

ハンズオンガイド

Day2 画面設計書とモック

Day1 の要件定義書から、画面一覧と遷移、工数入力画面の画面設計書を書き、動くモックで設計の抜けを見つけます。最後に請求回収の業務フローを描きます。

2026年10月15日（木）

Zoom

個人演習

Claude デスクトップアプリ

目次

本日のゴール

当日の流れ

Zoom での進め方

Day1 からの合流

D2-1 画面一覧と画面遷移

ねらい

できあがりの目安

使うファイルと作るファイル

STEP 1 現行画面の触り直し**STEP 2** 画面一覧と遷移図**STEP 3** 紐づけの抜け探し**STEP 4** 画面一覧の手直し

うまくいかないとき

早く終わった方へ

D2-2 工数入力画面の改修設計

ねらい

できあがりの目安

使うファイルと作るファイル

入力チェックの3つの書き分け

STEP 1 予定工数の項目定義5つ**STEP 2** 設計書の下書きと2つのチェック**STEP 3** Excel への変換と確認**STEP 4** 締め後と24時間チェックの判定

うまくいかないとき

早く終わった方へ

D2-3 設計書からの画面モック

ねらい

できあがりの目安

使うファイルと作るファイル

STEP 1 確かめたいこと3つ**STEP 2** モックの作成**STEP 3** ブラウザでの確認と差の洗い出し**STEP 4** v0.2 への反映

うまくいかないとき

早く終わった方へ

D2-4 請求回収の業務フロー

ねらい

できあがりの目安

使うファイルと作るファイル

STEP 1 手作業の拾い出し**STEP 2** 人と帳票の列挙と As-Is**STEP 3** To-Be の追加**STEP 4** 例外の確かめ

うまくいかないとき

早く終わった方へ

よくある質問

01

本日のゴール

Day1 の要件定義書を入力にして、工数入力画面（KIN030）の改修を画面設計書とモックまで落とします。最後に、経理の請求回収の業務フローを描きます。

持ち帰るもの

- ☐ 成果物/day2/画面一覧.md と 画面遷移.md 。現行5画面と新設1画面に、区分と対応する要件IDがある
- ☐ 成果物/day2/画面設計書_KIN030_v0.2.xlsx と、その元の 画面設計書_KIN030_v0.2.md 。項目定義、入力チェック、メッセージがそろっている
- ☐ 成果物/day2/mock_KIN030_工数入力_改修後.html 。ダブルクリックで開き、入力チェックが設計書のメッセージIDで出る
- ☐ 成果物/day2/業務フロー_請求回収.html 。As-Is と To-Be の2枚

Day3 は、この画面設計書とモックを入力に、設計のレビューと、要件との対応の確認をします。

02

当日の流れ

全体で [180min] です。講義のあとには、必ず演習か理解度チェックが入ります。

枠	中身	形式	時間
振り返り	Day1 の要件定義書の確認。未完成の方は合流版で合流	説明	[10min]
D2-1	画面一覧と画面遷移	個人演習	[30min]
D2-2	工数入力画面の改修設計	個人演習	[40min]
休憩			[10min]
D2-3	設計書からの画面モック	個人演習	[45min]
D2-4	請求回収の業務フロー	個人演習	[20min]
まとめ	設計書とモックのずれ、理解度チェック、Day3 の予告	まとめ	[10min]
質疑・アンケート		質疑	[15min]

Zoom での進め方

- 講師が Zoom で画面を共有し、同じ操作を見せながら進めます。手元では Claude デスクトップアプリと、エクスプローラー、メモ帳、LibreOffice を使います。画面が1枚の方は、Zoom の画面を小さくして横に並べると作業しやすくなります。
- 演習はすべて個人で進めます。発表や、ほかの受講者との話し合いはありません。
- 質問や、詰まったときは、Zoom のチャットに「D1-2 STEP 3 変換で文字化け」のように、演習の番号、STEP、起きていることを書いてください。講師が順に答えます。画面を見る必要があるときは、講師から画面共有をお願いすることがあります。
- 各 STEP の [Nmin] は目安です。時間になったら講師が次の STEP へ進めます。途中でも、次の STEP へ移って構いません。終わらなかった分は、研修のあとに同じ手順で続けられます。

03

Day1 からの合流

Day1 の要件定義書が v0.2 まで届かなかった方は、講師が配る合流版で Day2 を始めます。自分の v0.2 がある方は、そのまま自分のものを使ってください。

