

# Android App für Ersteichung

## Studienarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Herbstssemester 2018

Autoren: Stefano Kals, Valentina Merturi  
Betreuer: Prof. Dr. Markus Stolze  
Projektpartner: Mettler Toledo



## Abstract

Die Service-Techniker der Firma Mettler-Toledo (Schweiz) GmbH verwenden bei einer Ersteichung eine Desktop-Applikation namens *Miracal*. Dieses Programm ist nicht für solche Arbeiten vorgesehen und deshalb wird eine Applikation benötigt, die alle notwendigen Schritte abbildet und mit der keine Fehler gemacht werden können. Da alle Service-Techniker ein Android Smartphone besitzen, welche Bluetooth integriert haben und die METTLER TOLEDO Bluetooth Adapter (ACM360) als Kabelersatz anbieten, soll eine Lösung entstehen, die beides kombinieren kann. Die Applikation soll, mittels Bluetooth Low Energy, Verbindung mit dem Terminal oder der Waage aufnehmen, damit kommunizieren und den Benutzer durch den Ersteichungsprozess führen.

Unsere Lösung ist einmalig, weil solche Adapter bisher nur als Kabelersatz zwischen Terminal und Drucker oder Waage und Drucker eingesetzt wurden. Ziel dieses Projekts ist die Evaluation des Adapters, wie kompatibel er mit dem Terminal oder der Waage ist und die Umsetzung der Ersteichungsschritte, so dass die Applikation in Zukunft für die METTLER TOLEDO schweizweit von den METTLER TOLEDO Service-Technikern eingesetzt werden kann.

Mit dem aktuellen Stand der Applikation kann durchaus ein Pilot-Test durchgeführt werden. Wegen der Komplexität des Eichprozesses und der Anzahl an Waagentypen und deren verschiedenen Ausführungen von SICS Commands, mussten wir uns bei der Funktionalität einschränken und uns für diese Arbeit spezialisieren. Es steckt jedoch ein grosses Potential in der Nutzung dieser Technologie und unserer Applikation, welche als Ausgangspunkt für weitere Projekte als Basis verwendet und nach belieben erweitert und umgebaut werden kann.

# Management Summary

## Ausgangslage

Eine wichtige Produkt-Familie der METTLER TOLEDO sind Hochpräzisionswaagen. Für die Einrichtung einer solchen Waage muss eine Ersteichung nach Norm EN45501 durchgeführt werden. Diese Aufgabe wird von METTLER TOLEDO Service-Technikern durchgeführt. Aktuell wird dafür eine Windows-Anwendung genutzt, welche für die Waagen-Kalibrierung, aber nicht für die Ersteichung entwickelt wurde. Entsprechend ist die Nutzung der Anwendung unnötig kompliziert. Zudem ist die Software nicht direkt mit der Waage verbunden. Somit müssen alle Messergebnisse manuell in die Software eingetragen werden, was sehr fehleranfällig und einfach zu manipulieren ist. Diese Fehlerquelle soll minimiert und der Eichprozess soll mit einem dafür vorgesehenen Programm durchgeführt werden.

## Vorgehen und Technologien

Das Ziel dieser Arbeit war zweigeteilt. Zum einen sollte bestimmt werden, ob sich der Ablauf der Ersteichung auf einfache Art von einer Android App aus steuern lässt, zum anderen sollte analysiert werden, ob der von METTLER TOLEDO vertriebene Bluetooth LE Adapter (ACM360) zur automatisierten Kommunikation mit der Waage genutzt werden kann. Die Hoffnung war, dass aus der Arbeit eine App entsteht, welche über den Adapter alle relevanten Informationen im Eich-Prozess bezieht und darauf aufbauend Techniker Schritt für Schritt, effizient, fehler-minimierend und leicht verständlich durch den Eich-Prozess führt.

## Ergebnisse

Anfängliche Probleme mit der Kommunikation mit dem Bluetooth Adapter konnten erst spät überwunden werden. Aufgrund der verlorenen Zeit, konnte den Themen Effizienz, Fehler-Minimierung und Verständlichkeit der App nur weniger Aufmerksamkeit geschenkt werden. Trotzdem konnten in dem durchgeführten Usability Test und dem Usability Walkthrough wichtige Erkenntnisse gesammelt werden, welche in das Design des entwickelten Systems eingeflossen sind.

## Ausblick

Dieses System kann nach geringen weiteren Anpassungen für einen Pilot Test mit mehreren Service Technikern in der Schweiz eingesetzt werden. Eine Aufwandschätzung für die noch notwendigen Arbeiten wurde gemacht. Weitere wünschenswerte und mögliche Erweiterungen wurden als optionale Features dokumentiert.

# Inhaltsverzeichnis

|   | <b>Seite</b> |
|---|--------------|
| <b>1 Einleitung</b>   | <b>8</b>     |
| 1.1 Ausgangslage . . . . .  | 8            |
| 1.2 Zweck und Ziel . . . . .  | 8            |
| 1.3 Bereitgestellte Hardware . . . . .                                | 9            |
| 1.4 Lieferumfang . . . . .  | 9            |
| 1.5 Annahmen und Einschränkungen . . . . .                            | 9            |
| 1.6 Umfang . . . . .  | 9            |
| <b>2 Aufgabenstellung</b>   | <b>10</b>    |
| <b>3 Technologie</b>  | <b>13</b>    |
| 3.1 Bluetooth Low Energy (BLE) . . . . .                              | 13           |
| 3.2 Adapter (ACM360) . . . . .  | 14           |
| 3.3 SICS-Befehle . . . . .  | 14           |
| <b>4 Analyse</b>  | <b>16</b>    |
| 4.1 Domain Model . . . . .  | 16           |
| 4.2 Use Cases . . . . .   | 17           |
| 4.2.1 UC1: Verbindung zum Bluetooth-Adapter . . . . .                 | 17           |
| 4.2.2 UC2: Konfiguration der Waage auslesen . . . . .                 | 17           |
| 4.2.3 UC 3: SICS-Befehle an die Waage senden . . . . .                | 18           |
| 4.2.4 UC4: Ersteichungsprozess durchführen . . . . .                  | 18           |
| 4.2.5 UC5: Benutzerdaten verwalten . . . . .                          | 18           |
| 4.2.6 UC6: Zertifikat ablegen . . . . .                               | 19           |
| 4.2.7 UC7: Kundendaten verwalten . . . . .                            | 19           |
| 4.2.8 UC8: Konformitätszertifikat erstellen . . . . .                 | 20           |
| 4.2.9 UC9: Alle verfügbaren SICS-Befehle der Waage auslesen . . . . . | 20           |
| 4.2.10 Use Case Diagramm . . . . .                                    | 21           |
| <b>5 Architektur</b>  | <b>22</b>    |
| 5.1 Deployment . . . . .  | 22           |
| 5.2 Code Struktur . . . . .   | 23           |
| 5.3 Libraries . . . . .   | 24           |
| 5.4 Metriken . . . . .  | 25           |
| 5.4.1 Lines of Code . . . . .   | 25           |

|          |   |           |
|----------|---|-----------|
| 5.4.2    | Kopplung und Kohäsion . . . . .           | 25        |
| 5.4.3    | McCabe . . . . .                          | 25        |
| 5.5      | Eingesetzte Patterns . . . . .            | 26        |
| 5.5.1    | Observer Pattern . . . . .                | 26        |
| 5.5.2    | Adapter Pattern . . . . .                 | 26        |
| <b>6</b> | <b>Qualitätsmanagement</b>                | <b>27</b> |
| 6.1      | Nicht funktionale Anforderungen . . . . . | 27        |
| 6.1.1    | Usability . . . . .                       | 27        |
| 6.1.2    | Stability . . . . .                       | 27        |
| 6.1.3    | Portability . . . . .                     | 27        |
| 6.1.4    | Security . . . . .                        | 27        |
| 6.1.5    | Accuracy . . . . .                        | 28        |
| 6.1.6    | Accessibility . . . . .                   | 28        |
| 6.2      | Wireframes . . . . .                      | 28        |
| 6.2.1    | Erster Entwurf . . . . .                  | 28        |
| 6.2.2    | Zweiter Entwurf . . . . .                 | 29        |
| 6.3      | Usability Tests . . . . .                 | 30        |
| 6.4      | Code-Standards . . . . .                  | 30        |
| 6.5      | Code-Review . . . . .                     | 31        |
| 6.6      | Lint . . . . .                            | 32        |
| 6.7      | Commits . . . . .                         | 32        |
| 6.8      | CI/CD . . . . .                           | 32        |
| 6.9      | Tests . . . . .                           | 33        |
| <b>7</b> | <b>Development</b>                        | <b>34</b> |
| 7.1      | Tools . . . . .                           | 34        |
| 7.1.1    | Android Studio . . . . .                  | 34        |
| 7.1.2    | Visual Studio Code / TexStudio . . . . .  | 34        |
| 7.2      | Architekturprototyp . . . . .             | 34        |
| 7.3      | Applikation . . . . .                     | 35        |
| 7.4      | Testumgebung . . . . .                    | 35        |
| 7.4.1    | Server . . . . .                          | 35        |
| 7.4.2    | Simulation . . . . .                      | 35        |
| 7.5      | Anleitungen . . . . .                     | 36        |
| <b>8</b> | <b>Evaluation</b>                         | <b>37</b> |
| 8.1      | Kommunikation Adapter . . . . .           | 37        |
| 8.1.1    | Adapter . . . . .                         | 37        |
| 8.1.2    | Android Bluetooth Low Energy . . . . .    | 37        |
| 8.2      | Kommunikation Waage . . . . .             | 38        |
| 8.2.1    | Waage . . . . .                           | 38        |
| 8.2.2    | Simulation . . . . .                      | 39        |
| 8.3      | Eichprozess . . . . .                     | 40        |

|           |   |           |
|-----------|---|-----------|
| 8.4       | Graph Eichfehlergrenze . . . . .                | 40        |
| 8.5       | Zertifikat . . . . .                            | 40        |
| 8.6       | Fazit . . . . .                                 | 41        |
| <b>9</b>  | <b>Verzeichnisse</b>                            | <b>42</b> |
| 9.1       | Glossar und Abkürzungen . . . . .               | 42        |
| 9.2       | Abbildungsverzeichnis . . . . .                 | 45        |
| 9.3       | Tabellenverzeichnis . . . . .                   | 45        |
| 9.4       | Literatur . . . . .                             | 45        |
| <b>10</b> | <b>Appendix</b>                                 | <b>49</b> |
| 10.1      | Testprotokoll . . . . .                         | 49        |
| 10.2      | ACM360 Datasheet . . . . .                      | 50        |
| 10.3      | SICS Commands . . . . .                         | 52        |
| 10.4      | Benutzeranleitung . . . . .                     | 61        |
| 10.5      | Installationsanleitung . . . . .                | 69        |
| 10.6      | Lizenzen . . . . .                              | 71        |
| 10.7      | Metriken . . . . .                              | 74        |
| 10.8      | Detail Risiken . . . . .                        | 80        |
| 10.8.1    | Wöchentliche Risiko Analyse . . . . .           | 80        |
| 10.9      | Qualitätsmanagement Abschluss Screens . . . . . | 83        |
| 10.9.1    | Lint . . . . .                                  | 83        |
| 10.9.2    | Unit Tests . . . . .                            | 83        |
| 10.9.3    | Funktionale Tests . . . . .                     | 83        |
| 10.9.4    | Gitlab CI/CD Pipeline . . . . .                 | 84        |

# 1 Einleitung

## 1.1 Ausgangslage

Für eichrelevante Wägungen müssen Service Techniker, je nach Waagenausführung, eine Prüfung nach Norm EN45501 durchführen. Zurzeit wird eine Desktop-Applikation namens **Miracal** verwendet, das nicht für Ersteichungen ausgelegt ist, sondern für Kalibrierungen. Die Software wird sozusagen für geeichte Anwendungen missbraucht. Zudem werden alle Messergebnisse manuell in die Software eingetragen, was sehr fehleranfällig und einfach zu manipulieren ist.

## 1.2 Zweck und Ziel

Es soll eine Smartphone Applikation entwickelt werden, mit welcher der Service Techniker die Wäge-Prüfung nach Norm EN45501 durchführen kann. Die Mobile Applikation soll den Techniker durch den Prozess führen und die Werte von der Waage automatisch übertragen. Die Übertragung soll über Bluetooth Classic oder Bluetooth LE (Low Energy) funktionieren und über die METTLER TOLEDO internen Commands (SICS) kommunizieren. Die Kommunikation soll zwischen einem METTLER TOLEDO Bluetooth Adapter ACM360 und einem Samsung Galaxy S7 stattfinden. Nach abgeschlossener Prüfung wird ein Eichzertifikat erstellt und auf einem Fileserver als PDF abgelegt.

Dieses Projekt beinhaltet einerseits die korrekte Kommunikation über Bluetooth, sowie die erfolgreiche Kommunikation mit der Waage. Andererseits sollte die Mobile Applikation intuitiv und einfach zu bedienen sein.

In erster Linie geht es bei dieser Arbeit um eine Evaluation der Technologie (Adapter mit Bluetooth LE) in Verbindung mit Android. Aus diesem Grund liegt der Fokus der Arbeit in der Evaluation der Kommunikation dieser Technologie und der Implementierung eines Base Use Cases (Prozess Ersteichung). Da unser Rahmen zeitlich beschränkt ist und damit gerechnet werden kann, dass der Kommunikationsaufbau eine gewisse Zeit in Anspruch nehmen wird und der Eichprozess selber doch einiges an Implementationsaufwand verspricht, müssen gewisse Abstriche in anderen Punkten gemacht werden. Im Rahmen unserer Arbeit werden wir uns daher wohl oder übel auf ein gewisses Mass an Qualität beschränken müssen. Des weiteren werden wir eine Applikation schreiben, welche im Minimum mit einem Samsung Galaxy S7 (Android Version 8.0.0) und der zur Verfügung gestellten Waage umgehen kann. Eine weitere Einschränkung betrifft das GUI Testing, da das

GUI nicht im Fokus unserer Evaluation ist. Trotzdem möchten wir aber mittels Usability Tests sicherstellen, dass die Applikation am Ende einsatzfähig sein wird und der Service Techniker Freude daran hat.

Die METTLER TOLEDO bietet eine gigantische Auswahl an Waagentypen an. Damit wir nicht jede dieser erdenklichen Möglichkeit in unserer Applikation miteinbeziehen müssen, werden wir uns nur mit den Einbereichswaagen (Single-Range) auseinandersetzen.

### **1.3 Bereitgestellte Hardware**

- Samsung Galaxy S7, Android Version: 8.0.0
- Serial Bluetooth Adapter ACM360
- Tischwaage (Einfachwaage / single-range)

### **1.4 Lieferumfang**

- Android App für Ersteichung
- Evaluationsbericht
- Dokumentationen und Anleitungen

### **1.5 Annahmen und Einschränkungen**

Die Smartphone Applikation darf ausschliesslich zu internen Zwecken der Firma Mettler Toledo (Schweiz) GmbH verwendet werden.

### **1.6 Umfang**

Diese Arbeit ist in erster Linie eine Machbarkeitsstudie. In dieser evaluieren wir eine gewünschte Technologie des Kunden sowie deren Vor- und Nachteile dieser Technologie. Sofern es der Zeitumfang zulässt versuchen wir uns an Best-Practices im Software Engineering und Usability Design zu halten. Da dies in erster Linie jedoch eine Evaluation ist, wird auf eine produktive und komplett einsatzfähige Umsetzung der Applikation verzichtet. Ziel ist eine Applikation, welche für einen ersten Pilottest mit mehreren Personen über einen gewissen Zeitraum verwendet werden kann.

## 2 Aufgabenstellung

### Themenvorschlag für Studienarbeit Android Mobile Applikation

#### Ausgangslage

Für eich-relevante Wiegenungen müssen Service Techniker, je nach Waagen-Ausführung, eine Prüfung nach Norm EN45501 durchführen. Zurzeit wird eine Desktop-Applikation namens Miracal verwendet, welches für Kalibrationen entwickelt wurde, jedoch nicht für die Eichungen. Die Software wird sozusagen für geeichte Anwendungen missbraucht. Zudem werden alle Messergebnisse manuell in die Software eingetragen, was sehr fehleranfällig und einfach zu manipulieren ist.

#### Aufgabenstellung

Es soll eine Smartphone Applikation entwickelt werden, mit welcher der Service Techniker die Wäge-Prüfung nach Norm EN45501 durchführen kann. Die Mobile Applikation soll den Techniker durch den Prozess führen und die Werte von der Waage automatisch übertragen. Die Übertragung soll über Bluetooth Classic oder Bluetooth LE funktionieren (vorzugsweise BLE mit dem ACM360) und über die METTLER TOLEDO internen Comands (SICS) kommunizieren. Nach abgeschlossener Prüfung wird ein Eichzertifikat erstellt und auf einem Fileserver als PDF abgelegt.

Die Aufgabenstellung beinhaltet einerseits die korrekte Kommunikation über Bluetooth, sowie die erfolgreiche Kommunikation mit der Waage. Andererseits sollte die Mobile Applikation intuitiv und einfach zu bedienen sein.

#### Resultate

- Stabile Verbindung über Bluetooth (Plug & Play nach Pairing)
- Design nach MT Standard
- Geführter Prüfungsablauf
- Visuelle Darstellung der Eichfehlergrenze

#### Randbedingungen & Infrastruktur

Die benötigte Hardware (Terminal, Waage, Bluetooth-Adapter, Android Smartphone), die zu Testzwecken benötigt werden, wird von METTLER TOLEDO zur Verfügung gestellt.

#### Sonstige Hinweise

Die Smartphone Applikation darf ausschliesslich zu internen Zwecken verwendet werden.

Rev 001, 13.04.2018



---

## Aufgabenstellung Studienarbeit «Android-App für Waagen-Eichung»

---

### 1. Betreuer und Experte

Diese Arbeit mit dem Praxispartner Mettler Toledo durchgeführt und wird von Prof. Dr. Markus Stolze, HSR, IFS, [mstolze@hsr.ch](mailto:mstolze@hsr.ch) betreut.

*Industriepartner:*

Mettler Toledo (Schweiz) GmbH  
Im Langacher 44 CH-8606 Greifensee  
<http://www.mt.com>

*Ansprechperson:*

Franzis Perez  
Head of Engineering Geschäftsprozesse Industrie

### 2. Studierende

Diese Arbeit wird als Studienarbeit an der Abteilung Informatik durchgeführt von

- Valentina Merturi
- Stefano Kals

### 3. Ziele der Arbeit

Ziele der Arbeit sind dem Projektvorschlag der Mettler Toledo (Rev. 001, 13.04.2018, siehe Anhang) zu entnehmen. Diese Ziele sind durch die Studierenden in ein Anforderungsdokument abzubilden. Aus diesem Dokument ergibt sich die bewertungsrelevante Zielerreichung der Arbeit.

### 4. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen. Die zu erstellenden Dokumente bzw. Berichtsteile sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren.

### 5. Termine

Siehe Terminplan auf dem Skripteserver (Fachbereich/Bachelor-Arbeit\_Informatik/BAI/)

---

## 6. Beurteilung

Eine erfolgreiche Studienarbeit zählt 8 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von 30 Stunden.

Für die Beurteilung ist der verantwortliche Dozent zuständig.

| Gesichtspunkt   | Gewicht |
|---|---------|
| 1. Organisation, Durchführung   | 1/5     |
| 2. Berichte (Abstract, Mgmt Summary, technischer u. persönliche Berichte) sowie Gliederung, Darstellung, Sprache der gesamten Dokumentation | 1/5     |
| 3. Inhalt*)   | 3/5     |

\*) Die Unterteilung und Gewichtung von 3. Inhalt wird im Laufe dieser Arbeit präzisiert (u.A. durch die Bewertungsmatrix)

Im Übrigen gelten die Bestimmungen der Abteilung Informatik für Studienarbeiten.

Rapperswil, den 5. Dezember 2018.

