

Meine App

FÜR FORTNITE

Version 1.0.0, 9. Juli 2021 | Leandro Gilomen und Savio Elias Häsler

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc76714236)

[2 User Stories 3](#_Toc76714237)

[3 Mockups 4](#_Toc76714238)

[4 Technische Realisierung 5](#_Toc76714239)

[5 Testing 6](#_Toc76714240)

[5.1 Manuelle UI-Tests 6](#_Toc76714241)

[5.2 Testauswertung 7](#_Toc76714242)

[6 Fazit 8](#_Toc76714243)

# Abstract (Kurzbeschreibung)

Im üK 335 bekamen wir die Chance eine eigene Android Applikation zu programmieren. Wir bildeten eine zweier Gruppe und überlegten uns, was wir realisieren möchten. Da wir schon recht viel mit Datenbanken zu tun hatten, wollten wir eine API-Applikation erstellen. Als wir ein bisschen im Internet recherchierten, entdeckten wir die [Website Fortnite-API](https://fortnite-api.com/). Schlussendlich haben wir uns dazu entschieden. Unsere Applikation stellt immer den aktuellen Shop von Fortnite dar. Weiter kann man beim Klicken auf einen Shop Gegenstand mehr Details darüber erhalten. Auf einem letzten Tab werden dann noch aktuelle News vom Spiel dargestellt.

# User Stories

Als Benutzer möchte ich wissen, welche Skins im Shop sind, um das Budget planen zu können.

Als Benutzer möchte ich schnell einen Überblick über die Skins erhalten, um Zeit zu sparen.

Als Benutzer möchte ich die Seltenheitsfarbe der Skins sehen, um direkt zu wissen was für eine Seltenheit sie haben.

Als Benutzer möchte ich mehr Bilder zum Skin erhalten, um die verschiedenen Skin varianten sehen zu können.

Als Benutzer möchte ich den Skin genauer untersuchen, um mehr Informationen darüber zu erhalten.

Als Benutzer möchte ich den Spielgegenstand in einer Detailansicht ansehen können, um meinen Kaufentschluss zu bestätigen.

Als Benutzer möchte ich die Preise der Skins sehen, um sie miteinander zu vergleichen.

Als Benutzer möchte ich täglich die neusten Infos erfahren, um auf dem neusten Stand sein zu können.

# Mockups

1. Mainactivity

Zuoberst wird eine ActionBar dargestellt, wo man schnell von Fragment zu Fragment kommt. Die ActionBar beinhaltet einen grossen Titel und zwei Tab Titel. Die Tab Titel beschreiben jeweils das Fragment.

* 1. Shopfragment

Auf dem Shop Fragment werden die einzelnen Shop Gegenstände, wo heute gerade im Shop sind, dargestellt. Die Shop Gegenstände sind in einem Grid schön angeordnet, dass immer zwei nebeneinander platz haben. Diese passen sich auch dynamisch, je nach Bildschirmgrösse an. Ein Shop Gegenstand hat ein Bild, einen Namen und eine Seltenheitsstufe als Hintergrundfarbe von einem Skin, einem Emote, einer Spitzhacke usw. Mit einem Swipe nach links kann man vom Shop Fragment auf das News Fragment gelangen oder beim drücken auf den gewünschten Tab Titel in der ActionBar.

* 1. Newsfragment

Auf dem News Fragment werden die aktuellen Infos vom Spiel dargestellt. Ein Artikel hat eine Überschrift, ein passendes Bild, eine Subüberschrift und eine Beschreibung. Insgesamt werden immer vier Artikel auf dem News Fragment dargestellt. Umgekehrt zum Spop Fragment kommt man mit einem Swipe nach rechts vom News Fragment zum Shop Fragment oder beim Drücken auf den gewünschten Tab Titel in der ActionBar.

1. Detailactivity

Auf dem Detailactivity sieht man die Detailansicht von einem einzelnen Spiel Gegenstand. Zuoberst in der ActionBar hat es einen Pfeil, der zurück auf das Shop Fragment verweist. Daneben steht immer der Namen vom Spielgegenstand. Darunter sieht man einen ImageSlider, der mehr Bilder anzeigt. Wie zum Beispiel weitere Stilvarianten oder Accessoires vom Skin. Nachfolgend wir der Preis in Vbucks angezeigt und zu unterst einen kleinen Text über den Spiel Gegenstand. Auf diese Activity gelangt man, indem man auf den gewünschten Gegenstand drauf klickt. Zurückt geht es wie schon gesagt mit dem Pfeil oben links.

