kintoneカスタマイズ事例集

JavaScriptによるkintoneカスタマイズの考え方と事例集

株式会社ミナテック

はじめに

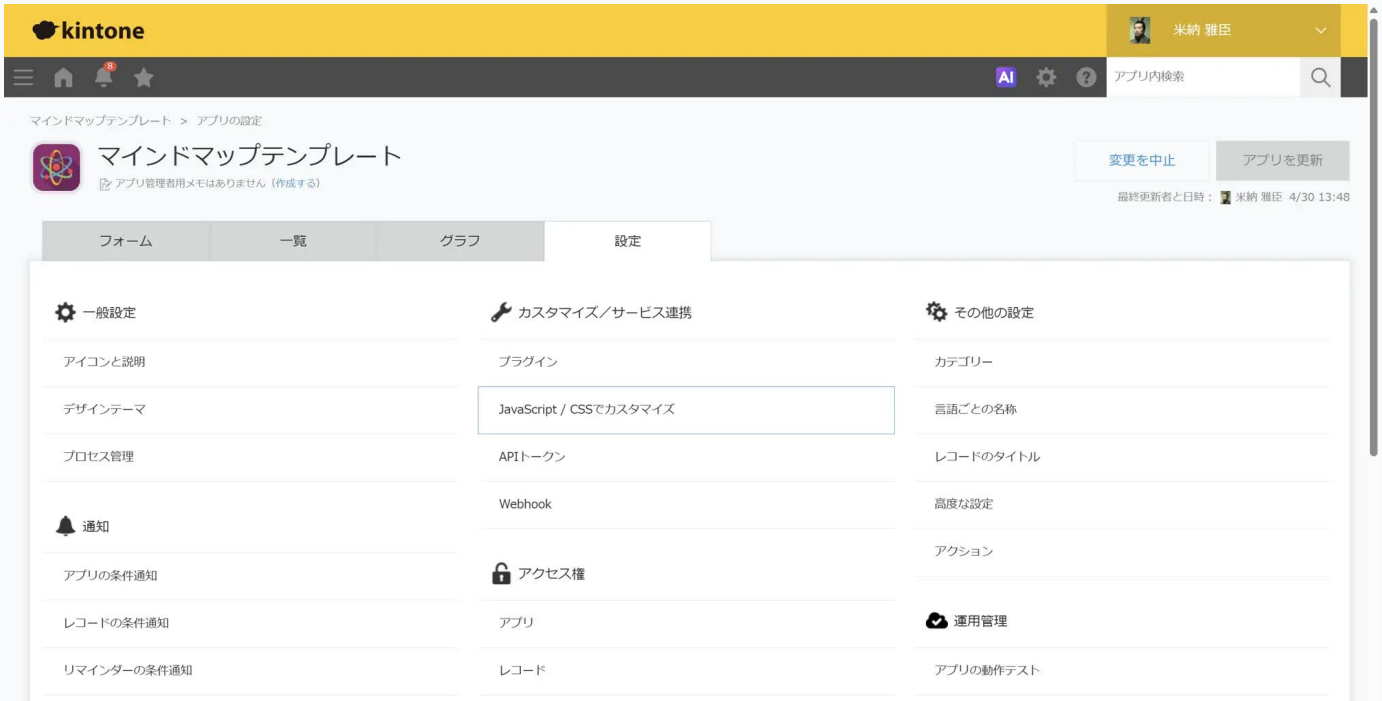
kintoneのカスタマイズ（ここではアプリに対するカスタマイズに限定します！）は、アプリの設定にJavaScriptファイルを組み込むことで実現できます。

2つの実装方法について

カスタマイズの実装方法は2通りあり、それぞれに特徴があります。

方法1：ファイルアップロード方式

[アプリ] → [設定] → [JavaScript / CSSでカスタマイズ] から、作成したJavaScriptファイルやCSSファイルをアップロードする方法です。



メリット：

- Visual Studioなどのエディタで編集するため、長めのコードでも編集しやすい
- コード補完や検索機能など、開発ツールの機能を活用できる

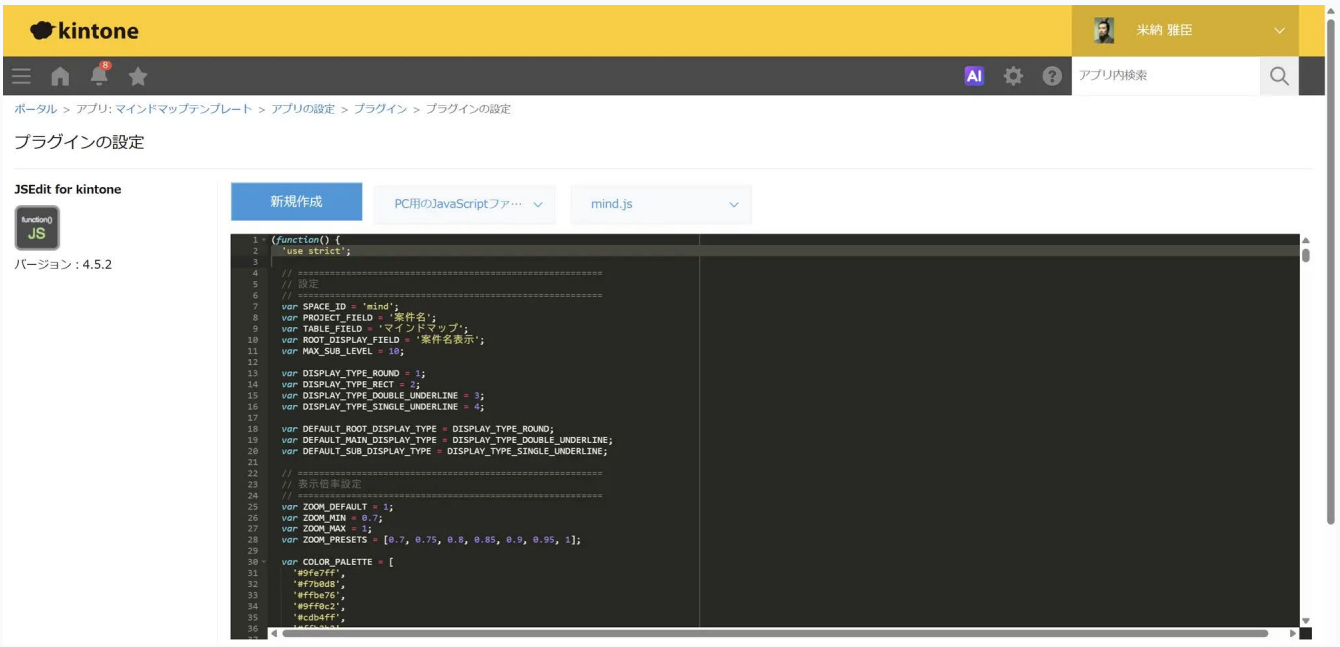
デメリット：

- ちょっとした変更でも「ファイル編集 → 保存 → アップロード」という手順が必要

実装方法解説

方法2：JSEdit for kintone プラグイン方式

JSEdit for kintone というプラグインをアプリに追加し、そのプラグイン内でコードを直接入力する方法です。（この方法でも最終的に [JavaScript / CSSでカスタマイズ] にコードが登録されます）



