Testing

Rahmenbedingungen

Gefordert sind automatische und manuelle Tests. Die automatischen Tests aka Unittest sollen mithilfe der JUnit 5 library geschrieben werden und automatisch ausgeführt werden. Des Weiteren sollen, die automatischen Tests Entwickler unabhängig ausgeführt werden respektive Gitlab (Aussage aus Einführung Source Code IP2). Die manuellen Tests sollen mithilfe einer Vorlage geschrieben werden. Siehe Testing

Testkonzept

Das Testing verifiziert Komponenten, Teileinheiten oder das ganze Produkt. Dadurch gilt das Testing als Beweis dafür, dass das Produkt die geforderten Anforderungen erfüllt und mit der gewünschten Zuverlässigkeit funktionieren kann. Des Weiteren kann durch das Testing Fehler und Regressionen eines Produktes aufdeckten. Es ist ein wichtiger Bestandteil der Entwicklung. Unser Testingkonzept baut auf den Qualitätszielen und Anforderungen auf.

Aufgrund der genannten Aspekten haben sich folgende Punkte kristallisiert: Testing sollte nach Umsetzung eines Features stattfinden respektive, wenn garantiert ist das eine gewisse Beständigkeit seitens des Features vorhanden ist. Es macht ja keinen Sinn etwas zu testen, welches sich grundlegen ändert und so die doppelte Arbeit zu haben. Daher werden die Tests in der Endphase angepackt, weil vorher die Struktur noch sehr volatil ist. Durch die Rahmenbedingungen ist gefordert, dass automatische Tests und manuelle Test eingesetz werden müssen. Aufgrund der Gegebenheiten werden die automatischen Tests aka Unittest den Code abdecken und die manuellen Tests das ganze Spiel, die Hardware und die Benutzeroberfläche. Anmerkung die manuellen Tests sollen die Funktionsweise und nicht die Usability testen, denn dafür werden separate Usability Test durchgeführt.

Unittests

Unittest sollen kleine Komponenten testen (bsp. Methoden). Anhand des Features werden alle Komponenten des Features separat in eine Unittest überprüft. Diese Tests werden automatisch durch eine Pipeline ausgeführt (Eine Pipeline ist eine fest definiert Reihenfolgen von Operationen, welche nach eine Event bspw. einem Push ausgeführt wird, (Siehe mehr bei der Dev Dokument) oder können manuell von einem Entwickler lokal angestossen werden um frühzeitig Regressionen oder Fehler zu entdecken. Dafür sollen sie folgendermassen aufgebaut sein:

- Sprechender und sinnvollen Namen (der Name soll beschreiben, was der Testfall macht)
- Für Argumente sollen Mocks oder kontrollierte Daten übergeben werden
- Einheiten oder Methoden, welche getestet werden müssen
- Der Testfall soll nach dem "Given When Then Pattern" aufgebaut sein
- Eine Behauptung anwenden

- Sie müssen isoliert durchführbar sein
- Die Vor- und Nachbedingungen sollen aus dem Kommentar der zu testenden Methoden abgeleitet werden

Beispiel Java

```
public class Tests {
  @Test
  public void testisOverdrawnBalance500AssertTrue() throws Throwable {
    // Given
    Customer customer = mockCustomer();
    int initialBalance = 0;
    Account underTest = new Account(customer, initial_balance);

    // When
    int balance = -500;
    boolean result = underTest.isOverdrawn(balance);

    // Then
    assertTrue(result);
  }
}
```

Hilfe Unittests

Damit das Schreiben der Test einfacher ist und der Norm entspricht, gibt es im Entwickler Projekt ein Template dazu.

- Dateien → UnitTest (Erstellt eine Unittestdatei)
- Methode → uTest (Erstellt ein Testmethode)

Dev Unittest templates [Detaillierter Beschreib der Templates]

Der folgende Blogeintrag von Parasoft "How to write test case for software examples tutorial" diente als Basis für das Konzept.

Manuelle Tests

Manuelle Tests testen das ganze Spiel, spezifische Hardware des Produktes und die Benutzeroberfläche des Produktes. Sie sollen folgende Informationen umfassen:

- Testfall-ID
- Testbeschreibung
- Vorbedingungen
- Testdaten
- Schritte
- erwartetes Ergebnis/Nachbedingung

- Pass/Fail

Dafür sind in IntelliJ Templates vorhanden, welches als Vorlage verwendet werden können (Test→TestCase und Testrun→Testrun). Zuerst muss der Test als **TestCase** erstellt werden, danach kann ein **Testrun** erstellt werden, welche den Durchlauf des Tests darstellt und in einem Testmandat gruppiert wird. Diese Testmandate werden dann vom Tester als Packet durchgeführt.

