Documentation

Table of Contents

Introduction

Purpose of the System

ACLReality app makes reliable and quick measurements in the treatment of patients with knee injuries, with the help of the accelerometer sensor and the depth camera of the iPhone, which is available on iPhone X and later versions. Nowadays doctors are very unsatisfied with the way of determining an injured knee. They use machinery such as the Rolimeter or the Latchmeter for measuring the translation of the tibia from the patella, while doing the Lachman test. The Lachman test is done to check for an anterior cruciate ligament (ACL) injury. Not only is this machinery expensive, but getting the correct leg translation fails sometimes and the procedure has to be repeated. In our app, the doctor can choose between two ways of completing a measurement. While doing the Lachman test, he can either use the Augmented Reality camera, where in a really quick way he can obtain many results and get an average of the translation, or use the Accelerometer sensors and get the slope of the test, which correlates with the laxity of the knee. In a couple of minutes the doctor will have a profile of the patient with all his injury related data, which he can track easily for all future sessions. The main users are the doctors who will use this app to increase efficiency in their day to day life and the stakeholders will be the clinics that will profit from the affordability of the app and the suppliers of the strap bands, markers or chest belt for the use of all possibilities.

Current System

Our customer presented us with an app "SmartKiRA that came with a device which supported the analysis of knee joint kinematics and dynamic laxities, while providing realtime graphics and information about the examination. It was produced by an Italian company named "Orthokey Italia s.r.l.", that developed it in 2012 but stopped the production and sales. This product can be found on http://www.smartkira.com/en/ .

Requirements

Visionary Scenarios

The requirements for the ACL Reality App will be categorized into functional and non-functional requirements.

First we'll lay out the functional requirements.

The first requirement is to present an Application that is able to recognize a rapture of the ACL. A ligament located in the knee. Furthermore the test used to detect this rapture should be the Lachman Test.

The second functional requirement includes the visualization of the data captured by the measurements of the app, as its important for the doctor to be able to base his resulting decision on the tests. Here we include our per-report comparison of the two knees and the progress over time concerning the patients recovery.

A third functional requirement for the app is the comparison of different patients results to determine differences in for example recovery speed. This is useful for the doctor in order to determine how well different treatments compare. We enable this in our app by a Comparison-feature for two patients visualizing the differences in measurements over time.

For the non-functional requirements our first one is performance and speed. The doctor needs to be able to open the app and do measurements quickly. Especially compared to using the Rollimeter the app should provide advantages in this area.

Anoother non-functioonal requirement should be robustness, especially against failures while taking the measurement. As the app is used in a time-restricted environment, for example when taking measurements on a patient, the app should deal with measurement errors by itself, so that the doctor doesn't need to put in much effort.

Our trailer as seen in the Design Review presents our visionary scenario.

User Interface

PatientList View

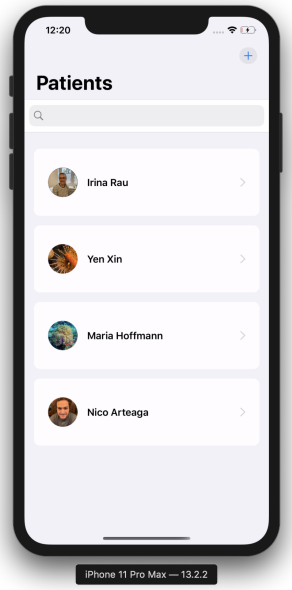


Fig. 3.1.1: Homescreen of the app with an overview of all patients.

AddPatient View

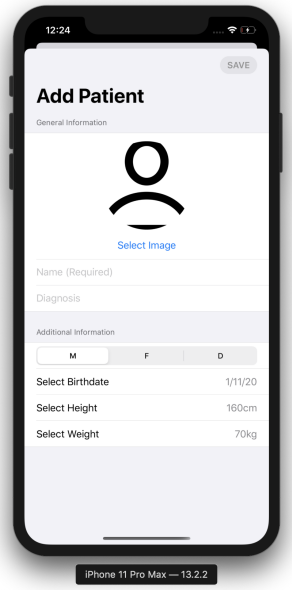


Fig. 3.1.2: By tapping on the top right on the home screen on the Plus-Button, you will get to the Add Patient View. A patient can be added to the PatientsList.

