physio2go

A Mobile Application for Remote Physiotherapy

Mobile Computing

Afonso Costa (2015231678)
Inês Félix (2015230932)
Pedro Amado (2015246176)
December 2019

Table of Contents

Physio2go: A Mobile Application for Remote Physiotherapy

Mobile Computing

Afonso Costa (2015231678)
Inês Félix (2015230932)
Pedro Amado (2015246176)

Abstract

Physiotherapy is a widely-used treatment for musculoskeletal disorder and limb and joint injuries. Some treatments require sophisticated devices for the procedure, while others only require aided-movement of the body part that is subjected to the treatment. Regarding the latter, we propose a mobile application to perform physiotherapy exercises on-the-go, using only the smartphone sensors and not requiring an active internet connection. This application is entitled *Physio2Go* and the present report is accompanied by the developed Android application. Here we thoroughly describe the planning, implementation and discuss several issues of the proposed physiotherapy framework.

1 Introduction

Physiotherapy is a treatment method that aims to recover the mobility of a limb, based on the science of the human movement. This science, which is entitled as kinesiology, is the scientific study of human (or non-human) body movement, that addresses the restoration of mobility, maintenance or maximisation of physical strength, to achieve a general well-being [1].

The physiotherapy itself encompasses several methods of treatment, from the assisted movement of the affected limb, to more evolved techniques such as electrical muscle stimulation. Furthermore, the rehabilitation sessions are often long and recurring, which sometimes can interfere with certain aspects of the patients life.

It is observable that the degree of injury can vary from total absence of movement to normal limb function, where the injury only presents itself when performing harsh physical activity. The latter case (and similar scenarios) is the one that we aim to address, in which the patient can move normally and has to do light physiotherapy, that does not require a physician or a doctor to be present.

In this report, we describe the development of a mobile application, entitled *Physio2Go*, that addresses the scenario specified above. The following pages will delve further on the development process and are structured as follows: Starting, Section 2 describes the idea and answers the postulated questions. Following, Section 4 provides a thorough description of the project planning and its architecture, including the UI/UX design developed for the application. Next, the Project Management methodology chosen is discussed in Section 3. Then, the implemented exercises are explained on Section 5, followed by the application specifications, in Section 6. Finally, a discussion of the application developed is provided on Section 7, where several security issues are discussed. The present report ends with Section 8, where some conclusions are drawn from this assignment.

2 Application Description

2.1 The idea

We started by brainstorming potential areas of application, where constructing a smartphone app would be beneficial. Since we are students of Biomedical Engineering, we thought it was reasonable to develop an app suitable for the healthcare field and with this in mind we were able to think back to a company, entitled Sword Health [2], that we had contacted with in a Biomedical Summit, some years ago.

This company, which claims to be “The first Digital therapist ever” provides a system that combines digital therapists with physical therapists [2]. The former is emulated by a smartphone or tablet that provides feedback to the user, while performing the prescribed exercises. These exercises are tracked by Motion Trackers, which connect to the Digital therapist (often a tablet) and send the exercise feedback to the physician who prescribes the physiotherapy plan. At this point, we found that it would be an interesting market opportunity to construct a smartphone app where, instead of using separate motion trackers, we could make use of the smartphone sensors, such as the accelerometer. This approach would mostly benefit from removing the need to carry any extra sensors along with the digital equipment. We consider that the smartphone could be strapped to the limb targeted with the exercise, collecting sensorial data and saving it in a server, which would later be analysed by the physician. The name given for the application reinforces that it is meant for physiotherapy and to be used on-the-go.

2.2 The three basic questions

In this section, we answer the three basic questions postulated in the Project Description [3], ensuring that our project respects the assignment guidelines.

Starting with the question “What do you want to create?”, our project is based on a physiotherapy system in which the exercises are prescribed by a physician and the patient is able to visualise and perform those exercises using only the smartphone.

Regarding the second question “What is it good for?”, we argue that it is good to ensure that the patients complete all planned physiotherapy sessions and do not skip any, due to other compromises or given the case of being out of town, where the health care institution is situated. By not skipping any session, the patient could have a quicker recovery from the injury, leading to higher chances of satisfaction from the patient.

Last, for the question: “Who are your potential users?”, we consider that this app is suitable not only for the elderly but also for younger demographics, which are also susceptible to musculoskeletal disorders and are constantly on-the-go. This application would be the most suitable for this scenario since there is no need to carry extra devices, apart from the personal smartphone. Additionally, the application was developed in such a way that there is no need for an active internet connection. The patient can perform the exercises offline and sync with the server later on, when there is an internet connection.

