Virtuelle Jukebox

Mobile App Client – Tools Resümee

David Böhm-Vrana

Mathias Dittrich

Tobias Egger

Paul Götzinger

Sophia Nunner

Projektnummer: PIE.XX.XXX

Version: 1.0

Datum: 18.12 2019

Änderungsverzeichnis

Versions	Datum	Änderung	Ersteller
1.0	18.12.2019	Erste Version	D. Böhm-Vrana, M. Dittrich, T. Egger, P. Götzinger, S. Nunner

Inhalt

1 Auflistung verwendeter Tools

1.1 Versionsverwaltung mit Git

- Repository gehostet auf GitHub (https://github.com/pagdot/Jukebox-App)
- Android Studio 3.5.3 Git Integration
- GitKraken 6.4.0
- Sourcetree 3.2.1

1.2 IDE

- Android Studio 3.5.3
- Kotlin Plugin Version 1.3.50-release-Studio3.5-1

1.3 Scrum Tool

- Trello mit Scrum by Vince Plugin (https://trello.com/b/VhfL13As/jukebox-app-woche-5-%F0%9F%94%A5-12-12-19-19-12-19)

1.4 GUI Mockup

- NinjaMock (https://ninjamock.com/s/LTMWMFx)

1.5 (Projekt)planung

- Projektstrukturplan: Microsoft Visio 2016
- Klassendiagramm: UMLet 14.3

2 Erfahrungen mit verwendeten Tools

2.1 Git

Positiv:

- Funktioniert wie erwartet. Keine größeren Probleme, weder in der Command Line, noch via GitKraken.

Negativ:

- Bei noch wenig Erfahrung mit Git kann leicht etwas schief gehen und es braucht einiges an Gewöhnung and die Funktionalität. Ein Git-Kurs im Bachelor wäre sehr von Vorteil!

2.2 GitKraken

Positiv:

- Angenehme UI
- Angenehme UX
- Macht was es soll

Negativ:

- Grundkenntnisse in Git müssen trotz GUI vorhanden sein.

2.3 Git-Integration von Android Studio

Positiv:

- Integriert in die IDE - kein zusätzliches Programm/Terminal-Befehle werden benötigt.
- Flotter Workflow bei Standard-Aktionen (Commit, Push, Pull)
- Kein Switchen zwischen Programmen nötig
- Gewohnte, angenehme Android Studio Bedienung
- Wie immer bei AS - viele Hilfestellungen im Web zu finden
- Ausgezeichnetes Merge-Tool

Negativ:

- Schlechte/gewöhnungsbedürftige Umsetzung bestimmter Funktionalitäten, zB. Auschecken eines Branches. Nach Einarbeitungsphase jedoch problemlos.
- Teilweise seltsames/fehlerhaftes Verhalten bei Pulls → Codeteile werden überschrieben, nicht nachvollziehbar automatisch gemerged

2.4 Android Studio

Positiv:

- Ausgezeichnete Autocompletion
- Ausgezeichnetes Syntax Highlighting
- Ausgezeichnete Emulator(Ausnahme in Negativ beschrieben), sowie Geräteintegration.
- Ausgezeichnete UI. Ansprechendes Design und übersichtliche/logische Strukturierung. Sehr gute Bedienbarkeit und kurze Einarbeitungszeit.
- Guter Debugger
- Guter Compiler. In der Regel treffen die Fehlermeldungen auch zu.
- Gute Integration des Compilers
- Verhältnismäßig geringe Downloadgröße(Ein paar 100 MB gegen die mehreren GB von Visual Studio)
- Gute Integration des Gradle-Systems
- Einfache Integration benötigter Libraries im Gradle System
- Auf Linux verfügbar
- Unkomplizierte Installation über snap
- Zahlreiche Hilfestellungen und Tutorials im Web
- Automatische Speicherung, gute Synchronisierung mit File-System

Negativ:

- Einige Emulatorimages(in der IDE selbst heruntergeladen) haben nicht funktioniert.
- Auslesen des Android-App-Logs nicht straightforward. Funktion hierfür meiner Meinung nach sehr versteckt.
- Nach Aktivierung dieser Auslese-Funktion überlappten sich bei mir der Debug-Tab und der Log-Tab, sodass immer nur einer der Tabs verwendet werden konnte, nicht aber beide.
- Lösung des Problems war die Verschiebung der Tabs, der Tab ließ sich nur leider nicht dort andocken, wo ich es wollte, damit war der Workflow gestört.

2.5 Kotlin

Positiv:

- Viel Syntactic Sugar
- Leistungsstarke Standard-Bibliothek
- Verhältnismäßig schnell zum Einarbeiten
- Gute Dokumentation
- Zahlreiche Hilfestellungen und Tutorials im Web zu allen benötigten Themen fürs Projekt
- Wenig Code für viel Funktionalität, mächtige Funktionen (Vgl. C++ mit Kotlin)

Negativ:

- Deep-Copy von Listen und Objekten stellt sich meiner Meinung nach schwieriger als in C++ dar.
- Obwohl schnell zum Einarbeiten, in die benötigte Denkweise kam ich erst spät rein.
- Übertriebene interne Nullpointer Checks und Asserts(Operatoren?. und !!.)
- Gewöhnungsbedürftige Syntax (Mischung von { } und ( ), zahlreiche Lambda- und verschachtelte Funktionen), teilweise sehr „Magic“

2.6 Trello

Positiv:

- Gut für die Organisation von Tasks innerhalb des Teams
- Ansprechendes Design und benutzerfreundliche UI. Intuitiv zu bedienen, übersichtlich

Neutral:

- Verhältnismäßig unwichtig und lediglich Zusatzaufwand, wenn man ohnehin der Einzige ist, der an einem gegebenen Unterprojekt arbeitet (z.B. Verwaltungsschicht). Arbeitspakete ohnehin im Projekthandbuch definiert. Zeiteinteilung ist in dem Fall ohnehin größtenteils mir selbst überlassen.

Negativ:

- Teils überladene UI

2.7 NinjaMock

Positiv:

- gratis für Grundfunktionalitäten
- Nicht überladene UI, einfach zu bedienen
- Webapplikation, keine Programminstallation nötig
- Gemeinsames, gleichzeitiges Arbeiten am gleichen Projekt möglich
- Alle nötigen Grundelemente vorhanden
- Speichern des Projekts in persönlichem Account
- Teilen von Projekten möglich zur gemeinsamen Bearbeitung
- Integrierte Demo-Funktion: Switchen zwischen Screens kann demonstriert werden
- Auch ohne Benutzeraccount kann ein geteiltes Projekt betrachtet werden

Neutral:

- Teilweise sehr auffällige Fehler, z.B.: fehlerhafte Synchronisation bei gemeinsamer Bearbeitung, bereits platzierte Objekte verschwinden oder können nicht mehr ausgewählt werden

Negativ:

- Allgemein trotzdem das beste (gratis) Mockup-Tool, das wir finden konnten. Die sporadischen Fehler können verkraftet werden und die gemeinsame Bearbeitung sowie das Teilen des Projekts zur Betrachtung für andere Projektmitglieder ist äußerst praktisch.