



Ulm University | 89069 Ulm | Germany

**Faculty of Engineering,
Computer Sciences and
Psychology**

Institute of Databases
and Information Systems
(DBIS)

Conception and realization of a REST-based application programming interface for intersession-management in psychology

Bachelor thesis at Ulm University

Submitted by:

Julian Huber

julian.huber@uni-ulm.de

Reviewers:

Prof. Dr. Manfred Reichert

Adviser:

Dr. Rüdiger Pryss

2019

Revision March 22, 2019

© 2019 Julian Huber

Satz: PDF-L^AT_EX 2_ε

Contents

1	Introduction	1
1.1	Related Work	1
1.2	Structure of the Thesis	2
2	Fundamentals	3
2.1	Setting and Approach	3
2.1.1	Collection of Patient Data	3
2.1.2	Usage of the Collected Data	4
2.2	Intersession Online: a Multi Device Platform	5
2.2.1	Mobile Applications	5
2.2.2	Web Application	6
2.2.3	Application Programming Interface	6
3	Requirements Engineering	7
3.1	Information Flow	7
3.2	Actors and User Groups	7
3.3	Requirements	8
3.3.1	Functional Requirements - General	8
3.3.2	Functional Requirements - Patient	8
3.3.3	Functional Requirements - Therapist	9
3.3.4	Functional Requirements - Researcher	10
3.4	Non Functional Requirements and Boundary Conditions	10
4	Software Development Process and Concept	13
4.1	Concepts and Tools	13
4.1.1	JavaScript Object Notation	13
4.1.2	Representational State Transfer - REST	14
4.1.3	API Documentation	14
4.2	Software Development Aspects	15
4.2.1	Agile Software Development	15
4.2.2	Continuous Integration with Bitbucket Pipelines	16

4.2.3	Testing	17
5	Implementation	19
5.1	Basic Structure	19
5.1.1	Database	19
	Accounts	20
	Questionnaires	20
	Contributions	20
	Interventions	21
	Seeding Data	21
5.1.2	Architecture and Classes	21
5.2	Programming Insights	22
5.2.1	Functionality and Difficulties	22
5.2.2	Pairing of Patient and Therapist	23
5.2.3	Intervention Assignments	24
6	Usage	27
6.1	Using the Application Interface as Developer	27
6.1.1	HTTP Requests	27
6.1.2	JSON Web Token	28
6.1.3	Offline Function	28
6.2	Current Usage Examples	29
6.2.1	Web Interface	29
6.2.2	Mobile Applications	29
7	Conclusion	31
A	Pictures and Screenshots	33
B	Code Snippets	47
	Bibliography	51

Glossary

anamnesis information gained by a physician by asking specific questions of a patient. 4, 5

API Application Programming Interface. 6, 7, 8, 10, 13, 14, 15, 17, 19, 21, 22, 27, 28, 29, 31

catamnesis the follow-up history of a patient after discharge from treatment. 4

CSV Comma separated values. 12

GUI Graphical User Interface. 15

HTTP Hypertext Transfer Protocol. 14, 19, 21, 22, 27

Intersession The time between two therapy sessions. 1, 3, 4, 5

JSON JavaScript Object Notation. 13, 14, 15, 27

JWT JSON Web Token. 28

ORM Object Relational Mapper. 21

REST Representational State Transfer. 14

XML Extensible Markup Language. 14

1 Introduction

Mobile software applications have become one enabler in the field of medical treatments. This is comprehensible, since it enables “open access to details about illnesses, diseases, health promotion and healthcare” [20]. Especially in the psychological branch, Home-Based Therapy and medical mobile apps are widespread. Furthermore, the analysis of patient data is difficult because it is rarely received digitally and can hardly be processed. Since 2017, the Alps-Adria-University Klagenfurt has been trying to improve the research progress in this field, exceedingly in the data collection of the patient’s Intersession processes. In detail, they want to develop a multi device platform to collect, analyse, illustrate and process the data of the patients as simple as possible.

In times of Big Data and Public Health Databases, the collection of information has carried through and there is a huge amount of ways to gather as much information as possible. The most common form is probably the Application Programming Interface, which enables easy communication between applications and offers a good possibility to save data to a server.

1.1 Related Work

Due to the big workload of analogue surveys, Johannes Schobel of Ulm University has been researching on a questionnaire framework for over five years [8]. By creating QuestionSys, he managed to build an “easy-to-use and self-explaining framework for creating, running, and evolving the questionnaires of psychological studies on mobile and smart devices” [29]. Smartphone assisted therapy approaches are a topic of research in more ways, one of them is MobileTx [6]. By using web services to connect the devices of researchers, therapists and patients, it tries to support the psychotherapy by sending homework into the everyday life of the user. Of course, this does not only include the sending of the work per se, but also figuring out suitable techniques, collecting data on several devices and analysing the patient’s behaviour. Obviously, this does not only help in psychological treatment, but can generally be

used in every medical field where homework is part of the therapy [25][28][26][27]. The University of Ulm also has more foundational research progresses, for example defining requirements for generic APIs, which are used for crowdsensing [21].

1.2 Structure of the Thesis

After giving an understanding of the study in which this project is created, the given environment is explained. After that, the functional- and non-functional requirements are listed, which are used to create the concept shown in the following Chapter. Then, the implementation itself is displayed, containing basic code structures and some inside views of the used algorithms. After explaining the usage of the API, this thesis ends with the conclusion and future possibilities.

2 Fundamentals

In the following, essential parts of the analysis and treatment of patients with mental illnesses, for example depression or anxiety disorders, are presented. Furthermore, the current approach of the therapists and researchers is shown, as well as how they want to modernise it by using the application “Intersession-Online”, which is created in this project.

2.1 Setting and Approach

Since many years, researchers in psychology are searching for the most efficient form of therapy, the time between its sessions was more or less neglected. Latest research however show more than ever, that the so called “Intersession processes”[11] are quite an important part of the recovery. In the course of his PhD thesis, Thorsten-Christian Gablonski is working on a study to “acquire and control Intersession processes” [15, p. 1]. To simplify the collection of patient data and to automate the interventions patients get, an online platform is developed. By defining different groups of patients, one that uses the platform and another one that doesn’t, comparable results are created.

2.1.1 Collection of Patient Data

As part of the therapy, patients have to fill out a considerable big amount of questionnaires. When it comes to the chronological aspect, the questionnaires are separated in four groups. Due to the temporal relevance of the completion, it is strictly prescribed, when a certain questionnaire is filled out. The times are defined as follows:

Group 1 Three times daily, once between 8:00 am and 11:59 am, once between noon and 3:59 pm and once between 4:00 pm and 8:00 pm. For each timeslot, the exact time is chosen randomly.

Group 2 Two hours after each therapy session, except if it would be after 8:00 pm, then it has to be filled out at 8:00 am the next day.

Group 3 Once every Saturday or Sunday, at a random time between 8:00 am and 8:00 pm.

Group 4 On the anamnesis and on the catamnesis. Also one, six and twelve months after the end of the therapy.

Which questionnaires are used, as well as their classification is described in Table 2.1. To ensure the safety of the patient's data, a code is used to label the contributions. The questionnaire results help the therapist to provide a therapy that fits the patient's needs.

Questionnaire	Group 1	Group 2	Group 3	Group 4
Intersession Fragebogen kurz (ISF-K) [17]			X	
Intersession Täglich	X			
Working Alliance Inventory (WAI) [31]		X		
SCL-K9 [19]		X	X	
Childhood-Trauma-Questionnaire (CTQ) [32]				X
Liste von Überzeugungsmustern [24]				X
Mentalization Questionnaire (MZQ) [18]				X
Relationship Questionnaire (RQ) [12]				X
Inventory of Personality Organisation (IPO-57) [14]				X
Persönlichkeitsinventar für DSM-5 (PID-5-BF) [33]				X
Health-49 (complete) [22]				X

Table 2.1: Grouping of the needed questionnaires

Because the fact, that every questionnaires have the same structure and just the number of answers varies, only two questionnaires are attached, the ISF-K in Figure A.1 and the SCL-K9 in Figure A.2.

2.1.2 Usage of the Collected Data

On the basis of the collected data, researchers of the Alps-Adria-University Klagenfurt investigate the impacts of Intersession processes and how they can be controlled. By

defining tasks for the time between two therapy sessions, they try to influence the patient's behaviour to regulate his Intersession experience. Besides that, the gathered information can be used for additional analysis of the correlation between collected data and outcome of the therapy.

2.2 Intersession Online: a Multi Device Platform

As a simplification of the research progress and to compare different ways of evaluating the patient's data, an online platform for multiple devices is developed. The application's purpose is to reduce the time that is spent on filling out and evaluating the questionnaires. On top of that, the customer's hope is to open new possibilities, such as displaying the collected data in new ways. To ensure a perfect workflow for all parties involved, the platform is split in three different parts:

Mobile Application Android- and iOS specific applications, that enable a handy way of filling out the questionnaires.

Web Platform By displaying the collected data to the therapists and researchers, the web platform offers a good way to get an overview of the patient's results.

API With an automated evaluation of the contributed questionnaires, the application programming interface connects the other parts to one platform.

Below, a small introduction to all parts is provided.

2.2.1 Mobile Applications

Filling out questionnaires is a major part when it comes to the collection of the patient's data. At the anamnesis, for example, 200 questions have to be answered. Adding the questionnaires of group one to three, the number grows to 300 in the first week. To ease that task, a mobile application is presented to the patients. Since a large amount of people have their smartphone around them most of the time, questionnaires can be filled from anywhere and, even more important, at an exact point of time. To ensure that, push notifications remind the patients to make their contributions. That improves the acquisition of the user's Intersession processes, one of the two aspects, that T.-C. Gablonski's study tries to improve. The other aspect, the control of those processes, can be enhanced by interventions based on the survey results, which are sent to the patient's smartphone. The application should be provided for Android and iOS.

2.2.2 Web Application

Not only the patients, but also the therapists and researches benefit from the application. By providing graphs of the patient's data, the results can be viewed in a helpful way without having to digitalize any data before. The platform also provides an overview of the patients that are connected with the therapist. On top of that, the used questionnaires can be displayed and relevant information of the therapist can be provided to his patients. To make sure that the therapist doesn't miss a contribution, a dashboard shows the latest notifications.

