

# 職務経歴書

---

## 基本情報

Key	Value
氏名	飯塚 大河 (Iizuka Taiga)
生年月日	1993/05/03
GitHub	tiizuka
Qiita	kakikukeko09
Zenn	kakikukeko09

## 保有スキル

### フロントエンド

- TypeScript を用いたモダンなフロントエンド開発
  - フレームワーク : Vue.js, React
  - UI : Element Plus, Vuetify
  - ツール : Vite
- デザインツールを活用した画面設計・仕様作成
  - Figma, draw.io を使用した UI デザイン・ワイヤーフレーム作成
- UI/UX 向上のためのリファクタリング経験

### バックエンド

- Ruby on Rails, Scala (Play Framework) , Kotlin (Ktor) , Go を用いた API 開発・設計
- OpenAPI (Swagger) を用いた API 仕様策定
- SQL の調査・改善 (パフォーマンス改善含む)

### ネイティブアプリ

- Swift / Kotlin を用いた iOS/Android アプリ開発 (両OS経験あり)
- BLE連携アプリの開発経験

### 品質・改善 / 推進

- QA体制の立て直し (テスト観点整備、回帰テスト自動化の推進)
  - MagicPod を用いた回帰テスト運用・整備
- AIコードレビュー導入 (CodeRabbit)
- レビュー体制の改善 (相互レビュー体制の整備、レビューワー負荷分散)

## 技術スタック

カテゴリ	技術・ツール
------	--------

---

カテゴリ	技術・ツール
言語	TypeScript, Ruby, Scala, Kotlin, Swift, Go
フロントエンド	Vue.js, React, Element Plus, Vuetify
バックエンド	Ruby on Rails, Play Framework, Ktor
ネイティブアプリ	Swift, Kotlin
API 設計	OpenAPI (Swagger)
デザイン・設計	Figma, draw.io
テスト・品質	MagicPod
CI/CD	Bitrise, GitHub Actions
クラウド	AWS Amplify, Amazon Cognito, API Gateway, AWS Lambda
その他	GitHub, Notion, Firebase, MySQL, PostgreSQL

## 職務経歴詳細

スプリームシステム株式会社（現：株式会社GROWTH VERSE）（2022/04～現在）

エンジニアとして主にフロントエンドの開発に従事しています。

### MA ツール クラウド版開発プロジェクト（2022/04～2023/04）

20年以上提供してきたオンプレミス型MA（マーケティングオートメーション）ツールのクラウドサービス化の立ち上げに参画しました。2022年12月にリリースし、その後は新機能追加・運用保守・PRレビュー、インターン生メンター等も担当しました。プロジェクト途中から参加し、複数画面・機能の設計～実装を担当しました。

- **プロジェクトカテゴリ**： Webサービス
- **担当工程**： 設計 / コーディング / テスト / 運用・保守
- **開発言語**： TypeScript (Vue.js)
- **使用技術**： AWS Amplify / Amazon Cognito（非Hosted UI） / Element Plus / Figma
- **経験した職種・役割**： フロントエンド / デザイン（画面設計）
- **チーム構成**： 8名（フロントエンド4名、バックエンド3名、プロダクトオーナー1名）
- **開発手法**： スクラム（1週間スプリント）
- **担当業務**：
  - Figmaを用いた画面設計、Web画面の実装、テスト、運用保守、PRレビュー
  - 遅延状態からの立て直しとして、ダイアログ・一覧等の共通コンポーネント化による省力化／工数削減を推進
  - PRレビューがフロントエンドリーダーに集中していたため、ランダムアサインによる相互レビュー体制を整備し負荷分散
  - 外部委託デザインが未完成の状態で、担当機能のUI設計から実装まで自走して対応
  - リリース後の認証基盤をAWS Cognitoへ移行する際、前職のVue + Amplify + 非Hosted UIの経験を活かして不具合なく移行を推進
  - 自社初のインターン生メンターとして育成・サポートを担当

## LINE ID連携 技術調査 (2023/01~2023/02)

異なるシステムに分散している会員データとLINEアカウントを紐付けるための技術調査を実施しました。LINE Messaging APIを利用した配信に必要な「LINE ID」を、お客様側で紐付ける運用が難しいケースが増加した背景を受け、**自社提供の仕組みで紐付けを実現する方針**の技術検証を担当しました。調査の一環として、ID連携用のbot serverを開発し、試験運用まで対応しました。