Patient View

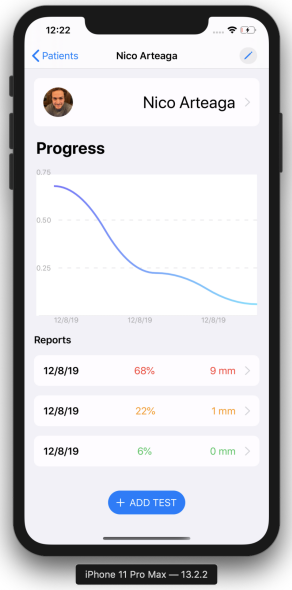


Fig. 3.1.3: By tapping on any patient in the Homescreen you will get an overview of a patient with his current progress and every report taken.

Edit Patient View

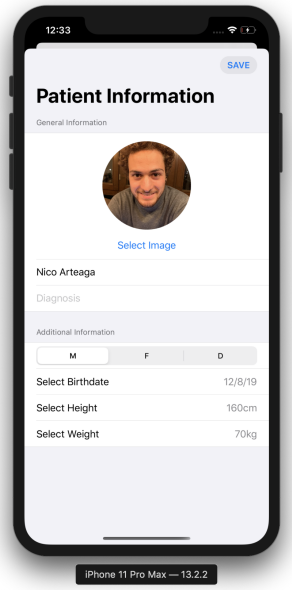


Fig. 3.1.4: By tapping on the top right in the Patient View, you will get to the Edit Patient View. An existing patient can be edited here again.

Augmented Reality View

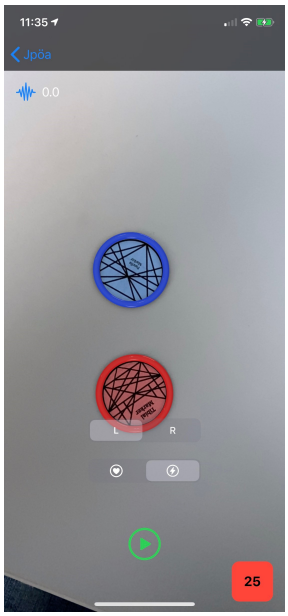


Fig. 3.1.5: By tapping on the Add Test Button in the Patient View and swiping to AR, you get to the AR View. A test can be performed here using Augmented Reality. The bottom right number indicates the distance of the phone to the markers. A green number will show a sufficient distance.

AR Report View (1)

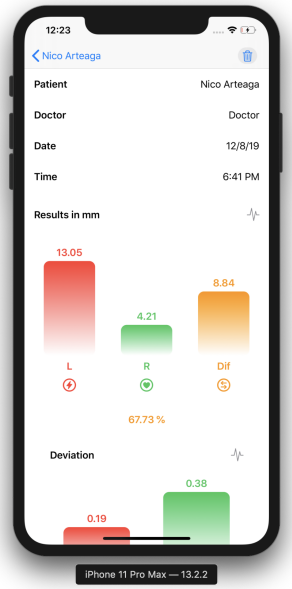


Fig. 3.1.6: After performing an AR Test, you get redirected to the AR Report View. It shows all metadata of an AR report. The result data is the most valuable one, since it shows each result of the legs and their difference.

AR Report View (2)

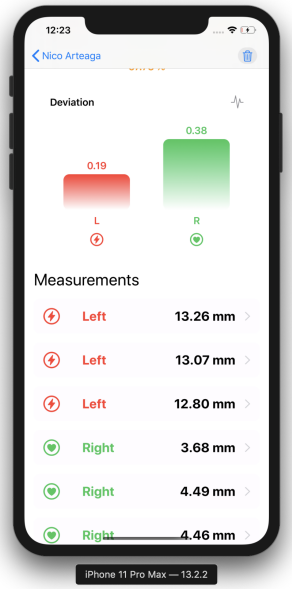


Fig. 3.1.7: Shows the deviation of all measurements, this is to ensure that valid measurements have been taken. Also shows all measurements taken with an option to look at each one in more detail.