# Technische Realisierung

Wir verwendeten für unser Projekt eine online Gratis verfügbare Fortnite API, diese gibt einem einen JSON String zurück welcher wir mit der Klasse NewsJSONParser auslesen. Das obige Klassendiagramm zeigt lediglich wie wird die News Daten auslesen und nicht zusätzlich noch den Shop. Um die Daten zu speichern erstellten wir die Klasse News in welche wir durch Project Lombok getter und setter automatisch erstellen liessen. Der Json String erhalten wir durch einen JsonObjectRequest welcher wir durch eine RequestQueue einen GET Request absetzten auf die Fortnite API. Die API enthält einen Link zu den Bildern wie auch Titel und Description der News

# Testing

## Manuelle UI-Tests

In diesem Kapitel definiert ihr die Tests die Ihr macht.   
Es müssen minimal 5 Unit-Tests, 3 automatische UI-Tests (Espresso) und 2 manuelle UI-Tests gemacht werden. Auf die Unit-Tests und die automatischen UI-Tests soll hier verwiesen werden, die manuellen UI-Tests hier definiert werden.

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | Testfallnummer (ST = Systemtest) |
| Anforderungen | Welche Anforderungen werden durch diesen Testfall abgedeckt. (User Stories) |
| Vorbedingungen | Was muss gegeben sein, damit dieser Test durchgeführt werden kann? |
| Ablauf | Welche Schritte werden bei der Durchführung des Tests durchlaufen? |
| Erwartetes Resultat | Was sollte nun passiert sein? |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | ST-01 |
| Anforderungen | US-01; US-03 |
| Vorbedingungen | In der Datenbank existiert ein Benutzer, welcher gesperrt ist. |
| Ablauf | 1. Die App wird gestartet damit das Login-Formular erscheint 2. Der korrekte Benutzername sowie das korrekte Passwort werden eingegeben. 3. Der Button mit dem Label „Login“ wird geklickt |
| Erwartetes Resultat | Ein Toast mit dem Text «Login erfolgreich» wird angezeigt. Die App wechselt zu der Ansicht mit den favorisierten Schwimmbäder |

## Testauswertung

Zusammenfassung aller durchgeführten Tests. Nur fehlgeschlagene Tests und Tests mit Bemerkungen müssen in der folgenden Tabelle aufgelistet werden.

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| ST-01 | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| … | … | … |
|  |  |  |

# Fazit

Wir sind der Meinung, dass uns dieses Projekt sehr gut gelungen ist. Wir konnten alle unsere Ziele erreichen und sind sehr zufrieden damit. Wir konnten uns gut an unser Mockup halten und konnten alle Komponenten die wir wollten implementieren. Weniger gut funktionierte das Testing. Wir konnten weder UI-Tests (Espresso) noch Unit-Tests durchführen. Irgendetwas funktionierte nicht und unsere Lehrkraft konnte uns leider auch nicht mehr weiterhelfen.

Mit dem Endergebnis sind wir sehr zufrieden. Wir finden, wir haben unsere Erwartungen übertroffen. Die Applikation ist schön und ansprechend gestaltet. Weiter funktioniert sie einwandfrei und flüssig.

In dieser üK-Woche haben wir sehr viel neue Dinge gelernt. Zuvor hatten wir noch nie mit API gearbeitet und es war eine komplett neue Erfahrung. Wir wollten extra keine Applikation mit Datenbank machen, weil wir dies schon zu genüge getan haben. Anfangs Woche war es noch sehr komplex doch mit der Zeit verstanden wir es immer wie besser. Was wir sicher Neues dazu lernten, ist

Wir konnten eigentlich alles was wir geplant hatten umsetzen. Weiter wollten wir noch einen Timer einbauen, der herunterzählt, bis der Shop wieder aktualisiert und neue Skins anzeigt. Dies konnten wir aber nicht erledigen, weil uns die Zeit fehlte. Aber es wäre eine Erweiterung in der Zukunft Wert. Schlussendlich sind wir sehr zufrieden mit dem Projekt und es bereitete viel Spass.