Eine Email-Client-App entwickeln

Noah Vogt & Simon Hammer

Geschrieben im Jahr 2021

Inhaltsverzeichnis

1 Vorwort

2 Einleitung

2.1 Ideenfindung

3 Ziel der Arbeit

was, wie und warum [machen wir das?]

4 Konzept der Arbeit

4.1 Funktionsweise

4.1.1 Vergleich mit Konkurrenz

4.2 Quellcode Modell

4.3 Philosophie (suckless)

4.3.1 Hintergründe, Technologisch, UNIX, KISS

4.4 Lizensierung

4.4.1 Hintergründe, philosophisch, technologisch

5 Arbeitsprozess

5.1 Hardware

5.1.1 Smartphones

für testing

5.1.2 PC / Laptop

für programmierung selber

5.2 Software

Aufgrund dessen, dass ein umfassendes Programm entstehen soll, wird auch Gebrauch von einigen anderen Programmen gemacht. In den nachfolgenden Seiten wird beschrieben welche Programme genutzt werden, wieso diese ausgewählt wurden und wie der Umgang mit Ihnen war.

5.2.1 Programme

5.2.1.1 Version Control System

5.2.1.1.1 Git Git und GitHub sind wohl die wichtigsten Programme die genutzt wurden. Sie sind Systeme, welche Fileordner (repository) verwalten können und sie für mehrere Computer zur verfügung stellen, wobei sie sehr viele praktische Funktionen mit sich bringen. Mit Git können repositorys local auf Computer oder Hardware geteilt werden, mit GitHub könne die repositorys auch über das Internet geteilt werden. Der einfachheitshalber wird nicht zwischen Git und GitHub unterschieden.

5.2.1.1.2 Github

5.2.1.2 IDE

android studio

5.2.1.3 Texteditor

vim

5.2.1.4 Android Emulator(s)

5.2.1.5 Compiler

gradle, maven

5.2.1.6 Installer / Packaging

erklären

5.2.1.6.1 signing keys

5.2.1.6.2 Android Debugging Bridge

5.2.1.7 Open Source Programme

Beim Programmieren kann es sehr Hilfreich sein Programme zu haben welche, ähnliche Funktionen haben wie das Programm welches entstehen soll. Solche Vorlagen können beliebig getestet und verändert werden. Simon hatte zu beginn Schwierigkeiten Java zu nutzen, um Programme zu schreiben, da er noch nicht viel Erfahrung mit dem Programmieren hatte. Um sich mit der Art der Sprache und des Programmierens vertieft auseinander zu setzen, begann er Email-Apps, welche Open Source waren, genauer zu Betrachten. Im folgenden Text werden wir diese Programme aufführen und zeigen für was wir sie gebraucht haben.

5.2.2 Librarys

5.2.2.1 RecyclerViewer

Der Recyclerviewer ist ein Behälter in welchen Daten gepackt werden. Er wird dem Layout hinzugefügt und hat eine grossen Vorteil gegenüber Listen. Eine Liste wird einmalig erstellt und komplett generiert. Das heisst es gibt Behälter, welche existieren, aber nicht auf dem Bildschirm angezeigt werden. Diese Behälter brauchen aber dennoch Speicherplatz, sind aber sinnlos. Hingegen der Recyclerviewer generiert nur so viele Behälter wie auf dem Bildschirm angezeigt werden können. Die Behälter, welche beim Scrollen am Bildschirm ende ankommen werden sogar wieder verwendet, mit neuen Information gespeist und am anderen Ende des Bildschirms angezeigt.
Bild von RecycleView
Ich (Simon) habe viel Zeit in einem template für den Recyclerviewer verbracht, da mir viele Internetseiten nicht helfen konnten. Es waren die ersten Schritte um richtig zu verstehen wie eine App funktioniert und wie sie Aufgebaut ist. Ich habe mich davor schon informiert jedoch wurde es mir dort richtig klar. Ich habe gelernt wie Behälter über einen Key von Java Files aufgerufen werden und mit Daten gespeist werden. Ebenso habe ich verstanden, weshalb in Programmen, basierend auf Java, viele Klassen erstellt erstellt werden müssen, weil jeweils nur einmal die Variablen, Funktionen und Konstruktor einer andere Klasse implementiert werden können. Das heisst, wenn eine Klasse zwei Funktionen aus zwei unterschiedlichen Klassen verwendet werden soll, muss eine der beiden Klassen die andere Klasse implementieren.

5.3 Recherche Tools / Quellen

5.3.1 Internet

duckduckgo, arch wiki, stackoverflow

5.3.2 Bücher, Artikel

5.4 Dokumentation(-stools)

5.4.1 Latex

5.4.2 Pandoc

6 Programmstruktur (=? Code)

6.1 Sicherheit / Security (Features)

6.1.1 PRIVATE MODE, Sandbox

6.2 Code Kompaktheit

6.2.1 Maintaining

6.2.2 less bug/error-prone

7 Umsetzung

7.1 neutraler Bericht

verweis texdiary

7.2 Beispiele aus der Umsetzung

7.2.1 Bugs

erklärung woher, warum falsch, wie gelöst

7.2.2 Probleme / Hiccups

Gründe

7.2.3 Kommunikation

8 Einschätzung / Schlussfolgerung

8.1 Fremdeinschätzung

8.2 Selbsteinschätzung

8.2.1 was lief gut

8.2.2 was lief schlecht

8.2.3 was würden wir gleich machen

8.2.4 was würden wir anders machen

8.2.5 abschliessende persönliche Schlussfolgerung

9 Danksagung