



# Sebastian Spies

SENIOR MOBILE APP DEVELOPER

## Profil

Erfahrener und kommunikativer Software-Entwickler mit dem Schwerpunkt auf nativer iOS & Flutter Programmierung sowie mit einer starken Affinität im Bereich UX/UI. Im Laufe meiner beruflichen Karriere konnte ich bereits in verschiedenen Bereichen fundiertes Fachwissen sammeln. Als meine größte Stärke sehe ich mein lösungsorientiertes Denken, welches es mir auch in Drucksituationen ermöglicht, stets den Überblick zu behalten.

## Ausbildung

**Bachelor of Science in Computervisualistik & Design, Hochschule Hamm-Lippstadt, Lippstadt**

SEPTEMBER 2011 – JULI 2015

## Berufserfahrung

**Senior Mobile App Developer, Freelancer**

SEPTEMBER 2023 – HEUTE

**Senior iOS Developer**

- TK-App
- STIHL connected App
- Dickert App

**Senior Flutter Developer**

- STIHL connected 2.0 App
- DWIDSapp
- SV Lippstadt 08 App

**Senior iOS Developer, grandcentrix GmbH, Remote**

AUGUST 2022 – SEPTEMBER 2023

**Senior iOS Developer**

- Leica FOTOS App
- RE.GUARD 2.0 App
- Entwicklung verschiedener Prototypen (PoC) für potentielle neue Produkte.

**iOS Developer, appsoluts GmbH, Düsseldorf**

OKTOBER 2020 – JULI 2022

**iOS Developer**

- Fortuna Düsseldorf App
- HappieHaus App
- Diloans App
- Sanguinum App

## Persönliche Angaben

Steinkribbenstraße 9

40597, Düsseldorf

Deutschland

+49 15678 661186

[sebastian@spies-digital-solutions.com](mailto:sebastian@spies-digital-solutions.com)

NATIONALITÄT

Deutsch

GEBURTSDATUM UND ORT

20.10.1991

Lippstadt

## Accounts

LinkedIn

GitHub

## Programmiersprachen & Tools

Swift

Dart

Javascript

Objective-C

PHP

Xcode, Visual Studio Code

Gitlab, Github

Fastlane

Figma

Mattermost, Slack

Google Workspace

Shell Scripting (Basics)

UIKit/ SwiftUI

CI/CD (Github Actions, GitLab

Runner, Azure Dev Ops)

Swift Package Manager,

CocoaPods

Unit/UI-Testing

NodeJS

Agile software development

(Scrum)

Jira & Confluence, Trello

## Fähigkeiten

Teamgedanke

## iOS Developer & Team Lead Dev, Adversign Media GmbH, Düsseldorf

JULI 2015 – SEPTEMBER 2020

### iOS Developer

- viewneo Player App
- Diwa App
- Entwicklung verschiedener Prototypen (PoC) für potentielle neue Produkte & Showcases für Messen.

### Team Lead

Leitung eines Entwicklungsteams von mehr als 10 Entwicklern.

- Verantwortlich für die Kommunikation zwischen DEV-Team mit anderen Teams.
- Ansprechpartner für Kunden

## Kurse

### iOS Lead Essentials, Essential Developer Academy

JANUAR 2022 – AUGUST 2022

## Projekte

### STIHL connected 2.0 App, Onsite + Remote

DEZEMBER 2024 – JANUAR 2026

#### Neuentwicklung der STIHL Connected App

- Verwendung von FlutterBluePlus für die Kommunikation mit IoT-Geräten
- Verwendung des HTTP-Protokolls und Rest-Services für die Kommunikation mit dem Backend
- Verwendung von Riverpod für das State-Management
- Unit-Tests & UI-Tests
- Fastlane, CI/CD (Github Actions, Azure Dev Ops), Git Flow, Jira & Confluence
- Verwaltung der CI/CD Architektur

### TK-App, Onsite + Remote

SEPTEMBER 2024 – MAI 2025

#### Weiterentwicklung der TK-App mit Fokus auf den TK-Gesundheitsmessenger

- Verwendung des HTTP-Protokolls und Rest-Services für die Kommunikation mit dem Backend
- Verwendung des Matrix-Protokolls für Chat-Kommunikation
- Verwendung von SwiftUI & UIKit für die Benutzeroberfläche, MVVM
- Unit-Tests
- Fastlane, CI/CD (Jenkins), CocoaPods, SPM, Combine, Git Flow, Jira & Confluence

### DWIDSapp, Remote

JULI 2024 – HEUTE

#### Implementierung & Wartung des Backends und der plattformübergreifenden App von Grund auf.

- Alleinige Verantwortung für die Architektur für App & Backend
- Backend-Implementierung mit NodeJS basierend auf dem Headless CMS strapi.
- App-Implementierung mit Dart & Flutter basierend auf dem BloC-Pattern.

Eigenverantwortung

Kommunikation

Führungsqualitäten

Analytische Fähigkeiten

Belastbarkeit

### Sprachen

Deutsch

Englisch

- Verwendung von http-Protokoll & rest-Services zur Kommunikation mit dem Backend.
- Verantwortlich für das Hosting des Backends und die Freigabe der Apps für Android & iOS.
- Unit-Tests
- Fastlane, CI/CD (Github-Actions)
- Die App soll im September in den Stores veröffentlicht werden.