メリット：

- プラグイン上で直接編集して保存すれば、すぐに適用される
- 手軽に試行錯誤できる

デメリット：

- プラグイン設定画面での編集になるため、長文のコードは視認性が悪い
- 検索機能がないため、編集箇所を目視で探す必要がある

どちらを選ぶべきか？

重めのカスタマイズなら **方法1（ファイルアップロード方式）** がおすすめ

軽めのカスタマイズなら **方法2（JSEdit for kintone プラグイン方式）** がおすすめ

カスタマイズを成功させるポイント

💡 AIを活用しましょう

ここまで読んで普段コーディングをしていない方は難しく感じたかもしれません。ですが、コーディング自体はAIにさせれば良いので、その心配は無用です。

重要なのは、以下の3点です

1. **kintoneの仕様を理解すること**
2. **kintoneカスタマイズのパターンを把握すること**
3. **実現したいことを明確にすること**

これらを理解し、AIに適切なプロンプトを渡せば完成度の高いコードが生成されます。そして、それこそが成功への近道です。

やってはいけないこと

以下のアプローチは、直接的にkintoneカスタマイズの近道にはなりません。

- kintoneカスタマイズを始めるために、JavaScriptの勉強から入る
- 他の人のカスタマイズ事例のコードを解析して、自分でコードを作ろうとする

これらの行為は、kintoneカスタマイズに十分習熟してから、時間に余裕があるときに取り組んでください。

💡 あなたがやるべき事は

「**AIを使いこなしkintoneカスタマイズを成功させる**」ことです。

1. kintoneカスタマイズの4つのパターン

kintoneのカスタマイズは、体系的に次の4つのパターンに分類できます。

① フィールド操作パターン

アプリ内のフィールドに対して操作を行うカスタマイズです。

具体例:

ルックアップで他アプリの内容を取得したフィールドは編集できません。例えば見積アプリで商品登録マスターから商品情報をルックアップで取得した場合、見積アプリ上で商品情報が編集できなくなります。

JavaScriptカスタマイズを使えば、このフィールドに対して入力を許可したり、フィールドの表示/非表示を制御できます。以下画像参照。ただし、これは本記事の本題ではないため、詳細は割愛します。


```
(() => {
  'use strict';
    const events = [
    'app.record.create.show',
    'app.record.edit.show',
    'app.record.create.change.商品',
    'app.record.edit.change.商品'
    ];

    kintone.events.on(events, event => {
      const record = event.record;

      for (let i = 0; i < record.商品.value.length; i++) {
        record.商品.value[i].value['商品名'].disabled = false;
        record.商品.value[i].value['型式・詳細'].disabled = false;
        record.商品.value[i].value['単位'].disabled = false;
        record.商品.value[i].value['単価自動'].disabled = false;
      }

      return event;

  });
})();
```

② スペース要素を活用した独自UI表示パターン

レコード詳細画面やレコード編集画面において、意図した場所に独自のUIを差し込むカスタマイズです。

スペース要素とは:

フォーム設定で使用できる特殊な要素で、通常はフィールド間の間隔調整に使用します。このスペースには「要素ID」を設定でき、この要素IDを目印にして、カスタマイズしたUIをそこに表示できます。

🎯 **これがわかると、kintoneでなんでも作れる世界が広がります!**

具体例: kintoneの標準機能「関連レコード一覧」は文字情報ベースの表示しかできません。しかし、スペース要素を配置し、そこにチャートなどのグラフィカルなUIをカスタマイズで表示させることができます。

スペースを配置すれば、そこにどんなUIでも表示できるのです。
これが重要なポイントです。

実装例：スペース要素を使ったData Grid表示

ステップ1：スペース要素の配置

チェックリスト判断のチェックボックスフィールドの下に、スペース要素を配置します。

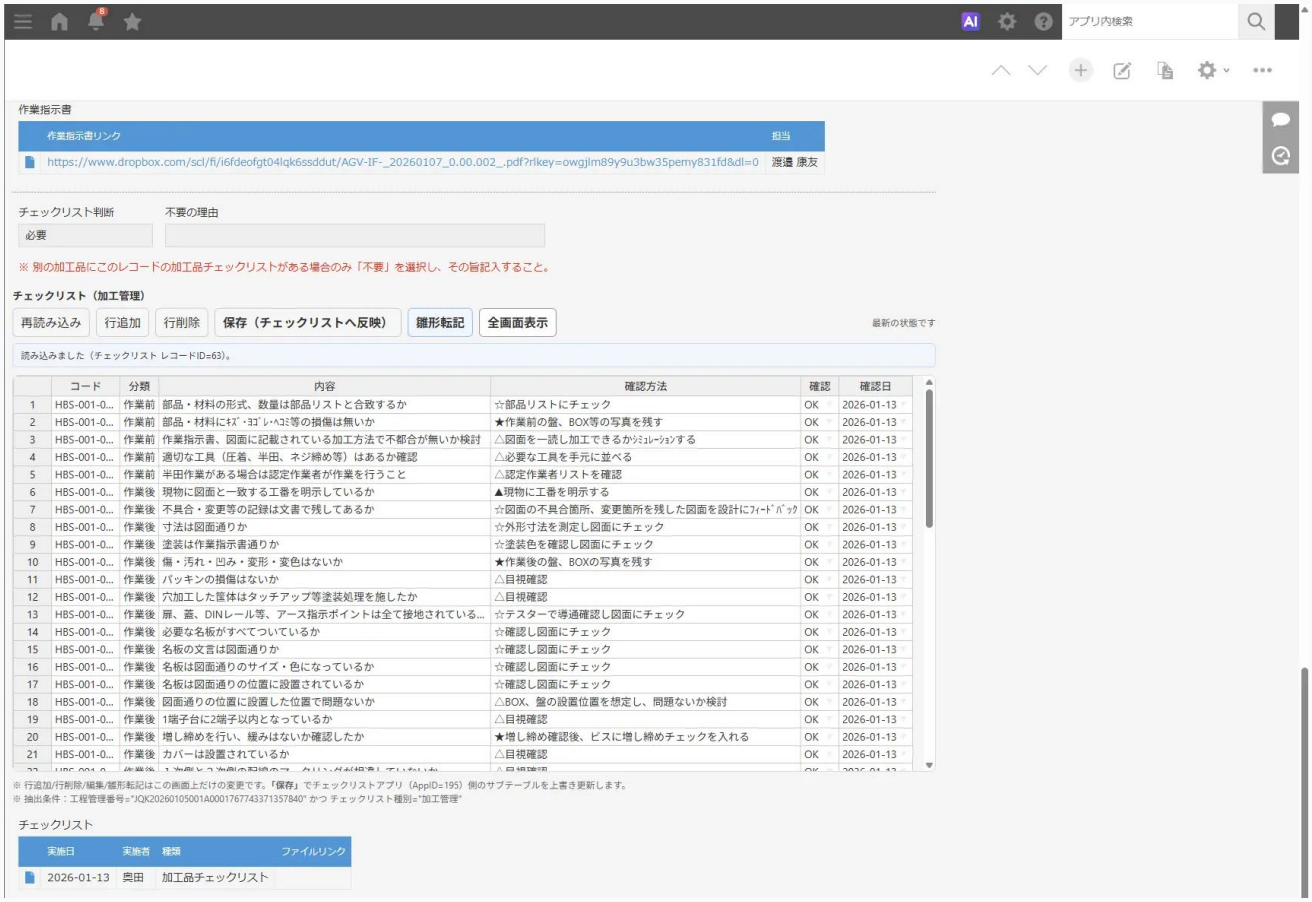
ステップ2：要素IDの設定

配置したスペースの設定画面で、要素ID（例：check_list ）を設定します。この要素IDが、カスタマイズUIを表示する際の目印となります。

実装例：スペース要素を使ったData Grid表示

ステップ3：カスタマイズの適用

JavaScriptで要素ID check_list を指定してData Gridを表示するコードを実装すると、画像のようなスプレッドシート風のUIが表示されます。



