fcdn.vow-cdn.com%2Fassets%2F11087137%2Ffile_icon_640.jpg&tr=3ef010c33e97d2930f1b82014e9e0c

7.2 Activity Diagram

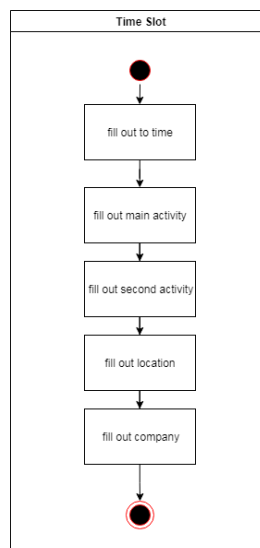
7.2.1 Before using the App



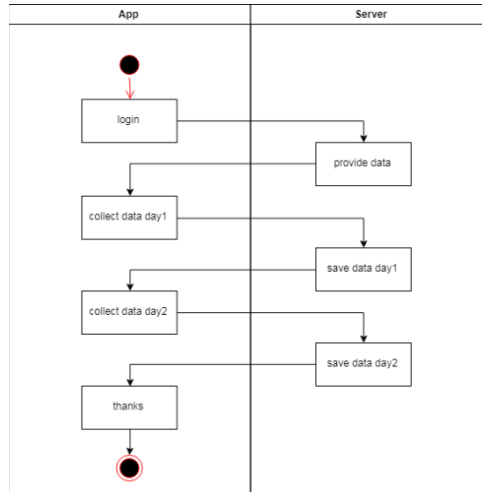
7.2.2 Using the app



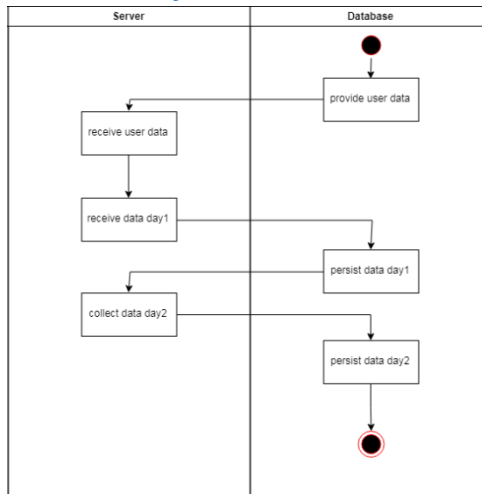
7.2.3 Filling a time slot