- **プロジェクトカテゴリ**： Webサービス / 自社プロダクト
- **担当工程**： 設計 / コーディング / テスト / 運用・保守 / その他（技術調査）
- **開発言語**： Go
- **経験した職種・役割**： バックエンド
- **チーム構成**： 4名（バックエンドエンジニア2名、プロダクトオーナー1名、自分）
- **担当業務**：
  - LINE ID連携の実現方式に関する技術調査（Messaging APIの取り扱い、連携方式の検討）
  - ID連携用bot serverの設計・開発、テスト、試験運用、運用保守
  - 当初想定（フロントでの連携画面作成）を見直し、公式LINEからのMessaging APIをバックエンドで直接受けて連携処理を行う構成へ方針転換

## AI コードレビュー導入 (2023/09~2023/10)

PRレビューにかかる時間の削減とコード品質の向上を目的として、AIによるコードレビュー機能（CodeRabbit）の導入を担当しました。企画から技術検証、導入、運用までを1名で一貫して推進しました。

- **担当工程**： 企画 / コーディング / テスト / 運用・保守
- **チーム構成**： 1名（単独対応）
- **使用技術**： GitHub Actions / YAML / CodeRabbit
- **担当業務**：
  - CodeRabbitの調査・技術検証、費用試算、導入方針の整理
  - GitHub Actions（YAML）による導入実装、動作検証、運用保守
  - 関係者への導入周知、開発チーム向け勉強会の開催（運用方法のトランスファー）
- **成果**：
  - 導入後数か月間の計測で、PRマージ数が導入前比で **10~20% 改善**
  - 開発チーム・プロダクトオーナーから、差分要約による確認コスト削減・可視化に有効とのフィードバックを獲得

## MA ツール 新機能開発、運用・保守 (2023/02~現在)

クラウド版 MA ツールのリリース後フェーズにおいて、新機能開発・運用保守を継続して担当しています。Web/アプリ接客機能、シナリオ承認機能を主担当として開発し、画面仕様レベルの要件整理から設計・実装・テストまで一貫して対応しています。

- **開発言語**： TypeScript（Vue.js）
- **使用技術**： AWS Amplify / Amazon Cognito（Not Hosted UI） / Element Plus / Figma
- **チーム構成**： 10名（フロントエンド3名、バックエンド2名、デザイナー2名、プロダクトオーナー1名、インターン2名）
- **開発体制**：
  - 1週間スプリント
  - Epic＝機能開発単位（期間は規模により可変）

- 複数Epicが並行稼働し、メンバーがEpicを掛け持ちすることがある
- Epicオーナーは持ち回り制のため、同時期に「あるEpicではオーナー」「別Epicでは実装担当」を兼務することがある
- **担当業務：**
  - 画面仕様レベルの要件整理、コンポーネント設計、API設計へのコメント・調整
  - 新機能実装（主担当：Web/アプリ接客、シナリオ承認機能）
  - 実装者としての単体・結合テスト
  - 運用保守（障害調査・修正、軽微改善）
  - Epicオーナー（持ち回り）：要件整理・仕様決め・進行管理
  - インターン生メンター：オンボーディング、タスク設計、週次1on1、PRレビュー
- **失敗と改善（負債解消）：**
  - Web/アプリ接客機能で、納期都合・UX統一の意図・関係者要望・仕様理解不足が重なり、Web向けとアプリ向けを同一画面・同一処理に統合して実装。ターゲット条件の持ち方やチャネル固有設定などオブジェクト構造の差分が大きく、条件分岐の増加により拡張性が低下し負債化。
  - 改善タイミングで自ら手を挙げ、フラグによる切替と段階リリースで影響を抑えつつ、新規作成→管理画面の順に分離。API/内部モデルも段階的に分割し、変更影響を分離できる構造へ整理。
  - 結果として機能追加が安全になり、バグが減少しデバッグが容易になった。

## QAチーム立て直し（2025/04～2025/12）

QA体制が急激に縮小し、リリース後インシデントが多発した状況で、開発（FE/BE）が稼働の50%を割いてQAチームの立て直しに参画しました（開発所属のまま支援）。週次で発生するEpicのリリースを継続しながら、テスト運用の再構築と回帰テスト自動化を段階的に推進しました。

- **使用ツール：** GitHub / Notion / MagicPod
- **チーム構成：**
  - QA：正社員1名＋契約社員1名
  - 開発支援：フロントエンド1名（自分）／バックエンド1名
- **担当業務：**
  - 既存テストケース・MagicPodの棚卸し（稼働状況の可視化、最新／要更新／不要の分類）と運用ルール整備
  - テスト観点の整備、受け入れ基準（完了条件）・バグ運用（優先度／トリアージ）・QA/開発の役割分担の明文化
  - 週次リリースを止めない前提で、短期はマンパワーQAで対応しつつ並行して運用を立て直し
  - MagicPodによる回帰テスト自動化の推進
    - 当初：MCPサーバー＋MagicPod組み込みAIエージェントで仕様書からテスト自動生成を検討（AI性能・トークン制約により断念）
    - 代替：単体回帰テストの網羅へ方針転換し、2025年12月末に整備完了
- **成果：**
  - 定量計測は未実施だが、体感としてリリース後インシデントはほぼ発生なくなり、顧客からの問い合わせも減少