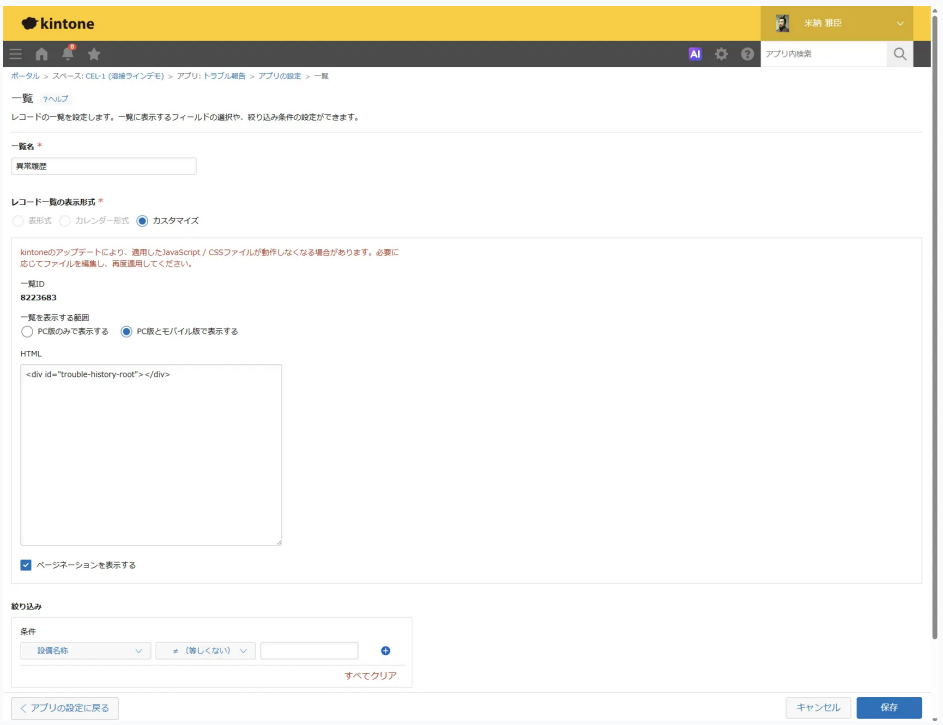
✨ ポイント

スペース要素に要素IDを設定することで、その場所を「アンカー」として、任意のカスタマイズUIを配置できます。この仕組みを理解すれば、kintoneの標準UIでは実現できない独自の表示や操作が可能になります。

③ レコード一覧画面の独自UI表示パターン

レコード一覧画面に独自のUIを表示するカスタマイズです。

アプリの一覧設定には「表形式」「カレンダー形式」「カスタマイズ」という形式があります。この「カスタマイズ」を選択することで、JavaScriptで作成した独自UIを表示できます。



活用シーン:

- 標準UIでは物足りない場合
- 他のアプリの内容も一覧に表示したい場合
- レコードを集約して独自の形式で表示したい場合

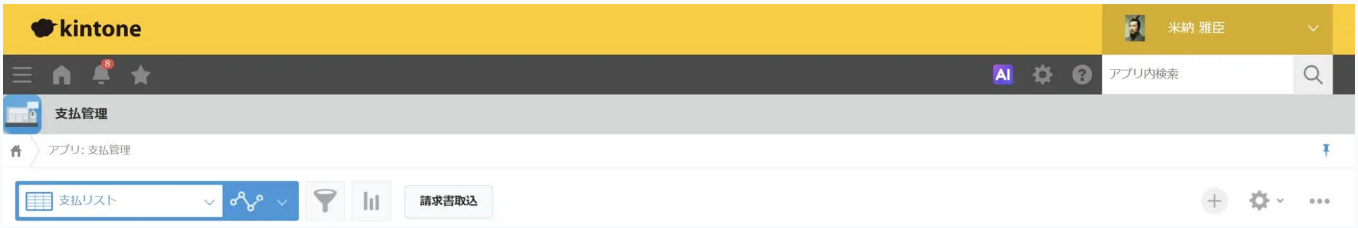
📌 重要

kintoneのレコード情報を表示する画面は、
・レコード詳細画面
・レコード編集画面
・レコード一覧画面
この３つの画面が基本になります。

④ 標準UI内へのボタン追加パターン

kintoneの標準UIの決まった場所にボタンなどのUIを追加するカスタマイズです。

例えば、レコード一覧画面の標準UI内にボタンを追加できます。スペース要素を目印にUIを表示するのと同様に、場所を指定できればどこにでも追加可能です。



活用シーン:

- 一括処理を実行するボタンを追加したい場合
- 外部システムとの連携ボタンを配置したい場合
- カスタム機能へのショートカットを作りたい場合

🎓 まとめ

ここまで紹介した4つのパターンを理解すれば、kintoneカスタマイズの大部分をカバーできます。特に②スペース要素の活用と③一覧画面のカスタマイズを習得することで、kintoneの可能性が大きく広がります。

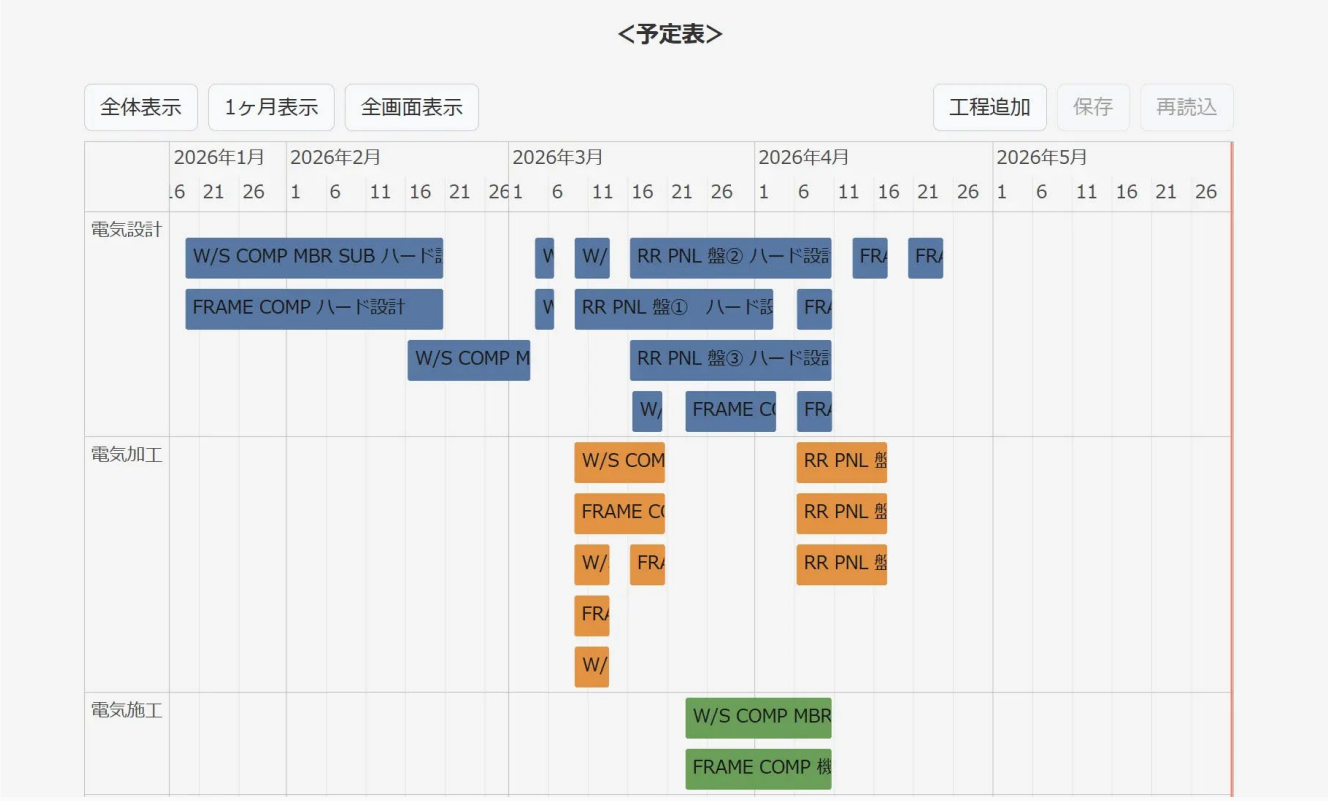
2. 具体的なカスタマイズ事例

ここからは、当社で実際に運用しているkintoneアプリに実装したカスタマイズ例を紹介します。

事例1：ガントチャート表示

🎯 実現したこと

プロジェクト管理アプリのレコード詳細画面に、工程管理アプリのデータを使ったガントチャートを表示し、編集も可能にしました。



背景と課題

プロジェクト管理アプリに関連レコード一覧で、設計期間の開始予定日、終了予定日を工程管理アプリから取得し表示していました。

[設計計画]