2.2.3 Application Programming Interface

As connection between Web- and Mobile Application, an Application Programming Interface (API) is provided. This API is not only responsible for the patient therapist connection and the evaluation of the questionnaires, but also ensures the privacy of the user data. All random and automated aspects are calculated on the server behind the API, for instance the exact time when the user is notified or which intervention is sent to the user. This Thesis is based on the implementation of the interface, so in the following chapters, the concept, the implementation and the software engineering process is described and explained. Because of the included logical unit, often used synonyms of the API are backend or server.

3 Requirements Engineering

Below, the demands and requirements of the customer, represented by Thorsten-Christian Gablonski, are listed. Not only the requirements itself, but also the general conditions are explained.

3.1 Information Flow

To grant a flawless information flow, several difficulties had to be overcome. Since the customer is a group of researchers from the Alps-Adria-University Klagenfurt, spontaneous meetings were hardly possible. Our first point of contact was Thorsten-Christian Gablonski, who provided us great support by making several meetings possible, either in person or via Skype. The most challenging aspect in obtaining the requirements have been the different fields of work. While I lack the psychological know-how, T.-C. Gablonski is missing the experience with the possibilities of multi device applications.

3.2 Actors and User Groups

To prevent confusion, relevant actors of the system are displayed in Figure 3.1. This cannot only contain human interaction partners, but also system elements like the implemented API. In Table 3.1, an overview of important actors is given. Another role, that is not displayed, but should be mentioned is the customer, which is the Alps-Adria-University, represented by Thorsten-Christian Gablonski. Not displayed in Table and Figure, but important as well, are the developers of Ulm University, Carsten Vogel, Réka Rejtő, Verena Pfaff, Miriam Gessler and me, supervised by Rüdiger Pryss.

Role	Explanation
Online platform	API, Mobile Application and Web Application combined to one platform. It is used, when multiple parts have to be taken into account or for imprecise explanations.
API	The Application Programming Interface serves as communication unit for Mobile- and Web Application. This does not only mean the interface as it is, but also the server that runs in the background. Synonyms can be the backend or server.
Mobile Application	The Android- or iOS Application, which is used by the patient. They both have the same possibilities, without difference, so they are combined into one actor.
Web Application	The frontend for therapists and researchers, reachable with by any browser.
User	A person, that uses the online platform.
Guest	A user, that is using the online platform, but is not logged in.
Patient	A logged in user, that is, will be or was in a therapy.
Therapist	A logged in user, that is a therapist.
Researcher	A logged in user, that is a researcher.

Table 3.1: Overview of actors and roles

3.3 Requirements

Because of the fact that this Thesis only covers the API, requirements of the Mobile- and Web Application are ignored in the following sections. Also keep in mind that the explanations might not be enough to reproduce the requirements for the other parts of the platform, since only the aspects of the programming interface are covered.

3.3.1 Functional Requirements - General

A large amount of requirements can be summarised for all user groups. Especially system functions regarding the account management are the same or very similar for every user, why they are collected in Table 3.2.

3.3.2 Functional Requirements - Patient

Since the patient does only interact with the mobile application, the requirements can be distinguished clearly from those of therapists and researchers. They are listed in

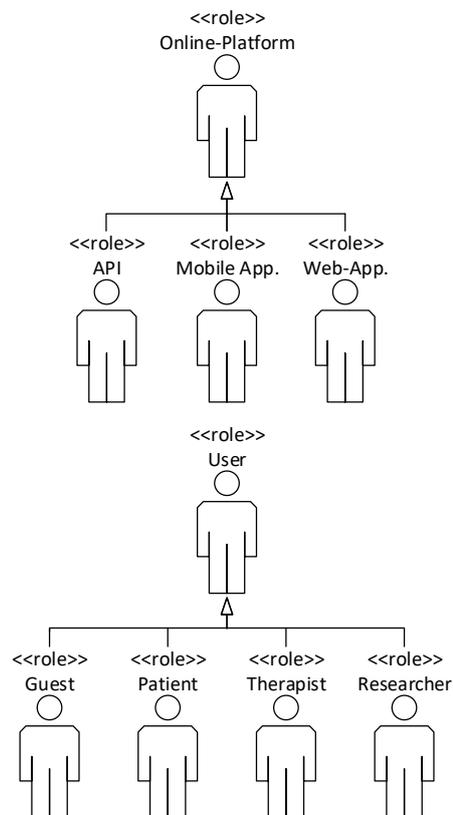


Figure 3.1: Actors and roles of the system

Table 3.3. Obviously, patients build the group of people, that participate in the study that is explained in Chapter 2.

3.3.3 Functional Requirements - Therapist

The most important part for the therapists is the displaying of data. Because of the usefulness of the bigger screen, all requirements in Table 3.4 are implemented as a part of a web application. The therapist does not have the possibility of logging in while using the mobile application. Furthermore, the therapist can only see the possibilities, he is allowed to use.

Requirement	Explanation
Login	The API offers a possibility to log into the system with the credentials, that are registered in the database.
Logout	When a user is logged into the system, there is a possibility to logout, which leads to a termination of the session.
Email verification	An Account is only usable, when it is created with an email that is verified. This is achieved by clicking on a link, which is sent to the given email address.
Password forgotten	In case of a forgotten password, the user can reset it.
Profile	The user can see his profile data. That does not only contain his personal information, but also his role in the system.

Table 3.2: General requirements

3.3.4 Functional Requirements - Researcher

Just like therapists, researchers use the web application. Only the possibilities, that can be used by the researcher should be shown, they are listed in Table 3.4.

3.4 Non Functional Requirements and Boundary Conditions

In contrast to the functional requirements, which have been listed quite strictly, the non functional requirements and the boundary condition where more loose. Nevertheless, some standards had to be observed.

Confidentiality The system should not make it possible for unauthorized users to claim any patient data.

Language Since most patients speak German, it is the only language used on the application.

Maintenance The university provides updates and fixes problems for 3 years.

Reliability Because of the short period of time in which questionnaires should be filled, the system does not have any major downtimes.

3.4 Non Functional Requirements and Boundary Conditions

Requirement	Explanation
Registration	Providing only email address and password, patients can register in the application.
Pairing	The patient has a possibility to specify, to which therapist he belongs.
Therapist data	There is a way to see the personal information of the therapist, that is connected with the patient.
Appointments	Patients can manage their appointments with the therapist. That means, appointments can be stored, changed and deleted.
Researcher access	To provide enough privacy, the user can grant and revoke the access to his provided contributions at any time. This only effects the access of the researcher, his therapist can still see the data.
Assignments	As listed in 2.1, the patient gets push notifications, when he has to fill out a certain questionnaire. This assignment expires after a prescribed duration.
Contributions	While an assignment is active, the patient can fill out a specific questionnaire and store the contribution into the system. This contribution can be shown by the patient.
Interventions	Depending on the results of the questionnaire contributions, the patient is assigned to interventions. They consist of small texts and questions, but have not to be answered. Interventions can be marked as fulfilled.
Deletion	The patient can delete his account at any time.

Table 3.3: Patient requirements

Requirement	Explanation
Registration	A therapist can register on the website. Unlike patients, they have to give further information of themselves, such as address and gender.
Therapist codes	To provide a possibility of connecting with a therapist, codes can be created. When a patient uses one of the therapist codes, his gender and date of birth are saved in the system. Therapist codes can also be deleted by the therapist.
Notifications	Therapists get notifications on the dashboard. They contain a heading and further information. Notifications are created, when one of his patients contribute a questionnaire, cancel the pairing or delete himself from the system.
Therapist data	To offer further information, therapists can change and provide more data of themselves. This data contains academic degree, name, address, gender and phone number.
Patient overview	An overview of the connected patients is provided. This contains not only the patient's data, but also his contributed questionnaires and a list of his interventions.
Unpair from patient	The therapist is able to unpair from a specific patient.

Table 3.4: Therapist requirements

Requirement	Explanation
Patient Overview	An overview of the patients is provided. This contains not only the patient's data, but also his contributed questionnaires and a list of his interventions. All patients are shown, but only the contributions of those, who grant researcher access, can be seen.
Therapist Overview	A list of all registered therapists is viewable. They can be activated or deactivated.
CSV Export	The contributions of patients can be downloaded as CSV. Only patients that provide researcher access are included in the CSV.

Table 3.5: Researcher requirements

4 Software Development Process and Concept

Since the project is not only a prototype, but is used by several patients with mental disorders, it was convenient to find several professional approaches instead of just starting to code. Below, some of those considerations are mentioned and explained.

4.1 Concepts and Tools

After defining functional and non-functional requirements, it is possible to find the right tools to realize the API. Main goals for the interface are being self explanatory and unambiguously, since it is used by different parties and should not have to be explained more than necessary. On the basis of that conditions, the following environment was chosen.

4.1.1 JavaScript Object Notation

As mentioned, the API should be built being self explanatory. This is one of the many advantages of the JavaScript Object Notation (JSON) format. Because of the fact, that the notation is in general a "sequence of tokens" [13], the data is readable by humans. That simplifies the usage of the API a lot for developers, since the answers are understandable. On top of that, JSON is known to be good practice in the online environment. It is widely used, not only because JavaScript uses the same structure and is used by almost every website. As shown in the Listing 4.1, JSON can display all needed data types.

Listing 4.1: JSON Code

```
1 //Object containing data of Julian Huber
2 {
3   "name": "Julian Huber",
4   "parents": {
5     "mother": "Heike",
6     "father": "Juergen"
7   },
8   "children": null,
9   "siblings": ["Daniel"],
10  "age": 27,
11  "height": 1.83,
12  "is_male": true
13 }
```

An alternative would have definitely been Extensible Markup Language (XML), but because of personal preference, JSON was chosen.

4.1.2 Representational State Transfer - REST

To get an ambiguous interface with a structured format, Representational State Transfer (REST) was taken as a guideline. For REST architectures, the most important idea is working with resources. Principally, every data object, such as users, contributions or others are treated as a unique resource. That opens many benefits, for example the following:

Unique Resource Identifier Every resource is accessible with a URI.

Statelessness No application session is needed, which opens the interface to be used by almost every device. Also, multiple devices can be used simultaneously.

Consistency By using standard methods like the Hypertext Transfer Protocol (HTTP) functions GET, POST, PUT, DELETE for every resource, the API is easy to use.

4.1.3 API Documentation

To make the API easy to understand and as a summary of the offered functions, a documentation with every available resource was created. To use an external graph-

ical interface, the resources are described with OpenAPI 3.0, a community driven, open specification under the Apache 2.0 license. By using OpenAPI, it is possible to offer a Swagger-UI [10] version of it. For each resource, a title, a description and parameters are listed. The most important part though is, that all possible responses are explained and an example value is given. The UI even goes so far as that every call can be done from the browser and the live response can be seen with status code, header and body. Because of the used JSON format, every answer can be described uniquely. For more impressions, see Figure A.3 and Figure A.4.