3 Project Management

One concern that we had from start was project management. Prior to any implementation, we established some guidelines and tools, in order to conduct the group in the development part of the project. We employed a simplified version of the Scrum methodology, given that we have a small team and project, for a short period of time. Scrum is an agile process whose goal is to deliver business value in short periods of time and incrementally [4]. Each period of time is a Sprint, which was planned by the group in a meeting. We assigned a member of the group as the Scrum master, who was responsible for creating the Tasks and entrusting them to the scrum team members, according to the sprint planning meeting. Here we defined 4 sprints, for the project development:

- S1 - Planning and implementation study;
- S2 - Coarse Implementation;
- S3 - Test phase and minor tweaks;
- S4 - Write report and submission.

Note that S4 was an ongoing sprint, being that we wrote several parts of the report alongside the application development.

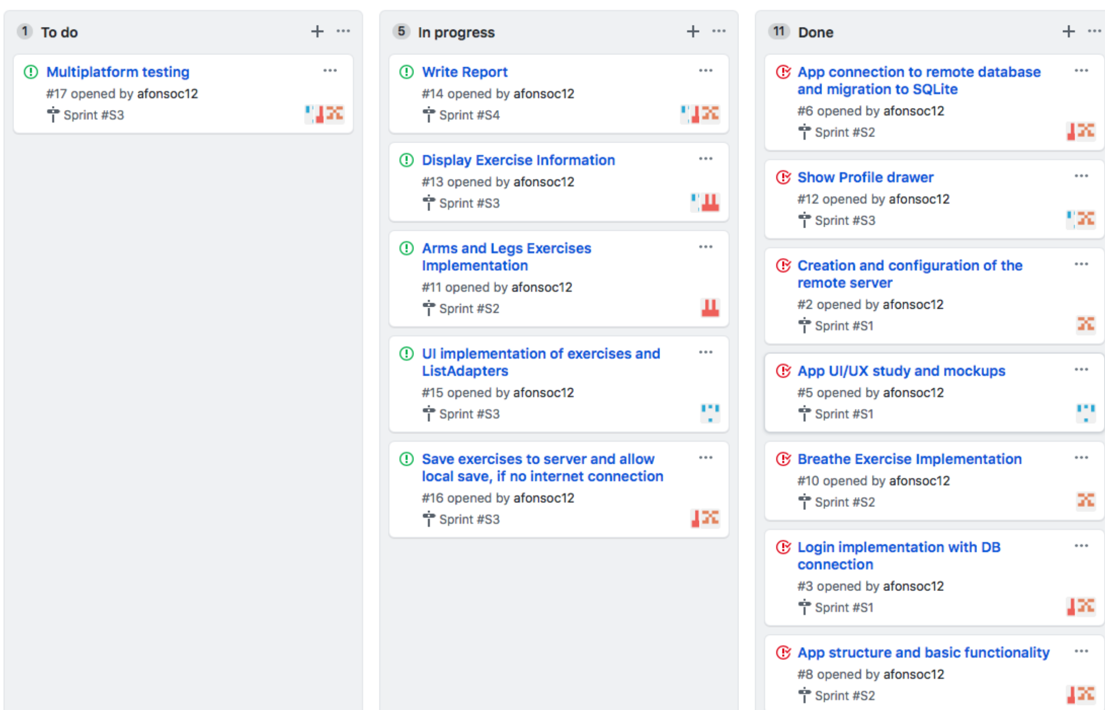

Figure 3.1: Kanban board to track the project status. We defined the standard *to do*, *in progress* and *done* columns.

Regarding the tools for the project management, we resorted to the Github's project management tools for this assignment. We created a repository[1] and a project[2] on this platform. The

[1] https://github.com/afonsoc12/CM_Project
[2] https://github.com/afonsoc12/CM_Project/projects/1

tasks were defined as issue[3] on Github, whereas the Sprints were defined by the Milestones. Each milestone had a deadline and several issues, which were previously defined by the group. To aid the management of the state of each Task, Github offers a Kanban tool, that is crucial to quickly understand the state of the project. A snapshot of the Github's Kanban board is shown on Figure 3.1.

Additionally, we present a Gantt chart, on Figure 3.2. We do not show any previsions on the estimated times of delivery of each sprint, since we adjusted the deadlines according to each team members necessities, to close the assigned tasks. For this reason, we only show the elapsed time of each sprint.

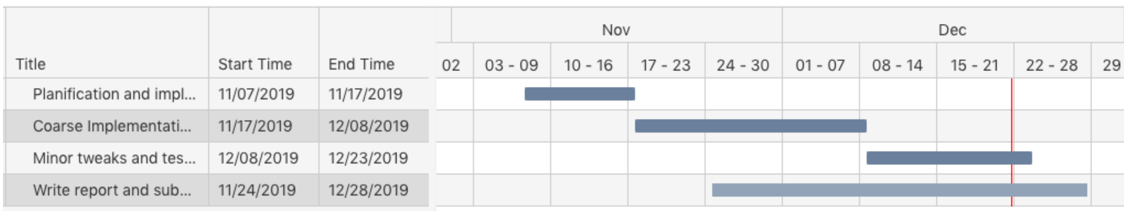

Figure 3.2: Gantt chart that shows the duration of each sprint in the project timeline.