Prof. Dr. Markus Stolze

## 3 Technologie

### 3.1 Bluetooth Low Energy (BLE)

Die Funktechnik, Bluetooth Low Energy (BLE), ist Teil der Bluetooth-4.0-Spezifikation vom Juni 2010[1]. Bluetooth LE unterstützt die typische Bluetooth-Architektur mit einer Host-CPU für den Upper Layer stack und einem Radio / Controller für den Lower Layer Stack. Beide sind über das Host Controller Interface (HCI) verbunden. Die Radio / Controller Hardware ist in den Varianten *Single-Mode Devices* oder *Dual-Mode Devices* verfügbar. Single-Mode-Hardware wird in Sensoren eingesetzt und unterstützt Bluetooth LE.

|                          |   |
|--------------------------|---|
| <b>Rollen</b>            | Geräte, die über Bluetooth LE kommunizieren, arbeiten entweder als Master oder als Slave. Diese Funktionen der Rollen bleibt mindestens für die Dauer der Verbindung bestehen.  |
| <b>States</b>            | Bluetooth LE kann unterschiedliche Zustände (StandBy, Advertiser, Scanner, Initiator, Connection) einnehmen. Die Master-Rolle ist ein Scanner (Scan für Advertiser-Events ohne Verbindungsaufbau) und Initiator (Aufbau einer Verbindung zu einem Advertiser). Die Eigenschaften eines Bluetooth LE Controllers sind abhängig vom Hersteller und haben eventuell Limitierungen betreffend der Anzahl unterstützter Verbindungen.                            |
| <b>Vergleich Classic</b> | Bluetooth Classic wird typischerweise zur Übertragung von Streaming Daten verwendet und verhält sich sehr robust bei Störungen durch andere Sender im gleichen Frequenzband. Kopfhörer sind die klassische Anwendung von Classic Bluetooth. Im Gegensatz zu Classic ist LE ideal für Produkte geeignet, die nur eine periodische Datenübertragung und kein kontinuierliches Datenstreaming erfordern. Dadurch eignet sich Bluetooth LE für IoT-Anwendungen. |

Zu ausführlicherem und tiefgründigerem Wissen sei auf den Artikel *Seamless and Secure Bluetooth LE Connection Migration* [2], den Artikel *Inside Bluetooth Low Energy* [3] sowie das Buch *Bluetooth Low Energy: A Technical Primer Your Guide to the Magic Behind the Internet of Things* [4] verwiesen.

## 3.2 Adapter (ACM360)

Der von der Firma METTLER TOLEDO entwickelte Bluetooth Adapter ACM360 ist eine industrielle Lösung zur Verbindung von Peripheriegeräten (wie Druckern oder andere Bluetooth 4.x Android Smart Geräten) und ist mit allen Mettler Toledo Terminals kompatibel. Der ACM360 ist so konzipiert, dass er eine ungestörte drahtlose Verbindung in rauen Umgebungen standhalten kann. Mehr Informationen können im Datasheet im Appendix 10.2 nachgelesen werden.

- Stromzufuhr**      Dadurch, dass der Adapter einen geringen Stromverbrauch hat, kann er direkt über die RS232-Schnittstelle [5] des Terminals (oder der Waage) mit Strom versorgt werden, was eine zusätzliche Stromversorgung überflüssig macht.
- Anwendungen**      Ursprünglich war der Adapter als Kabelersatz zwischen Drucker und Terminal (oder Waage) gedacht. Deshalb verarbeitet er auch bis zu 10 verschiedene Druckervorlagen. Doch nicht nur das, er kann auch mit mehreren Geräten verbunden werden. Des weiteren ist er für raue Industrieumgebungen gebaut und verfügt deshalb über einen IP65-Schutz.
- Konfiguration**      Die Rollenänderung des Adapters erfolgt über den Tastendruck. Über die Kontrollleuchte kann man den Status bzw. die aktuelle Rolle ermitteln. Genauere Angaben sind im User Manual des ACM360 Adapters von METTLER TOLEDO zu finden [6].
- Einschränkungen**      Der Adapter erlaubt Verbindungen in der Entfernung von max. 10 Metern. Zudem kann man nicht mehr als 16 Bytes Nachrichten übermitteln. Sollte eine Nachricht mehr Zeichen enthalten, so müssen weitere Notifikationen implementiert werden, um die gesamte Nachricht zu erhalten. Typischerweise werden solche Nachrichten (pro Zeile) immer mit einem Carriage-Return und Line-Feed abgeschlossen. Zudem wird die bestehende Verbindung nach 42 Sekunden automatisch beendet. Dieser Verbindungsabbruch ist nicht offiziell beschrieben. Wir sind im Internet vermehrt darauf gestossen und erlebten dies bei unseren Tests mit der Applikation. Die Dauer ist nicht überall gleich und wird vermutlich vom Bluetooth LE gesteuert.

## 3.3 SICS-Befehle

Alle METTLER TOLEDO Geräte unterstützen die standardisierten Befehle, die als SICS-Befehle bezeichnet werden. Man unterscheidet vier MT-SICS Levels, die abhängig vom Gerät sind.

- MT-SICS Level 0** Befehlssatz für die einfachsten Waagen (Wägezellen).
- MT-SICS Level 1** Erweiterung des Befehlssatzes für Standardwaagen, d.h. ohne integrierte Anwendungen.
- MT-SICS Level 2** Erweiterung des Befehlssatzes, speziell für die Produktfamilie.
- MT-SICS Level 3** Erweiterung des Befehlssatzes, für bestimmte Bereiche und Anwendungen der Produktfamilie.

Ein wesentliches Unterscheidungsmerkmal dieses Konzepts ist, dass die in MT-SICS Level 0 und 1 definierten Befehle für alle Waagen identisch sind. Sowohl die einfachste Waage als auch ein vollständig erweiterter Wägearbeitsplatz erkennt diese Befehle.

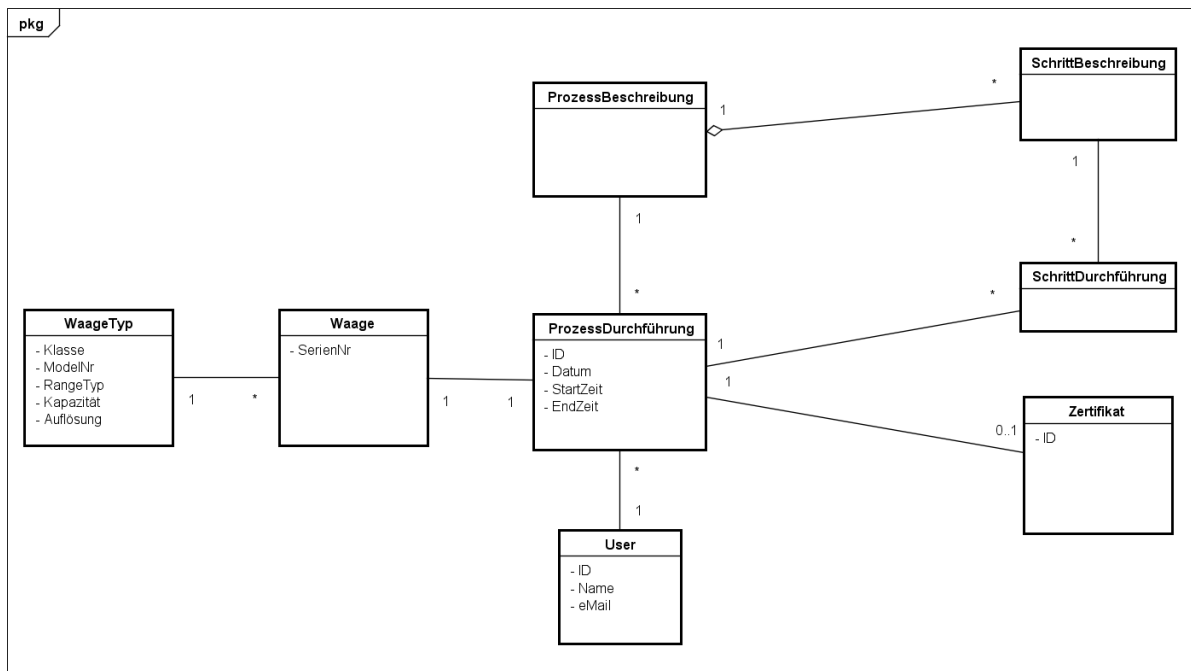
Obwohl alle Befehle gleich aufgebaut sind, sind deren Antworten unterschiedlich. Es gibt Befehle, die erhalten nur eine Zeile als Antwort, wobei andere bis zu sieben oder mehr Antwortzeilen erhalten. Jeder Befehl sowie die entsprechenden Antworten sind im SICS-Befehle-Dokument 10.3 aufgeführt.

# 4 Analyse

In diesem Kapitel befinden sich die Artefakte zu unserer Analyse, welche wir während dem Projekt fortlaufend ergänzt, erweitert und verbessert haben.

## 4.1 Domain Model

Das Domain-Model sieht wie folgt aus. Einige Punkte wurden während dem Projekt aus Zeitgründen vernachlässigt oder nach Absprache mit dem Kunden komplett weggelassen.



powered by Astah

Abbildung 4.1: Domain Model

## 4.2 Use Cases

Das Projekt besteht im Prinzip aus dem einzigen Use Case: Eine Ersteichung erstellen. Wir haben diesen Use Case jedoch in einzelne kleinere Use Cases unterteilt, damit wir eine bessere Übersicht über das Projekt haben und unsere Aufgaben einfacher verteilen können. Die einzelnen Use Cases wurden mit dem Kunden und die dafür verantwortlichen Personen in der ersten Kickoff-Sitzung besprochen und abgenommen. Man hat alle einzelnen Punkte ausführlich diskutiert und im Protokoll festgehalten.

### 4.2.1 UC1: Verbindung zum Bluetooth-Adapter

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Mittels Mobile-Applikation soll eine Bluetooth (LE) Verbindung zwischen dem Smartphone und dem Terminal bzw. der Waage entstehen.             |
| <b>Voraussetzungen</b>        | Der Vor- und Nachname muss beim Starten der App angegeben sein, sonst kommt man nicht zum nächsten Schritt.                                   |
| <b>Erfolgreiches Szenario</b> | Nach Eingabe der erforderlichen Daten, kann der Service-Techniker ein zur Verfügung stehendes Gerät auswählen und eine Verbindung herstellen. |
| <b>Umsetzung</b>              | Die Verbindung zum Bluetooth-Adapter wurde vollständig umgesetzt.   |

### 4.2.2 UC2: Konfiguration der Waage auslesen

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Über die Bluetooth Verbindung sollen die Konfigurationsdaten der Waage ausgelesen werden. Diese Informationen erhält man, indem man der Waage spezifische SICS-Befehle sendet (I0, I1, I2, I3, I4). |
| <b>Voraussetzungen</b>        | Es besteht eine Verbindung zum gewünschten Gerät.   |
| <b>Erfolgreiches Szenario</b> | Wenn man sich im Gewichtsfenster befindet, werden alle Daten (Seriennummer, max. / min. Gewicht, e-Wert, Geräte-Name) aus der Waage ermittelt und richtig dargestellt.                              |
| <b>Umsetzung</b>              | Das Auslesen der Waagen-Konfiguration wurde vollständig umgesetzt.  |

### 4.2.3 UC 3: SICS-Befehle an die Waage senden

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Die Applikation muss SICS-Befehle an die Waage senden und eine Antwort von der Waage erhalten können. Die SICS-Befehle werden von der Firma METTLER TOLEDO festgelegt (genauere Beschreibung der Befehle findet man im Anhang). |
| <b>Voraussetzungen</b>        | Es besteht eine Verbindung zum gewünschten Gerät.   |
| <b>Erfolgreiches Szenario</b> | Man kann die Waage mit den aufgeführten Tasten im Gewichtsfenster nullieren, tarieren und dessen Tara löschen. Intern werden die entsprechenden SICS-Befehle an die Waage gesendet und ausgeführt.                              |
| <b>Umsetzung</b>              | Das Senden der SICS-Befehle an die Waage wurde vollständig umgesetzt.   |

### 4.2.4 UC4: Ersteichungsprozess durchführen

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Der Service-Techniker muss durch den gesamten Ersteichungsprozess durchgeführt werden. Jeder Schritt der Ersteichung muss so abgebildet werden, dass das Ergebnis vom Service-Techniker als erfolgreiche Prüfung durchgehen kann. |
| <b>Voraussetzungen</b>        | Es besteht eine Verbindung zum gewünschten Gerät.   |
| <b>Erfolgreiches Szenario</b> | Der Benutzer wird Schritt für Schritt durch die Ersteichung geführt und die einzelnen Schritte sind klar verständlich.  |
| <b>Umsetzung</b>              | Der Ersteichungsprozess wurde vollständig umgesetzt, jedoch beinhaltet der Prozess einige Einschränkungen.  |

### 4.2.5 UC5: Benutzerdaten verwalten

|                     |  |
|---------------------|--|
| <b>Aktoren</b>      | Service-Techniker  |
| <b>Beschreibung</b> | Die Benutzerdaten, die zuerst vom Service-Techniker in der Applikation eingegeben werden müssen, sollen auch später modifiziert werden können. |

|                               |   |
|-------------------------------|---|
| <b>Voraussetzungen</b>        | Keine Voraussetzungen vorhanden.  |
| <b>Erfolgreiches Szenario</b> | Der Service-Techniker startet die App und muss direkt seinen Namen eintragen. Die Felder sind leer und sobald er auf 'Login' klickt, werden die Daten intern gespeichert und beim nächsten Start angezeigt. |
| <b>Umsetzung</b>              | Die Benutzerdaten können nur beim Starten der App geändert werden und nicht später während der Benutzung.   |

#### 4.2.6 UC6: Zertifikat ablegen

|                               |  |
|-------------------------------|--|
| <b>Aktoren</b>                | Service-Techniker  |
| <b>Beschreibung</b>           | Nach einer Ersteichung soll die erstellte PDF-Datei (Konformitätszertifikat) auf einem Server abgelegt werden (in diesem Projekt wird der HSR-Server verwendet).   |
| <b>Voraussetzungen</b>        | Der Benutzer hat die Ersteichung erfolgreich abgeschlossen.  |
| <b>Erfolgreiches Szenario</b> | Der Benutzer beendet die erfolgreich durchgeführte Ersteichung und überprüft die Zusammenfassung. Sobald er weiter geht, wird ein Zertifikat als PDF-Datei erstellt, die er bei Bedarf direkt anschauen kann. In diesem Schritt muss der Benutzer das Zertifikat mittels Schaltfläche auf den Server hochladen und kann anschliessend den Prozess beenden. |
| <b>Umsetzung</b>              | Das Zertifikat wird erfolgreich erstellt und hochgeladen.  |

#### 4.2.7 UC7: Kundendaten verwalten

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Die Kundendaten werden vom Service-Techniker vor der Ersteichung manuell eingetragen. Diese Daten müssen modifiziert werden können. |
| <b>Voraussetzungen</b>        | Keine Voraussetzungen vorhanden.  |
| <b>Erfolgreiches Szenario</b> | Der Benutzer kann die Kundendaten beliebig anpassen.  |
| <b>Umsetzung</b>              | Wurde aus Zeitgründen und in Absprache mit dem Kunden nicht umgesetzt.  |

#### 4.2.8 UC8: Konformitätszertifikat erstellen

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Nach erfolgreichem Abschluss der Ersteichung muss eine PDF-Datei (Konformitätszertifikat) erstellt werden können. In diesem Zertifikat werden alle wichtigen Informationen zur Ersteichung eingetragen. |
| <b>Voraussetzungen</b>        | Der Benutzer hat die Ersteichung erfolgreich abgeschlossen.   |
| <b>Erfolgreiches Szenario</b> | Der Benutzer kann ein Konformitätszertifikat erstellen, nachdem die Ersteichung abgeschlossen und die Zusammenfassung überprüft wurde.  |
| <b>Umsetzung</b>              | Das Erstellen eines Zertifikats wurde vollständig umgesetzt.  |

#### 4.2.9 UC9: Alle verfügbaren SICS-Befehle der Waage auslesen

|                               |   |
|-------------------------------|---|
| <b>Aktoren</b>                | Service-Techniker   |
| <b>Beschreibung</b>           | Der Service-Techniker soll in einer separaten Ansicht alle möglichen SICS-Befehle sehen und ausführen können. |
| <b>Voraussetzungen</b>        | Es besteht eine Verbindung zum gewünschten Gerät.   |
| <b>Erfolgreiches Szenario</b> | Der Benutzer kann im Gewichtsfenster alle verfügbaren SICS-Befehle abrufen und ausführen.                     |
| <b>Umsetzung</b>              | Wurde aus Zeitgründen und in Absprache mit dem Kunden nicht umgesetzt.  |

### 4.2.10 Use Case Diagramm

Das Use Case Diagramm wurde wie folgt zusammengestellt. Die Farben haben unterschiedliche Bedeutungen. Alle Use Cases, die grün eingefärbt sind, wurden komplett implementiert. Die Use Cases in der gelben Farbe waren optional und wurden entweder implementiert oder mit Begründung weggelassen. Der rote Use Case ist nice-to-have und würde nur dann implementiert werden, wenn noch genügend Zeit vorhanden wäre.

Ob die einzelnen Use Cases plangemäss implementiert wurden, ist im oberen Kapitel genauer beschrieben.



powered by Astah

Abbildung 4.2: Use Case Diagramm

## 5 Architektur

Dieser Teil der Arbeit beschreibt unsere erarbeitete Softwarearchitektur, die verwendeten Bibliotheken sowie Metriken am Ende des Projekts.

### 5.1 Deployment

Die Design-Entscheidung wurden uns mehrheitlich vom Kunden abgenommen. In der Aufgabenstellung ist klar ersichtlich mit welcher Hardware wir arbeiten sollen. Auf Grund dessen, dass die Service-Techniker alle Android-Smartphones benutzen, war auch diese Entscheidung bereits für uns getroffen worden. In nachfolgender Grafik 5.1 ist ersichtlich, wie unser Aufbau aussieht.

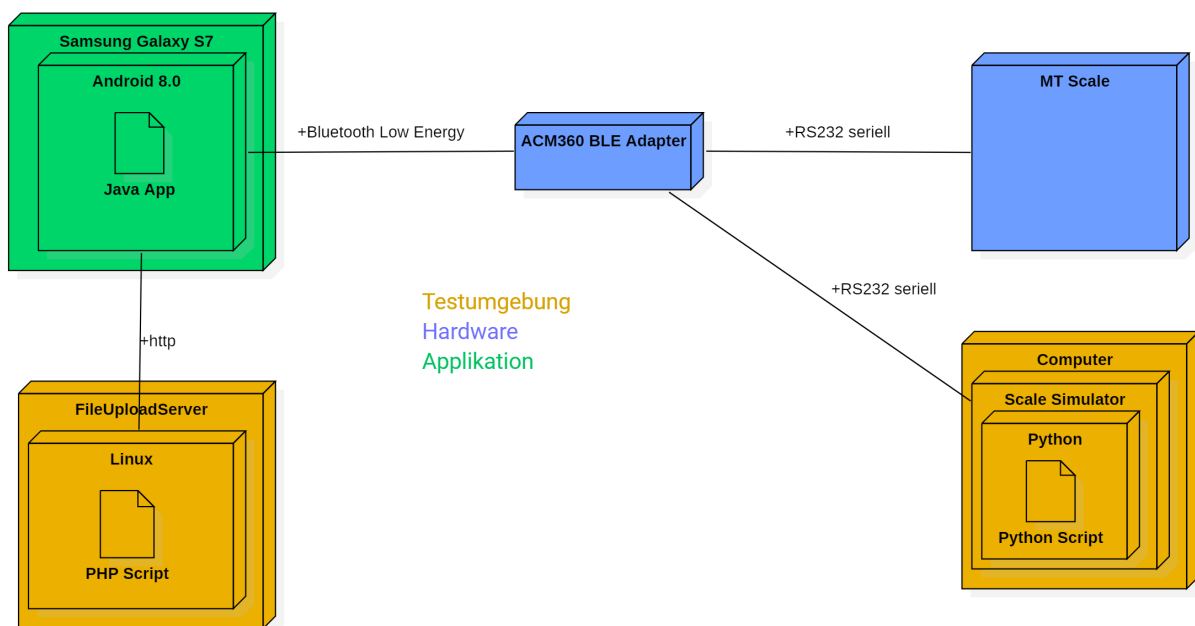


Abbildung 5.1: Deployment Diagramm

## 5.2 Code Struktur

In nachfolgender Grafik 5.2 ist ersichtlich, wie unsere Code Struktur der Android Applikation aussieht, wobei die Services unsere Businesslogik darstellen und Domainklassen ohne UI Funktionalität sind. Ausserdem ist ersichtlich, welche Bibliotheken verwendet wurden.