4.2 Software Development Aspects

Even though there haven't been guidelines as to which methods should be used, I tried to abide several modern approaches. In this chapter, they are presented.

4.2.1 Agile Software Development

To find an appropriate development approach, different circumstances have been taken into account. First of all, up to three different groups worked together simultaneously. This in fact excluded almost all linear approaches, because it would have been very hard for the other teams to develop their part. Due to the massive information procurement, the first parts of the API would have been usable after weeks, what makes an early merge of the Graphical User Interface (GUI) and the API impossible. On top of that, in my early researches, i came across an article of the Centers for Medicare & Medicaid Services [1]. In "Selecting a development approach" [9], they suggest the Spiral Method, which i tried to stick with.

Although a consistent duration could not be achieved for each iteration, an agile software development process has definitely been accomplished. Looking back, the implementation steps can be split in four major categories, that have been developed in an agile manner.

Account Management Not only contains Login, Registration and Logout, but also the Role System, that divides the users in patients and therapists. Also Email Verification, Password Management and Notifications are a part of it.

Questionnaires Consisting of Displaying the needed questionnaires in an appropriate way.

Contributions To save the patient data into the system, the filled out questionnaires have to be saved in the database.

Interventions Based on the contributed questionnaires, each patient that is currently in a therapy gets an intervention.

Of course, these steps have been divided in small portions to ensure structured developing. Every step contained not only planning and coding, but also creating a documentation 4.1.3 and writing tests 4.2.3. To provide the newest version to the team right away, a pipeline realized Continuous Integration 4.2.2.

4.2.2 Continuous Integration with Bitbucket Pipelines

By using a Bitbucket Pipeline, a notification was sent directly when a written unit test, described in Section 4.2.3, failed. Since the server machine was known from the beginning, the pipeline could be set to be a clone of the original environment. Sometimes, errors only occur at specific circumstances and would therefore not be detected, if the used Docker Container would not match the Deployment Server. To be more specific, a php7.2 Docker Image serves as a foundation of the pipeline. By setting up a MySQL 5.7 Service, the database settings correspond with the ones that are on the Deployment Server. More precisely, the pipeline did following things:

Prepare environment As mentioned, a Debian System runs on the docker image, containing php7.2. On top of that, composer is installed. The system is updated and the needed MySQL extensions are installed, as seen in lines 1-15 at B.1.

Set Framework Laravel, the used framework, works without an installation and is included in the Git Repository. To be able to run the tests, the right environment file has to be copied. Also, seeding data has to be included. These steps are described in lines 16-19 at B.1.

Run tests In line 20, the unit tests are finally run. A positive, as well as a negative example are shown in Figures A.6 und A.7. If a test fails, an email is sent to the person that started the pipeline.

At the end of the project, around 50% of the 227 pipelines failed, what means that a massive amount of errors have been detected with it. For the exact structure of the pipeline, see Listing B.1.

4.2.3 Testing

Before publishing a new function, PHPUnit Tests assured, that the written code was working as it should. Since testing was neither requested in the non-functional requirements, nor was it from the supervisor, a fast but effective approach to cover most of the test cases was the best solution. Therefore, equivalence classes have been searched, mainly given by the validation rules. Regarding to our project, tests did not only ensure the system functions directly after writing them, but by automatically call every test in the pipeline 4.2.2, malfunctions of older system parts are detected and can be fixed. A considerable big amount of seeding data enabled the possibility to define the tests as situations, that could occur in the later usage of the API. To keep the test cases structured, they are sorted in the same way the controllers are, as seen in Figure 4.1. Overall, around 7000 assertions in nearly 50 test methods are used.

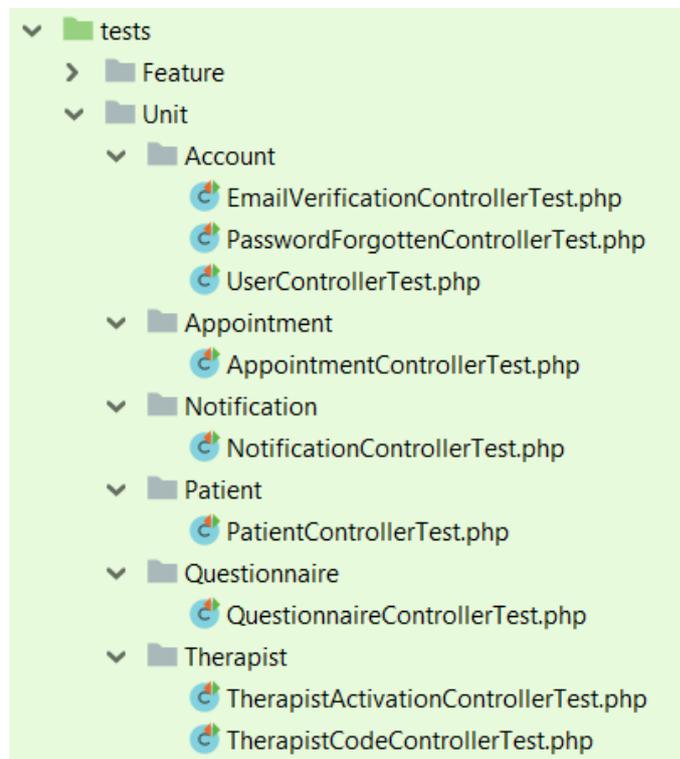


Figure 4.1: Structure of the unit tests

5 Implementation

Below, several aspects of the implementation are stated. To avoid problems, that are already solved, a Framework was used as code basis, the Laravel PHP Framework [3], version 5.7. Besides handling the database operations and basic HTTP-operations, it enables a possibility to import thousands of plugins.



Figure 5.1: Logo of the Laravel Framework

5.1 Basic Structure

In this section, the basic database- and code structure is explained to get an insight into the project's implementation. Keep in mind, that the API's documentation is not explained, since this is already done in Section 4.1.3.

5.1.1 Database

Server side, the data is stored in a MariaDB [5] database. Instead of creating the relations with pure MySQL, the Laravel Framework offers a convenient method to define and handle them. By creating Migration Files in PHP, the relations can not only be created, but can also be version controlled, so several developers can upgrade their developing machine easily to the newest schema. In general, some properties were kept consistent. The primary key, for example, is always an unsigned integer called ID and is given in every relation. Also, the times of the creation and the times of the last update are saved for every relation, that is why they are not shown in the figures. Like many other parts on the API, the database has four big parts: Accounts, Questionnaires, Contributions and Interventions.

Accounts

Regarding the system role, an account can be either patient, therapist or researcher. Some of the functions though have to be accessible for every account in the system, shown in Figure A.8. The essential part is the relation “Users”, containing not only the user’s email, with which he logs into the system, but also his password. Further, the user’s rights are shown, distinguishing between normal user, therapist or researcher. To provide entries in the user’s timeline, the Notifications relation can offer a link, a name and a description. On top of that, the boolean “clicked“ shows whether an entry has been seen yet, or not. Also, Password Reset- and Email Verification entries have to be accessible by every user.

Furthermore, there are some aspects that are only for the patient, described in Figure A.9. Not only Appointments, but also Assignments for Questionnaires are shown, as well as the patient’s Therapy, which contains the personal data.

To offer more information to the public, therapists can store address, gender and more in their Therapist Data, Figure A.10. A Therapist Code allows patients to connect with the therapist.

Questionnaires

Because of the fact that the questionnaires all have the same structure, the relations for them are quite easy to understand. The only part that varies is the number of given answers. As an example, see Figure A.1.

Figure A.12 contains the questionnaires, which have to be sorted into the four groups of Table 2.1. They all contain one or more sections, which have an optional description and can be ordered. Each section contains a set of questions and answers, both can be sorted, too.

Contributions

Because of the constant structure of the questionnaires, the contributions can be modelled straight forward. As seen in Figure A.13, one Contribution has a Foreign Key to the user, that contributed and to the Questionnaire. In addition, every contribution has Contribution Answers, each with a Foreign Key to the question and its answer.

Interventions

Because of the predefined set of Intervention Types, only a possibility of connecting it to a user at a certain time had to be implemented. As presented in Figure A.11, this includes a period of time as well as a processing status.

Seeding Data

It is easier to predict the API's behaviour, when there is as much data as there will be when it is used by real patients, that's why a big set of seeding data is provided. Before filling the tables, T. Gablonski estimated the amounts of users, so realistic information of for example users, questionnaires and appointments could be saved. By using the Faker Library [2], the data was random and did not have to be written manually. Another advantage was that the data could be reset and set again with ease. Also, writing tests was effortless, because there are already filled tables to work with.

5.1.2 Architecture and Classes

In general, Laravel ensures that a certain HTTP call is mapped to a specific controller method. This provides a regulated process, passes defined middleware and prevents confusion. The basic arrangement of classes is mostly given by the Laravel Framework. Basically, database, the app itself, configurations, tests and routes are separated. Every aspect is sorted in packages, named after the relating model, for example appointment, account or notification. While most of the bullets do not need further explanation, the app package is not as self-explanatory. It contains the following:

Models Every relation is defined as a model, implemented as php class. With the help of the built-in Object Relational Mapper (ORM), every database change can be done by simply changing the related model. Also, foreign key constraints are implemented, so that other models can be selected from the database without any SQL code.

Middlewares Contain methods, which may or may not be called before a controller is reached. For example, it is used to check for authentication or missing arguments.

Requests To keep track of the validation rules, every HTTP call is named as specific request. Also, validation error messages can be changed, as well as guards to check for the right authorization.

Controllers In the controllers, the actual logic is programmed. Because of the defined requests and resources, only the algorithm itself is defined in them, what makes the code clear and readable.

Resources By defining a consistent structure of the resources, it is possible to maintain the same build for every answer to a specific HTTP call.

As you can see, the process inside Laravel is well-defined. By keeping up this schedule, maintenance is easy and the code itself is transparent.

5.2 Programming Insights

Once the database relations were set up, most of the API's elements were quite feasible. Some aspects however had to be thought through more than others, because either they needed more complex algorithms or they had to be very reliable, because the patients get medical advices through them. In the following, some examples are shown to get an view inside of the project.