[3] `https://github.com/afonsoc12/CM_Project/issues`

4 Project Planning and Architecture

To develop the proposed application, there were several components that had to be architected and implemented. In this section we thoroughly review each component and provide some description of them.

4.1 Global System Architecture

As mentioned in Section 2.1, we developed a remote-based physiotherapy system. For this, we architected a system which is composed of three distinct groups:

- Client-side mobile application (for the patient);
- Client-side web application (for the doctor);
- A centralised server where both clients connect to retrieve/insert information.

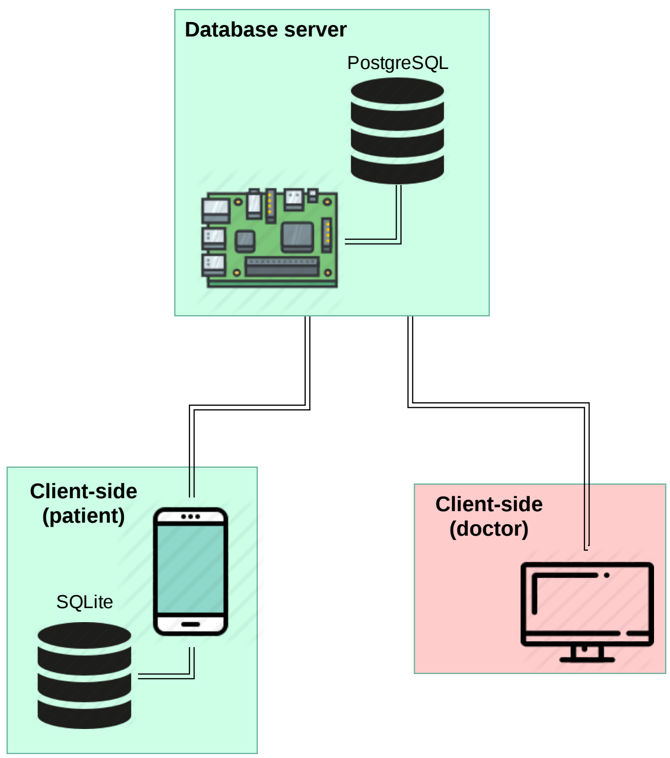

Figure 4.1: Physiotherapy application architecture. The green boxes represent the groups implemented, whereas the red one is the group not implemented (doctor's client-side).

These groups are illustrated in Figure 4.1. Starting with the server, initially, we thought of developing a web application using the Django Framework. In short, Django is a high-level Python Web framework, with a main feature of providing a back-office to a database, that is constructed by the framework, after an Object-Relationl Mapping (ORM) of the defined Python objects. However, due to time constraint-reasons and because this part of the app would reach slightly out of the scope of the present assignment, we decided to simplify and construct a PostgreSQL database (Section 4.2), using only Data Definition Languague (DDL) SQL queries. The not implemented doctor's client-side web app is highlighted in red on Figure 4.1. Still, we argue that this would be a good solution to develop the physician client-side web application.

4.2 Server Database

The database designed is composed of 5 tables. The tables and the respective UML class diagram is illustrated in Figure 4.2.

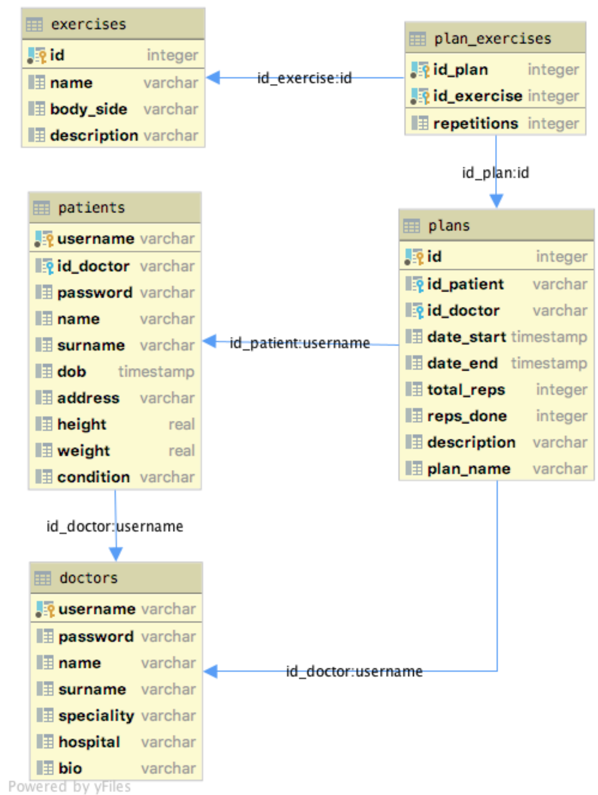

Figure 4.2: Database UML class diagram. It shows the 5 relations, primary keys and foreign keys.