Acceleration View

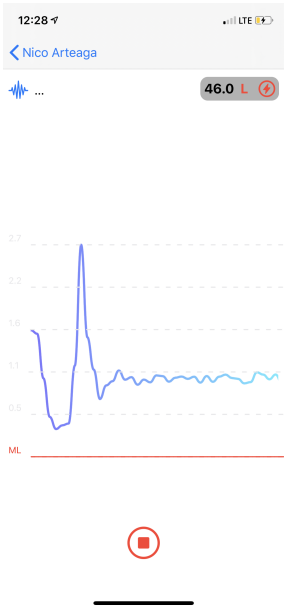


Fig. 3.1.8: By tapping on the Add Test Button in the Patient View and swiping to ACC, the ACCView is displayed. It shows the net acceleration while the test is performed.

ACC Report View

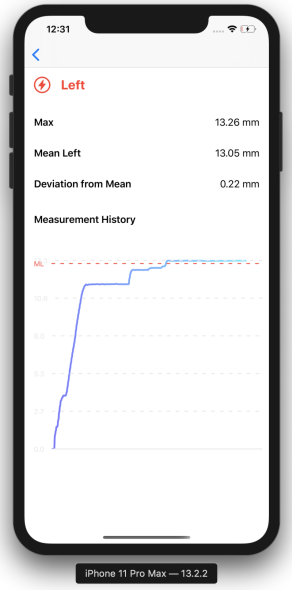
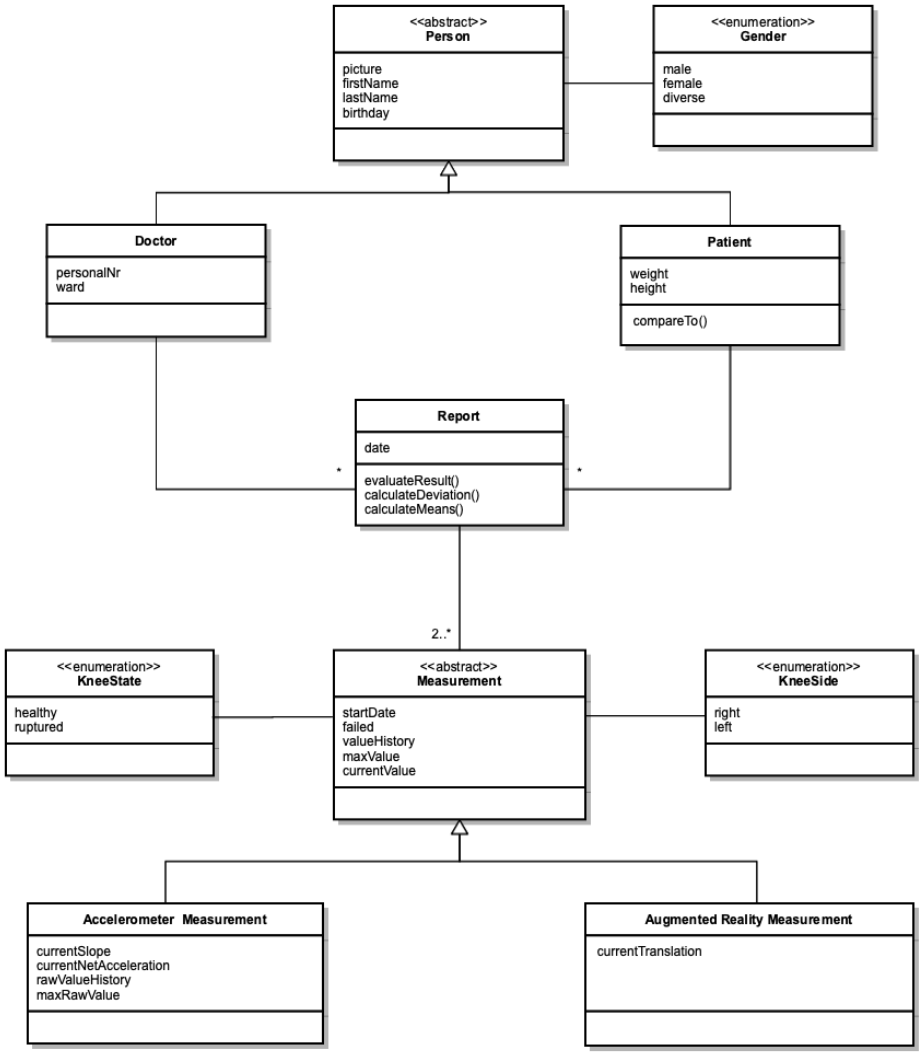


