Universität Ulm | 89069 Ulm | Germany

Faculty of
Engineering, Computer
Science and Psychology
Databases and Information
Systems Department

Conception and realization of a mobile data acquisition and assistance application for intersession processes of patients in psychotherapeutic treatments at the example of the iOS platform.

Master's thesis at Universität Ulm

Submitted by:
Carsten Vogel
carsten.vogel@uni-ulm.de

Reviewer:
Prof. Dr. Manfred Reichert
Dr. Rüdiger Pryss

Supervisor:
Dr. Rüdiger Pryss

2019

Version from August 12, 2019

Composition: PDF-LaTeX 2_ε

Abstract

Conventional effectiveness and impact factor studies in psychotherapy research deal mainly with the therapy session per se. In contrast, a current trend is the increasing focus on patient advancement between therapy sessions, the so-called intersession processes. Traditionally, patient data is collected and evaluated in the form of paper questionnaires. In the context of intersession research, where this is done just prior to the therapy session, this means that their results often can not be properly included right afterwards. With the proliferation of mobile devices such as smartphones, tablet computers, and wearables, mobile crowd sensing is a promising approach for capturing and analyzing large amounts of distributed data. This is attributed to the fact that modern mobile devices are equipped with unprecedented sensing, computing, and communication capabilities that allow them to perform complex tasks and provide countless possibilities for user interactions. Contemporaneous, in the course of digitization, both the topic of electronic health and mobile health (mHealth) are gaining increasingly more importance in the healthcare industry. Furthermore, simple and efficient interaction with mobile applications, as well as the exchange of information between health care provider, here the therapist, and the patients, are essential aspects in applications in the mHealth field. Properly implemented, this can both improve and simplify the patient's treatment process.

Within the scope of this thesis, in cooperation with the Institute of Psychology of the University of Klagenfurt, an mHealth application is developed, which allows to scientifically record intersession processes of patients in psychotherapeutic treatments. The patient automatically receives questionnaires via the mobile application, depending on therapy session dates and the results of previous evaluations, as well as manual interventions by the therapist. Thus, it should be significantly more easy and efficient for the therapist to collect and evaluate data on the patient's intersession processes and to prepare in advance for the upcoming therapy session.

Acknowledgments

At this point, I would like to thank all those involved, who have contributed to the success of this master's thesis through their professional and personal support. I appreciate the great cooperation between the members of the Intersession-Online team. Particularly noteworthy Mag. Thorsten-Christian Gablonski, representing the Department of Clinical Psychology, Psychotherapy and Psychoanalysis of the University of Klagenfurt, who advised me with valuable professional input, during the implementation and written elaboration. Above all, a very special thanks to my supervisor Rüdiger Pryss, from the Institute for Databases and Information Systems of the University of Ulm, who took care of me excellently over the period of my master's thesis, starting with the topic selection, to the submission of the drafting. Also, my thanks to all those who have participated in the testing phase of the resulting iOS application and thereby contributed to the success of this work. I would also like to thank Prof. Manfred Reichert from the Institute for Databases and Information Systems, for the approval of the work and the opportunity to write my master's thesis in this subject area. Last but definitely not least a big thank you to all my family members and closest friends, who always actively supported me during this time and gave me many tips and suggestions.

Contents

1

Introduction

The number of smartphone users trend displays a steady increase globally, which offers a wealth of new possibilities. For example, eMarketer forecasts that by the year 2020 in Germany alone there will be around 65 million, as shown in Figure 1.1 [1] and even 2.9 billion Smartphone users worldwide, as shown in Figure 1.2 [2].

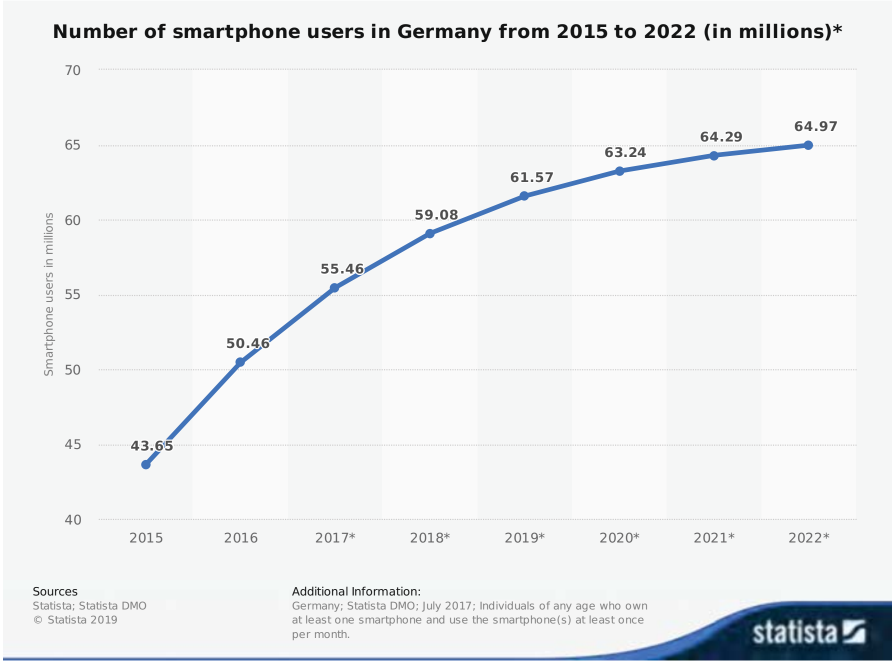
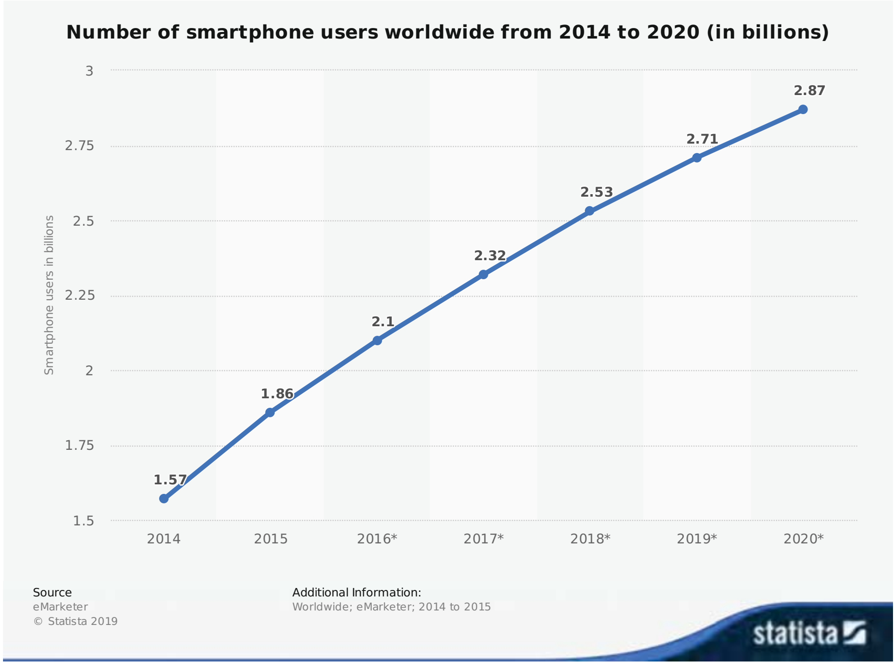


Figure 1.1: Number of smartphone users in Germany from 2015 to 2022 [1]

Contemporaneous, due to several factors, psychological assessments and interventions from the clinic extend further into everyday life. On one side, in the course of digitization and advances in mobile technology, both the topic of electronic health (eHealth) and



Source
eMarketer
© Statista 2019

Additional Information:
Worldwide; eMarketer; 2014 to 2015

Figure 1.2: Number of smartphone users worldwide from 2014 to 2020 [2]

mobile health (mHealth) are gaining increasingly more importance in the healthcare industry. On the other side, constrained clinical care, and consumer demand for contextualized, nonstigmatizing, and low-cost alternatives are beginning to change the face of psychological assessment and interventions. Mobile, social, and wearable technologies are now enabling individuals to measure and observe themselves, as well as to integrate countless forms of assistance. The massive data sets generated by self-tracking of several factors may eventually reorganize taxonomies of mental health concerns. Ultimately mobile therapies will emerge, involving contextually appropriate, appealing, and even dynamic feedback to provide support in the context of daily life. [3] [4]

Traditionally, data is collected in the form of paper questionnaires. In specific cases, for example the intersession research, this is done just prior to the therapy session, which means that their results often can not be properly included immediately. With the proliferation of mobile devices such as smartphones, tablet computers, and wearables, mobile crowd sensing (MCS) is a promising approach for capturing and analyzing large amounts of distributed data. This is attributed to the fact that modern mobile devices are equipped with unprecedented sensing, computing, and communication capabilities that allow them to perform complex tasks and provide countless possibilities for user interactions. [4]

Furthermore, simple and efficient interaction with mobile applications, as well as the exchange of information between health care provider (HCP), here the therapist, and the patients, are essential aspects in applications in the mHealth field. Properly implemented, this can both improve and simplify the patient's treatment process. The global digital health market in general is on the rise. Market researches predict a tremendous increase in the mHeath sector, based the estimation in Figure 1.3 [5] the value will double till 2021 and more than sixfold till 2025. This is also supported by the number of mHealth app downloads in the past years, raised by research2guidance, illustrated in Figure 1.4 [6]. Accourding to research2guidance one of the most attractive healthcare sectors for mHealth is mental health as shown in Figure 1.5 [7]. The work on psychotherapy between sessions is described by the intersession process, representing a relatively young field of research of the psychotherapeutic sciences. Intersession processes refer

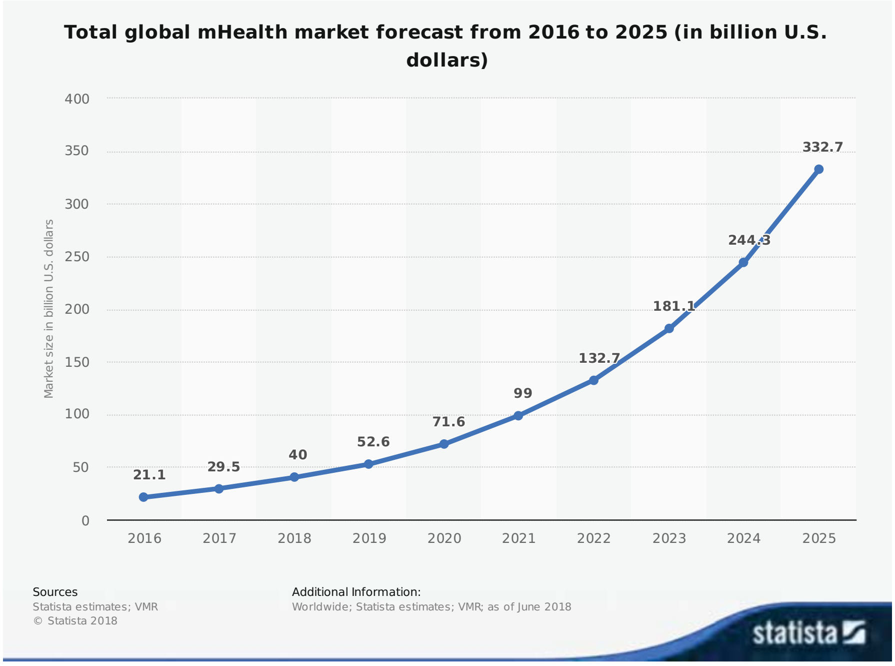


Figure 1.3: Total global mHealth market forecast from 2016 to 2025 [5]

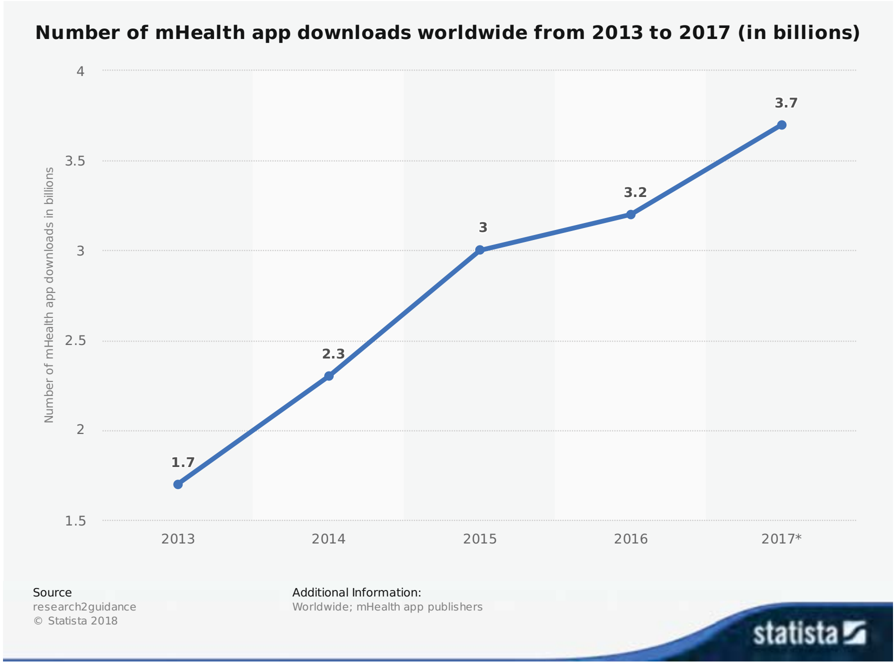


Figure 1.4: Number of mHealth app downloads worldwide from 2013 to 2017 [6]

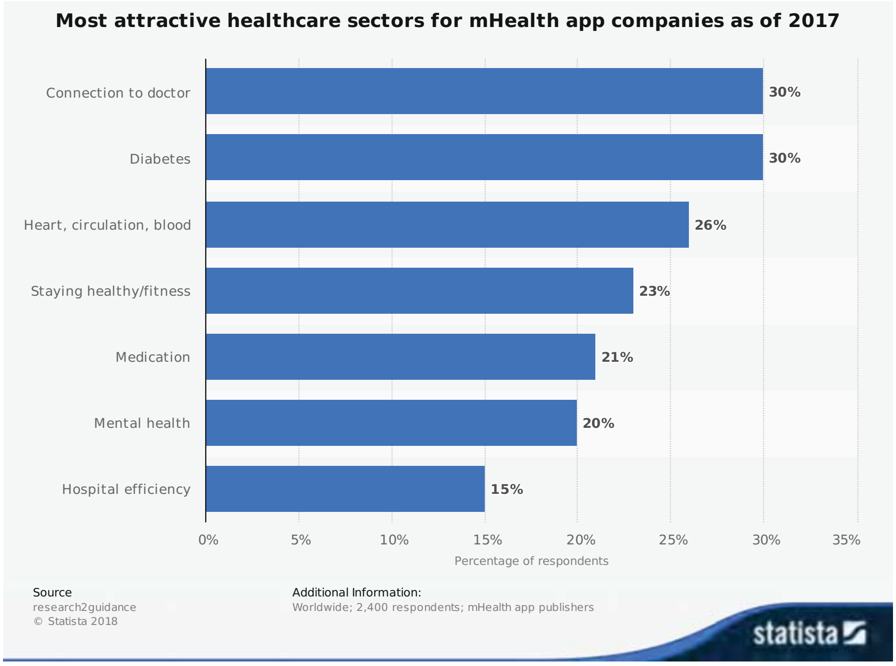


Figure 1.5: Most attractive healthcare sectors for mHealth app companies as of 2017 [7]

to representations of the therapeutic situation or the therapist, which includes thoughts, feelings, memories or dreams of the therapeutic situation or the therapist. [8] [9]

Since 2016, the Alpen-Adria-Universität Klagenfurt has been researching this topic and is currently endeavoring to advance the research process in this area. [9] Particular focus lies on the methodology for collecting and evaluating data of the patient's intersession processes, which can be improved tremendously by using modern digital aids. Therefore, a system is needed to scientifically record and understand intersession processes, which makes use of the advantages of the advancing digitalization and facilitates the tedious and inaccurate recordings. [10]

1.1 Problem statement

As a starting point for the topic of this thesis serves the general concept of the intersession processes and the need for a multi device eHealth platform, which can be used by researchers and therapists to collect, analyse, process and illustrate the patient data in a simple and fast manner. Furthermore, after a short test-phase, this technology will serve to conduct a three year field study on the topic of intersession processes. [9] [10] Details are elaborated in Chapter 2.

Therefore, within the scope of this thesis, in cooperation with the Institute of Psychology of the University of Klagenfurt, an mHealth application for the iOS platform is developed, which allows to scientifically record intersession processes of patients in psychotherapeutic treatments. The functionality of the iOS application needs to include the following crucial components:

- The completion of dynamically compiled and potentially appointment-dependent questionnaires
- Obtaining so-called interventions which can be triggered by a therapist manually or under certain conditions
- Managing appointments with the therapist
- Adequate user account and privacy management

- Synchronization with an back-end REST API [11] and offline usage
- iOS system notifications for certain events and tasks
- Suitable therapy information and user guidance

Thus, it should be significantly more easy and efficient for the therapist to collect and evaluate data on the patient's intersession processes and to prepare in advance for the upcoming therapy session.

1.2 Objective

The main goal of this thesis is the structured conception and realization of a mobile application, which later can be used for the purpose described in the beginning of Chapter 1 and in Section 1.1. The application, programmed with the modern programming language Swift 5[1], is intended to integrate seamlessly into the overall system, as described in Section 1.3 and enable patients with iOS devices to participate in the study described in Section 2.4. In addition to technical and subject-specific correctness of the application, the focus should also be on a particularly intuitive and simple usability, starting with registration, login and the profile management, to the answering of questionnaires and interventions, as well as setting up appointments. In order to make it as easy as possible for the participants of the study to access the iOS application, it should be published on the Apple App Store[2] before the study is started. In the long-term, assuming a successful study, the application could generally be available to patients undergoing psychotherapeutic treatment, if supported by the responsible therapist.

1.3 Classification in the overall system

The overall system is developed within the scope of an interdisciplinary project and supervised by Mag. Thorsten-Christian Gablonski[3] of the Department of Clinical Psy-

[1] https://developer.apple.com/swift/
[2] https://itunes.apple.com/us/app/intersession/id1457998965?l=de&ls=1&mt=8
[3] https://www.aau.at/team/gablonski-thorsten-christian/

chology, Psychotherapy and Psychoanalysis of the University of Klaagenfurt[4] and Dr. Rüdiger Pryss[5] of the Institute for Databases and Information Systems of the University of Ulm[6]. The project can be divided into four subprojects, represented by loosely coupled software components:

- iOS application developed with Swift 5
- Android application developed for Android 8.1 (API level 27)[7]
- REST web service realized with the Laravel 7.2 framework[8]
- Intersession-Online[9] web application for HCPs[10] written in Angular 5[11]

Both the iOS and Android application will be used by patients, whereas the web platform will mainly serve the purpose of administration and overview for researcher and therapists. These three components will communicate over a shared REST API of the web service, which also stores the patients data sets. This thesis describes the conception and realization of the iOS application, but at some points will elaborate cross-project decisions and concepts.

1.4 Structure of the thesis

This thesis is segmented into chapters representing the different steps in the development process of the resulting application – most building on their preceding chapters, some interdependent. For a better understanding of the work, Chapter 2 discusses the psychological foundations and Chapter 3 related topics of digital data collection, on which different parts of the work fall back. Requirements for the mobile application to be developed are discussed and defined in Chapter 4. Subsequently, Chapter 5 describes

[4] `http://wwwg.uni-klu.ac.at/psy/index.php?cat=inst&sub=abtx&abt=klin`, accessed 22.06.2019

[5] `https://www.uni-ulm.de/in/iui-dbis/mitarbeiter/mitarbeiter/ruediger-pryss/`

[6] `https://www.uni-ulm.de/in/iui-dbis/startseite/`

[7] `https://developer.android.com/about/versions/oreo/android-8.1`

[8] `https://laravel.com`

[9] `https://intersession.dbis.info/home`

[10] Healthcare Provider

[11] `https://angular.io`

the mobile applications design and prototyping process, focusing also on the software architecture and important aspects of the user interfaces. Chapter 6 highlights some of the more complex implementation issues. The completed application is presented in Chapter 7 and the result compared with the provided requirements in Chapter 8. Finally, in Chapter 9, the results of the work are summarized, some conclusions are made and an outlook on future projects is given.

2

Basics

This chapter discusses the scientific background, which is the basis for the desired iOS application as well as the overall system. The fundamentals are explained and terms introduced that contribute to the understanding of this work.

2.1 Theoretical background

As described in chapter 1.3, the mobile iOS application resulting from this work is part of an overall system, which is used to carry out the study described in [10].

2.1.1 Psychology, clinical psychology and psychotherapy

As an academic discipline of immense scope, psychology is the empirical science of behaviour and mind. Through the study of conscious and unconscious phenomena, as well as feelings and thoughts, human experiences and behaviors, their development in the course of life and all relevant internal and external causes or states should be described and explained. Being one of its disciplines, clinical psychology[1] [2] combines science, theory and clinical knowledge and aims to scientifically investigate biological, social, developmental and behavioural as well as cognitive and emotional basis of mental disorders. The goal is to understand, prevent, diagnose and treat these kind of disorders, such as depression and anxiety, to further the patient's subjective well-being and personal development. [12, 13]

[1] `https://www.apa.org/about/index`, (accessed 22.06.2019)

[2] `https://careersinpsychology.org/becoming-a-clinical-psychologist/`, (accessed 22.06.2019)

An even more specific branch of clinical psychology is the research field psychotherapy, which deals with different processes of psychotherapeutic procedures and their effectiveness. The therapy outcome and its context allows conclusions about the impact of various psychotherapeutic process variables. The concrete process variables must be examined in relation to the therapy result. Therefore modern process research now frequently occurs in combination with outcome research, called process-outcome research.[3]

2.1.2 Conventional research area of psychotherapy

Conventionally researchers dealt with the study of efficacy differences of various forms of therapy as well as with the direct and indirect effects of psychotherapy, mainly focusing on the therapy session per se. Since the actual therapy time represents only a rudimentary portion of the average wakefulness, it is reasonable to suppose that important processes relevant for the therapy outcome occur between the therapy sessions [14].

2.1.3 Intersession processes

In contrast, recent psychotherapy research has developed an increasingly strong focus on these, for a long time neglected, so-called intersession processes. This conceptualization from [15] describes the processing of the therapy and its associated content between the therapy sessions [8]. This includes all representations of spontaneous and intentional thoughts, memories, fantasies and feelings about the therapy and the therapist, especially in critical or emotionally difficult to handle situations. [16]. Various psychoanalytic and developmental psychological theories and concepts serve as theoretical basis for the intersession processes. [15] The aim is to solve the strict dichotomies between the paradigms, to combine different perspectives and to describe the relevant theoretical aspects as complementary explanatory approaches of the intersession processes. [8]

[3] `http://wwwg.uni-klu.ac.at/psy/index.php?cat=inst&sub=abtx&abt=klin`, (accessed 22.06.2019)

2.2 State of the art

Despite the inclusion of intersession processes in the Generic Model of Psychotherapy [17], there have been very few studies in this area, including 17 using quantitative and one using qualitative[4] methods [18]. Findings from previous research have been summarized in [10] and some relevant parts are covered by this section to get a rough impression of topics and issues of intersession research. These findings play an important role in the design of the mobile application regarding the temporal and visual representation of content and the content itself.

Early studies examining the nature and occurrence of intersession processes showed that 90% of patients thought about therapy and therapist between sessions [19]. These on average 30-60 second intersession experiences [20] occurred mostly immediately before and after a therapy session [21], but also in attenuated measure after completion of the psychotherapy [22]. Since a connection between intersession processes and therapy outcome has already been established, the question now arises why the processes over the intersession period have not yet been researched in greater extend. The gold standard for determining these is the Intersession Questionnaire [14], traditionally used immediately before the therapy session, which is often too short-term in practice. This can now be circumvented by the use of digital questionnaires on smartphones.

Recent studies, which researched relationships between intersession processes and other relevant constructs of psychotherapy, were often able to replicate the positive correlation to the therapeutic relationship [23, 16, 9, 24], as well as to the therapy outcome [25, 26]. This close connection also plays a crucial role in the design of the mobile application, especially for the interventions described in Section 4.2.6. [27]

Given that almost all previous studies on intersession processes have been of quantitative nature, a qualitative study of the relationship between the actual processes within a therapy session (in-session processes) and the intersession processes appears instructive [10]. In Chapter 4, the above findings are considered for the requirements engineering of a mobile application, to investigate the origin and how to maintain these.

[4] https://www.snapsurveys.com/blog/qualitative-vs-quantitative-research/

2.3 Problem statement of the overall study

So far, the nature and the occurrence as well as the connections to psychotherapy-relevant constructs or disturbance-specific differences were examined. However, not exactly what factors during therapy result in positive and frequent intersession processes, which in turn are associated with a positive therapeutic outcome. [10] Based on the current research state and [10], the following questions are relevant to this thesis:

- Can the frequency and positivity of intersession experiences be increased with the help of a smartphone application?
- Do patients with application guidance and higher intersession activity have a better outcome?

2.4 Study design

In this section, the study design of the dissertation project [10], which also includes the conception, realization and evaluation of the mobile iOS application resulting from this work, is summarized in a compact form. The purpose of this is to point out the basics for the development of the application and its purpose. The knowledge gained will be taken into account in subsequent chapters of this thesis.

2.4.1 Phase 1

In the first phase, the exact connections between the in-session processes of psychotherapy and subsequent intersession processes were exploratory investigated, examining particularly successful techniques and interventions to use for the mobile application. In-session processes were examined qualitatively, while intersession experiences were recorded and quantified using the Intersession Questionnaire [14]. At the beginning and end of therapy, patients filled out a whole stack of questionnaires, additionally prior to each individual psychotherapy session, they filled out the Intersession-Questionnaire, the short version of Symptom-Checklist (SCL-9) [28] (symptom development), and the

Working-Alliance-Inventory (WAI) [29] (therapeutic relationship) – quite similar to the resulting iOS application. Individual psychotherapy sessions were qualitatively evaluated in conjunction with intersession data. [10]

2.4.2 Phase 2

Aiming to capture intersession processes quickly and economically just prior to a therapy session, based on the results of the first phase, the mobile application for capturing and promoting them was developed, reviewed and revised by [10], hand in hand with this thesis. Therefore, corresponding requirements for the application have been defined, as discussed in detail in Chapter 4. Summative, for comprehensibility, practicability, functionality, therapeutic support and optics, rated and descriptively evaluated, the application was adapted according to the results (also see Chapter 8). [10]

2.4.3 Phase 3

In the third phase, the developed application is used in selected representative specialist clinics to compare the treatment outcome of patients with and without application use as part of a control group comparison with a sequential design plan[5]. Patients with interventions are expected to have

- a more stable intersession activity
- a better treatment outcome with regard to their symptom burden (before and after the treatment)
- a better therapeutic relationship (subjectively assessed success and relationship satisfaction)

compared to patients without interventions. Thus, these three criteria form the primary outcome. As a secondary outcome, the reflectivity, negative attitudes, structure level, traumatic experiences and attachment style are examined. This results in the dependent variables (DVs)

[5] Sequential Design: Patient cycles are alternately assigned to the experimental and control group until the required sample is obtained.

- intersession activity
- treatment success
- therapeutic relationship

and the independent variable (IV)

- setting (without vs. with application usage)

For all DVs the H1-Hypothesis[6] is assumed: Patients with application usage differ significantly from those without, in the primary outcome. [10]

2.5 Digitalization

Since this work represents an intersection between psychology and computer science, this section looks at the fundamentals of digitization in healthcare.

2.5.1 Electronic health

Encompassing health, information technology, and commerce, the term eHealth is derived from electronic health and describes a set of different concepts for delivering heath care by using information technology . In the course of digitization, the subject of eHealth has recently been playing an increasingly important role in the healthcare sector (see Chapter 1). Digitized patient data can be exchanged quickly and efficiently between HCPs, patients and other institutions for a variety of purposes, which allows completely new applications in this area. [30, 31, 32]

A special sub-category of eHealth is mobile health, or mHealth for short, which describes the use of a variety of mobile technologies that enable many applications in healthcare, public health, and health-related activities at an individual level. With eHealth applications, the need for physical attendance for caring or information exchange can in many cases be eliminated or immensely facilitated. However, especially in psychotherapy the immediate

[6] `http://www.real-statistics.com/hypothesis-testing/null-hypothesis/`

contact with the patient must not be missing. With the right applications, however, mHealth can also promote it and optimize the flow of a psychotherapy session. [32] For this purpose smartphones, tablets as well as wearables are utilized to run mobile applications, which can make a use of sensors, patient data, monitoring devices, social media, personalized health dashboards, and other applications connecting clinicians and patients in new efficient ways. [33, p. 243]

2.5.2 Opportunities of electronic questionnaires

In the health care sector in general and psychotherapy in particular, patient data is collected very frequently by means of questionnaires. For example, in the study described in Section 2.4.1, self-assessment questionnaires were used to collect patient data. Traditionally, data is collected in the form of questionnaires on paper and in the intersession research just prior to a therapy session, although these were often created in electronic form. As a consequence, this results in a time-consuming and costly manual re-digitization phase of the completed data, which also means that the results often can not be properly included in the therapy session directly afterwards [10]. A completely electronic data collection, on the other hand, enables an automated data analysis and thus saves valuable resources, especially in the context of large-scale studies as planned in Section 2.4.3 [10]. As a positive side effect the availability, reusability, comparability and hence the overall quality of the collected data sets highly benefits, without any loss to the professional accuracy. Digitalization also offers almost limitless possibilities for visualization and simplification of the datasets. In addition, a situation-specific dynamic adaptation of questionnaires is possible during operation, even abnormalities can be detected more easily and corresponding HCPs be automatically alerted. [34, 35, 36, 37] Patients frequently behave atypically in clinical environments, which is a huge impact factor on clinical research. This is where mHealth applications shine, collecting data in a less-constrained setting that reflects day-to-day behaviour more naturally. [38]

2.5.3 Mobile applications

Mobile applications are one trailblazer for electronic questionnaires and an integral part of our everyday lives (see Chapter 1). A mobile application is a software application designed to run on a mobile device such as a smartphone, tablet or smartwatch. The resulting application of this work is meant to be a native iOS application, which means, in contrast to a web- or hybrid-application, it is written in the iOS platform's corresponding native programming language Swift 5. Although this is associated with an increased development effort when dealing with several different target platforms and hence multiple codebases, it also entails a multitude of possibilities. For example, native programming enables access to the device's sensors and notificationcenter, interaction with other applications over the operating system (OS), and optimization capabilities in terms of processing power, speed, and energy consumption. Usually, the user is accustomed to a look and feel typical of the platform the application is running on, which usually can only be offered in its full extent by native applications. This topic is further discussed in Section 6.1. [39] [40]

2.5.4 Mobile data collection

The research field of mobile crowd sensing (MCS) is an important element of this work. Mobile crowd sensing or people/human-centric sensing in general is a relatively new paradigm taking advantage of large groups of omnipresent, independent, geographically dispersed, mobile devices, such as smartphones, tablets, wearables or even vehicles, to efficiently collect and evaluate data, therefore enabling countless applications with a common interest on a large scale. One of the most important aspects of MCS is human involvement, which also plays an essential role for this thesis in the form of mobile phone sensing (MPS) and data collection via questionnaires, both highly topical fields of research. Modern mobile devices offer a variety of sensors that can determine the orientation, movements, geographic location and environment of the device. In addition, front and rear camera, features such as image and face recognition and the microphone provide the ability to record voice. Interfaces like LTE, WLAN and Bluetooth enable versatile communication possibilities. [41, 42, 4, 43, 44]

Despite or even because it is a promising technology in the field of human data capture, there are several privacy concerns, which are discussed by [45, 46].

In [46] two classes of sensing paradigms for MCS are differed:

- Participatory sensing: Participants must consciously take action to decide when, where and how the application can capture which of their data. In the resulting mobile application of this work, the patient fills out questionnaires and has to actively click on a submission metaphor.
- Opportunistic sensing: The user is completely unconscious that the application may run in the background and can opportunistically collect data without the user being actively involved. Not the case in the context of this work.

MCS is used wherever large amounts of data need to be captured, making health care an ideal area of application. In Chapter 3 examples for MCS applications, frameworks and other related work to the context of this thesis will be presented. [47, 48, 49, 43, 38, 50]

3
Related Work

In this chapter various examples of related work from similar fields of research and topics are shown. The main focus is initially on work that emerged in the context of [10]. Furthermore, electronic data collection and mobile health applications will be discussed.