Starting with the *Patients* table, it holds all data from all patients. The primary key is the username. For the reason that each patient only has one doctor, the Doctor's primary key is also included in *Patients* table, as a foreign key. Similarly, the *Doctors* table holds the doctor's information and the primary key is also the respective username. Next, the *Exercises* table comprises all information regarding the implemented exercises, including the body side and its description. Each exercise is identified by a unique primary key. The *Plans* table contains the plans prescribed. It is composed by a unique identifier (primary key) and two foreign keys, that maps to the user (which the plan belongs to) and the doctor who prescribed it. Lastly, the table *Plan exercises* maps the exercises IDs to each plan, accompanied by the number of repetitions that should be performed. The primary key of this table is a shared primary key composed by the partial keys that refer to the plan ID and the exercise ID.

4.3 Mobile App UI/UX design

In order to provide the user with the best experience when using this app, it is important to focus on the graphic interface. We envisioned achieving something simple and intuitive, yet functional.

The chosen colour pallet is mostly light and neutral, but we also use accents of blue, that is a common colour in marketing on the medical field [5].

Designing and prototyping the application

We started by designing and prototyping[4] the mobile application interface using the software Adobe Experience Design (Adobe XD). Several mockups of the designed interface are shown on Figures 4.3, 4.4 and 4.5. After this step it was possible to have a complete view of the app functionality.

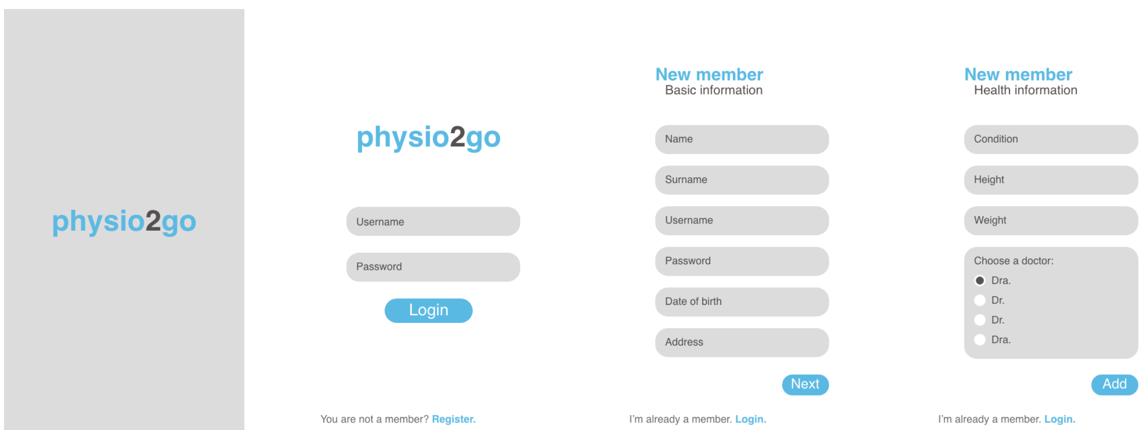

Figure 4.3: Splash, login and register screens (Adobe XD mockups).

The application initialises with a splash screen, where the logo is shown and then the user is prompted to a login activity. Here it is also the possible to redirect the user to the register section, for non-registered users, which allows the subject to signup on the *Physio2Go* platform. After a successful login, the logged in username is saved on `SharedPreferences` , all data concerning plans and exercises are stored on the local SQLite database and the application proceeds immediately to the Main Activity, where the prescribed plans are shown.

[4] `https://github.com/afonsoc12/CM_Project/blob/master/Prototype/Physio2Go%20-%20UI%20Mockups%20Demonstration.mp4`

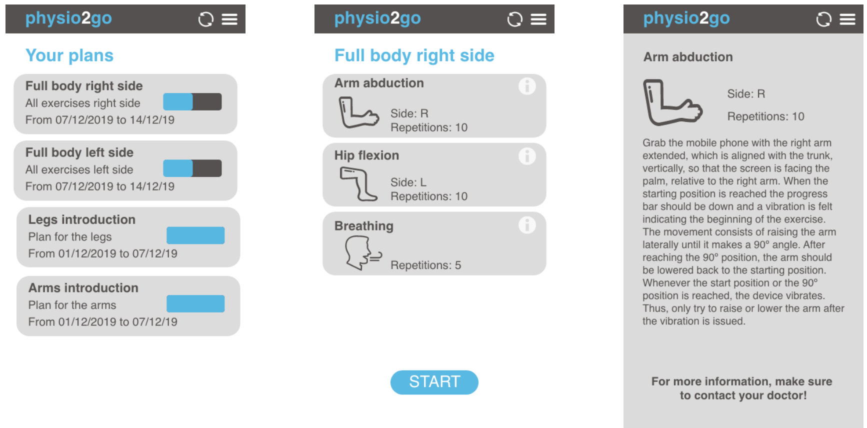

Figure 4.4: List of plans, plan's exercises and details of an exercise (Adobe XD mockups).