合流版を使う方の手順

講師が振り返りの時間に、要件一覧.md、要件定義書_v0.2.md、要件定義書_v0.2.docx を配ります。受け取り方は Zoom で案内します。

1 自分の途中のファイルが 成果物/day1/ にあれば、名前の末尾に _自分 を付けて残します。上書きで消さないためです。

2 受け取ったファイルを 成果物/合流用/ に置きます。

3 成果物/合流用/ の 要件一覧.md、要件定義書_v0.2.md、要件定義書_v0.2.docx を 成果物/day1/ にコピーします。

このガイドの送る文面は、どれも 成果物/day1/ のファイルを指しています。コピーしておけば、文面はそのまま使えます。

D2-1 画面一覧と画面遷移

[30min]

個人演習

ねらい

要件定義書から、改修する画面、新設する画面、影響を受ける画面を洗い出します。画面と要件を要件IDでつなぎ、どちらかに抜けがないかを見ます。

できあがりの目安

- ☐ 成果物/day2/画面一覧.md に、現行の5画面と新設の PJ010 が載っている
- ☐ 画面ごとに、区分（改修・新設・影響あり・影響なし）と、対応する要件IDがある
- ☐ 成果物/day2/画面遷移.md に遷移図があり、同じ図を描いた 成果物/day2/画面遷移.html をダブルクリックするとブラウザに図が出る

使うファイルと作るファイル

区分	ファイル	使い方
使う	成果物/day1/要件定義書_v0.2.md 成果物/day1/要件一覧.md	Day1 の成果物。合流版でも構いません
使う	現行画面/ の HTML	KIN010 から KIN050 の現行画面
作る	成果物/day2/メモ_D2-1.md	STEP 1 の手書きメモ
作る	成果物/day2/画面一覧.md	画面ごとの区分と要件ID
作る	成果物/day2/画面遷移.md 成果物/day2/画面遷移.html	遷移図。md は Mermaid、html はブラウザで見える用

現行画面を触り、変わる画面を書く

→

画面一覧と遷移図を作らせる

→

紐づけの抜けを探す

→

直して確定する

STEP 1 現行画面の触り直し

AIを開かない

[5min]

現行画面/index.html をダブルクリックして開き、5画面を順に触ります。要件定義書 v0.2 を横に開いておき、改修でどの画面が変わるかを考えます。

成果物/day2/メモ_D2-1.md に、画面ごとに1行ずつ書きます。区分（改修・新設・影響あり・影響なし）と、そう考えた理由です。

メモの形

KIN010 メニュー: 区分 / 理由
KIN020 日次勤怠入力: 区分 / 理由
KIN030 工数入力: 区分 / 理由
KIN040 交通費申請: 区分 / 理由
KIN050 月次締め・CSV: 区分 / 理由
PJ010 収支ダッシュボード: 新設 / 理由

Tips 「影響あり」は、画面そのものは直さなくても、改修の結果で表示や動きが変わる画面です。メニューに新しい画面へのリンクが増える、CSV の列が増える、などが当たります。

STEP 2 画面一覧と遷移図

[10min]

Claude に送る文面

@成果物/day1/要件定義書_v0.2.md
@成果物/day1/要件一覧.md
@現行画面/index.html

現行画面フォルダの画面の HTML を読み、改修後の画面一覧と画面遷移を作ってください。assets フォルダの中は読まなくて構いません。

- 成果物/day2/画面一覧.md に表を書く。列は 画面ID / 画面名 / 区分 / 対応する要件ID / 変わる理由
 - 区分は 改修・新設・影響あり・影響なし のどれか
 - 新設の PJ010 収支ダッシュボードも入れる
- 成果物/day2/画面遷移.md に、Mermaid で遷移図を書く
- 同じ図を 成果物/day2/画面遷移.html にする。外部のライブラリは使わず、HTML と SVG だけで描き、ダブルクリックで開けるようにする

私が現行画面を触って書いたメモは次のとおりです。区分がこれと違う画面があれば、理由を付けて教えてください。

[<ここに STEP 1 のメモを貼る>](#)

できたら、エクスプローラーで **成果物** → **day2** → **画面遷移.html** をダブルクリックして、図が出るかを確認めます。

Tips Mermaid は、文字で図を書く書き方です。Markdown の中に残しておくと、あとで Claude に図の中身を読ませたり直させたりしやすくなります。ただ、ブラウザは Mermaid をそのままでは図にできないので、見る用に HTML を別に作らせています。

STEP 3 紐づけの抜け探し

[10min]

画面一覧を読み、対応する要件IDが空いている画面に印を付けます。次に、どの画面にも出てこない要件を Claude に挙げさせます。

Claude に送る文面

@成果物/day2/画面一覧.md

@成果物/day1/要件一覧.md

要件一覧の要件IDのうち、画面一覧のどの画面にも出てこないものを挙げてください。
非機能や制約のように、画面ではなく別の場所で実現する要件は、別の表に分けてください。
画面一覧はまだ書き換えないでください。

