Requirements for a Flexible and Generic API Enabling Mobile Crowdsensing mHealth Applications

Rüdiger Pryss, Johannes Schobel, Manfred Reichert
Institute of Databases and Information Systems, Ulm University, Germany
{ruediger.pryss, johannes.schobel, manfred.reichert}@uni-ulm.de

Abstract—Presently, mHealth becomes increasingly important in supporting patients in their everyday life. For example, diabetes patients can monitor themselves by the use of their smartphones. On the other, clinicians as well as medical researchers try to exploit the advantages of mobile technology. More specifically, mHealth applications can gather data in everyday life and are able to easily collect sensor or context data (e.g., the current temperature). Compared to clinical trials, these advantages enable mHealth applications to gather more data in a rather short time. Besides, humans often behave atypically in a clinical environment and, hence, mHealth applications collect data in a setting that reflects the daily behavior more naturally. Hitherto, many technical solutions emerged to deal with such data collection settings. Mobile crowdsensing is one prominent example in this context. We utilize the latter technology in a multitude of large-scale projects to gather data of several chronic disorders. In the TrackYourTinnitus project, for example, we pursue the goal to reveal new medical insights to the tinnitus disorder. We learned in the realized projects that a sophisticated API must be provided to cope with the requirements of researchers from the medical domain. Notably, the API must be able to flexibly deal with requirement changes. The work at hand presents the elicited requirements and illustrate the pillars on which our flexible and generic API is built on. Although we identified that the maintenance of such an API is a challenging endeavor, new data evaluation opportunities arise that are promising in the context of chronic disorder management.

Index Terms—mHealth, mobile crowdsensing, mobile data collection, chronic disorder, RESTful API, requirements, flexibility

I. Introduction

The assessment of many chronic disorders is still a challenging issue for researchers, medical doctors, and healthcare professionals. While researchers mainly focus on how to reveal valuable data sources, medical doctors crave for new therapies in their daily clinical practice. In line with these demands, the ongoing proliferation of mobile devices constitutes a trend that may shift the way how patient data is collected. In particular, the collection procedure can be accomplished in everyday life, which offers new opportunities. For example, for many individuals suffering from a chronic disorder, their symptoms vary over time. However, established assessment methods neither systematically assess this moment-to-moment variability [1] nor environmental factors having an effect on many chronic disorders. However, information of individual fluctuations and the effect of environmental factors might represent important information for new research insights or be the basis for a more individualized treatment. In this context, new technical solutions emerged that cope with the aforementioned assessment demands. One promising approach for collecting ecological valid longitudinal datasets at rather low costs constitutes mobile crowdsensing [2], [3]. However, mobile crowdsensing constitutes only one example for a powerful mHealth technology. Generally speaking, mHealth aims at the use of mobile and wireless technology to support the achievement of health objectives. Considering the way that mHealth applications collect data in everyday life, it is important to know that mHealth data sets show a so-called high external validity. This means, medically, that the collected data reflects the real-world situation in the best possible way.

In the TrackYourTinnitus project (TYT) [4], [5], we developed a mobile crowdsensing platform with the goal to reveal new medical insights to the tinnitus disorder [6]. Interestingly, by using mobile crowdsensing technologies, we were already able to reveal new medical insights [7]–[10]. In general, mobile crowdsensing platforms constitute a valuable target to help patients to demystify their chronic disorder and to get better control of it. Moreover, researchers of the medical domain can be provided with additional data sources.

So far, the crowdsensing technology used for TYT has been adjusted and it is currently used in four additional research projects (cf. Table I). In these projects, we tackle medical questions on the loss of hearing, the management of stress and diabetes as well as the support of pregnant women.

Note that the developed technology comprises a website for user registration, two mobile applications (for iOS and Android) and a relational database (MariaDB) as the central repository storing the collected data. Further note that this setting is used for all of the five projects listed in Table I. In addition, again, in all projects, a RESTful API governs

Project	Medical Aspect	URL
TrackYourTinnitus	Tinnitus	http://www.trackyourtinnitus.org
TrackYourHearing	Hearing Loss	http://www.trackyourhearing.org
TrackYourStress	Stress	http://www.trackyourstress.org
Chrodis+	Diabetes	http://chrodis.eu
MyKind	Pregnant Women	http://www.mykind.info