The landing page after the login is a list with his plans and information such as the name, the description, the starting and finish dates and a progress bar that shows the percentage of repetitions of the plan the user has already completed. When a plan is pressed, the UI displays the exercises prescribed for that plan. Each exercise is clickable and displays the description and specifications of such exercise.

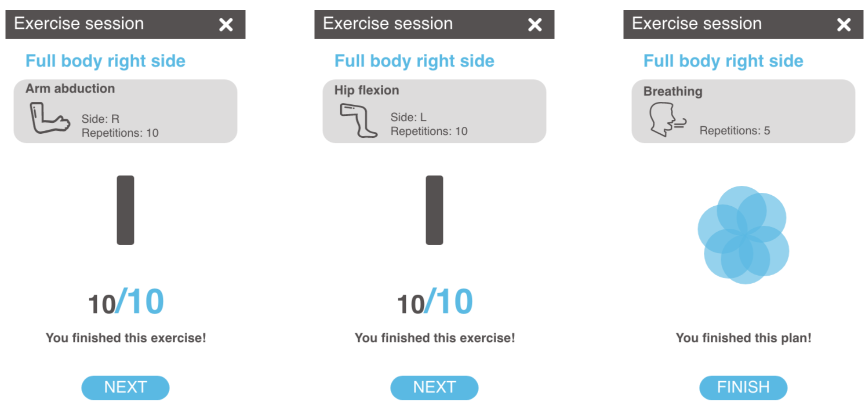

Figure 4.5: Exercise session of a plan (Adobe XD mockups).

For each exercise, there is a layout with its specifications, a progress bar that follows the movement or an image (in the case of the breathing exercise). When an exercise is finished, a `TextView`and button appear, allowing the user to proceed to next exercise or finish the current plan session (in the event that the current exercise is the last one on queue).

Final user interface

At the end of the development process, it was possible to obtain an Android based replication of the layouts presented in Figures 4.3, 4.4, 4.5, with only minor differences. Furthermore, we developed several new UI features, such as:

- A Menu with the options to see the user's profile, respective doctor's profile and logout (Figure 4.6);
- A Dialog Fragment to make sure the user wants to abort an exercise session (Figure 4.7).

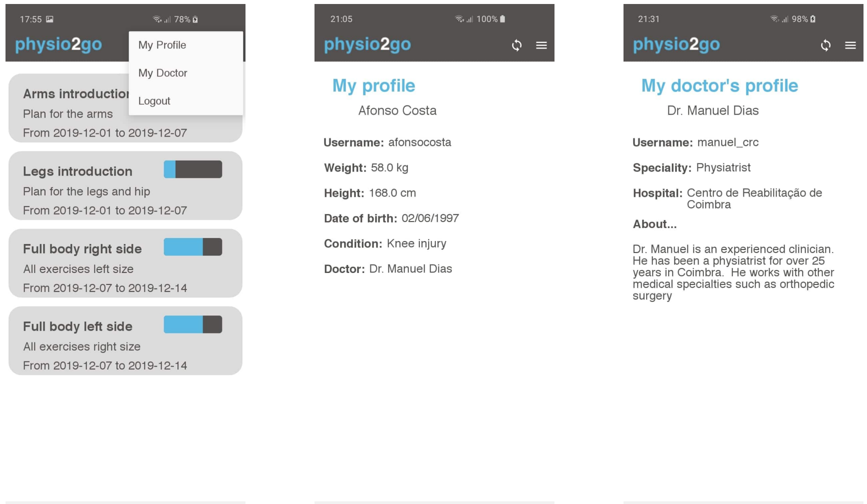

Figure 4.6: Menu options: user's profile and respective doctor's profile (Samsung S10).

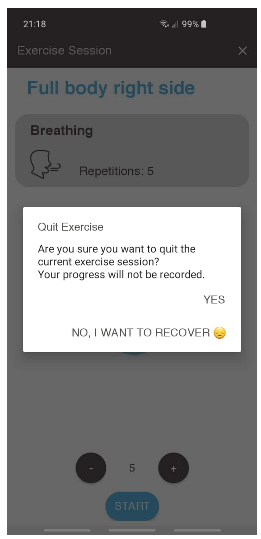

Figure 4.7: Dialog Fragment (Samsung S10).

5 Exercises Description

In order to supervise and assist the patient during their daily physical therapy exercises, *Physio2Go* aims to restore the physical, personal and social identity of patients, in the comfort of their own home or on-the-go, if it is preferable. The prescribed plans can contain up to 3 basic exercises, where two are for musculoskeletal recovery of arms and legs, specifically arm abduction and hip flexion, respectively and the other is a relaxing exercise, to regain the breathing rhythm. The following sections will delve further into the aspects of each of the aforementioned exercises, mainly its therapy utility, instructions to perform the exercise and mobile implementation.