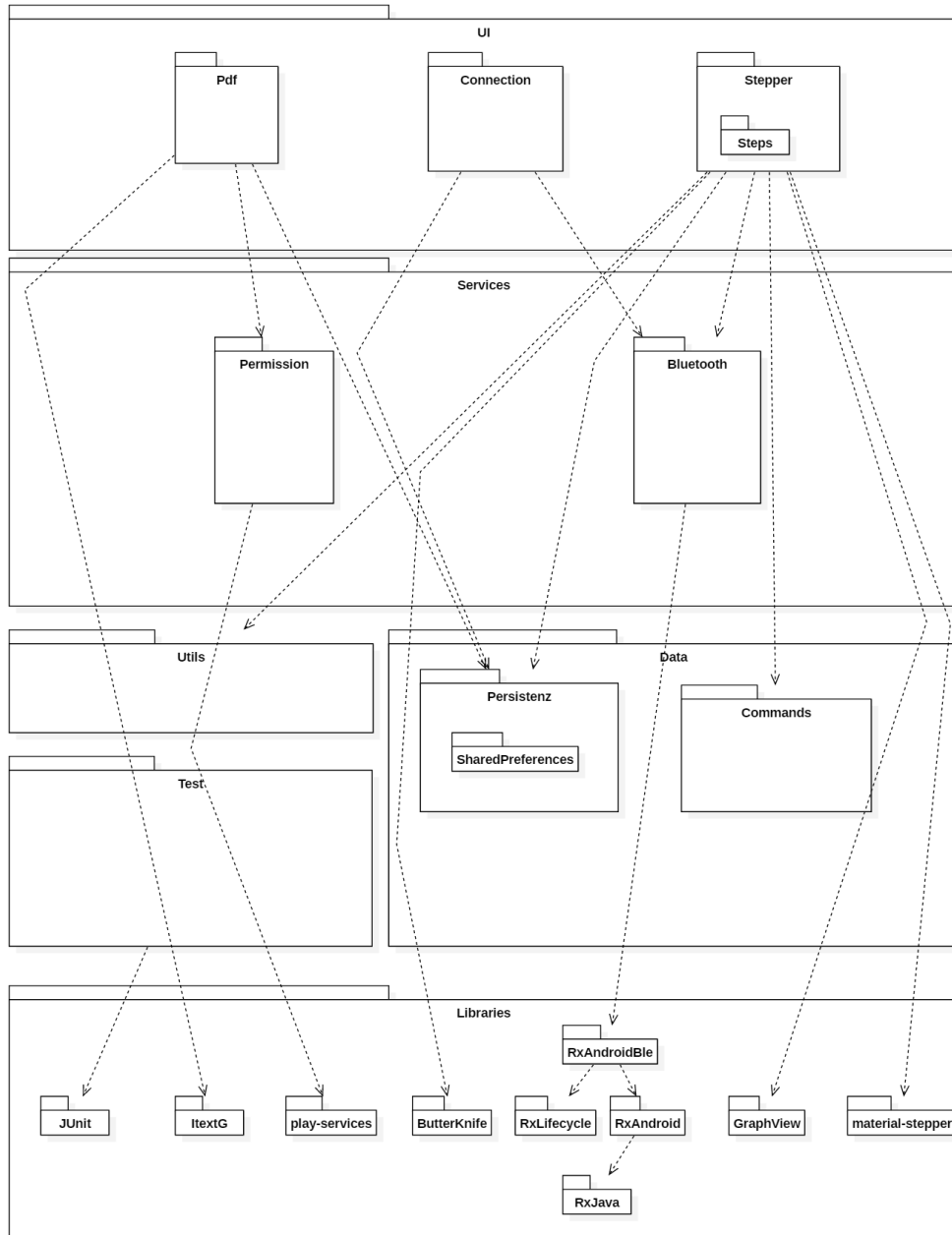


Abbildung 5.2: Package Diagramm

## 5.3 Libraries

Die verwendeten Bibliotheken werden im Folgenden aufgelistet und beschrieben. Ausserdem befinden sich die dazugehörigen Lizenzen der Bibliotheken im Appendix 10.6.

- |                                 |  |
|---------------------------------|--|
| <b>RxAndroidBle</b>             | Zum einfacheren Umgang des Bluetooth Low Energy Handlings in Android Applikationen, ist diese Bibliothek bestens geeignet. [7]                     |
| <b>RxAndroid</b>                | Diese Bibliothek ermöglicht ein vereinfachtes Scheduling und Observing auf dem Main Thread der Applikation. [8]                                    |
| <b>RxJava</b>                   | Diese Bibliothek ermöglicht es, asynchrone und eventbasierte Programme in Kombination mit dem Observer-Pattern zu erstellen. [9]                   |
| <b>Android Material Stepper</b> | Zur Anzeige einzelner Steps, via Tabs und Fragments im Material Design Style, wird diese Bibliothek eingesetzt. [10]                               |
| <b>ButterKnife</b>              | Diese Bibliothek ermöglicht das vereinfachte Auffinden von Ressourcen, Komponenten und Views. Es macht zudem den Code lesbarer und einfacher. [11] |
| <b>ItextG</b>                   | Das Erstellen und Anzeigen von PDFs in Android wird mit diese Bibliothek ermöglicht. [12]  |
| <b>GraphView</b>                | Mit dieser Bibliothek kann man verwenden, um mathematische Diagramme plotten und anzeigen. [13]  |
| <b>Play Services</b>            | Diese Bibliothek ermöglicht es, Permissions wie bei Google Maps mittels View und direkter User Interaktion anzuzeigen. [14]                        |

## 5.4 Metriken

In diesem Abschnitt befinden sich die Metriken wie *Lines of Code*, Anzahl Klassen und Anzahl *xml* Layout Dateien [15]. In der Grafik 5.3 sind diese Werte ersichtlich. Genaue Angaben zu den verschiedenen Dateien und ihren Namen befinden sich im Appendix unter 10.7. Die verschiedenen Metriken wurden mittels frei verfügbaren Android Studio[16] (JetBrains IntelliJ [17]) Plugins erstellt. Namentlich zu nennen sind dabei "SStatistic"[18] und "CodeMR"[19].

| Extension                                 | Count | Size SUM | Size MIN | Size MAX | Size AVG | Lines  | Lines MIN | Lines MAX | Lines AVG |
|---|-------|----------|----------|----------|----------|--------|-----------|-----------|-----------|
| <b>bat</b> (BAT files)                    | 1x    | 2kB      | 2kB      | 2kB      | 2kB      | 84     | 84        | 84        | 84        |
| <b>bin</b> (BIN files)                    | 12x   | 51,229kB | 0kB      | 14,616kB | 4,269kB  | 109386 | 1         | 44482     | 9115      |
| <b>gitignore</b> (GITIGNORE files)        | 1x    | 0kB      | 0kB      | 0kB      | 0kB      | 1      | 1         | 1         | 1         |
| <b>gradle</b> (GRADLE files)              | 2x    | 0kB      | 0kB      | 0kB      | 0kB      | 26     | 1         | 25        | 13        |
| <b>java</b> (Java classes)                | 29x   | 167kB    | 0kB      | 15kB     | 5kB      | 4547   | 6         | 413       | 156       |
| <b>lock</b> (LOCK files)                  | 5x    | 0kB      | 0kB      | 0kB      | 0kB      | 5      | 1         | 1         | 1         |
| <b>md</b> (MD files)                      | 1x    | 0kB      | 0kB      | 0kB      | 0kB      | 2      | 2         | 2         | 2         |
| <b>pro</b> (PRO files)                    | 1x    | 0kB      | 0kB      | 0kB      | 0kB      | 21     | 21        | 21        | 21        |
| <b>properties</b> (Java properties files) | 5x    | 1kB      | 0kB      | 0kB      | 0kB      | 31     | 0         | 14        | 6         |
| <b>xml</b> (XML configuration file)       | 34x   | 119kB    | 0kB      | 15kB     | 3kB      | 2305   | 4         | 322       | 67        |

Abbildung 5.3: Übersicht Metriken

### 5.4.1 Lines of Code

**Java**                      Insgesamt haben wir in unserem Projekt **4547 Zeilen** Code geschrieben, die in insgesamt **29 Klassen** unterteilt sind.

**XML**                        Mit einem Total von **2305 Zeilen** Code und insgesamt **34 XML Dateien** haben wir auch beim Layout grosse Zahlen erreicht.

### 5.4.2 Kopplung und Kohäsion

Mit der Ausnahme von drei Klassen (BluetoothLEService, ScanActivity und StepActivity), überzeugt der Code mit geringer Kopplung und hoher Kohäsion, welche dem erwarteten Ziel von guten Softwareprojekten entspricht. Der Grund, der oben erwähnten Klassen mit hoher Kopplung, ist vertretbar. Da dort die Kernelemente zur Kommunikation, sowie des Prozessablaufs stecken und die meisten anderen Android Klassen (wie Fragments und Activities) darauf angewiesen sind und eine hohe Kopplung (verglichen zum Rest zu diesem Zeitpunkt) nicht ausgeschlossen werden konnte. Dies ist sogar notwendig, um die Komplexität in diesem Fall durch weniger Klassen geringer zu halten. Die grafischen Darstellungen zu den verschiedenen Werten befinden sich im Appendix unter 10.7.

### 5.4.3 McCabe

Die gewichtete Summe aller Klassen, ist mit der McCabe Komplexität, auch Weighed Method Count (WMC) genannt, in unserem Fall bei allen Klassen auf einem sehr tiefen Niveau.

Die einzige Ausnahme ist die BluetoothLEService Klasse, die eine mittlere Komplexität aufweist, was durchaus vertretbar ist. Im Grossen und Ganzen entspricht dies einem gut wiederverwendbaren und wartbaren Code. Dies bedeutet auch, dass der Code weniger anfällig auf Fehler und grosse Änderungen ist. Eine grafische Übersicht ist wiederum im Appendix unter 10.7 zu finden.

## 5.5 Eingesetzte Patterns

### 5.5.1 Observer Pattern

Die Bibliotheken RxJava [9], RxAndroid [8] sowie RxAndroidBle [7] verwenden Subscriptions [20], um auf Events zu reagieren und erweitern das Observer Pattern so, dass mit Hilfe einfacher Befehle gearbeitet werden kann. Wir gehen davon aus, dass die Bibliotheken das korrekt implementiert haben. Für Informationen zu den genannten Bibliotheken sei auf den Abschnitt 5.3 weiter oben verwiesen.

### 5.5.2 Adapter Pattern

Für den Bluetooth Service Test wurde eigens ein Adapter geschrieben, welcher gemockte Daten liefert, vom echten Service ableitet und das gleiche Interface implementiert. Zu der Testbeschreibung siehe auch 6.9.

# 6 Qualitätsmanagement

## 6.1 Nicht funktionale Anforderungen

### 6.1.1 Usability

Der Text der Applikation ist von einem Meter Entfernung lesbar. Die Bedienbarkeit ist mit einer Hand möglich, jeder Button kann mit dem Daumen erreicht werden. Der Ablauf ist mit einzelnen Schritten geführt und der Abschluss des Prozesses ist durch einen Beenden Button klar ersichtlich.

### 6.1.2 Stability

Während dem Zeitraum der Prozessdurchführung soll stets eine Verbindung zum Adapter vorhanden sein. Diese Verbindung muss während einer Prozessdurchführung gewährleistet sein.

### 6.1.3 Portability

Es gibt keine Rahmenbedingung, welches SDK verwendet werden muss. Da aber neuere SDKs besser mit Bluetooth umgehen können, wir nur ein kleines Zielpublikum erreichen müssen und die Hardware der Enduser kennen, haben wir uns für ein relativ hohes SDK entschieden. Das minimale SDK[21], welches verwendet wird, ist SDK 25 [22]. Dies führt dazu, dass unsere Applikation ab Android Version 7.1 läuft, welches ungefähr 14.2% entspricht. Da die Applikation nur intern Verwendung findet und dies nur auf einem spezifischen Gerät, bei dem diese Mindest-Version gewährleistet ist, wird keine anderweitige Portabilität gewährleistet. Dies wurde auf einem Samsung Galaxy S7 mit Android 8.0 getestet und ist darauf lauffähig.

### 6.1.4 Security

Bei der Übertragung der Daten auf den File-Server wurde auf Sicherheitsmassnahmen wie SSL/TLS (https) verzichtet. Es ermöglicht uns, den Fokus auf die Technologie zu legen und die Vereinfachung der Testumgebung. Ausserdem werden keine sensiblen Kundendaten übertragen.

In der Android Applikation wurde jedoch darauf geschaut, dass die Daten nur isoliert intern und nicht von anderen Applikationen verwendet werden können. Dazu wurden Shared-

Preferences im Private Mode angelegt, sowie ein lokaler Broadcast Manager für Messages verwendet, welche nur dieser Applikation zur Verfügung steht.

### 6.1.5 Accuracy

Mittels der Bluetooth Low Energy Library ist stets gewährleistet, dass die Daten von der Waage zur Applikation korrekt übertragen werden, sofern Daten vorhanden sind und eine Verbindung mit dem Adapter besteht. Werden bei einem benötigten Vorgang wie Gewicht Lesen keine Daten übermittelt, wird dies dem Benutzer mittels eines Hinweises im Feld kenntlich gemacht.

### 6.1.6 Accessibility

Obwohl Accessibility keine Anforderung an unsere Applikation war, hat es uns doch interessiert und wir haben mittels **Google Accessibility Scanner** [23] des Google Play Stores [24] unsere Applikation auf Fehler durchsucht. Bei einzelnen Elementen kann durchaus noch ein stärkerer Kontrast erzeugt werden. Für den Kunden scheint dies zur Zeit jedoch kein Problem oder Bedürfnis zu sein.

## 6.2 Wireframes

### 6.2.1 Erster Entwurf

Folgende Darstellung zeigt den ersten Wireframe-Entwurf an, der noch bis vor dem Usability Test 2 galt.

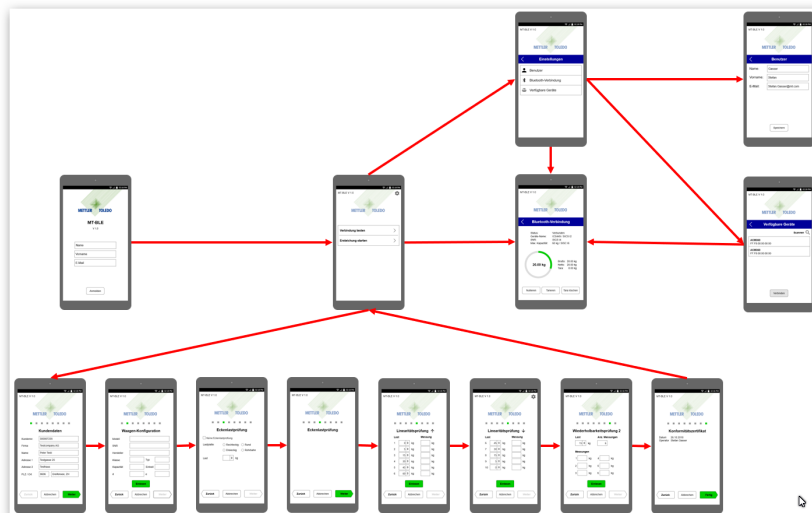


Abbildung 6.1: erster Entwurf

In der Abbildung in der Mitte ist der Einstiegsscreen zu sehen. Der Benutzer wird aufgefordert, seine Benutzerangaben einzutragen. Danach gelangt man zum Hauptfenster, wo man

die Verbindung testen, die Ersteichung starten oder in die Einstellungen gelangen kann. In den Einstellungen hat man die Möglichkeit die Benutzereingaben zu ändern, die verfügbaren Geräte (ACM360-Adapter) zu sehen und die Verbindung zu testen.

Wählt man im Hauptfenster «Ersteichung starten», so wird man durch den Ersteichungsprozess (in insgesamt acht Schritten) geführt. Jede Eingabe wird auf ihre Korrektheit überprüft und erst am Schluss wird das Konformitätszertifikat erstellt. Dabei wird eine PDF-Datei generiert, die anschliessend auf den von uns verwendeten HSR-Server hochgeladen wird.

## 6.2.2 Zweiter Entwurf

Nachdem man den Usability Test 2 mit den Benutzern durchgeführt hat, erkannte man einige unglückliche Konstruktionen. Diese möchte man nun mit dem zweiten Entwurf verbessern. Dementsprechend sieht der neue Prozess wie folgt aus.

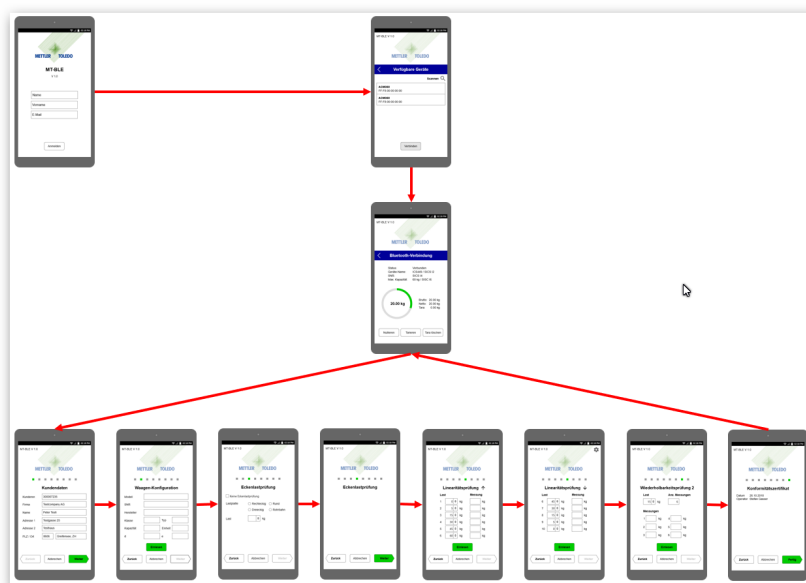


Abbildung 6.2: zweiter Entwurf

Was uns sofort auffällt, ist, dass man weniger Fenster hat. Man hat keine unnötigen Funktionen und Tasten mehr, somit ist der Vorgang eindeutig. Der Benutzer startet hier ebenfalls mit der Eingabe der Benutzerdaten. Danach wird er aufgefordert, den gewünschten Adapter auszuwählen und sich damit zu verbinden. Sobald die Verbindung besteht, gelangt der Benutzer zum nächsten Fenster. Dort wird ihm das aktuelle Gewicht angezeigt und er hat die Möglichkeit direkt die wichtigsten SICS-Befehle auf der Waage auszuführen. Zu diesen Befehlen gehören das Trieren, Nullieren sowie Tara löschen. Zudem kann er nur von dieser Ansicht aus eine Ersteichung durchführen. Beim Prozess der Ersteichung hat sich jedoch nichts geändert und ist deshalb gleich wie beim ersten Entwurf.

## 6.3 Usability Tests

**Usability Test 1** Der erste Usability Test war mehr ein Hallway Testing. Wir haben einem Benutzer (Service-Techniker) die Wireframes als clickable Exe-Datei zur Verfügung gestellt und ihm die Aufgabe gestellt, eine Ersteichung durchzuführen. Der Test dauerte nur wenige Minuten und wir konnten sehr schnell feststellen, dass der Prozess an sich so den Anforderungen entspricht. Der Benutzer fand sich sehr schnell zurecht und er konnte die Aufgabe meistern. Durch diesen Usability Test erkannten wir, wo die Schwierigkeiten dieser Applikation liegen und mussten uns somit bei diversen Schritten einschränken.

**Usability Test 2** Den zweiten Usability Test führten wir in der Construction Phase durch. Unsere Aufgabenstellung war folgende:

*«Stell dir vor, du wärst jetzt für eine Ersteichung bei einem Kunden (hier im Büro) disponiert. Du hast auf deinem Smartphone unsere Mobile Applikation MT-BLE bereits installiert und musst nun damit arbeiten. Bitte führe eine Ersteichung durch.»*

Diesen Test führten wir mit drei verschiedenen Personen aus und es stellte sich heraus, dass einige Fenster, die wir zu Beginn definiert hatten, nicht notwendig waren. Somit würde der gesamte Prozess der Applikation wesentlich einfacher werden. Mit dem Feedback aller Personen haben wir neue Ziele festgelegt.

## 6.4 Code-Standards

Wir haben uns am Google Java Style Guide [25] gehalten. Nicht jeder Bereich entspricht genau den Richtlinien, trotzdem haben wir das so gut wie möglich versucht weiterzuführen.

