

職務経歴

プロジェクト名
資産管理アプリ開発

プロジェクト概要
概略

WebViewを表示させているiOSアプリのフルリプレイス、およびAndroidアプリの新規開発。
社内の資産管理アプリとなり、社内備品管理や組織管理が主な機能。

期間
2021年10月～2021年12月

メンバー構成
- アプリエンジニア：1名
- バックエンドエンジニア：1名
- デザイナー：1名

自身の役割
- Flutterを用いたスマホアプリ開発
- 顧客折衝

使用技術
- Flutter

アピールポイント

RiverpodとHooksによる状態管理

StateProvider, StateNotifierProvider, Hooksを使い分け、StatefulWidgetクラスを使用しないことで最低限のリビルドのみ行われる実装に成功。

アトミックデザインを意識したUI設計

ボタンやモーダルなど、複数画面で使われるウィジェットを関数化し、引数によりプロパティを変更して表示できるように。

画面ファイルの可読性が向上し、また結果として開発スピードの向上にも繋がり納期の1週間前倒しで実装完了。

プロジェクト名
契約書管理アプリ開発

プロジェクト概要
概略

WebViewを表示させているiOSアプリのフルリプレイス、およびAndroidアプリの新規開発。
契約書を撮影またはギャラリーからサーバーへアップロードし、保存・管理するtoB向けアプリケーション。

期間

2021年6月～2021年10月

メンバー構成

- アプリエンジニア：1名
- バックエンドエンジニア：1名
- デザイナー：1名

自身の役割

- Flutter, Swift, Kotlinを用いたスマホアプリ開発
- 顧客折衝

使用技術

- Flutter
 - SQLite
- Swift
- Kotlin

アピールポイント

ネイティブコード(Swift/Kotlin)とOpenCVによるOS毎のカメラ機能開発

WeScan(Swift)ライブラリとSmartPaperScan(Kotlin)ライブラリを拡張し、カメラUIの調整、矩形の切取、回転、コントラスト調整、ソート機能などを実装。

元々のiOSアプリでは撮影した画像をメモリに保存する仕様になっていたため、Android版も同じ仕様で実装していたが、端末の一部でメモリのオーバーフローが発生する状態になり、SQLiteを活用する設計に変更。画像が複数枚表示されるUIではリサイズ済みのサムネイル画像を表示させる工夫も行い、動作のモタつき解消と安定化を図った。

共通処理はFlutterに集約

Swift, Kotlinそれぞれのライブラリで撮影・加工した画像はデータの型を統一し、MethodChannelでFlutter側に渡すことでアップロード処理はFlutter側で実行。

極力Flutterで処理させることでネイティブコードを極限まで減らし、コード量が少なくメンテナンス性の高いコードを意識。

プロジェクト名

ビジネス用途通話アプリ開発

プロジェクト概要

概略

ネット回線を利用した通話特化型のビジネス用途アプリ開発。
自社開発サービスとして1から立ち上げ。

期間

2020年4月～2021年5月

メンバー構成

- アプリエンジニア：1名
- バックエンド/インフラエンジニア：1名
- フロントエンドエンジニア：1名
- デザイナー：1名

自身の役割

- Flutter,Swiftを用いたスマホアプリ開発
- Appleストアへの申請・リリース対応
- プロジェクトリーダー

使用技術

アプリ

- Flutter
- Swift

バックエンド

- Flask

DB

- MySQL

アピールポイント

全体

プロジェクトリーダーとして社内上層部と開発チームのブリッジ役となり、スケジュール調整や仕様を固めるためのドキュメント作成などに尽力。

→齟齬が発生している場合に早急に気づいて改善に繋がった

開発においてはFlutter,Swiftどちらも実務未経験であったが、公私共にキャッチアップしリリースまで完了。

通話

WebSocket,WebRTCの技術を主に英語ドキュメントを読み漁る事でキャッチアップ。

このアプリは転送機能を有しており、callerとcalleeの通話以外に第三者の概念が存在することから、単純にWebSocketで相手の状態をするだけでは不完全であった