5.1 Explanation of the Exercise

In order to start a plan session (do all exercises within a plan) the *Start* button should be pressed. This button starts an exercise session, where all exercises are done. If an exercise is finished, a *Next* button is displayed that advances to the next exercise and so on. When reaching the last exercise in the session, a *Finish* button pops-up instead, that saves the session in the database (if there is an internet connection), or locally (to be synced later with the server).

Arm Abduction Exercise

Before starting the exercise, it is recommended to read the information about the current exercise and check which arm will be subjected to physical therapy, where an “L” indicates the left arm, whereas the “R” stands for the right arm. It is also advisable to lock the rotation and perform the exercises while standing.

The arm exercise is developed for the patient to perform a lateral movement with the arm. The patient grasps the mobile phone with the arm extended, which is aligned with the trunk, vertically, so that the screen is facing the palm, relative to the arm that will be subjected to physiotherapy. When the starting position is reached the progress bar should be down and a vibration is felt indicating the beginning of the exercise. The movement consists of raising the limb laterally until it reaches a 90°angle. During the ascending movement, the progress bar will be increasing until it is full, meaning the 90°angle has been reached. At this moment, the phone will vibrate, indicating that the arm should be lowered back to the starting position. Whenever the start position or the 90°position is reached, the device vibrates. Thus, only try to raise or lower the arm after the vibration is issued.

Note that the mobile phone should be kept perpendicular to the arm throughout the entire exercise. This routine is repeated until the number of prescribed repetitions is reached.

Hip Flexion Exercise

Similar to the arm exercise, it is advisable to read the exercise instructions, check which leg is the exercise suited for and lock automatic rotation of the device. It is also recommended to perform this exercise while standing.

This exercise consists in lifting the leg vertically until it reaches a 90°angle. Then, hold on for 4 seconds and return to the starting position. The patient should place the mobile phone in the pocket or leaning against the leg, which is parallel to the limb, with the back of the mobile phone pressed against the thigh, in order to be able to read what is on the smartphone’s display. Similar to the arm exercise, there will be a progress bar that is empty at the starting position. When this position is reached the mobile phone will vibrate, indicating the beginning of the exercise. Then the leg should be lifted vertically, until the thigh is at a 90°angle with the calves, with the knee at its midpoint.

During the movement, a progress bar is increasing increasing, indicating that the 90°angle has been achieved when it reaches its maximum, along with a vibration signal. This signal indicates that this position should be held for 4 seconds. When the time has passed, a long vibration is issued, meaning that the initial position should be returned. As previously mentioned, the smartphone will vibrate at both start and end positions, where a larger vibration is felt whenever the leg is supposed to be lowered (after the 4 seconds hold on).

This exercise is then repeated until all prescribed repetitions have been performed. Note that it is advised to keep the phone parallel to the leg throughout the exercise.

Breathing Exercise

This exercise aims to calm the patient by synchronising the breathing rhythm. Here, the subject should inhale and exhale synchronised with the image movement, where the fade-out should represent an exhalation and the fade-in, an inhalation. The recommended repetitions are displayed, but the patient is encouraged to increment or decrement the number of inhalation/exhalation repetitions, using the `+` and `-` buttons. These two phases of breathing take five seconds each.

5.2 Physical Therapy Utility

Arm Abduction Exercise Utility

The shoulder is a ball and socket joint. There are 5 linked bone groups and 4 joints in total, allowing the shoulder to perform more complicated movements. This means that the ligaments and muscles are extremely important to produce and control shoulder movements. The rotator cuff muscles are commonly involved in shoulder problems [6]. It is common to have referred pain from a shoulder problem, which often radiates down to the upper arm, although usually not past the elbow. Pain may also be felt at the back, around the shoulder blade or in more specific locations, if one of the smaller joints is affected.

Exercise is often a helpful treatment for shoulder pain and will provide the joint with strength and flexibility. This one can help to mobilise the shoulder joint and activate the muscles in the rotator cuff, which help to support the shoulder.

A little post-exercise discomfort is often common and not a sign of damage. If the patient experiences pain that regularly lasts for more than 30 minutes after exercise and feels that the overall pain is worsening, the patient must stop all exercises and seek medical advice [6].

Hip Abduction Exercise Utility

The hip flexors are a group of muscles in the front of the hip that act to lift the knee and bring the thigh towards the abdomen. The major muscles making up the hip flexors that the exercise focuses on are the *iliacus, iliopsoas* and the *rectus femoris.* The latter is a "two-joint muscle" because it crosses both the hip and knee joints [7]. As a group, the hip flexors have attachments on the lumbar spine, the pelvis and the femur. In addition to their major function of flexing the hip, their attachment on the spine makes them an important part of the core muscles and spinal stabilisers [7].