**Quelldatei** Der Name der Quelldatei besteht aus einem case-sensitiven Namen der enthaltenen obersten Klasse (von der genau eine existiert) und der Erweiterung '.java'. Die Quelldateien werden in UTF-8 kodiert. Zudem kommen keine Leerzeichen im Namen einer Quelldatei vor und werden in Zeichenketten escaped. Allgemein wird für jedes Zeichen eine spezielle Escape-Sequenz verwendet und keine Oktal-Schreibform.

**Struktur der Quelldateien** Eine Quelldatei beinhaltet Lizenz- oder Urheberrechtshinweise (falls vorhanden), Package-Statements, Import, Statements und genau eine Top-Level-Klasse, in dieser Reihenfolge. Zudem trennt eine Leerzeile

jeden vorhandenen Abschnitt.

Jede Top-Level-Klasse befindet sich in einer eigenen Quelldatei.

|                              |  |
|------------------------------|--|
| <b>Formatierung</b>          | Den Code haben wir gemäss Standard-Formatierung im Android Studio formatiert. Für jedes Statement (if, else, for while, etc.) werden geschweifte Klammern verwendet, auch wenn das Statement aus einer Zeile besteht oder leer ist. Nach jedem Statement folgt ein Zeilenumbruch. Die Zeilen sind limitiert auf 100 Zeichen, ausser es ist überhaupt nicht möglich. Jede Variablendeklaration ist in einer eigenen Zeile.  |
| <b>Namensgebung</b>          | Die Namen für Variablen, Methoden und Klassen sind aussagekräftig. Es werden keine Präfixe oder Suffixe verwendet. Package Namen beinhalten Kleinbuchstaben, wobei aufeinanderfolgende Wörter einfach aneinandergereiht werden. Klassennamen werden in <i>UpperCamelCase</i> geschrieben. Testklassen werden so benannt, dass sie mit dem Namen der Klasse, die getestet wird, beginnen. Methodennamen, nicht-konstante Feldnamen, Parameter sowie lokale Variablen werden in <i>lowerCamelCase</i> geschrieben. Konstantennamen verwenden <i>constant case</i> und beinhalten nur Grossbuchstaben, wobei die Worte mit Unterstrichen getrennt werden. |
| <b>Programming Practices</b> | Jede Exception wird abgefangen und im Log eingetragen oder eine Fehlermeldung angezeigt, wenn nötig.   |
| <b>Javadoc</b>               | Die Dokumentierung einzelnen Blöcke erfolgt mittels Javadoc. Diese Beschreibungen beginnen mit <code>/**</code> und enden mit <code>*/</code> . Beim entsprechenden Block wird die Funktionalität, sowie die Rückgabewerte und / oder Parameter beschrieben. Eine Beschreibung wird für jede public Klasse und jedem public Member vorausgesetzt und nur bei Methoden vorhanden, die keine Supertype-Methode überschreibt  |

## 6.5 Code-Review

Während des Projekts konnte der Code zweimal von einem Institutsmitarbeiter begutachtet werden. Der erste Termin war beim Abschluss des Architekturprototyps beim Meilenstein *M2 End of Elaboration*. Dabei kamen einige wichtige Punkte, die es zu beheben galt, zum Vorschein. Dies wurden nach dem Review wunschgemäss umgesetzt und in die Applikation eingepflegt. Der zweite Termin war unmittelbar nach dem Meilenstein *M3 Architektur / Usability Test 2*. Dort wurde auf die Verbesserungen und Umsetzungen des ersten Code Reviews eingegangen, welche wir zu einem grossen Teil direkt umsetzen konnten, sowie auf kleinere weitere Punkte, welche es noch umzusetzen galt.

## 6.6 Lint

Zur Qualität des Codes setzen wir auf Lint [26], welches im CI/CD Prozess integriert ist. Das finale Ergebnis ist im Abschnitt 10.9 zu sehen.

## 6.7 Commits

Wir haben uns zum Ziel gesetzt, viele und kleine Commits sowie Branches für Features, Bugfixes sowie Refactorings zu erstellen und einen Merge-Request auf den Master zu erzeugen. Nur dieser relevante Merge in den Master Branch wird dann mittels Gitlab CI/CD getestet und schliesslich erstellt. Bevor dies geschieht, wird der Merge-Request jedoch dem jeweils anderen zugewiesen, um Änderungen anzusehen und das 4-Augen-Prinzip einzuhalten.

## 6.8 CI/CD

### Continuous Integration (CI)

Wir verwenden Continuous Integration (CI) mit Pipelines auf GitLab [27] zur Steigerung der Qualität der Applikation durch automatisiertes Linting. Ausserdem ermöglicht es die Ausführung automatisierter Tests.

Das Linting [26] sowie die Android Unit Tests laufen vollständig automatisiert ab.

Da uns leider keine geeignete Hardware zu Beginn der Studienarbeit in den Räumen oder privat zur Verfügung stand, konnten wir via Continuous Integration keine funktionalen Android Tests durchführen, welche Teil des Continuous Integration Zyklus gewesen wären. Diese Tests setzen einen Emulator mit einem Intel® Prozessor sowie unterstützter Intel® VT-x Virtualisierung voraus. Die HSR Server, welche wir erhalten haben, ermöglichten dies nicht und auch für die öffentlichen Runner von Gitlab war keine dem entsprechende Hardware vorhanden. Aus diesen Gründen mussten wir hier auf das funktionale Testing verzichten und dies von Hand via Smartphone oder PC Emulator selbst vollziehen.

### Continuous Delivery (CD)

Wir verwenden Continuous Delivery mit Pipelines auf GitLab. Bei jeglichen Commits auf den Master-Branch wird die Applikation automatisch gebildet und als Artefakt zum Download angeboten. Das entstandene Artefakt ist eine apk [28] Datei welches via ADB [29] auf einem Android Smartphone installiert werden kann.

## 6.9 Tests

Unser Testplan besteht aus Unit Tests, welche die *Commands* und *Utils* Klassen beinhalten, die die nicht UI relevanten Sachen sowie funktionellen Tests, welche die Services mittels Emulator testen. Besonders zu erwähnen gilt es hierbei, dass wir für den Bluetooth Service Test einen Adapter zur vorhandenen BluetoothService Klasse schreiben mussten, um gemockte Daten zu erfassen. Die erfolgreich durchlaufenden Tests nach dem Code Freeze sind im Appendix 10.9 ersichtlich.

# 7 Development

Dieser Teil beschreibt die Tools, das Vorgehen und die Umsetzung der Applikation.

## 7.1 Tools

### 7.1.1 Android Studio

Um Android Applikationen zu erstellen, eignet sich das eigens von Google bereitgestellte Android Studio [16], welches auf IntelliJ IDEA [17] aufbaut.

### 7.1.2 Visual Studio Code / TexStudio

Zur Dokumentation verwenden wir LaTeX [30], welches wir wahlweise entweder mittels Visual Studio Code oder TexStudio [31] bearbeiten und erstellen können. Dazu benötigen wir jedoch noch einen Tex Compiler. Wir haben uns für MikTeX [32] entschieden, da dies eine komfortable LaTeX Paketverwaltung mit sich bringt und Pakete im Hintergrund bei Bedarf nachlädt.

## 7.2 Architekturprototyp

Da zu Beginn des Projekts noch wenig Android Erfahrung vorhanden war, wurde bewusst auf die noch sehr neue und junge Programmiersprache Kotlin [33] verzichtet und der Fokus ganz auf die Android Java Programmierung gesetzt. Ein weiterer Grund, der dagegen gesprochen hat, war die Vermutung nach fehlenden Dokumentationen und Beispielen in Kotlin.

Es wurde sehr früh damit begonnen, den ersten Durchstich zu erreichen. Dafür wurden die ersten Konzepte in Android umgesetzt, um den Architekturprototypen zu realisieren. Dabei konnte gezeigt werden, dass die Kommunikation mit dem Adapter zur Waage und zurück möglich ist.

Zusätzlich zur Kommunikation wurde mittels Prototyp das Erstellen eines PDFs sowie das Hochladen auf einen Server demonstriert, da dies Kernfunktionen des Produkts darstellen. Dass diese Komponenten funktionieren, konnte auch mit dem Prototyp gezeigt werden.

## 7.3 Applikation

Im ersten Teil der Konstruktionsphase wurde die Applikation nach Wunsch soweit implementiert, um einen ersten richtigen Usability Test zu machen. Die Applikation wurde mit den notwendigen Ersteichungsschritten, den Settings sowie dem Verbindungstest ausgestattet. Die Wireframes sind im Abschnitt 6.2 und die Usability Tests im Abschnitt 6.3 beschrieben.

Im zweiten Teil der Konstruktionsphase, wurde anhand des zweiten Usability Tests die Verbesserungen eingepflegt. Dabei ist besonders zu vermerken, dass der Kunde nun keine Settings mehr wünscht und auch der Verbindungstest wegfallen soll. Der Kunde möchte die Applikation so einfach wie möglich halten und direkt zur Ersteichung gelangen können. Auch hier sei auf den Abschnitt 6.3 verwiesen sowie die Usability Protokolle im Appendix.

## 7.4 Testumgebung

Dieser Teil beschreibt die implementierte und verwendete Test-Umgebung, welche nicht Teil der Applikation oder der Requirements war, aber für uns von zentraler Wichtigkeit war, um den Upload Use Case, sowie Fehler und Probleme zu testen und zu überprüfen.

Im Folgenden werden der Server und die Simulation, ihr Aufsetzen und die Verwendung kurz beschrieben.

### 7.4.1 Server

Mit Hilfe von Docker wurde auf einem HSR Server ein Apache Server mit PHP erstellt. Dahinter läuft ein File Upload Server, welcher darauf wartet, PDF Dateien entgegenzunehmen, diese abzuspeichern und als Antwort einen öffentlichen Link zurückzuschicken. Wir haben zur Vereinfachung und für unsere Evaluation auf eine verschlüsselte Verbindung mittels SSL/TLS verzichtet, da es auch hier nur um eine Machbarkeitsstudie geht. In der produktiven Applikation sollte jedoch zwingend zum Schutz der Daten eine verschlüsselte Verbindung verwendet, sowie die finale URL zum effektiven Server angepasst werden. Dies sollte dringend vor dem Pilottest stattfinden.

### 7.4.2 Simulation

Zur Simulation der Waage wurde mittels Python 3 ein Script erstellt, welches die Waage mit all ihren zu erwartenden SICS Befehlen simuliert. Dabei wird der ACM360 Adapter mittels seriellem RS232 an den PC angeschlossen und via seriellem COM Port und dem Python Script angesteuert. Das Resultat ist eine Waagensimulation, welche genau wie die echte auf SICS Commands wartet und darauf antworten kann.

## 7.5 Anleitungen

Zur Benutzung der Applikation wird eine Benutzeranleitung mitgeliefert, welche im Appendix unter 10.4 betrachtet werden kann. Zur Weiterentwicklung der Applikation wird eine Installationsanleitung mitgeliefert, welche ebenfalls unter 10.5 betrachtet werden kann.

# 8 Evaluation

## 8.1 Kommunikation Adapter

### 8.1.1 Adapter

- Problem** Da man den Bluetooth Adapter ACM360 in verschiedene Modi versetzen kann, muss man sicherstellen, dass der Adapter als Slave definiert ist und sich im SSP-Modus befindet. Die Umschaltung von einem Modus in einen anderen gestaltete sich relativ schwierig. In der mitgelieferten Benutzeranleitung des Adapters ist die Umstellung in verschiedenen Sprachen beschrieben. Die Modus-Umstellung ist zwar beschrieben, doch man muss sehr genau auf die einzelnen Schritte achten, denn es kommt nicht nur auf den Tasten-Druck an, sondern auch auf den Zeitabstand dazwischen [34].
- Test** Um zu testen, ob der richtige Modus eingestellt ist, klickt man auf die einzige Taste, die am Adapter zu finden ist. Beim Klicken auf dem Adapter-Knopf muss die Lampe zwei Mal hintereinander Gelb aufleuchten.
- Lösung** Der Adapter wurde mehrmals zurückgesetzt. Um sicher zu gehen, dass dieser Adapter nicht per Zufall kaputt ist, haben wir einen anderen Adapter von einem METTLER TOLEDO Mitarbeiter erhalten. Dieser war schon im gewünschten Modus, doch auch den anderen konnten wir nach einigen Versuchen wie gewünscht konfigurieren.

### 8.1.2 Android Bluetooth Low Energy

- Problem** Eine der Kernanforderungen war es, sich mit dem Adapter zu verbinden und Waage Daten auszulesen. Dabei wurde uns sehr schnell bewusst, wie schwierig es ist, mit den Android eigenen API Befehlen, dem Bluetooth Manager sowie der sehr Low-Level nahen Implementation, die geforderten Anforderungen umzusetzen. Wir stiessen schnell an unsere Grenzen, wegen fehlendem Observer Handling sowie der Möglichkeit eines reaktiven Verhaltens, welches uns ermöglichen würde, mit States, Observer und Subjekten umzugehen. Oft wurde nach kurzer Zeit die Verbindung unerwarteterweise getrennt und es konnte für kurze Zeit keine Verbindung mehr hergestellt werden.

## Lösung

Mittels Nachforschungen im Internet sind wir auf die reaktive Bibliotheken rxJava und rxAndroid gestossen sowie das für unsere Anforderungen perfekte Bibliothek RxAndroidBle, welches die reaktiven Bibliotheken verwendet sowie eine einfache Schnittstelle zur Bluetooth Low Energy Kommunikation bietet. Die Beschreibung zu den Bibliotheken befinden sich im Abschnitt 5.3. Diese sind zudem unter der Apache Lizenz 2.0, im Abschnitt 10.6 beschrieben und frei verfügbar.

## 8.2 Kommunikation Waage

### 8.2.1 Waage

#### Problem

So wie der Adapter besitzt auch die Waage verschiedene Modi. Bei der Waage ist es jedoch so, dass nur ein bestimmter Modus, der Dialog-Modus, in den meisten Fällen verwendet wird. Daher haben wir zu Beginn keine Einstellungen der Waage verändert. Unsere Befehle (z.B. 'S', um das Gewicht zu ermitteln) konnten nicht via Bluetooth-Adapter an die Waage weitergeleitet und verarbeitet werden. Wir wussten nicht, ob es an der Waagen-Simulation liegt oder vielleicht am Terminal. Somit mussten wir klarstellen, ob es eventuell am Terminal selber liegt.

#### Test

Nachdem wir einen anderen Adapter verwendet hatten und bei diesem die Ergebnisse genau gleich ausgewertet wurden, entschlossen wir uns, die Einstellungen der Waage genauer unter die Lupe zu nehmen. Wir stellten den Modus von Dialog zu Demand um. Mit dieser Konfiguration konnten wir unsere Basis-Befehle (Gewicht 'S', Trieren 'TA', Nullieren 'Z', Tara löschen 'TAC') an die Waage übermitteln. Von allen weiteren Befehlen wie z.B. 'I6' zum Ermitteln der maximalen Kapazität, erhielten wir keine Antwort. In der Benutzeranleitung der Waage ist jedoch nur der Dialog-Modus beschrieben. Trotz langen Diskussionen mit den Service-Technikern und den Mitarbeitern aus der Werkstatt wusste niemand so recht, was dieser Modus ist und weshalb einzelne Befehle so gesendet werden können.

Da wir zusätzlich verunsichert waren, ob das Terminal ein Problem darstellt oder gewisse Befehle auf Grund des Simulators nicht richtig funktionieren, schlossen wir die Waage über ein RS232 [5] Kabel direkt an den Computer an. Über ein Terminal Tool (YAT), womit man serielle Verbindungen testen kann, sendeten wir einzelne SICS-Befehle an die Waage. Die Antworten waren genau die, die wir erwartet hatten. Also lag das Problem definitiv nicht am Terminal oder an der Simulation.

**Lösung** Wir sind alle Benutzeranleitungen und Dokumentationen durchgegangen, die eventuell weiterhelfen. Zum Schluss kamen jemand darauf, dass man vielleicht ein Carriage-Return und Linefeed dem Befehl anhängen

## 8.2.2 Simulation

**Problem** Während des Projekts sind wir auf ein grosses Hindernis gestossen, welches uns die Arbeit erschwert hat. Mitten im Projekt ist die Kommunikation via Bluetooth Low Energy zum Adapter ausgefallen. Dadurch wurden wir gezwungen, mittels Ausschlussverfahren herauszufinden, wo das Problem liegt. Dabei gab es mehrere Möglichkeiten. Einerseits konnte es am Adapter selber liegen oder andererseits zwischen Adapter und Waage oder als dritte Möglichkeit gar an der Waage selber. Die Weiterarbeit wurde dadurch blockiert und wir mussten uns was ausdenken.

Ausserdem hatten wir nur eine Waage, dafür mehrere Adapter.

**Test** Um die Möglichkeiten der Reihe nach zu verifizieren und wo möglich auszuschliessen, haben wir neben dem Test mit der Waage am PC, auch eine Simulation in Python erstellt, bei welchem der Adapter an den PC seriell angeschlossen werden konnte, um Signale zu empfangen und sichtbar zu machen.

**Lösung** Mit diesem Setup konnte herausgefunden werden, dass der Adapter in einem falschen Modus ist und welchen es zu setzen galt, wie schon weiter oben beschrieben. Ausserdem konnte damit gezeigt werden, dass die Kommunikation zwischen Adapter und Waage nicht das Problem war, sondern die Waage selber ebenfalls in einem falschen Modus war. Somit konnten wir dank der Simulation erkennen, dass Befehle via Bluetooth LE durch den Adapter geleitet werden und am anderen Ende herauskommen.

Die Simulation bot sich ausserdem als Lösung an, parallel an der Applikation zu arbeiten, ohne mehrere Waagen zu halten.

Die weiterarbeit wurde dadurch auch nicht mehr blockiert.

## 8.3 Eichprozess

**Problem** Die Eichprozessschritte bilden den Kern unserer Applikation. Da es wichtig erschien, eine gute Prozessdurchführung zu liefern, welche einfach und intuitiv sowie den Google Material Standards entspricht, kam uns schnell der Gedanke, eine geeignete Bibliothek dafür zu suchen.

**Lösung** Wir stiessen bei der Suche auf die *Android Material Stepper*[10] Bibliothek, welche im Abschnitt 5.3 beschrieben wird. Sie bringt die nötigen Anforderungen zur Umsetzung mit, welche wir mittels Testimplementation im Prototypen zu Beginn verifizieren konnten. Er ist sehr dynamisch und gut erweiterbar und anpassbar. Sie ist zudem unter der Apache Lizenz 2.0, die im Abschnitt 10.6 beschrieben wird, frei verfügbar.

## 8.4 Graph Eichfehlergrenze

**Problem** Da eine weitere Anforderung das Aufzeigen der Eichfehlergrenze war, wir dies in Android nur mittels Canvas selber zeichnen konnten und dies ein erheblicher Mehraufwand zu bedeuten hätte, mussten wir uns auch hier auf die Suche nach etwas Effizienterem machen.

**Lösung** Die Lösung war auch hier eine OpenSource Bibliothek, die sich *GraphView*[13] nennt und schon im Abschnitt 5.3 beschrieben wird. Sie erfüllt durch seine Flexibilität und der dynamischen Anpassungsmöglichkeiten alle Anforderungen, um den Eichfehler in einem Graph zu plotten und darzustellen. Sie ist zudem unter der Apache Lizenz 2.0, die im Abschnitt 10.6 beschrieben wird, frei verfügbar.

## 8.5 Zertifikat

**Problem** Als wichtige Anforderung wird auch das Erstellen einer PDF-Datei (Konformitätszertifikat) vorausgesetzt. Alle Eichschritte sollen in dieser Datei zusammengefasst und gespeichert werden. Um ein solches Dokument effizient zu erstellen, haben wir im Internet nach einfachen Wegen gesucht.

**Lösung** Wir wurden im Internet auf die Library *iTextG* aufmerksam. Mit dieser Library konnten wir innerhalb kurzer Zeit ein PDF erstellen und speichern. Ein Nachteil davon ist jedoch, dass eine kommerzielle Lizenz benötigt wird. Da wir in unserer Studienarbeit keine App verkaufen

oder effektiv benutzen und wir höchstens einen Pilottest damit starten, ist die freie Nutzung noch im erlaubten Rahmen. Sollte diese App aber weiterentwickelt und von der METTLER TOLEDO produktiv eingesetzt werden, so muss diese Lizenz dringend miteinbezogen werden. Genaue Informationen zur Lizenz findet man im Abschnitt 10.6.