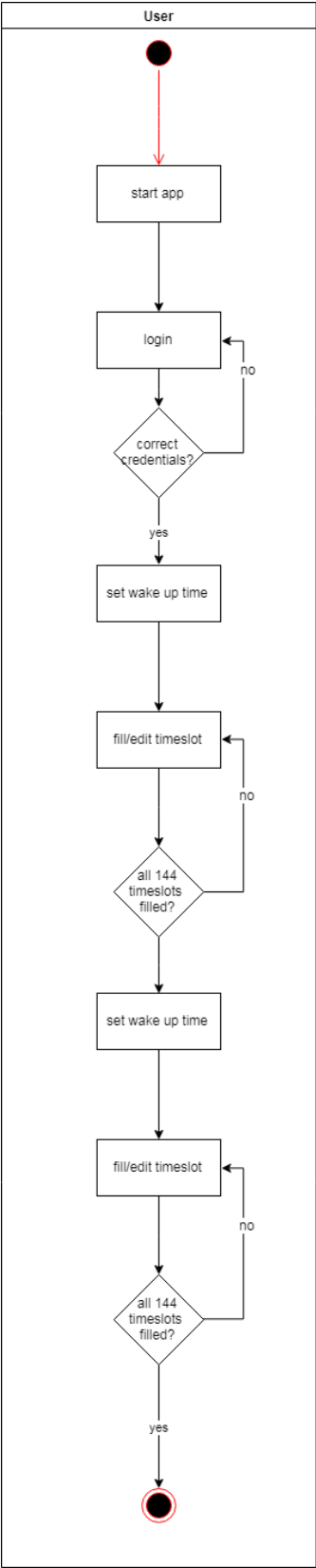
7.2.4 App server state activity



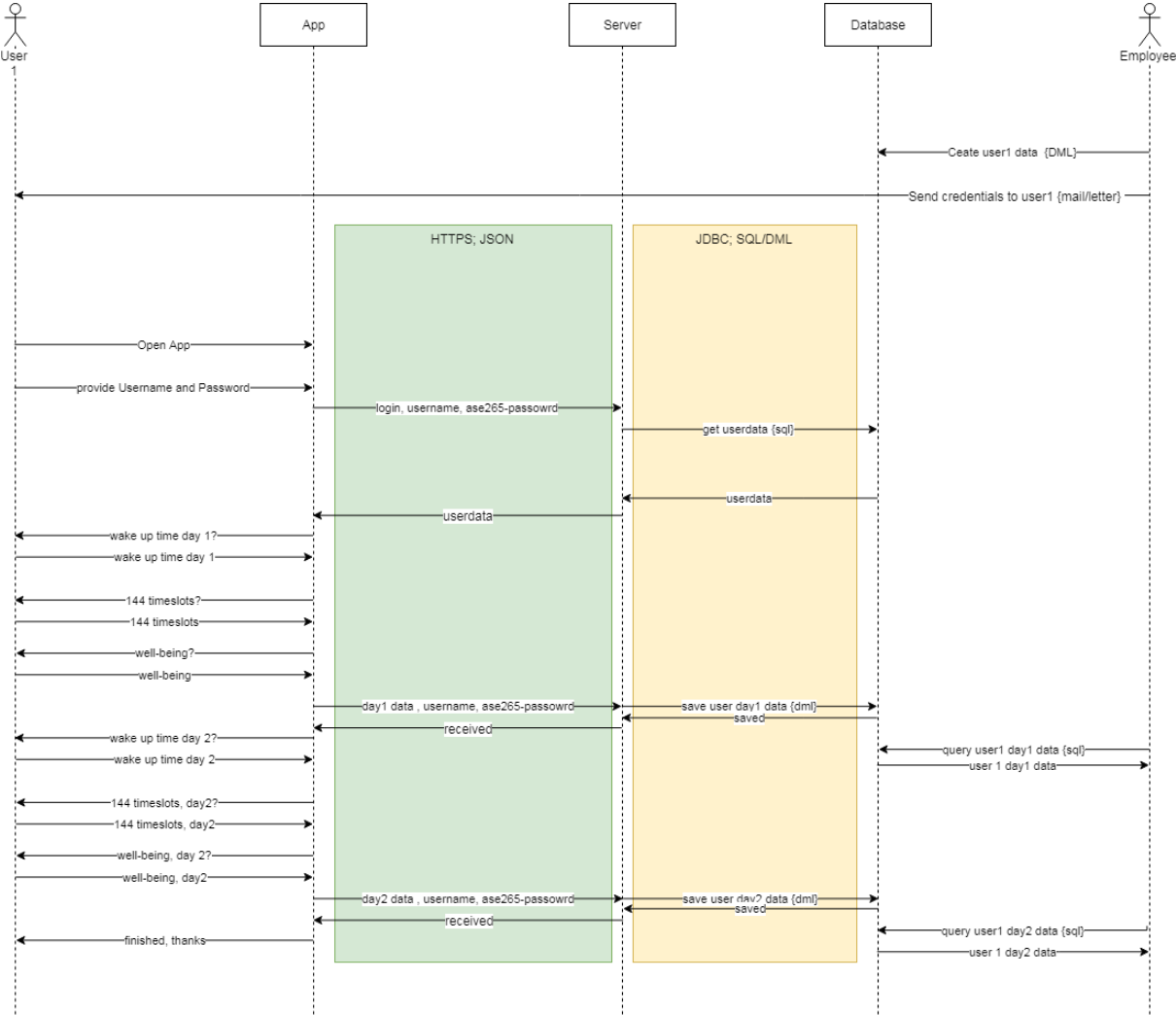
7.2.5 Backend server database activity



7.2.6 User activity



7.3 Communication Diagram



7.4 Implementation details

The sources of the service can be found at https://github.com/statistikat/STAT_time .