5.2.1 Functionality and Difficulties

In short, it was possible to implement every requirement described in Chapter 3. Therefore, a full account system, containing patients, researchers and a way of connecting them is available. Questionnaires are stored in the database and provided to therapists and researchers. On top of that, they are assigned to the patients regarding the rules described in Table 2.1. Appointments can be created and deleted, which is an important aspect, because some of the questionnaire assignments depend on them. After getting an assignment, patients can store the filled out questionnaire as contribution in the system. Once a week, the contributions are evaluated and interventions are assigned to the users.

The steady changes of how an implementation part should be provided to the end user has been the most difficult part. It was hard to offer both, the new functions planned for the current iteration and the last functions accommodated to the new requirements. As a result, some structures may seem to be contra-intuitive. As an

example, this effect can be seen in the database structure. While only taking into account the user relations in Figure A.10 and Figure A.9, it may seem that it is possible for a user to be both, therapist and patient. This is in case a remnant from a customer request, that should enable a patient's view to the therapists and researchers. It is not possible in the current system.

5.2.2 Pairing of Patient and Therapist

When the data is exported in a later stage of the study, it is not allowed to match contributed data with personal information of any patient. Nevertheless, gender and date of birth are important for the evaluation, so they have to be linked to a therapy. To enable that kind of anonymisation, a special pairing process had to be established. Until now, every patient has a code, which is not unique. It is formed by taking the first letter of the last name and the date of birth as simple numbers. For example, if a John Doe is born on July the 1st, 1955, his code would be "D010755".

In the system, this approach is continued. To simplify this aspect for the patient though, the therapists keep in track of the code creation. By connecting the code assignment with the pairing of patient and therapist, these steps are combined and the effort is halved. In detail, there is a relation in the database called "Therapist-Codes". Each code belongs to the therapist that creates it. The Therapist Code itself is a unique random pairing ID between 10000000 and 99999999. When calling the URI with a post to `"/api/therapistcode/"` containing a code of the described form, it creates a new entry in the database. To prevent confusion, the model's structure is illustrated in Listing 5.1.

By using the created random pairing ID, the patient can pair with the therapist and the code is assigned to the patient in one step.

Listing 5.1: Therapist Code structure

```
1 {
2   "id": 14,
3   "pairing_id": "10000012",
4   "codename": "C081104"
5 },
6 {
7   "id": 17,
8   "pairing_id": "10000015",
9   "codename": "G240511"
10 }
```

To keep it simple, the pairing ID is not changeable, neither is the code for the patient. Therefore, Therapist Codes are only creatable and deletable. Of course, only accounts, that are marked as therapist can manage Therapist Codes.

5.2.3 Intervention Assignments

One way of using the collected data is to distribute specific questions or exercises for the patients. This is an automatic process, that automatically evaluates the contributed questionnaires. Important for that part is the questionnaire “Intersession Fragebogen kurz (ISF-K)” [17] and “Symptom Checkliste K9” [19]. Being in Group 3, both have to be filled out once over the weekend, further information can be found in Table 2.1. Before implementing, a list of interventions and when they are used was given by T. Gablonski, so in fact the largest part was to evaluate the contributions and assign an intervention to the patient according to the given list. Since this is a scheduled task, it was possible to use the pre-implemented tool “Task Scheduling” [4] by Laravel. For that, the Framework uses a cron job, which is described as “command to an operating system or server for a job that is to be executed at a specified time” [7] by the Oxford Dictionary. While this task is called every minute, Laravel simply uses the exact current server time to define, which schedule is performed. This makes it possible to perform a task exactly once a week, at the same minute, which is used to assign the interventions automatically, at the same time once a week.

For the algorithm itself, some preconditions regarding the therapy and the contributed questionnaires are given, they are described in Table 5.1.

Those conditions, as well as the validation of the contributions, are double checked

Precondition	Explanation
Patient	The user has to be registered in the system and without therapist or researcher rights.
Therapy	The patient has to be within a therapy, meaning that his therapy data contains a date for the start, but not for the end.
Contributions	Both questionnaires required, "ISF-K" and "SCL-K9", have to be contributed the weekend before.

Table 5.1: Preconditions of the Intervention Assignment Algorithm.

before the actual evaluation starts. To keep it structured, every step is separated in an own function. The steps are as follows.

assignInterventionToUser Is used as a wrapper for the called functions and throws the corresponding exception if one of the used functions fails. First, it gets the latest needed contributions and validates them, then it evaluates them by calling the function "calculateSciMeanValue", "calculatelsfValue" and "calculateHigherItem" and assigns the correct intervention with the help of the function "createAssignmentFromScores".

calculateSciMeanValue Given a contribution and the questionnaire SCL-K9, to which the contribution belongs, it calculates the average of the answer's values and returns it. It is shown in Listing B.2.

calculatelsfValue Also using the contribution and the right questionnaire, it calculates the sum of the answer's values and returns it, as shown in Listing B.3.

calculateHigherItem Checks, if Item 7 or Item 8 of a questionnaire is higher.

createAssignmentFromScores Finally, the appropriate intervention is chosen and is assigned to the user at a random time between the time it is chosen and the next appointment with the therapist.

Finally, the outcome is tracked and emailed to all researchers. Because of privacy reasons, neither the chosen intervention, nor any personal information of the patients are sent. The sole purpose of the mail is to call attention to the exceptions that occur in the system.

6 Usage

As mentioned, the API has its own documentation 4.1.3, so the usage is quite straightforward. Nevertheless, some characteristics deserve to be mentioned. There are already mobile applications and a web interface in use, further information is given in Section 6.2.

6.1 Using the Application Interface as Developer

The interface is reachable online, deployed on the servers of the Ulm University. To get proper results, the Hypertext Transfer Protocol should be used.

6.1.1 HTTP Requests

Principally, the API is accessible by the most common HTTP requests, GET, POST, PUT and DELETE. The implementation can only handle the JSON format, that's why an accept header with the content "application/json" is recommended. Not only are all responses designed that way, but also the request body is expected to be in the same format, so a Content-Type header containing "application/json" is required for the best user experience as well.

The API documentation 4.1.3 has listed every possible request, each with the needed request bodies and the possible response codes, especially the errors that can occur. The most common status responses are:

- 200/201** General "ok" status code, is mostly used, when a request was successful.
- 401** Is used, when a user is unauthorized to perform the action.
- 422** If a validation rule is violated, a response with the status code 422 is sent, containing the parameter that is not compatible.
- 500** In case of an error that can not be defined, the Laravel Framework uses the code 500.

6.2 Current Usage Examples

Alongside the API, three different systems that use the interface have been developed. This is mainly because of the separation of patients, therapists and researchers. The Mobile Applications are only used by patients, while the Web Interface is for the other system roles.

6.2.1 Web Interface

In the system, therapists and researchers mostly analyse user data or keep track of their patient's contributions and diagrams. Because of that, a mobile application is not needed, it would simply be too small to recognize the important parts. As a solution, a web interface was implemented, containing account management, overview of the patients and their contributions and the other needed functions. Some insights of the Therapist Code creation can be seen in Figure A.15, a preview of the diagrams is shown in Figure A.14.

6.2.2 Mobile Applications

The main part for the patients is, as often mentioned, the notifications they get to their phones to fill out the questionnaire. Also, interventions work according the same principle. Those are the main functions the user can get, except for small micro actions, such as pairing with the therapist. Since those two main actions only work on a mobile device, no other possibility is needed, every function works with the mobile application. Both, android and iOS have been implemented natively and the results can be seen in Figure A.16 and A.17.

7 Conclusion

The goal of this thesis was to develop an application programming interface, which can be used to investigate intersession processes. In the course of a whole software engineering lifecycle, this goal could be reached. After creating a first concept, it was elaborated until every stakeholder was pleased. The implementation itself was a lot of work, but the outcome was a success, since T. Gablonski is pleased with it and the interface will be deployed and used on patients. The written tests also confirm, that the behaviour of the implementation is as it should be. Both groups, the creators of the Mobile Applications, as well as the developers of the Web Interface were able to integrate the API, which proves the usability. Of course, there have been problems in many steps, but all in all I think not only the outcome, but also the development itself was quite satisfying.

The next steps include the collection of data. Since this was a work from scratch, there are no results yet. Soon, the system will be used and patients of the Alps-Adria-University Klagenfurt contribute their data. By granting access to researchers, knowledge of better treatments can be achieved. Since the intersession processes are documented precisely, the influence of it to the therapy outcome can be investigated. Furthermore, the basic structure of the project can be used for many other fields of use. Physiotherapy, for example, often keeps track of the patient's pain. By making use of daily questionnaires, this project could easily be adjusted to the new needs.

Personally, I learned a lot regarding APIs and applications in the medical sector. While I was familiar with the knowledge of creating interfaces, database relations and the coding standards, I underestimated the workload regarding formal obligations. So for me, creating the platform was a successful and educational project.

A Pictures and Screenshots

<i>Platzhalter Logo</i>	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.

⊖ = gar nicht ⊕ = selten ⊗ = manchmal ⊙ = oft ⊚ = sehr oft

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

Figure A.1: Intersession-Fragebogen Kurzversion (ISF-K)



SCL-K-9

Im Nachfolgenden finden Sie eine Liste von Problemen und Beschwerden, die man manchmal hat. Bitte lesen Sie die einzelnen Beschwerden sorgfältig durch und entscheiden Sie, wie Sie seit der letzten Einzel-Therapiesitzung durch diese Beschwerden gestört oder bedrängt worden sind. Überlegen Sie bitte nicht erst, welche Antwort den „besten Eindruck“ machen könnte, sondern antworten Sie so, wie es für Sie persönlich zutrifft.