## 8.6 Fazit

Zu Beginn hatten wir den Eindruck, dass die Kommunikation zwischen dem ACM360 Bluetooth Adapter und der Waage schnell und einwandfrei funktionieren würde, weil wir auch schon sehr früh Erfolg damit hatten. Doch als die Probleme mit den verschiedenen Modi und Befehlsangaben auftauchten, hat uns das gehörig verunsichert. Deshalb zogen wir auch voreilige Bilanz und dachten, dass die Kommunikation mit diesem Adapter sich auch in Zukunft schwierig gestalten würde. Doch nach diesen Tests und den Lösungen, die im Abschnitt 8.1 genauer beschrieben sind, stellte sich heraus, dass der ACM360 Bluetooth Adapter für eine solche Anwendung, die wir nun geschrieben haben, völlig ausreichend ist.

Die Android Applikation hat auf jeden Fall Potential. Auf Grund der Feedbacks der METTLER TOLEDO Mitarbeiter schliessen wir daraus, dass sich unsere Evaluation bezüglich dem Adapter und auch der Implementierung der Bluetooth-Kommunikation zwischen der Waage und dem Adapter durchaus gelohnt hat. Man wird Schritt für Schritt durch den Ersteichungsprozess geführt und hat keinen Schreibaufwand, womit die Fehlerquelle massiv minimiert wird. Zudem gewinnt man durch die einfache Handhabung Zeit und aufgrund der Führung werden keine relevanten Schritte übersehen.

# 9 Verzeichnisse

## 9.1 Glossar und Abkürzungen

### **ACM360**

Der Bluetooth Low Energy Adapter von Mettler Toledo

### **ADB**

Android Debug Bridge (adb) [29]

### **Android SDK**

Android Software Development Kit

### **Android Studio**

Android Studio Entwicklungsumgebung [16]

### **APK**

Android Application Package [28]

### **ButterKnife**

ButterKnife Library [11]

### **Code-Freeze**

Endgültige Zeitpunkt in dem der Quellcode nicht mehr verändert wird [35]

### **Confluence**

Kolaborationssoftware für Teams [36]

### **Continuous Delivery**

Techniken und Werkzeuge zum Verbessern der Auslieferung von Software[37]

### **Continuous Integration**

Fortlaufendes Zusammenfügen von Komponenten in den Master Branch[38]

### **Durchstich**

Zeitpunkt in dem der Architekturprototyp funktioniert [39]

### **Extensible Markup Language XML**

Erweiterbare Auszeichnungssprache kurz XML zur Darstellung hierarchisch strukturierter Daten in Textform [15]

### **Functional Test**

Prüfung einer Funktionseinheit gegen deren funktionale Anforderungen [40]

### **Gitlab**

Webanwendung zur Versionsverwaltung für Softwareprojekte auf Basis von Git [27]

### **Graph View**

Graph View Plotting Library [13]

### **IEEE**

The world's largest technical professional organization for the advancement of technology[41]

### **IEEE Inside BLE**

Inside Bluetooth Low Energy [42]

### **IntelliJ IDEA**

Entwicklungsumgebung für KVM von JetBrains [17]

### **itextG**

ItextG PDF Library [12]

### **Jira**

Software zur Vorgangs- und Projektverfolgung für agile Softwareentwicklungsprojekte [43]

### **Kotlin**

Programmiersprache Kotlin [33]

### **LaTeX**

LaTeX Typesetting System [30]

### **Lint**

Werkzeug zur statischen Codeanalyse [26]

### **Material Stepper**

Android Material Stepper Library [10]

## **MiKTeX**

Aktuelle Implementation von TeX und LaTeX [32]

## **Pilot Test**

Möglichkeit zur Validation des Produkts nach Abgabe [44]

## **Play Services**

Play Services Library [14]

## **Rclone**

Rclone ist ein Kommandozeilenprogramm zur Synchronisation von Dateien von und zu Cloud-Lösungen [45]

## **RS232**

Serielle Schnittstelle zum physischen Austausch von Daten [5]

## **RxAndroid**

RxAndroid Library [8]

## **RxAndroidBle**

RxAndroidBle Library [7]

## **RxJava**

RxJava Library [9]

## **SICS**

Mettler Toledo Standard Interface Command Set

## **TexStudio**

Schreibprogramm, um LaTeX Dokumente zu erzeugen und zu bearbeiten [31]

## **UAT**

User Acceptance Test [46]

## **Unit Test**

Modultest zum Test funktionaler Einzelteile [47]

## **Usability Test**

Test zur Gebrauchstauglichkeit

## **Wireframes**

Schematische Darstellung eines frühen konzeptionellen Entwurfs

## 9.2 Abbildungsverzeichnis

|       |  |    |
|-------|--|----|
| 4.1   | Domain Model                               | 16 |
| 4.2   | Use Case Diagramm                          | 21 |
| 5.1   | Deployment Diagramm                        | 22 |
| 5.2   | Package Diagramm                           | 23 |
| 5.3   | Übersicht Metriken                         | 25 |
| 6.1   | erster Entwurf                             | 28 |
| 6.2   | zweiter Entwurf                            | 29 |
| 10.1  | Projekt hinzufügen                         | 69 |
| 10.2  | Git Projekt hinzufügen                     | 69 |
| 10.3  | Gradle Build starten                       | 70 |
| 10.4  | Erfolgreiches Build Resultat               | 70 |
| 10.5  | Java Dateien                               | 74 |
| 10.6  | XML Dateien                                | 79 |
| 10.7  | Lint Check Abschluss                       | 83 |
| 10.8  | Unit Test Check Abschluss                  | 83 |
| 10.9  | Functional Test Bluetooth Check Abschluss  | 84 |
| 10.10 | Functional Test Permission Check Abschluss | 84 |
| 10.11 | Gitlab Pipeline Check Abschluss            | 84 |

## 9.3 Tabellenverzeichnis

|      |                         |    |
|------|-------------------------|----|
| 10.1 | Testprotokoll Resultate | 49 |
|------|-------------------------|----|

## 9.4 Literatur

- [1] *Elektroniknet BLE 4.0*, [Online; aufgerufen am 14. Dezember 2018]. Adresse: <https://www.elektroniknet.de/elektronik/kommunikation/bluetooth-low-energy-in-smartphones-wie-funktioniert-das-103061.htm>.
- [2] S. R. Hussain, S. Mehnaz, S. Nirjon und E. Bertino, „Seamless and Secure Bluetooth LE Connection Migration“, in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, Ser. CODASPY '17, Scottsdale, Arizona, USA: ACM, 2017, S. 147–149, ISBN: 978-1-4503-4523-1. DOI: 10.1145/3029806.3029840. Adresse: <http://doi.acm.org/10.1145/3029806.3029840>.
- [3] G. Panwar und S. Misra, „Inside Bluetooth Low Energy (Gupta, N.) [Book Review]“, *IEEE Wireless Communications*, Jg. 24, Nr. 4, S. 2–3, Aug. 2017, ISSN: 1536-1284. DOI: 10.1109/MWC.2017.8014283.

- [4] T. Gaitatzis, A. Ward und L. Manning, *Bluetooth Low Energy: A Technical Primer Your Guide to the Magic Behind the Internet of Things*. CANADA: ISBN Canada, 2017, ISBN: 1775128083.
- [5] *RS232*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: <https://kompendium.infotip.de/rs-232-die-serielle-schnittstelle.html>.
- [6] M. T. GmBH, *ACM360 Bluetooth Adapter User Manual*, Im Langacher 44, CH-8606 Greifensee, Switzerland, 2018.
- [7] *RxAndroidBle Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/Polidea/RxAndroidBle>.
- [8] *RxAndroid Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/ReactiveX/RxAndroid>.
- [9] *RxJava Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/ReactiveX/RxJava>.
- [10] *Android Material Stepper Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/stepstone-tech/android-material-stepper>.
- [11] *ButterKnife Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/JakeWharton/butterknife>.
- [12] *ItexG PDF Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/itext/itextpdf>.
- [13] *Graph View Plotting Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://github.com/jjoe64/GraphView>.
- [14] *Play Services Library*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://developers.google.com/android/guides/overview>.
- [15] *Extensible Markup Language XML*, [Online; aufgerufen am 16. Dezember 2018]. Adresse: [https://de.wikipedia.org/wiki/Extensible\\_Markup\\_Language](https://de.wikipedia.org/wiki/Extensible_Markup_Language).
- [16] *Android Studio Entwicklungsumgebung*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://developer.android.com/studio/>.
- [17] *Capable and Ergonomic IDE for JVM*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://www.jetbrains.com/idea/>.
- [18] *IntelliJ Plugin Statistic*, [Online; aufgerufen am 16. Dezember 2018]. Adresse: <https://plugins.jetbrains.com/plugin/4509-statistic>.
- [19] *IntelliJ Plugin CodeMR*, [Online; aufgerufen am 16. Dezember 2018]. Adresse: <https://plugins.jetbrains.com/plugin/10811-codemr>.
- [20] *Subscribe*, [Online; aufgerufen am 18. Dezember 2018]. Adresse: <http://reactivex.io/documentation/operators/subscribe.html>.
- [21] *Android SDK*, [Online; aufgerufen am 3. Dezember 2018]. Adresse: <https://developer.android.com/studio/releases/sdk-tools>.

- [22] *Android 7.1 SDK 25 Nougat*, [Online; aufgerufen am 3. Dezember 2018]. Adresse: <https://developer.android.com/about/versions/nougat/android-7.1>.
- [23] *Google Accessibility Scanner*, [Online; aufgerufen am 17. Dezember 2018]. Adresse: <https://support.google.com/accessibility/android/faq/6376582?hl=de>.
- [24] *Google Play Store*, [Online; aufgerufen am 17. Dezember 2018]. Adresse: <https://play.google.com/store?hl=de>.
- [25] *Google Java Style Guide*, [Online; aufgerufen am 14. Dezember 2018]. Adresse: <https://google.github.io/styleguide/javaguide.html>.
- [26] *Lint*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: [https://de.wikipedia.org/wiki/Lint\\_\(Programmierwerkzeug\)](https://de.wikipedia.org/wiki/Lint_(Programmierwerkzeug)).
- [27] *Gitlab Versionsverwaltung*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://gitlab.com/>.
- [28] *Android Application Package*, [Online; aufgerufen am 28. November 2018]. Adresse: [https://en.wikipedia.org/wiki/Android\\_application\\_package](https://en.wikipedia.org/wiki/Android_application_package).
- [29] *Android Debug Bridge (adb)*, [Online; aufgerufen am 28. November 2018]. Adresse: <https://developer.android.com/studio/command-line/adb>.
- [30] *LaTeX Typesetting System*, [Online; aufgerufen am 28. November 2018]. Adresse: <https://www.latex-project.org/>.
- [31] *TexStudio LaTeX Schreibprogramm*, [Online; aufgerufen am 28. November 2018]. Adresse: <https://www.texstudio.org/>.
- [32] *MiKTeX*, [Online; aufgerufen am 25. November 2018]. Adresse: <https://miktex.org>.
- [33] *Programming Language Kotlin*, [Online; aufgerufen am 27. November 2018]. Adresse: <https://kotlinlang.org/>.
- [34] —, *ACM360 Bluetooth Adapter User Manual Page 18*, Im Langacher 44, CH-8606 Greifensee, Switzerland, 2018.
- [35] *Code-Freeze*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: <https://de.wikipedia.org/wiki/Code-Freeze>.
- [36] *Confluence - Team Collaboration Software*, [Online; aufgerufen am 25. November 2018]. Adresse: <https://www.atlassian.com/software/confluence>.
- [37] *Continuous Delivery*, [Online; aufgerufen am 5. Dezember 2018]. Adresse: <https://www.atlassian.com/continuous-delivery/ci-vs-ci-vs-cd>.
- [38] *Continuous Integration*, [Online; aufgerufen am 5. Dezember 2018]. Adresse: <https://www.atlassian.com/continuous-delivery/ci-vs-ci-vs-cd>.
- [39] *Durchstich*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: [https://de.wikipedia.org/wiki/Prototyping\\_\(Softwareentwicklung\)](https://de.wikipedia.org/wiki/Prototyping_(Softwareentwicklung)).

- [40] *Funktionstest*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: <https://de.wikipedia.org/wiki/Funktionstest>.
- [41] *ieee*, [Online; aufgerufen am 15. Dezember 2018]. Adresse: <https://www.ieee.org/>.
- [42] *ieee-inside-ble*, [Online; aufgerufen am 15. Dezember 2018]. Adresse: <https://ieeexplore.ieee.org/document/8014283>.
- [43] *Jira*, [Online; aufgerufen am 25. November 2018]. Adresse: <https://www.atlassian.com/software/jira>.
- [44] *Pilot Testing*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: <https://www.nngroup.com/articles/pilot-testing/>.
- [45] *Rclone*, [Online; aufgerufen am 2. Dezember 2018]. Adresse: <https://rclone.org/>.
- [46] *UAT*, [Online; aufgerufen am 18. Dezember 2018]. Adresse: [https://de.wikipedia.org/wiki/Akzeptanztest\\_\(Softwaretechnik\)](https://de.wikipedia.org/wiki/Akzeptanztest_(Softwaretechnik)).
- [47] *Unit Test*, [Online; aufgerufen am 19. Dezember 2018]. Adresse: [https://en.wikipedia.org/wiki/Unit\\_testing](https://en.wikipedia.org/wiki/Unit_testing).

# 10 Appendix

## 10.1 Testprotokoll

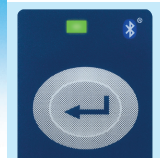
Am 18. Juni wurde die Zielerreichung getestet.

| Typ | Anforderung                                     | Test       | Resultat     |
|-----|---|------------|--------------|
| A   | Adapter ACM360 verwendet                        | siehe 5.1  | OK           |
| A   | Samsung Galaxy S7 mit Android 8.0 lauffähig     | siehe 6.1  | OK           |
| FA  | Wäge-Prüfung durchführen (Prozess durchführen)  | UAT        | OK           |
| FA  | Werte von Waage automatisch auslesen            | UAT        | OK           |
| FA  | Übertragung mittels Bluetooth LE (oder Classic) | UAT        | OK           |
| FA  | SICS Commands senden/empfangen                  | UAT        | OK           |
| FA  | Eichzertifikat erstellen                        | UAT        | OK           |
| FA  | Eichzertifikat hochladen                        | UAT        | OK           |
| FA  | Visuelle Darstellung Eichfehlergrenze           | UAT        | OK           |
| NFA | Stabile Bluetooth Verbindung                    | siehe 6.1  | OK           |
| NFA | Design MT Standard                              | siehe 6.1  | OK           |
| NFA | intuitiv und einfach bedienbar                  | siehe 6.1  | OK           |
| NFA | geführter Prüfungsablauf                        | siehe 6.1  | OK           |
| NFA | Material Designrichtlinien eingehalten          | siehe 6.4  | teilweise OK |
| NFA | Unit Tests laufen zu 100%                       | siehe 10.9 | OK           |
| NFA | Funktionale Tests (Bluetooth Service) zu 100%   | siehe 10.9 | OK           |
| NFA | Funktionale Tests (Permission Service) zu 100%  | siehe 10.9 | OK           |
| NFA | Keine Lint Fehler                               | siehe 10.9 | OK           |

Tabelle 10.1: Testprotokoll Resultate

## 10.2 ACM360 Datasheet

### ACM360 Bluetooth Adaptor



## Versatile Wireless Solution

### Fast and Flexible Peripheral Connectivity

#### Low Power Consumption

- These industrial Bluetooth adaptors (Version 4.1) feature very low power consumption and are powered directly via the terminal's RS232 interface. The user is independent of any additional power supply.
- Arrange your working space however you wish via a stable data transmission distance of up to 10 meters.

#### Easy Installation and Pairing

- The ACM360 Bluetooth adaptor allows for simple installation – just press the adaptor's multifunction key sequentially and you are ready to go.
- The additional available bracket allows mounting the adaptor at any convenient position. This makes the ACM360 a truly versatile wireless solution.

#### Flexible Label Printing

- The adaptor stores up to 10 different label templates. This is a perfect enhancement especially for our economic weighing terminals.
- The "multi to one" mode connects several terminals to one printer, reducing the amount of cabling and total number of printers needed.

#### Intelligent LED Interfacing

- The LED indicator changes colors depending on the status of the device.
- Get current status immediately by pressing the multi-function key.



#### ACM360 Bluetooth Adaptor Fast and Convenient Wireless Connection

The ACM360 is an industrial solution to connect peripherals such as printers or other Bluetooth 4.x Android smart devices to any METTLER TOLEDO terminal in a fast and convenient way. In addition to establishing terminal-to-terminal communication, the ACM360 is designed to provide an undisturbed wireless connection in rough environments.

Thanks to its low power consumption, the ACM360 can be directly powered over the terminal's RS232 interface, which makes an additional power supply obsolete. It also offers several multi-function features:

- Adaptor grants stable Bluetooth communication, but also handles your print templates
- Fast cable-free communication between METTLER TOLEDO terminals and a broad range of peripherals
- Suitable for harsh industrial environments (IP65)

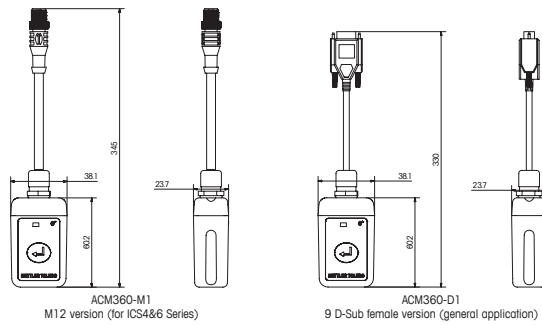
**METTLER TOLEDO**



**ACM360 Bluetooth Adaptor**

**Technical Data**

| Item                            | Parameter   |
|---------------------------------|---|
| Bluetooth Version               | Bluetooth version 4.1 (only compatible with Bluetooth version 4.x device) |
| Average power consumption       | 10mA; ~50mW   |
| Standard communication distance | ~10m (powered via RS232 signal)   |
| Baud rate                       | 4800, 9600, 19200, 38400, 57600, 115200bps                                |
| Work mode                       | SSP or Label print  |
| Housing                         | ABS   |
| Ingress Protection              | IP54 (DB9 version)  |
|                                 | IP65 (M12 version)  |
| Inner antenna                   | Output power  |
|                                 | Sensitivity   |
| Power supply                    | RS232 signal or 5 ~ 9 VDC from RS232 Pin 9                                |
| Key                             | One multi-function key  |
| Indicator                       | Three color LED indicator (red, yellow, green)                            |
| Accessory                       | Bracket (ABS)   |



[www.mt.com](http://www.mt.com)   
 For more information



**Mettler-Toledo (Changzhou)**  
 111 West Taihu Road Changzhou  
 Jiangsu 213125 China  
 Tel. +86-519-8664-2040  
 Fax +86-519-8664-1991

Subject to technical changes.  
 © 2017 Mettler-Toledo (Changzhou)  
 Printed in P. R. China 2017/03

## 10.3 SICS Commands

Level 0 commands

### 4.2 Command description

#### 4.2.1 IO – Inquiry of all implemented MT-SICS commands

|                 |   |   |
|-----------------|---|---|
| Command         | IO  | Send list of all implemented MT-SICS commands   |
| Response        | IO_B_X_"1. command"<br>IO_B_X_"2. command"<br>...<br>IO_B_X_"last command"<br>IO_A<br>IO_I  | x = number of the MT-SICS level the command belongs to<br><br>Command executed successfully<br>Command understood but not executable at present |
| <b>Example</b>  |   |   |
| Command         | IO  | Send list of all implemented MT-SICS commands   |
| Response        | IO_B_O_"IO"<br>...<br>IO_B_1_"D"<br>...<br>IO_B_3_"CLR"<br>...<br>IO_A  | Level 0 command IO implemented<br><br>Level 1 command D implemented<br><br>Level 3 command CLR implemented<br><br>Command executed successfully |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• The IO command lists all commands implemented in the present software.</li> <li>• All level 0 commands are listed in alphabetical order before all commands of level 1 etc. This order corresponds to the listing order in this manual.</li> <li>• Level 2 commands are specific for a complete product family.</li> <li>• Level 3 commands are an enhancement of level 2 but only specific for certain applications of the product family.</li> </ul> |   |



