

# 職務経歴書

## PR

大学在学中にプログラミングに興味を持ち、KotlinでのAndroidアプリ制作を通じて、エンジニアとしてのキャリアをスタートしました。その後、都内の受託開発企業で半年間のアルバイトを経て、AIチャットボットを提供するベンチャー企業の受託開発部門に参画し、Androidのビジネスチャットアプリ開発に従事しました。現在は、Flutterを用いて、建設業向けの施工管理サービスのアプリ開発に携わっています。

FlutterとAndroidのネイティブアプリ開発を得意としており、詳細設計からテスト・運用保守の工程までの業務フローを経験し、既存機能の改修および新規機能開発の経験があります。

未経験の技術に対しても、キャッチアップしながら業務を進めることが可能です。例として、FlutterとFirebaseを独学で学び、個人制作を行いました。また、ビジネスチャットアプリの開発でUIテストの自動化について、自身で公式ドキュメント等を調べながら実装しました。

これらの経験やスキルにより、スマホアプリ開発にて未経験の分野やツールを習得しながら、既存機能の改修および新規機能の開発まで対応できます。

## スキルセット

### 言語

Dart | Android Java | Java

### フレームワーク・ライブラリ

Flutter | Spring Boot

### RDB/NoSQL

SQLite | Realm | Oracle DB | Firestore | GraphQL

### クラウド

Firebase(Firestore, Storage, Authentication)

## 主な業務経歴

### 建設業向け施工管理サービスのスマホアプリ開発(2022/12 - 現在)

#### 【プロジェクト概要】

建設業向け施工管理サービスのスマホアプリの新規機能開発、運用保守

#### 【チーム情報】

Flutter2名、バックエンド&フロントエンド1名、ディレクター2名（全5名）

#### 【担当役割】

アプリ開発（Flutter）

#### 【使用した技術・ツール】

Flutter | Provider | Hive | Android Studio | Xcode | GraphQL | GitHub | Slack | Zenhub

#### 【担当業務】

- ・既存機能の不具合改修および仕様変更の実装。お知らせの既読処理の不具合改修、PDFビューアの不具合改修、スマホとタブレットでの文字の大きさを統一する等に対応。

#### 学んだこと

- ・自社サービスを開発しているWeb系企業の環境では、サービスを成功に繋げるために様々な意見を取り入れるため、自分のアイデアや意見を積極的に伝えられるようになった。

#### 苦勞したこと

- ・iOSアプリの開発は未経験のため、iOS特有のリリース作業や環境によるビルドエラーの解消等の理解に時間がかかった。

---

### 自動車運転免許管理システムの開発支援 (2022/03 - 2022/06)

#### 【プロジェクト概要】

自動車運転免許管理システムの大規模なバグ改修。

#### 【チーム情報】

フロントエンド40名以上、バックエンド40名以上、ディレクター約10名（全100名以上）

#### 【担当役割】

バックエンド

#### 【使用した技術・ツール】

Java(Spring Boot) | Oracle DB | JBoss | JaCoCo | GitLab | Subversion | Microsoft Teams | Redmine

#### 【担当業務】

- ・システムの有識者が作成したバグ修正方法の詳細設計書を基に、免許情報の検索・登録処理APIのバグを修正。
- ・バグ修正箇所の単体・結合テストのテスト仕様書作成およびテストを実施。テストの実施対象は修正箇所だけであまり多くなかったため、JUnitは使わず手動でEclipseのデバッグモードを用いて実施。JaCoCoでカバレッジを取得し、テスト実施のエビデンスとしてサーバーログ、修正箇所のスクリーンショット等をExcelに記録した。

#### 工夫したこと

- ・プロジェクトの進行が遅れていたため、自分のタスクを早く終え、チームメンバーのタスクを積極的に支援するよう努めた。
- ・詳細設計の不備を早期に発見および改修し、余計なバグ発生を未然に防止した。
- ・単体テストを教材で事前学習し、テスト全体の流れとテスト手法を早く理解することで、スムーズにテストの作成・実施を行うことができた。

#### 学んだこと

- ・実務でのバックエンド開発は本プロジェクトが初。過去に独学で簡易なWebアプリを制作したが、フレームワーク未使用で設計等は意識せずに開発していたため、本プロジェクトでフレームワーク(Spring Boot)を用いたMVCモデルの実装方法について知見を得た。
- ・単体テストと結合テストも初経験で、仕様書を作成する過程で同値分割や境界値分析の概念に触れたことにより、テストを意識してバグが出ないように考えながら実装できるようになった。