挙げた要件を1件ずつ見て、**メモ_D2-1.md** に扱いを書きます。どの画面に足すか、画面ではない場所で実現するか、Day1 の要件の書き方が粗すぎたのか。

STEP 4 画面一覧の手直し

AIを開かない

[5min]

STEP 1 の自分の区分と Claude の区分が違った画面、STEP 3 で扱いを決めた要件を、自分で書いてから送ります。

Claude に送る文面

@成果物/day2/画面一覧.md

次のとおり画面一覧を直し、画面遷移.md と 画面遷移.html も合わせて直してください。

<ここに、直す画面と区分、足す要件IDを自分で書いて貼る>

できていれば完了

- ☐ 要件IDが空いている画面は、「影響なし」の画面だけになっている
- ☐ どの画面にも出てこない要件には、画面ではない実現場所が書いてある
- ☐ **画面遷移.html** を開き直すと、直した内容が図に出ている

うまくいかないとき

画面遷移の HTML が白い画面になる

「**成果物/day2/画面遷移.html**」をブラウザで開いたら白い画面でした。外部のファイルやライブラリを読みに行っていないか、スクリプトの誤りがないかを確かめて直してください」と送ります。ブラウザで F12 を押し、**コンソール** に赤い文字が出ていたら、それを コピーして一緒に貼ると早く直ります。

HTML を開くとメモ帳が開く

右クリック → **プログラムから開く** → Chrome か Edge を選びます。

AI がメモにない要件を足してくる

画面一覧に、要件のない画面や機能（ログイン画面、承認画面など）が増えていたら、「要件IDのない画面を挙げ、要件定義書のどこから来たかを示してください」と送ります。示せなければ消させます。既存の認証を使う KINTAI では、ログイン画面は改修の対象外です。

現行画面を全部読むのに時間がかかる

文面の「assets フォルダの中は読まなくて構いません」が効いていないと、CSS やスクリプトまで読みに行きます。止まっているように見えたら、「画面の HTML の5ファイルだけ読めば足ります」と送ってください。

早く終わった方へ

追加と考察

月次締め・CSV の画面（KIN050）は、改修の影響を受けるかを考えます。CSV 出力に予定工数の列を足すのか。パートナー実績の CSV 取込は、収支ダッシュボードの原価とどうつながるのか。自分の答えと Claude の区分が同じでも、理由まで同じかを見比べてください。

発展課題

画面ごとに、使う人と操作できることの表を作らせます。

Claude に送る文面

@成果物/day2/画面一覧.md
@成果物/day1/要件定義書_v0.2.md

画面ごとに、使う人（メンバー・PM・管理部）と、その人が操作できることの表を作ってください。
要件定義書に根拠がない権限は「要確認」にしてください。保存先は 成果物/day2/画面別権限.md です。

Day1 で食い違いとして挙げた話題に関わる画面の行を、特に見てください。

D2-2 工数入力画面の改修設計

[40min]

個人演習

ねらい

KIN030 の改修を、項目定義、入力チェック、メッセージまで書きます。入力チェックは TERASOLUNA の書き分けに沿って、単項目、相関、業務ロジックの3つに分けます。

できあがりの目安

- ☐ 成果物/day2/画面設計書_KIN030_v0.1.xlsx が LibreOffice Calc で開ける
- ☐ 項目定義の全項目に、型・桁・必須・初期値・入力者がある
- ☐ 入力チェックが、単項目3つ以上、相関3つ以上、業務ロジック1つ以上ある
- ☐ すべての入力チェックにメッセージIDがあり、メッセージの一覧に文言がある

使うファイルと作るファイル

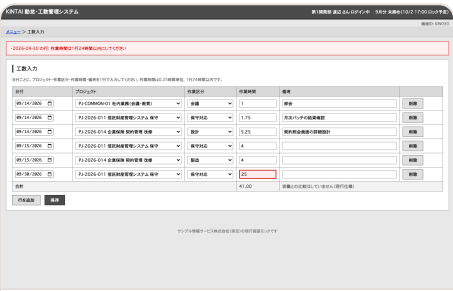
区分	ファイル	使い方
使う	テンプレート/画面設計書テンプレート.md	見出しと表の形をこのとおりにします
使う	資料/参考/TERASOLUNA_入力チェックと例外の書き分け.md	入力チェックの書き分けの決まり
使う	成果物/day1/要件定義書_v0.2.md	Day1 の成果物か合流版
使う	現行画面/KIN030_工数入力.html	今の項目と入力チェック
作る	成果物/day2/メモ_D2-2.md	STEP 1 の手書きの項目定義と、STEP 4 の判定
作る	成果物/day2/画面設計書_KIN030_v0.1.md 成果物/day2/画面設計書_KIN030_v0.1.xlsx	md が正本。xlsx は変換したもの