7.4.1 Basic authentication

The app uses “Basic authentication” via HTTPS which is one of the standard authentication mechanisms on the internet .In combination these technologies provide confidentiality.

Basic authentication uses a username and a password to authenticate a user at the back end of the service.

https://en.wikipedia.org/wiki/Basic_access_authentication

<https://en.wikipedia.org/wiki/HTTPS>

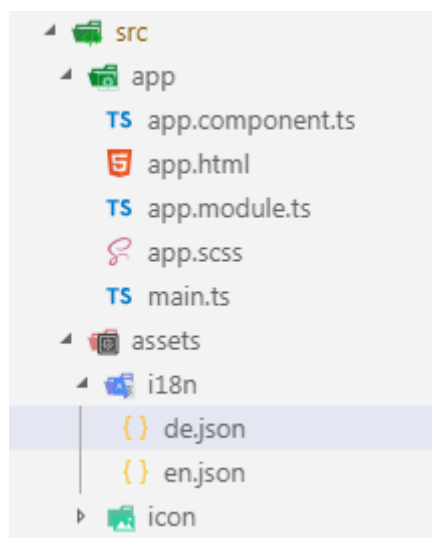
7.4.2 Hashing

For encrypting the user credentials of the app SHA256 is used. The encrypted password is in the database as well as in the local storage of the smartphone. SHA256 hashing is a save method for encrypting sensitive information on the web.

https://en.wikipedia.org/wiki/Secure_Hash_Algorithms

7.4.3 Internationalization and Localization

For internationalization and localization the web standard “i18n” was implemented. This means that by changing a file in the project of the app normally called “XX.json” (XX for the language, e.g.: “en.json”, “de.json”). This file can be edited in the asses of the app, containing a folder called “i18n”.



The file is structured as a key-value map. The left hand of each line contains a placeholder within the app. The right side provides the text in the language. Sample:

```
src > assets > i18n > {} enjson > ...
1 {
2   "title-survey": "Time Use Survey",
3   "title-timeusage": "Time Use",
4   "common": {
5     "cancel": "Cancel",
6     "done": "Done",
7     "back": "Back",
8     "ok": "OK",
9     "next": "Next",
10    "choose": "Please select",
11    "take": "Take",
12    "reset": "Reset",
13    "close": "Close",
14    "finish": "Finish",
15    "privacy": "Privacy",
16    "privacy-url": "https://www.statistik.at/web_en/about_us/data_protect
--

src > assets > i18n > {} dejson > ...
1 {
2   "title-survey": "Zeitverwendungserhebung",
3   "title-timeusage": "Zeitverwendung",
4   "common": {
5     "cancel": "Abbrechen",
6     "done": "Fertig",
7     "back": "Zurück",
8     "ok": "OK",
9     "next": "Weiter",
10    "choose": "Bitte auswählen",
11    "take": "Übernehmen",
12    "reset": "Zurücksetzen",
13    "close": "Schließen",
14    "finish": "Beenden",
15    "privacy": "Datenschutz",
16    "privacy-url": "https://www.statistik.at/web_en/about_us/data_protection
--
```

The structure of the file is identically for all languages. Changes in the structure of this files lead to changes in the app. If texts have to be split up

https://en.wikipedia.org/wiki/Internationalization_and_localization

<https://en.wikipedia.org/wiki/JSON>

https://en.wikipedia.org/wiki/Attribute%E2%80%93value_pair

8 General conclusions and future steps

In a world where a lot of respondents are used to a technology that makes their life easier in their daily life it is especially necessary to reduce respondent's burden and achieve high quality statistics in that way. With the Grant of 2017 we were able to develop an App and to test it in a smaller test environment. The detailed results of this test can be found in the report 2017 from Statistics Austria (see list of reference).

