

上流工程 要件定義・設計 生成AI活用研修

Day2

画面設計書とモック

株式会社インフォメーション・ディベロプメント様
2026年10月15日（木） オンライン



本資料の二次転載禁止について

本資料の著作権は株式会社ギブリーに帰属します。受講者の学習を目的として提供するものです。

本資料の全部または一部を、当社の許諾なく複製、転載、配布、公衆送信することを禁じます。
社内での共有をご希望の場合は、事務局を通じてご相談ください。

個人での録画・録音はご遠慮ください。

講師紹介



人的資本インテリジェンス部門講師・生成AI
エバンジェリスト

安田 光喜

Mitsuki Yasuda

公立はこだて未来大学 大学院（メディアデザイン領域）を卒業し、街づくり会社での複合商業ビルの立ち上げにおける情報インフラ整備やデザイン全般業務の責任として関与。その後デザイン事務所を立ち上げ、飲食店を中心としたデザイン業務や美術館展示のアートコンテンツ開発などを行う。また、地域ブランディングにも注力し、市主催の企画運営業務を行なった。2018年10月『小中学生向けのプログラミング教育』に注目し、函館市で初めての小中学生向けプログラミングスクール

『D-SCHOOL北海道』を立ち上げて運営。その後北海道のドラッグストア『サッポロドラッグストアー』からスクールの買収を受け、教育を専門とするグループ会社『株式会社シーラクス』にて技術開発責任者として運営。スクール設計・運営、20以上の自治体連携（イベントや出張教室運営）、大人向けプログラミングスクールメンターなどさまざまな『次世代IT教育』に取り組む。

Dify公式資料の日本語翻訳や、SecondMeアンバサダーなど生成AI最新トレンドに注力。

得意領域

- ・生成AIをはじめとした世界最先端トレンド
- ・生成AIを用いた新規事業の開発、運営、補助
- ・アプリケーション開発/自動化プログラム環境構築
- ・教育（大学講師、研修、スクール運営など）

実績

- ・生成AI研修
- ・大手企業、外資、教育機関(小中高大)でのITトレンド講演
- ・自治体と連携した地域教育実績
- ・上場企業社員対象のリスキリング支援

メッセージ

みなさんがこれから生成AIをパートナーに新しい時代を生きていけるように、3日間全力でサポートしていきます！
なにかご不明点などありましたら、講師陣になんでも気軽にご質問ください。

[#ClaudeCode](#) [#Codex](#) [#Cursor](#) [#Antigravity](#)
[#全国統一生成AI活用技能試験S1](#)
[#生成AIパスポート](#) [#G検定](#)

本日のゴール

要件定義書から画面設計書を書き モックで**設計の抜け**を見つけられる

設計書は Excel、モックは HTML の形で手元に残します。

今日の入力

Day1 に作った要件定義書です。仕上がっていない方は、講師が配る合流版から始められます。

3回の見取り図

Day1 の要件定義書が**今日の画面設計の入力**になります

Day1

10月8日（木）

ヒアリングメモから要件定義書へ

作るもの 要件一覧、要件定義書、確認事項



Day2

10月15日（木）

画面設計書とモック

作るもの 画面一覧、KIN030 の画面設計書、モック、業務フロー

今日



Day3

10月22日（木）

設計レビューと収支ダッシュボード

作るもの レビュー記録、対応表、収支ダッシュボード、レビュー会資料

本日の進め方

前半で設計書を書き後半でモックにして確かめます

コマ	中身	時間
振り返り	Day1 の要件定義書の確認、合流版の案内	[10min]
D2-1	画面一覧と画面遷移	[30min]
D2-2	工数入力画面の改修設計	[40min]
休憩		[10min]
D2-3	設計書からの画面モック	[45min]
D2-4	請求回収の業務フロー	[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	[10min]
質疑・アンケート		[15min]

Day1 で作ったもの

今日は**この3つ**を開いた状態から始めます

成果物/day1/ のファイル	中身
要件一覧.md	要件25件以上、全件に根拠のファイル名と行番号
要件定義書_v0.2.md と .docx	抜け漏れを反映した版。Claude に渡すのは md
確認事項.md	相手・聞くこと・なぜ を5件以上