入力チェックの3つの書き分け

種類	判定に使うもの	例
単項目チェック	その項目の値だけ	必須、桁数、0.25時間単位

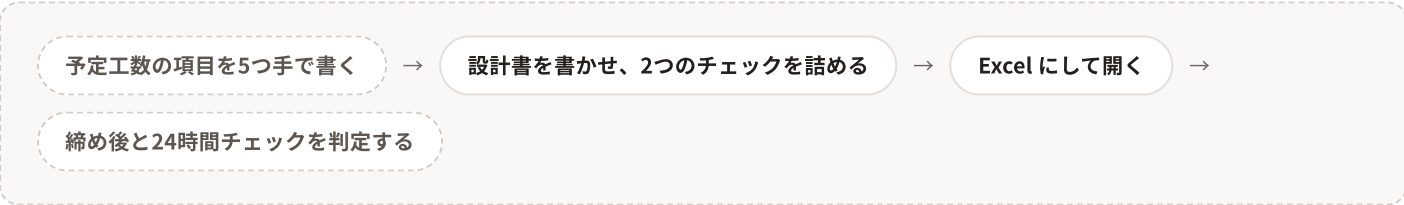
種類	判定に使うもの	例
関連チェック	同じ画面にある複数の項目の値	開始日が終了日より前か、同じ画面の行どうしを比べる
業務ロジックチェック	画面の外にある情報。データベースや他の画面の状態	締め済みの月か、プロジェクトの期間内か

現行の KIN030 のチェックは、必須、0.25時間単位、1行24時間以内の3つだけです。エラーは、画面の上部に赤字で並びます。改修後もこの出し方を続けるかは、設計書で決めることの1つです。



現行の KIN030。必須の項目が空いた行を保存したときのエラー

TERASOLUNA では、この3つを書く場所も、エラーの返し方も分けています。詳しい書き分けは、配布フォルダの参考資料資料/参考/TERASOLUNA_入力チェックと例外の書き分け.md にあります。STEP 4 で、24時間のチェックがどれに当たるかを自分で決めます。



STEP 1 予定工数の項目定義5つ

AIを開かない

[7min]

予定工数を入れるための項目を、手で5つ書きます。成果物/day2/メモ_D2-2.md に次の表の形で書いてください。

メモに貼る表の形

項目名	型	桁	必須	初期値	誰が入れるか
---	---	---	---	---	---

予定工数の粒度の決め方:

最後の1行には、予定工数を誰がどの単位で入れるのかを、どう決めたかを書きます。Day1 で食い違いとして挙げた話題です。要件一覧のその行を読み返してから決めてください。決めきれないなら、仮に採る案と、誰に確かめるかを書きます。

STEP 2 設計書の下書きと2つのチェック

[15min]

設計書を書かせる

Claude に送る文面

@テンプレート/画面設計書テンプレート.md
@資料/参考/TERASOLUNA_入力チェックと例外の書き分け.md
@成果物/day1/要件定義書_v0.2.md
@現行画面/KIN030_工数入力.html

KIN030 工数入力の、改修後の画面設計書を Markdown で書いてください。保存先は 成果物/day2/画面設計書_KIN030_v0.1.md です。

- テンプレートの ## 見出しと表の列をそのまま使う
- 項目定義は、全項目に 項目ID / 項目名 / 型 / 桁 / 必須 / 初期値 / 入力者 を書く
- 入力チェックは、参考資料の書き分けに沿って 単項目・相関・業務ロジック に分け、それぞれに種類の列を付ける
- すべてのチェックにメッセージIDを付け、メッセージの表に文言を書く
- 予定工数の項目は、私が決めた次の定義を使う。違う方がよいと思う点は、書き換えずに別に指摘する

<ここに STEP 1 の5項目と、予定工数の粒度の決め方を貼る>

入力チェックの扱いを詰める

改修依頼書にある24時間のチェックと、Day1 で入力チェックについて食い違いとして挙げた話題を、どう扱うかを Claude と詰めます。要件一覧のその行を読み返し、自分の考えを1〜2行で書いてから送ってください。

Claude に送る文面

@成果物/day1/要件一覧.md

改修依頼書の「1日の作業時間の合計が24時間を超えたら入力エラー」と、要件一覧で入力チェックについて「食い違い」にした要件の、2つの扱いを詰めたいです。

それぞれについて、単項目・相関・業務ロジックのどれに当たるか、エラーで止めるか警告にとどめるかの案を2つずつ出し、その案で困る人を書いてください。

設計書はまだ書き換えないでください。

私の今の考えは次のとおりです。

<ここに自分の考えを1〜2行で書く>

返ってきた案のうち、自分の考えと違うものを1つ選び、どこが違うのかを **メモ_D2-2.md** に書いておきます。STEP 4 で使います。

STEP 3 Excel への変換と確認

[10min]