3.1 In the scope of this project

As already elaborated in Chapter 1 and 2 this work is part of an overall system designed for research purposes and based on the research field of [10]. In the context of this overall system, an API based on the REST paradigm has been developed in [51]. The resulting API is visualized in 3.1 with the interactive tool Swagger[1] and can be explored online[2]. Through this interface, all involved components of the entire system, including the web and android applications [52, 53], which can be seen in Figure 3.2 and 3.3, can easily communicate with each other via JSON[3] over HTTP/2[4] and persist data on a server intended for this purpose. [51] describes the software development process of this REST-API and its functions. In addition to account and privacy management, the most important functions of the API include the retrieval of assigned interventions and questionnaires as well as the receipt of corresponding completed submissions. When and which interventions or questionnaires are available is decided by an algorithm derived from rules defined by the author of [10], which calculates this based on the

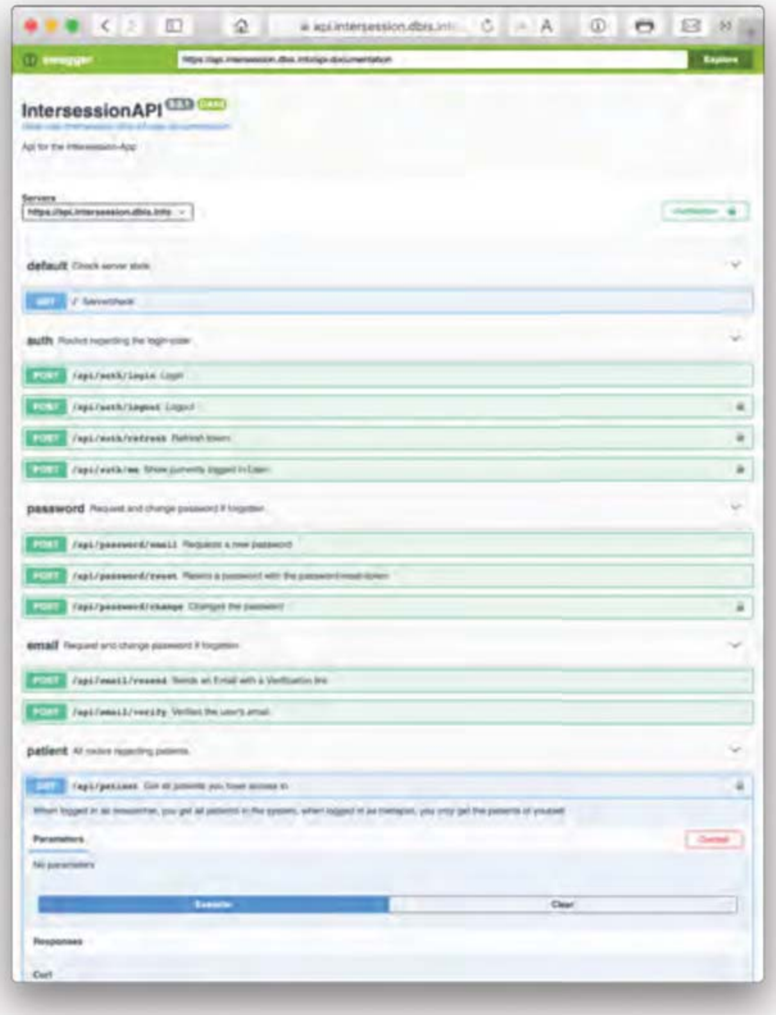
[1] https://swagger.io

[2] Intersession-Online REST-API with Swagger, as of 30.06.2019: https://api.intersession.dbis.info/swagger/

[3] JavaScript Object Notation: https://www.json.org

[4] Hypertext Transfer Protocol Version 2 defined in https://tools.ietf.org/html/rfc7540 and https://tools.ietf.org/html/rfc7541

completed questionnaires of patients and their self-specified therapy session appointments. Therapists and researchers have access to relevant patient data over the web application seen in Figure 3.2.



swagger https://api.intersession.dbis.info/api-documentation Explore

IntersessionAPI

Api for the Intersession-App

Servers
https://api.intersession.dbis.info

default Check server state
GET / ServerCheck

auth Routes regarding the login-state
POST /api/auth/login Login
POST /api/auth/logout Logout
POST /api/auth/refresh Refresh token
POST /api/auth/me Show currently logged in User

password Request and change password if forgotten
POST /api/password/email Requests a new password
POST /api/password/reset Resets a password with the password-reset-token
POST /api/password/change Changes the password

email Request and change password if forgotten
POST /api/email/resend Sends an E-mail with a Verification link
POST /api/email/verify Verifies the user's email

patient All routes regarding patients
GET /api/patient Get all patients you have access to

When logged in as researcher, you get all patients in the system, when logged in as therapist, you only get the patients of yourself

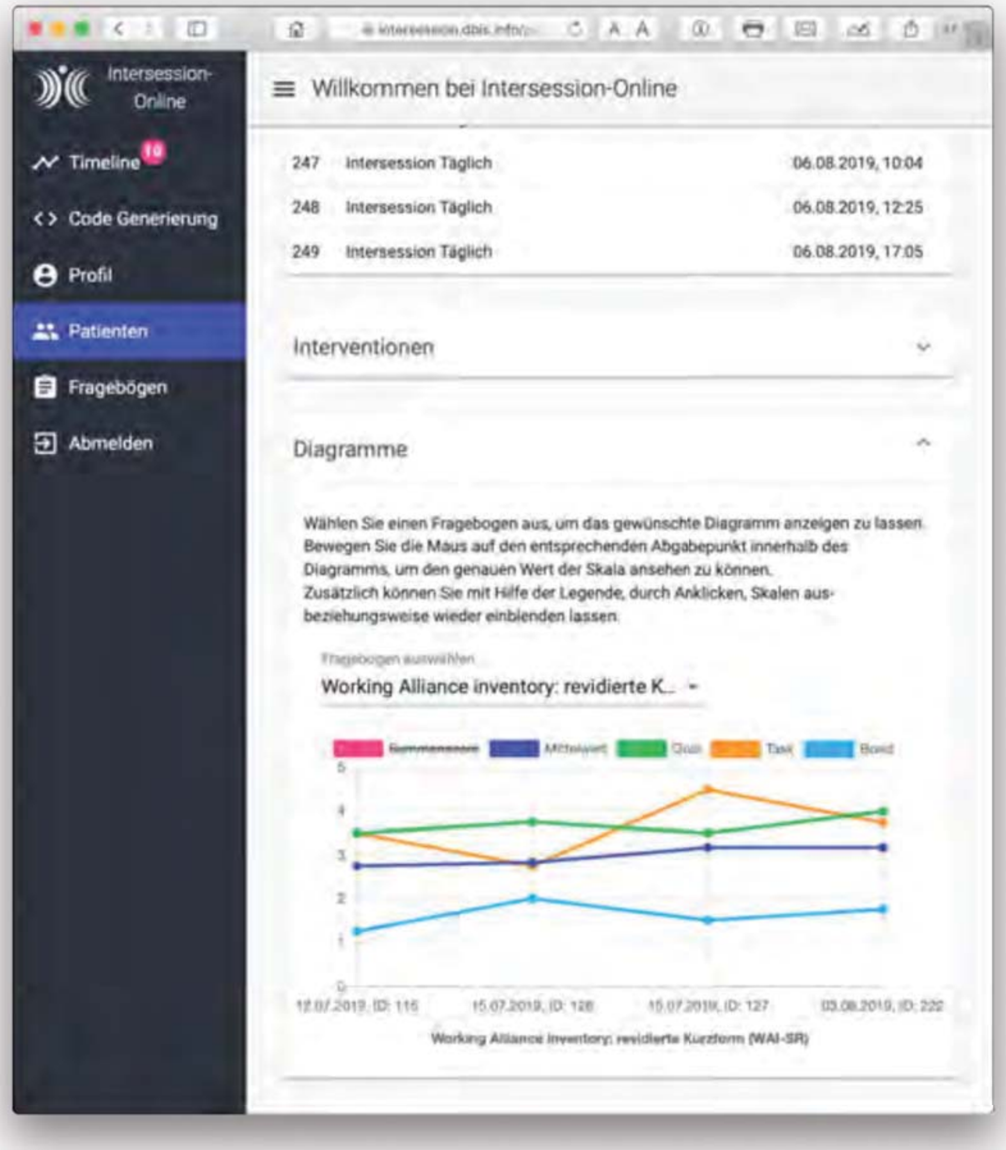
Parameters

No parameters

Responses

Curl

Figure 3.1: Intersession-Online REST-API with Swagger, by [51]; screenshot taken 06.08.2019



Willkommen bei Intersession-Online

247	Intersession Täglich	06.08.2019, 10:04
248	Intersession Täglich	06.08.2019, 12:25
249	Intersession Täglich	06.08.2019, 17:05

Wählen Sie einen Fragebogen aus, um das gewünschte Diagramm anzeigen zu lassen. Bewegen Sie die Maus auf den entsprechenden Abgabepunkt innerhalb des Diagramms, um den genauen Wert der Skala ansehen zu können.
Zusätzlich können Sie mit Hilfe der Legende, durch Anklicken, Skalen aus- beziehungsweise wieder einblenden lassen.

Fragebogen auswählen: Working Alliance inventory: revidierte K...

Figure 3.2: Intersession-Online Angular web application – patient profile, by [52]; screenshot taken 30.06.2019

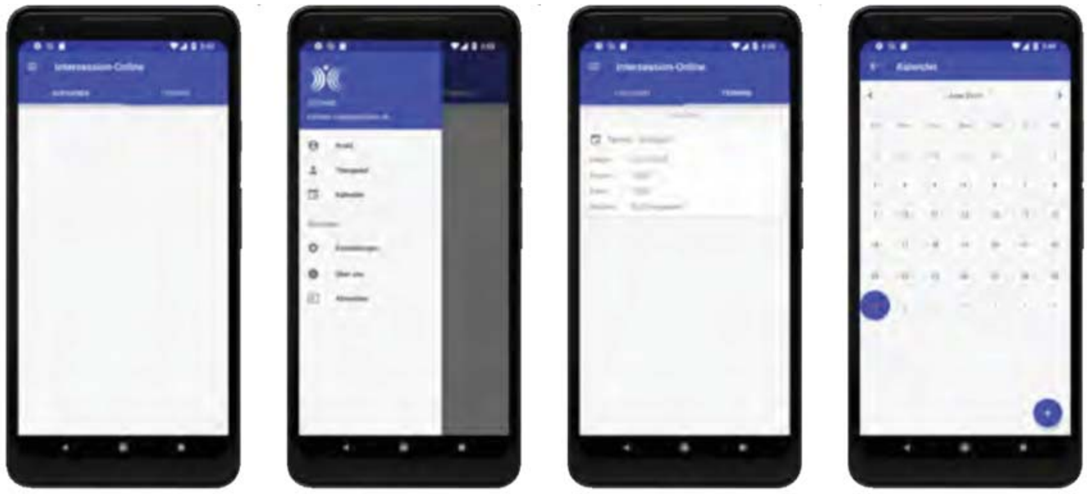


Figure 3.3: Intersession-Online Android mobile application, by [53]; screenshots taken 30.06.2019

3.2 Other related research topics

Before formulating the requirements resulting from Chapter 2 and Section 3.1, first various related work in the area of data collection, questionnaires, eHealth and mHealth is discussed. The insights gained will be incorporated into the development process of the iOS application in the Chapters 4, 5 and 6.

3.2.1 Digital data collection

The Institute of Databases and Information Systems (DBIS) at the University of Ulm has several research topics regarding digital data collection and produced numerous contributions to this scientific field. Due to the high cost of analogue surveys a questionnaire framework was researched and developed over a period of more than six years [48]. This generic framework enables the creation, execution and development of questionnaires, among other, for psychological studies with smartphone technologies and other mobile devices. [48, 54]

QuestionSys[5] is a generic framework for a variety of healthcare scenarios where large scale mobile data collection and sensor data integration is required. One goal of its model-driven approach is to empower healthcare experts to create these applications on their own with reasonable efforts and avoiding time consuming as well as costly development, implementation or maintenance processes for its end users. An important component of the platform-independent system is therefore an easy-to-use configurator that allows these experts to quickly create digital questionnaires that can then be filled out utilizing smartphones. [55, 56, 57, 58, 59]

Another DBIS paper [38] discusses the basic requirements for flexible and generic APIs for the purpose of mobile crowd sensing in the mHealth field of application. The resulting API is based on three essential criteria:

- REST as architectural style
- JSON for data exchange between components
- Laravel framework as basis for the implementation

This is almost identical to the API discussed in Section 3.1 and the resulting overall system, developed to track the individual daily situation, has quite a lot in common with the overall system this works iOS application is part of. The system is currently used for applications such as Track your Tinnitus[6] (TYT) or TrackYourStress[7] (TYS). [38]

Another research group tries to lower the burden of entry into MCS for human-subject researchers, who lack a technical orientation, with their system called Sensus. [49]

3.2.2 mHealth applications, mental health and eHealth in general

In the context of eHealth, the DBIS has several research cooperation projects. For example they research new ways to treat tinnitus patients by using eHealth tools like mobile applications and web-based platforms. Over the last few years, numerous applications and concepts have been developed in this field of research, which brought

[5] https://www.uni-ulm.de/in/iui-dbis/forschung/laufende-projekte/questionsys/
[6] https://www.trackyourtinnitus.org/de/home
[7] https://www.trackyourstress.org/home

new insights into technical issues as well as user behaviour. One of the basic ideas here is that patients monitor themselves in everyday life. In the form of different exercises, questions tasks and even games related to their symptoms and habits, the patients test themselves. They receive practical tips as needed, can interact with their respective HCP and follow their own progress. The HCP can view the collected data online and get an idea of the condition of the patient. [4, 43, 50, 60, 61, 62, 63, 64]

An iOS mHealth application with some capabilities similar to those resulting from this work is TinnitusTips. This applications utilizes the generic questionnaire API mentioned in 3.2.1 for tinnitus self assessment. Furthermore, the patient also has an HCP available, as well as a feedback function with helpful tips for dealing with tinnitus. [65]

It is important to note that the applications described are usually designed to support ongoing treatments and can not completely replace them. To avoid legal problems with the release of an eHealth/mHealth application it is crucial to make the application usable only under supervision of an suitable HCP. Therefore, use of the application should be possible for the patient only if authorized by the appropriate supervising HCP. In the scope of this thesis and the resulting iOS application the HCP[8] is a psychologist or the person responsible for the psychological treatment / care of the corresponding patient.

There is also great potential for mobile applications in the mental health area. Mental health mobile apps offer consumers the opportunity to empower themselves, reduce stigma associated with finding mental health services, use self-monitoring opportunities, improve communication between patients and HCP, and improve psychological services. Another promising use of smartphones is in the behavioural health: External hardware Devices such as biofeedback sensors can be connected to mobile phones to further enhance their capabilities. [32, 66, 67]

Although this is a promising field of research, the benefits of such applications must still be carefully weighed against ethnic concerns, such as privacy, informed consent, security of confidential data, and potential harm. Ultimately, it is up to the responsible HCP to clarify and inform. Patient preparedness and preferences also influence the involvement and impact of mHealth in psychiatry. There are a variety of scientific publications on

[8] Who is considered a Health Care Provider/Practitioner? `https://hr.berkeley.edu/node/3777`

ethnic, technical and developmental policies in the field of mobile and mental health applications. [32, 66, 67, 68]

3.3 Conclusions

The network interface design of the mobile application, resulting from this work, must obviously be geared to the circumstances given by [51], see Section 3.1. The design of the mobile iOS application should fit into the visual concept of the other user interfaces, originated in [52, 53], of the overall system, although operating system specific and creative differences may exist.

In addition, insights from 3.2 regarding user interface design, software architecture, code design and general aspects of application development, especially for the mHealth area, are also considered. Various empirical values in dealing with dynamically generated questionnaires serve as a reference to the developed application.

In the analysis of other applications, a particularly important point has emerged, which also represents one of the biggest challenges to the implementation: The user experience is greatly improved by short loading times. Therefore, the application should never block the user interface while waiting for server responses and, if necessary, work completely server-independent in an offline mode. The keyword here is background synchronization.

4

Requirements Engineering

The disciplined and systematic approach to identifying, documenting, analyzing, auditing, reconciling and managing requirements under customer-oriented, technical and economic objectives is called requirements engineering (RE) [69, ch. 2.4].

This chapter defines the requirements for the iOS application developed in the scope of this thesis. They emerged from several cooperative meetings with the customer, represented by Thorsten-Christian Gablonski, and were refined iteratively until reaching the required degree of detail. First of all, some essential definitions are set up before discussing the functional and non-functional requirements in terms of their internal and external manifestations. Finally, the most important boundary conditions are taken into consideration and defined.

4.1 Definitions

For a better understanding of this chapter, important terms are defined below.

4.1.1 Vocabulary

Overall System The overall system refers to the totality of all software components developed for the scientific work of [10] and their functional interaction. This includes the API [51], the web application [52], the Android application [53] as well as the iOS application of this master's thesis.

Application In the further course of this work, the term application always refers to the iOS application unless this is defined more precisely.

Logged-in The patient has successfully entered the credentials, so the application can access personal data.

Online The patient is logged-in and the application is open.

4.1.2 Users

For the application's design, it is crucial to know which users have to be handled. There are three groups of users in the overall system: Researchers, therapists and patients. Theoretically there is a fourth group, the admins. Since admins only maintain the system, but do not use its functions, they can be disregarded. While scientists and therapists are essentially taking an observational role through the web application, patients are the actual users of the application. Patients are people undergoing psychotherapeutic treatment and can come from all age and occupational groups as well as social strata. Since the study can only be conducted with smartphone owners, the basic handling of iOS applications, but not a deeper technical understanding can be assumed. It is equally important to consider their mental condition and to make the look and feel accordingly as pleasant as possible – even for people with visual and motoric limitations. The information presented should therefore be limited to the essentials and visualized large enough. For the remainder of the work, the term user is used as a synonym for patient, unless otherwise specified.

4.2 Functional requirements

Functional requirements describe, function-oriented and in the language of the product, what the product does. For example, in form of functions, operational sequence descriptions, use cases, and scenarios determining how a system should respond to specific inputs. Furthermore, they can be tracked during development, validated and, if necessary, verified by means of automated tests. [69, ch. 2.3] The following defines the technical requirements that the system must meet.

4.2.1 Manage patient account

Account creation The user can create a new account. For this purpose, the privacy policy must be accepted, an unregistered, valid email address specified and a password with repeated input set. It is possible to not share patient data with researchers. After successful creation, the email address can be validated.

Email validation The patient must confirm the provided email address via a validation email. This as well as the validation status has to be communicated visually. Furthermore, there must be the possibility to enter a new email address, if there is a typo. After successful validation, the login screen should appear.

Login The patient can log in with a validated account by providing the credentials (email address and password), which the application can remember on desire.

Logout A logged-in patient can log out manually.

Password reset The patient is offered the opportunity to request the resetting of the account password, stating the corresponding email address.

Profile view The logged-in patient can view his profile data at any time and, if an internet connection exists, change the password. Profile data contains at least the email address and the password and, if the patient is paired (see Section 4.2.2), additionally date of birth, gender, beginning and end of therapy as well as the user ID.

4.2.2 Pairing

In order to use the application's actual functionalities with a validated account, the patient must connect to a therapist. For this purpose, the patient receives from his therapist a numerical code in writing, also called pairing-code, which must be stated together with his birthday and gender. The view for this process will appear after the first successful login, but may also be aborted and later reopened from the therapist profile.

4.2.3 Overview

An overview indicates pending tasks and upcoming appointments and provides access to the therapist profile.

4.2.4 Therapist profile

If the patient is “paired” (see Section 4.2.2), the therapists profile information can be displayed, containing the full name with academic degree, email, gender, medical office address and the corresponding telephone as well as an emergency number. Otherwise reference is made to “pairing”. The profile can be conveniently added to the Contacts.

4.2.5 Handling questionnaires

Questionnaire overview Both pending and completed questionnaires should be listed. The patient should be made aware of the name, deadline and progress per entry.

Questionnaires completion The patient should be able to fill in various questionnaires with the application. The progress of filling in should be displayed and automatically saved locally. When dealing with long questionnaires, a search function for omitted questions seems useful. If all questions have been answered, the questionnaire can be submitted, which prevents further editing. If connected to the internet, completed questionnaires have to be sent the server.

Questionnaire structure The application must be able to display generic defined questionnaires, with the following properties:

- Questionnaire A questionnaire can have a name and a description text. This can be followed by a ordered list, containing any number of sections.
- Section Each section can also have its own description text, followed by a horizontally arranged pane with any number of answer options, which are ordered and arranged side by side.

- Answer option Each answer option can contain an option text of any length. The pane with the answer options is followed by a list, containing any number of questions.
- Question Each question can have a question text. Regarding the number of answer options of the current section, there must be a corresponding number of choices under each question text.
- Choice According to the “Single-choice principle”, the selection of one choice excludes the others, whereby choices may also be undone.
- Conditionally displayed sections Depending on the previously selected choices for questions in other sections, it must be possible to automatically hide and display content-dependent sections accordingly.

With the properties listed above it is possible to display all required questionnaire combinations. The questionnaires in Appendix B.1 to B.4 are representative for those.

Questionnaire generation With the exception of the Intersession-Daily Questionnaire (Daily), all questionnaires are generated and provided by the server, which also sets the delivery period. For server-technical reasons the Dailies are generated and stored only locally, until completed by the patient. They are available for 30 minutes each and occur three times a day in distinct time periods (08:00–11:59, 12:00–15:59 and 16:00–19:59). If a Daily would start, before the previous ended, its starting time is moved to the future about the difference, to prevent intersections.

4.2.6 Handling interventions

Intervention overview Both new and completed interventions should be listed and identified accordingly.

Intervention reception and completion Determined by the collected data of the questionnaires, interventions are sent to the patient. Interventions received can, depending on their kind, be executed and completed, which prevents further processing. In order to prevent the patient from giving socially unacceptable textual answers, they are not synchronized with the server, which also means that the therapist can not access them. The only thing sent to the server is if the patient has opened and completed the intervention. Entered data may only be stored locally and must not be available to others – the patient must be kept aware of this. This is an important data protection issue and critical to the course of therapy.

Intervention structure Interventions can occur in three different forms:

- Commendation Just a commendation text related to the course of therapy.
- Task A description text defining a task for the patient. In addition, after completing, this should be noted down via a switch.
- Question A question text with a multi-line text input field for the answer.

All three types of interventions should be confirmed after editing.

4.2.7 Setting appointments

It has been explicitly requested and is of great importance for the study that patients register and manage appointments themselves to stimulate the intersession processes.

Appointment entry The Patient can create, show, update and delete appointment entries for the next meeting with the therapist. Each of these entries requires a title, as well as a start and end date and may contain a comment. The title is at least one character long and entered via a single line text field. The dates are entered over a date picker with a five minutes time interval, which appears directly below the date to be edited. The internal logic must prevent and point out invalid date combinations. The duration is shown, which should be at least five minutes as well. Finally, a multiline text area provides enough space for the optional comments.

Appointment list The patient can scroll through a list of all the appointments and filter out the past ones. If the list is displayed as a whole, upcoming dates should be apparent.

Calendar view A calendar view, in which appointments can be displayed per day, serves as an overview. The patient can jump to the current date and next upcoming appointment.

4.2.8 Notifications

The application notifies the user about new questionnaires and interventions at the time of availability via system notification. In addition, upcoming appointments must be announced one hour before the start. Notifications for individual functionalities can be deactivated separately.

4.2.9 Information, settings and other options

Notification settings Via a settings menu, the patient should be able to activate / deactivate notifications for questionnaires, interventions and appointments individually.

Profile settings For reasons of data protection, the patient must be able to deactivate the local storage of interventions after logging out. The same applies to the login credentials of the user account. It must also be possible to completely and permanently delete the user account with all associated data.

Data privacy statement The patient must always have access to the data privacy statement.

Data privacy The patient may at any time revoke or permit the use of personal data.

General information The application provides the opportunity to learn about intersession processes, the project, its sponsor OeNB[1]and the team involved.

Imprint and contact Furthermore, an imprint and contact data must be available.

4.2.10 Data synchronization and offline operation

Data synchronization The application should synchronize with the server at regular intervals and at appropriate events. This data synchronization process should always run in the background and should not interrupt user interaction, if possible.

[1] Oesterreichische Nationalbank: `https://www.oenb.at`

Offline operation The full range of functionalities should be largely available even without an existing Internet connection. This includes completing and submitting questionnaires, viewing and finishing interventions, as well as all appointment management functionalities. Accordingly, all affected data sets must be reconciled with the server at a later time, taking into account any inconsistencies.

Multiple device usage Properly implemented, this should enable the patient to parallelly use of the application on multiple devices. The Intersession-Daily Questionnaire is a special case because it is only stored locally until it is submitted, and therefore, until this time, it is not synchronized between two applications.

4.2.11 Error messages, warnings and hints

The patient should be informed by appropriate error messages, warnings and instructions whenever this makes sense. These should be kept as short as possible and their occurrence limited to a minimum. In addition, only helpful information should be provided.

4.3 Non-functional requirements

Non-functional requirements, also called quality requirements, describe qualitative characteristics of the system or its components and supplement the functional requirements defined in Section 4.2. These can only be described from the perspective of the system. [69, ch. 2.3] The non-functional requirements for the system are defined below.

4.3.1 Look and feel

Intuitive controls The application must be intuitive to use and therefore should not contain exotic controls. This should prevent an unpleasant training phase. In particular, the registration and login process should not be an inhibition threshold to the use of the application.

Appealing design The graphical user interface (GUI) design should be visually appealing and limited to the essentials.

4.3.2 Platform

The application should support iOS 12 and be optimized for appropriate devices. Significant differences in screen sizes, alignment modes, and presentation layouts, as well as the notch of some models, have to be considered.

4.3.3 Performance

GUI elements should react quickly and user interaction should not be interrupted. Expensive operations and general synchronization must therefore be outsourced to parallel background threads.

4.3.4 Economical resource consumption

Device resources should be treated sparing to save energy. Therefore, a balanced synchronization rate with the server is recommended. Unnecessarily complicated visual effects are to be avoided, too.

4.3.5 Reliability, robustness and functional safety

The visually represented and user-intended should correspond to the actually performed operations. Once performed, operations must be permanent. The application must be crash save against direct or indirect impacts, such as unreliable user input or a faulty internet connection. Therefore, all textual inputs must use pre-formatted text input fields, while incorrect inputs must be treated accordingly and made known to the user. In short, the application should be bug-free.

4.3.6 Information security

Confidential data must be transmitted encrypted and only authorized persons may view it. Login credentials may only be stored in encrypted form.

4.3.7 Modularity, exchangeability, extensibility and maintainability

Through a modular structuring and clean interface definition of the application, it should be possible to easily exchange, expand and maintain individual components. A strict separation of GUI, logic, database and network interface is therefore important. Logically related functions must be bundled accordingly.

4.3.8 User guidance

Despite the possibility of deleting the user account, this should be preferable prevented by appropriate measures. The rejection or revocation of participation in the study is possible, but should be, as far as possible, prevented by appropriate dialogues. The patient should always be aware of what the disabling of certain settings or the general denial of notifications entails.

4.4 Boundary conditions

Boundary conditions are requirements limiting the way in which the system under consideration can be realized, and supplement both functional and non-functional requirements [69, ch. 2.3]. The most relevant boundary conditions are described in this section.

4.4.1 Medical applications

Legal requirements for medical applications are particularly stringent for stand-alone applications. For this reason, it has been decided that all relevant functionalities may only be used in combination with a participating therapist to avoid legal restrictions.

4.4.2 Timeframe and procedure

The application development timeframe provides for an official release in May 2019. Therefore fast prototyping is recommended. Some paper sketches of crucial components should give a rough impression and serve to communicate the basic functionalities and appearance to the customer. With an early high fidelity prototype the implementation should be done in time.

4.4.3 Organization

In regular meetings via Skype, in person or email, progress is discussed and questions are clarified. New insights and results from the meetings are recorded in a structured way.

4.4.4 Documentation

Adequate code documentation is mandatory. In addition, a short manual for the application could be created.

4.5 Overview and prioritization of requirements

Agile planning is required. By prioritizing the requirements, possible delays are mitigated by omitting low-priority requirements. The prioritization must never be done solely from a technical point of view. Therefore, visual components that can be presented to the customer early in the development phase have a high priority. [69, ch. 7.4]

Table 4.1 summarizes the requirements of sections 4.2, 4.3 and 4.4 and classifies them according to the MoSCoW principle [70, ch. 4.3] into the following categories:

- Must have The system will not work without this minimum usable subset of fundamental requirements.

- Should have Important requirements without which the system is still functional and usable.
- Could have Functions that can easily be omitted in the current development increment.
- Want to have, but not this time (won't have) Desirable functions that can be included in future development steps.

Ref.	Requirement group	Requirement	Priority
4.2.1	Manage patient account	Account creation	Must
		Email validation	Must
		Login	Must
		Logout	Must
		Password reset	Should
		Profile view	Must
4.2.2	Pairing		Must
4.2.3	Overview		Could
4.2.4	Therapist profile		Must
4.2.5	Handling questionnaires	Questionnaire overview	Should
		Questionnaires completion	Must
		Questionnaire structure	Must
		Questionnaire generation	Must
4.2.6	Handling interventions	Intervention overview	Should
		Intervention reception and completion	Must
		Intervention structure	Must
4.2.7	Setting appointments	Appointment entry	Must
		Appointment list	Must
		Calendar view	Should
4.2.8	Notifications		Must

Ref.	Requirement group	Requirement	Priority
4.2.9	Information, settings and other options	Notification settings	Should
		Profile settings	Must
		Data privacy statement	Should
		Data privacy	Must
		General information	Must
		Imprint and contact	Must
4.2.10	Data synchronization and offline operation	Data synchronization	Must
		Offline operation	Must
		Multiple device usage	Could
4.2.11	Error messages, warnings and hints		Must
4.3.1	Look and feel	Intuitive controls	Should
		Appealing design	Should
4.3.2	Platform		Must
4.3.3	Performance		Should
4.3.4	Economical resource consumption		Could
4.3.5	Reliability, robustness and functional safety		Must
4.3.6	Information security		Must
4.3.7	Modularity, exchangeability, extensibility and maintainability		Must
4.3.8	User guidance		Could
4.4.1	Medical applications		Must
4.4.2	Timeframe and procedure		Should
4.4.3	Organization		Should
4.4.4	Documentation		Should

Table 4.1: Requirements prioritization according to the MoSCoW principle [70, ch. 4.3]

5 Design

This chapter discusses the design of the mobile application resulting from the requirements engineering of Chapter 4. First, organizational aspects of the design phase are taken into consideration, then important design decisions are explained before the results of the prototyping phases are presented. Finally, the architecture of the application is introduced.

5.1 Organizational aspects

Due to the timeframe of the project, rapid prototyping was obligatory. Therefore, paper sketches were first made (see Section 5.3.2), based on them, mock-ups were designed (see Section 5.3.3)) and finally a high fidelity prototype was implemented (see Section 5.3.4). Essentially over the first few weeks of the project, meetings were held every two weeks, incorporating new aspects into the design.

5.2 Design decisions

This section discusses some of the most relevant design decisions that have been made during the project and that specifically affect the mobile application.

5.2.1 Medical applications

As as explained in Section 4.4.1, therapy relevant functions must only be usable in combination with a participating therapist. This must be taken into consideration for the design of the GUI and especially the planning of the applications architecture.

5.2.2 The Intersession-Daily Questionnaire

Due to the backend API not providing new Dailies, the mobile application has to generate them itself, locally. The algorithm is based on the rules defined in Section 4.2.5 (questionnaire generation) and generates Dailies for n days in advance:

Step 1: Check the date of the last Daily. If its day x, where the time is 00:00, is n days in advance to the current date's day c, where the time is also 00:00, the algorithm finishes. If the last Daily is in the past or there is none, use the current date's day c instead.

Step 2: Generate a random starting time s_1 in the first time period of day $x + 1day$ and add 30 minutes to get the end time $e_1 = s_1 + 30min$. This results in the starting date $sd_1 = x + 1day + s_1$ and end date $ed_1 = x + 1day + s_1 + 30min = sd_1 + 30min$.

Step 3: Generate a random starting time s_2 in the second time period of day $x + 1day$. If $s_2 < e_1$ then set $s_2 = e_1$. Then set the second end time e_2 to $e_2 = s_2 + 30min$.

Step 4: Use the principle of step 2 for the third time frame, with e_2 as lower boundary for the third start time s_3.

Step 5: Go to step 1.