überhaupt nicht	ein wenig	ziemlich	stark	sehr stark
0	1	2	3	4

Wie sehr litten Sie in den letzten sieben Tagen unter ...						
1	... Gefühlsausbrüchen, gegenüber denen Sie machtlos waren?	0	1	2	3	4
2	... dem Gefühl, dass es Ihnen schwer fällt etwas anzufangen?	0	1	2	3	4
3	... dem Gefühl, sich zu viele Sorgen machen zu müssen?	0	1	2	3	4
4	... Verletzlichkeit in Gefühlsdingen?	0	1	2	3	4
5	... dem Gefühl, dass andere Sie beobachten oder über Sie reden?	0	1	2	3	4
6	... dem Gefühl, gespannt oder aufgeregt zu sein?	0	1	2	3	4
7	... Schweregefühl in den Armen oder den Beinen?	0	1	2	3	4
8	... Nervosität, wenn Sie alleine gelassen werden?	0	1	2	3	4
9	... unter Einsamkeitsgefühlen, selbst wenn Sie in Gesellschaft sind?	0	1	2	3	4

Figure A.2: SCL-K9

therapist_codes Creating and using therapist_codes to pair with a therapist ∨

- GET** /api/therapistcode Gets all therapist_codes 🔒
- POST** /api/therapistcode Saves a new therapist_code 🔒
- GET** /api/therapistcode/{pairing_id} Gets the status of a therapycode 🔒
- PUT** /api/therapistcode/{pairing_id} uses the therapistcode to pair with a therapist 🔒
- DELETE** /api/therapistcode/{pairing_id} deletes a therapistcode 🔒
- GET** /api/therapistcode/{pairing_id}/pdf Gets a pdf with the pairing_id and the codename 🔒

notifications A notification is a message for the user, to see when something new accures >

appointments an appointment is a meeting with the therapist ∨

- GET** /api/appointment Gets appointments of the logged in User 🔒
- POST** /api/appointment Saves an appointment into the system 🔒
- PUT** /api/appointment/{id} Changes an appointment 🔒
- DELETE** /api/appointment/{id} Deletes an appointment 🔒

Figure A.3: Example of Swagger-UI

The screenshot displays a Swagger API resource for a POST endpoint. At the top, the method is labeled 'POST' and the path is '/api/therapistcode', with a description 'Saves a new therapist_code'. Below this, a note states: 'Given a codename, it saves a new therapist_code with a random pairing_id that belongs to the logged in therapist'. The 'Parameters' section is empty, with a 'Try it out' button. The 'Request body' section is marked as 'required' and has a dropdown menu set to 'application/json'. An 'Example Value' is shown as a JSON object:

```
{  "codename": "A210591"}
```

. The 'Responses' section is a table with columns for 'Code', 'Description', and 'Links'. It lists two responses: a 200 status with the description 'Successfully created' and 'No links', and a 401 status with the description 'Access token is missing or invalid' and 'No links'. Both response entries have a dropdown menu set to 'application/json'. The 200 response also includes a note 'Controls Accept header.' and an 'Example Value' showing a JSON object with 'message', 'codename', and 'pairing_id' fields.

Code	Description	Links
200	Successfully created	No links
401	Access token is missing or invalid	No links

Figure A.4: Overview of a Swagger-Resource

Pipeline	Status	Started	Duration
#227  contribution with details Julian Huber  6516e11  master	 Successful	10 minutes ago	1 min 4 sec
#226  api documentation for the contribution filter Julian Huber  30919a6  master	 Successful	26 minutes ago	1 min 27 sec
#225  api documentation for the contribution filter Julian Huber  9891105  master	 Failed	7 days ago	53 sec
#224  add questionnaire_id filter for getting contributions Julian Huber  6366a83  master	 Successful	7 days ago	1 min 47 sec
#223  Swagger documentation for interventions (get all, seen and fulfilled) Julian Huber  c3aae6c  master	 Successful	14 days ago	1 min 7 sec
#222  Intervention markable as seen and fulfilled Julian Huber  9b809d9  master	 Failed	14 days ago	1 min 17 sec
#221  interventions seeder Julian Huber  8f3137a  master	 Successful	15 days ago	1 min 16 sec
#220  fix notification in case of no therapist found Julian Huber  2fc0ca3  master	 Successful	16 days ago	1 min 12 sec
#219  Add notification for therapist, when his patient deletes his account Julian Huber  2d39f16  master	 Failed	16 days ago	58 sec
#218  fix therapies, so they start at the actual time, not some time in the ... Julian Huber  982ad0a  master	 Successful	16 days ago	2 min 21 sec