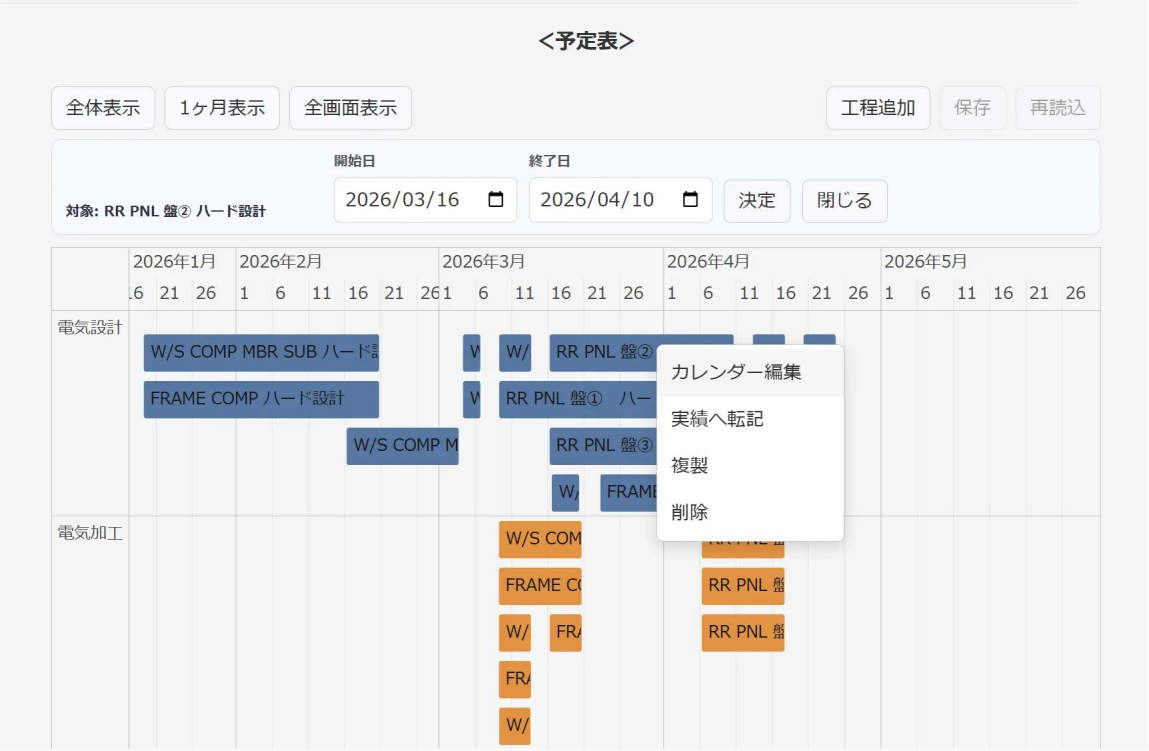
内容	担当	開始予定日時	終了予定日時	論理工数
Aeon2 ICN1 PLCプログラム設計	米納 雅臣	2025-12-29 21:33	2026-01-31 17:00	25 工数
Aeon2 ICN2 PLCプログラム設計	米納 雅臣	2026-02-02 8:00	2026-03-06 17:00	25 工数

これでは、アイテムが多くなった時の期間の重なりや、各アイテムのフェーズがわかりずらかったので、ガントチャートUIを作ってレコード詳細画面に表示するようにしました。

事例1：実装内容

詳細仕様

1. プロジェクト管理アプリの「管理番号」フィールドを抽出キーとして使用
2. 工程管理アプリから条件が一致するレコードの内容、担当、開始予定日時、終了予定日時を取得
3. 取得したデータを指定したスペース要素IDにガントチャートUIを表示
4. ガントチャートは全体表示、1ヶ月表示を切り替えるボタンを実装
5. 工程追加ボタンにより、新規のアイテムをこの画面から追加できるように実装
6. 全画面表示ボタンにより、ガントチャートを全画面表示できる機能を実装
7. ガントチャートを左クリックすると、工程管理アプリのレコード詳細画面が開く
8. ガントチャートを右クリックすると、チャート操作ウインドウを表示
9. カレンダー編集機能により、アイテムの開始予定日時、終了予定日時を編集できるようにした



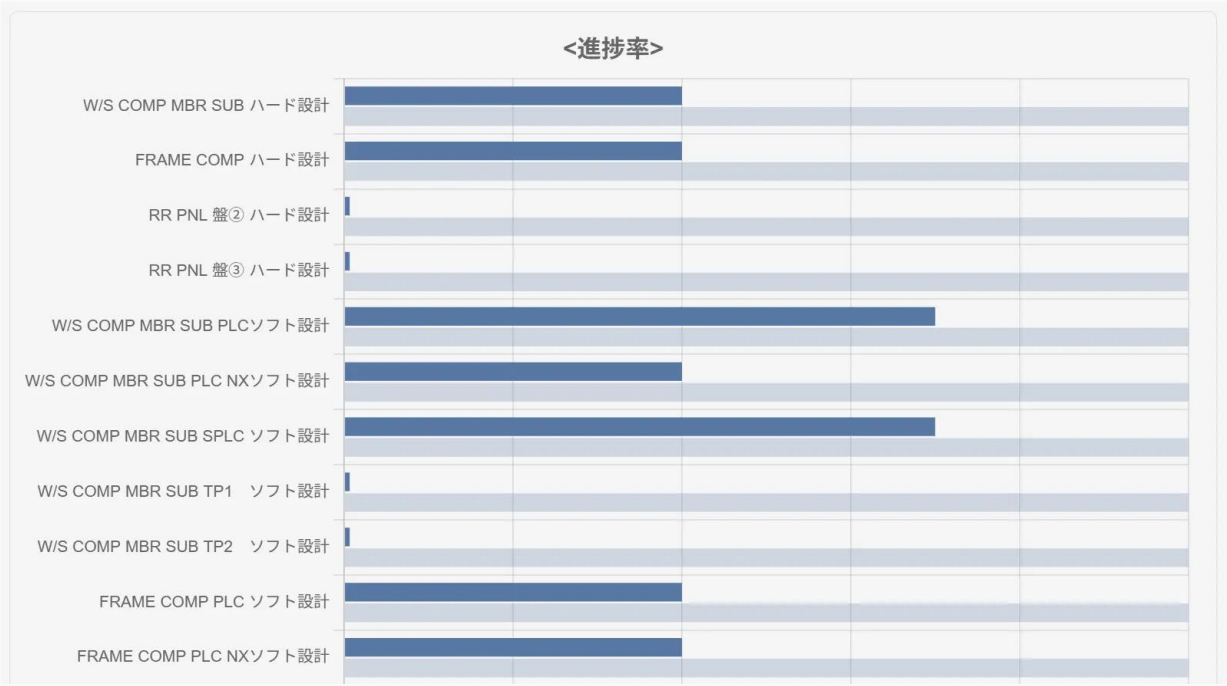
💡 実装のポイント

このようなUIを実装する際は、全画面表示機能を追加することをおすすめします。アプリ内の表示領域は限られているため、情報量が多くなると使いづらさを感じるからです。

事例2：進捗率グラフ表示

🎯 実現したこと

プロジェクト管理アプリのレコード詳細画面に、設計管理アプリのデータと、工程管理アプリのデータを使った進捗率と目標進捗率をグラフ表示する機能を実装しました。グラフをクリックすると対象のレコード編集画面に飛ぶようにすることで、プロジェクト管理アプリから設計管理の進捗率を編集するという動線を作りました。