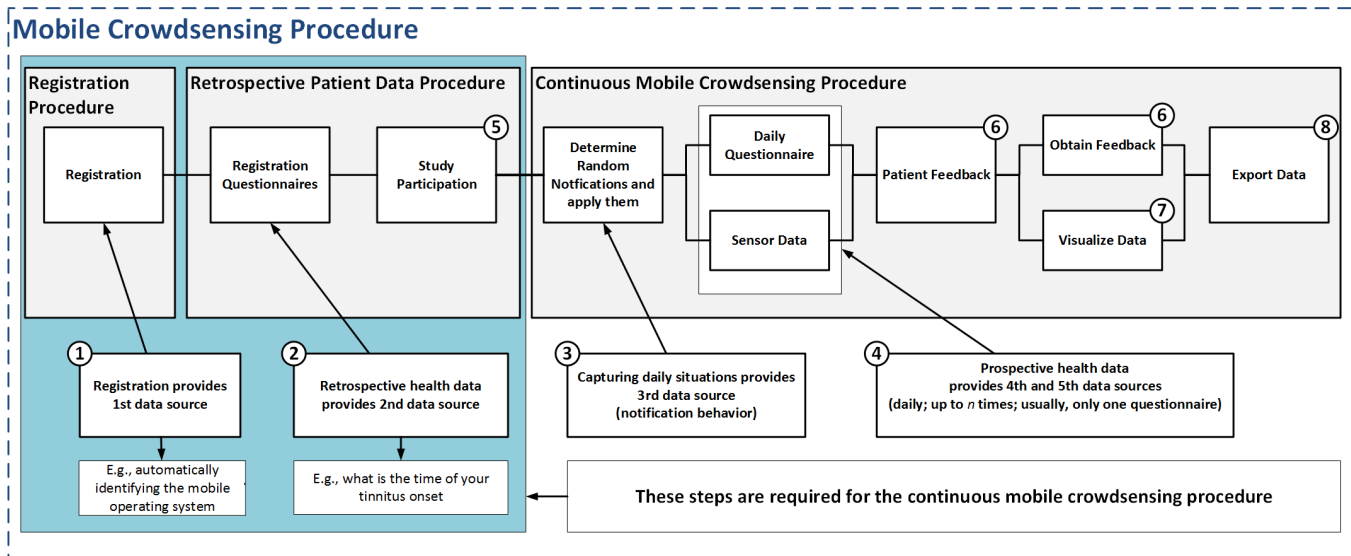TABLE I: Mobile Crowdsensing Projects

Fig. 1: Mobile Crowdsensing mHealth Data Collection Procedure

the communication between the aforementioned components. As the crowdsensing platform is used in different medical contexts, these projects particularly revealed that the design of the used API is crucial when maintaining the mobile applications to collect the data. On one hand, the frequent updates of the underlying mobile operating systems should be considered by the API design. On the other, the frequent requests of the medical domain experts to change existing features or add new ones must be considered when designing the API. For example, the medical doctors requested a feature to enable studies for different patient groups.

Based on the insights of the projects shown in Table I, as well as the ongoing efforts to improve the RESTful API, the work at hands presents a set of requirements that must be considered to maintain a mobile crowdsensing platform in the context of mHealth data. More specifically, the set of requirements can be used to realize the kind of crowdsensing features that allow for capturing the moment-to-moment variability of patients suffering from a chronic disorder or people that want to cope with their health condition on a daily basis.

The remainder of this paper is structured as follows. In Section II, we briefly illustrate the developed mobile crowdsensing platform along TrackYourTinnitus and explain the aspects that are relevant in the context of this paper. Section III presents the elicited requirements, while related work is discussed in Section IV. Finally, a summary and an outlook are provided in Section V.

II. MOBILE CROWDSENSING PLATFORM BACKGROUNDS

In this section, the features of the developed mobile crowdsensing platform, which are relevant for the requirements presented in this paper, will be briefly discussed. Besides the variety of used mHealth technologies, research in general is premature with respect to data that is gathered by utilizing smart mobile devices to achieve health objectives due to several reasons. First of all, it must be considered whether existing evaluation methods fit to this kind of gathered data. Important questions are, for example, how to cope with dropouts or how to deal with patients that provide data with challenging time gaps. However, methods like the Ecological Momentary Assessment (EMA; also known as: *ambulatory assessment & experience sampling*) have the potential to support clinicians in assessing symptoms or making diagnoses. In EMA, the variable in question (e.g., symptoms) is assessed repeatedly in daily life [11]. Repeatedly, in turn, not only means that the assessment is accomplished once a day, it includes that the assessment is done multiple times per day. Consequently, variations during the day can also be captured. Therefore, EMA is the basis of TYT and the other four realized crowdsensing projects to capture the moment-to-moment variability of the patients.

Technically, the developed crowdsensing technology is built on four components. First, it offers a website for user registration and other user-related features (e.g., data visualization). Second, it offers an Android and iOS mobile application. Third, a relational database (i.e., MariaDB) is used for the central repository for the data collected [5]. Fourth, a RESTful API is provided that enables the communication between the mobile applications, the website, and the database. In general, the technology was developed to track the individual daily situation of its registered users. The tracking is based on a set of existing and individually developed questionnaires. In addition, for some of the realized projects, sensor measurements are accomplished. For example, in TYT, the environmental sound level is measured when tracking the users. However, in TrackYourStress, the GPS position is measured. Regarding the practical use of the crowdsensing platform, the procedure depicted in Fig. 1 is accomplished by all crowdsensing users. More detailed information about the TYT procedure, for example, can be found at

https://www.trackyourtinnitus.org/process.pdf.

Following this procedure, the crowdsensing users accomplish three fundamental phases. First, they have to register (cf. Fig. 1, ①). This can be accomplished by using the website, the Android or the iOS app. Second, users have to fill in so-called registration questionnaires (cf. Fig. 1, ②). The latter capture the current health situation, demographic data (e.g., date of birth), and retrospective reports. For example, in TYT, users have to provide demographic data and fill in the "Mini-TQ-12" questionnaire [12], which measures tinnitus-related psychological problems. The completion of these registration questionnaires is a fundamental prerequisite for users who want to use the features of the continuous mobile crowdsensing procedure (cf. Fig. 1; Steps ③ and ④). During this phase, first of all, users have to accept or adjust a notification schema. The notification schema determines how often and in what way (i.e., fixed or random points in time) the daily assessment questionnaire is applied (cf. Fig. 1, ③). In TrackYourStress, for example, the number of daily assessments is restricted to 20 times per day. However, in TYT, it is restricted to 12 times per day. The maximum number of assessments, in turn, is individually specified for each of the presented projects by the healthcare professionals we are collaborating with. Note that the number of used assessment questionnaires also differs per project. In TYT, only one assessment questionnaire is used, whereas TrackYourStress uses 3 assessment questionnaires. In turn, the number of assessment questionnaires determines also the number of used notification schemata. Consequently, in TYT, only one schema is used, while TrackYourStress makes use of three schemata for each of the assessment questionnaires.