Claude に送る文面

成果物/day2/画面設計書_KIN030_v0.1.md を tools/md_to_xlsx.py で Excel にして、成果物/day2/画面設計書_KIN030_v0.1.xlsx に保存してください。

変換ツールは、Markdown の `##` 見出しごとに1枚のシートを作り、見出しの下で最初の表をそのシートに書きます。表の前の文は、シートの上に注記として入ります。

- 1 コマンドの確かめが出たら、`tools/md_to_xlsx.py` と `成果物/day2/` の中のファイルになっているかを見て `許可` を押します。
- 2 `画面設計書_KIN030_v0.1.xlsx` を LibreOffice Calc で開き、下のシートの見出しがテンプレートの `##` 見出しと同じ数だけ並んでいるかを見ます。
- 3 入力チェックのシートで、色の付いた見出しの行を選んで `データ` → `オートフィルター` を押し、種類の列で絞り込んで、単項目・相関・業務ロジックの数を数えます。

見るところ

- ☐ 項目定義のシートで、型・桁・必須・初期値・入力者に空いたセルがない
- ☐ 入力チェックの数が、できあがりの目安に届いている
- ☐ 入力チェックのメッセージIDが、すべてメッセージのシートにある

STEP 4 締め後と24時間チェックの判定

AIを開かない

[8min]

2つのことを自分で決め、`メモ_D2-2.md` に決めたことと理由を書きます。

- 1 締めたあとに、工数を画面で直せるか。Day1 で食い違いとして挙げた話題の1つです。確認がとれるまで、設計書で何を仮に採るか、確認事項のどれに当たるかを書きます。
- 2 24時間のチェックは、相関チェックか、業務ロジックチェックか。判定に使う値がどこにあるかで決めます。

判定の手がかり

相関チェックは、画面に出ている値だけで判定できるものです。24時間の判定に使う値が、工数入力画面に出ている、同じ日の行だけでそうかを考えてください。

決めたことを書いてから、設計書に反映させます。

Claude に送る文面

成果物/day2/画面設計書_KIN030_v0.1.md を、次の判定に合わせて直してください。直したら tools/md_to_xlsx.py で xlsx も作り直してください。

<ここに、締め後の扱いと、24時間チェックの種類を、理由と一緒に自分で書いて貼る>

うまくいかないとき

python が見つからない

「py で実行してください」と送ります。それでも動かないときは、Windows の **設定** → **アプリ** → **アプリの詳細設定** → **アプリ実行エイリアス** で **python.exe** と **python3.exe** をオフにして、Claude デスクトップアプリを開き直してください。

Excel のシートが1枚しかない

Markdown の見出しが **##** になっていないか、見出しのすぐ下に表がないと、シートが分かれません。「## 見出しごとに表が1つずつあるように、**成果物/day2/画面設計書_KIN030_v0.1.md** の形を直してから変換し直してください」と送ります。

変換で文字化けする

Calc で化けているときは、Markdown をメモ帳で開き、化けていないかを見ます。Markdown をメモ帳で書き換えた場合は、**エンコード** を **UTF-8** にして保存し直してから変換します。

LibreOffice で開けない

xlsx が 0 KB なら変換が失敗しています。Claude の応答に出たエラーの文を貼り、「変換のエラーを直してください」と送ります。ダブルクリックで別のアプリが開くときは、右クリック → **プログラムから開く** → **LibreOffice Calc** です。

AI がメモにない要件を足してくる

項目定義に、要件定義書にない項目（承認者、承認日時など）が増えていたら、「要件定義書に根拠のない項目を挙げてください」と送り、消すか要確認にします。

早く終わった方へ

追加と考察

1日の合計のチェックをエラーで止めると、困る人は誰かを考えます。夜間の障害対応で、勤怠と工数を別の日に入れる人。複数のプロジェクトを兼ねていて、月末にまとめて按分する人。止めた場合と警告にとどめた場合で、管理部の月末の作業がどう変わるかまで書いてください。

発展課題

画面の注意書きを、メンバー向けに書かせます。

Claude に送る文面

@成果物/day2/画面設計書_KIN030_v0.1.md

工数入力の画面の上に出す注意書き（ヘルプ文）を、メンバー向けに100字以内で書いてください。予定工数と実績の違いと、入力チェックでエラーになる場合が、読めば分かるようにしてください。

候補を3つ出し、それぞれ何を削ったかを書いてください。

100字に収まらない説明は、画面の設計そのものが分かりにくいという合図でもあります。

D2-3 設計書からの画面モック

[45min]