背景と課題

プロジェクト管理アプリに関連レコード一覧で、設計管理アプリから、内容、担当、進捗ステータス、進捗率を取得し表示していました。

[設計進捗]

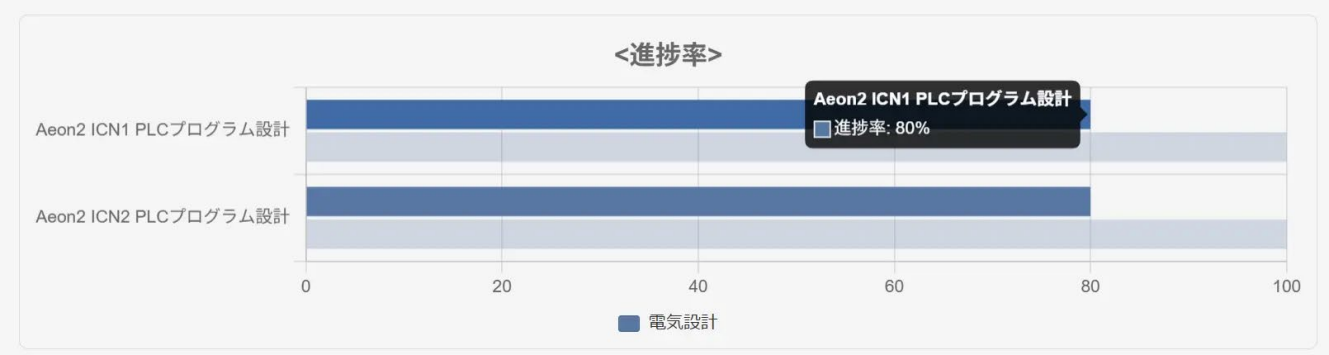
内容	担当	進捗ステータス	進捗
Aeon2 ICN1 PLCプログラム設計	米納 雅臣	受取資料確認　資料作成完了　OUTPUT作成　レビュー完了　プレリリース	80 %
Aeon2 ICN2 PLCプログラム設計	米納 雅臣	受取資料確認　資料作成完了　OUTPUT作成　レビュー完了　プレリリース	80 %

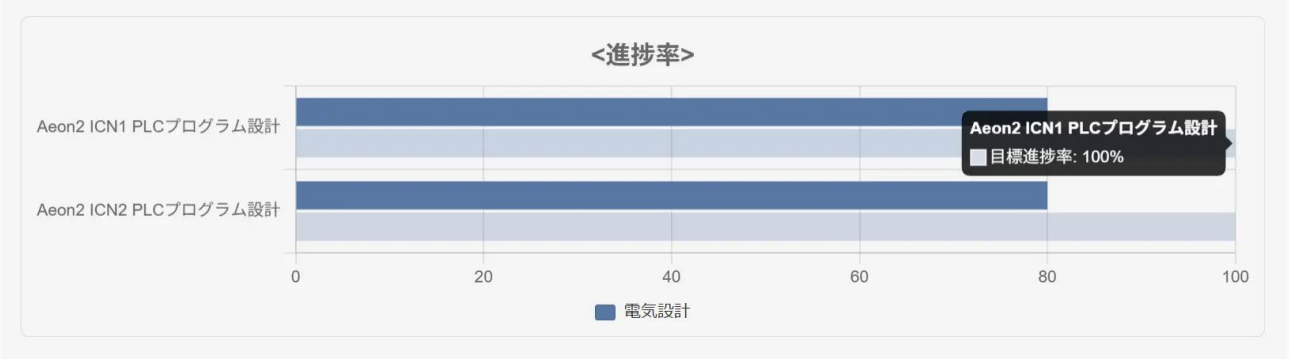
これでも、進捗率は分かりますが、現時点で進捗が進んでいるのか、遅れているのか、分かりずらさを解消する為、進捗率グラフを表示するようにしました。

事例2：実装内容

詳細仕様

1. プロジェクト管理アプリの「管理番号」フィールドを抽出キーとして使用
2. 設計管理アプリから条件が一致するレコードの内容、担当、進捗率を取得
3. 設計アイテムと一致する工程管理アプリのレコードの開始予定日時、終了予定日時を取得
4. 取得したデータを指定したスペース要素IDに進捗グラフUIを表示
5. 設計管理アプリのデータはチャート上部の進捗率として使用
6. 工程管理アプリのデータはチャート下部の目標進捗率として使用
7. チャートを左クリックすると、設計管理アプリのレコード詳細画面が開く
8. チャートにカーソルを合わせると、進捗率数値を表示





💡 実装のポイント

2つのアプリから必要な情報を取得し、かつ現在日時から目標進捗率を表示することで、進捗度合の視認性を向上させることに成功しました。

事例3：レコード入力アドバイス表示

🎯 実現したこと

設計管理アプリのレコード入力状態から、入力アドバイスを表示するようにしました。アドバイスの内容に応じて、選択した画像が表示されるようにし、正常、警告、異常の状態をユーモラスに伝えられるようにしました。

背景と課題

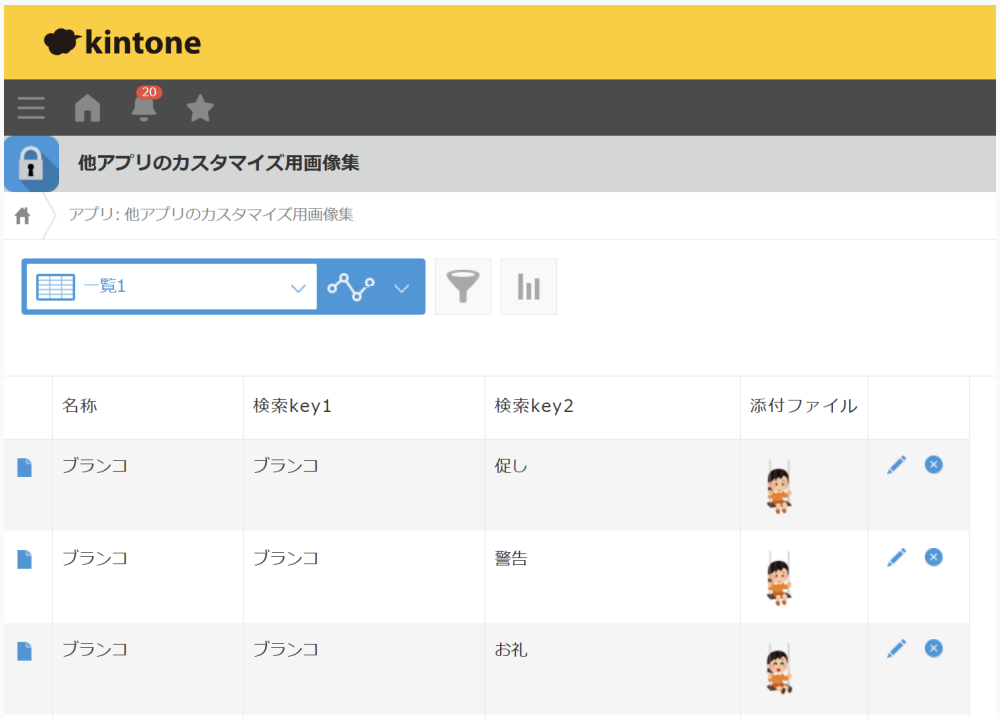
複雑なkintoneアプリは入力する人の理解度により、間違った内容を入力されたり、入力して欲しいフィールドに入力してもらえなかったり、その都度指摘し修正してもらう必要があり困っていました。
ある程度自動で指摘する機能をと思いこの機能を実装しました。

