

# 職 務 経 歴 書

2021年08月01日現在

氏名 宋 竹梅

## ■職務要約

2016年モンスターラボ成都支社へ入社しフロントエンド開発を担当。  
2017年5月からはフロントエンドリーダーとして、フロントエンドの開発チームの品質向上に貢献。  
2018年12月モンスターラボ日本会社へ転籍後は、多国籍開発チームのフロントエンドテックリードとして、プロジェクト単体だけではなく、会社全体の開発品質向上に貢献する。

## ■得意分野

### 技術・知識

- Web develop: ★★★★★ Typescript / ECMAScript / HTML5&CSS3
- Framework&Library: ★★★★★ React.js(Gatsby.js)+Redux(Rematch)+immutable.js+react-route
- Build tools:★★★★★ Webpack / Parcel / Rollup / Gulp
- Automated test: ★★★★★ Jest / React Testing Library / Nightwatch
- Back-end & Infra: ★★★★★ Node.js / AWS / Heroku / Firebase
- Others: ★★☆☆☆ Git flow / Docker / Linux basic commands / Flutter
- English: ★★★★★
- Japanese: ★★★★★

### 幅広いフェーズ経験

- プロジェクトのタイプに応じて適切なアーキテクチャを選択
- フロントエンドチームでの複数人での作業経験・デザインチームとバックエンドチームとの連携経験
- 技術指導および、人材の育成

## ■職務経歴詳細

2018年12月～現在 株式会社モンスターラボ

事業内容:フロントエンドテックリードとして、アーキテクチャーとコーディング

| 期間 | プロジェクト内容                    |
|----|-----------------------------|
|    | コーヒーラベルのカスタマイズWeb App PC・SP |

|  |   |
|--|---|
| <p>2021年04月<br/>～<br/>2021年07月<br/>(4ヶ月)</p> | <p><b>[概要]</b></p> <p>コーヒーショップのユーザーが注文する様子を再現したウェブアプリで、ユーザーはテキストを入力したり、画像を選択したりして、コーヒーのラベルをカスタマイズすることができます。<br/>React hooksのbest practiceを使用。</p> <p><b>[使用ツール]</b></p> <ul style="list-style-type: none"> <li>● Gatsby.js + redux(rematch) + immer..js + typescript</li> </ul> <p><b>[担当フェーズ]</b></p> <ul style="list-style-type: none"> <li>● フロントエンドのアーキテクチャー</li> <li>● コーディング・テスト</li> </ul> <p><b>[開発規模]</b></p> <ul style="list-style-type: none"> <li>● フロントエンド (2名)</li> <li>● バックエンド (2名)</li> <li>● テスト (2名)</li> <li>● プロジェクト全体 ( 10名)</li> </ul> <p>immer.jsを使用しており、Immutable dataを導入しています。ページに深いコンポーネントツリーがある場合、再レンダリングの判断を正確に行うことができ、子コンポーネントへの変更がトップダウンですべてのコンポーネントの再レンダリングを引き起こすことはありません。<br/>画像のリンクをクリックして画像をダウンロードするためのnpm packageは、以前のプロジェクトで使用したものです。<br/>このプロジェクトでは、デザインチームと密接に連携し、多くのデザイン変更がありました。コンポーネントのステートレスな処理と、state treeをフラットにより、頻繁な設計変更にも容易に対応できました。</p> |
| <p>2020年12月<br/>～<br/>2021年02月<br/>(2ヶ月)</p> | <p><b>ファンクラブWeb App (Admin+Client)    PC・SP</b></p> <p><b>[概要]</b></p> <p>ファンクラブの新規プロジェクトです。React hooksのbest practiceを使用。</p> <p><b>[使用ツール]</b></p> <ul style="list-style-type: none"> <li>● Admin側 : React.js + redux(redux-saga) + react-router + antd</li> <li>● Client側 : React.js + redux(redux-saga) + react-router + less</li> </ul> <p><b>[担当フェーズ]</b></p> <ul style="list-style-type: none"> <li>● フロントエンドのアーキテクチャー</li> <li>● Sprintのタスクを分割</li> <li>● コーディング・テスト</li> <li>● code review</li> </ul>   |

