

stock-simulator

Jan Müller
Jonas Thelemann
Juri Lozowoj
Lucas Held

27. März 2020

Inhaltsverzeichnis

1	Einleitung	4
2	Technische Details	4
2.1	Architektur	4
2.2	APIs	4
2.2.1	IEX Cloud	4
2.2.2	CoinGecko	4
2.3	Bibliotheken	4
2.3.1	Android Jetpack	4
2.3.2	Moshi	5
2.4	Room	5
2.4.1	Retrofit	5
3	Funktionalität	5
4	Teamwork	5
5	Zusammenfassung	5
6	Fazit	5
7	Ausblick	5

```

1 private suspend fun StockbrotQuote.executeSellOrder(quote:
  ↳ Quote) {
2     val depotQuote = accountRepository.depotQuoteBySymbol(
  ↳ id) ?: return
3     if (quote.latestPrice >= minimumSellPrice) {
4         val amount = depotQuote.amount
5         Timber.i("Bot is selling $amount for ${quote.
  ↳ latestPrice}")
6         accountRepository.sell(quote, amount)
7     }
8 }

```

Listing 1: example kotlin code

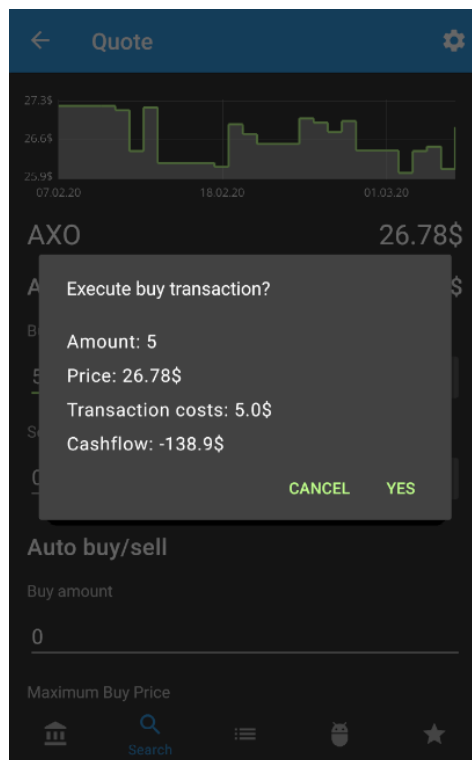


Abbildung 1: example image

1 Einleitung

Aufgabe der Veranstaltung „Code-Camp Context-Awareness 1“ war die Entwicklung eines Börsensimulator Spiels für Android Smartphones. Dabei stellte das Verwenden von echten Aktienkursen, mit aktuellen Daten, eine wichtige Anforderung an das Spiel dar. Diese Daten sollten als Basis für alle weiteren Funktionen dienen. Mithilfe des zur Verfügung stehenden Spielgeldes soll der Benutzer die Möglichkeit haben, Aktien oder Kryptowährungen zu kaufen. Um das Kaufen von bestimmten Aktien und Kryptowährungen zu ermöglichen, wurde eine Suchfunktion gefordert, welche die verfügbaren Elemente filtert und anzeigt. Um das Budget immer im Blick zu behalten, muss es Depotübersicht, sowie eine Anzeige für den aktuellen Kontostand und dessen Verlauf geben. Eine Historie soll die Käufe und Verkäufe in der Vergangenheit darstellen. Als Darstellungsform der Kurse und des Kontoverlaufs sind Graphen zu wählen. Wie auf gängigen Tradingplattformen soll auch der Simulator mit jedem Kauf- oder Verkauf Transaktionskosten berechnen. Ein Bot soll das Traden übernehmen, falls dies vom Benutzer gewünscht wird. Entsprechende Zielwerte sollen für jede Aktie oder Kryptowährung anpassbar sein. Damit der Benutzer die Möglichkeit hat, das Spiel neu zu beginnen, muss die Anwendung eine Option bieten, den Spielstand zurückzusetzen. Zusätzlich zu den bisher genannten Hauptfeatures, wird mindestens ein Zusatzfeature gefordert, welches eine nützliche Erweiterung für die Anwendung darstellt. Ziel des Spiels ist es, das Startkapital im Laufe der Zeit möglichst stark zu vermehren.

2 Technische Details

TODO

2.1 Architektur

TODO

2.2 APIs

TODO

2.2.1 IEX Cloud

TODO

2.2.2 CoinGecko

TODO

2.3 Bibliotheken

TODO

2.3.1 Android Jetpack

TODO

2.3.2 Moshi

TODO

2.4 Room

TODO

2.4.1 Retrofit

TODO

3 Funktionalität

TODO

4 Teamwork

TODO

5 Zusammenfassung

TODO

6 Fazit

TODO

7 Ausblick

TODO