---

## 株式会社ソニックス（2018/06～2022/03）

エンジニアとしてサーバー、ネイティブアプリの開発に従事しました。

## タクシー配車アプリ 開発支援プロジェクト（2018/09～2020/03）

タクシー配車アプリの保守・継続開発を担当しました。新機能追加、バグ修正、パフォーマンス改善を通じてサービス品質の向上に貢献しました。アサイン当初はアプリ／サーバーサイドの保守対応を中心に担当し、その後体制再編により提携タクシー会社向け業務支援システムの新規開発（サーバーサイド／フロントエンド）に参画しました。

- **プロジェクトカテゴリ：** Webサービス / スマホアプリ / 業務システム / 受託開発
- **担当工程：** 要件定義 / 設計 / コーディング / テスト / 運用・保守
- **開発言語：**
  - iOS：Swift / RxSwift
  - Android：Kotlin
  - バックエンド：Ruby（Ruby on Rails） / Scala（Play Framework）
  - フロントエンド：TypeScript（React） / JavaScript
- **使用技術：** GitHub / OpenAPI（Swagger） / SQL / PostgreSQL / draw.io / Git
- **経験した職種・役割：** バックエンド / フロントエンド / アプリ開発（iOS） / アプリ開発（Android）
- **特徴：**
  - 10年以上運用されているシステムがあり、複数リポジトリ・複数DBが複雑に連携する構成
  - SQLの難読化が顕著で、調査・改修に高度な読解が必要
  - コードレビューは「自社内の相互レビュー」→「お客様側開発チームのレビュー」の2段階方式
- **担当業務：**
  - （保守・継続開発）GitHub Issueベースで週次割り当てタスクに対応
    - iOSアプリのバグ修正（UIの文言見切れ修正、地図画面のピン位置ずれ修正 等）
    - Rails製APIの改修（レスポンスデータ、バッチ処理の修正）
    - 難読化SQLの簡略化によるパフォーマンス改善
  - （新規開発）提携タクシー会社向け業務支援システムの開発
    - ScalaによるAPI開発、TypeScript（React）による業務画面開発
    - ドライバー一覧画面、ページング／ソート／削除機能の実装
- **成果：**
  - アサイン2ヶ月後には、週次タスクを水～木曜までに完了・マージし、追加タスクを継続的にアサインされる状態へ移行

## e-bike連動アプリ 開発プロジェクト（2020/03～2021/06）

BLE通信で電動バイクと連携するスマートフォンアプリの開発・運用を担当しました。他社が開発した初期バージョンを引き継ぎ、継続的な機能追加・バージョンアップを行い、各アプリストアへリリースしました。リリース後はTrelloを用いたKPTで振り返りを行い、継続的改善を推進しました。

- **プロジェクトカテゴリ：** Webサービス / スマホアプリ / 受託開発
- **担当工程：** 設計 / コーディング / テスト / 運用・保守
- **開発言語：**
  - フロントエンド：TypeScript（Vue.js）
  - Android：Kotlin
  - iOS：Swift
  - バックエンド：Kotlin（Ktor）
- **使用技術：** MySQL / Firebase / Bitrise / Figma / OpenAPI（Swagger） / draw.io / JSON
- **経験した職種・役割：** バックエンド / フロントエンド / アプリ開発（iOS） / アプリ開発（Android）
- **特徴：**
  - 他社開発の初期版を引き継ぎ、継続的にバージョンアップしてストアリリース
  - リリース後にKPT形式の振り返りを実施し、改善を継続

- **担当業務：**
  - UI設計（Figma）、フロー図作成（draw.io）
  - アプリ開発（Android/iOS：比率ほぼ半々）
  - API開発（Ktor）、DB修正、テスト、運用保守
  - マイナーアップデート途中から参画し、既存コード・資料の読み込みと言語学習によりキャッチアップして独力で開発を推進
- **開発における工夫：**
  - 実装者のスキル差やiOS/Androidの仕様差による手戻りを減らすため、「実装担当が詳細設計を作成 → テックリーダー+マネージャーがレビュー」プロセスを導入し、意図の可視化と認識齟齬の低減を実現
- **メンバーへの働きかけ：**
  - 技術・ライブラリが多岐にわたり新規参画のハードルが高かったため、増員時にオンボーディング／開発力向上／レビュー観点共有を目的としてモブプログラミングを提案・実施
- **学び：**
  - Mapbox、Charts等のライブラリ活用を通じて、アプリ開発の表現力・技術スタックの幅を拡張