Testplan

Der Testplan sieht vor, dass sobald eine gewisse Beständigkeit herrscht, das ist gegen Schluss des Projektes, die Test geschrieben und geprüft werden. Dabei wird zwischen Unit- und manuelle Tests unterschieden. Die Manuellen Tests sind die Abnahmetests.

Testpakete

Die Testfälle der Unit und Manuellen Tests/Abnahmetest wurden in Pakete eingeteilt. Diese Pakete dienen zur besseren Verwaltung.

Table 1. Testpakete

Paket	**Beschreibung**	**Autor**
Unittest: Audio	Audiofunktionalität testen	Andri Pieren
Unittest: Wechsel Statemachine	Wechsel der einzelnen Screens testen	Andri Pieren
Unittest: Team	Die Funktionalität für den Teammodus testen	Philip Gertsch
Unittest: Einzel	Die Funktionalität für den Einzelmodus testen	Philip Gertsch
Unittest: Quiz	Die Funktionalität für das Laden des Quizzes testen	Sabrina
Unittest: QR-Code	Die Funktionalität für das Erstellen des QR Code testen	Sabrina
Manuelle Test: Config	Die Applikation konfigurieren	Andri Pieren
Manuelle Test: Versus	Den Teammodus ganz durchspielen testen	Andri Pieren
Manuelle Test: Einzel	Den Einzelmodus ganz durchspielen	Andri Pieren

Testplan Unittest

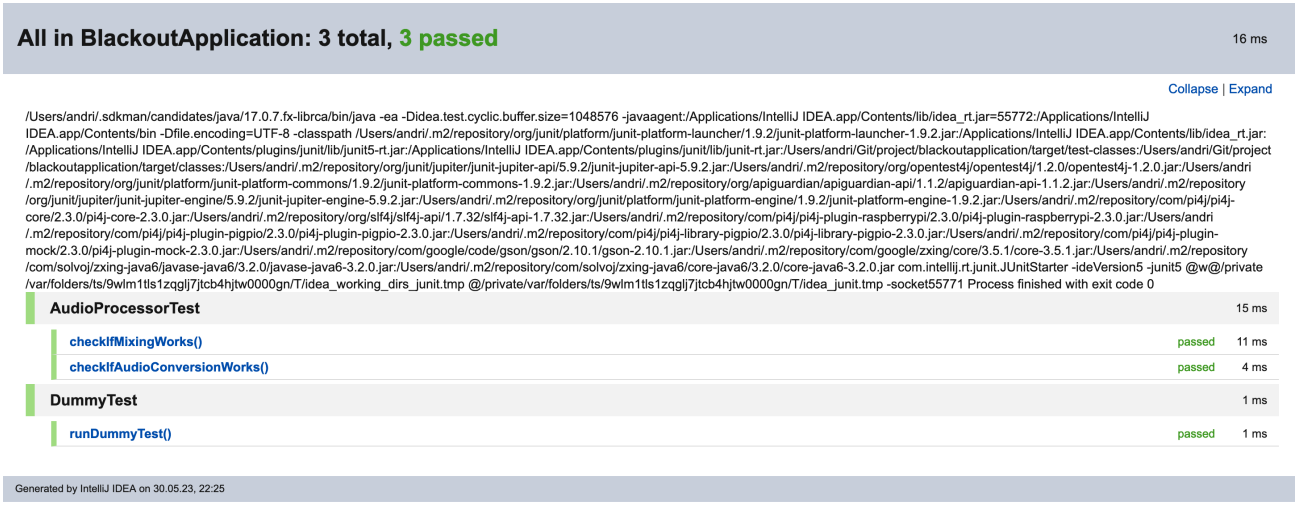
Der Testplan Unittest zeigt auf bis wann die einzelnen Pakete umgesetzt für die Unittests umgesetzt sind.

Table 2. Testplan Unittest

Datum	**Paket**
31.05.2023	Unittest: Audio → erreicht

Datum	Paket
31.05.2023	Unittest: Wechsel Statemachine → nicht erreicht
31.05.2023	Unittest: Team → nicht erreicht
31.05.2023	Unittest: Einzel → nicht erreicht
31.05.2023	Unittest: Quiz → nicht erreicht
31.05.2023	Unittest: QR-Code → nicht erreicht

Resultat der Unittests



Testplan Manuelle Test

Der Testplan Manuelle Test zeigt auf bis wann die einzelnen Pakete umgesetzt für die Manuellen Test umgesetzt sind.

Table 3. Testplan Unittest

Datum	Paket
30.05.2023	Manuelle Test: Config
30.05.2023	Manuelle Test: Versus
30.05.2023	Manuelle Test: Einzel

Testmandat der Abnahmetest/Manuellen Test

Die Testmandate sind die Durchläufe der Manuellen Tests, welche den Personen zugeordnet sind. Diese Personen führen in ihrem Testmandat den Testfall für das definierte Paket durch.