< **1** 2 3 4 5 ... 23 >

Figure A.5: Bitbucket Pipelines

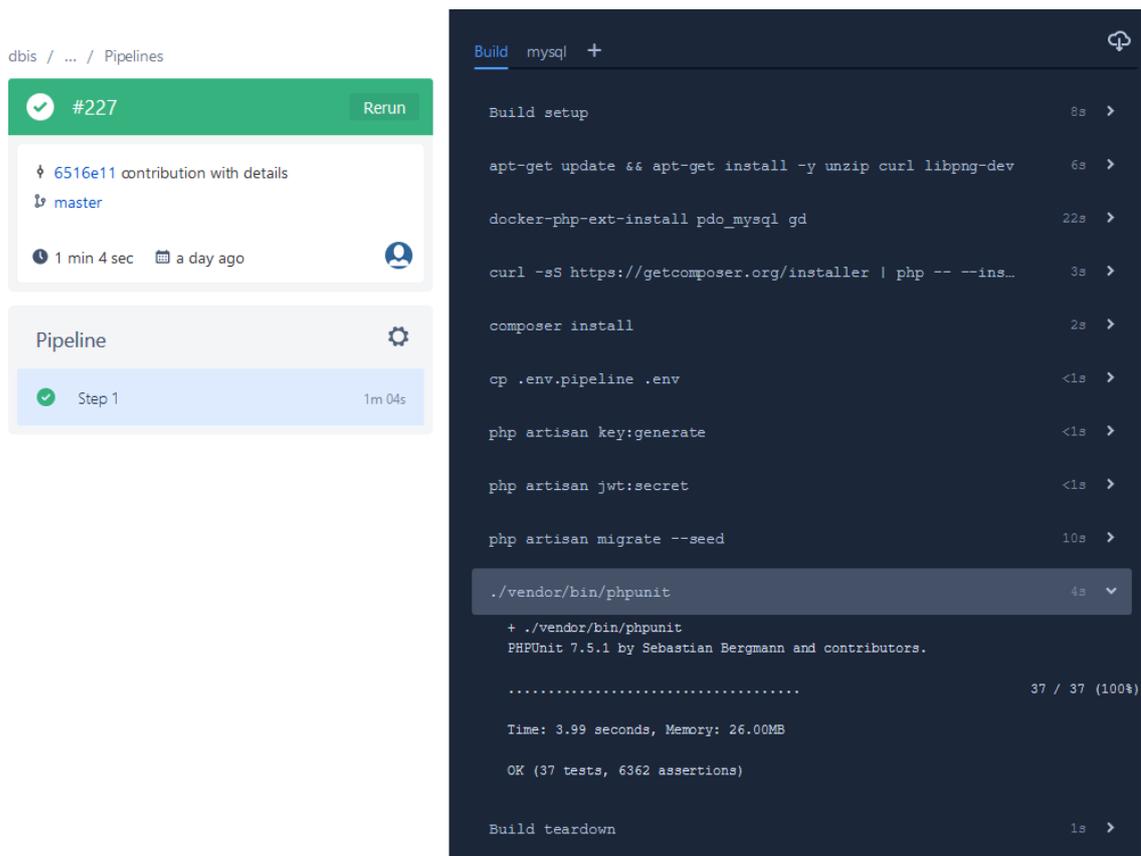


Figure A.6: Successful Bitbucket Pipeline

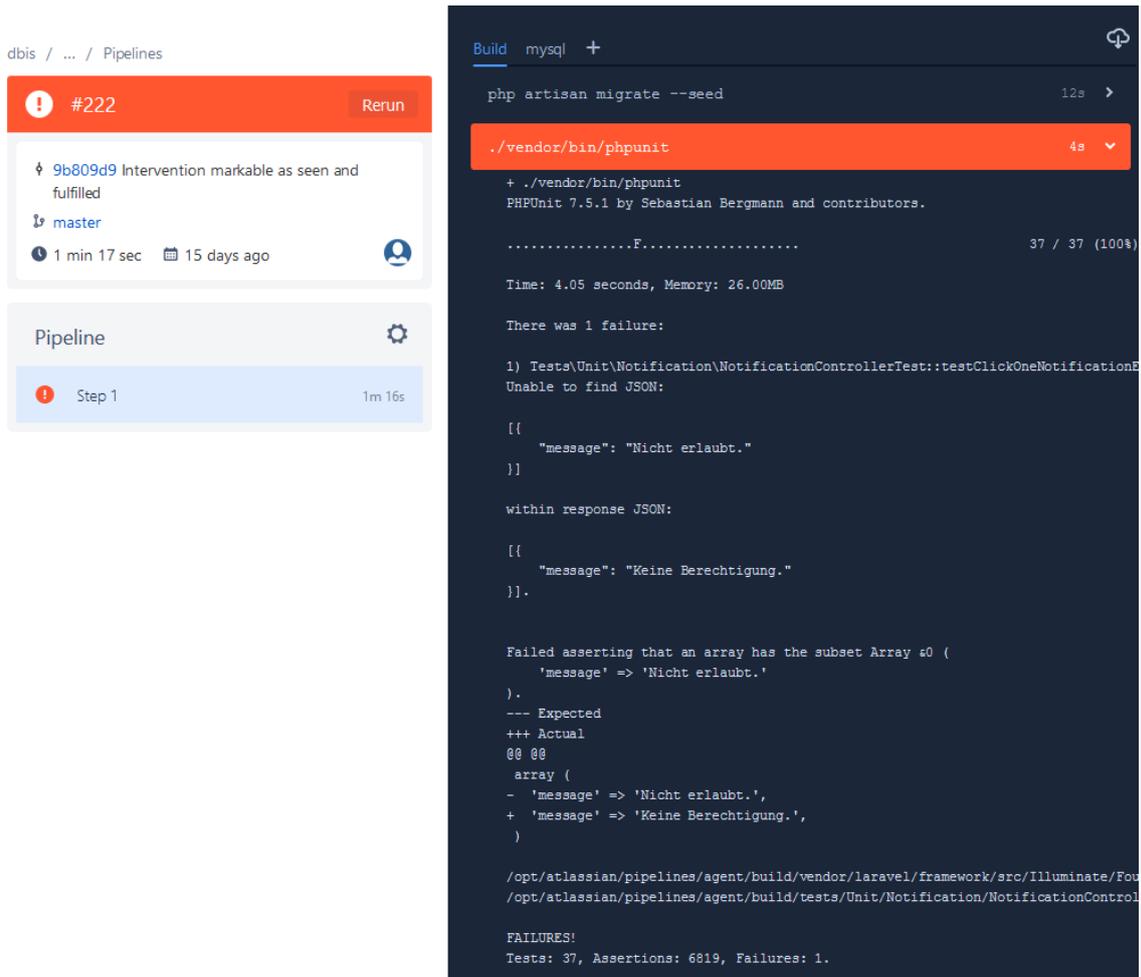


Figure A.7: Failed Bitbucket Pipeline



Figure A.8: User database, general

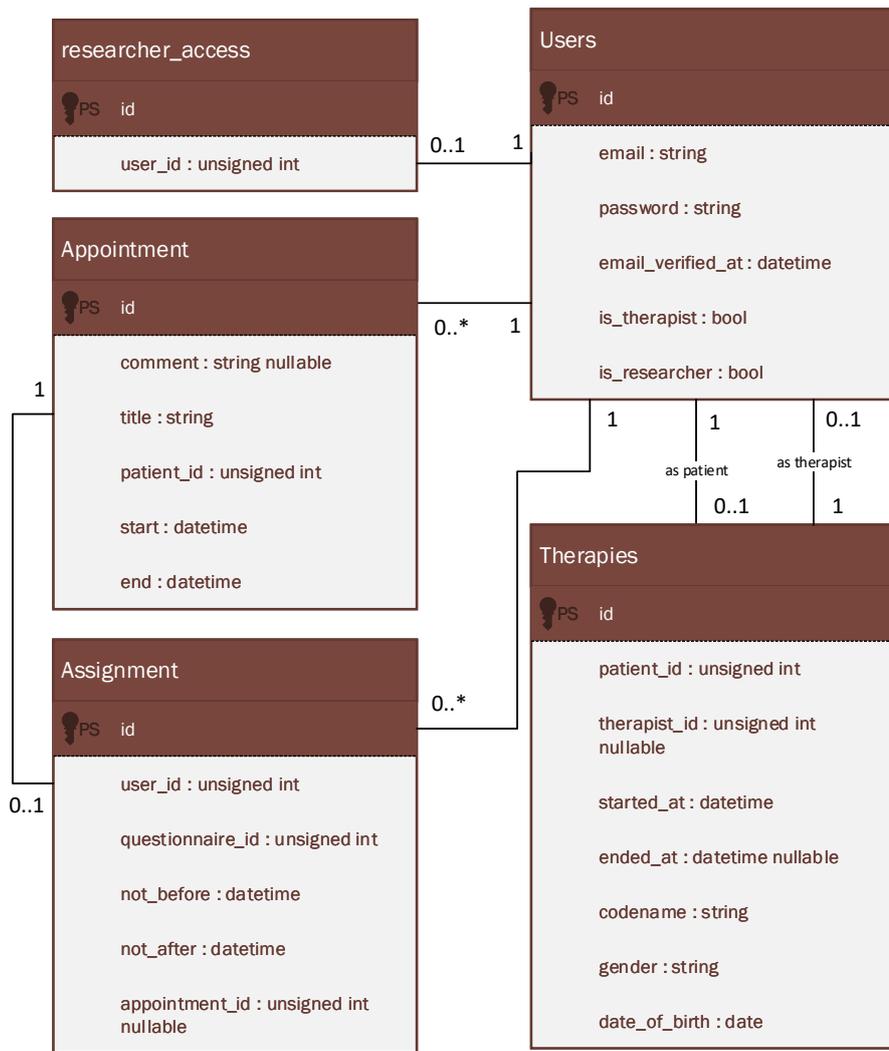


Figure A.9: User database, patients

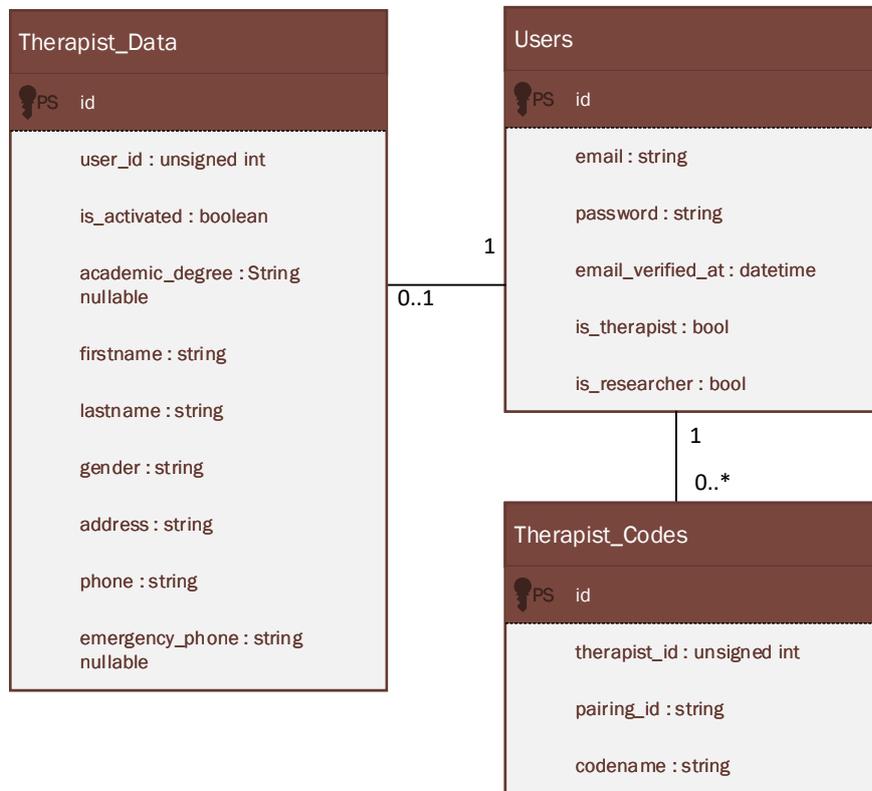


Figure A.10: User database, therapists

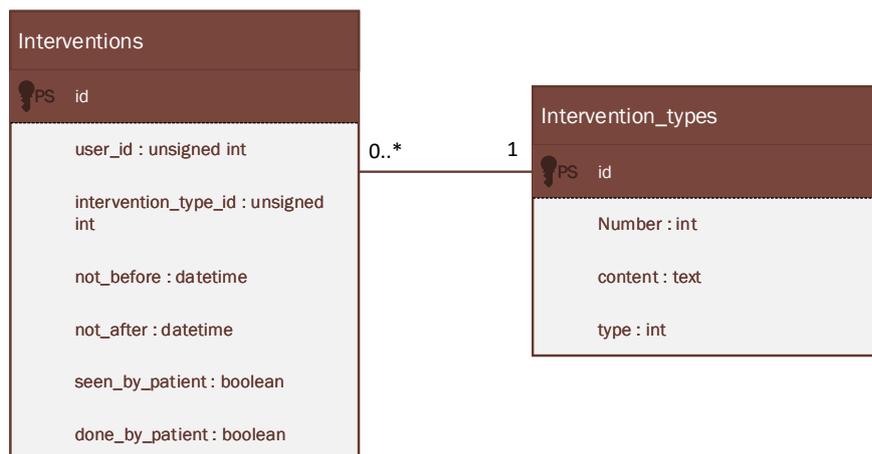


Figure A.11: Intervention database

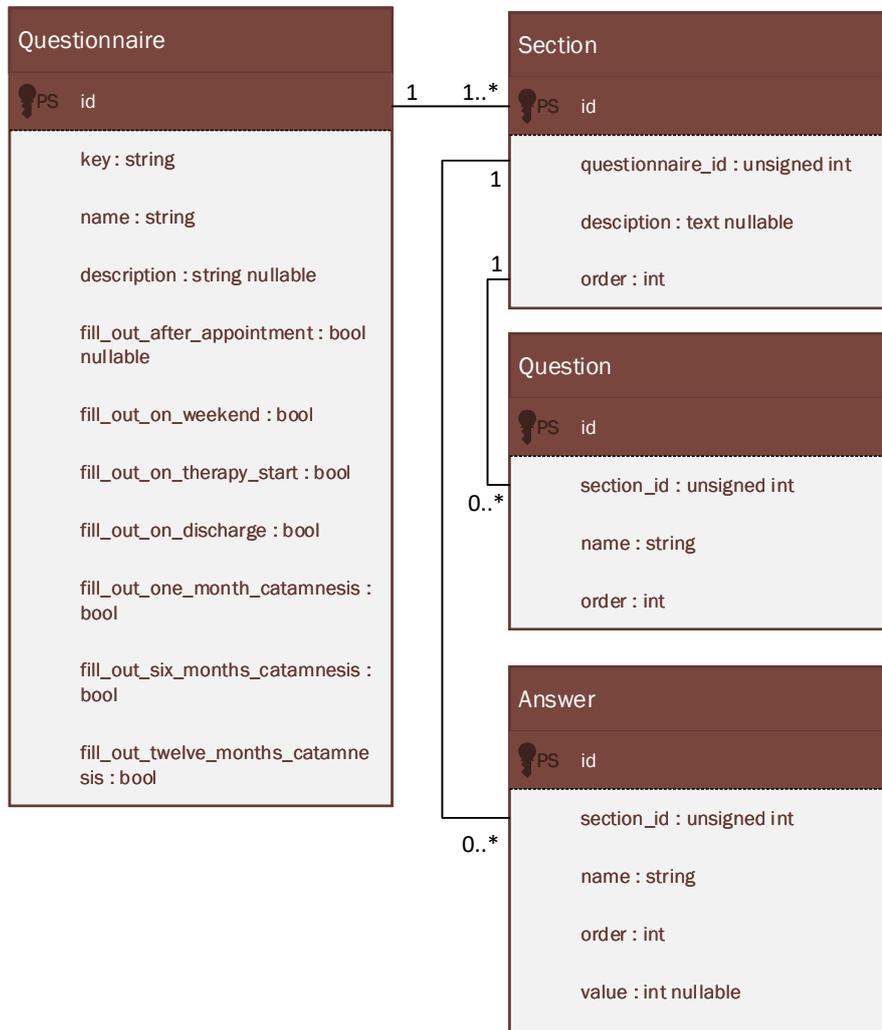


Figure A.12: Questionnaire database

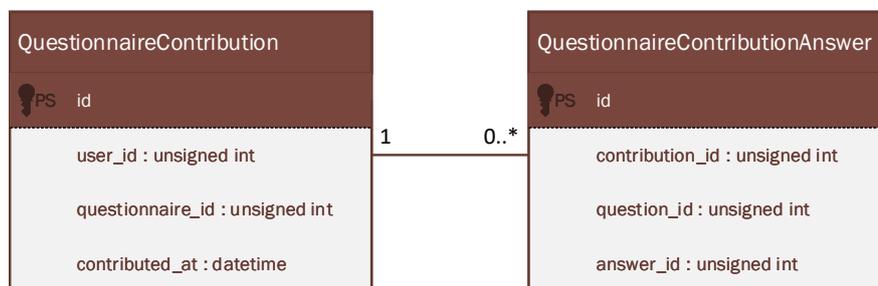


Figure A.13: Contribution database

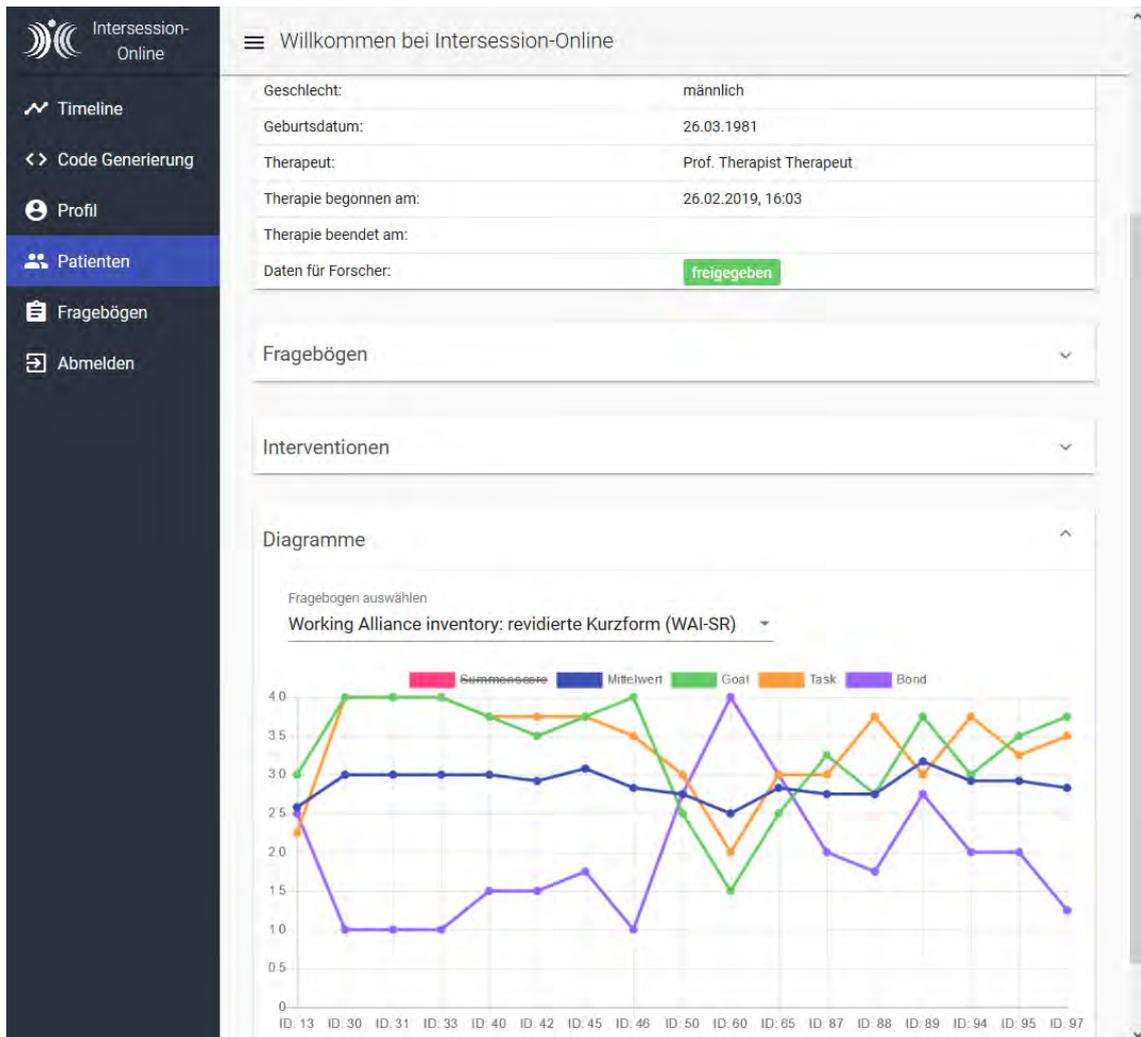


Figure A.14: Diagrams of contributed questionnaire answers [23].

Intersession-Online

☰ Willkommen bei Intersession-Online

↶ Timeline
↷ Code Generierung

👤 Profil
👥 Patienten
📄 Fragebögen
🚪 Abmelden

↷ Hier finden Sie alle Informationen zum Pairing-Prozess mit Ihren Patientinnen und Patienten.

Damit sich Ihre Patientin / Ihr Patient mit Ihnen via Intersession-Online verbinden kann, ist es notwendig einen Code zu generieren. Zur Code-Generierung benötigen Sie den PatientInnen-Code, der sich wie folgt zusammensetzt:

Erster Buchstabe des Nachnamens + das Geburtsdatum der/des PatientIn.

Hier ist ein Beispiel für Max Mustermann, geboren am 01.08.1977: M010877

Geben Sie in die untere Zeile den PatientInnen-Code ein und klicken Sie auf „Code generieren“. In der darunterliegenden Übersicht können Sie den PatientInnen-Code als PDF-Datei abrufen. Bitte fragen sie Ihre/n PatientIn, ob diese/r ein Android- (Symbol) oder iPhone-Smartphone (Symbol) verwendet und klicken Sie auf das entsprechende Symbol.

Bitte drucken Sie die PDF-Datei anschließend aus und geben Sie diese der/dem PatientIn mit. Dort findet die/der PatientIn eine ausführliche Schritt-für-Schritt-Erklärung zum Pairing-Prozess.

Sobald sich ein/e PatientIn mit Ihnen verbunden hat, verschwindet der generierte Code aus der unteren Liste.

Patientencode eingeben

T161193

PDF herunterladen:

Alternativ können Sie die PDF-Datei auch jederzeit über die unten aufgeführte Liste herunterladen.

Code generieren

Pairing-Codes anzeigen ^

Patienten-Codename	Pairing-Code	PDF-Download		
J200905	10000010			
C081104	Therapeuten-Code wurde erstellt. OK			

Figure A.15: Creation of a Therapist Code [23].



Figure A.16: Calendar and Exercise functions in the Android Application [16].

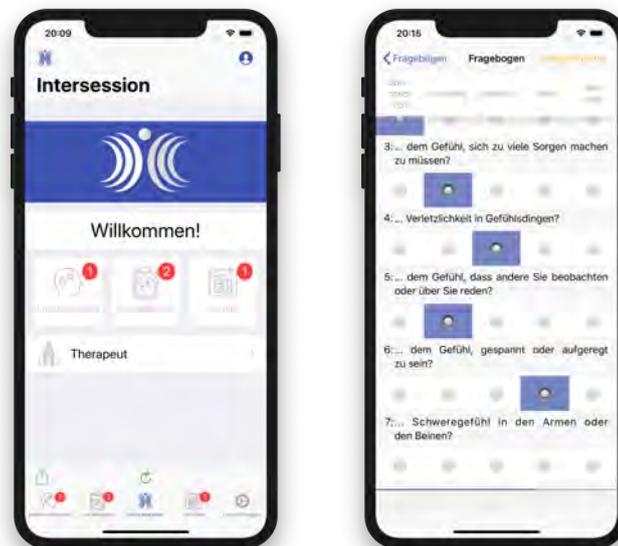


Figure A.17: Overview and questionnaire in the iOS Application [30].

B Code Snippets

Listing B.1: Bitbucket Pipeline

```
1 image: php:7.2-fpm
2
3 pipelines:
4   default:
5     - step:
6       caches:
7         - composer
8       script:
9         - apt-get update && apt-get install -y unzip
10        curl libpng-dev
11        - docker-php-ext-install pdo_mysql gd
12        - curl -sS https://getcomposer.org/installer