最初に確かめること

要件定義書の要件ID の振り方を見ておきます。今日の画面一覧と画面設計書は、この要件ID で要件とつながります。

Day1 が途中の方は合流版から始めてください

Day1 の成果物が
そろっている



自分の 成果物/day1/ をそのまま使います
送る文面のファイルの場所も変えません

Day1 の成果物が
途中で止まっている



合流版を 成果物/day1/ にコピーして使います
講師が研修の始めに配ります

合流用に入るファイル

要件一覧.md

要件定義書_v0.2.md

要件定義書_v0.2.docx

成果物/合流用/ に置いたあと 成果物/day1/ にコピーすれば、ガイドの送る文面をそのまま使えます。

今日使うファイル・作るファイル

今日作るものは全部 **成果物/day2/** に置きます

使うもの	演習
成果物/day1/要件定義書_v0.2.md、成果物/day1/要件一覧.md	D2-1・D2-2
テンプレート/画面設計書テンプレート.md	D2-2
資料/参考/TERASOLUNA_入力チェックと例外の書き分け.md	D2-2
現行画面/ の HTML	D2-1～D2-3
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	D2-4
作るもの（成果物/day2/）	演習
メモ_D2-1.md ～ メモ_D2-4.md	各演習の STEP 1
画面一覧.md、画面遷移.md、画面遷移.html	D2-1
画面設計書_KIN030_v0.1.md と .xlsx	D2-2
画面設計書_KIN030_v0.2.md と .xlsx、mock_KIN030_工数入力_改修後.html	D2-3
業務フロー_請求回収.html	D2-4

D2-1

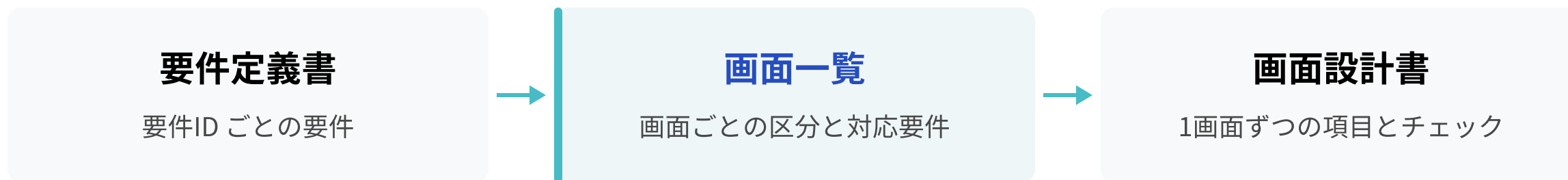
画面一覧と画面遷移

[30min]

要件定義書から、改修・新設・影響を受ける画面を洗い出します。

画面一覧から始める理由

設計は影響する画面を**全部並べてから**始めます



1画面ずつ書き始めると起きること

改修の中心の画面だけを詳しく書き、データを受け取る側の画面を見落とします。
工数入力に項目が増えたとき、月次締め CSV 出力は変わらないのか。
この問いは、画面を一覧に並べたときに初めて出てきます。

現行の5画面と新設の1画面

現行画面は [現行画面/index.html](#) から開けます



KIN010 メニュー（実画面）

KIN030 は、日付ごとにプロジェクト・作業区分・作業時間・備考を1行ずつ入れる画面です。PJ010 は画面が無く、改修依頼書 KINTAI-2026-03 の3番目で新しく作ります。

画面ID	画面名	使う人
KIN010	メニュー	全員
KIN020	日次勤怠入力	全員
KIN030	工数入力	全員
KIN040	交通費申請	全員
KIN050	月次締め・CSV	PM・管理部
PJ010	収支ダッシュボード	改修依頼で新設

画面の区分

影響なしも確かめた結果として一覧に残します

区分	意味
改修	画面の項目や動きを変える
新設	画面を新しく作る
影響あり	画面そのものは直さないが、改修の結果で表示や動きが変わる
影響なし	変わらないことを確かめた。理由を1行書く