For the application of the assessment questionnaires, notification features for both Android and iOS, as well as a notification algorithm were realized [10]. The algorithm is used if the user decides for random notifications. Note that the latter is the default application setting. If random notifications are used, then up to the specified number and on a random basis, the user is notified to fill in the assessment questionnaires. This procedure particularly ensures that users cannot foresee the time when being asked, on the other it ensures that they are captured in various daily situations. Note that this randomized approach was implemented to realize EMA. The approach to randomly apply the assessment questionnaires is illustrated by Fig. 2. It works on both mobile operating systems in exactly the same ways. After a notification appears, the user may click on it. In the latter case, the mobile application is started (if not already running) and the assessment questionnaire is directly displayed to the user. Then, he or she can fill in the questionnaire and finally save the entered data. After saving the questionnaire data, the mobile application proceeds in different ways, depending on the project and the decision of the healthcare professionals. In TYT, for example, the mobile application is terminated 3 seconds later after saving the assessment questionnaire. Within these 3 seconds, the result is transferred to the backend if the mobile app is online; otherwise the result is locally stored until the device gets
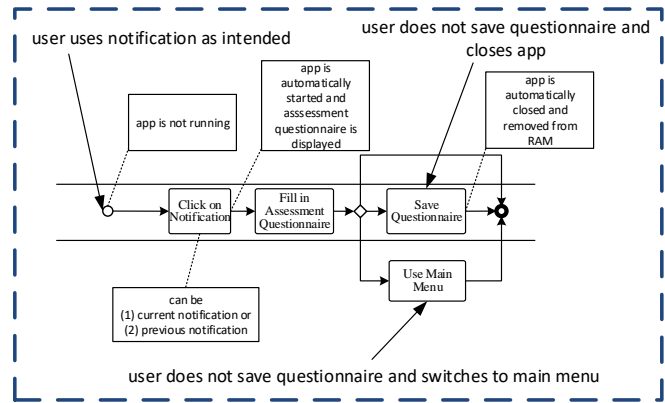

Fig. 2: Possible User Action after a Notification

an online connection. In TrackYourStress, in turn, the mobile app proceeds to the main menu of the mobile application. Again, the decision is made by the responsible healthcare professionals.

As can be also seen from Fig. 2, the user might click on a current notification or previous ones. The latter situation might occur if the user did not click on a previous notification. Each time a notification is not clicked, the mobile application memorizes and stores it in the local message store. If a user clicks on the local message store, he or she may see these previous notifications. Note that the allowed amount of *previous* notifications messages differs between the two mobile operating systems. As we store separate timestamps for the notification as well as the saving time of each processed assessment questionnaire, users can also fill in questionnaires related to previous notifications.

With respect to the API requirements, some additional aspects have to be briefly discussed. First, as can be obtained from Fig. 1, 5 data sources to evaluate user data are provided: Data that can be gathered during the (1) registration, (2) when filling in the registration questionnaires, (3) when analyzing whether or not users actually use the random notifications to fill in the assessment questionnaires, (4) when analyzing data provided by the assessment questionnaires, (5) and data that is coming from the sensor measurements. The latter is accomplished while filling in the assessment questionnaires. For example, in TrackYourStress, the smart mobile device of a user records the GPS position of the user if this option has been activated. These data sources, in turn, are the fundament that characterizes the opportunities of our mobile crowdsensing mHealth platform. For example, results on prospective reports vs. retrospective ratings of tinnitus variability and tinnitus-stress associations can be obtained [10]. In general, only little is known to what degree such retrospective reports reflect the actual experiences made in everyday life. This shows the potential of EMA, mobile crowdsensing and the applied procedure we developed as shown by Fig. 1. Moreover, four more features were integrated into the platform.

1) Users can enter studies (cf. Fig. 1, ⑤). When partici-

pating in a study, the gathered user data is assigned to a study. In addition, studies can have their own notification schemata. The latter, in turn, can be fixed (i.e., user cannot change it) or pre-specified by a helathcare professional. In addition, it can be determined whether or not a user can change the schema.

2) Users can be provided with feedback (cf. Fig. 1, ⑥). Thereby, the API offers features to automatically calculate feedbacks based on the five discussed data sources.

3) Users can visualize their gathered data using the website or the mobile applications (cf. Fig. 1, ⑦). This required the API feature to obtain the required data from the database as the mobile applications do not store the processed data permanently.

4) Users can export their data to a CSV file (cf. Fig. 1, ⑧). Again, a corresponding feature was integrated into the API.

Another important issue for the API constitutes the management of questionnaires. Notably, the projects we conduct with the help of the developed mobile crowdsensing platform address different medical aspects (cf. Table I). Therefore, the used questionnaires differ by magnitude of orders. Consequently, the API must provide, in the best case, a generic approach to manage the different medical settings. In addition, if swift changes to the user feedback calculations are necessary, the API should cope with this requirement as well. Many other examples occurred when using the mobile crowdsensing platform in practice. Moreover, the crowdsensing platform provides multilingual support. The offered languages, in turn, depend on the needs of the project. In TYT, currently, English, German and Dutch are supported.