Fig. 3.1.9: After performing an ACC Test, you get redirected to the ACC Report View. It shows a more detailed view of the measurement. Interesting here is the graph that shows the history of values while the test was taken.

Analysis

Analysis Object Model



A person can either be a patient or a doctor. The doctor is the one that performs the Lachmann test and the patient is the one the test is performed on. To track the progress of each patient, one patient can have multiple reports. Also one doctor is linked to multiple reports, since you want to trace back, who performed the test.

One Report has to at least consist of two measurement, since the comparison of both knees is needed to calculate the result of the test.

A measure can either be an Acceleration Measurement or an AR Measurement, since both solutions have been implemented and have to be evaluated in the long run, which one is suited better to track the progress of ACL patients.

Glossary

Term	Definition
Patient	A person with knee injury. Each patient may have several reports and a progress tracking.
Doctor	A qualified person working at the clinic to treat patients. The doctors create new reports by examining the patients.

Report	Contains measurement data (either Acceleration Measurement Data or AR Measurement Data) obtained from the examinations of a patient's both knees by the doctor. The patient have several reports (before/after operation). The report can be saved and stored in the patient profile.
Acceleration Measurement Data	Data obtained from the accelerometer by examining one knee of the patient (e.g., net acceleration in each direction (x,y,z) over time)
AR Measurement Data	Data obtained using AR by examining one knee of the patient, contains translation of the tibia over time
Knee Measurement Data	Describes the state of the knee (healthy/ruptured, left/right, translation, rotation, acceleration).
Examination	Lachmann Test on one knee; performed to quantify the anterior cruciate ligament rupture

System Design

Overview

Our main components are the Accelerometer and Augmented Reality Tracker, which are used to acquire the data necessary for our two strategies. Furthermore the App contains components to collect the measurements, manage the report, doctor and patient data and visualise the data to the user. In the future it is planned that the app is fully integrated in the clinic software Tomedo, which holds all patient and doctor data.

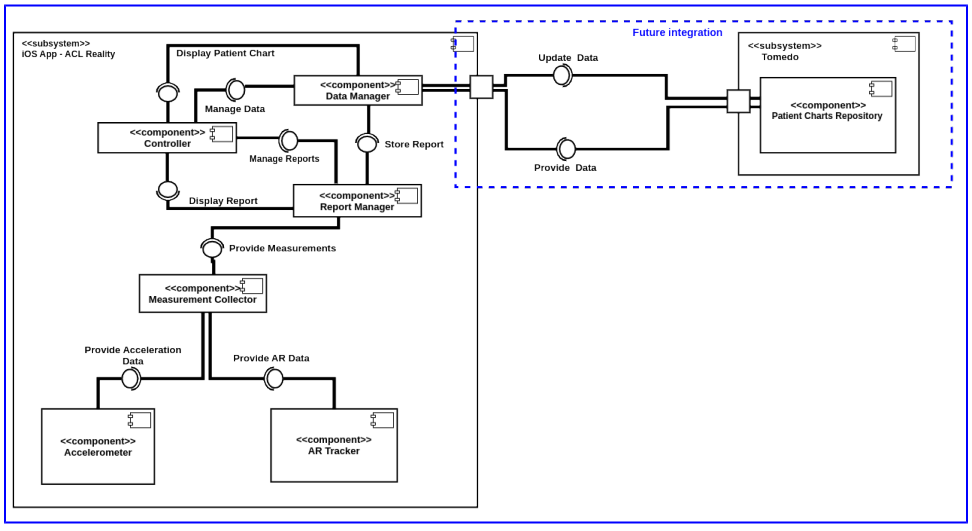
Accelerometer Component

For this approach the iPhone is held by the doctor on the patients Tibia, while the doctor performs multiple Lachman tests. During the test, the Accelerometer Component acquires the acceleration data and provides it through CoreMotion. CoreMotion provides the acceleration data in three axes, which we use to compute the net acceleration so that the test works regardless of iPhone positioning.