### e-bike連動アプリ メール送信予約システム開発プロジェクト（2021/06～2021/08）

e-bike連動アプリ利用者に対して、利用規約改定や新サービス開始のお知らせを配信するためのメール送信予約システムを新規開発しました。既存のユーザー管理画面への機能追加として実装し、短納期（約1ヶ月半）・2名体制（FE/BE各1名）で開発を完遂しました。

- **プロジェクトカテゴリ：** Webサービス / 受託開発
- **担当工程：** 設計 / コーディング / テスト
- **開発言語：** TypeScript（Vue.js）
- **経験した職種・役割：** フロントエンド / デザイン（画面設計）
- **特徴：**
  - 既存ユーザー管理画面への機能追加として実装
  - 2名体制（フロント/バック各1名）、納期約1ヶ月半
  - 画面デザインの提供なし（ゼロベースでUI/UX設計）
  - 検索条件に合致したユーザーへ最大約20万通の一斉送信
  - アプリ設定言語に応じた多言語配信（9言語）
- **担当業務：**
  - 画面設計（Excel）
  - DB定義書の作成
  - OpenAPI（Swagger）を用いたAPI設計
  - Web画面（Vue + TypeScript）の実装、テスト
  - Vue + TypeScript経験が浅い状況から、チュートリアル等で学習しながら実装を推進
- **開発における工夫：**
  - バックエンド担当と連携し、テーブル設計からボトムアップで設計・実装を推進
  - 設計成果物を都度お客様に共有してフィードバックを取り込み、完成イメージの齟齬を抑えて手戻りを最小化
- **学び：**
  - OpenAPI（Swagger）を用いたAPI設計の実務経験を通じて、設計精度の重要性を獲得
  - Vue + TypeScript（SPA）開発の基礎的なアーキテクチャ理解を深化

### 移動で貯まるポイントをマイルに交換できるモバイルアプリ Web画面開発プロジェクト（2021/09～2021/11）

モバイルアプリから外部ブラウザで利用するマイページおよび、管理者向けアドミン画面の開発を担当しました。マイページではプロフィール編集・マイレージクラブ連携・無料/課金プラン選択・決済機能を提供し、アドミン画面ではポイントレート設定やお知らせ配信などの管理機能を提供しました。

- **プロジェクトカテゴリ**： Webサービス / 受託開発
- **担当工程**： コーディング
- **開発言語**： TypeScript (Vue.js)
- **使用技術**： Vuetify / AWS Amplify / Amazon Cognito / API Gateway / AWS Lambda / Stripe
- **経験した職種・役割**： フロントエンド
- **特徴**：
  - 認証にAWS Cognitoを採用（アドミン：Hosted UI、マイページ：非Hosted UI）
  - マイページはスマホ（特にiOS）前提のレスポンス設計、アドミンはWeb専用
  - 課金プランの決済にStripeを利用
  - 参加期間1ヶ月半で、マイページ12画面+アドミン16画面を実装
- **担当業務**：
  - マイページ画面の実装全般
  - お客様対応（進捗報告、質疑応答）を通じた不明点解消・仕様の詰め
  - アドミン画面のマネジメント業務（工数見積、タスク割り振り、進捗管理、技術サポート、PRレビュー）
- **開発における工夫**：
  - お客様側の設計が未確定な状態から開始したため、Slack（必要に応じてハドル）で密に連携し認識齟齬を抑止
  - Composition APIでメソッド共通化を進め、限られた期間で実装量を最大化
- **学び**：
  - Vue + Amplify + 非Hosted UI構成の実装事例が少ない中で、Cognito不具合も含めた調査・対応を通じて、Amplify×Cognitoの制限や挙動に関する知見を獲得

## e-bike 走行データ解析・管理サービス フルリブレースプロジェクト（2021/11～2022/03）

走行データをアカウントに紐づくクラウドにアップロードし、走行データを解析・管理するサービスのシステムをフルリブレースするプロジェクトに参画しました。画面設計およびAPI/DB設計を担当しました。

- **担当工程**： 設計
- **担当業務**：
  - 画面設計
  - API 設計
  - DB 設計

---

## 強み

- フロント、バック、アプリ（両OS）といった広い領域の開発経験
- 開発に加えて、レビュー体制改善やQA立て直しなど「プロセス改善」を推進できる

## 意欲・興味

- アーキテクチャやライブラリ選定など設計分野のスキル向上
- Go など未経験／経験浅い言語での開発領域拡大

- 継続的改善（品質・開発体制の改善）を伴うプロジェクト推進