## Dickert App, Remote

DEZEMBER 2023 – DEZEMBER 2024

### Wartung & Weiterentwicklung der Dickert App

- Update auf die Neueste iOS & Swift Version
- Implementierung kleinerer neuer Features
- CoreBluetooth für Kommunikation mit BLE Modul

## STIHL connected App, Onsite + Remote

SEPTEMBER 2023 – AUGUST 2024

### Weiterentwicklung der STIHL Connected App

- Verwendung von CoreBluetooth für die Kommunikation mit IoT-Geräten
- Verwendung des HTTP-Protokolls und Rest-Services für die Kommunikation mit dem Backend
- Verwendung von SwiftUI & UIKit für die Benutzeroberfläche, MVVM
- Unit-Tests
- Fastlane, CI/CD (Github Actions, Azure Dev Ops), CocoaPods, SPM, RxSwift, Combine, Git Flow, Jira & Confluence
- Verwaltung der CI/CD Architektur

## SV Lippstadt 08 App, Remote

OKTOBER 2022 – JULI 2023

### Implementierung des Backends und der plattformübergreifenden App von Grund auf.

- Alleinige Verantwortung für die Architektur für App & Backend
- Backend-Implementierung mit NodeJS basierend auf dem Headless CMS strapi.
- App-Implementierung mit Dart & Flutter basierend auf dem BloC-Pattern.
- Verwendung von http-Protokoll & rest-Services zur Kommunikation mit dem Backend.
- Verantwortlich für das Hosting des Backends und die Freigabe der Apps für Android & iOS.
- Unit-Tests
- Fastlane, CI/CD (Github-Actions)

## Leica PHOTOS App

JANUAR 2023 – AUGUST 2023

### Implementierung neuer Funktionen für diese App (Scrum)

- Arbeiten mit dem Bluetooth-, WiFi- und MFI-Protokoll für die Kamerakommunikation.
- Implementierung von zwei PoCs für neue Kameras, um die Machbarkeit der Implementierung in die App zu prüfen.
- Modularisierung der App.
- Arbeiten mit MVVM und Coordinator-Pattern
- Unit-Tests

- Fastlane, CI/CD (Github Actions), CocoaPods, SPM, RxSwift, Combine, Git Flow, Jira & Confluence

## **RE.GUARD 2.0 App**

AUGUST 2022 – JANUAR 2023

### **Implementierung der RE.GUARD 2.0 App (Scrum)**

Arbeiten mit dem Bluetooth-Protokoll für die Kommunikation mit den IoT-Geräten.

- Arbeiten mit MVVM und Coordinator-Pattern
- Unit-Tests, UI-Tests
- Fastlane, CI/CD (Github Actions), SPM, Combine, Git Flow, Jira & Confluence

## **Fortuna Düsseldorf App**

OKTOBER 2020 – JULI 2022

### **Die App von Grund auf neu erstellt**

- Alleinige Verantwortung für die Architektur der App
- Einrichten einer vollständigen CI/CD-Pipeline (Gitlab & GitlabRunner)
- Verwendung von http-Protokollen und Rest-Services zur Kommunikation mit dem Backend.
- Verwendung von UIKit & MVVM-Muster
- CoreData für lokale Speicherung
- Unit- und UI-Tests
- Fastlane, CI/CD (GitLab Runner), CocoaPods, Git Flow, Trello

## **HappieHaus App**

OKTOBER 2020 – JULI 2022

### **Die App von Grund auf entwickelt**

- Alleinige Verantwortung für die Architektur der App
- Einrichten einer vollständigen CI/CD-Pipeline (Gitlab & GitlabRunner)
- Verwendung von http-Protokollen und Rest-Services zur Kommunikation mit dem Backend.
- Verwendung von SwiftUI, MVVM-Muster & CoreData für die Datenspeicherung
- Unit- und UI-Tests
- Fastlane, CI/CD (GitLab Runner), CocoaPods, SPM, Combine, Git Flow, Trello

## **Diloans App**

OKTOBER 2020 – JULI 2022

### **Die App von Grund auf entwickelt**

- Alleinige Verantwortung für die Architektur der App
- Einrichten einer vollständigen CI/CD-Pipeline (Gitlab & GitlabRunner)
- Verwendung von http-Protokollen und Rest-Services zur Kommunikation mit dem Backend.
- Verwendung von SwiftUI, MVVM-Muster & CoreData für die Datenspeicherung
- Unit- und UI-Tests
- Fastlane, CI/CD (GitLab Runner), CocoaPods, SPM, Combine, Git Flow, Trello

## **Sanguinum App**

OKTOBER 2020 – JULI 2022

### **Die App von Grund auf entwickelt**

- Alleinige Verantwortung für die Architektur der App
- Einrichten einer vollständigen CI/CD-Pipeline (Gitlab & GitlabRunner)

- Verwendung von http-Protokollen und Rest-Services zur Kommunikation mit dem Backend.
- Verwendung von UIKit, Storyboard, MVC-Muster & CoreData für die Datenspeicherung
- Unit- und UI-Tests
- Fastlane, CI/CD (GitLab Runner), CocoaPods, SPM, Git Flow, Trello

## **Diwa App**

SEPTEMBER 2019 – SEPTEMBER 2020

### **Die App von Grund auf entwickelt**

- Alleinige Verantwortung für die Architektur der App
- Verwendung von http-Protokoll & Rest-Services zur Kommunikation mit dem Backend.
- Bluetooth-Kommunikation mit dem viewneo Diwa
- UIKit mit MVC-Muster
- CocoaPods, Jira & Confluence, Git Flow

## **viewneo Player App**

JULI 2015 – SEPTEMBER 2020

### **Die App von Grund auf entwickelt**

- Alleinige Verantwortung für die Architektur der App
- Verwendung von http-Protokoll & Rest-Services zur Kommunikation mit dem Backend.
- UIKit mit MVC-Muster
- CocoaPods, Jira & Confluence, Git Flow