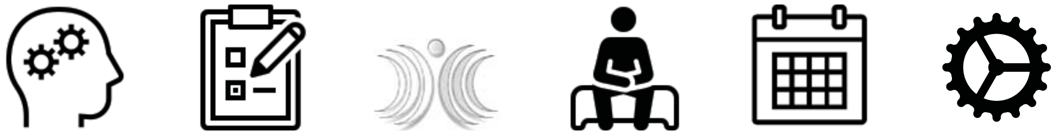
This is a crucial part of the application. Considering that there must be also kept track of the database and time zones, this relatively simple algorithm has a high potential for misbehaviour and hence was explained here in greater detail.

5.2.3 Notifications

In order to realize the notification functionality required in section 4.2.8 without real push notifications, an algorithm with predefined rules has been developed that links questionnaires, interventions and appointments from the database to local notification via IDs. It should be noted that Dailies have random start times and therefore can not be repeated, also only 64 notifications can be listed at the same time. The scheduling algorithm is called whenever the content of relevant database tables changes.

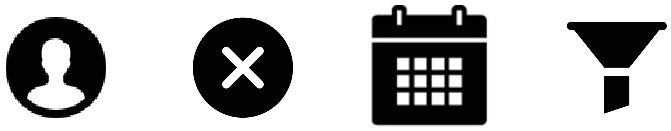
5.2.4 Icons

In order to obtain an appealing and memorable look and feel, icons were selected which represent suitable metaphors for the associated function. The chosen icons were downloaded from Icons8[1] and supplement the standard iOS icons provided by Apple. The application icon was designed by [52]. Most relevant icons are displayed in Figures 5.1, 5.2, 5.3 and 5.4. Affected by this are the tab bar (5.1a, 5.1b, 5.1c, 5.1e, 5.1f), navigation bar (5.2a, 5.2b, 5.2c, 5.2d) and table view (5.1d, 5.3a, 5.3b, 5.3c, 5.3d, 5.3e, 5.3f) icons as well as the login, registration and home screen (5.4a, 5.4b, 5.4c). If a GUI element is selected or activated, the icon changes its colour correspondingly. Most of the icons are used in combination with text to prevent confusion.



(a) Interventions (b) Questionnaires (c) Intersession (d) Therapist (e) Appointments (f) Settings

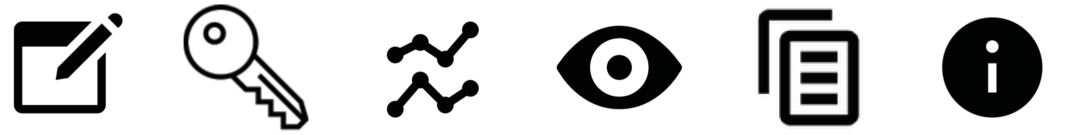
Figure 5.1: Icons – tab bar and main functionalities



(a) Profile (b) Cancel (c) Calendar (d) Filter

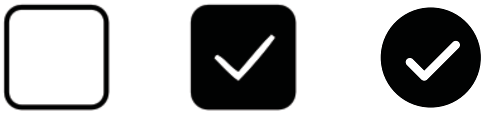
Figure 5.2: Icons – navigation

[1] Icons8: https://icons8.com



(a) Tasks (b) Credentials (c) Therapy data (d) Data protection (e) Data privacy statement (f) About us

Figure 5.3: Icons – table view

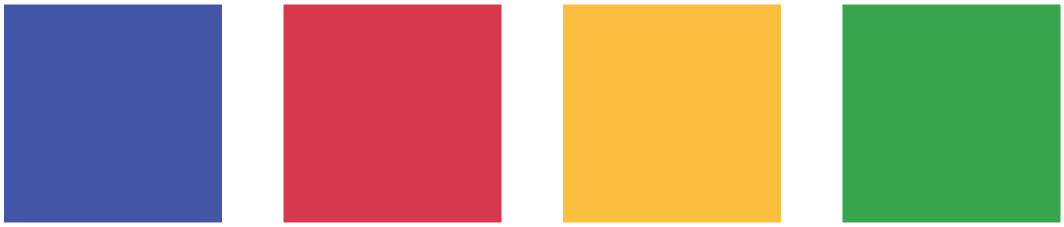


(a) Checkbox unchecked (b) Checkbox checked (c) Okay

Figure 5.4: Icons – other

5.2.5 Colours

The main application colours are presented in Figure 5.5 and replace the standard Apple colours of the GUI elements. Figure 5.5a shows the main colour of the application from which most shades are derived. The other colours in Figures 5.5b, 5.5c and 5.5d are inspired by the Bootstrap[2] colour theming for danger, warning and success.



(a) Theme base (b) Error, danger (c) Incomplete, warning (d) Ok, valid, success

Figure 5.5: Colours

[2] Bootstrap 4: `https://getbootstrap.com/docs/4.0/utilities/colors/`

5.2.6 Screen sizes

The design of the mobile application must consider the different screen sizes of supported iPhones and iPads, as well as changes to the display orientation and associated layout adjustments. The basis for this is the adaptivity and layout overview[3] of the Apple iOS Human Interface Guidelines [71] and an online article[4] summarizing relevant information about screen resolutions in pixels and points.

5.3 Prototyping

The mobile application was designed in constant communication with the customer and with the non-functional requirements (see Section 4.3) as well as Apple's Human Interface Guidelines for iOS devices in mind, to ensure a familiar, pleasant and easy usability.

5.3.1 iOS Human Interface Guidelines

The iOS Human Interface Guidelines are a collection of iOS design themes and principles that are highly recommended by Apple and should be considered when developing iOS applications. These principles should help to provide high quality applications. [71]

According to Apple [71] the three primary themes differentiating iOS from other platforms are defined as follows:

- Clarity "Throughout the system, text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design. Negative space, color, fonts, graphics, and interface elements subtly highlight important content and convey interactivity." [71]

[3] `https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/adaptivity-and-layout/`, (accessed 31.07.2019)

[4] `https://kapeli.com/cheat_sheets/iOS_Design.docset/Contents/Resources/Documents/index`, (accessed 31.07.2019)

- Deference “Fluid motion and a crisp, beautiful interface help people understand and interact with content while never competing with it. Content typically fills the entire screen, while translucency and blurring often hint at more. Minimal use of bezels, gradients, and drop shadows keep the interface light and airy, while ensuring that content is paramount.” [71]
- Depth “Distinct visual layers and realistic motion convey hierarchy, impart vitality, and facilitate understanding. Touch and discoverability heighten delight and enable access to functionality and additional content without losing context. Transitions provide a sense of depth as you navigate through content.” [71]

In addition, the overview lists six design principles that should improve the impact and reach of iOS applications [71]:

- Aesthetic integrity Appearance and behaviour depend on the function.
- Consistency The application should behave and look like the user expects it and is used to. A consistent terminology as well as the use of system elements, etc. is therefore recommended.
- Direct manipulation Screen contents should be able to be manipulated directly and the impact of these actions should be immediately visible.
- Feedback Briefly highlighting interactive elements as they are touched, progress indicators for long-running processes, and animations, confirm user interaction and inform the user.
- Metaphors Familiar metaphors in the physical interaction with the screen, like dragging, swiping, toggles, slider or pickers, accelerate the learning process.
- User control The application can suggest actions or inform about consequences, but the user should always be in control.

5.3.2 Simple paper sketches

The big advantage in paper sketches is that they can be quickly made and the loss is not dramatic if discarded for whatever reasons. This prevents the inhibition of the customer for constructive criticism and thus can avert significant late effects in the development process.

During the first two meetings, paper sketches were crafted in collaboration with the customer to catch the basic functionalities of the application. An example for such a paper sketch is shown in Appendix A.1, where the basic process of the user registration is defined.

5.3.3 Mock-ups

A mock-up is "a model of something, which shows how it will look or operate when it is built, or which is used when the real thing is not yet available"[5]. Therefore the goal of this chapter is to make the functionality and appearance of the application understandable to the customer. For this purpose the mock-up tool NinjaMock[6] was utilized. The resulting mock-ups are presented in Figure 5.6, 5.7 and 5.8.

Launch & welcome When launching the application for the first time, the loading screen (5.6a) transitions into the welcome screen (5.6b), which will present two options – the login and the registration (5.6c) – when tapped.

Login In the login view (5.6d) an already registered patient can log in with the user credentials (5.6e).

Registration In the registration process, the user is first presented information (5.6f) about the application in general and its purpose, followed by the data privacy statement (5.6g) and finally a view for entering new account credentials (5.6h). Before proceeding to create a new account the data privacy statement must be accepted (5.6i). Here the user has the option to not participate in the study. If the necessary data is complete, the register button now is clickable (5.6j). Before the registration is complete, the user must confirm the specified email address via a link sent by email. The application shows the status of this process and provides adequate options as well as useful hints (5.6k). As

[5] https://dictionary.cambridge.org/de/worterbuch/englisch/mock-up, definition for mock-up in business English (accessed 11.08.2019))

[6] NinjaMock: https://ninjamock.com/

soon as the email address is confirmed, the application indicates that (5.6l) and offers a direct link to the login screen with completed credentials (5.6m).

Pairing In order to use the therapeutic functionalities of the application, the user must undergo the pairing process. In the first step, the account is connected to a therapist account via a code (5.6n, 5.6o). In a second step, birthdate and gender must be stated (5.7a, 5.7b and 5.7c). When all data is complete, the process can be finished (5.7d).

Home When logged in after launching, the first view is the home tab (5.7e) with an overview of the notifications and a link to the therapists view. This view was added relatively late in the development process after some changes were made to the tab structure. All relevant functionalities can be reached via the main menu bar.

Profile In the profile (5.7f) user data can be viewed and the password changed.

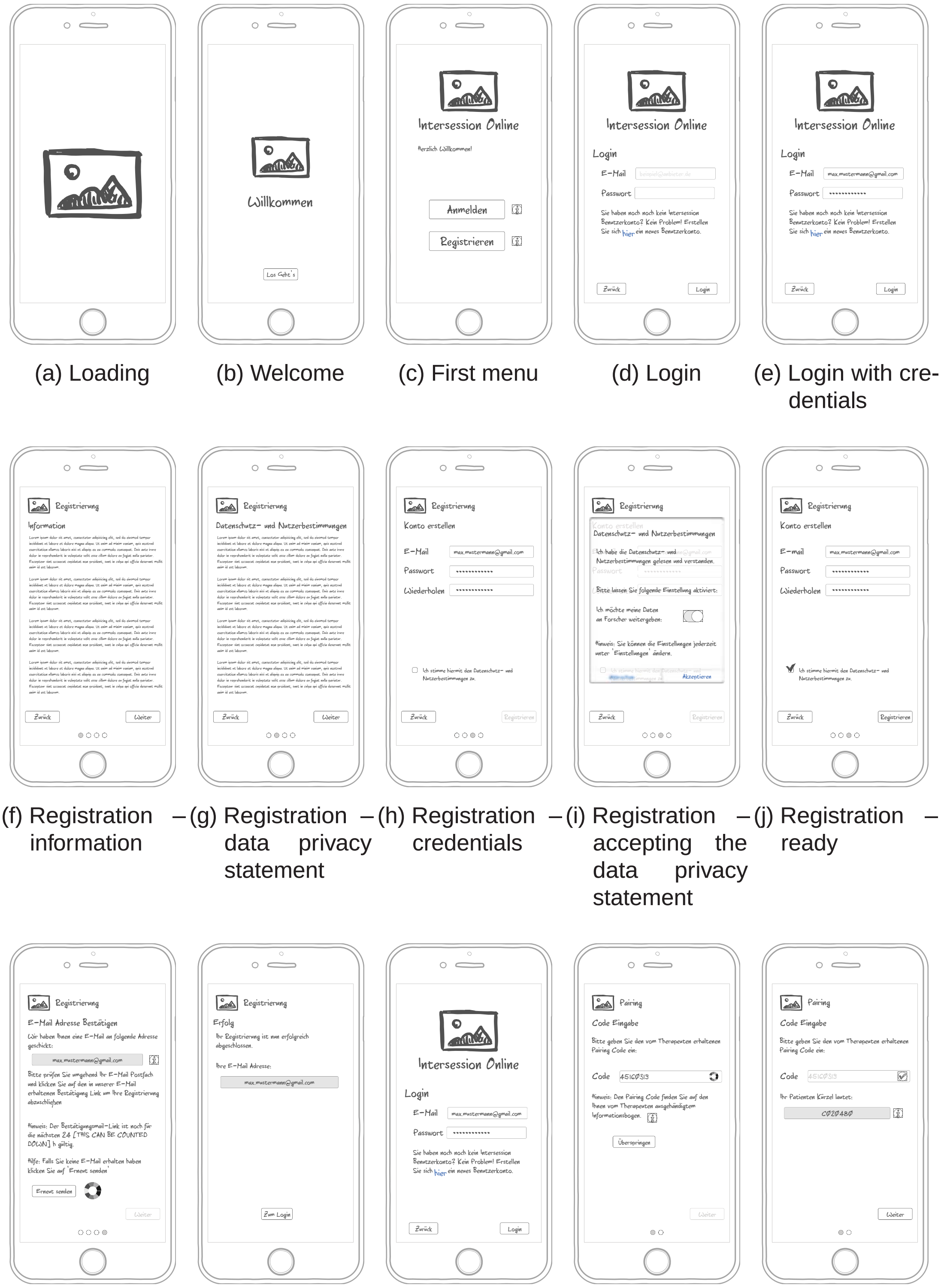
Tasks The task tab contains two segments, one for the questionnaires (5.7g) and one for the interventions (5.7h). Later in the development process, these segments received their own tabs. In addition to some questionnaire information, a progress bar indicates how many questions have been answered.

Therapist In the therapist view, either the indication that the account is not linked to a therapist's account (5.7i) or the contact information of the associated therapist is displayed (5.7j).

Appointments In the appointment tab, the user can view upcoming and past appointments in either a calendar (5.7k) or list view (5.7l). Figure (5.7m) shows the view which serves to create, display, update and delete appointments.

Settings The settings tab (5.7n) provides access to various information views and several options to configure the profile, notifications (5.7o) and privacy.

Questionnaire Figure 5.8a shows the general structure of a questionnaire. A progress bar and a counter indicate how many questions are answered/open. When all questions are answered, the done button is clickable as shown in Figure 5.8b. Figures 5.8c, 5.8d and 5.8e show how the Intersession-Daily Questionnaire, which contains conditional questions, is displayed.

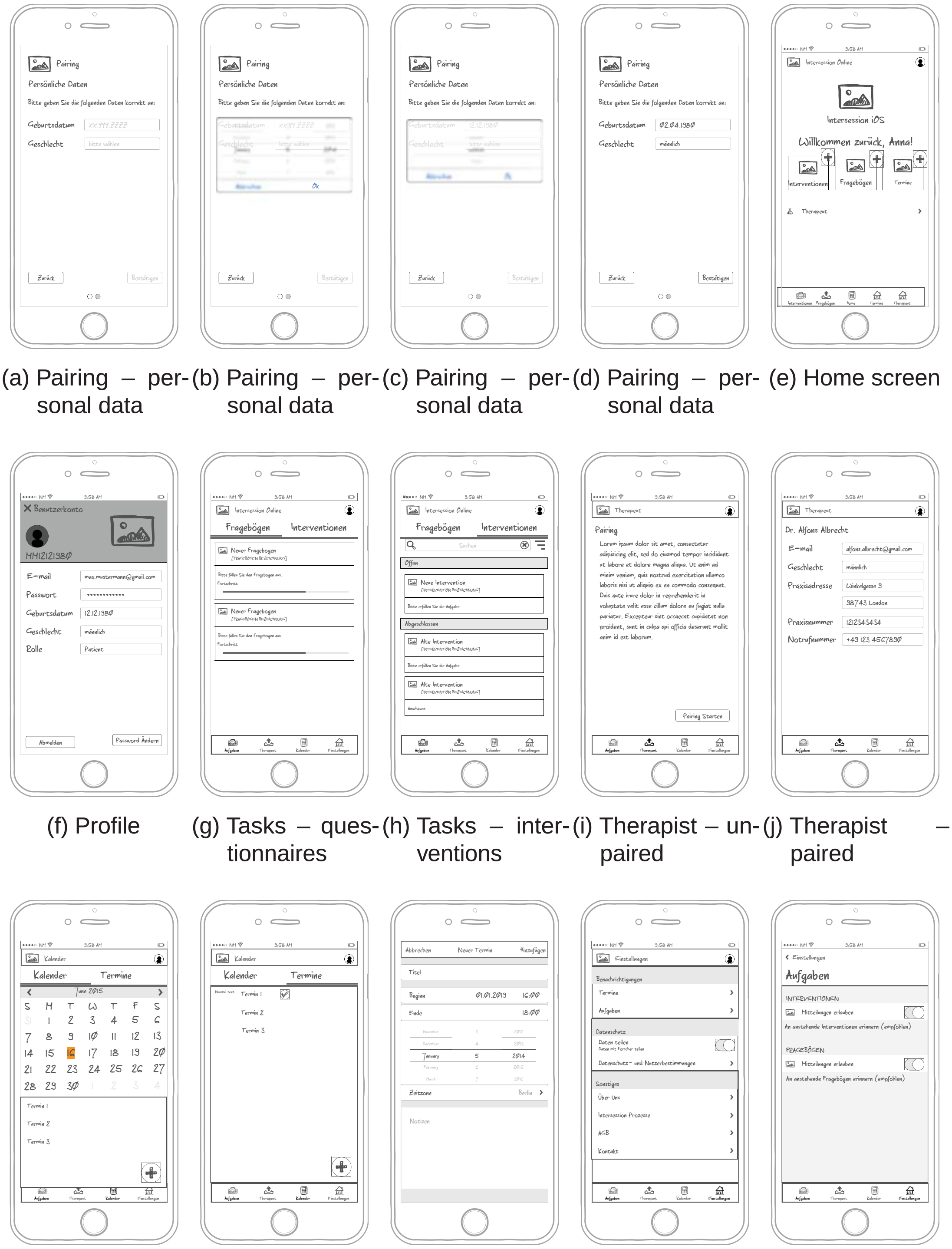

Willkommen
Los Geht's
Intersession Online
Herzlich Willkommen!
Anmelden
Registrieren
Login
E-Mail
Passwort
Sie haben noch kein Intersession Benutzerkonto? Kein Problem! Erstellen Sie sich hier ein neues Benutzerkonto.
Zurück
Login
Registrierung
Information
Datenschutz- und Nutzerbestimmungen
Weiter
Konto erstellen
Wiederholen
Ich stimme hiermit den Datenschutz- und Nutzerbestimmungen zu.
Ich habe die Datenschutz- und Nutzerbestimmungen gelesen und verstanden.
Bitte lassen Sie folgende Einstellung aktiviert:
Ich möchte meine Daten an Forscher weitergeben:
Hinweis: Sie können die Einstellungen jederzeit unter "Einstellungen" ändern.
Akzeptieren
E-Mail Adresse Bestätigen
Erneut senden
Erfolg
Ihre Registrierung ist nun erfolgreich abgeschlossen.
Ihre E-Mail Adresse:
Zum Login
Pairing
Code Eingabe
Bitte geben Sie den vom Therapeuten erhaltenen Pairing Code ein:
Code
Überspringen
Ihr Patienten Kürzel lautet:

(a) Loading (b) Welcome (c) First menu (d) Login (e) Login with credentials

(f) Registration – information (g) Registration – data privacy statement (h) Registration – credentials (i) Registration – accepting the data privacy statement (j) Registration – ready

(k) Registration – email verification (l) Registration – complete (m) Login after registration (n) Pairing – code (o) Pairing – code confirmed

Figure 5.6: Mock-ups – Part 1



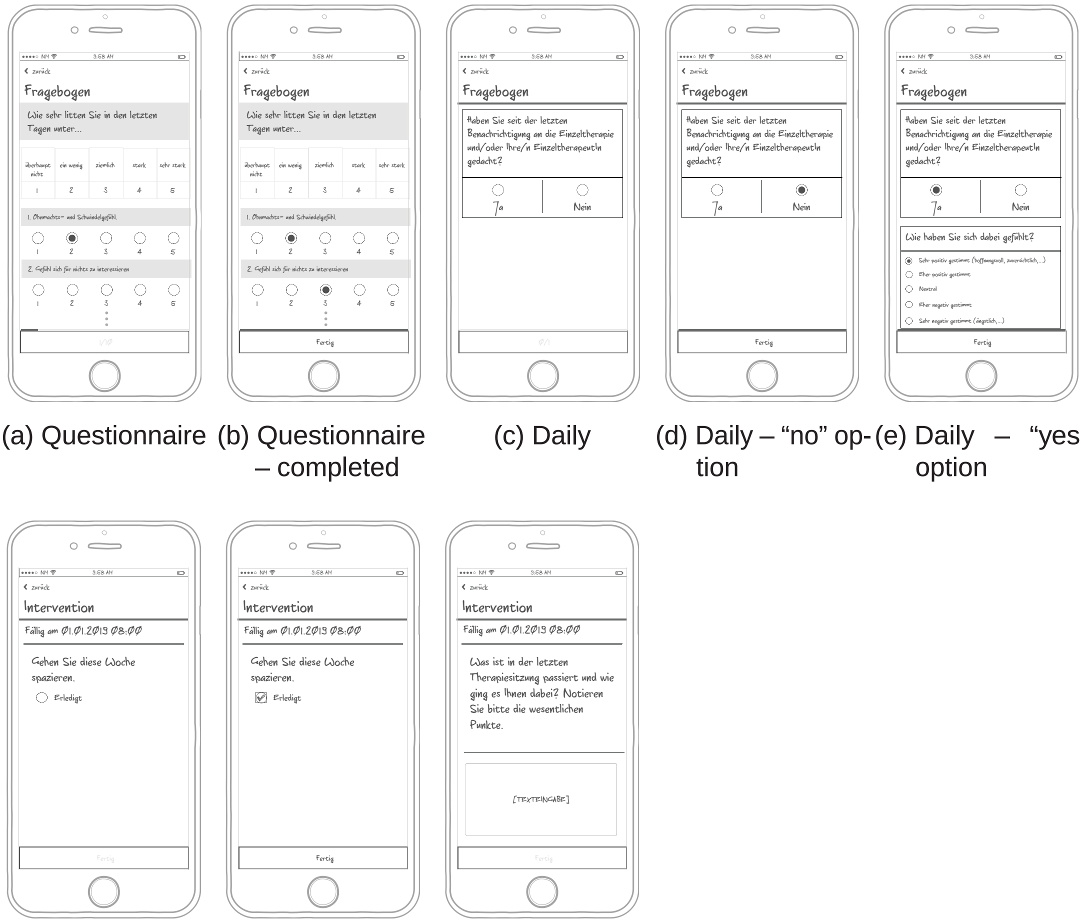
(a) Pairing – personal data (b) Pairing – personal data (c) Pairing – personal data (d) Pairing – personal data (e) Home screen

(f) Profile (g) Tasks – questionnaires (h) Tasks – interventions (i) Therapist – unpaired (j) Therapist – paired

(k) Appointments – calendar (l) Appointments – list (m) Appointments – create new (n) Settings (o) Settings – detail

Figure 5.7: Mock-ups – Part 2

Intervention Figures 5.8f and 5.8g represent an intervention with a task to fulfil. Figure 5.8h shows an intervention with a question and text field.


zurück
Fragebogen
Wie sehr litten Sie in den letzten Tagen unter...
überhaupt nicht
ein wenig
ziemlich
stark
sehr stark
1 2 3 4 5
1. Ohnmachts- und Schwindelgefühl.
2. Gefühl sich für nichts zu interessieren
1/10
Fertig
Haben Sie seit der letzten Benachrichtigung an die Einzeltherapie und/oder Ihre/n EinzeltherapeutIn gedacht?
Ja
Nein
0/1
Wie haben Sie sich dabei gefühlt?
Sehr positiv gestimmt (hoffnungsvoll, zuversichtlich,...)
Eher positiv gestimmt
Neutral
Eher negativ gestimmt
Sehr negativ gestimmt (ängstlich,...)
Intervention
Fällig am 01.01.2019 08:00
Gehen Sie diese Woche spazieren.
Erledigt
Was ist in der letzten Therapiesitzung passiert und wie ging es Ihnen dabei? Notieren Sie bitte die wesentlichen Punkte.
[TEXTEINGABE]

(a) Questionnaire (b) Questionnaire – completed (c) Daily (d) Daily – “no” option (e) Daily – “yes” option

(f) Intervention (g) Intervention – task fulfilled (h) Intervention – question

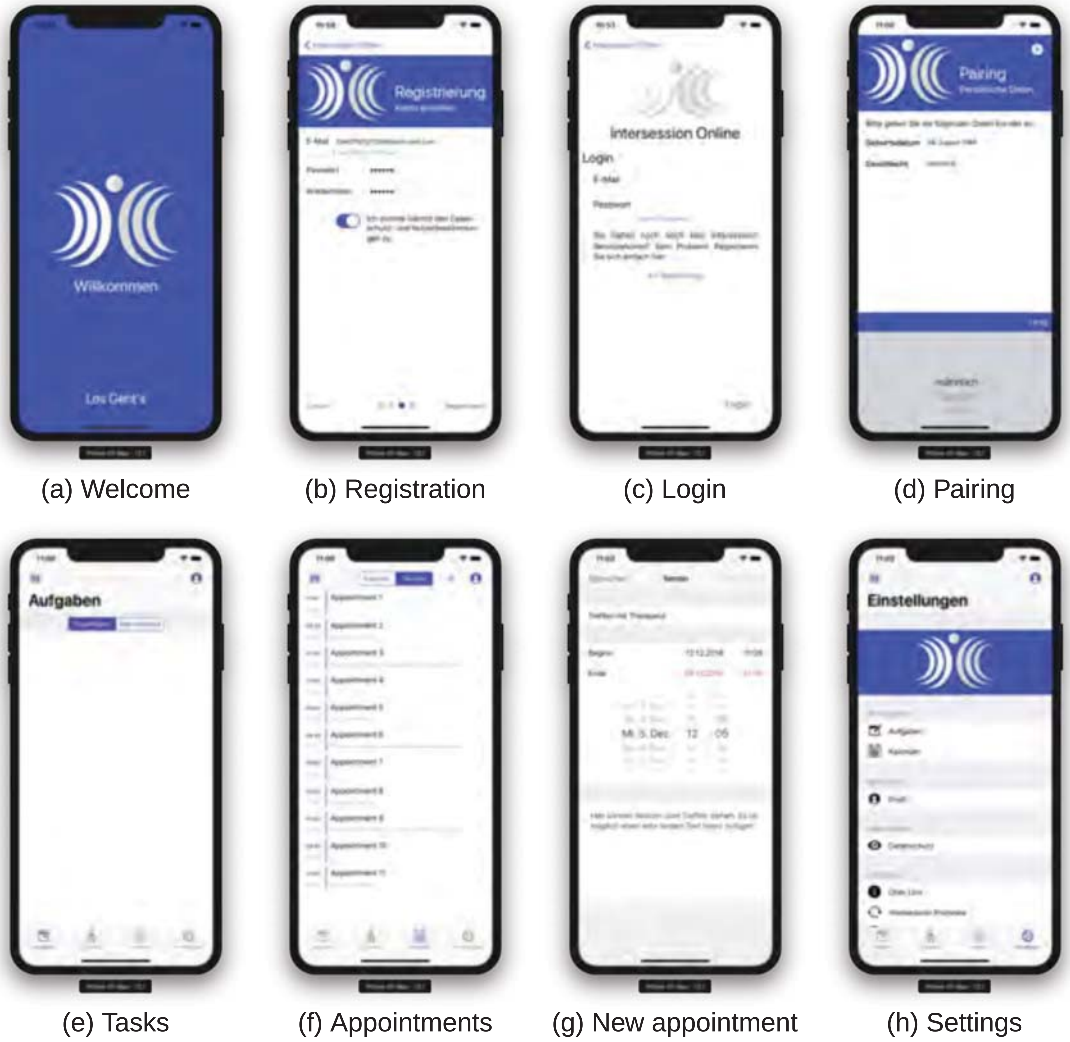
Figure 5.8: Mock-ups – Part 3

5.3.4 High-fidelity prototype

Due to a relatively compact timeframe for the development of the application and the positive feedback on the mock-ups, the design phase has been accelerated by an early implementation of a high-fidelity prototype[7]. This approach not only allowed an early experience of the look and feel of the application, but also saved a lot of time communicating with the customer, which was used in the subsequent implementation. As a positive side effect, the first concrete results could be shown on an early presentation held by the customer about his dissertation progress. Also the communication with the backend system could be tested relatively early and therefore allowed a close cooperation with its developer.

Figure 7.1 shows some screenshots from the high-fidelity prototyping phase. It is to be noted that these screenshots do not represent the final state of the implementation, but give a rough impression of the GUI and a feel of the menu navigation of the application. In Figure 5.9a to 5.9d the first steps with the application, from the welcome screen, over the registration and login, to the pairing process, are shown. Figure 5.9e shows the task view before it was split into a questionnaire and an intervention view, later, during the project progression. Figures 5.9f and 5.9g show a list of appointments and their creation. Figure 5.9h presents the settings tab.

[7] Low-fidelity vs. high-fidelity prototyping: https://www.invisionapp.com/inside-design/low-fi-vs-hi-fi-prototyping/ (accessed 11.08.2019)



(a) Welcome (b) Registration (c) Login (d) Pairing

(e) Tasks (f) Appointments (g) New appointment (h) Settings

Figure 5.9: High-fidelity prototype – early stage screenshots of the implementation

5.4 Architecture

This chapter introduces the underlying architecture model, which serves as the basis for the implementation discussed in Chapter 6. Several relevant aspects, reaching from the GUI, over the data model to the logic, are discussed. Therefore Figure 5.10 provides a simplified overview of the entire architectural model of the application, including the interface for communicating with the backend API. All UML diagrams were created using the web-based tool LucidChart[8].

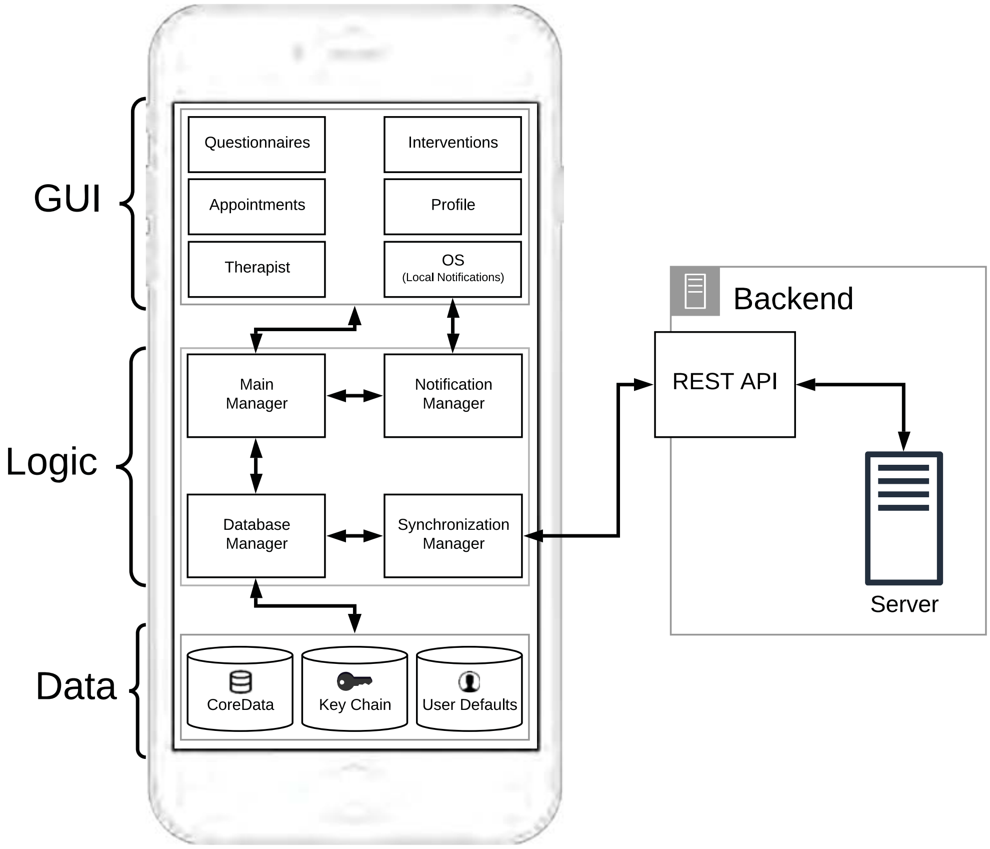


Figure 5.10: Architecture overview

[8] LucidChart: `https://www.lucidchart.com/`

5.4.1 Graphical user interface

Derived from the mock-ups shown in Figures 5.6, 5.7 and 5.8, two UML structure diagrams were created to illustrate the accessibility of the individual views of the application, which the user can navigate through. Figure 5.11 shows which views the user can access, if not yet logged in – essentially the login and registration views.