Level 0 commands

#### 4.2.4 I3 – Inquiry of terminal software version and type definition number

|                 |   |   |
|-----------------|---|---|
| Command         | I3  | Inquiry of terminal software version and type definition number   |
| Response        | I3_A_"aa-bb-cc.cc.cc-dd-e"  | aa Product family<br>S4 ICS4__ series<br>S6 ICS6__ series<br>bb Weighing interface<br>DC Digicell<br>ID IDNet<br>??? SicsPro<br>cc.cc.cc Software version<br>dd Application abbreviation<br>BW Basic weighing<br>CT Counting<br>MF Multi functional<br>OU Over/Under Checkweighing<br>SW Straight weighing<br>e Application level<br>2 ICS_2_<br>3 ICS_3_<br>4 ICS_4_<br>6 ICS_6_<br>8 ICS685<br>I3_I Command understood but not executable |
| <b>Example</b>  |   |   |
| Command         | I3  | Inquiry of terminal software version and type definition number   |
| Response        | I3_A_"S4-DC-01.01.00-OU-4"<br>I3_A_"S6-DC-01.01.00-MF-4"  | ICS449<br>ICS685  |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• For more information on the software version refer to the corresponding Service Manual.</li> </ul> |   |

#### 4.2.5 I4 – Inquiry of serial number

|                 |   |   |
|-----------------|---|---|
| Command         | I4  | Inquiry of serial number  |
| Response        | I4_A_"text"<br>I4_I   | Serial number as text<br>Command understood but not executable at present |
| <b>Example</b>  |   |   |
| Command         | I4  | Inquiry of serial number  |
| Response        | I4_A_"1234567"  | Serial number of the scale is 1234567                                     |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• The serial number agrees with that on the model plate and is different for every scale.</li> <li>• The serial number can be used e.g., as a device address in a network solution.</li> <li>• The response to I4 appears unsolicited after switching on and after the reset command @.</li> </ul> |   |

### 4.2.6 S – Send stable weight value

|                 |   |   |
|-----------------|---|---|
| Command         | S   | Send the current stable net value   |
| Response        | S_S_Weight value_Unit<br>S_I<br>S_+<br>S_-  | Current stable weight in the unit currently set for Unit 1<br>Command understood but not executable at present<br>Scale in overload range<br>Scale in underload range |
| <b>Example</b>  |   |   |
| Command         | S   | Send a stable weight value  |
| Response        | S_S_100.00_g  | The current stable weight value is 100.00 g   |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• The duration of the timeout depends on the scale type.</li> <li>• To send the stable weight value in the currently displayed unit, see SU command in level 2.</li> </ul> |   |

### 4.2.7 SI – Send weight value immediately

|                 |   |   |
|-----------------|---|---|
| Command         | SI  | Send the current weight value irrespective of stability   |
| Response        | S_S_Weight value_Unit<br>S_D_Weight value_Unit<br>S_I<br>S_+<br>S_-   | Current stable weight in the unit currently set for Unit 1<br>Dynamic (unstable) weight in the unit currently set for Unit 1<br>Command understood but not executable at present<br>Scale in overload range<br>Scale in underload range |
| <b>Example</b>  |   |   |
| Command         | SI  | Send a stable weight value  |
| Response        | S_D_129.07_g  | The current dynamic weight value is 129.07 g  |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• The response to the command SI is the last internal weight value (stable or dynamic) before receipt of the command SI.</li> <li>• To send the weight value immediately in the currently displayed unit, see SIU command in level 2.</li> </ul> |   |

Level 0 commands

#### 4.2.8 SIR – Send weight value immediately and repeat

|                |   |   |
|----------------|---|---|
| Command        | SIR   | Send the weight value repeatedly irrespective of stability  |
| Response       | S_S_Weight value_Unit<br>S_D_Weight value_Unit<br>S_I<br>S_+<br>S_-   | Current stable weight in the unit currently set for Unit 1<br>Dynamic (unstable) weight in the unit currently set for Unit 1<br>Command understood but not executable at present<br>Scale in overload range<br>Scale in underload range |
| <b>Example</b> |   |   |
| Command        | SIR   | Send current weight values  |
| Response       | S_D_____129.07_g<br>S_D_____129.03_g<br>S_S_____129.11_g<br>S_S_____129.51_g<br>S_D_____128.05_g<br>...   | The scale sends stable or unstable weight values continuously.  |
| Comments       | <ul style="list-style-type: none"> <li>• SIR is overwritten by all send commands and hardware break and hence cancelled.</li> <li>• To send weight values immediately in the currently displayed unit, see SIRU command in level 2.</li> <li>• The number of weight values per second depends on the scale type.</li> </ul> |   |

#### 4.2.9 Z – Zero

|                |  |  |
|----------------|--|--|
| Command        | Z  | Zero the scale when stable   |
| Response       | Z_A<br>Z_I<br>Z_+<br>Z_-   | Command executed successfully<br>Command understood but not executable<br>Upper limit of zero setting range exceeded<br>Lower limit of zero setting range exceeded |
| <b>Example</b> |  |  |
| Command        | Z  | Zero the scale   |
| Response       | Z_A  | Command executed successfully  |
| Comments       | <ul style="list-style-type: none"> <li>• The tare memory is cleared during zero setting.</li> <li>• The duration of the timeout depends on the scale type.</li> <li>• The zero point determined during switching on is not influenced by this command, i.e., the measurement ranges remain unchanged.</li> </ul> |  |

### 5.2.5 T – Taring

|                |  |   |
|----------------|--|---|
| Command        | T  | Tare a stable weight value  |
| Response       | T_S_Weight value_Unit<br><br>T_I<br>T_+<br>T_-   | Taring performed, tare weight value in the unit currently set under Unit 1<br><br>Command understood but not executable<br>Upper limit of taring range exceeded<br>Lower limit of taring range exceeded |
| <b>Example</b> |  |   |
| Command        | T  | Tare  |
| Response       | T_S_.....100.00_g  | The scale is tared and has a value of 100.00 g in the tare memory   |
| Comments       | <ul style="list-style-type: none"> <li>• The tare memory is overwritten by the new tare weight value.</li> <li>• The duration of the timeout depends on the scale type.</li> <li>• Clearing tare memory: see command TAC.</li> <li>• The weight value is always sent in Unit 1.</li> </ul> |   |

### 5.2.6 TA – Inquiry/presetting of tare weight value

|                |  |  |
|----------------|--|--|
| Command        | TA<br>TA_Weight value_Unit   | Inquiry of a tare weight value<br>Entry of a tare preset value in the unit currently set under Unit 1  |
| Response       | TA_A_Weight value_Unit<br><br>TA_I<br><br>TA_L   | Taring performed with the preset tare value in the unit currently set under Unit 1<br><br>Command understood but not executable (scale is currently executing another command, e.g., zero setting, or timeout as stability was not reached)<br><br>Command understood but not executable; wrong or missing parameter |
| <b>Example</b> |  |  |
| Command        | TA_100.00_g  | Tare 100.00 g  |
| Response       | TA_A_.....100.00_g   | The scale has 100.00 g in the tare memory  |
| Comments       | <ul style="list-style-type: none"> <li>• The tare memory will be overwritten by the preset tare weight value.</li> <li>• The keyed-in tare value will be rounded by the scale to the current readability. The response weight value is rounded in Unit 1.</li> <li>• The preset value may be entered in any supported unit.</li> <li>• The taring range is specified to the scale type.</li> </ul> |  |

### 5.2.7 TAC – Clear tare value

|          |                |  |
|----------|----------------|--|
| Command  | TAC            | Clear tare value   |
| Response | TAC_A<br>TAC_I | Tare value cleared, 0 is in the tare memory<br>Command understood but not executable |

Level 2 commands

**6.2.34 SX – Send stable weighing data**

|                |  |  |
|----------------|--|--|
| Command        | SX   | Send current stable weighing data with the currently displayed unit  |
| Response       | SX_S_x1_x2_x3_x4_x5_x6<br><br>SX_I<br>SX_+<br>SX_-   | Current stable gross, net and tare weights<br>x1 = gross (G.....)<br>x2 = unit of gross weight<br>x3 = net (N.....)<br>x4 = unit of net weight<br>x5 = tare (T.....)<br>x6 = unit of tare weight<br><br>Command understood but not executable<br>Scale in overload range<br>Scale in underload range |
| <b>Example</b> |  |  |
| Command        | SX   | Inquiry of current stable weighing data  |
| Response       | SX_S_G_2.00030_kg_N_1.99970_kg_T_0.00060_kg<br>Gross weight = 2.00030 kg, Net weight = 1.99970 kg, Tare weight = 0.00060 kg  |  |
| Comments       | <ul style="list-style-type: none"> <li>No response until weight is stable.</li> <li>The duration of the timeout depends on the scale type.</li> <li>See section 2.2.4 for response examples for S commands.</li> </ul> |  |

**6.2.35 SXI – Send weighing data immediately**

|                |   |   |
|----------------|---|---|
| Command        | SXI   | Send current weighing data immediately with the currently displayed unit  |
| Response       | SX_S_x1_x2_x3_x4_x5_x6<br>SX_D_x1_x2_x3_x4_x5_x6<br><br>SX_I<br>SX_+<br>SX_-  | Current stable gross, net and tare weights<br>Current unstable (dynamic) gross, net and tare weights<br>x1 ... x6: see command SX<br><br>Command understood but not executable<br>Scale in overload range<br>Scale in underload range |
| <b>Example</b> |   |   |
| Command        | SXI   | Send the current weighing data immediately  |
| Response       | SX_D_G_2.00030_kg_N_1.99970_kg_T_0.00060_kg<br>SX_S_G_2.00033_kg_N_1.99972_kg_T_0.00061_kg  | Dynamic weight values<br>Stable weight values   |
| Comments       | <ul style="list-style-type: none"> <li>Like SX command, but response at any time (stable or unstable result).</li> <li>See section 2.2.4 for response examples for S commands.</li> </ul> |   |

Level 3 commands

### 7.2.8 I6 – Inquiry of the scale build parameters

|                 |   |   |
|-----------------|---|---|
| <b>Command</b>  | I6  | Inquiry of the scale build parameters   |
| <b>Response</b> | I6_IB_P<br>I6_MAX_Max value_Unit<br>I6_MIN_Min value_Unit<br>I6_TH_pt value_Unit<br>I6_Ri_resolu_Unit/maxval_Unit<br><br>I6_IE_nd<br><br>IE | ICS scale family<br>Max value = maximum capacity<br>Min value = minimum capacity<br>pt value = maximum preset tare<br>Ri: Range information<br>Ri = 0           Single Range<br>Ri = 1, 2, 3    Partial Range, Interval info<br>resolu: resolution of Range/Interval Ri<br>maxval: max value of Range/Interval Ri:<br><br>nd: approved resolution<br>nd = 0 d        non approved scale<br>nd = 1 d        display value equals verification interval<br>nd = 10 d       class II scale with extended resolution<br><br>End of parameters |
| <b>Example</b>  |   |   |
| <b>Command</b>  | I6  | Inquiry of scale build parameters   |
| <b>Response</b> | I6_IB_P<br>I6_MAX_15.000_kg<br>I6_MIN_0.002_kg<br>I6_TH_15.000_kg<br>I6_R1_0.002_kg/6.000_kg<br>I6_R2_0.005_kg/15.000_kg<br>I6_E_0d<br>IE   | ICS scale family<br>Max. capacity   15.000 kg<br>Min. load        0.002 kg<br>Max. preset tare 15.000 kg<br>Range 1         resolution 0.002 kg / Max. value 6.000 kg<br>Range 2         resolution 0.005 kg / Max. value 15.000 kg<br>Non approved scale<br>End of parameters  |

**7.2.9 I10 – Inquiry/setting of device identification**

| <b>Inquiry</b>  |   |  |
|-----------------|---|--|
| Command         | I10   | Inquiry of the device identification   |
| Response        | I10_A_`text`<br>I10_J   | <b>text</b> represents the device identification<br>Command understood but not executable  |
| <b>Example</b>  |   |  |
| Command         | I10   | Inquiry of the device identification   |
| Response        | I10_A_`My scale`  | Device identification is <b>My scale</b>   |
| <b>Setting</b>  |   |  |
| Command         | I10_`text`  | Set the <b>text</b> for device identification  |
| Response        | I10_A<br>I10_J<br>I10_L   | Device identification has been set<br>Command understood but not executable<br>Command understood but not executable; wrong or missing parameter |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• Up to 40 characters can be entered as <b>text</b>.</li> <li>• The set device identification is retained even after the reset command @.</li> <li>• The scale can display, transfer and print the device identification.</li> </ul> |  |

**7.2.10 I11 – Inquiry of the model designation**

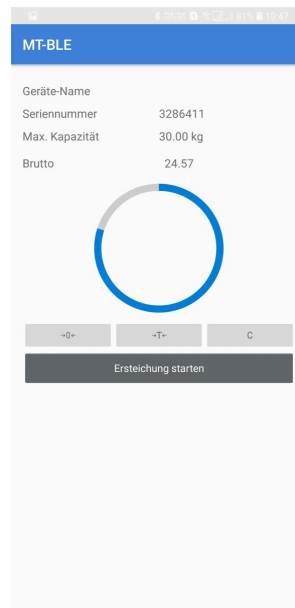
| Command         | I11   | Inquiry of model designation of the weighing terminal   |
|-----------------|---|---|
| Response        | I11_A_`text`<br>I11_J   | <b>text</b> represents the model designation<br>The model designation cannot be transferred at present as another operation is taking place |
| <b>Example</b>  |   |   |
| Command         | I11   | Inquiry of model designation  |
| Response        | I11_A_`ICS429a-BB30/c`  | The scale is an ICS429a-BB30/c  |
| <b>Comments</b> | <ul style="list-style-type: none"> <li>• The scale can display the device name.</li> <li>• The device name has a max. length of 40 characters.</li> <li>• The device name cannot be changed.</li> </ul> |   |

## 10.4 Benutzeranleitung

In diesem Benutzerhandbuch wird davon ausgegangen, dass ein geeignetes APK unserer Applikation vorhanden ist und der User weiss wie das APK installiert wird.

Customer Engineering

### Android App für Ersteichung Bedienungsanleitung



18071601, Internal Project  
Version 1.0.0



## Inhalt

|          |                             |          |
|----------|-----------------------------|----------|
| <b>1</b> | <b>Allgemein</b>            | <b>3</b> |
| 1.1      | Voraussetzungen             | 3        |
| <b>2</b> | <b>Ersteichungssoftware</b> | <b>4</b> |
| 2.1      | Login                       | 4        |
| 2.2      | Verbindungsaufbau           | 4        |
| 2.3      | Gewichtsfenster             | 5        |
| 2.4      | Ersteichung                 | 5        |
| 2.4.1    | Einstellungen der Waage     | 5        |
| 2.4.2    | Eckenlastprüfung            | 6        |
| 2.4.3    | Linearitätstest             | 6        |
| 2.4.4    | Wiederholbarkeitstest 1 & 2 | 7        |
| 2.4.5    | Eichfehlergrenze            | 7        |
| 2.4.6    | Zertifikat                  | 8        |
| 2.4.7    | Ersteichung beenden         | 8        |

## 1 Allgemein

Die Service-Techniker der Firma Mettler Toledo (Schweiz) GmbH verwenden bei einer Ersteichung eine Desktop-Applikation namens Miracal. Dieses Programm ist nicht für solche Arbeiten vorgesehen und deshalb wird eine Applikation benötigt, die alle notwendigen Schritte abbildet und mit der keine Fehler gemacht werden können. Da alle Service-Techniker ein Android Smartphone besitzen, welche Bluetooth integriert haben und die Mettler Toledo Bluetooth Adapter (ACM360) als Kabelersatz anbieten, soll eine Lösung entstehen, die beides kombinieren kann. Die neue Ersteichungs-Applikation soll mittels Bluetooth (LE) Verbindung mit dem Terminal bzw. mit der Waage kommunizieren und den Benutzer durch den Ersteichungsprozess führen.

Unsere Lösung ist einmalig, weil solche Adapter bisher nur als Kabelersatz zwischen Terminal / Waage und Drucker eingesetzt wurden. Mit dieser App kann man ganz einfach und schnell eine Ersteichung durchführen, jedoch nur für Single-Range Waagen der Klasse 2.

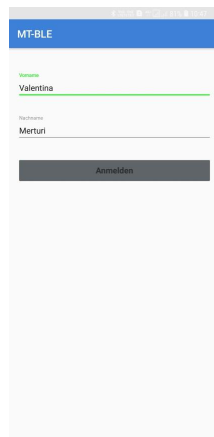
### 1.1 Voraussetzungen

Damit diese App überhaupt funktioniert, wird folgendes vorausgesetzt:

- Android-Smartphone (mind. Version 8.0)
- Bluetooth Adapter
- Single-Range Waage der Klasse 2.

## 2 Ersteichungssoftware

### 2.1 Login



Der Vor- sowie der Nachname müssen einmal beim Öffnen der Applikation eingegeben werden. Diese Daten werden lokal auf dem Smartphone abgelegt und beim nächsten Aufruf geholt.

Beim Betätigen der Anmelde-Taste werden diese Angaben wieder intern gespeichert.

Während des Gebrauchs der Applikation können die Namen nicht geändert werden. Diese Angaben werden für das Zertifikat am Schluss einer Ersteichung verwendet und müssen bei Bedarf beim Starten der Applikation angepasst werden.

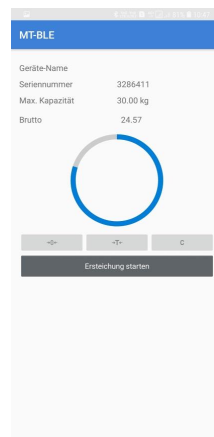
### 2.2 Verbindungsaufbau



Sobald man die Login-Daten eingegeben hat, gelangt man zu diesem Fenster. Hier werden alle verfügbaren Bluetooth Adapter (nur ACM360) aufgelistet. Es kann sein, dass es etwas länger dauert, bis ein Adapter angezeigt wird. Hierfür kann man auf die Schaltfläche "Aktualisieren" klicken, um den Vorgang etwas zu beschleunigen.

Wenn man ein verfügbares Gerät auswählt, wird direkt eine Verbindung aufgebaut. Dies dauert in der Regel ein paar Sekunden. Sollte kein Verbindungsaufbau möglich sein, so wird eine Fehlermeldung unten im Display angezeigt.

### 2.3 Gewichtsfenster



Beim Verbindungsaufbau werden direkt einige SICS-Befehle an die Waage gesendet und das Gewichtsfenster erscheint. In diesem Fenster werden die Seriennummer, die maximale Kapazität und das aktuelle Gewicht angezeigt, die über Bluetooth von der Waage übermittelte wurden.

Der Kreis in der Mitte stellt das maximale Gewicht dar. Der blaue Balken verändert sich an Hand des Gewichts, das auf der Waage liegt.

Mit den einzelnen Tasten unterhalb des Gewichtskreises kann man die Waage direkt ansteuern. Die erste Taste (>0<) führt eine Nullierung durch, die zweite (>T<) tariert die Waage und mit der dritten Taste (C) wird das aktuelle Tara gelöscht.

Belätigt man die unterste Taste "Ersteichung starten", so gelangt man in das nächste Fenster, in dem die Ersteichung beginnt.

### 2.4 Ersteichung

#### 2.4.1 Einstellungen der Waage



Der erste Schritt in der Ersteichung ist die Ansicht der Waagen-Einstellungen. Diese Angaben werden zwar alle im vorherigen Fenster von der Waage ausgelesen, doch hier erneut zusammengefasst.

### 2.4.2 Eckenlastprüfung



Bei der Eckenlastprüfung kann man wählen, ob man diese Prüfung durchführen möchte oder nicht.

Entscheidet man sich für die Eckenlastprüfung, so muss man die Gewichte der Reihe nach an den angegebenen Ecken platzieren. Das Gewicht wird berechnet und ist ein Drittel der maximalen Kapazität. Wenn man mit dieser Gewichtsangabe nicht zufrieden ist, kann man sie beliebig anpassen.

Sobald man ein Gewicht auf der angegebenen Ecke platziert hat, klickt man auf die Schaltfläche "Lesen", um den Gewichtswert von der Waage zu ermitteln. Dieser Wert wird direkt beim aktuellen Ecken eingetragen.