→WebRTCのDataChannelを活用し、callerとcalleeに第三者の状態を知らせる事に成功

またiOSの通話履歴との連携にも成功し、履歴をタップすることで当該アカウント(callerまたはcallee)への発信画面に遷移可能に

状態管理

変更があった際に画面全体ではなく変更箇所のみUIをリビルドさせるため、当時開発者からの人気が高く定評があったProviderを採用。

日本でのFlutter認知度が低い時期であり、情報収集が難しい状況であったが、動作検証するための画面を用意し、挙動を都度確認しながら実装。

プロジェクト名

宿泊顧客分析・販促システム開発

プロジェクト概要

概略

宿泊施設に宿泊した顧客のアンケート収集や分析、さらにアンケートで収集したデータを元に販促も可能にしたWebアプリ開発

期間

2020年2月～2020年11月

メンバー構成

- フロント/バックエンドエンジニア: 1名
- インフラエンジニア: 1名

自身の役割

- JavaScript/Slimを用いたフロントエンド開発(管理画面/ユーザー画面両方)
- RoRを用いたAPI開発
- DB設計
- 顧客折衝

使用技術

バックエンド

- Ruby on Rails

フロントエンド

- JavaScript
 - Ajax
- Slim
- MDBootStrap

DB

- MySQL

アピールポイント

全体

「クライアントの要望ヒアリング→工数見積→実装」まで、上司と相談しながらではあるがほぼ一人で完結。

バックエンド

インバウンド対応要望があり、リクエストヘッダーのAccept-Languageの内容によりブラウザに表示される言語を切り替える事で、多言語対応サービスに。

フロントエンド

アンケートフォームの項目が多く、submit時にバリデーションエラーが発生するとサーバーからのレスポンス待ち時間が発生し、UXが悪いという課題に対してフロントエンドのみでも全フォームにバリデーションを掛けるように対応。

販促機能としてアンケート回答者にDM送信機能を搭載し、送信先ターゲットとして「地域/国 x 年齢/性別」の選択による抽出を可能に。その際Ajaxを使うことで、現在の抽出結果件数がリアルタイムで分かるように工夫し、クライアントから絶賛頂いた。

プロジェクト名

LINE風チャット&SNS統合アプリ開発

プロジェクト概要

概略

大手運送業者のドライバー間や本社⇄ドライバー間のコミュニケーション量増加を目的としたアプリ開発。

テキストによるチャット機能に加え、掲示板やポイント交換によるインセンティブ機能を搭載

期間

2019年11月～2020年4月

メンバー構成

- PM： 2名
- バックエンドエンジニア： 2名
- フロントエンドエンジニア： 1名
- アプリエンジニア： 1名
- インフラエンジニア： 1名

自身の役割

- RoRを用いたAPI開発
- DB設計

使用技術

バックエンド

- Ruby on Rails
 - Action Cable

フロントエンド

- Vue.js
- Chart.js

DB

- MySQL

その他

- Swagger

アピールポイント

「DB/モデル設計→もう1名のバックエンドエンジニアとFB」を高速で回すことで手戻りが少なく済み、納期の2週間前倒しでの納品に成功。

具体的には今までDB/モデル/コントローラーの実装まで行ってからPRを出していたサイクルを、DB/モデル設計時点でドラフトPRを出す方針に変え、レビュー通過後に実装に着手するサイクルに変更。

またプロジェクトの途中でフロントエンドエンジニアの手が回らなくなり、2ヶ月ほどVueとChart.jsによる管理画面作成の実装にも貢献し、プロジェクトを円滑に進めることに成功。

APIの仕様はSwaggerで都度共有し、フロントエンドとアプリエンジニアにとって分かりやすいドキュメントを意識。

プロジェクト名

WordPressによるサイト構築、および改修

プロジェクト概要

概略

WordPressを利用した企業HPの新規作成や改修、静的サイト・CMS(Blue Monkey)からWordPressへの移行。
おおよそ5案件ほど担当。

期間

2019年2月～2019年10月

メンバー構成

- ディレクター： 1名
- WordPressエンジニア： 1名
- デザイナー兼コーダー： 1名

自身の役割

- PHP/WordPressによるWordPressテーマ、および管理画面作成
- コーダーが作成した静的ページをWordPressに組み込み

使用技術

- PHP
- WordPress
- JavaScript

アピールポイント

クライアントの細かい要望に対し、プラグインのカスタマイズなどを織り交ぜながら対応。

開発環境にはVagrantを使用し、PHPやWordPressのバージョンが異なる複数案件を同時実装させることで、開発スピードを向上させ納期通りの納品を達成。