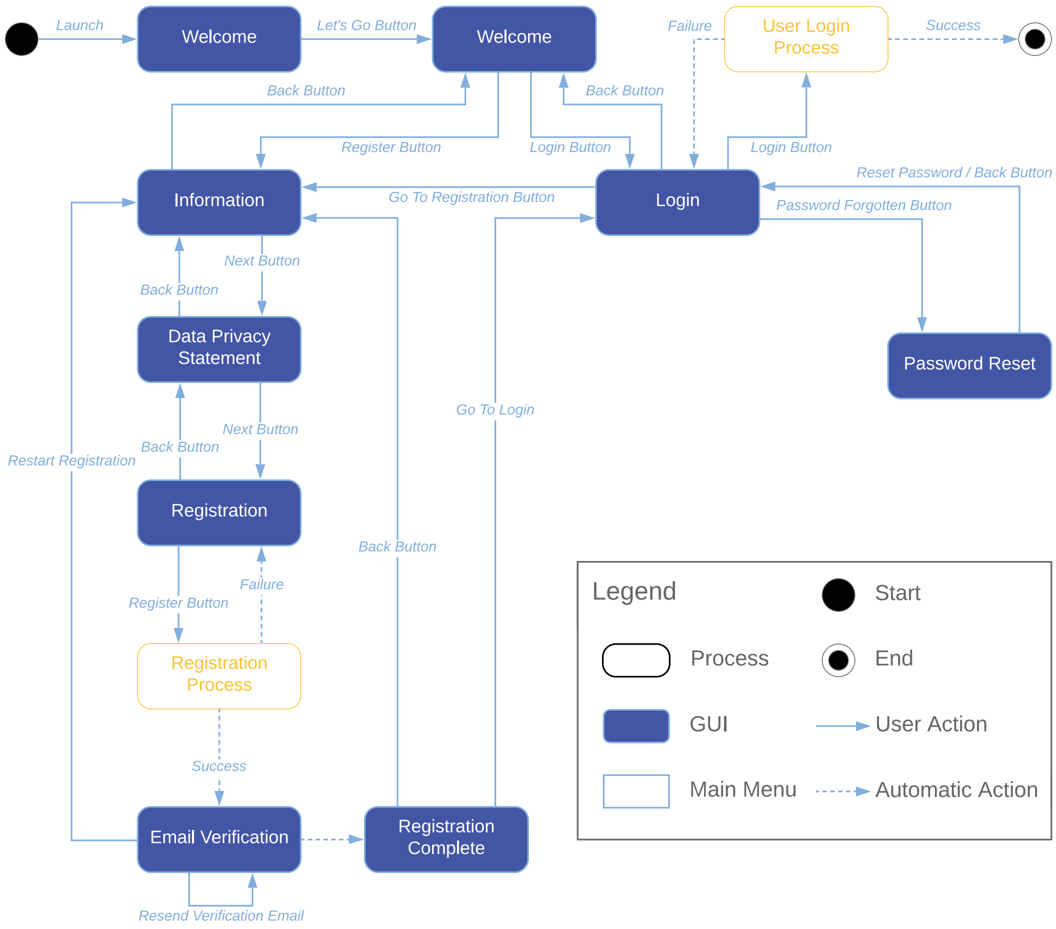


Figure 5.11: Application structure – all views the user can reach when not logged in

Figure 5.12 illustrates the application's navigation paths for already logged-in users. Starting from the intersession tab, all main functionalities can be accessed via the main menu bar. This includes the profile, the therapist, the questionnaires, the interventions, the appointments and the settings.

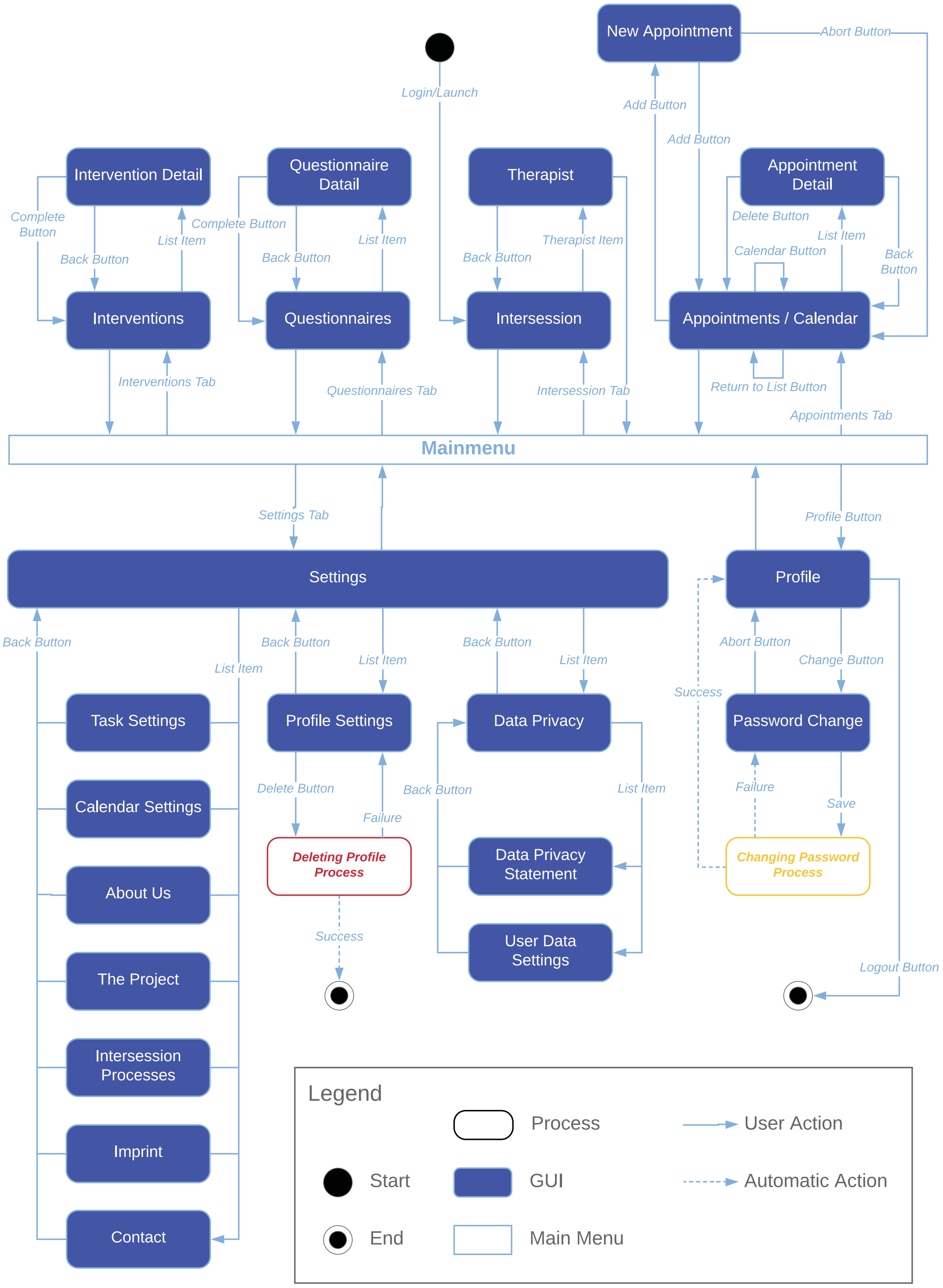


Figure 5.12: Application structure – all views the user can reach when logged in

5.4.2 Data model

Before the logic of the application is presented in section 5.4.3, the underlying data model follows first. Figure 5.13 shows the database schema of the questionnaires, interventions and appointments. Here, the modular structure of the questionnaires and their corresponding relations are illustrated, as explained in the Sections 4.2.5, 4.2.6 and 4.2.7. It should be noted that this scheme allows to link a questionnaire to an assignments or a contribution as well as to store a questionnaire as template independently. At the simplest level, assignments are bound to questionnaires, which still have to be filled in, and contributions link already completed questionnaire data.

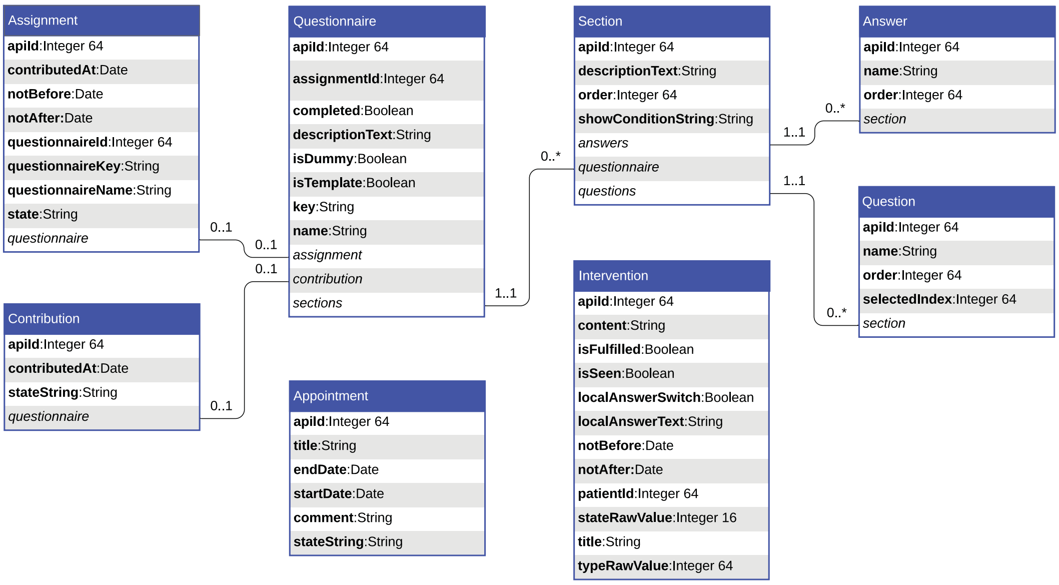


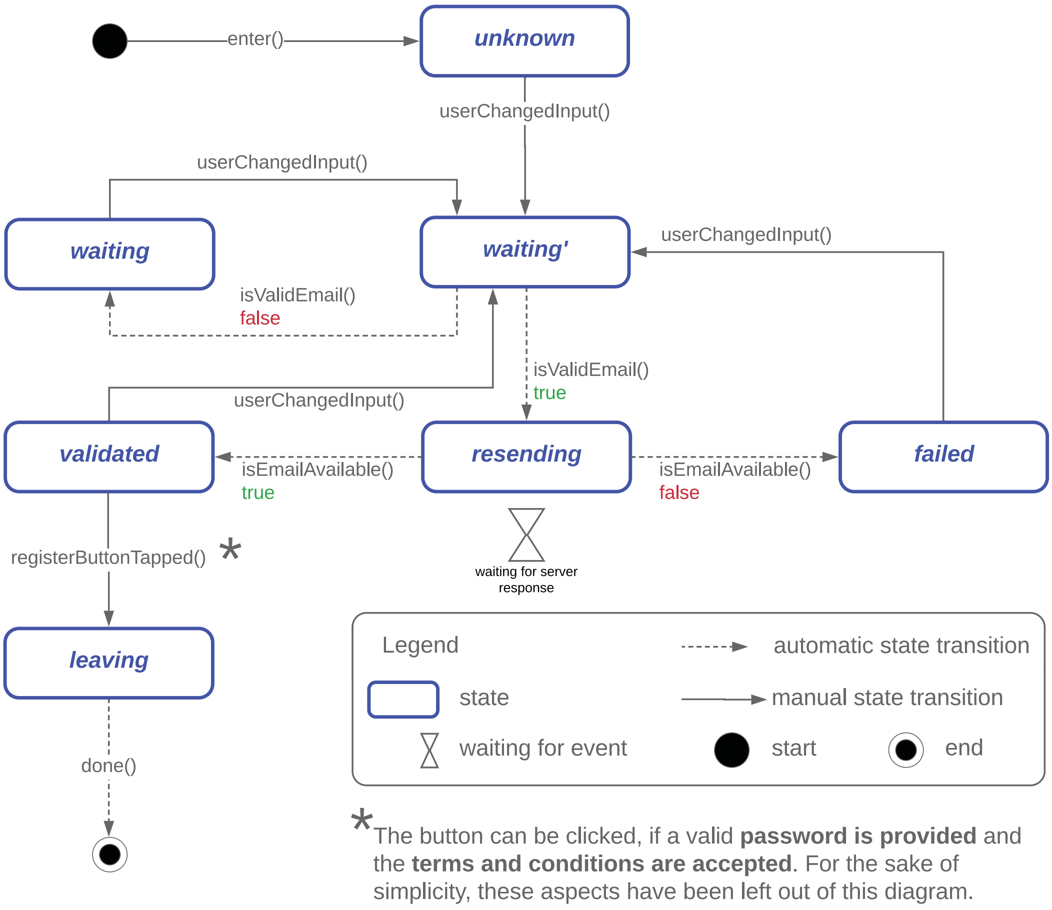
Figure 5.13: Database – entity relationship diagram

5.4.3 Logic

For the application to cope with different situations, several logic components had to be designed. In Section 5.4.4 the mechanism behind specific GUI elements and in Section 5.4.5 the basic concept for the data synchronization with the backend API is explained.

5.4.4 GUI element states

Most interactive GUI elements of the application are bound to state machines, which determine the visual appearance and the interaction possibilities. Inputs of the user or given events, like the ending of a long running process, server responses or the loss of the internet connection, can change these states. By way of example, Figure 5.14 shows a simplified version of the possible states and their corresponding state transitions of the registration button as a UML state diagram. The state of the registration button depends on the user inputs in the “email”, “password” and “password confirmation” text fields, a checkbox and the server response to the “email available” request to the REST API.



*The button can be clicked, if a valid **password is provided** and the **terms and conditions are accepted**. For the sake of simplicity, these aspects have been left out of this diagram.

Figure 5.14: Register button states

5.4.5 Synchronisation and CAP theorem

One of the biggest architectural challenges is the data synchronization with the backend REST API and therefore had to be planned very precisely in advance to the implementation. Because push notifications are not available, periodic or manual requests to the server in a background thread are needed to compare local data with server data.

CAP theorem The three basic constraints of a distributed system are its consistency, availability and partition tolerance – in short CAP. The CAP theorem states, that in a distributed system only two of them can apply at the same time. For this mobile application the most important constraint is the availability. Patients must always be able to access and edit their questionnaires, interventions and appointments, even if no active internet connection is given. This leads to the second required constraint, the partition-tolerance. Data sets can be modified locally in the mobile application while being offline, and hence have to be synchronized later with the server. This results in a partitioning between the server and the mobile application, but has the advantage that the data on the server can be used by a third system participant. Therefore, an algorithm and data structure to tolerate and resolute possible data conflicts must be implemented to ensure the required availability and partition tolerance (AP). [72, 73, 74]

Figure 5.15 shows states and corresponding transitions of an intervention object for the local and backend database. Only “seen” and “submitted” are considered, as explained in Section 4.2.6. The synchronize() method is periodically called by the application and represents a rather complex algorithm to synchronize local and backend data. This method can fail for various reasons, which will result in no change to the local state. A state is always defined by the combination of the local and the external state of a data object. Here, it is important to note, that the methods synchronize(), openIntervention() and submit() are actively called by the mobile application, which therefore has full control over them. The mobile application itself has no influence on external state changes made by the server, such as externalDelete() and externalUpdate(), and therefore can only interrogate external changes via synchronize(). For the sake of simplicity externalUpdate() is left out of Figure 5.15. Following these principles, Figures 5.16 and 5.17 show the states for questionnaire and appointment data objects.

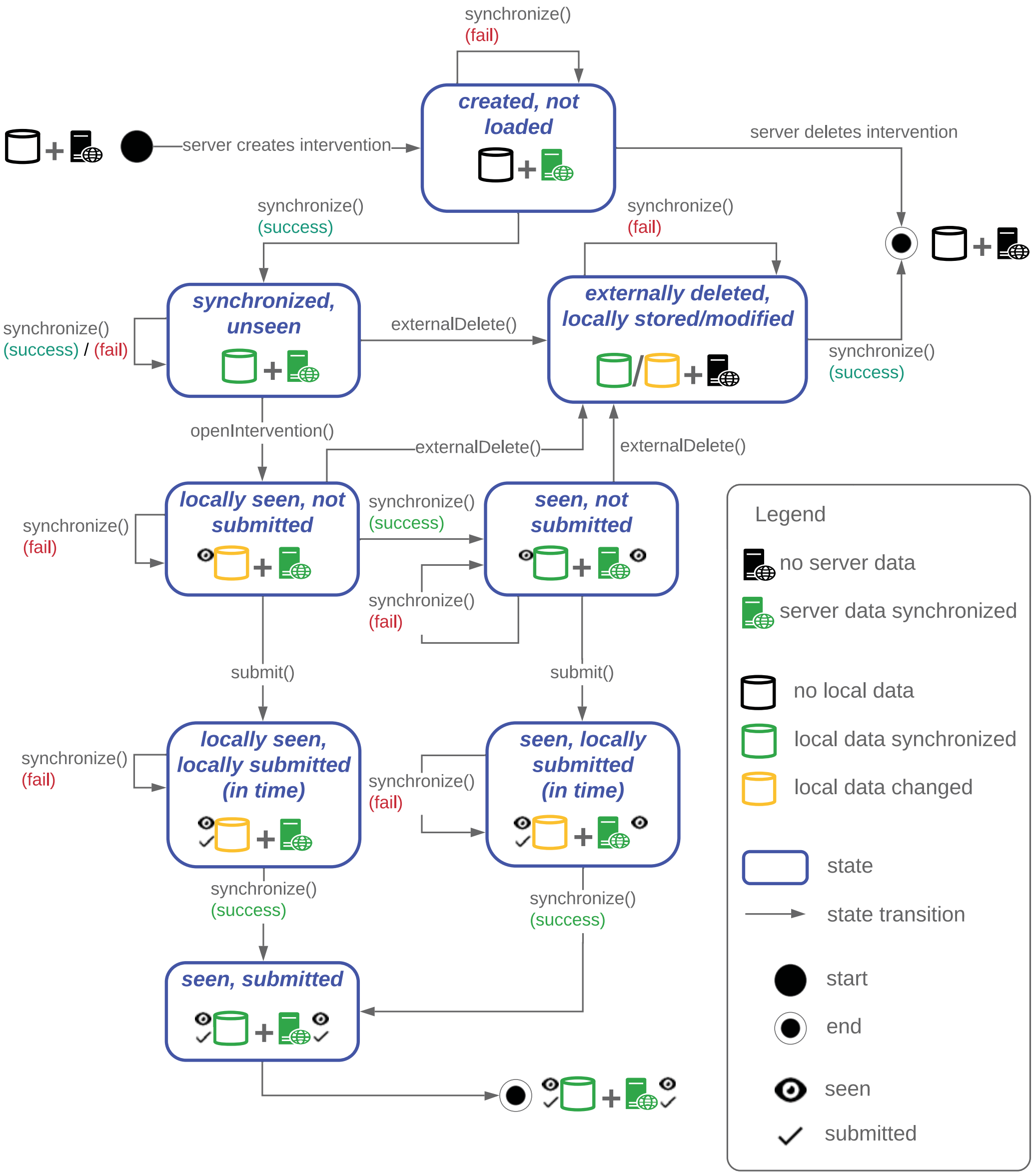


Figure 5.15: Intervention states

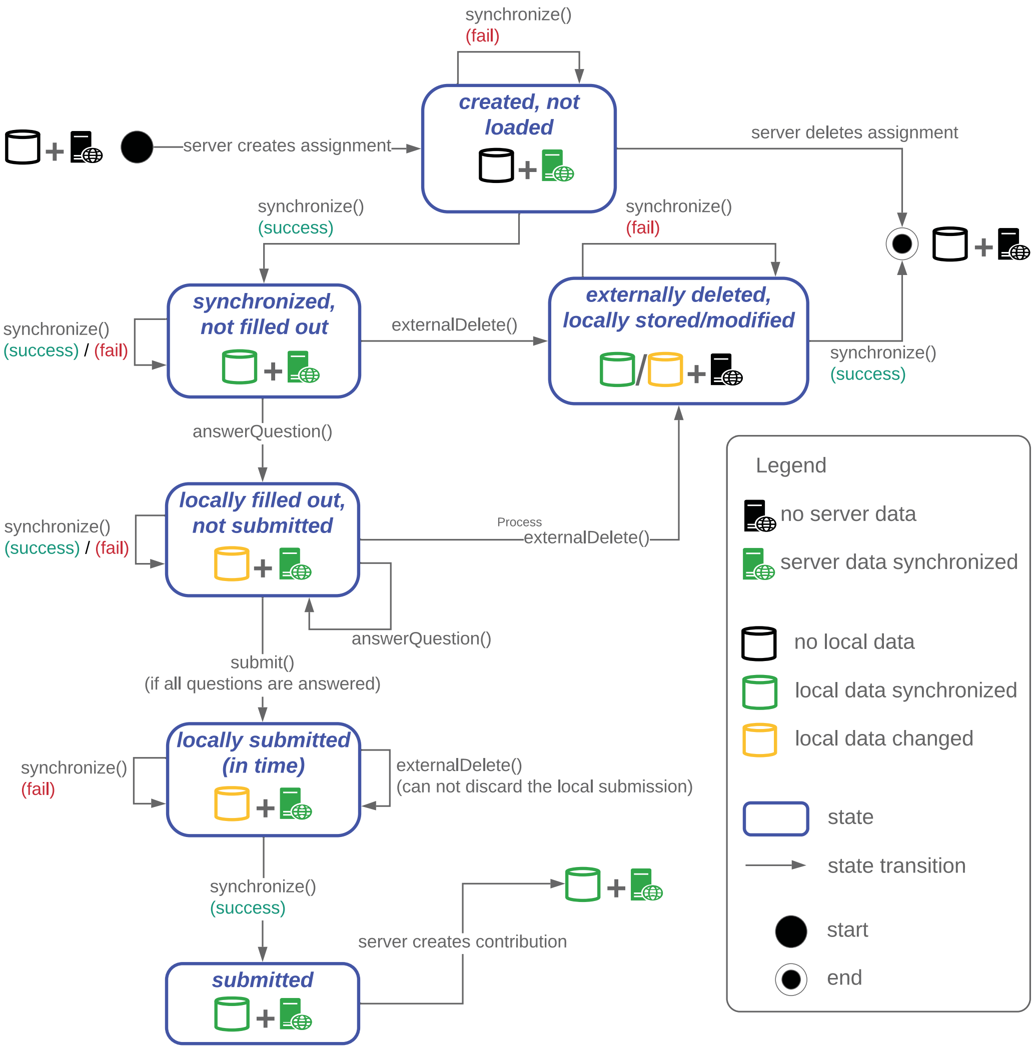


Figure 5.16: Questionnaire states

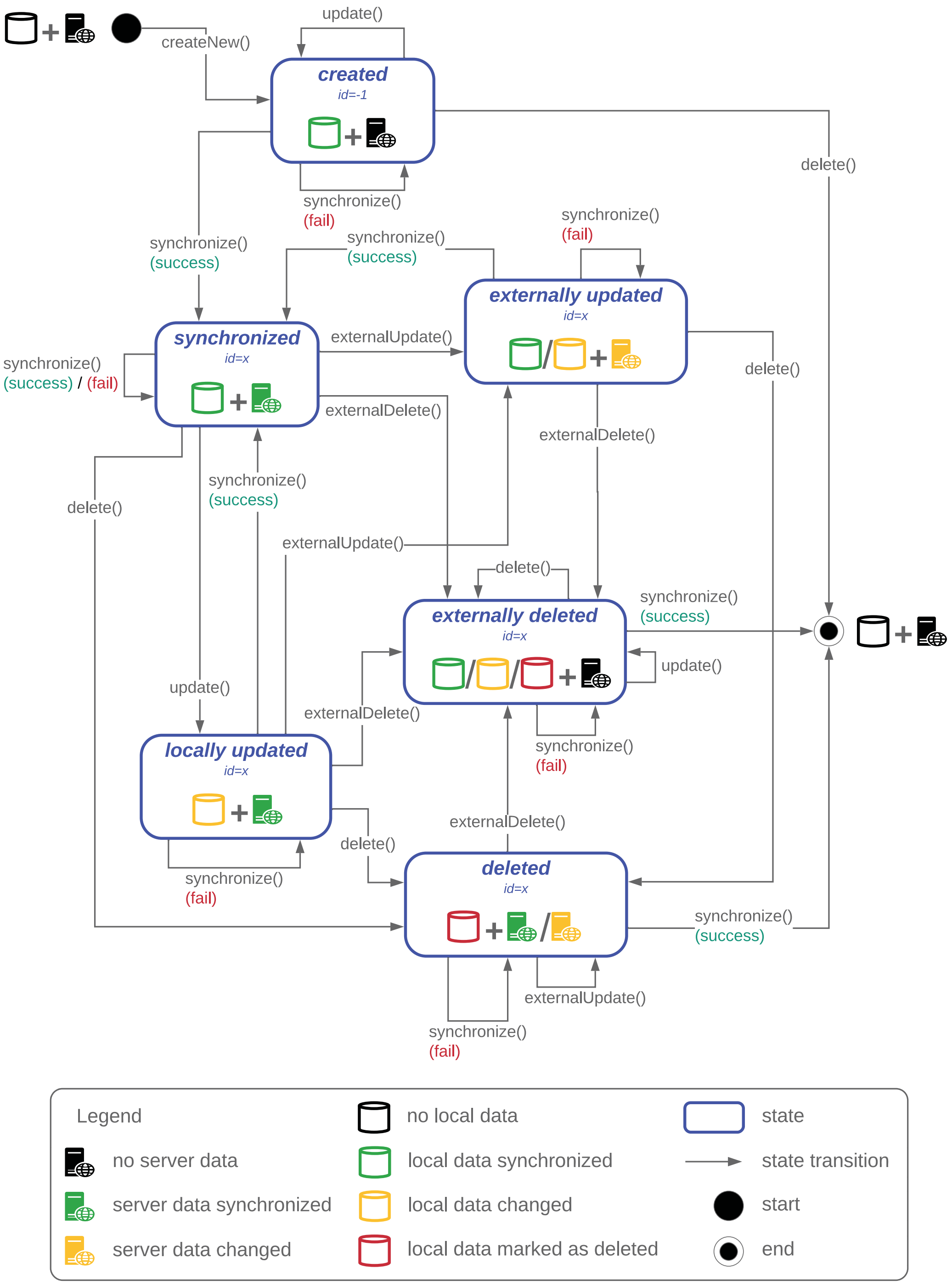


Figure 5.17: Appointment states

The synchronization process is shown in Figures 5.18, 5.19 and 5.20 in the form of UML sequence diagrams, using the example of the questionnaires. This process consists of three consecutive steps:

Step 1: The list of assignments is loaded from the server and compared to the local one. If an assignment is only in the server list, it is added to the local one and marked as new. In case both lists contain the same assignment nothing has to be done. And if only the local list contains an assignment, it must be marked as deleted.

Step 2: In parallel threads all local assignments are examined:

- Case 1 An assignments is new.
  The associated questionnaire must be loaded from the server, if not already present in the local database. When finally the questionnaire is locally available, the assignment can be marked as ready. The loading process can fail, which terminates this thread.
- Case 2 An assignments is ready.
  The associated questionnaires is already loaded, hence nothing has to be done.
- Case 3 An assignments is deleted.
  If the associated questionnaires was completed by the patient in time, it must be submitted to the server. If the submission was successful or the local completion to late / not done at all, the data has to be locally removed afterwards. The submission can fail, which terminates this thread.

Step 3: The list of contributions is loaded from the server and the local one adapted to the server version. In parallel threads all local contributions are examined and missing questionnaire and contribution data loaded. The loading process can fail, which terminates this thread.