Augmented Reality Tracker Component

For this approach we use ARKit with image recognition. We place two markers on the patients leg, one on the Tibia and one on the Patella. When the doctor now performs the Lachman test ARKit is able to provides us with the relative vector data of the markers by knowing the exact size of the markers combined with the depth sensor of the iPhone. We save the base-vector in the first few frames and compute the translation movement every frame using the current vector between the marker points and span a triangle with the base-vector. By knowing the exact length of the vectors and the angle between them, we are able to calculate the exact translation movement by using the cosine rule.

Subsystem Decomposition



Description

The main goal is to determine a qualitative result of the Lachman test using an iPhone. This means the app should be able to tell whether an ACL is ruptured or not. To accomplish this we chose two different independently working strategies which are resembled in two subsystems (Accelerometer and Augmented Reality).

Accelerometer subsystem

The Accelerometer subsystem encapsulates the iPhones Gyroscope sensor and the on top laying framework core motion, which provides the App with the acceleration data in three axes.

Augmented Reality subsystem

The Augmented Reality subsystem encapsulates the iPhones depth sensor and the on top laying framework RealityKit, which provides the app with relative vectors of all included image anchors as well as the relative camera position.

Knee App

Measurement Collector

Accelerometer:

The Measurement Collector automatically detects when a new measurement has started and ended by using a threshold on the acceleration data provided from the accelerometer subsystem.

Augmented Reality:

The Measurement Collector automatically detects when a new measurement has started and ended by using combined thresholds. These thresholds consists of a minimum time for a measurement, a maximal translation movement which has to be reached and a lower bound, where the current translation movement has to fall below to end a measurement.

Report Manager

The Report manager creates a new Report based on one of the two strategies and starts a session for the measurement collector. It also provides the functionality to swap the measured knee within the session and computes a result of the test when the session has ended. Result implicates the finalized version. A report is either of type ARMeasurement or AccMeasurement and computes the result from all measurements in this report.

Report Manager (Result computation)

Accelerometer:

As a healthy ACL stops the translation movement of the Tibia relative to the Patella during the Lachman test, we see a firm endpoint in the acceleration data of the measurements. When the ACL is ruptured the translation movement is stopped by the soft tissue around the Tibia. By using statistical analysis on all acquired measurements we can determine a qualitative result.

Augmented Reality:

Here we calculate the means and standard deviations of all measurements on both knees. The means are then subtracted from each other to determine a quantitative result as absolute value in millimetres and relative value. (e.g. Left injured knee 11mm and right healthy knee 3mm Dif. 8mm / 72,7%). The standard deviation provides additional information for the user, as high standard deviation can indicate that a few measurements went wrong or the doctor used different forces to compute these measurements.

Data Manager

The Data manager associates reports with the corresponding doctor and patient. It also cares about the data persistence. Which is currently achieved by saving the data locally in a json format.

UI

Visualises the data. Doctors can choose or create patients, with which they can start a new report based on a chosen strategy.

Tradeoffs and Decisions

Strategy Pattern

As both ideas are quite promising we chose to integrate both using the strategy pattern. A doctor can decide which strategy he is using for the examination.