Wurden alle Ecken gewogen, so kann man mit der "Next"-Taste zum nächsten Schritt weiterfahren.

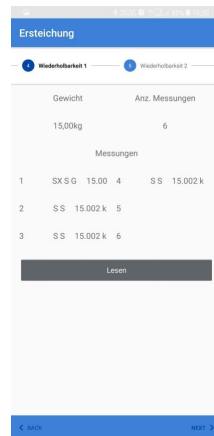
### 2.4.3 Linearitätstest



Beim Linearitätstest werden die Gewichte ebenfalls alle schon vorberechnet. Man wird hier aufgefordert der Reihe nach vom ersten bis zum letzten Gewicht die aktuellen Gewichtswerte von der Waage zu lesen. Die Werte werden auch hier automatisch in die Felder eingetragen.

Sobald man alle Werte hat, kann man weiterfahren zum nächsten Schritt.

### 2.4.4 Wiederholbarkeitstest 1 & 2



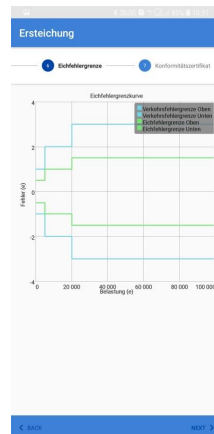
|           | Gewicht       | Anz. Messungen |
|-----------|---------------|----------------|
|           | 15,00kg       | 6              |
| Messungen |               |                |
| 1         | SXS G 15.00 4 | S S 15.002 k   |
| 2         | S S 15.002 k  | 5              |
| 3         | S S 15.002 k  | 6              |

Im Eichprozess gibt es zwei Wiederholbarkeitstests. Im ersten Test, wie im linken Bild angezeigt, arbeitet man mit der Hälfte der maximalen Kapazität. Dieser Wert wird berechnet. Im zweiten Wiederholbarkeitstest arbeitet man mit der maximalen Kapazität.

In beiden Fällen muss man die gleiche Anzahl Messungen durchführen und auslesen. Diese werden wie im vorherigen Schritt der Reihe nach eingetragen.

Um zum nächsten Schritt zu gelangen, klickt man auf die "Next"-Taste.

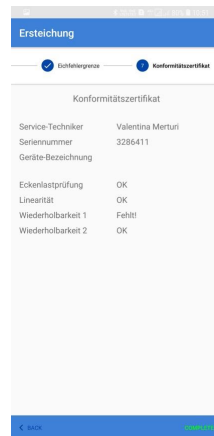
### 2.4.5 Eichfehlergrenze



Die Eichfehlergrenze wird an Hand des e-Werts berechnet und in diesem Graph dargestellt. Da wir in dieser Version nur Single-Range Waagen der Klasse 2 verwenden, sieht der Graph überall in etwa gleich aus.

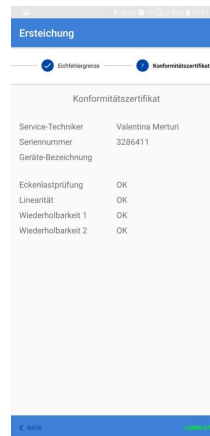
Mit "Next" gelangt man zum letzten Schritt.

### 2.4.6 Zertifikat

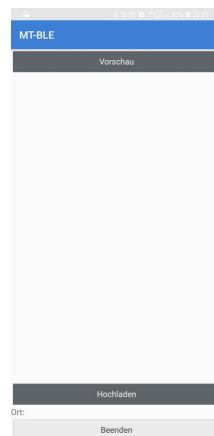


In diesem Schritt werden alle vorherigen Schritte zusammengefasst. Wenn bei einem Schritt nicht alle Messungen durchgeführt wurden, so erscheint in der Zusammenfassung ein "Fehl!" beim entsprechenden Bereich (links aufgeführt).

Auf der rechten Seite ist eine erfolgreiche Durchführung zusammengefasst. Alle Felder beinhalten ein "OK". Somit kann man die Ersteichung abschliessen mit "Complete".



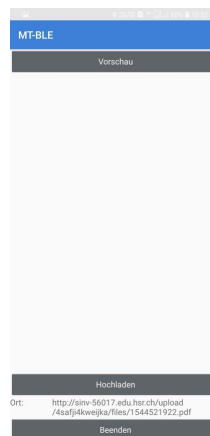
### 2.4.7 Ersteichung beenden



Hier befindet man sich nicht mehr im Ersteichungsprozess. In diesem Fall wurde das Zertifikat schon erstellt und man hat die Option, dieses Dokument anzuzeigen, in dem man auf die Taste "Vorschau" klickt.

Nun muss man noch das Zertifikat auf den File-Server hochladen, in dem man auf die Taste "Hochladen" klickt. Wenn dieser Vorgang erfolgreich war, erscheint der neue Pfad direkt unterhalb, wie im rechten Bild dargestellt.

Danach wird die "Beenden"-Taste wieder aktiviert und man kann die Ersteichung abschliessen.



Rev 001, 13.12.2018

## 10.5 Installationsanleitung

Zur Installation muss zwingend Android Studio 3.2 oder höher installiert werden. Ebenfalls vorhanden sein muss ein SDK ab Version 28 sowie die neuesten Build Tools ab Version 28. Sinnvoll für die funktionalen Tests ist ein Emulator, welcher einen Intel Prozessor voraussetzt.

Nachdem das Projekt vom Repository geklont oder sonst wie beschafft wurde, kann es in Android Studio mittels *Open an existing Android Studio project* geöffnet werden wie in der Grafik 10.1 zu sehen ist. Alternativ kann auch direkt vom Repository importiert werden wie die Grafik 10.2 zeigt.

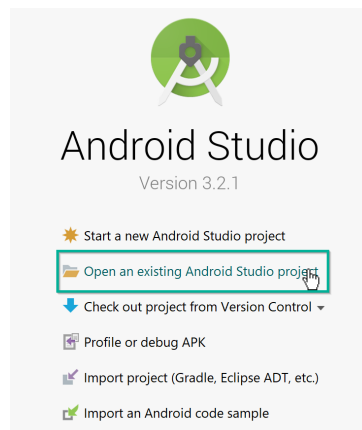


Abbildung 10.1: Projekt hinzufügen

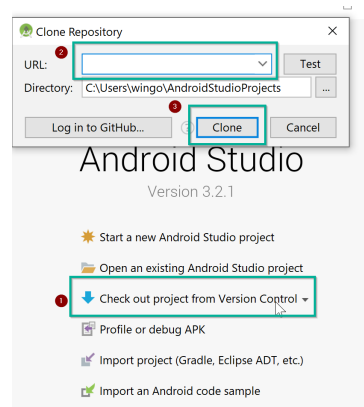


Abbildung 10.2: Git Projekt hinzufügen

Nach dem Hinzufügen sollte als erstes eine Projektsynchronisation durchgeführt werden wie sie im folgendem Bild 10.3 zu sehen ist. Dabei werden zunächst automatisch als erstes die Gradle Dependencies heruntergeladen sowie einige Prüfungen durchlaufen, was einige Zeit in Anspruch nehmen kann.

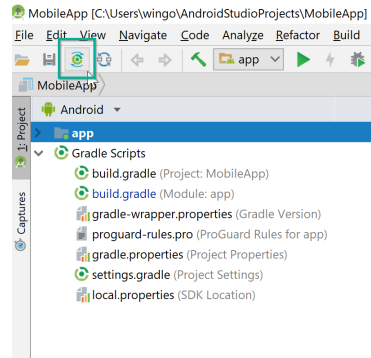


Abbildung 10.3: Gradle Build starten

Hat alles geklappt, sollte das Ergebnis wie in Bild 10.4 aussehen und die Entwicklung kann beginnen.

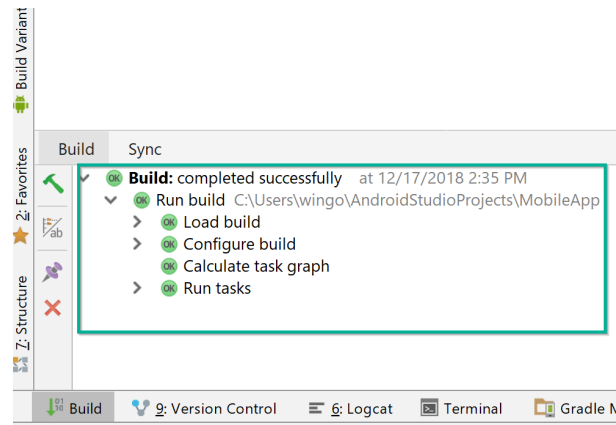


Abbildung 10.4: Erfolgreiches Build Resultat

## 10.6 Lizenzen

```
1 rxandroidble@1.7.0
2 Copyright 2016 Polidea Sp. z o.o
3
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8     http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License.
15
16 rxandroid@1.2.1
17 Copyright 2015 The RxAndroid authors
18
19 Licensed under the Apache License, Version 2.0 (the "License");
20 you may not use this file except in compliance with the License.
21 You may obtain a copy of the License at
22
23     http://www.apache.org/licenses/LICENSE-2.0
24
25 Unless required by applicable law or agreed to in writing, software
26 distributed under the License is distributed on an "AS IS" BASIS,
27 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
28 See the License for the specific language governing permissions and
29 limitations under the License.
30
31 rxjava@2.2.2
32 Copyright (c) 2016-present, RxJava Contributors.
33
34 Licensed under the Apache License, Version 2.0 (the "License");
35 you may not use this file except in compliance with the License.
36 You may obtain a copy of the License at
37
38 http://www.apache.org/licenses/LICENSE-2.0
39
40 Unless required by applicable law or agreed to in writing, software
41 distributed under the License is distributed on an "AS IS" BASIS,
42 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
43 See the License for the specific language governing permissions and
44 limitations under the License.
45
46 material-stepper@4.3.1
47 Copyright 2016 StepStone Services
48
49 Licensed under the Apache License, Version 2.0 (the "License"); you may not use
```

```
↪ this file except in compliance with the License. You may obtain a copy of the
↪ License at

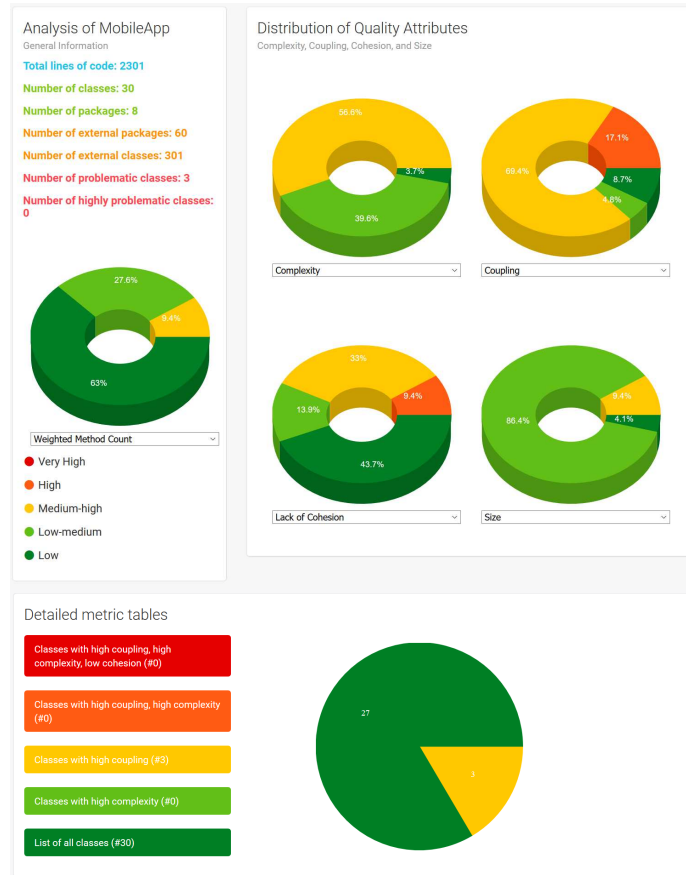
50
51     http://www.apache.org/licenses/LICENSE-2.0
52
53 Unless required by applicable law or agreed to in writing, software distributed
↪ under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
↪ CONDITIONS OF ANY KIND, either express or implied. See the License for the
↪ specific language governing permissions and limitations under the License.
54
55 butterknife@8.8.1
56 Copyright 2013 Jake Wharton
57
58 Licensed under the Apache License, Version 2.0 (the "License");
59 you may not use this file except in compliance with the License.
60 You may obtain a copy of the License at
61
62     http://www.apache.org/licenses/LICENSE-2.0
63
64 Unless required by applicable law or agreed to in writing, software
65 distributed under the License is distributed on an "AS IS" BASIS,
66 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
67 See the License for the specific language governing permissions and
68 limitations under the License.
69
70 itextg@5.5.10
71 iText is licensed as AGPL software.
72
73 AGPL is a free / open source software license.
74
75 This doesn't mean the software is gratis!
76
77 Buying a license is mandatory as soon as you develop commercial activities
↪ distributing the iText software inside your product or deploying it on a
↪ network without disclosing the source code of your own applications under the
↪ AGPL license. These activities include:
78
79     offering paid services to customers as an ASP
80     serving PDFs on the fly in the cloud or in a web application
81     shipping iText with a closed source product
82
83 graphview@4.2.2
84 Copyright 2016 Jonas Gehring
85
86 Licensed under the Apache License, Version 2.0 (the "License");
87 you may not use this file except in compliance with the License.
88 You may obtain a copy of the License at
89
90     http://www.apache.org/licenses/LICENSE-2.0
91
92 Unless required by applicable law or agreed to in writing, software
93 distributed under the License is distributed on an "AS IS" BASIS,
```

```
94 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
95 See the License for the specific language governing permissions and  
96 limitations under the License.  
97  
98 play-services@16.0.0  
99 Except as otherwise noted, the content of this page is licensed under the Creative  
    ↪ Commons Attribution 3.0 License, and code samples are licensed under the  
    ↪ Apache 2.0 License. For details, see our Site Policies. Java is a registered  
    ↪ trademark of Oracle and/or its affiliates.  
100  
101 Last updated August 23, 2018.
```

















































































## 10.7 Metriken

| Source File                       | Total Lines | Source Code Lines | Source Code Lines (%) | Comment Lines | Comment Lines (%) | Blank Lines | Blank Lines (%) |
|-----------------------------------|-------------|-------------------|-----------------------|---------------|-------------------|-------------|-----------------|
| BluetoothLEService.java           | 413         | 295               | 71%                   | 75            | 18%               | 43          | 10%             |
| BluetoothLEServiceTest.java       | 330         | 305               | 92%                   | 0             | 0%                | 25          | 8%              |
| ButterKnifeFragment.java          | 25          | 18                | 72%                   | 0             | 0%                | 7           | 28%             |
| CommandsTest.java                 | 48          | 36                | 75%                   | 0             | 0%                | 12          | 25%             |
| ConnectionActivity.java           | 265         | 224               | 85%                   | 0             | 0%                | 41          | 15%             |
| FakeBluetoothLEService.java       | 308         | 280               | 91%                   | 5             | 2%                | 23          | 7%              |
| GenericFileProvider.java          | 6           | 4                 | 67%                   | 0             | 0%                | 2           | 33%             |
| IBluetoothService.java            | 40          | 21                | 52%                   | 0             | 0%                | 19          | 48%             |
| Keys.java                         | 32          | 24                | 75%                   | 0             | 0%                | 8           | 25%             |
| LoginActivity.java                | 166         | 147               | 89%                   | 0             | 0%                | 19          | 11%             |
| PDFActivity.java                  | 181         | 157               | 87%                   | 0             | 0%                | 24          | 13%             |
| PermissionService.java            | 223         | 151               | 68%                   | 50            | 22%               | 22          | 10%             |
| PermissionServiceTest.java        | 72          | 48                | 67%                   | 12            | 17%               | 12          | 17%             |
| ScanActivity.java                 | 171         | 149               | 87%                   | 3             | 2%                | 19          | 11%             |
| ScanResultsAdapter.java           | 112         | 90                | 80%                   | 0             | 0%                | 22          | 20%             |
| SICSCommand.java                  | 176         | 45                | 26%                   | 129           | 73%               | 2           | 1%              |
| StepActivity.java                 | 136         | 109               | 80%                   | 6             | 4%                | 21          | 15%             |
| StepFragmentAdapter.java          | 92          | 84                | 91%                   | 0             | 0%                | 8           | 9%              |
| StepFragmentCertificate.java      | 110         | 93                | 85%                   | 0             | 0%                | 17          | 15%             |
| StepFragmentEdgeLoadStandard.java | 216         | 180               | 83%                   | 0             | 0%                | 36          | 17%             |
| StepFragmentGraphView.java        | 147         | 127               | 86%                   | 0             | 0%                | 20          | 14%             |
| StepFragmentLinearity.java        | 238         | 202               | 85%                   | 0             | 0%                | 36          | 15%             |
| StepFragmentRepeatabilityOne.java | 213         | 178               | 84%                   | 0             | 0%                | 35          | 16%             |
| StepFragmentRepeatabilityTwo.java | 203         | 170               | 84%                   | 0             | 0%                | 33          | 16%             |
| StepFragmentScaleSettings.java    | 92          | 77                | 84%                   | 0             | 0%                | 15          | 16%             |
| StepName.java                     | 12          | 7                 | 58%                   | 0             | 0%                | 5           | 42%             |
| UploadActivity.java               | 207         | 180               | 87%                   | 0             | 0%                | 27          | 13%             |
| Utils.java                        | 233         | 138               | 59%                   | 67            | 29%               | 28          | 12%             |
| UtilsTest.java                    | 80          | 62                | 78%                   | 0             | 0%                | 18          | 22%             |
| <b>Total:</b>                     | <b>4547</b> | <b>3601</b>       | <b>79%</b>            | <b>347</b>    | <b>8%</b>         | <b>599</b>  | <b>13%</b>      |