Finally, to give an impression how the platforms are used in practice, we provide some data on the usage of TYT. Regarding the latter, the current number (July 2018) of registered users is 3403 and they are coming from over 80 countries. Thereby, 822 of the users provide the major part of the data, which are 56401 filled out assessment questionnaires. It is further noteworthy that users from seven countries provide the major part of all data gathered. To be more precise, most of the collected data is provided by users from Germany, followed by the US, Netherlands, the UK, Switzerland, Canada, and France.

III. SET OF API REQUIREMENTS

Based on the procedure shown in Fig. 1, the API for the mobile crowdsensing platform was designed and realized. Technically, we decided to build the API based on three pillars. First, the API shall follow the REST architectural style. Second, for exchanging data between the API, the mobile applications, and the website, JSON shall be used. Third, the Laravel framework [13] shall be the basis to implement the API. This section presents the set of elicited requirements to provide a powerful API that enables the illustrated mobile crowdsensing procedure (cf. Fig. 1). On top of that, we briefly convey how to cope with flexibility demands on one hand and the provision of a generic API that can be used in different
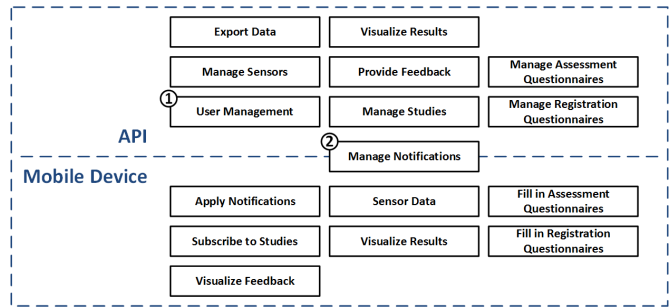

Fig. 3: Set of Realized Features

contexts (cf. Table I) on the other. Regarding the set of features that are provided by the API, consider Fig. 3. It reflects the current feature set of the developed API. In addition to already mentioned features, two more aspects are shown in Fig. 3. First, the API must provide a proper user management (cf. Fig. 3, ①). Second, the management of notifications is a joint feature between the mobile applications and the API (cf. Fig. 3, ②). That means, some features to apply the notifications as presented are not provided by the API and the mobile applications manage these features themselves. For example, the algorithm to calculate the notifications is solely executed on the mobile device. So far, we decided not to transfer the calculation results back to the API. However, each data set of the assessment questionnaires that is transferred to the API has a field that stores the timestamp when the user was notified. In addition, we manage a field that stores the timestamp when a data set was saved in the MariaDB database.

Based on the mobile crowdsensing procedure shown in Fig. 1 and the discussed features shown in Fig. 3, the set of requirements, which is currently addressed by the API, were derived (cf. Table II). Note that we omit a presentation how data is anonymized and secured. However, these aspects are also of utmost importance and considered by the API.

While maintaining the mobile crowdsensing platforms shown in Table I, we experienced change requests to these set of requirements by the healthcare professionals on a frequent basis. For example, a request emerged that the registration questionnaires shall appear again after a certain period that is depending on the amount of filled in assessment questionnaires by a user. Another request emerged by integrating gamification aspects that award the amount of processed questionnaires by
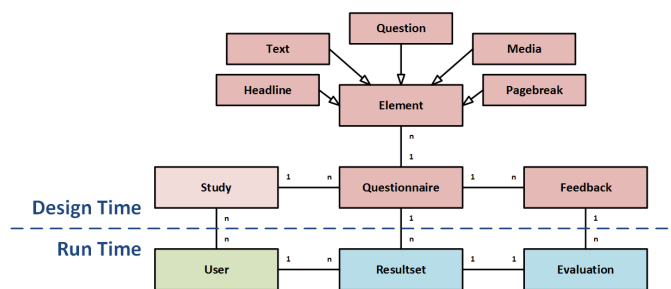

Fig. 4: Data Model

Requirement	Description
User Management	The API must support a feature that users can register to the platform, change their personal data, remove themselves, or reset their password.
Study Management	The API must support a feature that users can be assigned or removed from studies.
Export Management	The API must support a feature that users and healthcare professionals can export data to a CSV file.
Sensor Management	The API must support a feature that data from the assessment questionnaires can be enriched with data that is gathered by a sensor. In addition, it must be possible to specify that studies require a sensor measurement. Consequently, only users that have this sensor built in in their smart mobile device and actually accept the measurement can participate in such studies.
Notification Management	The API must support a feature that notification settings can be obtained by the mobile applications.
Questionnaire Management	The API must support a feature that users can fill in registration and assessment questionnaires. The feature must also consider (1) that the assessment questionnaire can only be used after the corresponding registration questionnaires are completed, (2) the different types of notifications, (3) the studies, and finally (4) the sensor measurement.
Data Visualization	The API must support a feature that users can visualize their gathered data.
Feedback Management	The API must support a feature that healthcare professionals can provide feedback to users. More specifically, it must be possible to add rules or calculations that (1) determine user feedback only based on the data of the user that requests the feedback or (2) determine user feedback based on a comparison with data of other (i.e., peer) users.