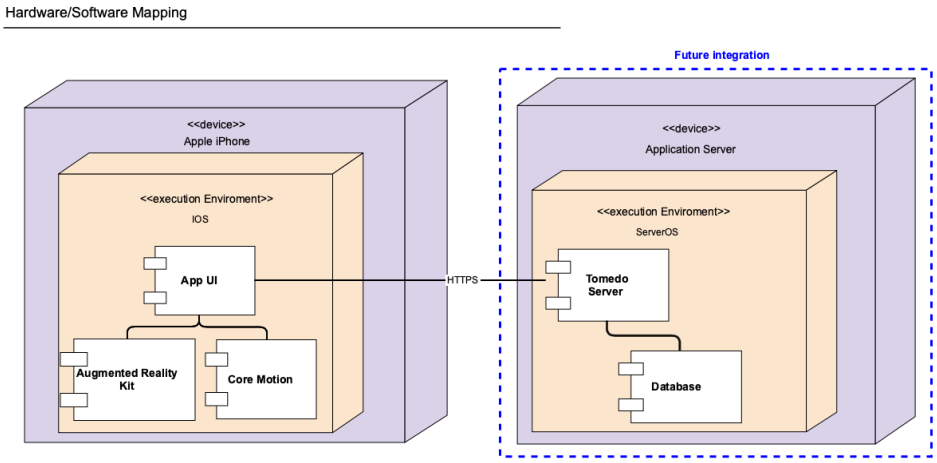
Tomedo subsystem

Tomedo is a clinic software used in many clinics. It holds data about all patients and employees of a clinic / doctors office. The vision here is to integrate the ACL Reality app seamlessly with the Tomedo backend as diagnostic tool. As the Tomedo API is not open source and we couldn't establish a contract with Tomedo in the short time, we chose to save the data locally in the initial version, especially considering that the app is a proof-of-concept.

Data Manager

The Data manager will manage retrieval and upload of updated data in the future.

Hardware/Software Mapping



Description

Client

Hardware: iPhone X or higher (needs to have a depth sensor)

Software: iOS 12 or higher

Communication Protocol

Will likely be HTTPS on a RESTful API

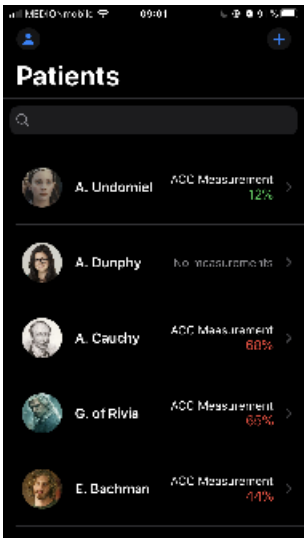
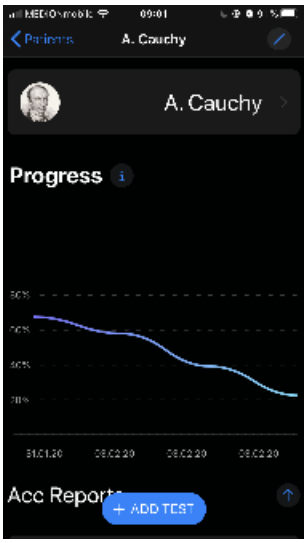
Persistent Data Management

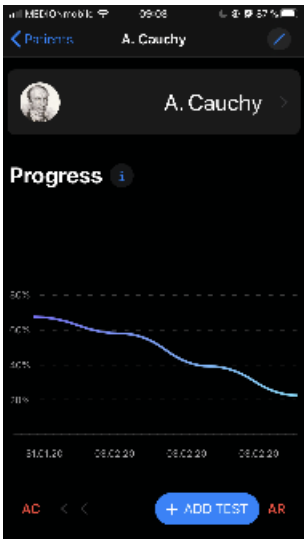
Description

The data classes are serialised into JSON and saved locally on the device.