Performing this exercise as a part of your cross-training routine will increase flexibility and strength of the hip flexors and help to counteract muscle imbalances in the hip. Keep in mind that the hip flexors are shortened in the sitting position, which consists in one more reason to lead experts to recommend not being sited for extent periods of time.

Breathing Exercise Utility

Breathing heals on many levels and understanding how it performs this function is good for both mental and physical well-being. Breathing constantly converts our life-sustaining energy, taking in oxygen, invigorating red blood cells and expelling carbon dioxide, which is a metabolic waste product. This exercise will improve the respiratory system, calm the nervous system, strengthen the lymphatic system, release muscle tension and improve the cardiovascular system.

6 Application Specifications

6.1 Database Server

When developing the app, the database server (a PostgreSQL 11 database) was running on a Raspberry Pi (RPi) (Model B Rev 2), with Raspbian 10 (buster) as the Operating System. It can be accessed using a dynamic DNS domain `http://acosta-server.ddns.net:5432`, provided by No-ip. This domain points to a home router (where the RPi is connected), that maps the 5432 port to the RPi's `54322` port, to access the database. The router's port `22` is also opened (and mapped to the same port on the RPi) to allow accessing the RPi, using the Secure Shell protocol (SSH). Note that the performance of the RPi is not satisfactory, mainly due to its 512MB of RAM, but is acceptable for this project.

However, after the application was finished, we migrated the database to a cloud-server, specifically a Digital Ocean droplet, whose public IP is `157.245.36.217`. The server is running the same PostgreSQL version, on Ubuntu 18.04 operating system.

Regarding the mobile app, we developed it to have API 23 (Android 6.0 Marshmallow) as the minimum SDK, which covers around 62.6% of the devices, according to Android Studio. Also, the target SDK defined was API 29 (Android 10), since this is the Android version of the device where the application will be deployed for the final presentation. All layouts were targeted for high resolution screens, with a diagonal of around 6 inches. However, we tested the application with 3 physical devices and 3 emulated devices, specifically:

Physical Devices:

- Samsung S10
- Samsung A5 2016
- Huawei P9 Lite

Emulated Devices:

- Pixel 3 XL
- Nexus 4
- Nexus 10 Tab

Credentials

In order to login onto the platform, we created two user accounts for the Professors to test the application:

Professor Tiago:

- **Name:** *prof_tiago*
- **Password:** *123proftiago*

Professor João:

- **Name:** *prof_joao*
- **Password:** *123profjoao*

6.2 Exercises Specifications

Arms and Legs Exercises

1. Through the `newInstance` static method, that is present on each fragment, we receive as an argument the exercise object that will be performed. Then, a new Fragment object is created (for the respective exercise) and the bundle that contains the Exercise object is set as an argument of the fragment. Next, on `onCreateView` the respective layout for the fragment is inflated and the necessary information is retrieved from the Exercise object, such as the number of repetitions, respective body side and its description.

2. On the onCreateView method the SensorManager class is instantiated, providing the SensorEvent as the context, followed by the definition of the type of sensor to be used (TYPE_ACCELEROMETER). The maximum value for the progress bar and the initial value are also set.

3. On the onResume method the sensor listener is registered and the parameter SENSOR_DELAY_NORMALis defined. It specifies the rate of information captured by the accelerometer. The SensorEventListener interface is implemented in this class, in order to access values read by the sensor. Furthermore, the SensorEventListeneris unregistered on onPausemethod, to avoid excessive battery drainage.

4. The onSensorChange method represents the basis of the exercise. It receives a SensorEventas an argument, where the acceleration values on each component x, y and z are extracted. Whenever there is a change in the values read by the accelerometer the onSensorChange method is called, allowing to control the device values, during the exercise execution.

5. By setting some conditions, such as odd or even and limb side, it is possible to issue phone vibrations, whenever it is desirable, for both cases. These cases are represented by the if statements in the source code. Note that we have accounted for some uncertainty, so that the user does not have to place the mobile phone in an exact position, increasing the ease of use.

6. Then, the repetitions counter is incremented throughout the exercise. When the exercise is finished a button pops-up, displaying *Next* if there are more exercises on the plan or *Finish* if the current exercise is the last. If the latter event occurs, a finish message is sent through an interface, finishing the activity and saving the exercise session on the server.

Breathing Exercise

1. The newInstance method receives the exercise as an argument, which is put on a bundle, similar to the Arm and Leg exercises. Following, on onCreateView method the layout is inflated and the exercise information is shown. It includes two buttons to increment or decrement the number of breathing repetitions. When the *Start* button is pressed, the animation starts.