TABLE II: Set of API Requirements

a user. Although our realized mobile crowdsensing platform is beyond the status to cope with all these requirements, the data model shown in Fig. 4 revealed its benefits in this context. It enables us to flexibly cope with arising requirements on one hand and allows for the usage of heterogeneous questionnaires on the other. Notably, as can be seen in Fig. 4, we maintain only one questionnaire type. However, to cope with the requirements of the healthcare professionals, a questionnaire can comprise various types of elements. These types, in turn, are used by the mobile applications to dynamically render a questionnaire. This enables us to change questionnaires without changing the implementation of the mobile applications.

To get a better impression of the used data model, Listing 1 shows an example for a JSON document that is used for TYT. The shown JSON document illustrates the meta data of the questionnaire object used in our mobile crowdsensing platforms.

Listing 1: JSON Example of Questionnaire Metadata

```
1  GET /api/v1/questionnaires/6?token=TOKEN
2  HTTP/1.1 Host: www.tyt.dev Accept-Language: en
3  {
4      "data": {
5          "type": "questionnaires",
6          "id": "6",
7          "attributes": {
8              "name": "TrackYourTinnitus",
9              "title": "TrackYourTinnitus",
10             "origin": "internal",
11             "is_active": 1,
```

```
12             "can_use": [
13                 {
14                     "name": "gps",
15                     "type": "amount",
16                     "value": 3
17                 },
18                 {
19                     "name": "microphone",
20                     "type": "each",
21                     "value": 5
22                 }
23             ],
24             "is_onetime": 0,
25             "is_multiple": 1,
26             "points": 0,
27             "schedule": [
28                 {
29                     "type": "between",
30                     "every": 1,
31                     "start_day": 0,
32                     "start": "08:00:00",
33                     "end": "20:00:00",
34                     "amount": 5
35                 }
36             ],
37             "is_schedule_changeable": 1,
38             "description": "",
39             "introtext": "",
40             "outrotext": ""
41         },
42         "meta": {
43             "likes": 0,
44             "avg": 0
45         },
46         "links": {
47             "self": "www.tyt.dev/api/v1/questionnaires/6"
48         }
49     },
50     "meta": {
51         "count": 1,
52         "include": [],
53         "translated": [
54             "title",
55             "description",
56             "introtext",
57             "outrotext"
58         ]
59     }
60 }
```

When using this object in practice, the general connection procedure between the smart mobile device and the API is as follows:

- After the first time a user enters the application, the smart mobile device is requesting from the API whether more than the default study exist. If so, the user can choose the study he or she wants to enter. If the user does not choose a study even one exists, he or she is enrolled into the default study. If only the default study exists, the user cannot decide which study should be entered. Note that a user can be enrolled in more than one study.

- Then, based on the chosen study, the mobile application requests from the API whether registration questionnaires exist, which is based on the flag $is\_onetime$. If this flag indicates 1 for a questionnaire of the study the user is enrolled to, then a registration questionnaire was found. This decision must be obtained by the smart mobile device for all questionnaires of a study. Therefore, the API provides a feature that provides all questionnaires of a study.

- If registration questionnaires exist, then the mobile device requests the structure (i.e., what elements of Fig. 4 are used) and the appearance (e.g., slider usage for the answer of a question) of all these questionnaires from the API. Listing 2 shows four element examples of a TYT questionnaire. More specifically, it shows (1) a headline element as well as a (2) question for which the answer shall be presented as a $SingleChoice$ user interface element on the smart mobile device of a user. Furthermore, (3) a question for which the answer shall

be rendered as a *Slider* user interface element as well as (4) a question for which the answer shall be rendered as a *SAMScaleFace* user interface element on the smart mobile device of the user. Note that the *SAMScaleFace* user interface element represents a particular scale (i.e., the Self-Assessment Manikin scale; SAM) that is often used in psychology [14]. Further note that the Android and iOS applications of the developed mobile crowdsensing platform use this *questiontype* information to dynamically render the (registration) questionnaires.

- Then, the mobile application ensures that all registration questionnaires must be completed before proceeding with the mobile crowdsensing procedure (cf. Fig. 1).
- For the mobile crowdsensing procedure, again, it is requested which questionnaires exist for the enrolled studies. This is indicated by the flag *is_multiple*. If the flag indicates 1, then an assessment questionnaire is found. Based on this information, the mobile device accomplishes two actions. First of all, for each assessment questionnaire, the smart mobile device offers a notification schema. Second, for each assessment questionnaire, notifications will be created based on the specified notification schema by the smart mobile device user.

Based on this procedure, we basically ensure a high flexibility of the realized API for the various types of mobile crowdsensing projects with different EMA requirements. More specifically, the following aspects ensure this flexibility. First of all, questionnaires – independently of being a registration or assessment questionnaire – are locally downloaded to the mobile device. The rendering of the questionnaires is dynamically accomplished based on the *questiontype* information we have presented. However, each time the mobile applications are started, the mobile device determines (i.e., by using features of the API) whether changes to the questionnaires have been made or not. This may occur if the involved healthcare professionals want to adjust an ongoing study. Then, the adjusted questionnaires are downloaded by the smart mobile device. If a registration questionnaire was changed, then the mobile application goes back to the registration questionnaire procedure, *iff* new questions have been added or existing ones require more information for an answer. If an assessment questionnaire has been changed or added, then the mobile applications also dynamically adapts to the new situation. These dynamic adaptations have three important prerequisites, which are satisfied by the API:

- The information how the questionnaire shall be rendered is only determined by information that are stored by the API. For user interface elements that cannot be satisfied by default elements of the underlying mobile operating system (e.g., SAM scale), we have developed respective features for Android and iOS.
- If questionnaires are changed, we manage the version of a questionnaire. This way, the healthcare professional is able to determine which data set belongs to which questionnaire version.