事例3：実装内容

詳細仕様

1. 設計管理アプリの入力状態の判定ロジックと表示するアドバイスコメントの関係性を纏めました
2. 入力状態の状態遷移として、OK、FAULT、WARNINGを用意
3. OK、FAULT、WARNINGの時に表示する画像を登録する為の別アプリを作成
4. 判定ロジックによるアドバイスコメントと画像を指定したスペース要素IDに表示するUIを実装
5. 画像は複数登録してあれば、ドロップダウンでユーザーが切り替えられるようにしました

別アプリで画像を表示する為のライブラリ的アプリを作成



💡 実装のポイント

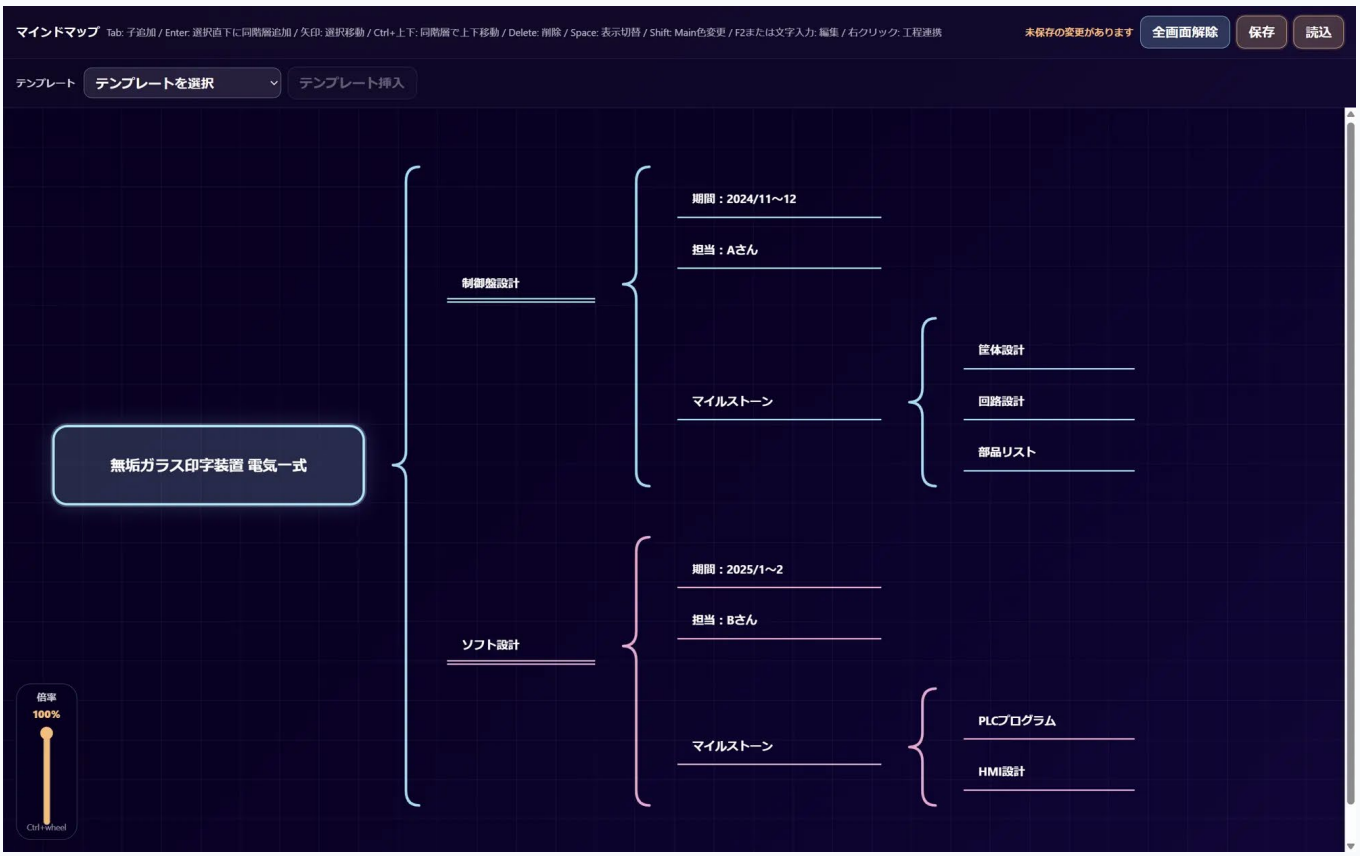
入力状態を判定するロジックと、それに応じたアドバイスコメントをAIに表などで説明すると、正確な判定ロジックを構築しやすいです。

画像ファイルを表示させるカスタマイズには、画像ファイルのライブラリ的アプリを作り、そこに添付ファイルで画像を格納することで、引っ張ってくることが出来ます。画像の検索性を上げる為に名称や分類などのフィールドを設定することも忘れずに！

事例4：マインドマップ入力機能

🎯 実現したこと

設計管理アプリのレコード詳細画面に、マインドマップUIによる
プロジェクト概要入力機能を実装しました。



背景と課題

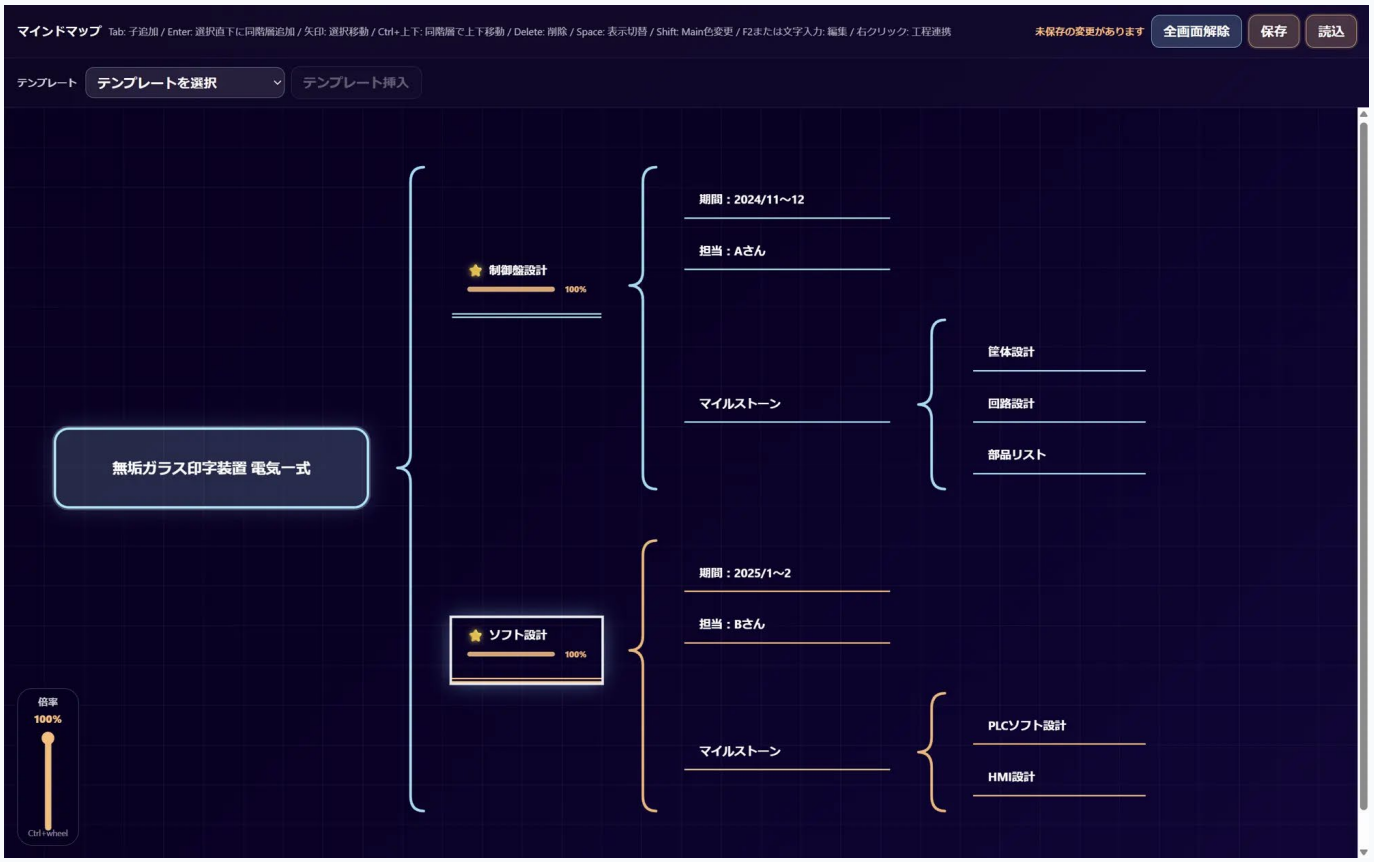
プロジェクト管理アプリにプロジェクトの概要を文字列（複数行）フィールドに入力するようにしていましたが、入力する人の文章作成能力により、詳しく書ける人もいれば、そうでない人もいて、管理者が一見してプロジェクトの概要を把握するのが困難になっていました。