個人演習

ねらい

書いた設計書が画面として成り立つかを、動くモックで確かめます。モックを作る途中で AI が決めたことを拾い、設計書の書き漏らしとして戻します。

できあがりの目安

- ☐ 成果物/day2/mock_KIN030_工数入力_改修後.html がダブルクリックで開く
- ☐ 入力チェックのエラーが、設計書のメッセージIDつきで出る
- ☐ 設計書とモックの差が2件以上見つかったり
- ☐ 差を戻した 成果物/day2/画面設計書_KIN030_v0.2.xlsx がある

使うファイルと作るファイル

区分	ファイル	使い方
使う	成果物/day2/画面設計書_KIN030_v0.1.md	D2-2 で作ったもの
使う	現行画面/KIN030_工数入力.html	見た目をそろえる元
作る	成果物/day2/メモ_D2-3.md	確かめたいこと3つと、差の表
作る	成果物/day2/mock_KIN030_工数入力_改修後.html	動くモック
作る	成果物/day2/画面設計書_KIN030_v0.2.md 成果物/day2/画面設計書_KIN030_v0.2.xlsx	差を戻した設計書。Day3 の入力になります

確かめたいことを3つ書く

→

モックを作らせる

→

ブラウザで触り、差を表にする

→

設計書を v0.2 にする

STEP 1 確かめたいこと3つ

AIを開かない

[5min]

モックで確かめたいことを3つ、成果物/day2/メモ_D2-3.md に書きます。「画面を作る」のが目的ではなく、設計書の中で自信のないところを確かめるのが目的です。

書き方の例を見る

- 予定と実績が、同じ行で見比べられるか
- 24時間を超えたときのエラーが、画面のどこに出るか
- 締めたあとの月は、入力できないようになっているか

STEP 2 モックの作成

[15min]

Claude に送る文面

@成果物/day2/画面設計書_KIN030_v0.1.md
@現行画面/KIN030_工数入力.html

設計書どおりに、改修後の工数入力画面のモックを作ってください。保存先は 成果物/day2/mock_KIN030_工数入力_改修後.html です。

- HTML と JavaScript だけで作る。1ファイルで完結させ、外部のライブラリと Web サーバは使わない
 - 見た目は現行画面とそろえる。現行画面の CSS は、中身をこのファイルに書き写す
 - 画面の項目には、設計書の項目IDを id 属性と、項目の横の小さなラベルの両方で付ける
 - 入力チェックは設計書のとおりに動かし、エラーはメッセージIDと文言で出す
 - 設計書に書いていないことを決めたときは、その箇所の HTML に「<!-- 設計書に無い: 何を決めたか -->」のコメントを残す
- このモックで確かめたいことは次の3つです。これが確かめられる作りになしてください。

<ここに STEP 1 の3つを貼る>

Tips 「設計書に無い」のコメントは、STEP 3 で AI が勝手に決めたところを拾うための目印です。モックは、設計書の空白を AI がもっともらしく埋めて動かします。埋めたことが見えないと、書き漏らしに気づけません。

STEP 3 ブラウザでの確認と差の洗い出し

[15min]

- 1 エクスプローラーで `mock_KIN030_工数入力_改修後.html` をダブルクリックし、Chrome か Edge で開きます。
- 2 STEP 1 の3つを順に試し、結果を `メモ_D2-3.md` に書きます。
- 3 わざと誤った入力をします。1行に 25 時間、0.3 時間、1日の合計が24時間を超える入力、締めたあとの月への入力。どのメッセージIDが、画面のどこに出るかを見ます。

4

設計書に書いていないのに AI が決めたところを探します。初期値、行の並び順、エラーの出る位置、ボタンの名前。

AI が決めたところは、Claude にも挙げさせます。

Claude に送る文面

@成果物/day2/mock_KIN030_工数入力_改修後.html
@成果物/day2/画面設計書_KIN030_v0.1.md

モックの中の「設計書に無い」のコメントを全部挙げ、設計書のどの章に書くべきことを表にしてください。
コメントを付け忘れたところ（初期値、並び順、エラーの出る位置など）も、設計書と見比べて挙げてください。

自分で見つけた差と、Claude が挙げた差を、**メモ_D2-3.md** の1つの表にまとめます。

メモに貼る表の形

箇所	モックでの動き	設計書の記述	どうするか
---	---	---	---
	書いていない		

STEP 4 v0.2 への反映

[10min]

表の「どうするか」は自分で書きます。モックの動きを設計書に書き足すのか、モックの方が誤りで直すのか、要確認として残すのか。

Claude に送る文面

@成果物/day2/画面設計書_KIN030_v0.1.md
@成果物/day2/mock_KIN030_工数入力_改修後.html

次の表のとおり、設計書を直して 成果物/day2/画面設計書_KIN030_v0.2.md として保存し、tools/md_to_xlsx.py で 成果物/day2/画面設計書_KIN030_v0.2.xlsx も作ってください。v0.1 は残してください。