- Only based on the two flags *is_onetime* and *is_multiple* (including the data model shown in Fig. 4) of the questionnaire object, the powerful procedure shown in Fig. 1 can be flexibly ensured.

Altogether, the set of requirements and pursued procedures presented in this section, combined with a generic data model for questionnaires, revealed to be a reasonable and flexible solution to maintain mobile crowdsensing platforms that gather mHealth data in the scope of EMA.

Listing 2: JSON Example of Questionnaire (Excerpt)

```
 1 GET
 2 /api/v1/questionnaires/6/structure?token=TOKEN HTTP/1.1 Host: www.tyt.dev
 3 Accept-Language: en
 4 ...
 5 {
 6     "data": [
 7         {
 8             "type": "elements/elements",
 9             "id": "80",
10             "attributes": {
11                 "content": {
12                     "headline": "Track Your Tinnitus Questionnaire"
13                 },
14                 "name": "Headline01",
15                 "elementtype": "elements/headlines"
16             },
17             "links": {
18                 "self": "www.tyt.dev/api/v1/elements/elements/80"
19             }
20         },
21         {
22             "type": "elements/elements",
23             "id": "81",
24             "attributes": {
25                 "content": {
26                     "question": "Did you perceive the tinnitus right now?",
27                     "answers": [
28                         "Yes",
29                         "No"
30                     ],
31                     "label": "s01",
32                     "questiontype": "SingleChoice",
33                     "values": [
34                         "YES",
35                         "NO"
36                     ],
37                     "condition": null,
38                     "required": 1,
39                     "restricted_to": null
40                 },
41                 "name": "Q01",
42                 "elementtype": "elements/questions"
43             },
44             "links": {
45                 "self": "www.tyt.dev/api/v1/elements/elements/81"
46             }
47         },
48         {
49             "type": "elements/elements",
50             "id": "83",
51             "attributes": {
52                 "content": {
53                     "question": "How stressful is the tinnitus right now?",
54                     "answers": [
55                         {
56                             "value": 0,
57                             "label": "not stressful"
58                         },
59                         {
60                             "value": 100,
61                             "label": "maximal stressful"
62                         }
63                     ],
64                     "label": "s03",
65                     "questiontype": "Slider",
66                     "values": {
67                         "min": 0,
68                         "max": 100,
69                         "step": 1
70                     },
71                     "condition": null,
72                     "required": 0,
73                     "restricted_to": null
74                 },
75                 "name": "Q03",
76                 "elementtype": "elements/questions"
77             },
78             "links": {
79                 "self": "www.tyt.dev/api/v1/elements/elements/83"
80             }
81         },
82         {
83             "type": "elements/elements",
84             "id": "84",
85             "attributes": {
86                 "content": {
87                     "question": "How is your mood right now?",
88                     "answers": [],
89                     "label": "s04",
```

```
90                    "questiontype": "SAMScaleFace",
91                    "values": {
92                        "min": 1,
93                        "max": 9,
94                        "step": 1
95                    },
96                    "condition": null,
97                    "required": 0,
98                    "restricted_to": null
99                },
100                "name": "Q04",
101                "elementtype": "elements/questions"
102            },
103            "links": {
104                "self": "www.tyt.dev/api/v1/elements/elements/84"
105            }
106        },
107        {
108            "type": "elements/elements",
109            "id": "85",
110            "attributes": {
111                "content": {
112                    "question": "How is your arousal right now?",
113                    "answers": [],
114                    "label": "s05",
115                    "questiontype": "SAMScaleBody",
116                    "values": {
117                        "min": 1,
118                        "max": 9,
119                        "step": 1
120                    },
121                    "condition": null,
122                    "required": 0,
123                    "restricted_to": null
124                },
125                "name": "Q05",
126                "elementtype": "elements/questions"
127            },
128            "links": {
129                "self": "www.tyt.dev/api/v1/elements/elements/85"
130            }
131        },
132 ...
133 }
```

Finally, we conceive the architecture of the API (cf. Fig. 5). It API comprises 7 core layers. The first layer governs the authentication and authorization (cf. Fig. 5, ①). The second layer manages the requests from the mobile applications and the website as well as the corresponding responses (cf. Fig. 5, ②). The third and fourth layers (cf. Fig. 5, ③ & ④) manage the studies and the users, while the fifth layer enables the feedback feature (cf. Fig. 5, ⑤). Finally, layers six and seven manage the persistent storage of the data collected.

IV. RELATED WORK

In this section, we discuss prior research on mobile crowdsensing, EMA, mHealth, and works that discuss requirements of a technical solution in this context. Mobile crowdsensing has become an important research topic in various scenarios [2], [15]. However, mobile crowdsensing applications less exist in the medical domain although their use is promising for many questions [3], [7], [16], [17]. One reason might be that many obstacles still exist or that they are less understood. On top of this, to maintain a mobile crowdsensing platform that is able to gather data with different mobile operating systems is a challenging endeavor. In the context of EMA, other studies exist that use this method in the context of chronic disorders [18], [19]. Moreover, further aspects are investigated using EMA approaches [20]–[23]. In general, EMA approaches are considered to offer unprecedented opportunities to study clinical symptoms under ecologically valid conditions [24], even though the utilization of its possibilities is still premature.