Abbildung 10.5: Java Dateien



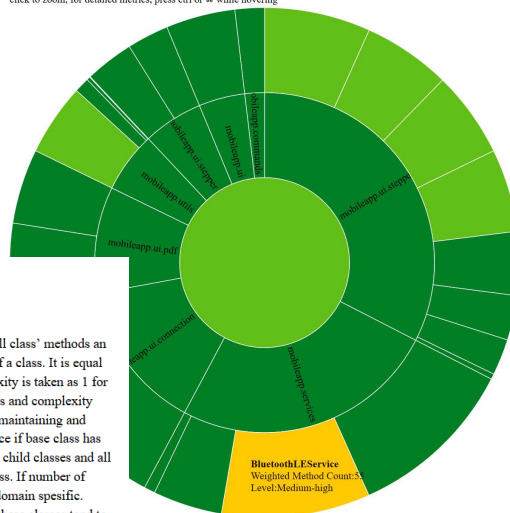
| Classes with high coupling, high complexity, low cohesion (#0) |                      |          |            |                  |      |     |             |             |                  |             |     |
|--|----------------------|----------|------------|------------------|------|-----|-------------|-------------|------------------|-------------|-----|
| ID   | CLASS                | COUPLING | COMPLEXITY | LACK OF COHESION | SIZE | LOC | WMC         | RFC         | CBO              | LCAM        |     |
| Classes with high coupling, high complexity (#0)               |                      |          |            |                  |      |     |             |             |                  |             |     |
| ID   | CLASS                | COUPLING | COMPLEXITY | LACK OF COHESION | SIZE | LOC | CBO         | WMC         | RFC              | NOM         |     |
| Classes with high coupling (#3)                                |                      |          |            |                  |      |     |             |             |                  |             |     |
| ID   | CLASS                | COUPLING | COMPLEXITY | LACK OF COHESION | SIZE | LOC | CBO         | CBO APP     | CBO LIB          | RFC         |     |
| 1  | BluetoothLEService   |          |            |                  |      | 217 | 22          | 0           | 22               | 81          |     |
| 2  | ScanActivity         |          |            |                  |      | 105 | 20          | 3           | 17               | 39          |     |
| 3  | StepActivity         |          |            |                  |      | 71  | 21          | 4           | 17               | 44          |     |
| Classes with high complexity (#0)                              |                      |          |            |                  |      |     |             |             |                  |             |     |
| ID   | CLASS                | COUPLING | COMPLEXITY | LACK OF COHESION | SIZE | LOC | WMC         | RFC         | NOM              | NOF         | DIT |
| List of all classes (#30)                                      |                      |          |            |                  |      |     |             |             |                  |             |     |
| ID   | CLASS                | COUPLING | COMPLEXITY | LACK OF COHESION | SIZE | LOC | COMPLEXITY  | COUPLING    | LACK OF COHESION | SIZE        |     |
| 1  | BluetoothLEService   |          |            |                  |      | 217 | medium-high | high        | high             | medium-high |     |
| 2  | ScanActivity         |          |            |                  |      | 105 | medium-high | high        | low              | low-medium  |     |
| 3  | StepActivity         |          |            |                  |      | 71  | medium-high | high        | medium-high      | low-medium  |     |
| 4  | FakeBluetoothLESe... |          |            |                  |      | 247 | medium-high | medium-high | low              | low-medium  |     |
| 5  | ConnectionActivity   |          |            |                  |      | 167 | medium-high | medium-high | low              | low-medium  |     |
| 6  | UploadActivity       |          |            |                  |      | 124 | medium-high | medium-high | low              | low-medium  |     |
| 7  | PDFActivity          |          |            |                  |      | 109 | medium-high | medium-high | low-medium       | low-medium  |     |

|    |                      |   |   |   |   |     |             |             |             |            |
|----|----------------------|---|---|---|---|-----|-------------|-------------|-------------|------------|
| 7  | PDFActivity          |    |    |    |    | 109 | medium-high | medium-high | low-medium  | low-medium |
| 8  | PermissionService    |    |    |    |    | 101 | medium-high | medium-high | low         | low-medium |
| 9  | LoginActivity        |    |    |    |    | 101 | medium-high | medium-high | low-medium  | low-medium |
| 10 | StepFragmentAdapter  |    |    |    |    | 61  | medium-high | medium-high | low         | low-medium |
| 11 | StepFragmentLinea... |    |    |    |    | 155 | low-medium  | medium-high | medium-high | low-medium |
| 12 | StepFragmentRepea... |    |    |    |    | 128 | low-medium  | medium-high | medium-high | low-medium |
| 13 | StepFragmentEdgeL... |    |    |    |    | 126 | low-medium  | medium-high | medium-high | low-medium |
| 14 | StepFragmentRepea... |    |    |    |    | 121 | low-medium  | medium-high | medium-high | low-medium |
| 15 | StepFragmentGraph... |    |    |    |    | 93  | low-medium  | medium-high | medium-high | low-medium |
| 16 | StepFragmentCerti... |  |  |  |  | 65  | low-medium  | medium-high | medium-high | low-medium |
| 17 | ScanResultsAdapter   |  |  |  |  | 57  | low-medium  | low-medium  | low-medium  | low-medium |
| 18 | StepFragmentScale... |  |  |  |  | 53  | low-medium  | low-medium  | low-medium  | low-medium |
| 19 | Utils                |  |  |  |  | 105 | low-medium  | low         | low         | low-medium |
| 20 | ButterKnifeFragment  |  |  |  |  | 8   | low-medium  | low         | low         | low        |
| 21 | GenericFileProvider  |  |  |  |  | 1   | low-medium  | low         | low         | low        |
| 22 | Upload               |  |  |  |  | 61  | low         | low         | low         | low        |
| 23 | SICSCCommand         |  |  |  |  | 43  | low         | low         | low         | low        |
| 24 | Keys                 |  |  |  |  | 22  | low         | low         | low         | low        |
| 25 | IBluetoothService    |  |  |  |  | 16  | low         | low         | low         | low        |
| 26 | ViewHolder           |  |  |  |  | 10  | low         | low         | low         | low        |

|    |                      |   |   |   |   |   |     |     |     |     |
|----|----------------------|---|---|---|---|---|-----|-----|-----|-----|
| 27 | StepName             | ■ | ■ | ■ | ■ | 5 | low | low | low | low |
| 28 | PermissionBinder     | ■ | ■ | ■ | ■ | 3 | low | low | low | low |
| 29 | BluetoothLeBinder    | ■ | ■ | ■ | ■ | 3 | low | low | low | low |
| 30 | OnAdapterItemClic... | ■ | ■ | ■ | ■ | 2 | low | low | low | low |

Class Metric:  | 
 Package Metric:  | 
 Project Metric:

click to zoom, for detailed metrics, press ctrl or # while hovering



### Weighted Method Count (WMC)

Scope: Class

**Description:** The weighted sum of all class' methods an represents the McCabe complexity of a class. It is equal to number of methods, if the complexity is taken as 1 for each method. The number of methods and complexity can be used to predict development, maintaining and testing effort estimation. In inheritance if base class has high number of method, it affects its' child classes and all methods are represented in sub-classes. If number of methods is high, that class possibly domain specific. Therefore they are less reusable. Also these classes tend to more change and defect prone.

**Related Quality Attributes:** Complexity, Size

[Click for more information](#)

Metric thresholds are defined as:

- Very high: >120
- High: 101 - 120
- Medium-high: 50 - 101
- Low-medium: 20 - 50
- Low: 0 - 20

| Source File                 | Total Lines | Source Code Lines | Source Code Lines [%] | Comment Lines | Comment Lines [%] | Blank Lines | Blank Lines [%] |
|-----------------------------|-------------|-------------------|-----------------------|---------------|-------------------|-------------|-----------------|
| activity_connection.xml     | 152         | 128               | 84%                   | 0             | 0%                | 24          | 16%             |
| activity_login.xml          | 74          | 66                | 89%                   | 0             | 0%                | 8           | 11%             |
| activity_pdf.xml            | 8           | 7                 | 88%                   | 0             | 0%                | 1           | 12%             |
| activity_scan.xml           | 27          | 24                | 89%                   | 0             | 0%                | 3           | 11%             |
| activity_step.xml           | 23          | 21                | 91%                   | 0             | 0%                | 2           | 9%              |
| activity_upload.xml         | 58          | 48                | 83%                   | 0             | 0%                | 10          | 17%             |
| AndroidManifest.xml         | 67          | 60                | 90%                   | 0             | 0%                | 7           | 10%             |
| circle_shape.xml            | 11          | 9                 | 82%                   | 0             | 0%                | 2           | 18%             |
| colors.xml                  | 6           | 6                 | 100%                  | 0             | 0%                | 0           | 0%              |
| dimens.xml                  | 6           | 6                 | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 101         | 101               | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 9           | 9                 | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 6           | 6                 | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 93          | 93                | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 9           | 9                 | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 5           | 5                 | 100%                  | 0             | 0%                | 0           | 0%              |
| group-index.xml             | 10          | 10                | 100%                  | 0             | 0%                | 0           | 0%              |
| ic_launcher_background.xml  | 4           | 4                 | 100%                  | 0             | 0%                | 0           | 0%              |
| ic_launcher_round.xml       | 5           | 5                 | 100%                  | 0             | 0%                | 0           | 0%              |
| ic_launcher.xml             | 5           | 5                 | 100%                  | 0             | 0%                | 0           | 0%              |
| item_scan_result.xml        | 42          | 36                | 86%                   | 0             | 0%                | 6           | 14%             |
| master-index.xml            | 117         | 117               | 100%                  | 0             | 0%                | 0           | 0%              |
| progress_bar.xml            | 18          | 17                | 94%                   | 0             | 0%                | 1           | 6%              |
| provider_paths.xml          | 4           | 4                 | 100%                  | 0             | 0%                | 0           | 0%              |
| sdk-registry.xml            | 143         | 142               | 99%                   | 0             | 0%                | 1           | 1%              |
| step_certificat.xml         | 174         | 149               | 86%                   | 0             | 0%                | 25          | 14%             |
| step_edge_load_standard.xml | 148         | 124               | 84%                   | 0             | 0%                | 24          | 16%             |
| step_graph_view.xml         | 17          | 15                | 88%                   | 0             | 0%                | 2           | 12%             |
| step_linearity.xml          | 322         | 271               | 84%                   | 0             | 0%                | 51          | 16%             |
| step_repeatability_one.xml  | 211         | 179               | 85%                   | 0             | 0%                | 32          | 15%             |
| step_repeatability_two.xml  | 210         | 179               | 85%                   | 0             | 0%                | 31          | 15%             |
| step_scale_settings.xml     | 118         | 100               | 85%                   | 0             | 0%                | 18          | 15%             |
| strings.xml                 | 69          | 68                | 99%                   | 0             | 0%                | 1           | 1%              |
| styles.xml                  | 33          | 25                | 76%                   | 0             | 0%                | 8           | 24%             |
| <b>Total:</b>               | <b>2305</b> | <b>2048</b>       | <b>89%</b>            | <b>0</b>      | <b>0%</b>         | <b>257</b>  | <b>11%</b>      |

Abbildung 10.6: XML Dateien

## 10.8 Detail Risiken

### 10.8.1 Wöchentliche Risiko Analyse

- Woche 1-3** In den ersten drei Wochen wurde die Erstellung der Risiken erarbeitet. Architektur und Kommunikation BLE scheinen weitaus das grösste wahrscheinliche Risiko zu sein und auch den grössten Impact zu haben [SKM-31 Kommunikation Android Bluetooth Low Energy] [SKM-39 Architektur].
- Woche 4-5** Mittlerweile funktioniert die Kommunikation des Prototyps einwandfrei. Daten können geschickt und geholt werden. Die meisten Risiken wurden daraufhin herabgestuft, besonders die Risiken bezüglich Kommunikation. Da die Architektur weitestgehend noch fehlt, bleibt das Risiko dort hoch. [SKM-31 Kommunikation Android Bluetooth Low Energy] [SKM-39 Architektur].
- Woche 6** Leider haben wir Probleme mit dem Adapter und es ist nicht mehr möglich, Daten zu empfangen oder zu senden. Auch mit älteren Code-Commits ist zurzeit keine lauffähige Demo möglich. Als Test wurden auch einige andere Bluetooth Low Energy Android Demo Apps getestet und diese funktionieren ebenfalls nicht. Wir evaluieren gerade mit Hochdruck, was das Problem sein könnte. Daher wurde BLE Kommunikation Impact und Probability wieder erhöht [SKM-31 Kommunikation Android Bluetooth Low Energy].
- Woche 7** Nachdem wir unzählige Versuche unternommen haben, haben wir bis jetzt leider immer noch keinen Erfolg mit der Kommunikation erzielt. Wir haben nach Handbuch alle möglichen Adapter Modi durchgespielt und keinen Erfolg in der Wiederherstellung der Kommunikation verzeichnen können. Jedoch konnten wir mittlerweile einen Adapter mit RS232 zum Anschluss an ein RS232 zu USB Adapter auftreiben, um diesen an den PC anzuschliessen und zu testen. Rudimentär wurde dieser mittels Putty im seriellen Modus getestet und wir bekommen die gesendeten String Commands in der Console des PCs als Output. Der FakeBluetoothLeService läuft mittlerweile auch einwandfrei und ist sauber implementiert. Damit ist es uns möglich mit den verschiedenen anderen Requirements unabhängig weiterzuarbeiten. Wir sind ab hier also nicht mehr an das Funktionieren der Waage direkt gebunden [SKM-31 Kommunikation Android Bluetooth Low Energy].
- Woche 8** Die gleichen Probleme bestehen weiterhin, wie in den vorigen Wochen bis zur Mitte der Woche. Danach haben wir die Waage noch einmal auf

Demand-Mode gesetzt und zu unserem Erstaunen konnten wir wieder kommunizieren. Ausserdem wichtig zu erwähnen ist, dass der Adapter nun im SSP Mode (bei einmaligem Click zweimaliges gelbes Blinken) ist [**SKM-31** Kommunikation Android Bluetooth Low Energy].

#### Woche 9

Die BLE Probleme scheinen gelöst zu sein und ein Bugfix für das runOnUiThread() wurde behoben, damit Notifications wieder empfangen werden können. Es wurde ein Adapter an den PC angeschlossen und ein kleines Python Script geschrieben, welches mit dem Adapter via serielle Kommunikation kommuniziert und quasi eine Waage simuliert. Dadurch kann nun auch ohne Waagen-Hardware, aber mit Adapter, getestet und gearbeitet werden [**SKM-31** Kommunikation Android Bluetooth Low Energy].

Die Risiken zur Kommunikation mit BLE und Kommunikation Dongle wurden getrennt, um die Probleme zu abstrahieren. Ausserdem können die Risiken nun wieder etwas gesenkt werden, da wir dem Problem auf der Spur sind [**SKM-31** Kommunikation Android Bluetooth Low Energy] [**SKM-104** Kommunikation zwischen Adapter und Waage].

#### Woche 10

Der PC Waagen Simulator wurde noch etwas verbessert, indem man Gewichte einstellen und verändern kann. Um dies zu ermöglichen mussten wir es thread-fähig machen. SICS Commands wie I2 oder I4 werden noch immer nicht von der Waage entgegengenommen oder eine Antwort zurückgesendet. Wir suchen fieberhaft nach der Ursache [**SKM-31** Kommunikation Android Bluetooth Low Energy] [**SKM-104** Kommunikation zwischen Adapter und Waage].

#### Woche 11

Das Problem mit dem Abrufen der Waagen Einstellungen mittels I2, I4 Befehlen konnte nun eruiert werden. Ausserdem wissen wir nun, wieso die Waage nicht im Dialog Modus sondern nur im Demand Modus antwortet. Die Konstellation der Adaptereinstellung und der Waageinstellung haben es uns sehr erschwert, herauszufinden, wieso die Kommunikation teilweise nicht mehr funktioniert hat. Jedoch ist nun klar, dass die Antwort im Prinzip sehr simple war. Im Demand Modus können Befehle ohne Lineending geschickt werden und die Befehle T, TAC, SI und P gehen dabei reibungslos. Im Dialog Modus war uns lange nicht klar, dass dort das Lineending zwingend mitgeschickt werden muss. Mit diesem Breaking Change in der Übermittlung durch anhängen und mitschicken dieses Lineendings ist es uns nun gelungen, alle Werte wunschgemäss abzufragen [**SKM-31** Kommunikation Android Bluetooth Low Energy] [**SKM-104** Kommunikation zwischen Adapter

und Waage].

Der vorläufige Fokus beim momentanen Abschluss der Applikation ist es, diese in einen konsistenten Zustand zu bringen, um mögliche Abstürze zu vermeiden [**SKM-44 Instability**].

Schliesslich ist uns ausserdem bewusst geworden, wie wichtig ein guter Zugang zu Dokumenten und FactSheets der verwendeten Hardware wäre und auch eine gute Kommunikation beim Kunden [**SKM-40 Mangel an Dokumenten zu benutzter Hardware**].

## Woche 12

Die Applikation ist in einem guten Zustand, aber lange noch nicht perfekt. Es gibt noch einige Bedingungen, welche aus Zeitmangel nicht mehr komplett überprüft oder abgefangen werden können. Die Applikation ist aber im Grossen und Ganzen in einem Stadium, in dem ein erster Pilot Test durchgeführt werden kann. Die grössten Risiken bleiben zur Zeit die Stabilität und das Exception-Handling durch die unzähligen Möglichkeiten an verschiedenen Antworten [**SKM-44 Instability**] [**SKM-36 Bedingungsprüfungen**].

Der Mangel an Dokumenten wird herabgestuft, da wir nicht mehr darauf angewiesen sind [**SKM-40 Mangel an Dokumenten zu benutzter Hardware**].

Die Aufwandschätzung wird erhöht, da uns die Zeit davonläuft und wir bald schon unsere 480h erreicht haben. Da wir einige Zeit in die Kommunikationsprobleme mit dem Adapter sowie zwischen Adapter und Waage stecken mussten, wurde der Aufwand unterschätzt [**SKM-43 Aufwandschätzung**].

## 10.9 Qualitätsmanagement Abschluss Screens

Hier befinden sich die Abschluss Screens nach dem Code Freeze.

### 10.9.1 Lint

In der Abbildung 10.7 ist ersichtlich, dass keine Fehler oder Warnungen mehr vorhanden sind.

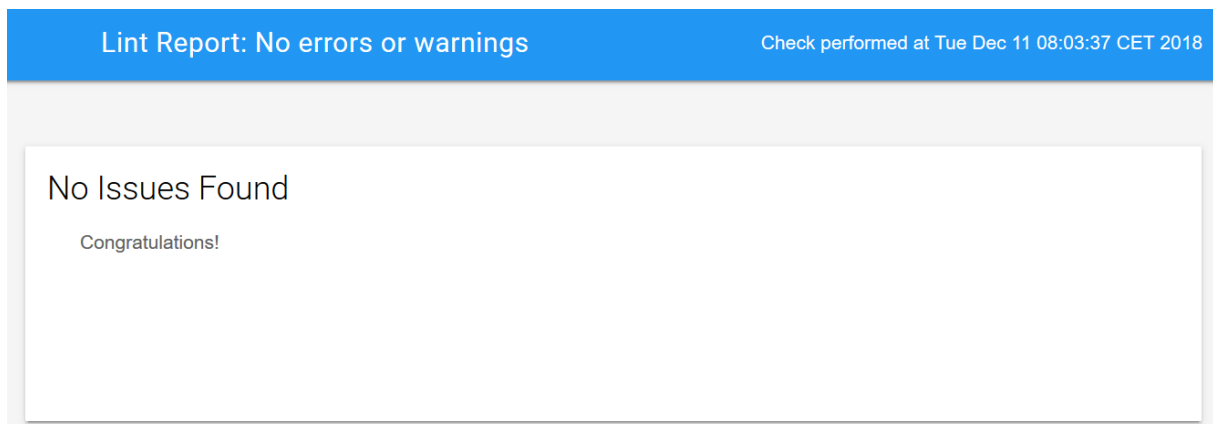


Abbildung 10.7: Lint Check Abschluss

### 10.9.2 Unit Tests

In der Abbildung 10.8 ist ersichtlich, dass alle Unit Tests zu 100% durchlaufen.

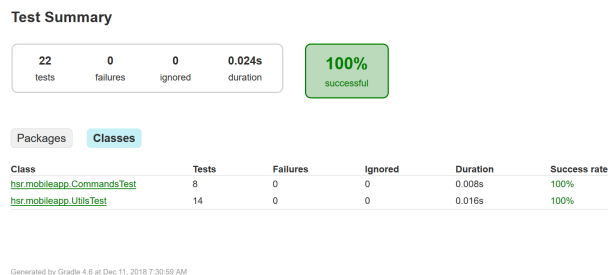


Abbildung 10.8: Unit Test Check Abschluss

### 10.9.3 Funktionale Tests

In der Abbildung 10.9 sowie 10.10 ist ersichtlich, dass alle funktionellen Tests zu 100% durchlaufen.

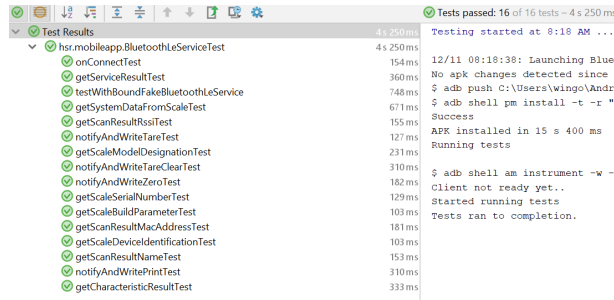


Abbildung 10.9: Functional Test Bluetooth Check Abschluss

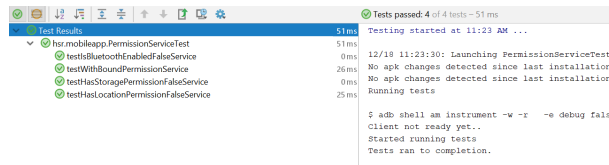


Abbildung 10.10: Functional Test Permission Check Abschluss

### 10.9.4 Gitlab CI/CD Pipeline

In der Abbildung 10.11 ist ersichtlich, dass beim Abschluss die Pipeline komplett und ohne Fehler durchläuft.

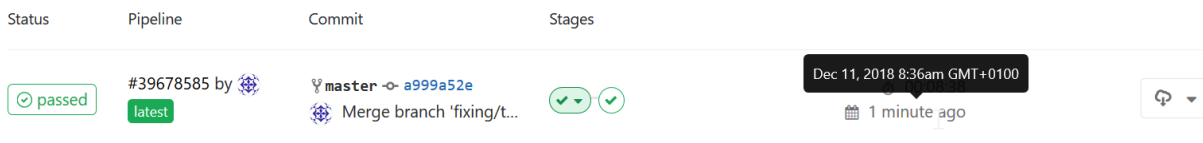


Abbildung 10.11: Gitlab Pipeline Check Abschluss