画面一覧.md の列

画面ID	画面名	区分	対応要件ID	理由
------	-----	----	--------	----

対応要件ID は要件定義書の要件ID です。1つの画面に複数の要件がつながってかまいません。

画面遷移の書き方

遷移は Mermaid の文字で残し見る用に HTML も作ります

画面遷移.md (現行の遷移を書いた例)

```
flowchart LR
```

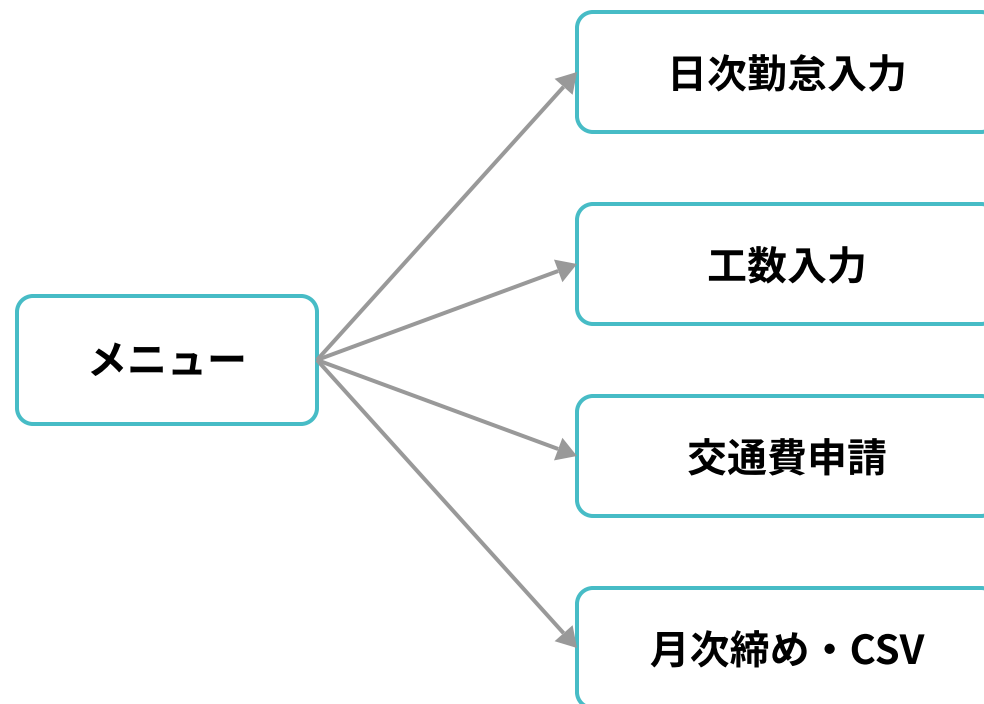
```
KIN010[メニュー] --> KIN020[日次勤怠入力]
```

```
KIN010 --> KIN030[工数入力]
```

```
KIN010 --> KIN040[交通費申請]
```

```
KIN010 --> KIN050[月次締め・CSV]
```

ブラウザは Mermaid をそのままでは図にできません。
見る用に 画面遷移.html を別に作らせます。
矢印に名前を付けるときは A -->|保存| B と書きます。



同じ遷移を 画面遷移.html に描いた図

出典：Mermaid 「Flowcharts Syntax」 <https://mermaid.js.org/syntax/flowchart.html> (2026年9月24日確認)

改修・新設・影響を受ける画面を一覧で洗い出します

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。

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.html をダブルクリックするとブラウザに図が出る

区分	ファイル
使う	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/ の HTML
作る	成果物/day2/ の メモ_D2-1.md、画面一覧.md、画面遷移.md、画面遷移.html

最初の5分は AI を開かずに現行画面を触ります

■ 自分の手で ■ AIと一緒に

STEP1
[5min]

STEP2
[10min]

STEP3
[10min]

STEP4
[5min]