Testmandate: Paket «Manuelle Test: Config» (bis und mit 31.05.2023)

Testmandat

Table 4. Testmandat A (Andri)

Datum	Testrun	Result	Link
30.05.2023	Philip und Andri		Testrun

Testdurchlauf vom Test: System Check

Datum	31.05.2023
Verantwortlich	Andri Pieren
Tester	Andri Pieren

System Check durchführen

Beschreibung	
In diesem Testfall wird geprüft, ob die System-Checks erfolgreich durchgeführt werden können.	
Vorbedingung	
Blackout Applikation ist auf dem Raspberry Pi gestartet. Tastatur und Maus angeschlossen	
Testdaten	
Keine vonnöten	
Schritte	**Resultat**
1. Der erste Screen bietet eine Möglichkeit das System zu konfigurieren?	Ja
2. Auf dem aktiven Screen in den Configurations-Screen wechseln.	-
3. Button sollte innerhalb von 2. Sekunden reagieren.	Ja
4. Das System wechselt in den Configurations-Screen.	Ja
5. Das System zeigt einen Texteditor für das Ändern der Config an	Ja
6. Das System zeigt eine Liste von verfügbaren Audiogeräten an.	Ja
7. Das System zeigt einen Testmodus für das Audio an.	Ja
8. Config werte anpassen und Blackout verlassen mit dem Exit knopf	-
9. Blackout starten und prüfen ob die Config übernommen worden ist	Ja

Status → Test has

Die Bestimmung des Bestanden / Nicht Bestanden-Status hängt davon ab, wie das erwartete Ergebnis und
das tatsächliche Ergebnis miteinander verglichen werden.

Gleiches Ergebnis = Bestanden/Pass → **Unterschiedliche Ergebnisse** = Fehlschlagen/Fail →

Eventuelle Bemerkungen

Testmandate: «Manuelle Test: Versus» (bis und mit 31.05.2023)

Table 5. Testmandat A (Andri)

Datum	Testrun	Result	Link
30.05.2023	Philip und Andri		Testrun

Testdurchlauf vom Test: Team Andri

Datum	31.05.2023
Verantwortlich	Andri Pieren

Team

Beschreibung	
In diesem Testfall wird geprüft, ob der Versusmodus gespielt werden kann	
Vorbedingung	
Blackout Applikation ist auf dem Raspberry Pi gestartet.	
Testdaten	
Keine vonnöten	
Schritte	**Resultat**
1. Blackout im Versusmodus starten und den Zufallsmodus wählen.	-
2. Fragen werden angezeigt.	Ja
3. Alle Fragen beantworten und Antworten merken	-
4. End-Screen mit QR-Code wird angezeigt.	Ja
5. QR-Code scannen und prüfen, ob die richtigen Resultate angezeigt werden.	Ja

Status → Test has

Die Bestimmung des Bestanden / Nicht Bestanden-Status hängt davon ab, wie das erwartete Ergebnis und
das tatsächliche Ergebnis miteinander verglichen werden.

Gleiches Ergebnis = Bestanden/Pass → **Unterschiedliche Ergebnisse** = Fehlschlagen/Fail →

Eventuelle Bemerkungen

Testmandate: «Manuelle Test: Einzel» (bis und mit 31.05.2023)

Table 6. Testmandat A (Andri)

Datum	Testrun	Result	Link
30.05.2023	Philip und Andri		Testrun

Testdurchlauf vom Test: Einzel

Datum	31.05.2023
Verantwortlich	Andri Pieren

Einzel Modus spielen

Beschreibung	
In diesem Testfall wird geprüft, ob der Einzelmodus gespielt werden kann	
Vorbedingung	
Blackout Applikation ist auf dem Raspberry Pi gestartet.	
Testdaten	
Keine vonnöten	
Schritte	**Resultat**
1. Blackout im Einzelspielermodus starten und den Zufallsmodus wählen.	-
2. Fragen werden angezeigt.	Ja
3. Alle Fragen beantworten und Antworten merken	-
4. End-Screen mit QR-Code wird angezeigt.	Ja
5. QR-Code scannen und prüfen, ob die richtigen Resultate angezeigt werden.	Ja

Status → Test has

Die Bestimmung des Bestanden / Nicht Bestanden-Status hängt davon ab, wie das erwartete Ergebnis und
das tatsächliche Ergebnis miteinander verglichen werden.

Gleiches Ergebnis = Bestanden/Pass → **Unterschiedliche Ergebnisse** = Fehlschlagen/Fail →

Eventuelle Bemerkungen