13        | php -- --install-dir=/usr/local/bin
14        --filename=composer
15        - composer install
16        - cp .env.pipeline .env
17        - php artisan key:generate
18        - php artisan jwt:secret
19        - php artisan migrate --seed
20        - ./vendor/bin/phpunit
21   services:
22     - mysql
23
24   definitions:
25     services:
26       mysql:
27         image: mysql:5.7
28         environment:
29           MYSQL_RANDOM_ROOT_PASSWORD: 'yes'
30           MYSQL_ALLOW_EMPTY_PASSWORD: 'yes'
31           MYSQL_DATABASE: 'intersession'
32           MYSQL_USER: 'intersessiondb'
33           MYSQL_PASSWORD: 'test'
```

Listing B.2: Calculation of the SCL-K9 value

```
1  /**
2  * Calculates the score of the SCL-k-9 by calculating
3  * the average of the chosen answer-values
4  *
5  * @param Contribution $contribution
6  * @param Questionnaire $scl_k9
7  * @return float|int
8  */
9  private function calculateSclMeanValue
10     (Contribution $contribution, Questionnaire $scl_k9)
11  {
12     //Check if the questionnaire is valid
13     $this->validateContribution($contribution, $scl_k9);
14
15     //SCL Score is calculated by getting the average points.
16     //The value can be between 0 and 4.
17     $contributionAnswers = $contribution->contributionAnswers()->get();
18
19     //count the answers and make the sum of the values to get the average
20     $answerCount = 0;
21     $valueSum = 0;
22     foreach ($contributionAnswers as $contributionAnswer) {
23         try {
24             $chosenAnswer = $contributionAnswer->answer()->firstOrFail();
25             $answerCount++;
26             $valueSum += $chosenAnswer->value;
27         } catch (ModelNotFoundException $exception) {
28             throw new Exception("Eine Antwort konnte nicht gefunden werden.");
29         }
30     }
31     return ($valueSum / $answerCount);
32 }
```

Listing B.3: Calculation of the ISF-K value

```
1 /**
2  * Calculates the score of the ISF-K by calculating
3  * the sum of the chosen answer-values
4  *
5  * @param Contribution $contribution
6  * @param $isf_k
7  * @return int
8  */
9 private function calculateIsfValue(Contribution $contribution, $isf_k)
10 {
11     //Check if the questionnaire is valid
12     $this->validateContribution($contribution, $isf_k);
13
14     //ISF Score is calculated by getting the sum.
15     $contributionAnswers = $contribution->contributionAnswers()->get();
16
17     // make the sum of the values to get the score
18     $valueSum = 0;
19     foreach ($contributionAnswers as $contributionAnswer) {
20         try {
21             $chosenAnswer = $contributionAnswer->answer()->firstOrFail();
22             $valueSum += $chosenAnswer->value;
23         } catch (ModelNotFoundException $exception) {
24             throw new Exception("Eine Antwort konnte nicht gefunden werden.");
25         }
26     }
27     return $valueSum;
28 }
```


