

サイネージアプリ(2022-12~ 2ヶ月)

プロジェクト概要

管理画面にてプレイリストを作成し、Android端末を通してモニタに出力し続けるアプリです。主にフランチャイズ経営の飲食店や多数の店舗を抱えるショッピングモール向けに販売を目的としたサービスになります。

- **プロジェクトカテゴリ:**
 - WEBサービス, スマホアプリ, 自社プロダクト, パッケージ開発
- **担当工程:**
 - 要件定義, 設計, コーディング, テスト, 運用/保守
- **経験した職種・役割:**
 - フロントエンド, バックエンド, インフラ, アプリ開発(Android)
- **使っていた技術:**
 - PHP, CakePHP, ReactNative, Python, Apache, MySQL
- **チーム情報:**
 - 実装 x1名 (自分)
 - 端末選定 x1名

機能まとめ

- **要件定義**
 - コンテンツ内容の確認 (動画,画像,url,テキスト,,)
 - コンテンツ表示方法の確認 (表示間隔, モニタ向き,,)
 - 端末環境の確認 (端末スペック,通信環境,,)
 - 編集権限の確認 (本社と支店とそれぞれの関係)
- **技術選定**
 - 端末の選定 (候補: シングルボードコンピュータか、Android端末)
 - Android開発言語の選定 (候補: Kotlin,ReactNative,Cordova)
 - 動画編集言語 (Python)
- **DB設計**

管理者 -< 店舗 -< 端末 - プレイリスト -<

- **サーバー構築**
 - マニュアルに従ってサーバーを構築します
 - 既にサーバー構築を自動化するためのスクリプトを自作しているため、作業負担はほぼありません。

実装内容

1. 管理画面の開発(設計したDBに沿ってコンテンツ登録ができること)

- 使用した技術:
 - CakePHP, Shell
- フロントサイドのカスタマイズ
 - テンプレートにある程度用意されていますが、次の2点を追加しました。
 - 直感的に操作できるリスト形式の入れ替え機能

- 動画編集処理の進捗状況が可視化されるようプログレスバー

2. プレイリスト拡張子の利用(m3u8)

- 使用した技術:
 - ReactNative, FFmpeg
- 実装目的:
 - ライブ配信で使われている拡張子(m3u8)を利用し、クライアント側の読み込み速度を安定させること。
- 成果
 - 大きい動画ファイルを定期的に読み込む時と比べると、安定して読み込むことができます。
- 課題
 - 一見問題ないように見えたが、やはり負担が大きいです。CDNを利用していないためサーバー側の負担も計り知れません。

3. プレイリストのオフライン対応(プレイリストデータ・データ構造を外部ストレージに保存する)

- 使用した技術:
 - ReactNative, react-native-fs
- 実装目的:
 - クライアントのさまざまな環境を考慮し、通信環境に依存しない機能が必要でした。
- 課題・問題点
 - 基本的なコンテンツ表示に使われる`import`に変数が利用できない事実に躓きました。バイナリ型としてローカル保存することも考えましたが、コンテンツデータを外部ストレージに保存することで実現しました。

4. プレイリスト統合ツール開発(プレイリスト内のすべてのコンテンツを1つの動画に変換する)

- 使用した技術:
 - Python, MoviePy
- 実装目的:
 - 多少なりともクライアント側の負荷を減らすため。
- 成果
 - クライアント側の負担を大きく減らすことができました。責任を分散することでテストも容易になりました。(動画が構成されていない時はPython側、想定通りに表示されないときはReact側)

成果

- **ReactNative,Androidの開発経験**
 - 画像はimportするもので、importは変数を利用できないという固定概念が変わりました。Android端末への苦手意識もなくなり、どんどん活用していきたいと考えています。
- **Pythonの開発経験**
 - 端末のスペックによるものだから暗転状態が発生するのは仕方がないと諦めず、サーバー側での動画編集処理を提案し、実現することができました。このような技術的な課題を解決するには、フルスタックの経験と技術が重要だと感じました。

教育支援システム(2022-06~ 4ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 業務システム, 受託開発)
- 担当工程: (設計, コーディング, テスト, 運用/保守)
- 経験した職種・役割: (フロントエンド, バックエンド, インフラ)
- 使っていた技術: (PHP, React, 自然言語処理, AWS, MySQL)

課題に対する生徒の意見を集計することがメインのシステムになります。

チームの構成と立ち位置

- CSSとHTMLの担当 x2名
- CSS以外の担当 x1名 (自分)