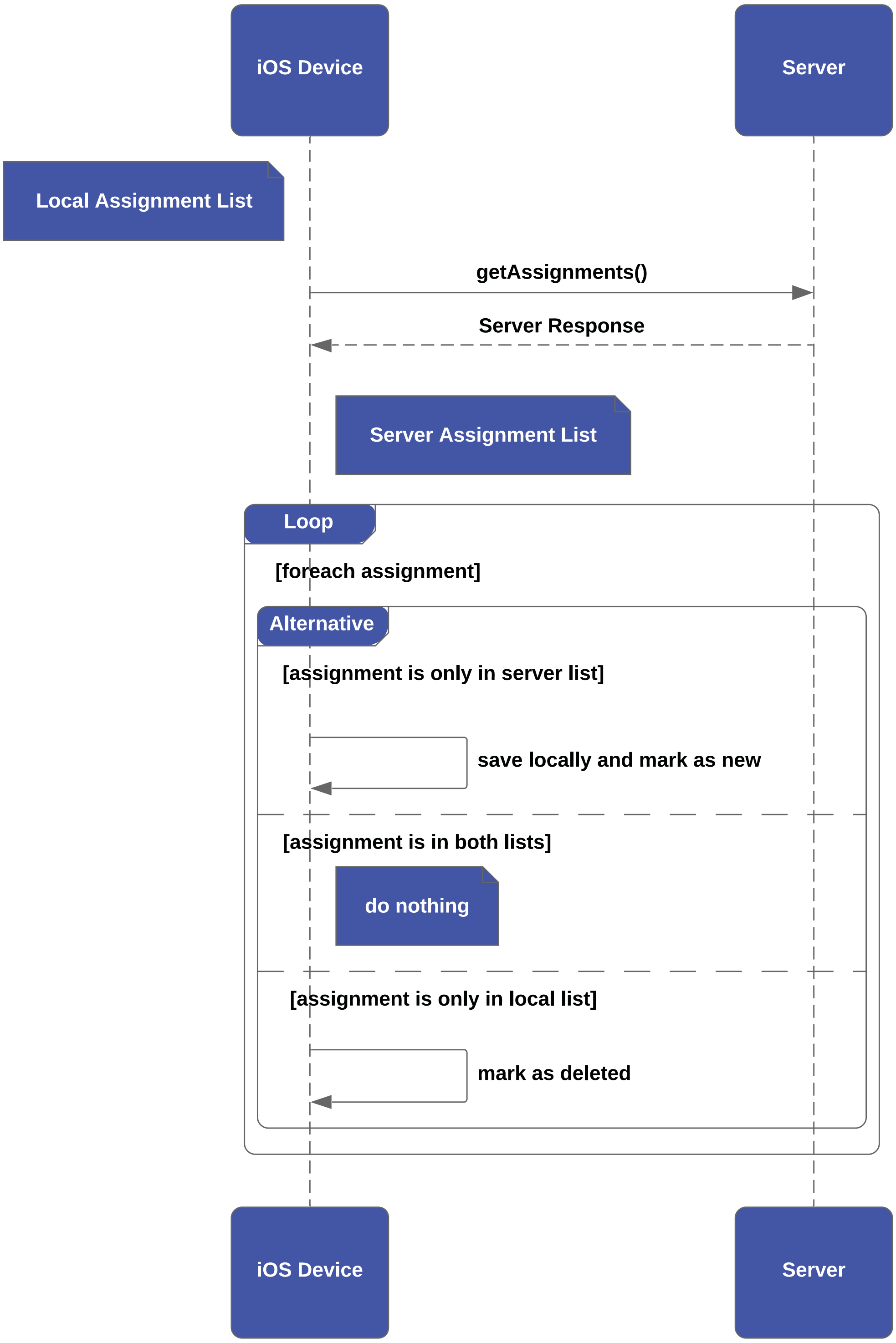


Figure 5.18: Assignments synchronization – sequence diagram

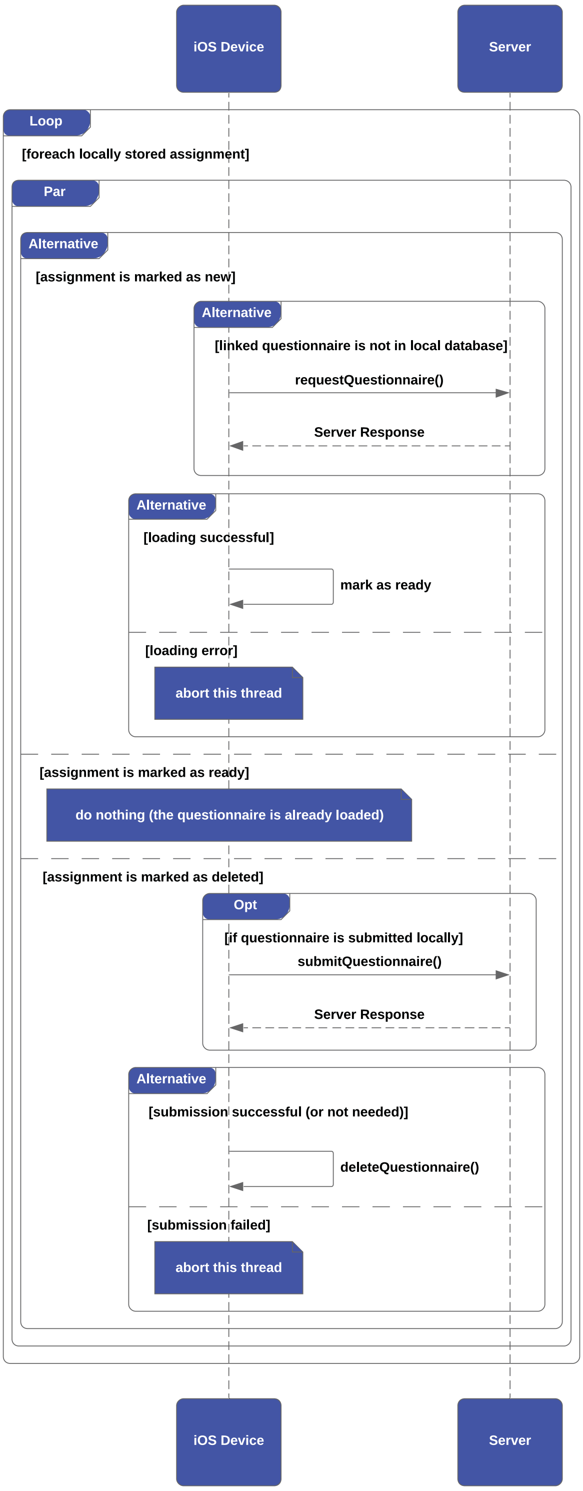

iOS Device
Server
Loop
[foreach locally stored assignment]
Par
Alternative
[assignment is marked as new]
Alternative
[linked questionnaire is not in local database]
requestQuestionnaire()
Server Response
Alternative
[loading successful]
mark as ready
[loading error]
abort this thread
[assignment is marked as ready]
do nothing (the questionnaire is already loaded)
[assignment is marked as deleted]
Opt
[if questionnaire is submitted locally]
submitQuestionnaire()
Server Response
Alternative
[submission successful (or not needed)]
deleteQuestionnaire()
[submission failed]
abort this thread
iOS Device
Server

Figure 5.19: Questionnaires synchronization – sequence diagram

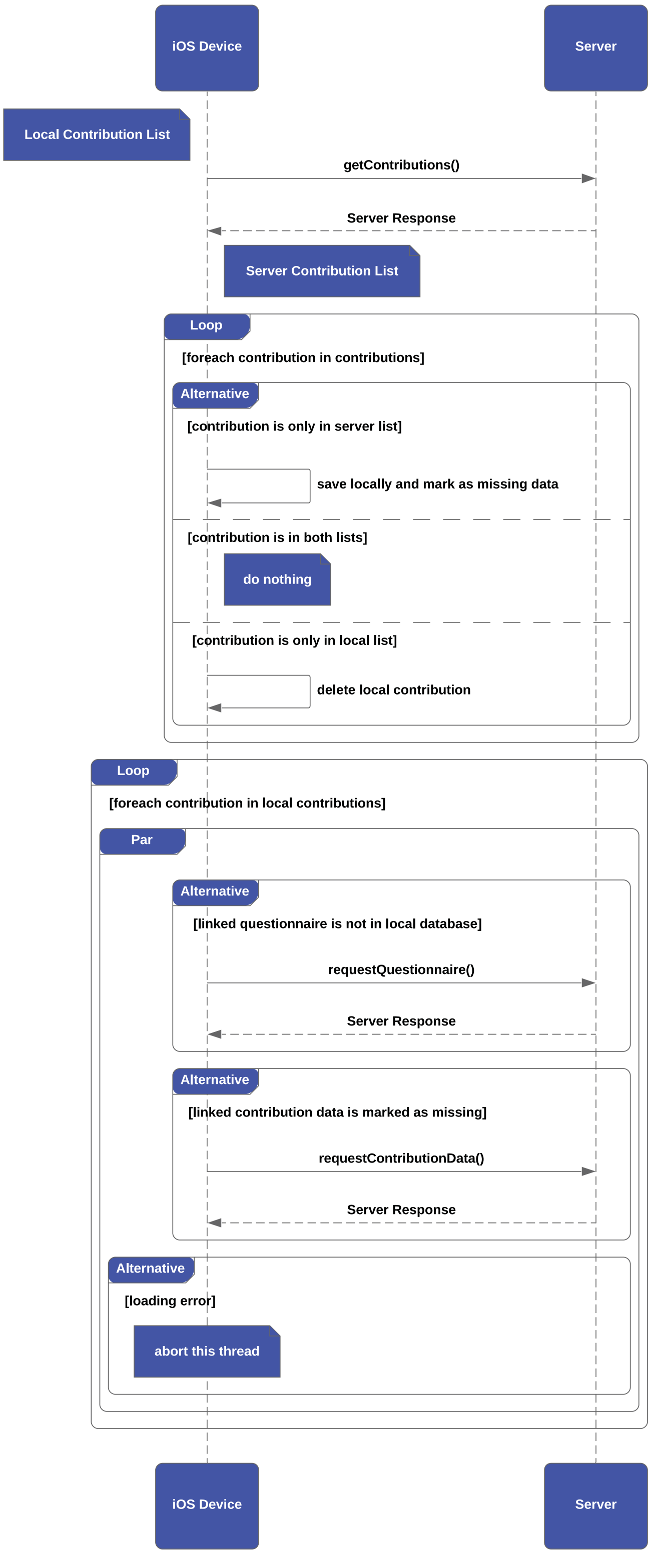

iOS Device
Server
Local Contribution List
getContributions()
Server Response
Server Contribution List
Loop
[foreach contribution in contributions]
Alternative
[contribution is only in server list]
save locally and mark as missing data
[contribution is in both lists]
do nothing
[contribution is only in local list]
delete local contribution
Loop
[foreach contribution in local contributions]
Par
Alternative
[linked questionnaire is not in local database]
requestQuestionnaire()
Server Response
Alternative
[linked contribution data is marked as missing]
requestContributionData()
Server Response
Alternative
[loading error]
abort this thread
iOS Device
Server

Figure 5.20: Contributions synchronization – sequence diagram

6

Implementation

This chapter introduces the technical implementation of the requirements defined in Chapter 4 and the designs modelled in Chapter 5. In addition to various technical aspects, selected essential components of the implementation are discussed in detail.

6.1 Technology and tooling

Before the implementation could begin, some fundamental decisions had to be made regarding the technology and tooling.

The very basic thing to consider for the implementation are the programming language and frameworks suitable for the target platform. In the case of the iOS platform, various approaches are initially available. The most common approaches for iOS are native, (progressive) web-based, hybrid or cross-platform code bases. Since the Android version was already in progress, this had a strong impact on the decision, so that a cross-platform development with Xamarin[1] or React Native[2] did not seem reasonable. While web-based applications are automatically cross-device and hot-pluggable, they lack some key features, such as push notification, offline availability, and native user experience, and are therefore out of question. Hybrid apps can overcome some of these issues, but still lack in performance, have a high error potential caused by the interface of native and web-based components, and tend not to have a familiar look and feel.

[1] https://dotnet.microsoft.com/apps/xamarin

[2] https://www.reactnative.com

For these and other reasons a native a code base was the best solution. The modern programming language Swift 5[3] was preferred over Objective-c[4].

Based on that decision Xcode[5] was the integrated development environment of choice. As a distributed version management system, Git[6] was chosen in combination with the web-based file hosting service for software development projects BitBucket[7] and the Git GUI client GitKraken[8]. To handle third party code libraries the decentralized dependency manager Carthage[9] was chosen over Cocoapods[10] and the relatively new Swift Package Manager[11]. The main reasons for that decision was the support of all relevant third party libraries with this manager and its simplicity. Other tools like the Terminal[12], VSCode[13], Postman[14] and Swagger[15] were used for several reasons as well as the Xcode iOS Simulator and real iOS hardware for testing purposes. To structure the development process the BitBucket issue tracker was utilized and to document the project a Confluence[16] page created.

6.2 Conventions and principles

In order to produce reusable, maintainable and readable program code and a clear project structure several patterns and principles were applied.

As basis of a good code design served the Apple API Design Guidelines[17] with the following fundamental goals:

[3] https://swift.org
[4] https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/
[5] https://developer.apple.com/xcode/
[6] https://git-scm.com
[7] https://bitbucket.org/
[8] https://www.gitkraken.com
[9] https://github.com/Carthage/Carthage
[10] https://cocoapods.org
[11] https://swift.org/package-manager/
[12] https://support.apple.com/de-de/guide/terminal/welcome/mac
[13] https://code.visualstudio.com
[14] https://www.getpostman.com
[15] https://swagger.io
[16] https://www.atlassian.com/de/software/confluence
[17] https://swift.org/documentation/api-design-guidelines/

- Clarity at the point of use
- Clarity is more important than brevity
- Write a documentation comment

Supplementary the implementation strongly follows the DRY [18] principle to reduce redundancy, whenever possible. Therefore, components are divided into small logically connected reusable units. Whenever more than two almost identical code snippets occurred, they where rewritten to one generic component. The above-mentioned guidelines also correlate with the KISS[19] principle. This keeps the code simple and makes it understandable.

Also various programming patterns where followed with a certain extent of consciousness like singleton, as shown in Listing 1, MVC, delegation and observer for example. Components are mostly designed following SOLID[20]:

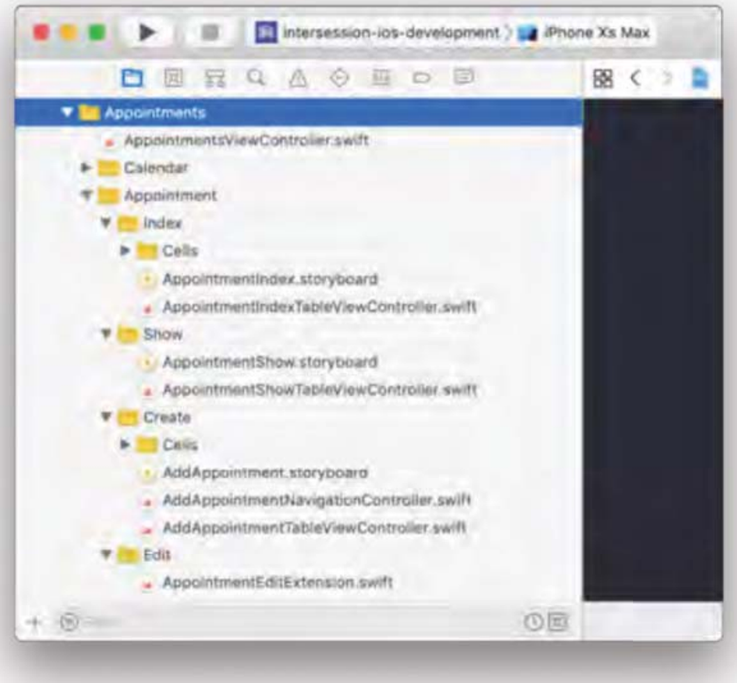
- Single responsibility: a class has a single responsibility
- Open–closed: open for extension, closed for modification
- Liskov substitution: subtypes can replace instances without altering the correctness
- Interface segregation: not needed here
- Dependency inversion: details should depend upon abstractions, not vice versa

The general folder structure and the class naming convention of the project is oriented on web-development projects with RoR[21] or Laravel[22], but transferred to a Swift Project, as Figure 6.1 shows. This principle is mostly used for lists, treated as the "index", with detail views as the "show" equivalent for their entries, etc., resembling the CRUD[23] methods. Therefore grouping logical related components is, preferred over a grouping by MVC.

[18] Don't Repeat Yourself
[19] Keep It Simple, Stupid
[20] https://williamdurand.fr/2013/07/30/from-stupid-to-solid-code/
[21] Ruby on Rails: https://rubyonrails.org
[22] https://laravel.com
[23] Create, Read Update, Delete

```
class DatabaseSemaphore {

    // The DispatchSemaphore only provides one execution slot,
    // hence parallel calls will be executed sequentially.
    private static let defaultNumberOfParallelThreads: Int = 1

    // MARK: - Properties

    private static let databaseSemaphore: DatabaseSemaphore = {
        return DatabaseSemaphore(numberOfParallelThreads:
  ↪ defaultNumberOfParallelThreads)
    }()

    private let semaphore: DispatchSemaphore

    // MARK: - Initialization

    private init(numberOfParallelThreads: Int) {
        self.semaphore = DispatchSemaphore(value:
  ↪ numberOfParallelThreads)
    }

    // MARK: - Functions - Accessors

    fileprivate class func enter() {
        // Thread must wait for a free slot in the semaphore.
        self.databaseSemaphore.semaphore.wait()
    }

    fileprivate class func leave() {
        // Thread releases the reserved slot in the semaphore.
        self.databaseSemaphore.semaphore.signal()
    }

}
```

Listing 1: Database semaphore realised as singleton provides the static methods enter() and leave() and prevents instantiating other objects of itself by private init(...)



```
Appointments
    AppointmentsViewController.swift
  Calendar
  Appointment
    Index
      Cells
      AppointmentIndex.storyboard
      AppointmentIndexTableViewController.swift
    Show
      AppointmentShow.storyboard
      AppointmentShowTableViewController.swift
    Create
      Cells
      AddAppointment.storyboard
      AddAppointmentNavigationController.swift
      AddAppointmentTableViewController.swift
    Edit
      AppointmentEditExtension.swift
```

Figure 6.1: Folder and class naming conventions

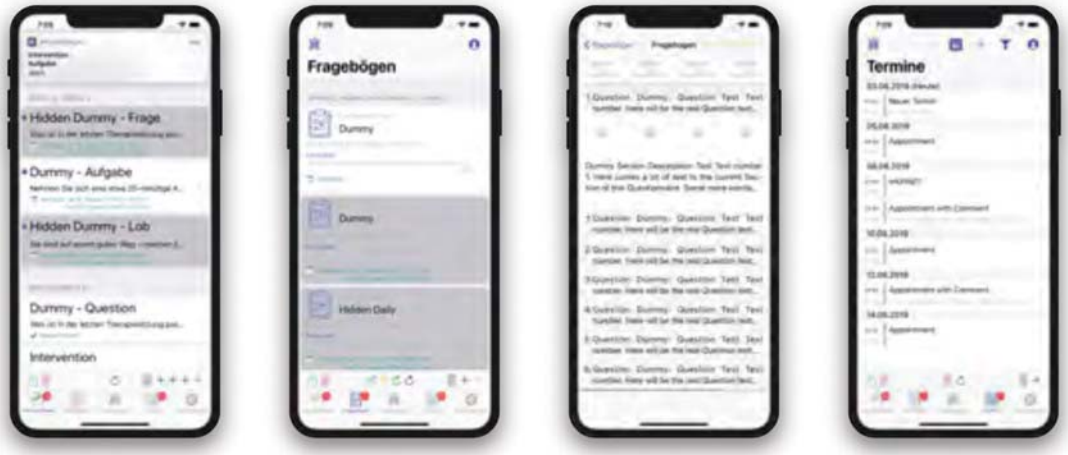
6.3 Data management and security

The application differentiates three major ways of persisting data, as shown in Figure 5.10. The highly confidential login credentials and the login token are securely stored in the iOS Keychain. Settings made in the application are not confidential and therefore can be persisted in the UserDefaults. To store large data sets, such as questionnaire or intervention responses and appointments, the application uses the Coredata library.

One of the biggest concerns here is: What happens to the data, when the user logs out? If the user logs out manually, the local CoreData database is wiped with the exception of the intervention data. Since the intervention data is only held locally and therefore would be lost, the patient must activate the deletion by setting an appropriate setting. In the event that another account logs in on the same device on which foreign intervention data is still stored, this data will be removed. By logging out and in again data is reloaded from the server. Login credentials are kept in the iOS Keychain if the user set the corresponding setting and are overwritten by logging in with an other account. If the login token expires the user has to log in again.

6.4 Development vs. production build

Since the application is capable to present dynamically generated questionnaires, which may not be available in the backend, it was necessary to setup and implement a testing scheme. An Xcode scheme defines a collection of build targets and the configuration to use when building them. Using different configurations for development and production builds, the application adds or removes toolbars with function buttons. Figure 6.2 visualizes the application when build with the development configuration. With these buttons different interventions (6.2a), questionnaires (6.2b, 6.2c) and appointments (6.2d) can be generated to test the system. Also several other tasks like a partial or complete database wipes, the deletion of the login token and manual synchronization can be performed. These features help to examine system and are especially useful for testing the notification feature (6.2a) and corner cases of questionnaires (6.2c). Also hidden elements are made visible (6.2b).



(a) Interventions (b) Questionnaires (c) Corner case questionnaires (d) Appointments

Figure 6.2: The application build with the development configuration.

In order to realize the schema change at the push of a button, various configurations had to be made. Two Xcode schemes were configured, each with an own ".xcconfig"-file to store properties, which had to be registered to the projects "Info.plist"-file. For the signing process two different accounts were deposited, one for debug and one for release builds. Finally, an adapter class was implemented for the comfortable use of property values.

6.5 Selected implementation aspects

This section copes with interesting aspects of the implementation at the example of three important components of the application.

6.5.1 UI interaction state machines at the example of the registration view

Section 5.4.4 described the principle of state machines behind the appearance of interactive GUI elements. This section examines the implementation of the state machine showed in Figure 5.14 in greater detail.

The first component is a data type definition of possible GUI states as shown in Listing 2. For each state a human readable description text was defined, which can displayed to the user if needed. Note that the "failed" state has two arguments, the API status code and the corresponding message, which both are put into the description.

In Listing 3 a property of the above data type is declared, which implements the Swift "didSet"-method to react on state changes. According to the current state various properties are set, the corresponding method to update control elements of the view is called and background processes are started. The "canProceed" property indicates whether or not the user can click the register button. The "setControlsFor" methods style the GUI representing the current state and enable or disable certain controls, as shown in Listing 4. Note that the callback of methods like "checkEmailValidationStatusWithTimer()" can also change the "requestState".

```swift
enum EmailValidationState: CustomStringConvertible {
    case unknown
    case validated
    case waiting
    case resending
    case failed(
        API.Email.Resend.HTTPStatusCodes?,
        HTTPStatusCodeMessage?
    )
    case leaving

    var description: String {
        switch self {
        case .unknown: return ""
        case .validated: return "E-Mail-Adresse verfügbar"
        case .waiting: return "E-Mail-Adresse wird bereits verwendet"
        case .resending: return "Sende erneut..."
        case .failed(let statusCode, let message): return "Fehler mit Statuscode \(statusCode?.rawValue ?? -1) - Nachricht: \(message?.description ?? "")"
        case .leaving: return "Fertig..."
        }
    }
}
```

Listing 2: RegistrationValidateEmailAddressViewController.swift – requestState enum definition

6.5.2 Synchronization architecture

In this section the background synchronization and the communication with the backend API are discussed.

Background synchronization

To give a pleasant feeling of use, to avoid long loading times and not to interrupt the interaction flow nearly all synchronization processes are hidden in background threads, while the user is still able to use the application in its full extend. All components of the application, which have to be synchronized, register their synchronization method to the SynchronizationHandler, which takes care of the following aspects:

- Starting and stopping the synchronization process according to the application's life cycle[24].
- The synchronisation interval, which depends on whether or not the corresponding view is in the fore- or background.
- Handling instant synchronization operations, which occur when the user manually wants to synchronize, for example by pulling down a UITableView, or when the user modified the local data.

A big concern when implementing the synchronization feature was the spam filter of the backend API, which gets triggered by to many requests in a certain time interval. Essentially, this was about not to sync in too short intervals, but often enough to keep the application data up to date. The SynchronizationHandler can cope with that.

[24] "Managing Your App's Life Cycle": `https://developer.apple.com/documentation/uikit/app_and_environment/managing_your_app_s_life_cycle`

```swift
private var requestState: EmailValidationState = .unknown {
    didSet {
        NSLog(String(describing: self) + " - requestState:
 ↪  \(self.requestState)")
        switch self.requestState {
        case .unknown:
            self.canProceed = false
            self.setControlsForUnknown()
            self.checkEmailValidationStatusWithTimer()
        case .waiting:
            self.canProceed = false
            self.setControlsForWaiting()
            self.checkEmailValidationStatusWithTimer()
        case .validated:
            self.canProceed = true
            self.setControlsForEmailValidated()
            self.notifyAll()
        case .failed(let statusCode, let message):
            self.canProceed = false
            self.setControlsForFailed()
            self.showRequestStatus(
                code: statusCode?.rawValue,
                description: statusCode?.description,
                message: message
            ) {
                _ in

                self.checkEmailValidationStatusWithTimer()
            }
        case .resending:
            self.stopCheckEmailValidationStatusWithTimer()
            self.canProceed = false
            self.setControlsForResending()
            self.resendValidationEmail()
            break
        case .leaving:
            self.stopCheckEmailValidationStatusWithTimer()
        }
    }
}
```

Listing 3: RegistrationValidateEmailAddressViewController.swift – request state

```swift
private func setControlsForResending() {
    // email information
    self.emailInformationButton.isEnabled = false

    // checking status
    self.checkingStatusLabel.isHidden = true
    self.checkingStatusActivityIndicator.stopAnimating()

    // validated
    self.emailValidatedLabel.isHidden = true

    // resend
    self.resendButton.isEnabled = false
    self.resendButton.isHidden = false
    self.resendButton.isUserInteractionEnabled = false
    self.resendActivityIndicator.startAnimating()
}
```

Listing 4: RegistrationValidateEmailAddressViewController.swift – setting controls for resending

Synchronization API

The API to synchronize local with server data consists of multiple components:

- The main component is the NetworkHandler, containing the makeRequest(...) method, as shown in Listing 5. This method is at the heart of communicating with the REST API. Virtually any type of object that matches the Encodable property can be send to the API, and Decodable objects can be accepted from the server response accordingly. The sending and expected reception types can be set individually and are send as JSON in the HTTP body. Received responses are evaluated, the data decoded and handed over to the completionHandler method with a corresponding statusCode and/or if necessary a message object. Errors are intercepted and treated accordingly.
- Possible URLRequest objects are predefined in an API class, which also provides expected HTTPStatusCodes corresponding to these requests. This allows to create an URLRequest object for a specific questionnaires by simply calling let request = API.Questionnaires.get(id).
- The last part is the actual method initiating the communication and coping with the results, depending on the use case. In Listing 6 the corresponding version for loading a specific questionnaire is presented. The method takes a questionnaire ID, a success and an error handler as arguments, utilizes the URLRequest mentioned above and uses the NetworkHandler to make a request. The request result is treated asynchronously with the adequate handler.

```swift
static func makeRequest<T: Decodable, U: Encodable>(
    as requestFunction: (Data) -> URLRequest,
    for decodable: T.Type,
    with bodyData: U,
    completionHandler: @escaping (
        T?, Int?, HTTPStatusCodeMessage?
    ) -> Void
) {
    guard let jsonData = encodeJSON(bodyData) else {
        return
    }

    let request = requestFunction(jsonData)

    URLSession.shared.dataTask(with: request) {
        (data, response, error) in

        var statusCode: Int?
        let dataTaskIsValid = validateDataTask(data: data, response: response, error: error, statusCode: &statusCode)

        if dataTaskIsValid,
            let apiData = decodeJSON(decodable, from: data)
        {
            if let message = decodeJSON(
                HTTPStatusCodeMessage.self, from: data
            ) {
                completionHandler(apiData, statusCode, message)
            } else {
                completionHandler(apiData, statusCode, nil)
            }
        } else if dataTaskIsValid,
            let message = decodeJSON(HTTPStatusCodeMessage.self, from: data)
        {
            NSLog(message)
            completionHandler(nil, statusCode, message)
        } else {
            NSLog("ERROR: Request is not valid.")
            completionHandler(nil, statusCode, nil)
        }
    }.resume()
}
```

Listing 5: NetworkHandler.swift – API request method

```
internal static func getQuestionnaire(
    with id: String,
    success: @escaping (APIModels.Questionnaire?) -> (),
    error: @escaping (
        _ httpStatusCodes: API.Questionnaires.HTTPStatusCodes?,
        _ message: HTTPStatusCodeMessage?
    ) -> ()
) {
    let request = API.Questionnaires.get(id)

    NetworkHandler.makeRequest(
        as: request,
        for: APIModels.Questionnaire.self
    ) {
        (apiData, statusCode, message) in

        // Asynchronous.
        DispatchQueue.main.async {
            // Check if status code is a valid status code
            // from the API definitions.
            if apiData != nil || message != nil,
                statusCode != nil,
                let statusCode =
  ↪  API.Questionnaires.HTTPStatusCodes(rawValue: statusCode!)
            {
                switch statusCode {
                case .ok:
                    success(apiData)
                default:
                    error(statusCode, message)
                }
            } else {
                error(
                    nil,
                    HTTPStatusCodeMessage(
                        message: "Fatal Error",
                        errors: nil
                    )
                )
            }
        }
    }
}
```

Listing 6: Request for loading a single questionnaire with a specific ID

6.5.3 Algorithm to load questionnaires

The synchronization was essentially implemented as described in Section 5.4.5 and visualized in Figures 5.18, 5.19 and 5.20. The synchronization of questionnaires is tied to the synchronization of assignments and contributions. Since the entire process spans a gigantic tree of operations, all steps have been divided into 10-20 line methods, nested inside each other, some of which are shown below.

The synchronization process begins with the synchronization of the assignments. As the competing methods work on the same database, a semaphore will be entered during synchronization, as shown in Listing 7, which in turn enters the database semaphore already presented in Listing 1.

```
private static func assignments(
    in persistentContainer: NSPersistentContainer,
    completion: ((_ success: Bool) -> ())? = nil
) {
    persistentContainer.performBackgroundTask {
        context in
        AssignmentsSemaphore.enter {
            leave in
            // Execute critical, asynchonous operations here.
            Synchronize.assignments(in: context) {
                success in
                if let completion = completion {
                    completion(success)
                }

                // Leave the AssignmentsSemaphore.
                leave()
            }
        }
    }
}
```

Listing 7: SynchronizeAssignments.swift – entering the semaphore

In the next step, shown in Listing 8, a request to the backend API is made to receive all assignments. The Request.getAssignments(...) method works basically the same as the getQuestionnaire(...) method from Listing 6. Synchronize.assignments(with: assignments, in: context) updates the local database as shown in Figure 5.18, before calling assignmentsQuestionnaires(...).

```swift
internal static func assignments(
    in context: NSManagedObjectContext,
    completion: @escaping (_ success: Bool) -> ()
) {
    // Get all Assignments from the backend API.
    Request.getAssignments(success: {
        assignments in

        // Update database with assignments from backend.
        Synchronize.assignments(with: assignments, in: context) {
            // Synchronize Questionnaires.
            Synchronize.assignmentsQuestionnaires(in: context) {
                success in

                completion(success)
            }
        }
    }, error: {
        _,_ in

        completion(false)
    })
}
```

Listing 8: SynchronizeAssignments.swift – updating assignments in the local database

The last major step of the assignments synchronization process, is shown in Listing 9. First, all IDs are collected from questionnaires belonging to assignments in the local database for which there is no questionnaire relationship established. These IDs are then used to validate with the Synchronize.questionnaireTemplates(with: questionnaireIds, in: context) method whether the corresponding questionnaire templates exist in the local database or have to be loaded from the backend, in an extra step – not discussed here. If this method finishes, all assignments without questionnaire relationships are assigned newly created questionnaires, generated from the local templates with the method Questionnaire.fromTemplate(with: apiId, in: context).

```
private static func assignmentsQuestionnaires(
    in context: NSManagedObjectContext,
    completion: @escaping (_ success: Bool) -> Void
) {
    let assignmentsWithNoQuestionnaire =
  ↪ Assignment.allWithNoQuestionnaire(in: context)

    // Get all Questionnaire apiIds
    // of Assignments with no Questionnaire.
    let questionnaireIds = assignmentsWithNoQuestionnaire.map {
        // Get only the apiId.
        return $0.questionnaireId
    }

    // Get all the Questionnaire Templates
    // needed to present the Assignments.
    Synchronize.questionnaireTemplates(with: questionnaireIds,
  ↪ in: context) {
        for assignment in assignmentsWithNoQuestionnaire {
            // Get the apiId of the Questionnaire,
            // which should be added to the Assignment.
            let apiId = assignment.questionnaireId
            // Create and add a Questionnaire
            // from a Questionnaire Template.
            assignment.questionnaire =
  ↪ Questionnaire.fromTemplate(with: apiId, in: context)
        }
        Database.save(context)
        completion(true)
    }
}
```

Listing 9: SynchronizeAssignments.swift – loading, creating & assigning questionnaires

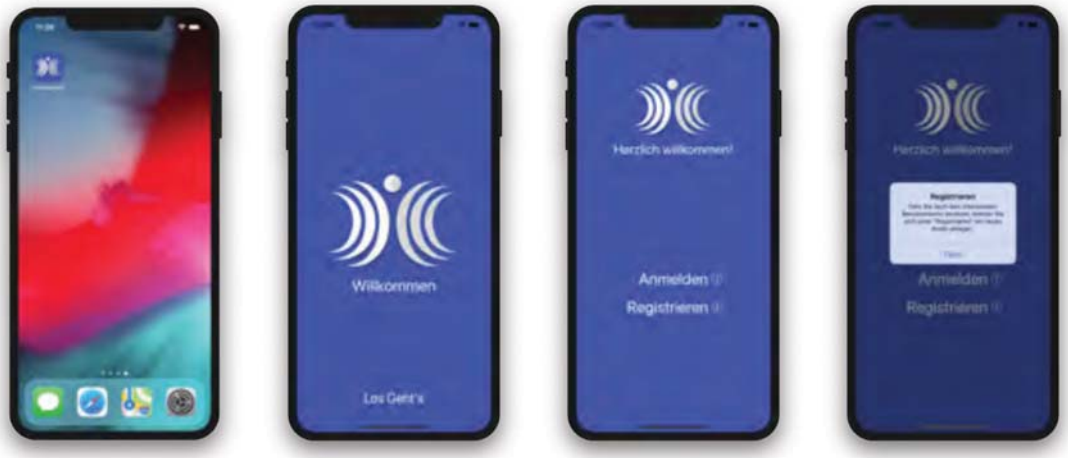
Based on a quite similar structure of individual synchronization steps, the contribution and further details of the questionnaire synchronization process, helper methods as well as database operations are not explained in greater extend at this point.

7 Presentation of the iOS Application

This chapter presents the finished iOS application in the form of a guided tour. Screenshots will showcase and explain all features step by step, also including optical differences between different iOS devices. Nevertheless, the functionality is the same on all devices. While screenshots embedded in this chapter were mostly taken on the iPhone Xs Max, those from the iPad Pro (12.9 inches) (3rd generation) and the iPhone 5s serve as a comparison.

7.1 First time usage

Figure 7.1 shows the views that are displayed to the user when the application is used for the first time. After launching the application from the iOS home screen (7.1a), the user is presented with a welcome view (7.1b). When proceeding to the first menu (7.1c), the user is offered the opportunity to log in with an existing account or to register a new one. Additional info buttons facilitate orientation (7.1d).