Regarding the elicitation of requirements for a mobile crowdsensing platform, including a discussion of API requirements, must be distinguished into two dimensions in the context of this paper. First, related works must be considered that discuss such requirements for mobile crowdsensing in general.
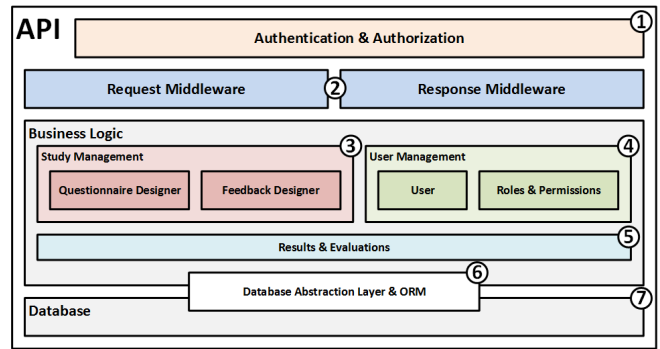

Fig. 5: API Architecture

Second, related works must be considered that particularly address mobile crowdsensing in the context of mHealth and chronic disorders. Concerning the first dimension, there exist lot of works that discuss the design of mobile crowdsensing architectures [16], [17], [25], [26]. Interestingly, only few works elicit requirements in this context [27]. In turn, an elicitation of requirements for an API as provided in this work is not presented in these works. Concerning the second dimension, less works can be found that focus on a combination of mobile crowdsensing, mHealth, and API requirements. For example, in [28], a mobile crowdsensing approach is discussed to conduct studies related to healthcare questions. Again, a detailed discussion of elaborated requirements is not provided.

Furthermore, works exist that cope with mHealth applications in more general. In [29], for example, the peculiarities of mobile operation systems are discussed for clinical interventions that are applied by the use of mobile applications. The authors describe that it costs a lot of time to cope with the different ways to develop mHealth applications. In addition, they state that it is naturally an interdisciplinary endeavor. In the context of mobile data collection for medical purposes, also approaches exist [30], [31] that discuss peculiarities of different mobile operating systems for the data collection procedure. However, these approaches do not present requirements how an API should be realized in this context. Furthermore, many works exist that aim to evaluate the differences and quality of mHealth applications that were developed for both mobile operating systems [32]–[34]. Again, a discussion of how to realize a proper API in this context is not provided. In addition, in [4], [5], the authors presented the overall architecture and its services used for TYT. However, the features implemented in the API as well as aspects of the data model were not presented in the aforementioned works [4], [5]. Altogether, the use of mobile crowdsensing in the context of mHealth data is still in its infancy, although the opportunities are very promising on the other.

V. SUMMARY AND OUTLOOK

This paper presented a set of requirements for a RESTful API, which was developed for mobile crowdsensing platforms that enable medical researchers to gather mHealth data in the context of EMA. We presented the advantages of mobile

crowdsensing in this context and discussed what we learned from our conducted projects. Based on this, the revealed mobile crowdsensing procedure was discussed and shown that it constitutes the basis for the realized API. A set of requirements for the API was presented and on top of this discussion, it was illustrated how an API that meet these requirements can be enhanced to flexibly cope with frequent requirement changes of the medical researchers. To be more specific, a proper data model is necessary to deal with these types of change requests. The way how such a data model may look like was illustrated along our used one. Moreover, we conceived the architecture of our flexible and generic API for mobile crowdsensing mHealth applications. We further discussed that technology as presented in the work at hand is still in its infancy. However, as the technology already showed for TYT that new medical insights can be revealed, an API and its requirements as shown here may be of interest for many researchers in future. Currently, we work on further technical API aspects. In addition, we conduct experiments to identify respective API metrics (e.g., performance, changeability). Furthermore, in the context of mobile crowdsensing, both from the medical as well as the technical perspective, differences between the mobile users can be also a promising question to answer [35]. Based on the latter work, we currently apply machine learning techniques to learn more about the users and their characteristics. Altogether, mobile crowdsensing technology constitutes a valuable target in the context of mHealth in general and chronic disease management combined with EMA in particular.

REFERENCES

[1] W. Schlee, R. Pryss, T. Probst, J. Schobel, A. Bachmeier, M. Reichert, and B. Langguth, "Measuring the moment-to-moment variability of tinnitus: the TrackyourTinnitus smart phone app," *Frontiers in Aging Neuroscience*, vol. 8, 2016.

[2] L. Shu, Y. Chen, Z. Huo, N. Bergmann, and L. Wang, "When Mobile Crowd Sensing Meets Traditional Industry," *IEEE Access*, 2017.

[3] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[4] R. Pryss *et al.*, "Mobile Crowdsensing Services for Tinnitus assessment and Patient Feedback," in *Int'l Conf on AI & Mobile Services*. IEEE, 2017, pp. 22–29.

[5] ——, "Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research," in *4th Int'l Conf on Mobile Services*. IEEE Computer Society Press, 2015, pp. 352–359.

[6] B. Langguth *et al.*, "Tinnitus: causes and clinical management," *The Lancet Neurology*, vol. 12, no. 9, pp. 920–930, 2013.