In general the app was working satisfyingly. All respondents received an invitation letter with the information for the download. All of them were familiar with downloading the app. The simple and consistent design of the app with its Icon led the participants through the app. All of the respondents were not cautious to fill out the diary and it was learning by doing. But they also indicated that they are already used to apps and have higher demands on the standard regarding the usability of an app. The loading time of an app and different devices with different operating versions can result in performance problems - especially for older mobile phones. A challenge the respondents faced were the 144 time slots. The aim to get more precise activities was hard to achieve. For the respondents it was difficult to differentiate between main and side activity. Persons who protocol at the evening could have help with a GPS-option. A list or a map of the places the person has been could be displayed and aid the recalling of activities. The app should give the participants immediate feedback. The detailed reconstruction of the day is burdensome and the users of an app are used to get an instant gratification. This gratification can be a graphic demonstration of the personal time for the different activities. But also the keyword "gamification" of apps should be taken into account without being childish or suspect. The app should make fun and give you some extra value immediately. In order to be close to the respondents` reality, questions could be adopted upon the done activities. Future research should cover this issue more in depth.

With the grant of 2019 it was possible to transform the App STAT_time technically in an open source version and internationalise it. The documentation should help other NSIs or interested Institutions to test STAT_time before it is used for a wider range of respondents.

For future versions we might consider also linking to other new technological developments like detection of the activity with sensors (sleeping time, etc.). There are already good Apps on the market that allow the user to track their online-time, their physical activities, the time the use for other Apps. They offer the possibility to track the time someone has used other on apps i.e. social media apps, mail-programs, or time for surfing on the internet. In fact many mobile phones offer this feature for their user weekly in a "screening time" summary. More features mean also a higher power consumption and leads to a performance issue of the app and of course it is a privacy issue linked with that.

Technically an App is never really ready because the stores constantly change requirement for upload and publication. Update of MacOS , Xcode, Android SDK, APIVersions or old version of libraries have to be replaced. Continuous development is necessary as frameworks update constantly.

Concluding it was a step forward to develop the next time use survey. In future it will be necessary to go more into the deep and concentrate more on the usability of the app. And before the fieldwork starts a new test phase should be definitely considered. Enough testing time is really necessary. Unforeseen bugs and challenges that can occur in the field phase need extra time for getting new

solutions. In general the project was again a valuable way to learn and to get new experiences in measuring time use with innovative tools.

9 List of References

EUROSTAT (Hrsg.) (2019): Harmonised European time use surveys; 2018 guidelines. Luxembourg: Office for Official Publications of the European Communities.

EUROSTAT (Hrsg.) (2009): Harmonised European time use surveys; 2008 guidelines. Luxembourg: Office for Official Publications of the European Communities.

EUROSTAT (Hrsg.) (2004): How Europeans spend their time; Everyday life of women and men. Luxembourg: Office for Official Publications of the European Communities.

Stiglitz, J. E., Sen, A., & Fitoussi, J. P. (2010). *Report by the commission on the measurement of economic performance and social progress*. Paris: Commission on the Measurement of Economic Performance and Social Progress.

United Nations (2005). *Guide to producing statistics on time use: measuring paid and unpaid work*. United Nations: New York. https://unstats.un.org/unsd/publication/SeriesF/SeriesF_93e.pdf

United Nations Secretariat (2012). *Report of the United Nations Expert Group Meeting on the Revision of the United Nations International Classification of Activities for Time Use Statistics (ICATUS)*, 11-13 June, 2012. United Nations: New York.

Statistic Austria (2017): Final Report “Preparatory work to revise the harmonised European Time Use survey for the implementation in the round 2020” of EUROSTAT-GRANT Agreement No. 07141.2016.001-2016.772.



AT_TUS_Grant_Report_2017_12_12.pdf

CSPA-compliant Documentation of STAT_time <https://www.statistical-services.org/>