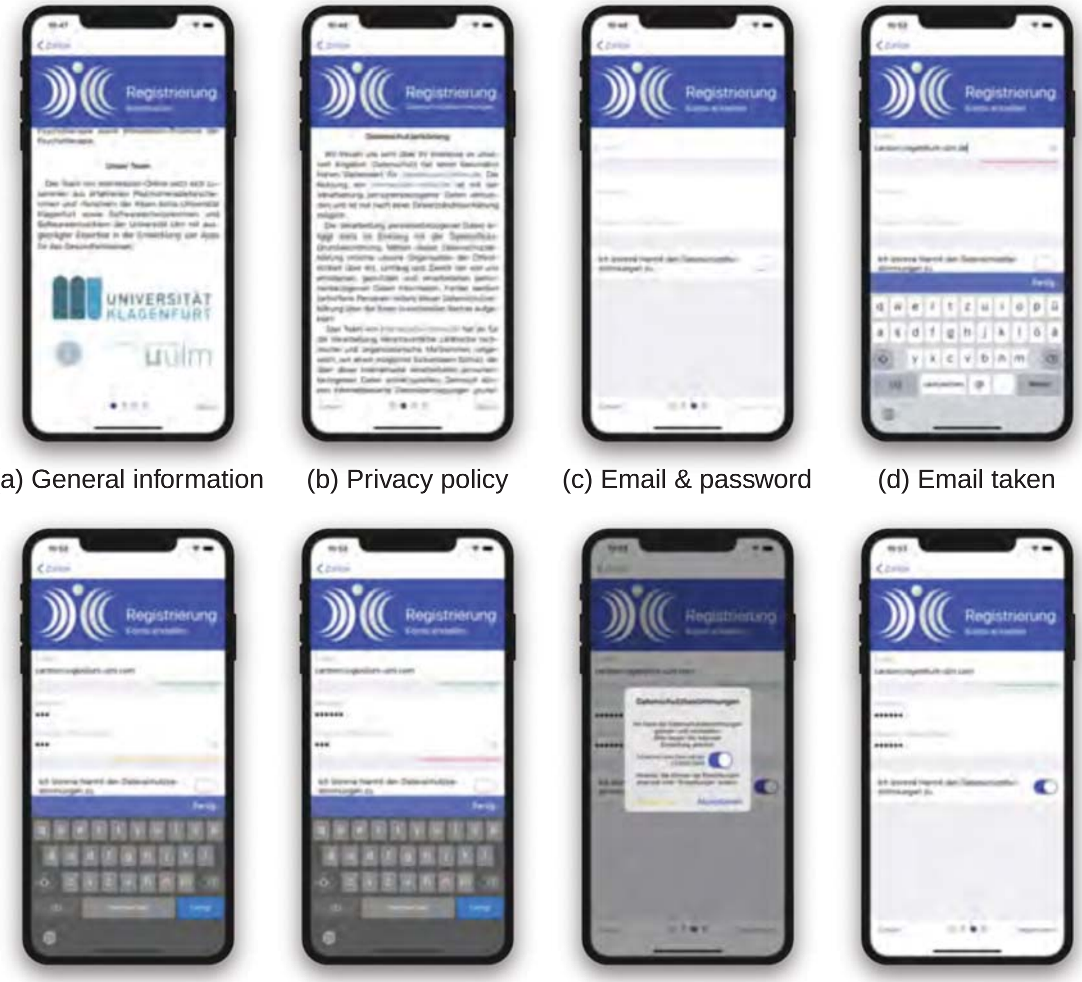
(a) iOS home screen (b) Welcome view (c) First menu (d) Info button information display

Figure 7.1: The first views of the application in portrait mode.

7.2 Registration

When registering a new account, the user is first presented some general information about the application context (7.2a) and then the privacy policy statement (7.2a). After that, an valid email address must be specified and a password set (7.2c). The user is informed if the email address has already been registered (7.2d) or if other entries are invalid (7.2e, 7.2f). Before the registration button can be clicked, it must be confirmed, that the privacy policy statement was read (7.2g). In the same view, the user has the possibility to not share his data with researchers. When all fields are filled in with valid information, the register button is now clickable (7.2h).

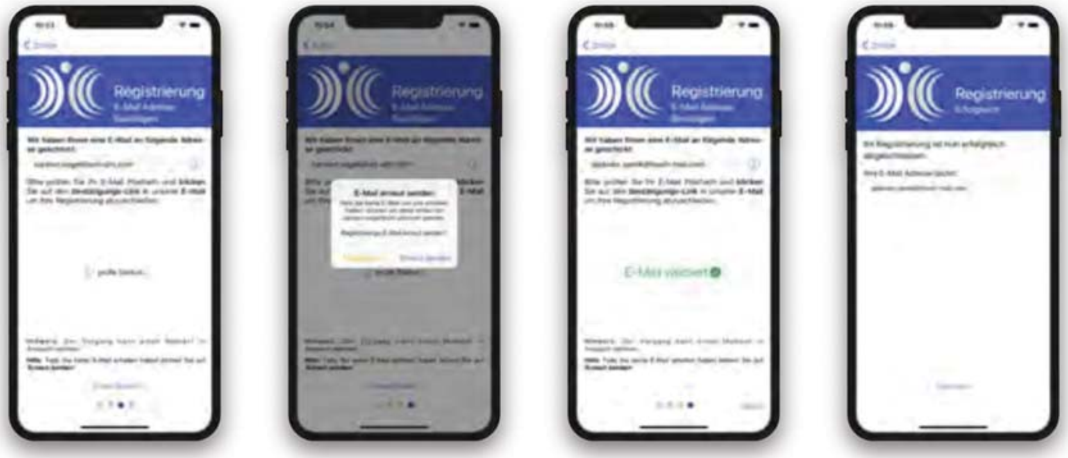
If the registration was successful, the user now must confirm the email address. The current state of the registration is always displayed to the user (7.3a). If there is a typo, the process can be aborted and a new email address can be specified. If the user does not receive a confirmation email, it can be requested again (7.3b). Once the email address has been confirmed, it will be displayed in the application and the user can complete the registration process (7.3c). Clicking the continue button shows the view displayed in 7.3d, where the user can directly jump to the login view, which is presented in Section 7.3.



(a) General information (b) Privacy policy (c) Email & password (d) Email taken

(e) Password too short (f) Passwords unequal (g) Privacy policy and researcher access confirmation (h) New account data valid & complete

Figure 7.2: Registration – creating a new user account



(a) Abort confirmation (b) Resend email (c) Email confirmed (d) Registered

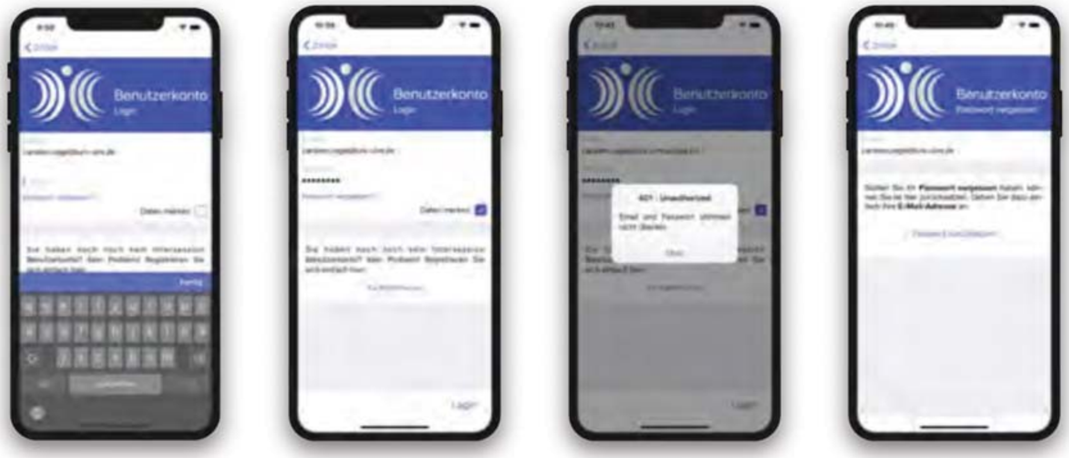
Figure 7.3: Registration – email confirmation

7.3 Login

Figure 7.4 shows the views related to the user login and the password reset. Over the login view (7.4a) user credentials can be entered as well as a check mark set to remember them. The credentials of a newly created user account are already filled in (7.4b), if this view was reached from the registration view directly. Clicking on the login button will either lead to a successful login or to an error message describing the problem, like displayed in Figure 7.4c, where the email and password combination is wrong.

If the user logs in with a new account, which is most likely not connected to a therapist, the pairing view (see Section 7.4) opens first. Otherwise the home screen (see Section 7.5) is shown.

If the user has forgotten the account password there is the option to reset it (7.4b). By clicking on the password forgotten button a view for the password reset is displayed. If the email address is registered in the system, a email with a link to reset the password is send to the corresponding user. Success or errors will be communicated to the user by appropriate alerts.

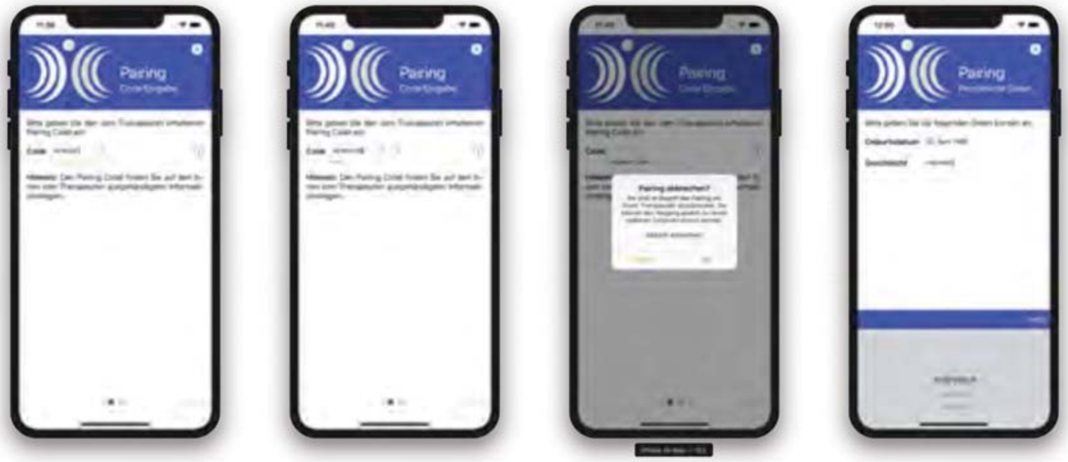


(a) Entering credentials (b) Login button (c) Login error (d) Password reset

Figure 7.4: Login

7.4 Pairing

The patient connects with the therapist during the pairing process, as shown in Figure 7.5. If a user logs in unpaired, the pairing view automatically opens and the user can enter the pairing code received in a PDF file (see Appendix A.2) from the therapist (7.5a). When a complete eight-digit code is entered, the application checks the code on the server (7.5b). The pairing process can be aborted any time (7.5c) and retrieved again, later, over the therapist view, described in Section 7.5. If the code is correct the user can proceed to enter age and gender on the next view (7.5d). Pressing a confirm button completes the process and, if successful, connects the user to the therapist as a patient.



(a) Entering the pairing code (b) Code verification (c) Abort pairing (d) Entering age and gender

Figure 7.5: Pairing

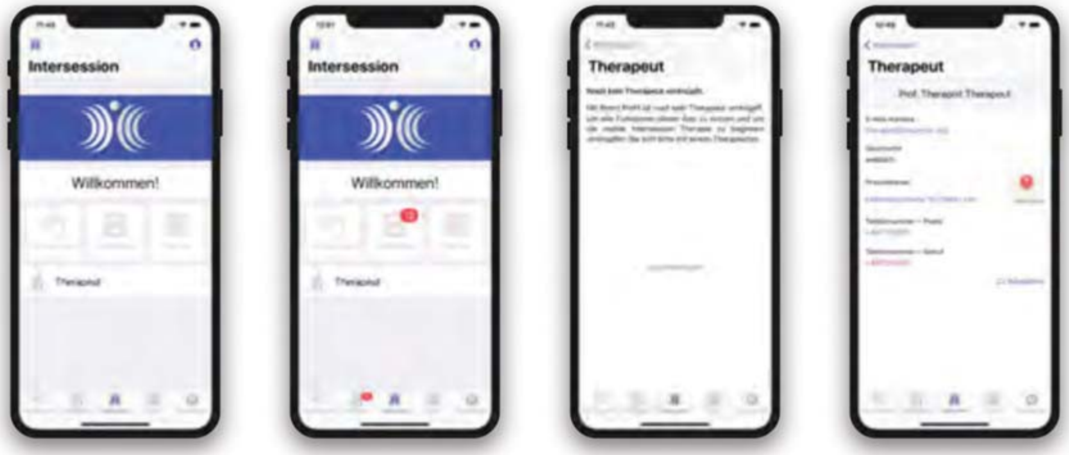
7.5 Home screen, unpaired and therapist view

After completing the pairing process or launching the application, while still being logged in, the patient first sees the main menu, with the home screen opened (7.6a). If already paired there may appear badges indicating how many interventions or questionnaires are currently available as well as appointments due (7.6b). The main menu bar contains five tabs. In addition to the home screen itself, the interventions, questionnaires, appointments and settings view can be accessed from here.

By clicking on the therapist entry on the home screen an user reaches either the unpaired view (7.6c) or the therapist view (7.6d), if already paired. By clicking on the pair button the pairing view (7.5a) opens and the user can proceed as described in Section 7.4.

In the therapist's view, the patient sees important information about the therapist. By simply clicking on one of the controls, it is possible to directly send a ready-made email to the therapist, to open the address directly in Apple Maps[1] or to make a call. The contact information can also be imported into the iOS contact list, with ease.

[1] https://www.apple.com/ios/maps/, accessed 09.07.2019



(a) Home screen (b) Badges (c) Unpaired view (d) Therapist view

Figure 7.6: Home screen, unpaired and therapist view

7.6 Profile

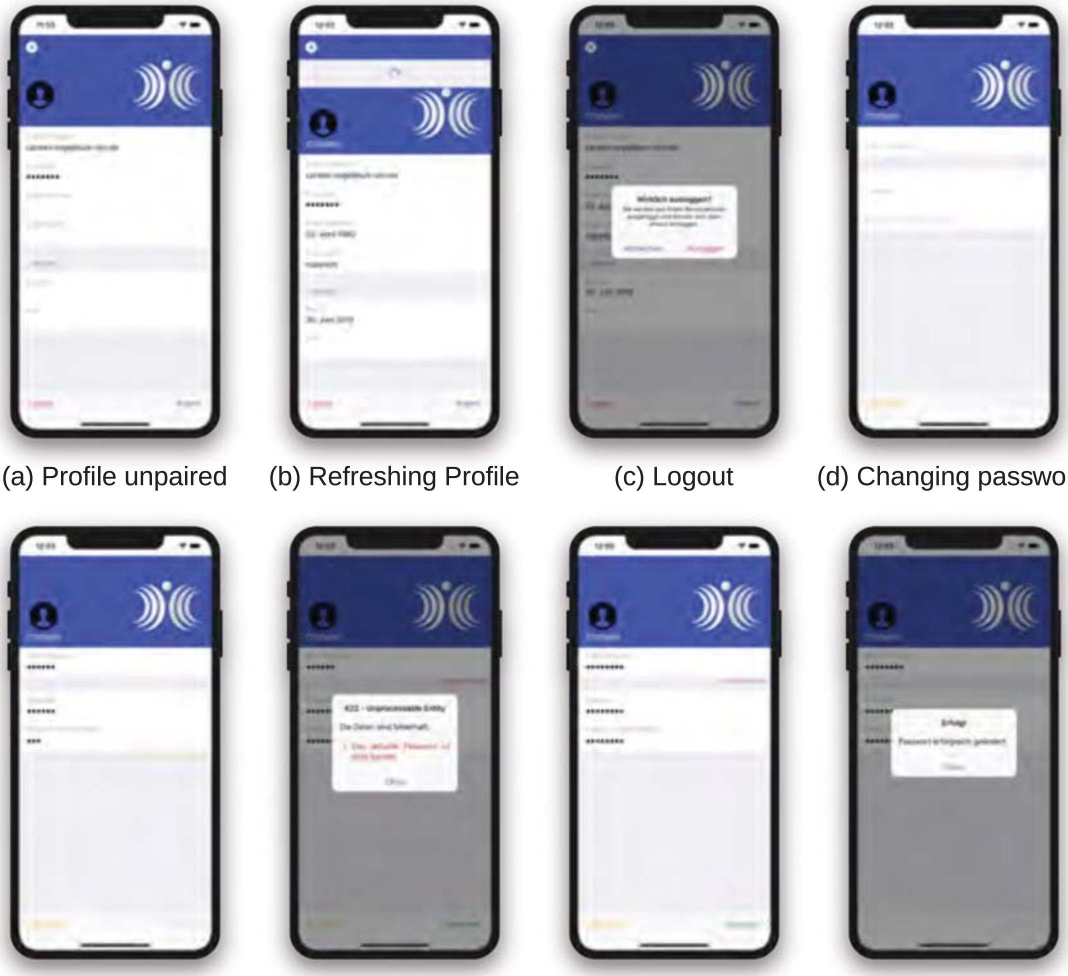
Figure 7.7 displays views related to the user profile.

Unpaired In the profile view, an unpaired user can only see the username and the masked password (7.7a).

Paired A paired user also sees personal information such as the user ID, date of birth and gender, and the beginning and end of the therapy (7.7b). By pulling down the profile view a spinning wheel appears and the data refreshes (7.7b).

Logout Via a logout button the user can log out of the account (7.7c).

Change password If the device is connected to the internet, the change button (7.7b), which opens a password form (7.7d), is clickable. To change the account password, the user is obligated to type in the current password as well as the new one twice, as confirmation. To guide the user, hints (7.7e), error alerts (7.7f) and warnings (7.7g) communicate information. If the password was successfully changed, this is also displayed to the user via an alert view (7.7h).


Wirklich ausloggen?

422 - Unprocessable Entity

Die Daten sind fehlerhaft.

Erfolg!

Passwort erfolgreich geändert

(a) Profile unpaired (b) Refreshing Profile (c) Logout (d) Changing password

(e) Password hint (f) Wrong password (g) Password warning (h) Password successfully changed

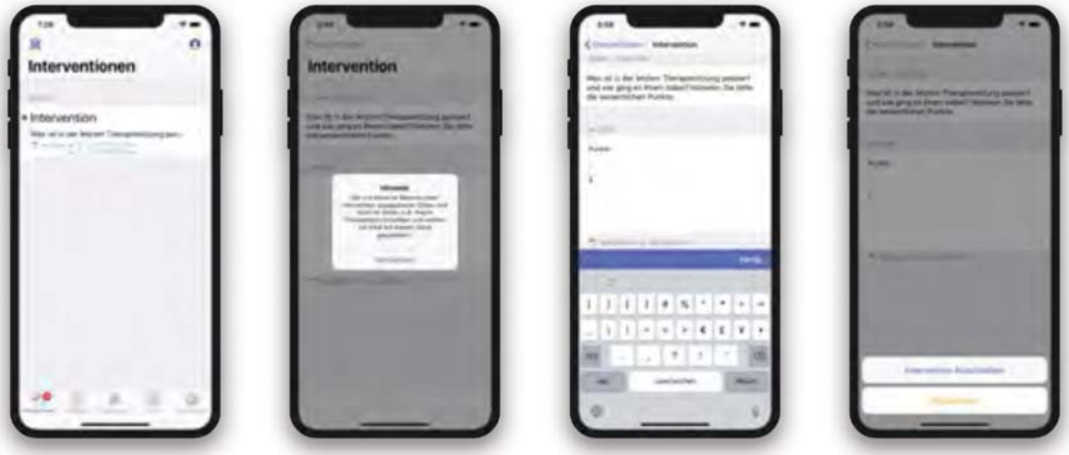
Figure 7.7: Profile

7.7 Interventions

This section looks at the interventions for patients, one of the three main functions of the application. Figures 7.8 and 7.9 visualize these.

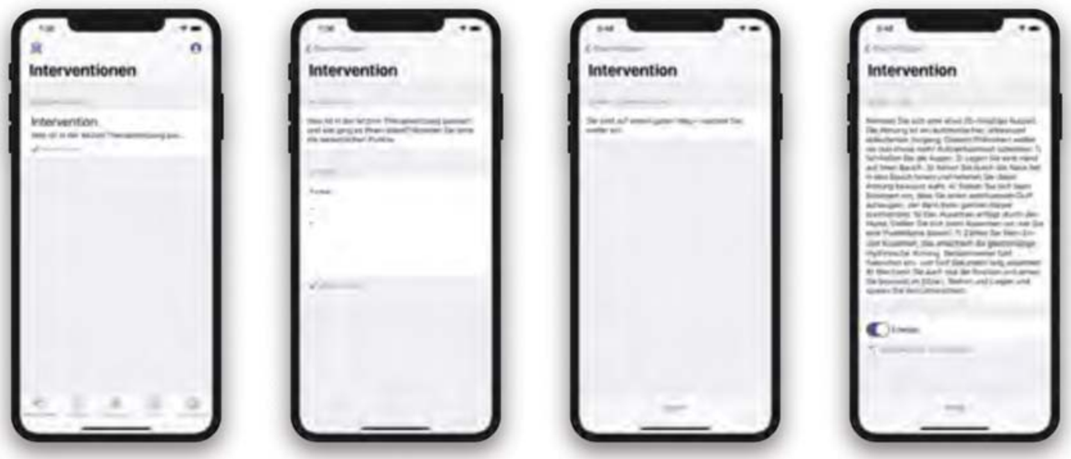
List view In a list view (7.8a) the patient sees his new interventions as well as all those that have already been answered. A round blue dot indicates whether an intervention has already been opened by the patient. Since interventions can only be stored locally, completed interventions are not synchronized between multiple devices, as explained in Section 4.2.6. However, the patient can still see if they have been read and answered. Pressing on one of the list entries opens the detailed view.

Detail view By opening the detail view of a new intervention an alert (7.8b) is shown, clarifying that the data is not shared with the therapist or the researchers. There are currently three different types of interventions. When editing a question intervention (7.8c), a text box must be completed before the intervention can be finished (7.8d). After finishing an intervention, it will be displayed in the list view (7.9a), where it can be opened again, but not edited any more (7.9b). A commendation intervention (7.9c) is just a motivational sentence for the patient, which does not require further editing. A task intervention (7.9d) can be marked as fulfilled before completion.



(a) List containing a new intervention (b) Privacy hint (c) Commenting (d) Confirming completion

Figure 7.8: Opening a new task intervention



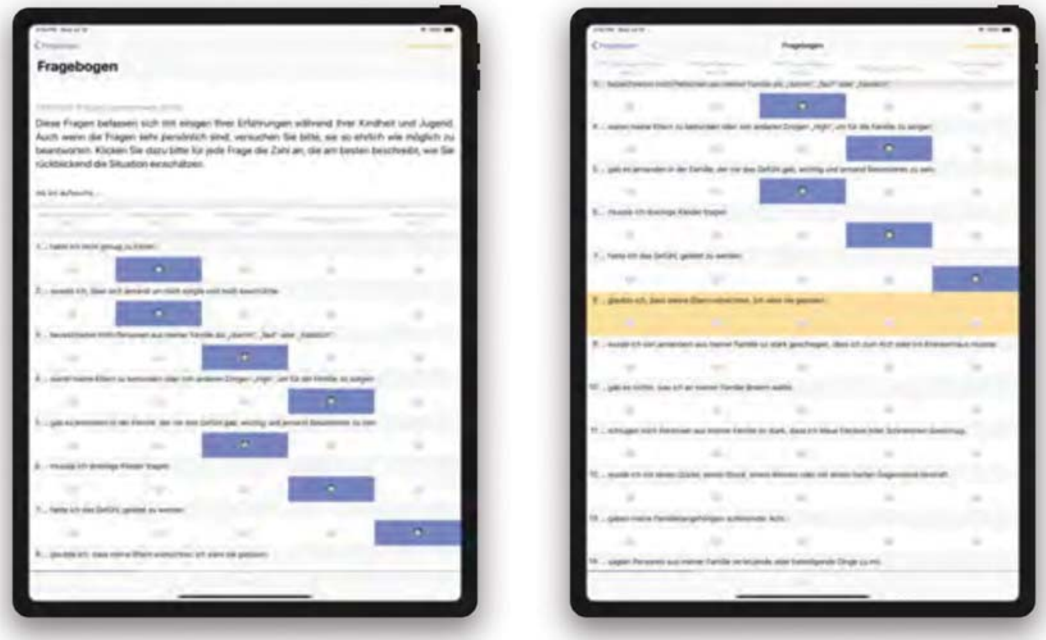
(a) List containing a completed intervention (b) Completed intervention (c) Commendation (d) Task

Figure 7.9: Completed and other types of interventions

7.8 Questionnaires

The second main function of the application is the completion of questionnaires. This is visualized in Figures 7.10, 7.11 and 7.12, for iPad and iPhone.

Detail view Questionnaires are filled in the detail view (7.10a). At the beginning of a questionnaire usually a short introductory text is to be seen. This is followed by one or more sections, each with its own text, a range of possible answers and a list of related questions, for each of which one of the answering options can be selected. An example of a questionnaire with multiple sections is shown in Figure 7.12a. Selecting an option is not permanent and can be changed by clicking on an other one or be reset by clicking again on the same. As with the interventions, any change is immediately persisted in the local database of the application. A button in the top right corner is clickable as long as there are unanswered questions left. Clicking this button will scroll to the first unanswered question in the current questionnaire and mark it yellow for a short period of time (7.10b). This is especially useful in large questionnaires. Note that the range of possible answers will stick to the top of a section to always be visible to the patient. When all questions have been answered, the progress bar at the bottom of the questionnaire disappears and the finish button becomes visible. An alarm prompts the patient to confirm the completion of the questionnaire (7.11a). Subsequently, the questionnaire will be transmitted and displayed in the overview as filled out accordingly (7.11b). There it can be viewed again, but can not be edited.



Fragebogen

Diese Fragen befassen sich mit einigen Ihrer Erfahrungen während Ihrer Kindheit und Jugend. Auch wenn die Fragen sehr persönlich sind, versuchen Sie bitte, sie so ehrlich wie möglich zu beantworten. Klicken Sie dazu bitte für jede Frage die Zahl an, die am besten beschreibt, wie Sie rückblickend die Situation einschätzen.

(a) Filling out a questionnaire

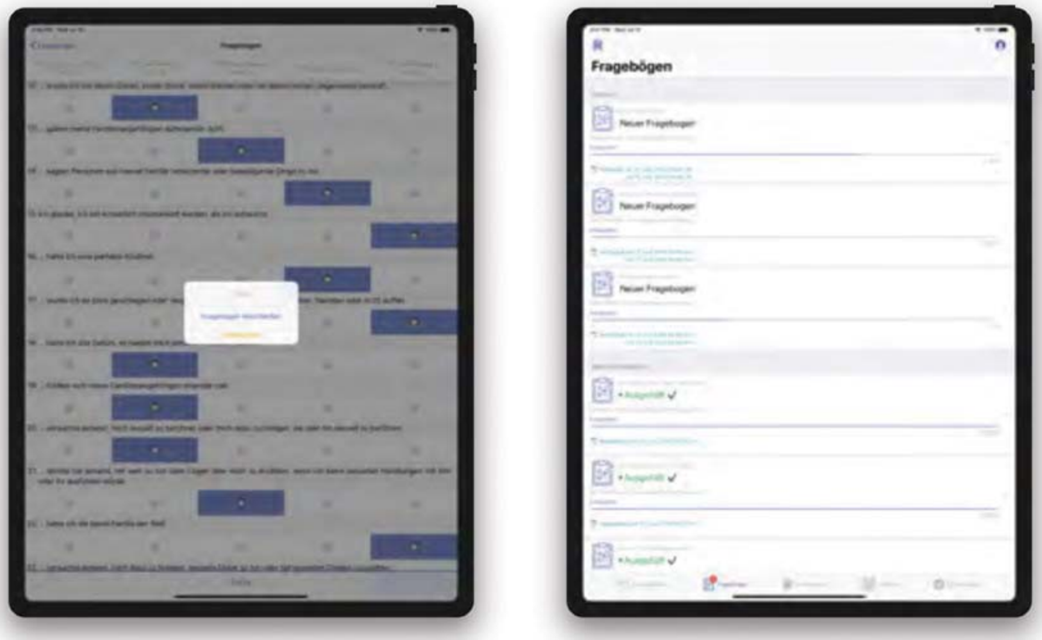
(b) Finding unanswered questions

Figure 7.10: Filling out questionnaires on the iPad

List view In a list view (7.11b), new and already submitted questionnaires are displayed to the patient. Here questionnaires can be selected and displayed in a detail view.

Daily questionnaire A special case for a questionnaire is represented by the Daily. This questionnaire has initially only one question with “yes” and “no” as possible answers (7.12b). When selecting “no” the questionnaire is complete and can be finished (7.12c). Selecting “yes” will instead open an additional section (7.12d). This new section does have an own range of possible answers. Here, it should be noted that the progress bar adjusts to the newly emerged questions accordingly.

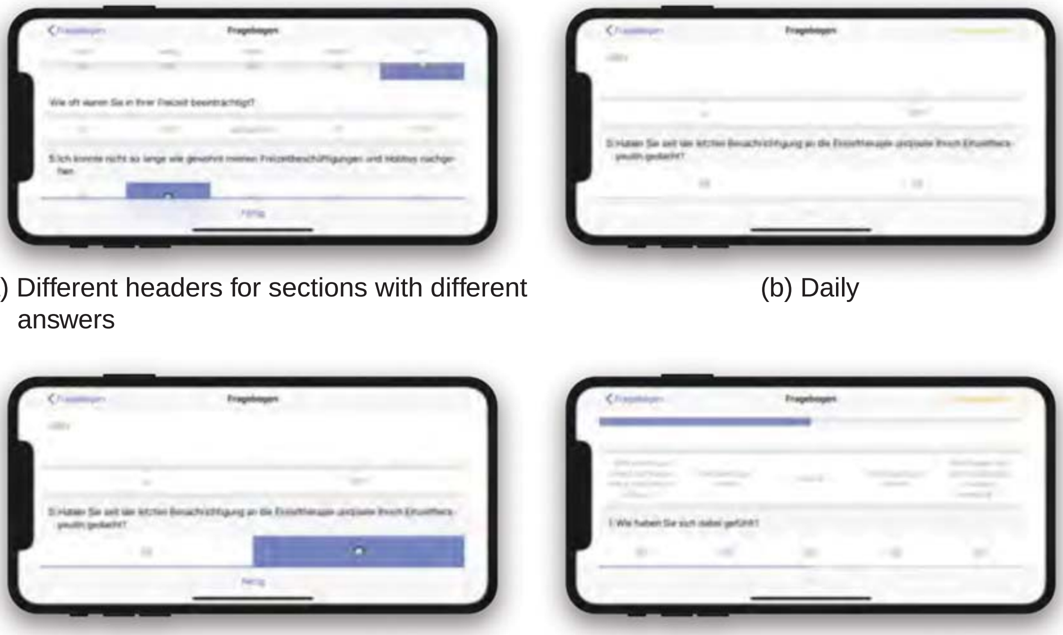
)



(a) Completing a questionnaire

(b) Questionnaire overview

Figure 7.11: Completing questionnaires and questionnaire overview on the iPad



(a) Different headers for sections with different answers

(b) Daily

(c) Daily: “No” answer

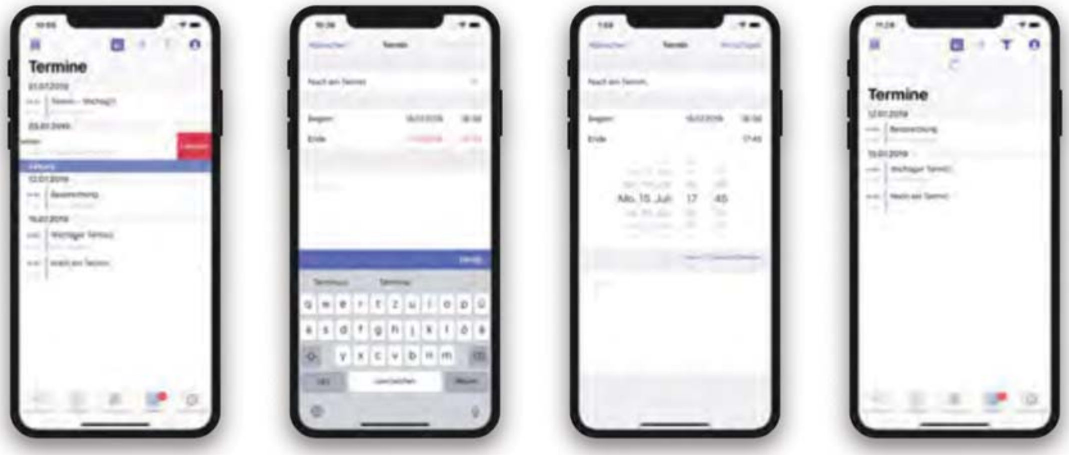
(d) Daily: “Yes” answer

Figure 7.12: Questionnaires – multiple sections and the Daily questionnaire

7.9 Appointments

The third main functionality of the application is the appointments and calendar view, where appointments can be managed, as seen in Figures 7.13 and 7.14.

List view All appointments are displayed in a list view (7.13a), with a blue-violet bar separating the old from the new ones. By default, past events are hidden by a set filter (7.13d). From this view the filter can be set and the calendar and create view reached. Clicking on an entry in this list opens the detail view for the corresponding appointment. By swiping from right to left the patient is asked whether or not to delete the appointment (7.13a). Both opening and deleting can also be done via a Force Touch menu. The list can be refreshed by dragging down(7.13d).