背景

当初、別の企業が開発していたシステムを改修する依頼でしたが、改修規模が大きく、扱い慣れていない技術を使用していたため、システムを作り直すことにしました。DB構造から見直しながらも、大きく仕様は変えないことを念頭に置いて開発を進めました。

開発内容・課題

1. 形態素解析 MeCabを利用し出現頻度の高いワードを算出しました。また、管理画面上からMeCabの辞書を更新することで、NGワードや造語にも対応しました。
2. 多種多様なポップアップ CakePHPでは大変な作業でしたが、フロント言語をReactにすることで難なく実装できました。(編集するとき、戻る時のアラート、ファイルをアップロードするとき)

内覧受付システム(2021-10~ 2ヶ月)

- 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, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, WordPress, Apache, MySQL)

スムーズな内覧受付機能を重視したホームページです。

チームの構成と立ち位置

- デザイナー x1名
- フロントエンド x2名
- テーマ制作、API開発 x1名 (自分)

開発内容・課題

1. 多言語化 1つの物件に対する項目量が多いため、日本語の枠の隣に英語用の枠を用意するのではなく、英語用のサイトとして、サイトごと複製することが最善の策だと判断しました。今思えば、DeepLで自動翻訳する策も視野に入れるべきだったかと悔いが残ります。

成果

REST APIを使う際のフロントエンド部署との連携の難しさと、実装した時の使いやすさを再認識しました。

そこで、まずはSwaggerを利用して可読性の高いAPI設計書を作成することと、自分自身でReactを学んで、連携まで実現することを目指すことに決めました。

販促システム(2021-07~ 2ヶ月)

- 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, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, SMS, OCR, Apache, ロードバランサ, MySQL)

抽選を行いキャッシュバックを約束することで、購買意欲を高めることを目的とするシステムです。

チームの構成と立ち位置

- CSSとHTMLの担当（外部）
- CSS以外の担当（自分）

開発内容・課題

1. 外部フロントエンジニアとの連携 自社のフロントエンドエンジニアと異なる書き方や仕様に戸惑いましたが、いい経験になりました。
2. 不正抽選の防止 不正抽選を防ぐための認証および景品の送信のためにSMSを利用しました。選定から行い、料金仕様とAPIの豊富さからTwilioを選びました。
3. 購入したかどうかを確認する作業の負担 LINEのOCRを導入することで、少しでもおかしい箇所を目視するだけで済む作業に簡略化しました
4. サーバーへの負荷 全国規模で一斉に応募されることが想定されるため、ロードバランサを使い負荷を分散しました。

学内システム(2021-03~ 4ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 業務システム, 受託開発)
- 担当工程: (設計, コーディング, テスト, 運用/保守)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, LDAPS, Apache, MySQL)

教職員と学生(約10,000人)へページを共有するシステムです。

チームの構成と立ち位置

- CSSとHTMLの担当 x2名
- CSS以外の担当 x1名（自分）

開発内容・課題

1. 権限 なにより権限が重視されるため、いつも以上に緊張感を持って実装しました。
 - IP制限
 - 操作ログ
 - 部門ごとの閲覧権限
 - ファイルごとのアクセス権限
 - 教職員が登録したページの承認処理等
 - 承認前後のページアーカイブ
2. 複数システムにわたるアカウント情報を一元管理する アカウント管理をしているLDAPサーバーと連携しました。LDAPの挙動を把握するため、別途LDAPS環境を構築しました。

見積システム(2020-06～ 3ヶ月)

- プロジェクトカテゴリ: (WEBサービス, 業務システム, 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)
- 経験した職種・役割: (バックエンド, インフラ)
- 使っていた技術: (PHP, Backlog, Apache, MySQL)

業務の効率化、見積案件の整理、それらに紐づく実働工数の可視化を図るシステムです。入社して間もない頃に開発したシステムですが、多くのスタッフに利用されておりいいものを作れたと実感しています。

チームの構成と立ち位置

- デザイナー x1名
- CSSとHTMLの担当 x2名
- CSS以外の担当 x1名 (自分)

開発内容・課題

1. 見積書・請求書の自動生成 登録した担当者、見積もり内容と金額から、担当者の判子が押された状態の見積書や請求書を生成します。
2. 工数管理 タスク管理に利用していたBacklogと連携させることで、担当者ごとの進捗具合や案件ごとの実働工数を算出しました。
3. サイト管理 システム上でサイトを登録し、SSLの有効期限を監視します。