モックも、直した設計書に合わせて直してください。

<ここに、差の表を「どうするか」まで書いて貼る>

できていれば完了

- ☐ **画面設計書_KIN030_v0.2.xlsx** が Calc で開ける
- ☐ 差の表のうち「設計書に書き足す」にした行が、v0.2 に入っている
- ☐ 直したモックを開き直すと、設計書のメッセージIDでエラーが出る

うまくいかないとき

モックが白い画面になる

ブラウザで F12 を押し、**コンソール** に赤い文字が出ていないかを見ます。出ていたらそれをコピーして、「モックを開いたら白い画面で、コンソールにこう出ていました」と貼って送ります。外部のファイルを読みに行っている、スクリプトの途中で止まっている、のどちらが多い原因です。

見た目が現行画面と違う

現行画面の CSS を、リンクで読み込もうとしていることがあります。「現行画面の CSS の中身を、このファイルの style に書き写してください」と送ります。

入力チェックが動かない、エラーが出ない

どの操作で、何が出るはずだったかを書いて送ります。「0.3 を入れて保存を押しても、E-KIN030-00x が出ません」のように、メッセージIDまで書くと早く直ります。

HTML を開くとメモ帳が開く

右クリック → **プログラムから開く** → Chrome か Edge を選びます。

AI がメモにない要件を足してくる

モックに設計書にない機能（一括登録、CSV 出力、グラフなど）が付いていたら、「設計書にない機能を挙げて、モックから外してください」と送ります。付いた機能が本当に要るなら、要件定義書に戻って要確認にします。

早く終わった方へ

追加と考察

このモックを見せたら、田中課長と佐藤さんはそれぞれ何と言いそうかを書きます。予定と実績を月末に見比べる PM と、締めと集計を受け持つ管理部では、同じ画面で見る場所が違います。2人の言いそうなのが、設計書のどこに効くかまで書いてください。

発展課題

締め状態を切り替えるボタンを付け、締めたあとの見え方を確かめます。

Claude に送る文面

@成果物/day2/mock_KIN030_工数入力_改修後.html

@成果物/day2/画面設計書_KIN030_v0.2.md

モックの上に「締め前」「締め後」を切り替えるボタンを付けてください。締め後は、設計書のとおりに入力できない項目を入力できないようにし、画面のどこかに締め状態を出してください。

設計書に締め後の見え方が書いていない箇所は、コメントで「設計書に無い」を残してください。

D2-4 請求回収の業務フロー

[20min]

個人演習

ねらい

業務フローを AI に描かせるときの渡し方を知ります。図にして、手作業と例外がどこにあるかを見ます。

できあがりの目安

- ☐ 成果物/day2/業務フロー_請求回収.html に、As-Is と To-Be の2枚の図がある
- ☐ 一部入金、振込手数料の差し引き、合算入金のうち、2つ以上が図に載っている
- ☐ To-Be の図の横に、手作業が減った箇所が書いてある

使うファイルと作るファイル

区分	ファイル	使い方
使う	資料/day2/ヒアリングメモ_04_管理部経理_山本さん_請求回収.md	経理のヒアリングメモ
使う	資料/データ/請求回収/請求一覧_202608.csv 資料/データ/請求回収/入金明細_202609.csv	発展課題で使います
作る	成果物/day2/メモ_D2-4.md	STEP 1 で拾った手作業
作る	成果物/day2/業務フロー_請求回収.html	As-Is と To-Be の図

手作業の箇所を拾う → 人と帳票を先に挙げさせてから As-Is → To-Be を足す → 例外が載っているかを見る

STEP 1 手作業の拾い出し

AIを開かない

[5min]

経理のヒアリングメモをメモ帳で読み、人が手で何かをしている箇所を3つ以上、成果物/day2/メモ_D2-4.md に書きます。1行に1つ、行番号を添えます。

Excel に転記している、目で突き合わせている、メールで問い合わせている、のような箇所です。入金が請求と合わないときの扱いも拾ってください。

STEP 2 人と帳票の列挙と As-Is

[8min]

いきなり図を描かせると、メモに出てこない部署や帳票が入ります。先に、登場する人と帳票を挙げさせ、自分の拾い出しと見比べてから描かせます。

Claude に送る文面

@資料/day2/ヒアリングメモ_04_管理部経理_山本さん_請求回収.md

請求回収の今の業務フロー（As-Is）を描きたいです。描く前に、登場する人・部署と、帳票・データを一覧にしてください。それぞれ、メモの行番号を付けてください。図はまだ描かないでください。