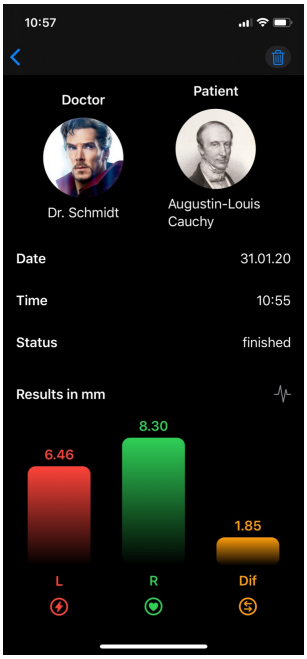
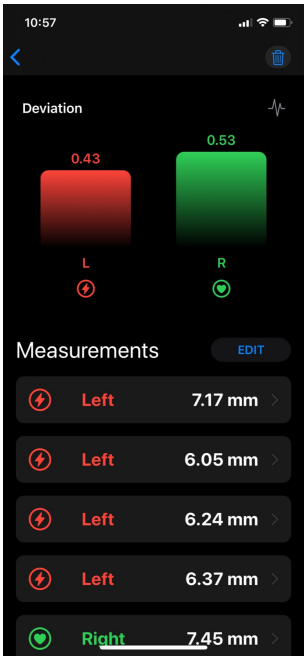
Demo Scenario

Scenario name	Taking Measurements with Augmented Reality and camera
Participating actor instances	Dr. Schmidt : Doctor, Augustin-Louis: Patient

Flow of events	1. Dr. Schmidt chooses his patient Augustin-Louis on the list of the patients		
		2. App opens Patient Chart of Augustin-Louis	
	3. Dr. Schmidt puts on the chest belt and places his iPhone on it		

4. Dr. Schmidt places AR markers on the Tibia and Patella of Augustin-Louis		
5. Dr. Schmidt swipes right to start the Augmented Reality Session		
	6. App starts AR Session.	

7. Dr. Schmidt takes 4 measurements on the injured knee and pauses the session to change the measured knee.		
8. Dr. Schmidt repeats the steps 4 and 7 for the healthy knee and finishes the session		

		9. App ends the session and then generates and opens the report.	 
Entry conditions	• doctor has an IPhone X or later • doctor has a chest belt • doctor has Tibia and Patella AR markers, which are compatible with the ACL Reality app • doctor has ACLReality app opened		
Exit conditions	• doctor has a report with evaluation of the knee injury of the patient		
Quality Requirements	• Augmented Reality markers are detected by the system • system detects the start and the end of each measurement (automatic control session) • acoustic feedback is synchronised with the motion detection		

Product Backlog

All required use cases that were required by the client have been implemented.

- A user (physician) can quantify anterior translation movements during the Lachmann test. This functionality works as good as the Rolimeter that is currently being used for the Lachmann test.
- The device is easy to use and work on a non-invasive basis (no penetration of the skin by parts of the device) in a clinical routine settings.
- Measurement results can be displayed on the app with graphs that help to understand the data that was recorded. The data is optimized and structured in a way that allows easy export to third-party software, e.g. Tomedo.

An additional optional use case, that a user (physician) can quantify the lateral translation or rotation movements (rapid i.e. reduction of the tibia during the pivot shift test) was not implemented due to the complexity of the pivot shift test.

Administrator Manual

Infrastructure Setup

For development of the app, an Apple device that is able to run macOS Catalina or newer is required. Xcode 11.2.1 or newer is required to build the app, since it relies on SwiftUI and RealityKit.

ACL Reality does not utilize a backend or any other kind of external dependency, the Xcode project includes all dependencies that are required to successfully build and run the app in its current state.

Please be aware that both the accelerometer as well as the augmented reality functionalities do not run on the simulator.

Requirements for the accelerometer functionality

- iPhone that supports iOS 13

Requirements for the augmented reality functionality

- iPhone that supports iOS 13. Additionally, for augmented reality to work well, it is advised to use an iPhone X or newer.

Deployment and Configuration

Steps to run ACL Reality on local device

1. Download/clone the git repository of the project
2. Open **sportclinic.xcworkspace** by double-clicking or using "Open.." in Xcode
3. Choose a suitable target device that is attached to the development machine in the top bar
4. Press the play button

Known Issues and Workarounds

Running the project in the Simulator **does not work** if the following line is uncommented in *sportclinic/ARMeasurementARController.swift*:

```
self.session.delegate = self
```

For UI testing purposes, it is advised to comment the line out for simulator runs:

```
//self.session.delegate = self
```

The augmented reality functionality will not work on the device if this line is commented out.

Third-party Components

SwiftLint 0.36.0 is used to keep the code consistent and clean.