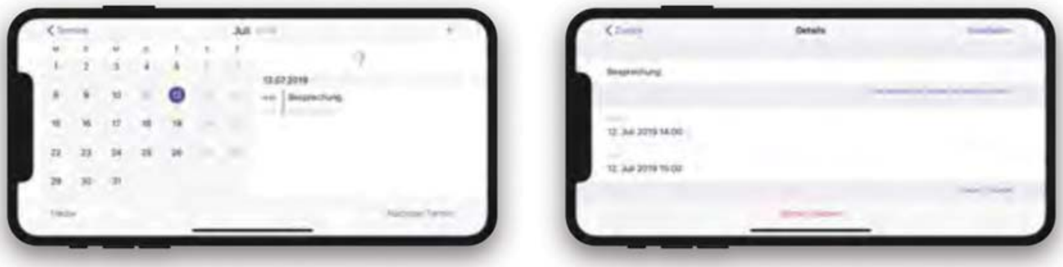
(a) Complete list (b) New appointment with warning (c) New appointment complete (d) Filtered list

Figure 7.13: Appointments

Create view Over the create view new appointments can be added (7.13b). The patient must set a title, start and end date. Faulty time configurations are coloured red and must be fixed to enable the add button. By tapping on the start or end date a date picker appears, animated (7.13c). As soon as the dates are correctly entered, the duration is displayed. Optionally notes can be added. By clicking the add button the appointment is created, synchronized and the patient can see it in the list (7.13d) or calendar view.

Calendar view The calendar view (7.14a) consists of a month view and a list of appointments. The list contains all appointments to the selected date in the month view. Additionally to the add button, there are also buttons to jump to the current date and the date of the next appointment. The month view can be scrolled vertically via a corresponding swipe gesture. The current date is marked red.

Detail view The detail view (7.14b) displays all information of an appointment. From here the update view can be reached or the delete button pressed.



(a) Calendar split view

(b) Appointment detail view

Figure 7.14: Calendar – horizontal orientation

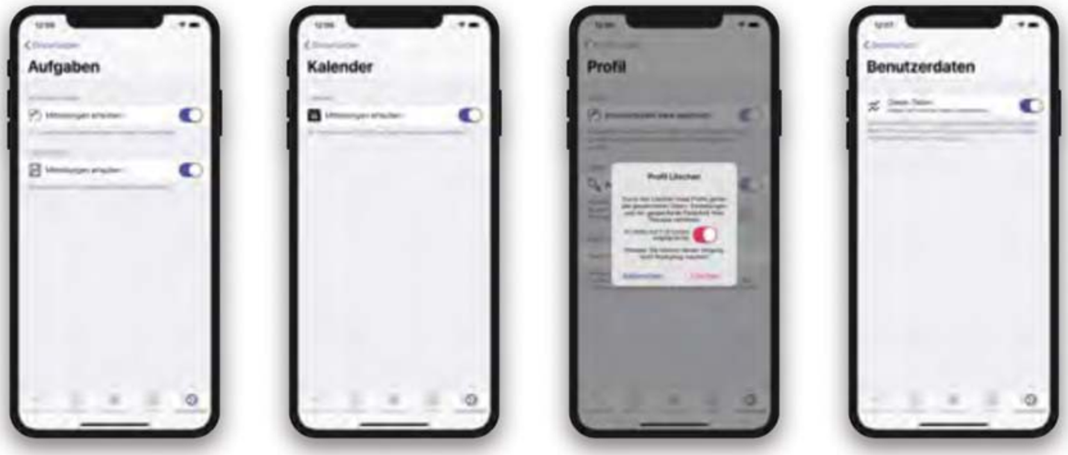
Update view The update view is basically the same as the create view except the update button. Appointments can be altered here.

7.10 Settings

Over the settings tab patients can customize the notifications and privacy settings as well as gather information about the project. This is shown in Figures 7.15 and 7.16.

Notifications For interventions, questionnaires and appointments, the notifications can be activated separately (7.15a, 7.15b).

Profile Over the profile settings (7.15c) it is possible to configure whether or not the interventions and user credentials are kept locally after logging out. With alerts, the user is confronted with the consequences of these actions. Also the patients profile, including all corresponding data sets, can be permanently deleted from the overall system. This



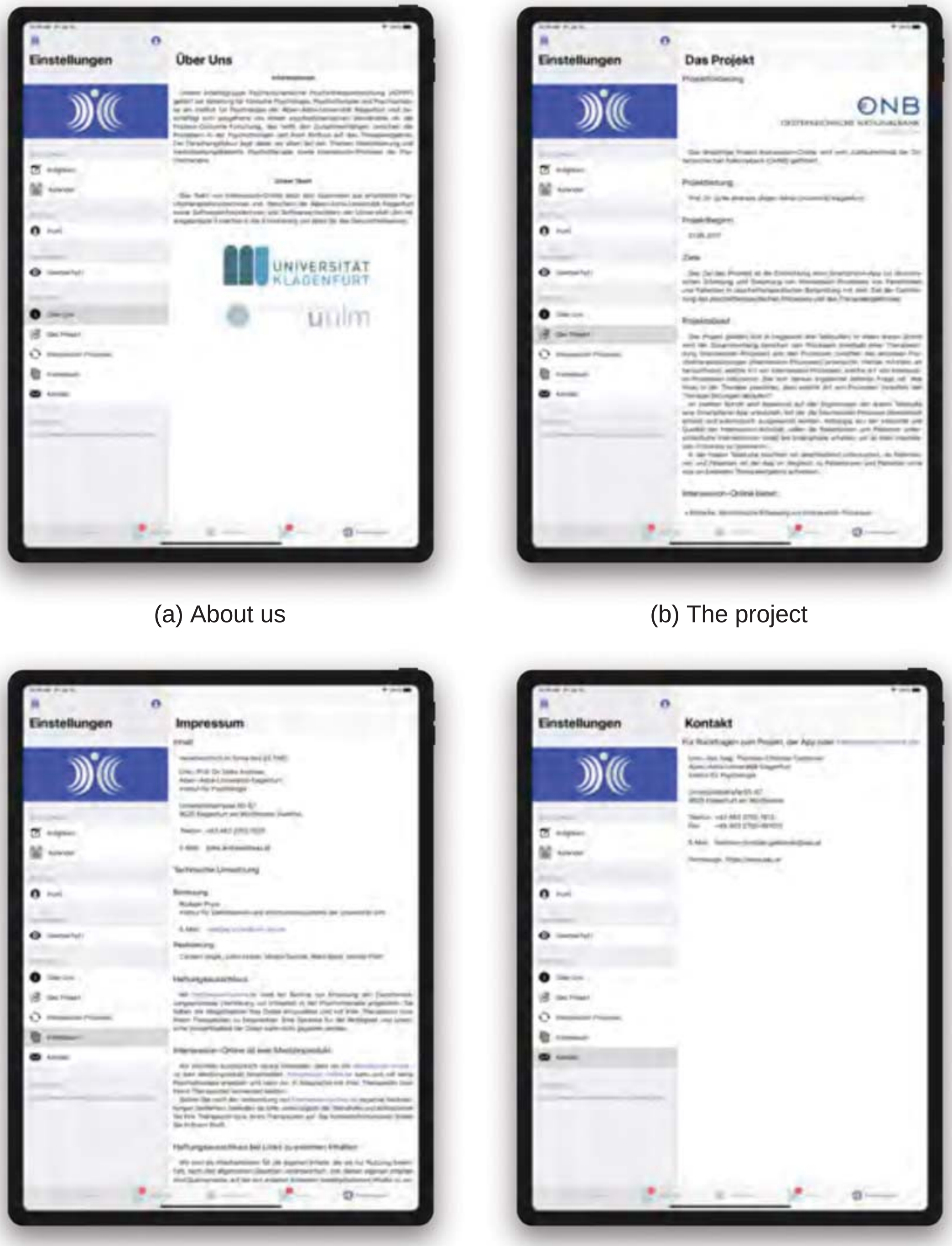
(a) Tasks (b) Calendar (c) Profile (d) Privacy

Figure 7.15: Settings

process is irreversible and must therefore be confirmed by activating a switch, clicking on the delete option and confirming again.

Privacy Over the privacy view the patient can access views of the data privacy statement and the corresponding settings (7.15d). An active internet connection is needed to alter this setting. Therefore the GUI visualizes changes with a spinning wheel or a disabled button, if offline.

Information In addition to the data privacy statement, there are also the categories “About Us” (7.16a), “The Project” (7.16b), “Intersession Processes”, “Imprint” (7.16c) and “Contact” (7.16d), giving the patient all needed information.



(a) About us

(b) The project

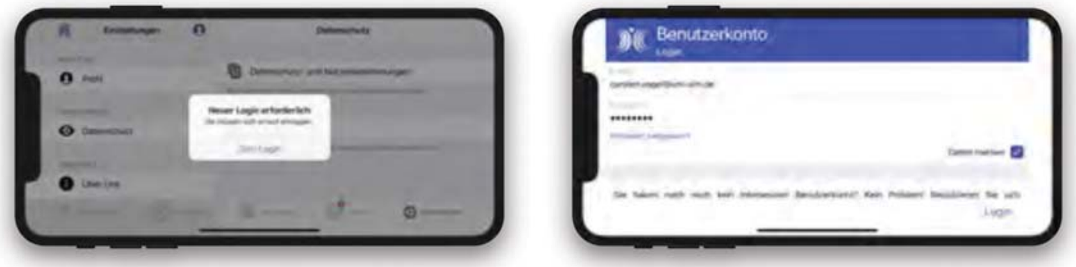
(c) Imprint

(d) Contact

Figure 7.16: Settings - iPad

7.11 Error messages

In rare cases, error messages may occur, for example if the server is not reachable or, as shown in Figure 7.17, the session token has expired (7.17a) and the user must log in again (7.17b).



Neuer Login erforderlich

Sie haben noch kein Benutzerkonto? Kein Problem! Registrieren Sie sich

(a) New login required

(b) Re-entering the user credentials

Figure 7.17: Error messages – landscape orientation

7.12 Inputs and feedback

All input fields are limited to meaningful user inputs and are checked strictly. By clicking on a input field bound to a keyboard input, when appearing, the displayed content is moved properly, to not cover up the input field. If an input field has been completely filled in, it is possible to jump to the next input field with the "Next" key or finish the input with "Done". Most input fields are fully customized with moving placeholder text and proper hints, warnings and error messages, directly below the input field.

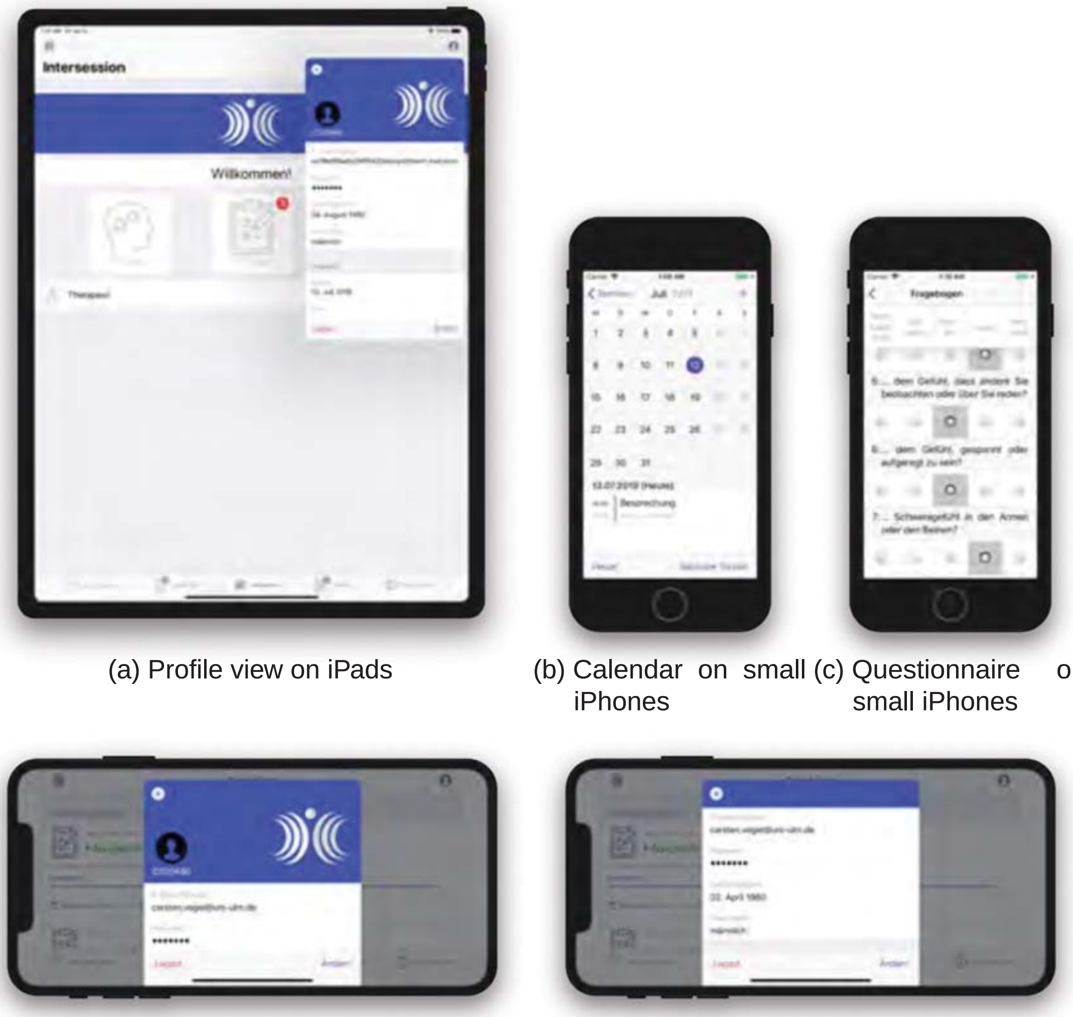
In many places, whenever useful, Force Touch gestures are available as an alternative way of interacting, to preview the next view and to display a corresponding option menu.

Progress bars and activity views keep the user aware of long running processes and background synchronization. Important hints, warnings and errors are presented to the user as alerts, interrupting the workflow and thus drawing the full attention the user.

7.13 Further impressions

This chapter finishes with a few more impressions from different screen sizes (iPhone 5s, iPhone XS Max and iPad Pro (12.9-inch) (3rd generation)), shown in Figure 7.18.

The profile hovers over whichever view it was opened from (7.18a). Even iPhones with small screens can properly display the calendar (7.18b) and questionnaires (7.18c) Widescreen iPhones have a special representation of the profile view (7.18d, 7.18e).



(a) Profile view on iPads

(b) Calendar on small iPhones

(c) Questionnaire on small iPhones

(d) Profile on wide iPhones in landscape mode

(e) Profile on wide iPhones in landscape mode – scrolled

Figure 7.18: Further impressions of the iOS application

8 Requirements Comparison

This chapter compares the developed application with the requirements defined in chapter 4. The assessment is based on objective comparisons between the implemented application, presented in Chapter 7, and the requirements as well as the customers assessment and user feedback from the test phase. The following distribution is used for a critical evaluation of the requirements:

- 100% The requirement is completely fulfilled.
- 90–99% The requirement is met, but there is room for improvement.
- 75–90% The requirement is largely met, but there are missing parts, that restrict the use or success marginally.
- 1–75% The requirement is at best partially fulfilled. The result has strong shortcomings and can not be used as intended.
- 0% The requirement was not fulfilled.

8.1 Functional requirements

Table 8.1 provides an overview of the fulfilment levels of the functional requirements.

First of all, it can be seen that all functions relating to the patient account and the information presentation, see 4.2.1 to 4.2.4 and 4.2.9, as well as the therapeutic functions of the application, see 4.2.5 to 4.2.7, are completely met.

Notifications offer all desired features, as described in Section 4.2.8, but, during the test phase, sporadically did not show up. This common issue in iOS development is further investigated and fixed in the the scope of this thesis.

Data synchronization, specified in Section 4.2.10, works flawlessly, reliably and can easily handle a variety of network and inconsistency problems. It is noteworthy that the number of requests to determine changes on the server could be improved via HTTP ETags[1], which are currently not provided by the server. Also related to the server, the usage of multiple devices does not include the locally generated Dailies. They will simply appear at different times per device and treated as independent questionnaires. Important here is that this is not a big problem, but in the worst case a little inconvenience.

The last functional requirement concerns the handling of error messages, warnings and notes, as defined in section 4.2.11. This requirement has been fully and correctly implemented, only in some places the frequency and meaningfulness of the texts could be slightly improved.

Ref.	Requirement group	Requirement	fulfilled
4.2.1	Manage patient account	Account creation	100%
		Email validation	100%
		Login	100%
		Logout	100%
		Password reset	100%
		Profile view	100%
4.2.2	Pairing		100%
4.2.3	Overview		100%
4.2.4	Therapist profile		100%
4.2.5	Handling questionnaires	Questionnaire overview	100%
		Questionnaires completion	100%
		Questionnaire structure	100%
		Questionnaire generation	100%

[1] Hypertext Transfer Protocol entity tag `https://tools.ietf.org/html/rfc7234`

Ref.	Requirement group	Requirement	fulfilled
4.2.6	Handling interventions	Intervention overview	100%
		Intervention reception and completion	100%
		Intervention structure	100%
4.2.7	Setting appointments	Appointment entry	100%
		Appointment list	100%
		Calendar view	100%
4.2.8	Notifications		90%
4.2.9	Information, settings and other options	Notification settings	100%
		Profile settings	100%
		Data privacy statement	100%
		Data privacy	100%
		General information	100%
		Imprint and contact	100%
4.2.10	Data synchronization and offline operation	Data synchronization	95%
		Offline operation	100%
		Multiple device usage	95%
4.2.11	Error messages, warnings and hints		90%

Table 8.1: Functional requirements fulfilment

8.2 Non-functional requirements

The non-functional requirements of Section 4.3 have all been fully met with two small exceptions, which are discussed below. Worth mentioning at this point is the strong positive feedback of the test group and the customer on the look and feel of the application.

The performance, see Section 4.3.3, may still be slightly improved, concerning the synchronization rate in the background thread. Overall performance improvement can be achieved through original Apple push notifications. However, a corresponding infrastructure was not available at the time of implementation and is not yet planned.

The application consists of several interchangeable modules, which encapsulate the individual functions of a feature and communicate with each other through interfaces. Whenever possible, functions were reused to avoid overhead. All features and their individual classes follow the same structure, which makes legibility much easier. Only a few of the classes, implemented at the beginning of the project, have become more unstructured, driven by the need for rapid progress for demonstration purposes and frequent adjustments to new circumstances. So, for these specific classes, there is room for improvement in terms of modularity, structuring and maintainability. In general, the code is well-structured, reusable, and uses common patterns for software development. It thus largely fulfils the requirements of Section 4.3.7.

Ref.	Requirement group	Requirement	fulfilled
4.3.1	Look and feel	Intuitive controls	100%
		Appealing design	100%
4.3.2	Platform		100%
4.3.3	Performance		95%
4.3.4	Economical resource consumption		100%
4.3.5	Reliability, robustness and functional safety		100%
4.3.6	Information security		100%
4.3.7	Modularity, exchangeability, extensibility and maintainability		85%
4.3.8	User guidance		100%

Table 8.2: Non-functional requirements fulfilment

8.3 Boundary conditions

In this area, too, all requirements were met as far as possible. The details are discussed below.

The most important boundary condition was described in Section 4.4.1 and is completely fulfilled. The application is available for everyone over the Apple App Store. Users can create an account and play around with some features, but cannot use any therapeutic functionalities without being paired with an therapist – just as intended.

The application was completed and released for internal beta-testing in time, as required in Section 4.4.2, and is available in the Apple App Store since mid may 2019. Nevertheless, bug fixes had to be added later to the notification functionality, expanding the development time over the time frame.

The organization and documentation, as required in Section 4.4.3 and 4.4.4, was done mostly in a very structured and comprehensibly way. All important decisions were noted down in Markdown and, for crucial components, summarized in special definition files. These Markdown-files can directly be imported to the online tool Confluence.

Ref.	Requirement group	Requirement	fulfilled
4.4.1	Medical applications		100%
4.4.2	Timeframe and procedure		85%
4.4.3	Organization		95%
4.4.4	Documentation		95%

Table 8.3: Boundary conditions fulfilment

9 Conclusion

Finally, in this chapter a conclusion is drawn from this master thesis, the contents are roughly summarized and a brief outlook on opportunities for improvement and future projects related to this work is given.

The work represents, as an interdisciplinary project between the fields of psychology and computer science, a success that can be attributed to the dedication and good cooperation of the persons involved. An overall system, for which this work represents an essential component, was realized for the acquisition and exploration of intersession processes. This work has resulted in many valuable experiences, from planning and collaborating with experts in another discipline, over the design and implementation phase, to the publication of the application in the Apple App Store.

9.1 Summary

Essentially, the first three chapters deal with the scientific foundations of psychology and computer science needed for this work. Chapter 1 first motivated the topic before gradually introducing problem statement. It addressed the research topic of intersession processes and described the digital platform that needs to be developed in this context. Furthermore, the role of this work for the overall system was defined and a brief overview of the chapters given. Subsequently, in Chapter 2 a deeper introduction to the core research topic was given, the basics explained and the current state of research in this field presented. A study outside the scope of this work was described, in which the iOS application to be developed is to be used. At the end of the first part of the thesis, in Chapter 3 related work concerning eHealth and digital data collection was discussed.

The middle part of the work addresses the conception and realization of the mobile application. Chapter 4 defines the functional and non-functional requirements, resulting from previous chapters, as well as the boundary conditions. Based on this, in chapter 5 a design of the mobile application was conceptualized, where important decisions of the design, the user interface as well as the architecture were discussed. Chapter 6 first dealt with technical details and then with the actual implementation of the application.

The last part of the thesis dealt with the results of the implementation. In chapter 7 the finished iOS application was presented in the form of a guided tour, before being evaluated in chapter 8.

9.2 Outlook

Finally, there are some ideas as to what can be improved in this project and what future projects could result from it.

9.2.1 Improvement opportunities

First of all, the requirements that are not 100% fulfilled can be worked through, as there is still a slight potential for improvement left. For example, it would be possible to go through all the error messages, warnings, and hints that are displayed across the applications and adjust them according to their quality. Here, key features include their triggers, the quality and accuracy of the text, and the frequency with which the messages occur. These may be annoying in some situations or may be missing elsewhere. Another possibility would be to improve the performance, here the algorithm for the background update of the application could be refined or even user settings for the frequency could be implemented. Also the classes mentioned in chapter 8.2 could be refactored to make the code clearer and more modular.

A server-side improvement would be server-generated dailies. For this purpose, only the local generation would have to be switched off in the code of the iOS application. Two big improvements would be the usage of eTags for less synchronization overhead and Apple Push Notifications, both associated with greater effort and in conjunction with implementation changes to the server. Another idea is to enable the server to save personal intervention data of the patient, without the therapists or the researches to have access. The advantage would be that locally stored data could be synchronized with other patient devices or restored in the event of a local loss. For this, the data would have to be encrypted locally on the device with the password of the patient and stored in this form on the server.

9.2.2 Opportunities for follow-up projects

In a follow-up project, in addition to the improvements described in Section 9.2.1, the following ideas could be realized.

Multilingualism In addition to the already implemented German language, other languages could be added. In fact the settings tab of the application is already partially implemented bilingual. However, full translation into other languages must also include a mechanism (such as sending a language parameter in the HTTP request) to retrieve questionnaires on the server.

Questionnaire module Also, the expansion of the questionnaire module would be conceivable. Here, the questionnaires could be supplemented by new types of questions or even inputs via peripheral devices. In addition, the basis for dependent questions, during the completion, has already been created. This too could be a starting point.

Other new functionalities In addition, the front camera could be utilized to recognize, by means of machine learning, emotions when filling in the questionnaires. Even at the meta level, more information about using the application or individual functions and the usage time can be collected. Comparisons of user behaviour and the user's mood between Android and iOS users, which is not uncommon for field studies, could be accomplished.

Areas of application It would be conceivable to adapt the entire system as part of further studies or even to transfer it to a new area of application.

Bibliography

[1] Statista; Statista DMO: Number of smartphone users in germany from 2015 to 2022 (in millions)*. Website (July 2017) Statistica, https://www.statista.com/statistics/467170/forecast-of-smartphone-users-in-germany/ [Online; accessed 06.06.2019].

[2] eMarketer: Number of smartphone users worldwide from 2014 to 2020 (in billions). Website (June 2016) Statistica, https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ [Online; accessed 06.06.2019].

[3] Morris, M.E., Aguilera, A.: Mobile, social, and wearable computing and the evolution of psychological practice. Professional Psychology: Research and Practice 43 (2012) 622

[4] Pryss, R., Reichert, M., Herrmann, J., Langguth, B., Schlee, W.: Mobile crowd sensing in clinical and psychological trials–a case study. In: 2015 IEEE 28th International Symposium on Computer-Based Medical Systems, IEEE (2015) 23–24

[5] Statista: Total global mhealth market forecast from 2016 to 2025 (in billion u.s. dollars). Website (June 2018) Statistica, https://www.statista.com/statistics/938544/mhealth-market-size-forecast-globally/ [Online; accessed 07.06.2019].

[6] research2guidance: Number of mhealth app downloads worldwide from 2013 to 2017 (in billions). Website (November 2017) Statistica, https://www.statista.com/statistics/625034/mobile-health-app-downloads/ [Online; accessed 07.06.2019].

[7] research2guidance: Most attractive healthcare sectors for mhealth app companies as of 2017). Website (November 2017) Statistica, https://www.statista.com/statistics/795584/top-healthcare-sectors-for-mhealth-app-companies-2017/ [Online; accessed 07.06.2019].

[8] Zeeck, A., Hartmann, A., Orlinsky, D.: Inter-session-prozesse. PPmP-Psychotherapie· Psychosomatik· Medizinische Psychologie 54 (2004) 236–242

[9] Andreas, S., Gablonski, T.C., Hiesberger, S., Senft, B., Koller, I., Schulz, H.: Intersession-processes in inpatient therapy: Are the individual or group therapy important for the therapeutic relationship and the outcome? PSYCHODYNAMISCHE PSYCHOTHERAPIE 15 (2016) 206–218

[10] Gablonski, T.C.: Intersession-online: Entwicklung einer smartphone-app zur systematischen erfassung und kontrolle von intersession-prozessen. PhD- exposé 15 (2019) 1–5

[11] Masse, M.: REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. " O’Reilly Media, Inc." (2011)

[12] Plante, T.G.: Contemporary clinical psychology. John Wiley & Sons (2010)

[13] Brain, C.: Advanced psychology: applications, issues and perspectives. Nelson Thornes (2001)

[14] Hartmann, A., Orlinsky, D.E., Geller, J.D., Zeeck, A.: Der inter-session-fragebogen (isf)–ein instrument zur erfassung von psychotherapierelevanten prozessen zwischen sitzungen. PPmP-Psychotherapie· Psychosomatik· Medizinische Psychologie 53 (2003) T67–T75

[15] Orlinsky, D.E., Geller, J.D.: Patients’ representations of their therapists and therapy: New measures. (1993)

[16] Hartmann, A., Orlinsky, D., Zeeck, A.: The structure of intersession experience in psychotherapy and its relation to the therapeutic alliance. Journal of clinical psychology 67 (2011) 1044–1063

[17] Orlinsky, D.E., Howard, K.I.: Process and outcome in psychotherapy. In: Garfield, S.L., Bergin, A.E. (Hg.): Handbook of psychotherapy and behavior change. 3. Auflage, S. 311–381, New York: Wiley (1986)

[18] Stewart, S., Schröder, T.: Emotional homework: A systematic literature review of patients' intersession experiences. Journal of Psychotherapy Integration 25 (2015) 236

[19] Orlinsky, D., Tarragona, M.: Patients' representations of the therapist and experiences in therapy sessions: Further findings. In: annual conference, Society for Psychotherapy Research, Toronto. (1989)

[20] Geller, J., Farber, B.: Factors influencing the process of internalization in psychotherapy. Psychotherapy Research 3 (1993) 166–180

[21] Geller, J.D., Cooley, R.S., Hartley, D.: Images of the psychotherapist: A theoretical and methodological perspective. Imagination, Cognition and Personality 1 (1981) 123–146

[22] Epstein, M.E.: Mental representations of the psychotherapeutic relationship during the post-termination period. PhD thesis, University of Chicago, School of Social Service Administration (1989)

[23] Owen, J., Quirk, K., Hilsenroth, M.J., Rodolfa, E.: Working through: In-session processes that promote between-session thoughts and activities. Journal of Counseling Psychology 59 (2012) 161

[24] Knox, S., Goldberg, J.L., Woodhouse, S.S., Hill, C.E.: Clients' internal representations of their therapists. Journal of Counseling Psychology 46 (1999) 244

[25] Zeeck, A., Hartmann, A.: Relating therapeutic process to outcome: are there predictors for the short-term course in anorexic patients? European Eating Disorders Review: The Professional Journal of the Eating Disorders Association 13 (2005) 245–254

[26] Hartmann, A., Orlinsky, D., Weber, S., Sandholz, A., Zeeck, A.: Session and intersession experience related to treatment outcome in bulimia nervosa. Psychotherapy: Theory, Research, Practice, Training 47 (2010) 355

[27] Gablonski, T.C., Senft, B., Andreas, S.: The relationship between intersession processes and level of personality functioning in patients with mental disorders. (under review)

[28] Franke, G.H., Derogatis, L.R.: SCL-90-R: symptom-Checkliste von LR Derogatis: deutsche version: manual. Beltz Test (2002)

[29] Wilmers, F., Munder, T., Leonhart, R., Herzog, T., Plassmann, R., Barth, J., Linster, H.W.: Die deutschsprachige version des working alliance inventory–short revised (wai-sr)–ein schulenübergreifendes, ökonomisches und empirisch validiertes instrument zur erfassung der therapeutischen allianz. Klinische Diagnostik und Evaluation 1 (2008) 343–358

[30] Oh, H., Rizo, C., Enkin, M., Jadad, A.: What is ehealth (3): a systematic review of published definitions. Journal of medical Internet research 7 (2005) e1

[31] Eysenbach, G.: What is e-health? Journal of medical Internet research 3 (2001) e20

[32] East, M.L., Havard, B.C.: Mental health mobile apps: from infusion to diffusion in the mental health social system. JMIR mental health 2 (2015) e10

[33] Balgrosky, J.A.: Essentials of Health Information Systems and Technology. Jones & Bartlett Publishers (2014)

[34] Schobel, J., Pryss, R., Schlee, W., Probst, T., Gebhardt, D., Schickler, M., Reichert, M.: Development of mobile data collection applications by domain experts: Experimental results from a usability study. In: International Conference on Advanced Information Systems Engineering, Springer (2017) 60–75

[35] Abernethy, A.P., Ahmad, A., Zafar, S.Y., Wheeler, J.L., Reese, J.B., Lyerly, H.K.: Electronic patient-reported data capture as a foundation of rapid learning cancer care. Medical care (2010) S32–S38

[36] Bliven, B.D., Kaufman, S.E., Spertus, J.A.: Electronic collection of health-related quality of life data: validity, time benefits, and patient preference. Quality of Life Research 10 (2001) 15–21

[37] Pavlović, I., Kern, T., Miklavčič, D.: Comparison of paper-based and electronic data collection process in clinical trials: costs simulation study. Contemporary clinical trials 30 (2009) 300–316

[38] Pryss, R., Schobel, J., Reichert, M.: Requirements for a flexible and generic api enabling mobile crowdsensing mhealth applications. In: 2018 4th International Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS), IEEE (2018) 24–31

[39] Charland, A., Leroux, B.: Mobile application development: web vs. native. Communications of the ACM 54 (2011) 49–53

[40] Wasserman, T.: Software engineering issues for mobile application development. FoSER 2010 (2010)

[41] Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N.Y., Huang, R., Zhou, X.: Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. ACM Computing Surveys (CSUR) 48 (2015) 7

[42] Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine 49 (2011) 32–39

[43] Pryss, R., Schlee, W., Langguth, B., Reichert, M.: Mobile crowdsensing services for tinnitus assessment and patient feedback. In: 2017 IEEE International Conference on AI & Mobile Services (AIMS), IEEE (2017) 22–29

[44] Pryss, R., Probst, T., Schlee, W., Schobel, J., Langguth, B., Neff, P., Spiliopoulou, M., Reichert, M.: Mobile crowdsensing for the juxtaposition of realtime assessments and retrospective reporting for neuropsychiatric symptoms. In: 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), IEEE (2017) 642–647

[45] Pournajaf, L., Xiong, L., Garcia-Ulloa, D.A., Sunderam, V.: A survey on privacy in mobile crowd sensing task management. Dept. Math. Comput. Sci., Emory Univ., Atlanta, GA, USA, Tech. Rep. TR-2014-002 (2014)

[46] Ma, H., Zhao, D., Yuan, P.: Opportunities in mobile crowd sensing. IEEE Communications Magazine 52 (2014) 29–35

[47] Gaggioli, A., Pioggia, G., Tartarisco, G., Baldus, G., Corda, D., Cipresso, P., Riva, G.: A mobile data collection platform for mental health research. Personal and Ubiquitous Computing 17 (2013) 241–251

[48] Schobel, J., Ruf-Leuschner, M., Pryss, R., Reichert, M., Schickler, M., Schauer, M., Weierstall, R., Isele, D., Nandi, C., Elbert, T.: A generic questionnaire framework supporting psychological studies with smartphone technologies. (2013)

[49] Xiong, H., Huang, Y., Barnes, L.E., Gerber, M.S.: Sensus: a cross-platform, general-purpose system for mobile crowdsensing in human-subject studies. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM (2016) 415–426

[50] Pryss, R., Reichert, M., Schlee, W., Spiliopoulou, M., Langguth, B., Probst, T.: Differences between android and ios users of the trackyourtinnitus mobile crowdsensing mhealth platform. In: 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), IEEE (2018) 411–416

[51] Huber, J.: Conception and realization of a REST-based application programming interface for intersession-management in psychology (2019) Bachelor's Thesis, University of Ulm.