[7] T. Probst *et al.*, "Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the "TrackyourTinnitus" application," *Scientific reports*, vol. 6, 2016.

[8] ——, "Emotion dynamics and tinnitus: Daily life data from the "TrackyourTinnitus" application," *Scientific reports*, vol. 6, 2016.

[9] R. Pryss *et al.*, "Mobile Crowdsensing for the Juxtaposition of Realtime Assessments and Retrospective Reporting for Neuropsychiatric Symptoms," in *IEEE 30th Int'l Symp on Computer-Based Medical Systems*. IEEE, 2017, pp. 642–647.

[10] ——, "Prospective crowdsensing versus retrospective ratings of tinnitus variability and tinnitus–stress associations based on the TrackYourTinnitus mobile platform," *International Journal of Data Science and Analytics*, 2018.

[11] T. Trull and U. Ebner-Priemer, "Ambulatory assessment," *Annual review of clinical psychology*, vol. 9, pp. 151–176, 2013.

[12] W. Hiller and G. Goebel, "Rapid assessment of tinnitus-related psychological distress using the Mini-TQ," *Int J Audiol*, vol. 43, no. 10, pp. 600–604, 2004.

[13] X. Chen *et al.*, "Restful API Architecture Based on Laravel Framework," in *Journal of Physics: Conference Series*, vol. 910, no. 1. IOP Publishing, 2017, p. 012016.

[14] J. Morris, "Observations: Sam: the self-assessment manikin; an efficient cross-cultural measurement of emotional response," *Journal of advertising research*, vol. 35, no. 6, pp. 63–68, 1995.

[15] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *12th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2015, pp. 136–144.

[16] M. Demirbas *et al.*, "Crowd-sourced sensing and collaboration using twitter," in *International Symposium on a World of Wireless Mobile and Multimedia Networks*. IEEE, 2010, pp. 1–9.

[17] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[18] M. Wilson *et al.*, "Ecological momentary assessment of tinnitus using smartphone technology a pilot study," *Otolaryngology–Head and Neck Surgery*, p. 0194599815569692, 2015.

[19] J. Henry *et al.*, "Pilot study to evaluate ecological momentary assessment of tinnitus," *Ear and hearing*, vol. 32, no. 2, p. 179, 2012.

[20] R. Jamison *et al.*, "Electronic diaries for monitoring chronic pain: 1-year validation study," *Pain*, vol. 91, no. 3, pp. 277–285, 2001.

[21] M. Killingsworth and D. Gilbert, "A wandering mind is an unhappy mind," *Science*, vol. 330, no. 6006, pp. 932–932, 2010.

[22] S. Wenze and I. Miller, "Use of ecological momentary assessment in mood disorders research," *Clinical psychology review*, vol. 30, no. 6, pp. 794–804, 2010.

[23] P. Santangelo, M. Bohus, and U. Ebner-Priemer, "Ecological momentary assessment in borderline personality disorder: a review of recent findings and methodological challenges," *Journal of Personality Disorders*, vol. 28, no. 4, pp. 555–576, 2014.

[24] I. Myin-Germeys *et al.*, "Experience sampling research in psychopathology: opening the black box of daily life," *Psychological medicine*, vol. 39, no. 9, p. 1533, 2009.

[25] Y. Xiao *et al.*, "Lowering the barriers to large-scale mobile crowdsensing," in *Proc of the 14th WS on Mobile Computing Systems and Applications*. ACM, 2013, p. 9.

[26] X. Hu *et al.*, "Multidimensional context-aware social network architecture for mobile crowdsensing," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 78–87, 2014.

[27] ——, "Vita: A crowdsensing-oriented mobile cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 148–165, 2013.

[28] H. Xiong *et al.*, "Sensus: a cross-platform, general-purpose system for mobile crowdsensing in human-subject studies," in *Proc of the Int'l Joint Conf on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 415–426.

[29] D. Ben-Zeev *et al.*, "Strategies for mHealth Research: Lessons from 3 Mobile Intervention Studies," *Administration and Policy in Mental Health and Mental Health Services Research*, vol. 42, no. 2, pp. 157–167, 2015.

[30] J. Schobel, R. Pryss, M. Schickler, and M. Reichert, "Towards flexible mobile data collection in healthcare," in *29th International Symposium on Computer-Based Medical Systems*. IEEE, 2016, pp. 181–182.

[31] J. Schobel *et al.*, "Development of Mobile Data Collection Applications by Domain Experts: Experimental Results from a Usability Study," in *29th Int'l Conf on Advanced Information Systems Engineering*, ser. LNCS, no. 10253. Springer, 2017, pp. 60–75.

[32] A. Azfar, K. Choo, and L. Liu, "Forensic Taxonomy of Popular Android mHealth Apps," *arXiv preprint arXiv:1505.02905*, 2015.

[33] H. Seabrook *et al.*, "Medical Applications: a Database and Characterization of Apps in Apple iOS and Android Platforms," *BMC research notes*, vol. 7, no. 1, p. 573, 2014.

[34] B. Zapata *et al.*, "Empirical Studies on Usability of mHealth Apps: A Systematic Literature Review," *Journal of medical systems*, vol. 39, no. 2, p. 1, 2015.

[35] R. Pryss *et al.*, "Differences between Android and iOS Users of the TrackYourTinnitus Mobile Crowdsensing mHealth Platform," in *Proc. 31th IEEE International Symposium on Computer-Based Medical Systems*. IEEE Computer Society Press, 2018.