|  |   |
|--|---|
|  | <p><b>【開発規模】</b></p> <ul style="list-style-type: none"> <li>● フロントエンド（4名）</li> <li>● Appエンド（2名）</li> <li>● バックエンド（3名）</li> <li>● テスト（3名）</li> <li>● プロジェクト全体（15名）</li> </ul> <p>使用しているイメージダウンロードコンポーネントで、イメージをダウンロードできないという問題が発生しました。これは、画像がブラウザにキャッシュされていて、正常にダウンロードできなかったためです。解決した後、image urlで画像をダウンロードする方法をnpmパッケージを作りました。</p> <p>管理画面の記事プレビュー機能は、iframeで表示されます。プレビュー機能は、コンポーネントではなく、ページリンクを呼び出すことで最も簡単に実装できます。</p>  |
| <p>2020年06月<br/>～<br/>2020年08月<br/>(3ヶ月)</p> | <p><b>ライブ配信用のメッセージ機能 PC・SP</b></p> <hr/> <p><b>【概要】</b></p> <p>Serverlessを使用して新規のプロジェクト。各ライブ配信に対応した部屋があり、その部屋にアクセスしてメッセージを投稿することができます。ライブ配信でメッセージを投稿する機能は、10万人のユーザーがオンラインでメッセージを投稿するという性能テストに合格することが求められます。</p> <p>React hooksのbest practiceを使用。</p> <p><b>【担当フェーズ】</b></p> <ul style="list-style-type: none"> <li>● 技術調査</li> <li>● コーディング・テスト</li> </ul> <p><b>【開発規模】</b></p> <ul style="list-style-type: none"> <li>● フロントエンド（1名）</li> <li>● テスト（1名）</li> <li>● プロジェクト全体（4名）</li> </ul> <p>今回のプロジェクトで最も問題となったのは、いかにして性能試験をクリアするかということでした。当初はFirebaseを使用していましたが、ユーザー数が2万人に達したときに、データを取得できないというパフォーマンス上の問題が発生しました。AWS Amplifyに移行した後、1000本以上の書き込みが制限される問題が発生しました。ビジネスモデルを単純化して問題点を探てみると、コメントデータがルームIDに関連付けられていない場合、書き込み制限がないことがわかりました。このパフォーマンスの問題は、コメントからルームIDのキーを削除することで解決しました。</p> |
|  | <p><b>まもりあいJapan管理画面の構築 PC</b></p>  |

|                                    |  |
|------------------------------------|--|
| 2020年04月<br>～<br>2021年05月<br>(二週間) | <p>[概要]</p> <p>接触確認アプリ「まもりあいJapan」プロジェクトの管理画面の新規プロジェクト。React hooksのbest practiceを使用。 管理画面を素早く構築・公開することができました。<br/> <a href="https://github.com/mamori-i-japan/mamori-i-japan-admin-panel">https://github.com/mamori-i-japan/mamori-i-japan-admin-panel</a></p> <p>[使用ツール]</p> <ul style="list-style-type: none"> <li>● React.js + redux(redux-saga) + react-router + typescript + antd</li> </ul> <p>[担当フェーズ]</p> <ul style="list-style-type: none"> <li>● フロントエンドのアーキテクチャー</li> <li>● Sprintのタスクを分割</li> <li>● コーディング・テスト</li> <li>● code review</li> </ul> |
|------------------------------------|--|

## ■自己学習について

### フロントエンドコンポーネントの標準化

- OSSのソースコードの研究
- プロジェクト間の再利用可能なnpm packageの構築
- 自分のコンポーネントライブラリを完成させたい

### フロントエンドのパフォーマンスの最適化

- Immerを使用して、レンダリング効率を向上させる

## ■自己PR

### プログラミングの基礎がしっかりしている

- javascriptデザインモデルの深い理解
- プログラミングパラダイムの深い理解 (FP / OOP)
- React Hooks・Customize Hooks・High Order Component

### SPAの開発経験多数

2017年からReact.jsでプロジェクトを開発している。React.jsでの開発が得意

### 変化を受け入れる

- 勉強会等に参加したり、気になった新技術を積極的に勉強している
- 技術を共有