2. The animation is implemented with a library named *ViewAnimator* [8], which is an open-source code, licensed under the Apache License 2.0. The ViewAnimator library is utilised on an ImageView that fades in and out, to emulate inhalation and exhalation, and then repeats for the number of repetitions set. Note that on onCreateView method we also set the initial image with a simple animation, using the same library, specifically a x and y translation.

3. When the repetitions are completed, a button pops-up and acts as an interface, sending a message to the exercise activity with its status.

7 Discussion

The application appears to be working fine on all physical and emulated devices tested and no bugs were discovered up to this point, which does not mean that bugs or errors are not present. Still, there are several aspects worth mentioning, regarding the developed app.

First, we are aware that a web application to prescribe physiotherapy plans (by the doctor) would be beneficial. Nevertheless, as stated in Section 4.1 we did not have time to implement both applications. Nonetheless, we strongly believe that a Django web application would be suitable, due to its quick development process.

Regarding the security concerns, the passwords introduced on the login or signup are encrypted using the `SHA256` encryption algorithm. This is a good security practice since passwords are not sent unencrypted over the web. The server only holds the password's hashes, that are compared on login. Despite, the connection to the server is made via `DriverManager` class, which requires that both database username and password are hard-coded on the app's source code. We are aware that this is a security issue but we did not come up with another easily implementable solution. We argue that, even though this is a security concern, the objective of the assignment is to develop a mobile app and we can tolerate security breaches until a certain degree. Still, the hard-coded PostgreSQL user has limited capabilities (only `SELECT`, `INSERT` and `UPDATE` privileges), in order to minimise the nefast effects of a potential invasion. Possible solutions could be the implementation of several web-services or a REST API that offered the necessary `SELECT`, `INSERT` and `UPDATE` operations.

Next, the logic that defines an exercise could be improved for better movement detection. We only consider several thresholds to define the begin/end of an exercise, but this logic could be enhanced. We only implemented three exercises, but the current system was designed to allow easy integration of more exercises, being only necessary to define the new exercise's UI, business logic and insert the exercise's description into the database. In terms of user experience, it could benefit from a small video clip or an animation, demonstrating how the exercise should be performed, instead of a textual description, in order to facilitate the comprehension by the user and avoid performing the exercise incorrectly.

Furthermore, the storage of data from the exercise sessions could be extended, in order to also persist meta-data and sensor values. By doing so, the data is saved on the database and can later be analysed using Machine learning techniques, aiming to find patterns on the data, in order to improve the implemented exercises or for statistical analysis of the application users, in a hypothetical scalable scenario. Furthermore, a dynamic algorithm to define the sensors sensibility thresholds could be developed, if the data was collected, that would maximise the compatibility of the exercise with each specific patient.

8 Conclusion

At the end of this Mobile Computing project, we were able to develop an Android mobile application, that extended the task's objectives, including a remote server to rule the platform. The project management methodology adopted was an important asset, that focused on planning, deliverability and quality, that led to an application characterised by usefulness, practicality and ease-of-use. The group elements believe that this project achieved the postulated goals, as mentioned in Section 2.

By developing the present assignment, it was possible to broaden our knowledge in several topics lectured in the course, such as activities, fragments, databases and `AsyncTasks`. Following our workshop theme, we chose to make use of the smartphones' sensors, as means to tackle an opportunity on the Biomedical field, that we believe that has potential to be profitable, if further enhancements on the *Physio2Go* platform take place.

It was the first time that the team members implemented an application with both back-end and front-end, and we are very pleased with the final result. In sum, we consider that this course was a great complementary subject for our masters' degree. Concepts have been assimilated that will most certainly be useful for our future.

References

[1] Kinesiology. Wikipedia. Accessed: 2019-12-26. [Online]. Available: https://en.wikipedia.org/wiki/Kinesiology

[2] Welcome to the future of musculoskeletal care. Sword Health. Accessed: 2019-12-26. [Online]. Available: https://swordhealth.com/

[3] T. Cruz and J. Fernandes, "Mobile computing project," *Mobile Computing*, 2019.

[4] What is scrum? Scrum.org. Accessed: 2019-12-26. [Online]. Available: https://www.scrum.org/resources/what-is-scrum

[5] Why most medical logos are blue. Doctor Multimedia. Accessed: 2019-12-26. [Online]. Available: https://doctormultimedia.com/medical-logos-blue/

[6] Clinical education. Allied Health Professionals. Accessed: 2019-12-26. [Online]. Available: https://ahpsuffolk.co.uk/Home.aspx

[7] Hip flexors: strength and flexibility. Fleet Feet. Accessed: 2019-12-26. [Online]. Available: https://www.fleetfeet.com/s/hartford/sports-medicine/sports-medicine-corner/hip-flexors-strength-flexibility

[8] Viewanimator: A fluent android animation library. Github. Accessed: 2019-12-26. [Online]. Available: https://github.com/florent37/ViewAnimator