[52] Pfaff, V., Rejtö, R.: Angular Web Application for Intersession Processes. (2019) Software Engineering Project, University of Ulm.

[53] Gessler, M.: Android Mobile Application for Intersession Processes. (2019) Software Engineering Project, University of Ulm.

[54] Schobel, J., Pryss, R., Reichert, M.: Using smart mobile devices for collecting structured data in clinical trials: Results from a large-scale case study. In: 2015 IEEE 28th International Symposium on Computer-Based Medical Systems, IEEE (2015) 13–18

[55] Schobel, J., Probst, T., Reichert, M., Schickler, M., Pryss, R.: Enabling sophisticated lifecycle support for mobile healthcare data collection applications. IEEE Access 7 (2019) 61204–61217

[56] Schobel, J., Pryss, R., Schickler, M., Reichert, M.: A configurator component for end-user defined mobile data collection processes. In: International Conference on Service-Oriented Computing, Springer (2016) 216–219

[57] Schobel, J., Pryss, R., Schickler, M., Reichert, M.: A lightweight process engine for enabling advanced mobile applications. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer (2016) 552–569

[58] Schobel, J., Pryss, R., Wipp, W., Schickler, M., Reichert, M.: A mobile service engine enabling complex data collection applications. In: International Conference on Service-Oriented Computing, Springer (2016) 626–633

[59] Schobel, J., Pryss, R., Schickler, M., Reichert, M.: Towards flexible mobile data collection in healthcare. In: 2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS), IEEE (2016) 181–182

[60] Probst, T., Pryss, R., Langguth, B., Schlee, W.: Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the “trackyourtinnitus” application. Scientific reports 6 (2016) 20382

[61] Probst, T., Pryss, R., Langguth, B., Schlee, W.: Emotion dynamics and tinnitus: daily life data from the “trackyourtinnitus” application. Scientific reports 6 (2016) 31166

[62] Probst, T., Pryss, R.C., Langguth, B., Spiliopoulou, M., Landgrebe, M., Vesala, M., Harrison, S., Schobel, J., Reichert, M., Stach, M., et al.: Outpatient tinnitus clinic, self-help web platform, or mobile application to recruit tinnitus study samples? Frontiers in aging neuroscience 9 (2017) 113

[63] Schickler, M., Pryss, R., Reichert, M., Schobel, J., Langguth, B., Schlee, W.: Using mobile serious games in the context of chronic disorders: a mobile game concept for the treatment of tinnitus. In: 2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS), IEEE (2016) 343–348

[64] Schlee, W., Pryss, R.C., Probst, T., Schobel, J., Bachmeier, A., Reichert, M., Langguth, B.: Measuring the moment-to-moment variability of tinnitus: the trackyourtinnitus smart phone app. Frontiers in aging neuroscience 8 (2016) 294

[65] Kraft, R.: Conception and Realization of a Mobile Application Framework to Cope with the Communication Gap Between Health Care Manufacturer, Health Care Provider and Patient (2017) Master's Thesis, University of Ulm.

[66] Rickard, N., Arjmand, H.A., Bakker, D., Seabrook, E.: Development of a mobile phone app to support self-monitoring of emotional well-being: a mental health digital innovation. JMIR mental health 3 (2016) e49

[67] Agarwal, S., LeFevre, A.E., Lee, J., L'Engle, K., Mehl, G., Sinha, C., Labrique, A.: Guidelines for reporting of health interventions using mobile phones: mobile health (mhealth) evidence reporting and assessment (mera) checklist. bmj 352 (2016) i1174

[68] Hendrikoff, L., Kambeitz-Ilankovic, L., Pryss, R., Senner, F., Falkai, P., Pogarell, O., Hasan, A., Peters, H.: Prospective acceptance of distinct mobile mental health features in psychiatric patients and mental health professionals. Journal of psychiatric research 109 (2019) 126–132

[69] Ebert, C.: Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten. dpunkt. verlag (2019)

[70] Stapleton, J.: DSDM, dynamic systems development method: the method in practice. Cambridge University Press (1997)

[71] Apple Inc.: Human Interface Guidelines – iOS. Website (2019) `https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/` [Online; accessed 29.07.2019].

[72] Brewer, E.: Cap twelve years later: How the "rules" have changed. Computer 45 (2012) 23–29

[73] Brewer, E.A.: Towards robust distributed systems. In: PODC. Volume 7. (2000)

[74] Gilbert, S., Lynch, N.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. Acm Sigact News 33 (2002) 51–59

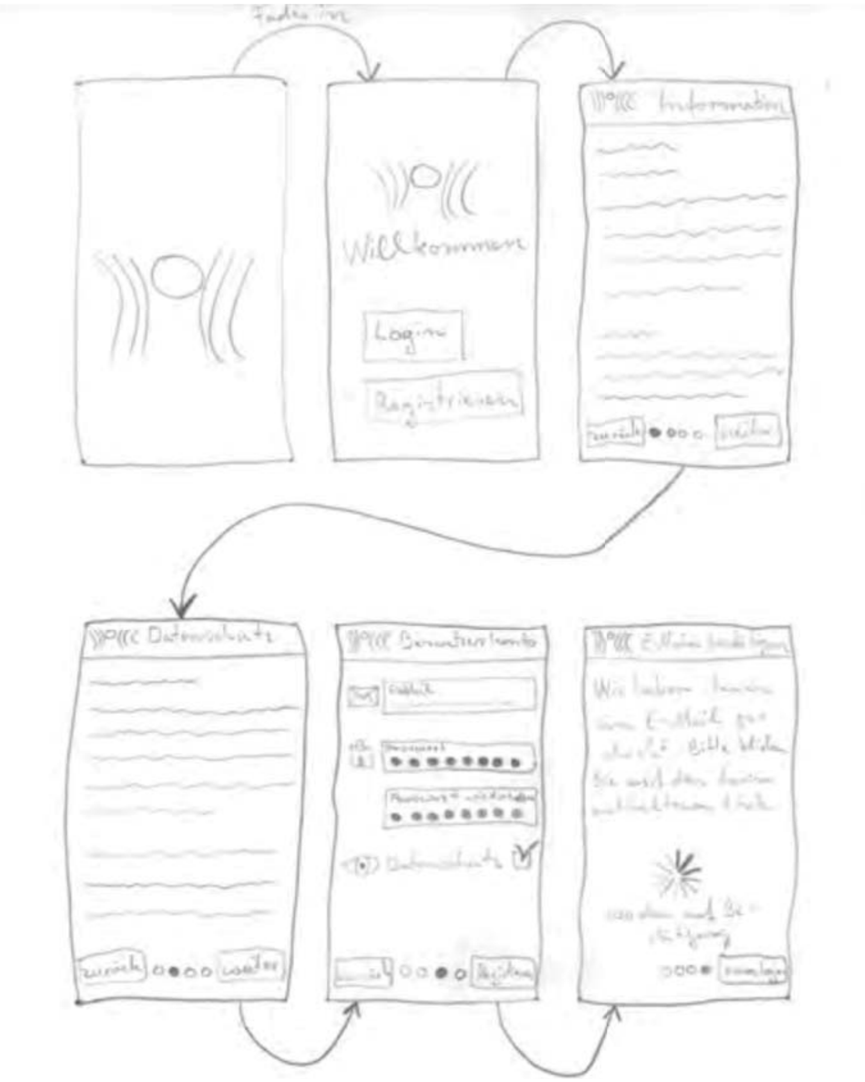
[75] Rabung, S., Harfst, T., Koch, U., Schulz, H.: Hamburger module (health-49). Website (November 2007) Hamburger Module, `http://www.hamburger-module.de/download/HEALTH-49.pdf` [Online; accessed 04.07.2019].

[76] Rabung, S., Harfst, T., Koch, U., Wittchen, H.U., Schulz, H.: „hamburger module zur erfassung allgemeiner aspekte psychosozialer gesundheit für die therapeutische praxis (health)"-psychometrische überprüfung eines neuen selbstbeurteilungsinstruments zur multidimensionalen erfassung psychosozialer gesundheit. Physikalische Medizin, Rehabilitationsmedizin, Kurortmedizin 17 (2007) 133–140

[77] Bartholomew, K., Horowitz, L.M.: Attachment styles among young adults: a test of a four-category model. Journal of personality and social psychology 61 (1991) 226

A

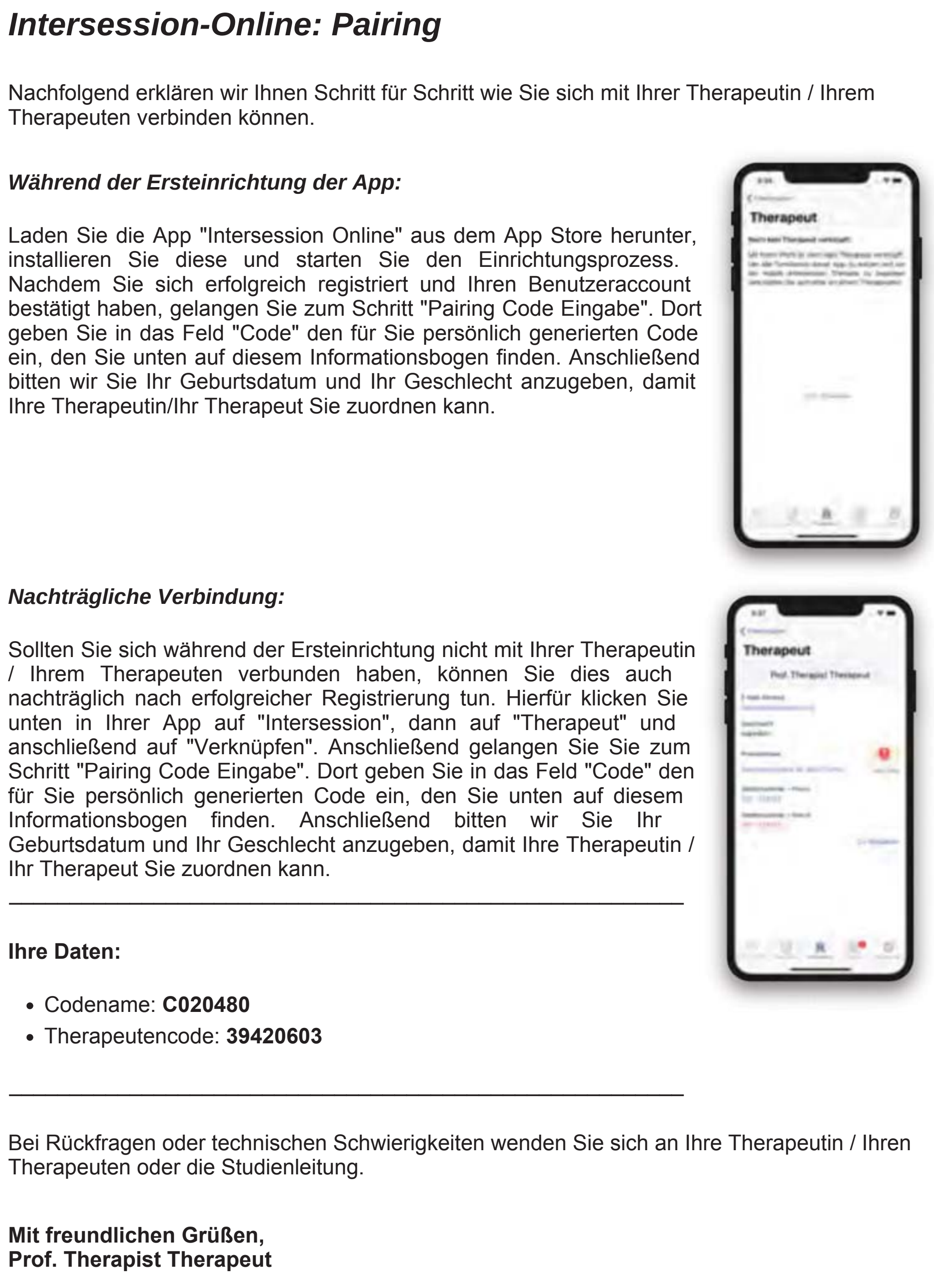
Screenshots, Pictures and PDFs



Wir haben Ihnen eine E-Mail geschickt! Bitte klicken Sie auf den darin enthaltenen Link.

warten auf Bestätigung

Figure A.1: Paper sketch – user registration



Therapeut
Therapeut
Prof. Therapist Therapeut

Intersession-Online: Pairing

Nachfolgend erklären wir Ihnen Schritt für Schritt wie Sie sich mit Ihrer Therapeutin / Ihrem Therapeuten verbinden können.

Während der Ersteinrichtung der App:

Laden Sie die App "Intersession Online" aus dem App Store herunter, installieren Sie diese und starten Sie den Einrichtungsprozess. Nachdem Sie sich erfolgreich registriert und Ihren Benutzeraccount bestätigt haben, gelangen Sie zum Schritt "Pairing Code Eingabe". Dort geben Sie in das Feld "Code" den für Sie persönlich generierten Code ein, den Sie unten auf diesem Informationsbogen finden. Anschließend bitten wir Sie Ihr Geburtsdatum und Ihr Geschlecht anzugeben, damit Ihre Therapeutin/Ihr Therapeut Sie zuordnen kann.

Nachträgliche Verbindung:

Sollten Sie sich während der Ersteinrichtung nicht mit Ihrer Therapeutin / Ihrem Therapeuten verbunden haben, können Sie dies auch nachträglich nach erfolgreicher Registrierung tun. Hierfür klicken Sie unten in Ihrer App auf "Intersession", dann auf "Therapeut" und anschließend auf "Verknüpfen". Anschließend gelangen Sie Sie zum Schritt "Pairing Code Eingabe". Dort geben Sie in das Feld "Code" den für Sie persönlich generierten Code ein, den Sie unten auf diesem Informationsbogen finden. Anschließend bitten wir Sie Ihr Geburtsdatum und Ihr Geschlecht anzugeben, damit Ihre Therapeutin / Ihr Therapeut Sie zuordnen kann.

Ihre Daten:

- Codename: **C020480**
- Therapeutencode: **39420603**

Bei Rückfragen oder technischen Schwierigkeiten wenden Sie sich an Ihre Therapeutin / Ihren Therapeuten oder die Studienleitung.

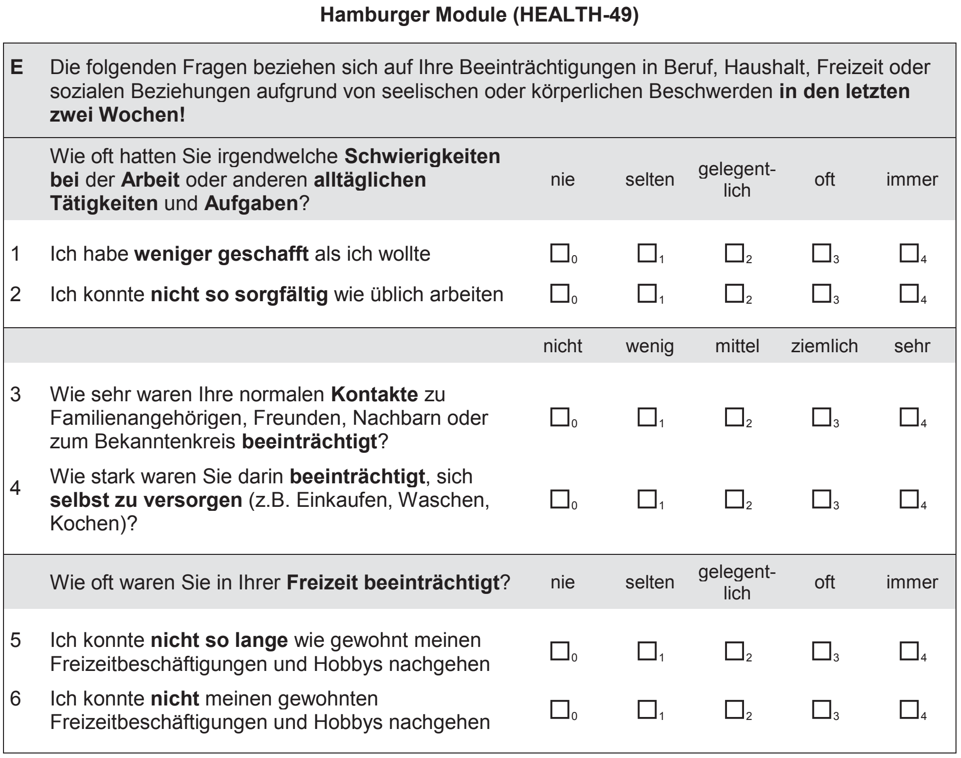
Mit freundlichen Grüßen,
Prof. Therapist Therapeut

Figure A.2: Intersession-Online: Pairing – therapist/pairing code [52]

B

Questionnaires

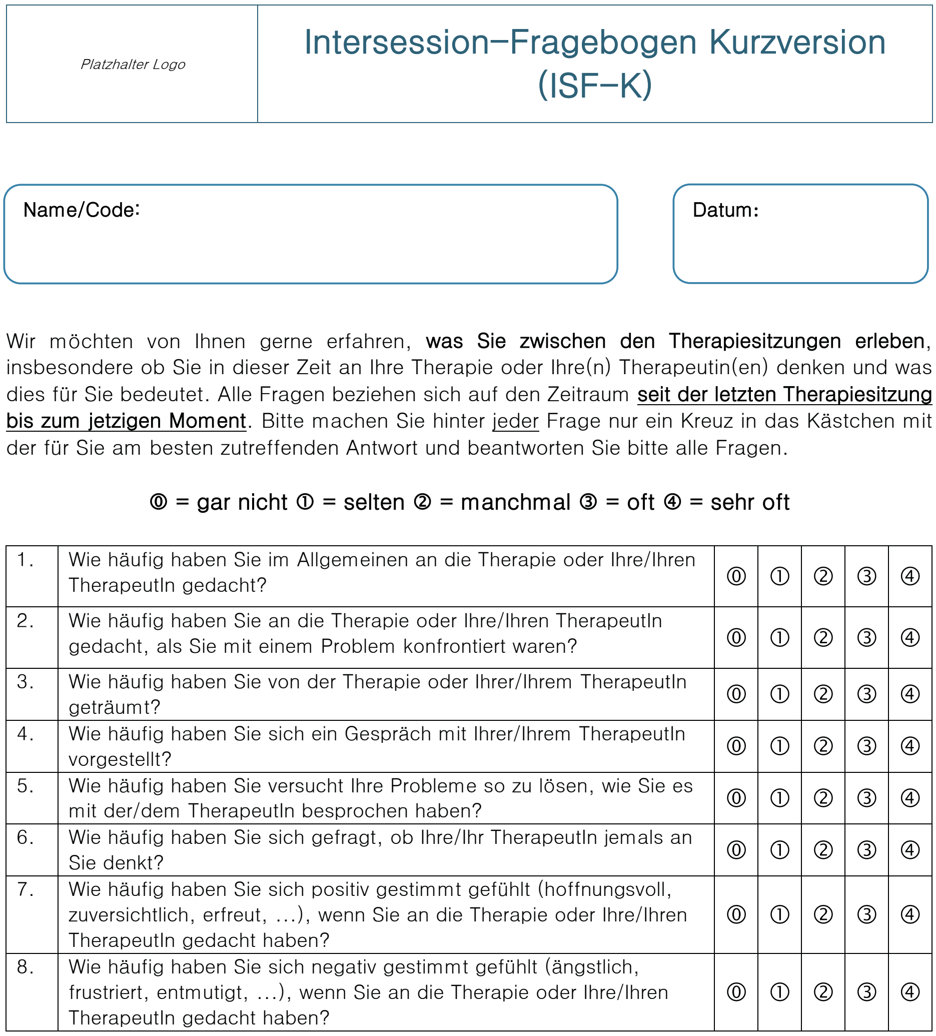
This appendix contains a selection of relevant questionnaires or excerpts thereof.



Hamburger Module (HEALTH-49)

E	Die folgenden Fragen beziehen sich auf Ihre Beeinträchtigungen in Beruf, Haushalt, Freizeit oder sozialen Beziehungen aufgrund von seelischen oder körperlichen Beschwerden **in den letzten zwei Wochen!**					
	Wie oft hatten Sie irgendwelche **Schwierigkeiten bei** der **Arbeit** oder anderen **alltäglichen Tätigkeiten** und **Aufgaben**?	nie	selten	gelegent-lich	oft	immer
1	Ich habe **weniger geschafft** als ich wollte	□0	□1	□2	□3	□4
2	Ich konnte **nicht so sorgfältig** wie üblich arbeiten	□0	□1	□2	□3	□4
		nicht	wenig	mittel	ziemlich	sehr
3	Wie sehr waren Ihre normalen **Kontakte** zu Familienangehörigen, Freunden, Nachbarn oder zum Bekanntenkreis **beeinträchtigt**?	□0	□1	□2	□3	□4
4	Wie stark waren Sie darin **beeinträchtigt**, sich **selbst zu versorgen** (z.B. Einkaufen, Waschen, Kochen)?	□0	□1	□2	□3	□4
	Wie oft waren Sie in Ihrer **Freizeit beeinträchtigt**?	nie	selten	gelegent-lich	oft	immer
5	Ich konnte **nicht so lange** wie gewohnt meinen Freizeitbeschäftigungen und Hobbys nachgehen	□0	□1	□2	□3	□4
6	Ich konnte **nicht** meinen gewohnten Freizeitbeschäftigungen und Hobbys nachgehen	□0	□1	□2	□3	□4

Figure B.1: Hamburger Module (HEALTH-49) – E [75, p. 3] designed by [76]



Platzhalter Logo

Intersession–Fragebogen Kurzversion (ISF–K)

Name/Code:

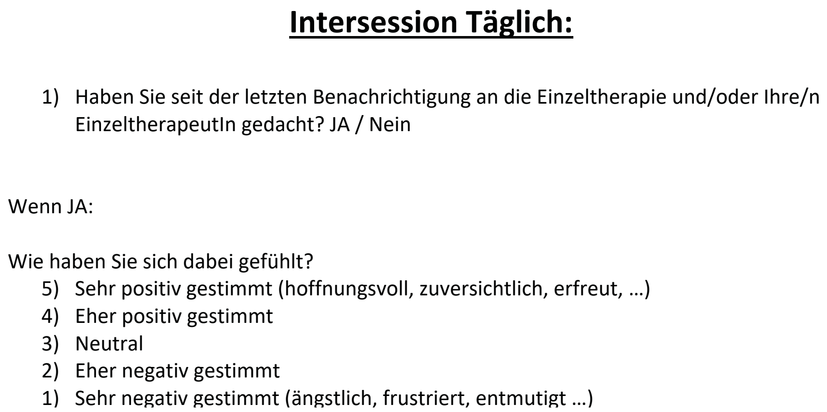
Datum:

Wir möchten von Ihnen gerne erfahren, **was Sie zwischen den Therapiesitzungen erleben**, insbesondere ob Sie in dieser Zeit an Ihre Therapie oder Ihre(n) Therapeutin(en) denken und was dies für Sie bedeutet. Alle Fragen beziehen sich auf den Zeitraum seit der letzten Therapiesitzung bis zum jetzigen Moment. Bitte machen Sie hinter jeder Frage nur ein Kreuz in das Kästchen mit der für Sie am besten zutreffenden Antwort und beantworten Sie bitte alle Fragen.

⓪ = gar nicht ① = selten ② = manchmal ③ = oft ④ = sehr oft

1.	Wie häufig haben Sie im Allgemeinen an die Therapie oder Ihre/Ihren TherapeutIn gedacht?	⓪	①	②	③	④
2.	Wie häufig haben Sie an die Therapie oder Ihre/Ihren TherapeutIn gedacht, als Sie mit einem Problem konfrontiert waren?	⓪	①	②	③	④
3.	Wie häufig haben Sie von der Therapie oder Ihrer/Ihrem TherapeutIn geträumt?	⓪	①	②	③	④
4.	Wie häufig haben Sie sich ein Gespräch mit Ihrer/Ihrem TherapeutIn vorgestellt?	⓪	①	②	③	④
5.	Wie häufig haben Sie versucht Ihre Probleme so zu lösen, wie Sie es mit der/dem TherapeutIn besprochen haben?	⓪	①	②	③	④
6.	Wie häufig haben Sie sich gefragt, ob Ihre/Ihr TherapeutIn jemals an Sie denkt?	⓪	①	②	③	④
7.	Wie häufig haben Sie sich positiv gestimmt gefühlt (hoffnungsvoll, zuversichtlich, erfreut, …), wenn Sie an die Therapie oder Ihre/Ihren TherapeutIn gedacht haben?	⓪	①	②	③	④
8.	Wie häufig haben Sie sich negativ gestimmt gefühlt (ängstlich, frustriert, entmutigt, …), wenn Sie an die Therapie oder Ihre/Ihren TherapeutIn gedacht haben?	⓪	①	②	③	④

Figure B.2: Intersession-Fragebogen Kurzversion (ISF-K) [14]



Intersession Täglich:

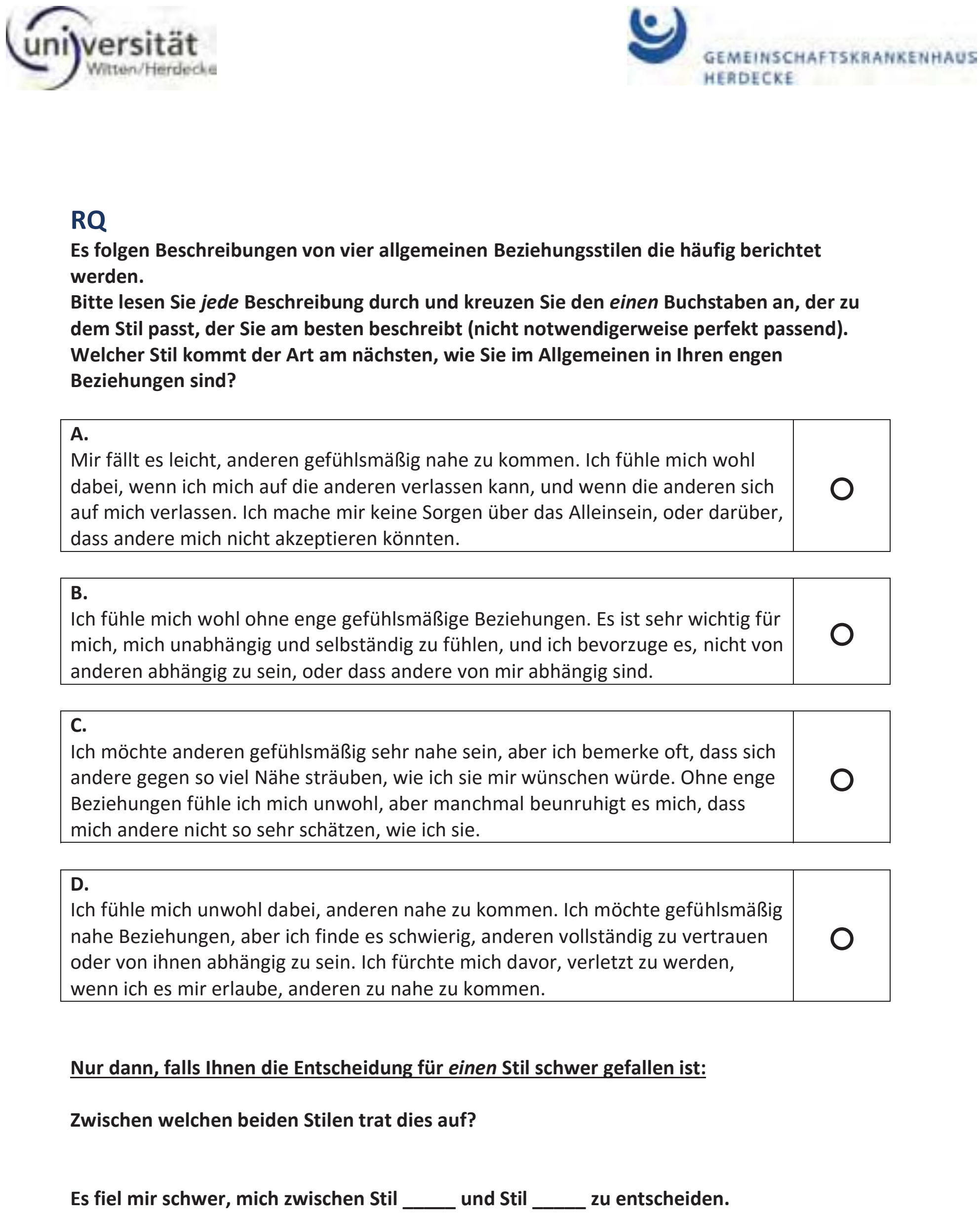
1) Haben Sie seit der letzten Benachrichtigung an die Einzeltherapie und/oder Ihre/n EinzeltherapeutIn gedacht? JA / Nein

Wenn JA:

Wie haben Sie sich dabei gefühlt?

5) Sehr positiv gestimmt (hoffnungsvoll, zuversichtlich, erfreut, …)
4) Eher positiv gestimmt
3) Neutral
2) Eher negativ gestimmt
1) Sehr negativ gestimmt (ängstlich, frustriert, entmutigt …)

Figure B.3: Intersession-Daily by Gablonski, T.-C.



universität Witten/Herdecke

GEMEINSCHAFTSKRANKENHAUS HERDECKE

RQ

Es folgen Beschreibungen von vier allgemeinen Beziehungsstilen die häufig berichtet werden.
Bitte lesen Sie *jede* Beschreibung durch und kreuzen Sie den *einen* Buchstaben an, der zu dem Stil passt, der Sie am besten beschreibt (nicht notwendigerweise perfekt passend).
Welcher Stil kommt der Art am nächsten, wie Sie im Allgemeinen in Ihren engen Beziehungen sind?

A. Mir fällt es leicht, anderen gefühlsmäßig nahe zu kommen. Ich fühle mich wohl dabei, wenn ich mich auf die anderen verlassen kann, und wenn die anderen sich auf mich verlassen. Ich mache mir keine Sorgen über das Alleinsein, oder darüber, dass andere mich nicht akzeptieren könnten.	○
B. Ich fühle mich wohl ohne enge gefühlsmäßige Beziehungen. Es ist sehr wichtig für mich, mich unabhängig und selbständig zu fühlen, und ich bevorzuge es, nicht von anderen abhängig zu sein, oder dass andere von mir abhängig sind.	○
C. Ich möchte anderen gefühlsmäßig sehr nahe sein, aber ich bemerke oft, dass sich andere gegen so viel Nähe sträuben, wie ich sie mir wünschen würde. Ohne enge Beziehungen fühle ich mich unwohl, aber manchmal beunruhigt es mich, dass mich andere nicht so sehr schätzen, wie ich sie.	○
D. Ich fühle mich unwohl dabei, anderen nahe zu kommen. Ich möchte gefühlsmäßig nahe Beziehungen, aber ich finde es schwierig, anderen vollständig zu vertrauen oder von ihnen abhängig zu sein. Ich fürchte mich davor, verletzt zu werden, wenn ich es mir erlaube, anderen zu nahe zu kommen.	○

Nur dann, falls Ihnen die Entscheidung für *einen* Stil schwer gefallen ist:

Zwischen welchen beiden Stilen trat dies auf?

Es fiel mir schwer, mich zwischen Stil _____ und Stil _____ zu entscheiden.

Figure B.4: Relationship Questionnaire (RQ) [77]

List of Figures

List of Tables

Name: Carsten Vogel Matriculation number: 766457

Honesty disclaimer

I hereby affirm that I wrote this thesis independently and that I did not use any other sources or tools than the ones specified.

Ulm, ...

Carsten Vogel