これらの課題を解決するため、マインドマップUIを実装しました。

事例4：実装内容

詳細仕様

1. マインドマップを構成する要素をサブテーブルとして作成
2. マインドマップの編集機能を実装
3. 指定したスペース要素IDにマインドマップUIを表示
4. マインドマップのアイテムと、設計管理アプリを紐づける仕組みを実装
5. マインドマップのアイテムに進捗率が表示される機能を実装



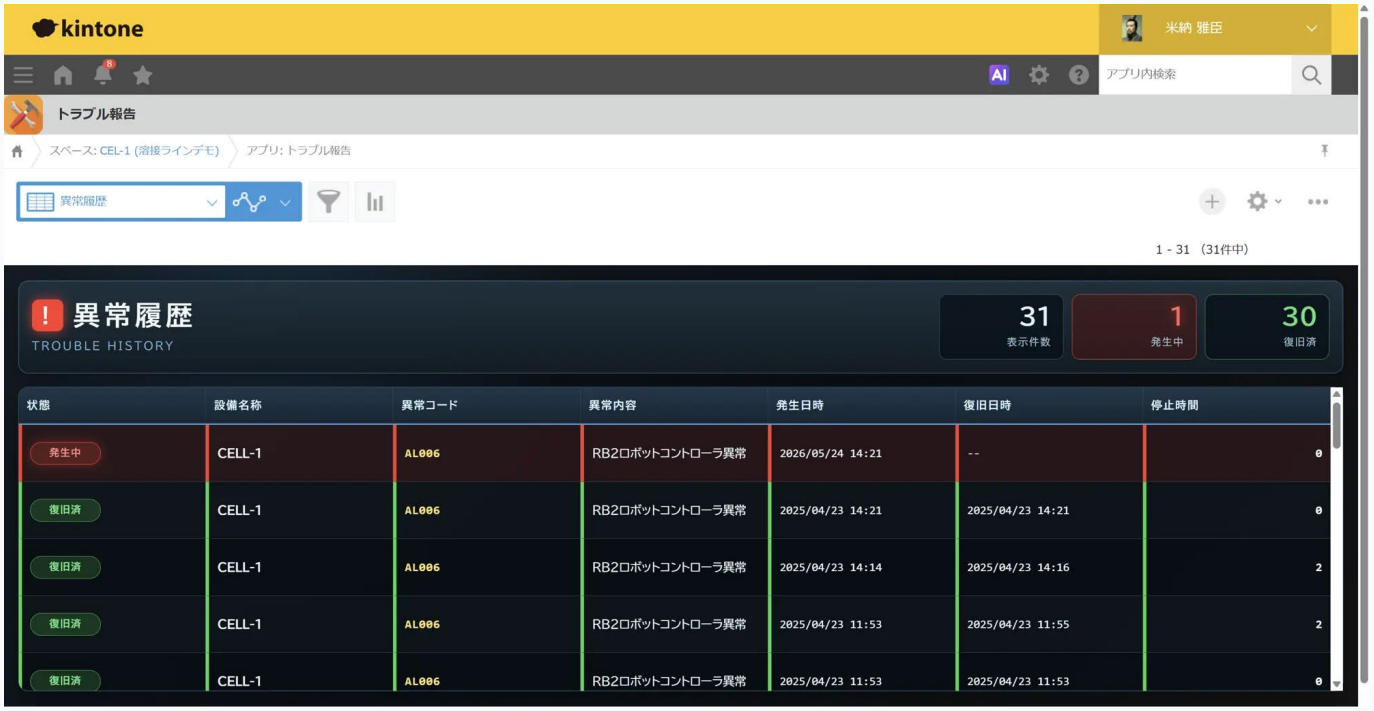
💡 実装のポイント

サブテーブルを「構造化されたデータの配列」として扱うことで、複雑なUI制御が可能になります。この手法は、マインドマップ以外にも、組織図やフローチャートなど、様々な視覚的表現に応用できます。

事例5：異常履歴画面のカスタマイズ

🎯 実現したこと

設備管理アプリの一覧画面を、FA（ファクトリーオートメーション）でよく見られる異常履歴画面のようなUIに変更し、直感的な操作性を実現しました。



背景と課題

製造現場では、設備の異常発生時に迅速な対応が求められます。しかし、標準のレコード一覧画面では表示色によるカテゴライズなどが出来ず、使いづらいという意見がありました。

これらの課題を解決するため、製造設備の異常履歴画面のUIをレコード一覧画面で表示するカスタマイズを行いました。

事例5：実装内容

詳細仕様

1. レコード一覧画面の表示形式を「カスタマイズ」に設定
2. 異常レコードを時系列で表示するリスト形式のUIを作成
3. 各異常項目に重要度に応じた色分けやアイコンを表示
4. 異常の行をクリックすると、詳細情報がポップアップで表示される機能を実装
5. PLC(Programmable Logic Controller)からkintone PLC connector(当社製品)を使用して、装置の異常内容を自動的にレコード登録