#### 苦勞したこと

- ・大規模なプロジェクトが初経験で、ウォーターフォール式の開発手法やExcelファイルで全てを管理する手法に慣れるのに苦勞した。その経験を通じて設計や記録をマメに残す習慣が身に付いた。

---

### ビジネスチャットのスマホアプリ開発およびUIリニューアル (2020/07 - 2021/09)

#### 【プロジェクト概要】

ビジネスチャットのスマホアプリのUIリニューアル、新規機能実装、UIテストの自動化。

#### 【チーム情報】

Android3名、iOS3名、バックエンド、ディレクター1名（全8名）

#### 【担当役割】

アプリ開発（Android）

#### 【使用した技術・ツール】

Android Java | Android Studio | SQLite | SkyWay | Espresso | Bitbucket | Sketch | Slack | Backlog | Redmine

#### 【担当業務】

- Androidアプリを新規UIデザインにリニューアル。デザイナーが作成した新規UIのデザインデータをデザインツール(Sketch)でSVGとして書き出し、チャット画面のレイアウトやボタン等の変更を行った。
- 個人間・複数人の通話機能の開発。SkyWayという音声・ビデオ通話のSDKを用いて実装。
- UIテストの自動化。テストフレームワークのEspressoを用いて、自動でUIの操作・判定を行うように実装。
- スマホアプリの新規画面遷移図を作成。アプリ画面のスクリーンショットを用いてExcelで作成した。

#### 学んだこと

- 本プロジェクトで初めてチケット管理ツール(Backlog)を使用し、タスクの詳細、担当者、期限等の情報をチケットで管理して開発を進める方法を覚えた。
- バックエンドエンジニアにJSONデータの値の変更を依頼したり、iOS版と実装が同じようになるように打ち合わせするなど、自分の担当分野以外のメンバーとコミュニケーションを逐次行った。
- 本プロジェクトで初めてコードレビューのレビュアー側を経験した。自分以外のコードをレビューすることで効率的なコーディングや、理解しやすい変数・メソッドの命名等を知ることができた。

#### 苦労したこと

- 本プロジェクトではRxJavaやRetrofitなどAndroidの主流なライブラリは使用されていなかった。Androidの標準APIや機能を使用しているため、複雑になりやすいAPIコールや画像取得周りの処理をシンプルに実装できるように努めていた。
- UIテストの自動化について経験のあるメンバーがいなかったため、自身で公式ドキュメントなどで調べながら実装していった。ここで技術のキャッチアップ力がより向上した。

---

## MAU500人以上のラジオストリーミングのスマホアプリ開発・運用保守 (2020/01 - 2020/05)

#### 【プロジェクト概要】

ラジオストリーミングのスマホアプリの新規機能実装、運用保守。

#### 【チーム情報】

Android5名、iOS6名（全11名）

#### 【担当役割】

アプリ開発（Android）

#### 【使用した技術・ツール】

Android Java | Android Studio | Realm | GitLab | Adobe XD | Microsoft Teams | Mattermost

#### 【担当業務】

- アプリ全体のバグ改修。初のエンジニア業務であったため、最初の1ヶ月はAndroid Java、Realm、Git、Adobe XDの基礎を覚えて、簡単なUI表示バグからビジネスロジックのバグ修正を対応した。
- 利用規約の同意機能の開発。アプリの初回起動、または既存ユーザーで利用規約に同意していない場合に、利用規約画面を表示する。同意したらトップ画面に遷移、同意しなければアプリを終了するように分岐させた。
- ジャンル検索機能の開発。新たにジャンル検索用の画面を作成し、各ジャンル名のタブを画面上部に配置し、選択したジャンルのラジオ番組が一覧で表示されるように実装。

#### 学んだこと

- 初めてエンジニアとして業務に従事したことで、独学では学べなかったAndroidの基礎技術、実務レベルのコーディングの方法、大量のコードから処理の流れを理解し、バグを発見する力とコツを身に付けた。
- チーム開発でのGitの運用方法（git-flow、ブランチ作成、マージリクエスト、コミット）を経験した。
- 「タスクの認識合わせ」「背景・意図・コード等を質問時に共有する」等のチーム開発のコミュニケーションを覚えた。