私がメモから拾った手作業は次のとおりです。一覧に漏れていたら足してください。

[<ここに STEP 1 のメモを貼る>](#)

一覧を見て、メモにない人や帳票が入っていたら消させます。それから図を描かせます。

Claude に送る文面

この一覧を使って、As-Is の業務フローを 成果物/day2/業務フロー_請求回収.html に描いてください。

- 人・部署ごとに横の帯（スイムレーン）を分ける
- 手作業の箱は色を変え、メモの行番号を添える
- 例外（一部入金、振込手数料の差し引き、合算入金）がメモにあれば、分かれ道として描く
- HTML と SVG だけで描き、外部のライブラリは使わない。ダブルクリックで開けるようにする

STEP 3 To-Be の追加

[5min]

Claude に送る文面

成果物/day2/業務フロー_請求回収.html の As-Is の下に、KINTAI と請求回収のデータがつながった後の To-Be の図を足してください。

図の横に、As-Is から手作業が減った箇所と、減らなかった箇所を箇条書きで書いてください。

To-Be はメモにない案を含んでよいですが、メモにない部分は箱の色を変えて区別してください。

ブラウザで開き、As-Is と To-Be が上下に並んでいることを確かめます。

STEP 4 例外の確かめ

AIを開かない

[2min]

図の中に、一部入金、振込手数料の差し引き、合算入金の分かれ道がいくつあるかを数えます。2つに届かなければ、メモの該当の行を見て、描かれていない例外を自分で書き出してから、描き足させます。

うまくいかないとき

図が白い画面になる

F12 の **コンソール** に出た赤い文字をコピーして、「業務フローの HTML が白い画面でした」と一緒に送ります。

箱の中の文字がはみ出す、線が重なる

「箱の幅を文字に合わせて広げ、線が箱と重ならないように並べ直してください」と送ります。帯（スイムレーン）の数が多いときは、画面の横幅に合わせるより、縦に長くしてよいと伝えると崩れにくくなります。

AI がメモにない業務を描いてくる

As-Is はメモにあることです。「As-Is の箱のうち、メモの行番号が付いていないものを挙げてください」と送り、消させます。To-Be は案なので、メモにないことが入って構いません。色で分かれているかを見てください。

早く終わった方へ

追加と考察

To-Be で新しく生まれた作業は何かを書きます。データがつながると、突き合わせの手作業は減ります。その代わり、取り込みの失敗を確かめる、マスタをそろえる、といった作業が増えます。増えた作業を誰が受け持つのかまで書いてください。

発展課題

請求一覧と入金明細を渡し、消し込みの突き合わせの決まりを書かせます。

Claude に送る文面

@資料/day2/ヒアリングメモ_04_管理部経理_山本さん_請求回収.md

@資料/データ/請求回収/請求一覧_202608.csv

@資料/データ/請求回収/入金明細_202609.csv

請求一覧と入金明細を突き合わせて消し込むときの決まりを、条件と処理の表で書いてください。

一部入金、振込手数料の差し引き、合算入金のそれぞれで、どう判定するかを入れてください。

決まりを当てはめると、実際のデータで判定できない入金は何件あるかも数えてください。保存先は 成果物/day2/消し込みの決まり.md です。

判定できない入金が残ったら、その入金を人がどう判断しているかを、ヒアリングで聞くことになります。

よくある質問

合流版は、どうやって受け取りますか

振り返りの時間に、講師が **成果物/合流用/** に入れるファイルを配ります。受け取り方は Zoom で案内します。自分の Day1 の成果物がある方は、使わなくて構いません。

Excel を LibreOffice Calc で直してもよいですか

見て確かめるのは Calc で構いませんが、直すのは Markdown にしてください。正本は Markdown です。Calc で直すと Markdown と食い違い、次に Claude に渡すときに古い中身を読ませることになります。

モックで作った画面は、そのまま本番に使えますか

使いません。モックは、設計書に書き漏らしがないかを確かめるための道具です。本番の KINTAI は JSP と Spring MVC で作られており、モックの HTML とは作りが違います。

Mermaid が分かりません

図を文字で書く書き方です。研修では書き方を覚える必要はありません。Markdown に Mermaid で残しておく、あとで Claude に図の中身を読ませたり直させたりしやすくなる、とだけ覚えておいてください。

テンプレートの見出しを変えたいです

研修中は変えずに進めてください。Day3 で、要件と画面設計書の対応表を作るときに、見出しがそろっていることを前提にしています。自分の現場で使うときは、テンプレートを直して使ってください。

モックをブラウザで開いたら、会社の設定で JavaScript が止まりました

講師に知らせてください。その間は、Claude に「モックの入力チェックが、どの操作でどのメッセージを出すかを表にしてください」と頼み、表で設計書との差を探して進められます。