Bibliography

- [1] *Centers for Medicare & Medicaid Services*. <https://www.cms.gov/>, . – accessed: March 3rd, 2019.
- [2] *Faker PHP Library*. <https://github.com/fzaninotto/Faker>, . – accessed: March 15th, 2019.
- [3] *Laravel PHP Framework*. <https://laravel.com/>, . – accessed: March 12th, 2019
- [4] *Laravel Task Scheduling*. <https://laravel.com/docs/5.8/scheduling>, . – accessed: March 17th, 2019
- [5] *MariaDB*. <https://mariadb.org/>, . – accessed: March 14th, 2019.
- [6] *MobileTx*. <https://www.uni-ulm.de/in/iui-dbis/forschung/laufende-projekte/mobiletx/>, . – accessed: March 20th, 2019
- [7] *Oxford Dictionary for cron*. <https://en.oxforddictionaries.com/definition/cron>, . – accessed: March 17th, 2019
- [8] *QuestionSys - A Generic and Flexible Questionnaire System Enabling Process-Driven Mobile Data Collection*. <https://www.uni-ulm.de/in/iui-dbis/forschung/laufende-projekte/questionsys/>, . – accessed: March 20th, 2019
- [9] *Selecting a development approach*. <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>, . – accessed: March 3rd, 2019.
- [10] *Swagger-UI*. <https://swagger.io/tools/swagger-ui/>, . – accessed: March 5th, 2019
- [11] ANDREAS, S. ; GABLONSKI, T.-C. ; HIESBERGER, S. ; SENFT, B. ; KOLLER, I. ; SCHULZ, H. : Intersession-Prozesse in stationärer psychotherapeutischer Behandlung: Ist die Einzel- oder die Gruppentherapie entscheidender für die therapeutische Beziehung und das Therapieergebnis? In: *PDP Psychodynamische Psychotherapie* (2016), S. 206–218

- [12] BARTHOLOMEW, K. ; HOROWITZ, L. M.: Attachment styles among young adults: a test of a four-category model. In: *Journal of personality and social psychology* 61 (1991), Nr. 2, S. 226
- [13] BRAY, T. : *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 8259. <http://dx.doi.org/10.17487/RFC8259>. Version: Dez. 2017 (Request for Comments)
- [14] DAMMANN, G. ; ZIMMERMANN, J. ; HÖRZ-SAGSTETTER, S. ; BENECKE, C. : IPO-2001 IPO-16. Inventar der Persönlichkeitsorganisation. In: *Diagnostische Verfahren in der Psychotherapie* (2016), Nr. 3, S. 288–293
- [15] GABLONSKI, T.-C. : *Intersession-online: Entwicklung Einer Smartphone-app Zur Systematischen Erfassung Und Kontrolle Von Intersession-prozessen*. – PhD-exposé
- [16] GESSLER, M. : *Mobile Application for Intersession Processes*. 2019. – Software Engineering Project
- [17] HARTMANN, A. ; ORLINSKY, D. E. ; GELLER, J. D. ; ZEECK, A. : Der Intersession-Fragebogen (ISF)–Ein Instrument zur Erfassung von psychotherapie-relevanten Prozessen zwischen Sitzungen. In: *PPmP-Psychotherapie· Psychosomatik· Medizinische Psychologie* 53 (2003), Nr. 11, S. T67–T75
- [18] HAUSBERG, M. C. ; SCHULZ, H. ; PIEGLER, T. ; HAPPACH, C. G. ; KLÖPPER, M. ; BRÜTT, A. L. ; SAMMET, I. ; ANDREAS, S. : Is a self-rated instrument appropriate to assess mentalization in patients with mental disorders? Development and first validation of the Mentalization Questionnaire (MZQ). In: *Psychotherapy Research* 22 (2012), Nr. 6, S. 699–709. <http://dx.doi.org/10.1080/10503307.2012.709325>. – DOI 10.1080/10503307.2012.709325. – PMID: 22867004
- [19] KLAGHOFER, R. ; BRÄHLER, E. : Konstruktion und Teststatistische Prüfung einer Kurzform der SCL-90–R. / Construction and test statistical evaluation of a short version of the SCL-90–R. In: *Zeitschrift für Klinische Psychologie, Psychiatrie und Psychotherapie* 49 (2000), 11, S. 115–124
- [20] LUPTON, D. : Apps as Artefacts: Towards a Critical Perspective on Mobile Health and Medical Apps. In: *Societies* 4 (2014), Nr. 4, 606–622. <http://dx.doi.org/10.3390/soc4040606>. – DOI 10.3390/soc4040606. – ISSN 2075–4698
- [21] PRYSS, R. ; SCHOBEL, J. ; REICHERT, M. : Requirements for a Flexible and Generic API Enabling Mobile Crowdsensing mHealth Applications. In: *4th Int'l Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS), RE'18 Workshops*

- [22] RABUNG, S. ; HARFST, T. ; KAWSKI, S. ; KOCH, U. ; WITTCHEN, H.-U. ; SCHULZ, H. : Psychometrische Überprüfung einer verkürzten Version der "Hamburger Module zur Erfassung allgemeiner Aspekte psychosozialer Gesundheit für die therapeutische Praxis" (HEALTH49). In: *Zeitschrift für Psychosomatische Medizin und Psychotherapie* 55 (2009), 04. <http://dx.doi.org/10.13109/zptm.2009.55.2.162>. – DOI 10.13109/zptm.2009.55.2.162
- [23] RÉKA REJTÖ, V. P.: *Web Application for Intersession Processes*. 2019. – Software Engineering Project
- [24] SAMMET, I. ; LEICHSENRING, F. ; SCHAUENBURG, H. ; ANDREAS, S. : Self-ratings of pathogenic beliefs: A study based on the psychodynamic control-mastery theory. In: *Psychotherapy Research* 17 (2007), Nr. 4, 494-503. <http://dx.doi.org/doi:10.1080/10503300601130854>. – DOI doi:10.1080/10503300601130854. – ISSN 1050–3307
- [25] SCHICKLER, M. ; REICHERT, M. (Hrsg.) ; PROBST, T. (Hrsg.) ; PRYSS, R. (Hrsg.): *Ein Rahmenwerk zur mobilen Unterstützung therapeutischer Interventionen*. <http://dbis.eprints.uni-ulm.de/1714/>. Version: November 2018
- [26] SCHICKLER, M. ; PRYSS, R. ; SCHOBEL, J. ; REICHERT, M. : Supporting Remote Therapeutic Interventions with Mobile Processes. In: *6th IEEE International Conference on AI & Mobile Services (IEEE AIMS 2017)*, IEEE Computer Society Press
- [27] SCHICKLER, M. ; PRYSS, R. ; SCHOBEL, J. ; SCHLEE, W. ; PROBST, T. ; REICHERT, M. : Towards Flexible Remote Therapeutic Interventions. In: *30th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2017)*, IEEE Computer Society Press
- [28] SCHICKLER, M. ; PRYSS, R. ; STACH, M. ; SCHOBEL, J. ; SCHLEE, W. ; PROBST, T. ; LANGGUTH, B. ; REICHERT, M. : An IT Platform Enabling Remote Therapeutic Interventions. In: *30th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2017)*, IEEE Computer Society Press
- [29] 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. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 69–69
- [30] VOGEL, C. : *Conception and realization of a mobile data acquisition and as-*

sistance application for intersession processes of patients in psychotherapeutic treatments at the example of the iOS platform. 2019. – Master Thesis in press

- [31] 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 schulensübergreifendes, ökonomisches und empirisch validiertes Instrument zur Erfassung der therapeutischen Allianz. In: *Klinische Diagnostik und Evaluation* 1 (2008), Nr. 3, S. 343–358
- [32] WINGENFELD, K. ; SPITZER, C. ; MENSEBACH, C. ; GRABE, H. ; HILL, A. ; GAST, U. ; SCHLOSSER, N. ; HÖPP, H. ; BEBLO, T. ; DRIESSEN, M. : Die deutsche Version des Childhood Trauma Questionnaire (CTQ): Erste Befunde zu den psychometrischen Kennwerten. In: *Psychotherapie Psychosomatik Medizinische Psychologie - PSYCHOTHER PSYCHOSOM MED PSYC* 60 (2010), 11. <http://dx.doi.org/10.1055/s-0030-1247564>. – DOI 10.1055/s-0030-1247564
- [33] ZIMMERMANN, J. ; ALTENSTEIN, D. ; KRIEGER, T. ; HOLTFOORTH, M. ; PRETSCH, J. ; ALEXOPOULOS, J. ; SPITZER, C. ; BENECKE, C. ; KRUEGER, R. ; MARKON, K. ; LEISING, D. : The structure and correlates of self-reported DSM-5 maladaptive personality traits: Findings from two german-speaking samples. In: *Journal of Personality Disorders* 28 (2014), Nr. 4, S. 518–540

List of Figures

3.1	Actors and roles of the system	9
4.1	Structure of the unit tests	17
5.1	Logo of the Laravel Framework	19
A.1	Intersession-Fragebogen Kurzversion (ISF-K)	33
A.2	SCL-K9	34
A.3	Example of Swagger-UI	35
A.4	Overview of a Swagger-Resource	36
A.5	Bitbucket Pipelines	37
A.6	Successful Bitbucket Pipeline	38
A.7	Failed Bitbucket Pipeline	39
A.8	User database, general	40
A.9	User database, patients	41
A.10	User database, therapists	42
A.11	Intervention database	42
A.12	Questionnaire database	43
A.13	Contribution database	43
A.14	Diagrams of contributed questionnaire answers [23].	44
A.15	Creation of a Therapist Code [23].	45
A.16	Calendar and Exercise functions in the Android Application [16].	46
A.17	Overview and questionnaire in the iOS Application [30].	46

List of Figures

List of Tables

2.1	Grouping of the needed questionnaires	4
3.1	Overview of actors and roles	8
3.2	General requirements	10
3.3	Patient requirements	11
3.4	Therapist requirements	12
3.5	Researcher requirements	12
5.1	Preconditions of the Intervention Assignment Algorithm.	25

Name: Julian Huber

Student ID: 869019

Declaration

I hereby declare that I have developed and written the enclosed Bachelor Thesis by myself and have not used sources or means without declaration in the text. This Bachelor Thesis has never been used in the same or in a similar version to achieve an academic grading or was published elsewhere.

Ulm,

Julian Huber