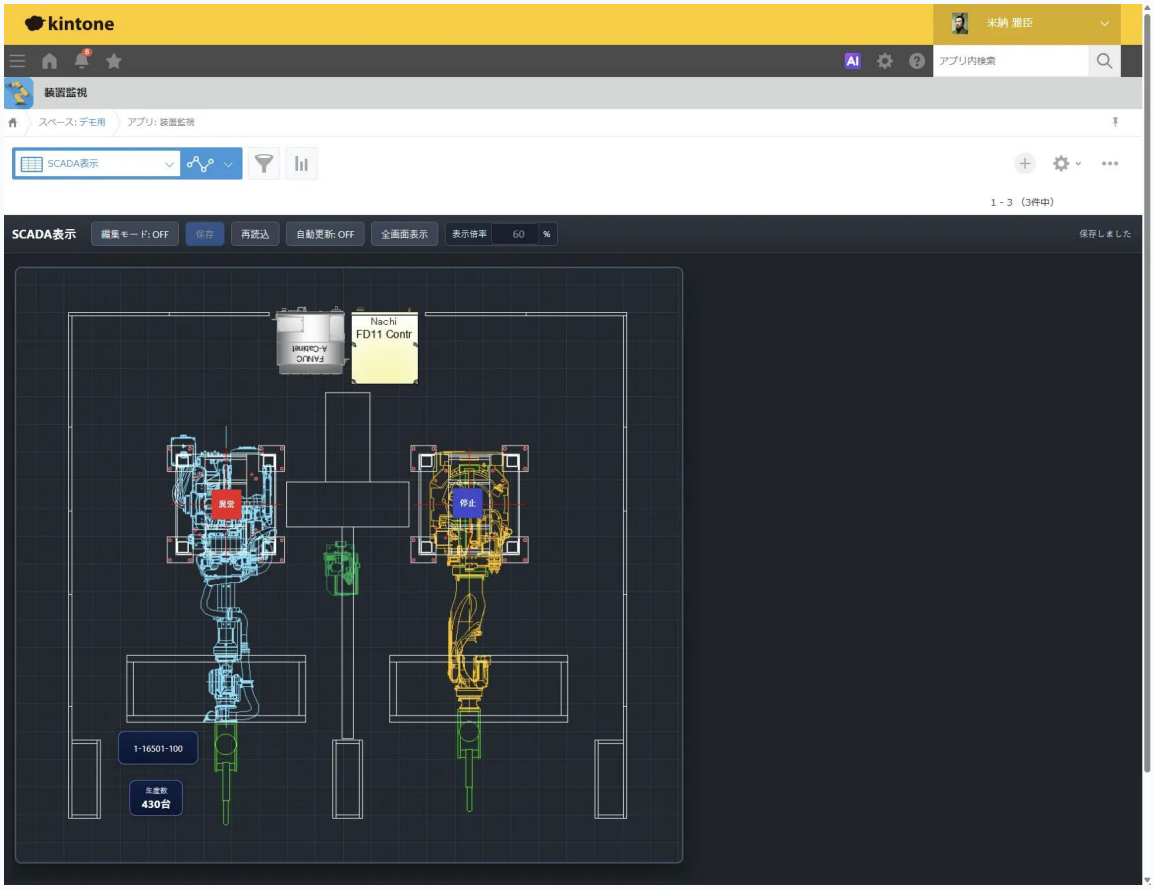
💡 実装のポイント

現場で使い慣れたUIを再現することで、操作の習熟時間を大幅に短縮できます。またポップアップでその後の処置内容を表示することで、トラブル時の処置がスムーズになります。

事例6：装置監視画面のカスタマイズ

🎯 実現したこと

装置の状態を表示するアイテムを実際の装置レイアウト上にグラフィカルに表示し、ユーザー自身でランプや数値、文字表示などのレイアウトを自由にカスタマイズできる機能を実装しました。



背景と課題

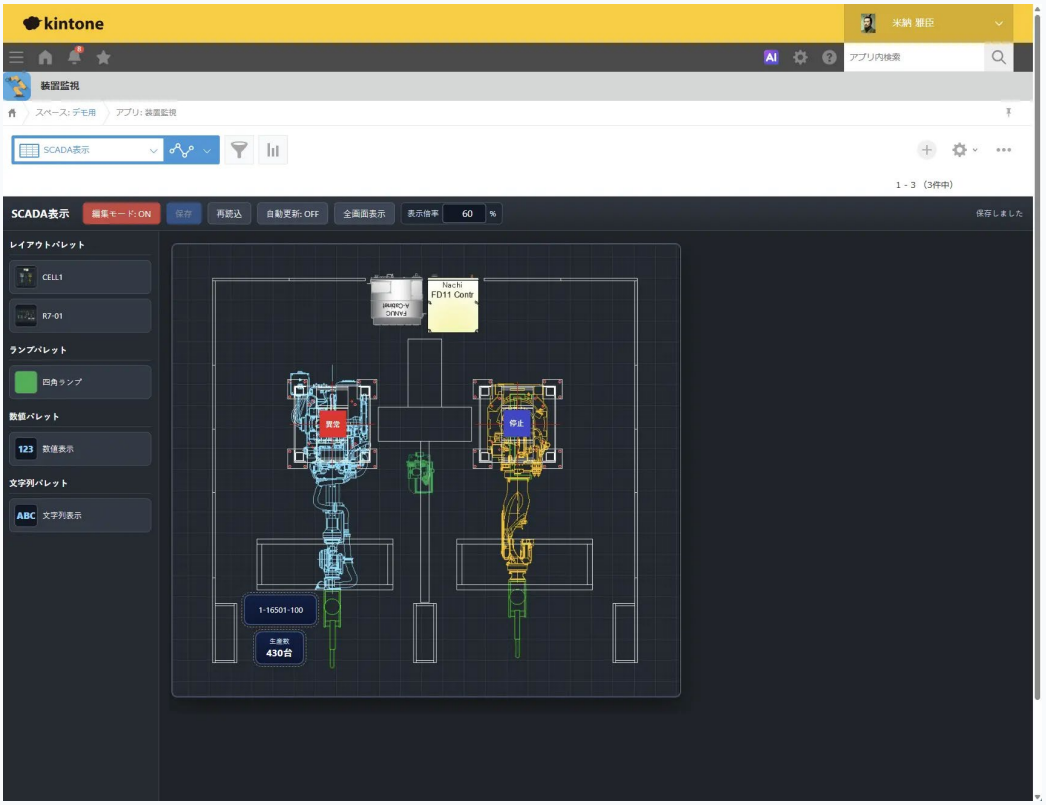
従来は装置の状態を文字列フィールドで「運転」「停止」「異常」と表示していました。

これではあまりに味気が無いのと、面白みに欠けました。

事例6：実装内容

詳細仕様

1. 装置の実際のレイアウトを背景として設定
2. フィールドと連動して表示状態が変わる以下のアイテムを配置可能にした
   ・ランプ表示
   ・数値表示
   ・文字表示
3. ランプはランプ画像登録アプリを作成することで、形状の違うランプを登録できる
4. ユーザーが自分でアイテムを配置・編集できるレイアウト機能を実装
5. PLC(Programmable Logic Controller)からkintone PLC connector(当社製品)を使用して、装置の状態を自動的にレコード登録



💡 実装のポイント

装置の状態をグラフィカルに表現することで、視認性が大幅に向上しました。また、ユーザー自身でレイアウトを変更できる機能により、装置の改造や構成変更にも柔軟に対応できるようになりました。

最後に

本記事で紹介したカスタマイズ事例は、JavaScriptの専門開発エンジニアではなくても、kintoneの仕様理解とAI活用により構築できたものです。

成功の鍵は以下の3点にあります。

1. kintoneの仕様と機能を正しく理解すること
2. 実現したい内容を明確に整理すること
3. AIに適切なプロンプトを渡すこと

はじめはAIにどのような、指示を出したらいいか分からないかも知れませんが、それも含めて、大まかな内容でAIにどのような情報が必要か聞いて進めればいいです。まずは行動あるのみ。と言いたいところですが、最後の最後に！
いきなり本番環境で初心者がカスタマイズするのはNGです！
必ず面倒でも、カスタマイズしたいアプリのコピーアプリを作成して、その環境で問題ないことを確認し、本番環境に導入してください。

🚀 まずは実践してみましょう

テスト用のアプリを作成し、簡単なカスタマイズから始めてみてください。
本記事で紹介したパターンを参考に、少しずつ実装を試していくことで、kintoneカスタマイズのスキルが自然と身についていきます。