■ **STEP1** **現行画面の触り直し**
[5min] 5画面を触り、区分と理由を メモ_D2-1.md に1行ずつ書きます

■ **STEP2** **画面一覧と遷移図**
[10min] 画面一覧.md、画面遷移.md、画面遷移.html を作らせます

■ **STEP3** **紐づきの抜け探し**
[10min] どの画面にも出てこない要件を挙げさせ、扱いを決めます

■ **STEP4** **画面一覧の手直し**
[5min] 直す画面と区分を自分で書いてから、直させます

STEP 1 で書いたメモを差し込んでから送ります

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。

1. 成果物/day2/画面一覧.md に表で書く。列は 画面ID / 画面名 / 区分 / 対応する要件ID / 変わる理由
2. 成果物/day2/画面遷移.md に、Mermaid で遷移図を書く
3. 同じ図を 成果物/day2/画面遷移.html にする。HTML と SVG だけで描く

私が現行画面を触って書いたメモは次のとおりです。

<ここに STEP 1 のメモを貼る>

要点だけ載せています。全文はハンズオンガイドの STEP 2 からコピーします。

入力欄で @ を打つと、ファイルを指定できます。

AI の一覧は要件ID の紐づけから疑います

見る観点

- 要件ID の紐づけが無い画面はないか
- どの画面にもつながらない要件はないか
- 影響なしの画面に、確かめた理由が書いてあるか

よくある落とし穴

- 要件定義書にない画面を AI が足してくる
- 区分が全部「改修」になり、差が見えない
- 画面遷移.html が白い画面になる。F12 のコンソールの赤い文字を貼って直させる

早く終わった方は月次締め画面への影響を考えます

追加と考察

CSV 取込の画面（KIN050）は、今回の改修の影響を受けるかを理由とともに書きます。

発展課題

画面ごとに、使う人（メンバー・PM・管理部）と操作できることの表を AI に作らせます。

月次締め・CSV

利用者: PM・管理部。締め状態の確認、勤怠・工数 CSV の出力、パートナー実績 CSV の取込を行います。

9月分 未締め(10/2 17:00 ロック予定)

対象月	2026年9月
締め状態	9月分 未締め(10/2 17:00 ロック予定)
ロック予定	2026-10-02 17:00(翌月2営業日目)

CSV 出力

9月分の勤怠・工数を CSV でダウンロードします。

CSV 出力

パートナー実績 CSV 取込

協力会社から受け取った週単位の実績 CSV を取り込みます。

Choose File No file chosen

取込

KIN050 月次締め・CSV（実画面）

D2-2

工数入力画面の改修設計

[40min]

予定工数を足す 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。

工数入力画面 KIN030 の現行

現行のチェックは作業時間の3つだけです

工数入力

日付ごとに、プロジェクト・作業区分・作業時間・備考を1行で入力してください。作業時間は0.25時間単位、1行24時間以内です。

日付	プロジェクト	作業区分	作業時間
09/14/2026	PJ-COMMON-01 社内業務(会議・教育)	会議	1
09/14/2026	PJ-2026-011 信託財産管理システム 保守	保守対応	1.75
09/14/2026	PJ-2026-014 企業保険 契約管理 改修	設計	5.25
09/15/2026	PJ-2026-011 信託財産管理システム 保守	保守対応	4
09/15/2026	PJ-2026-014 企業保険 契約管理 改修	製造	4
合計			16.00

行を追加

保存

現行画面/KIN030_工数入力.html（実画面）

現行のチェック

- 1 作業時間は必須
- 2 0.25時間単位
- 3 1行あたり24時間以内

改修で足すもの

予定工数の入力と、1日の作業時間の合計が24時間を超えたときのエラーです。

合計の行は表示するだけで、
実働との比較はしていません。

画面設計書のシート構成

Markdown の ## 見出し1つが Excel の1シートになります

画面概要

項目定義

入力チェック

メッセージ

画面遷移

テーブル変更

シート	書くこと
画面概要	画面ID・区分・対象要件・締め後の扱い・性能
項目定義	全項目の型・桁・必須・初期値・入力者
入力チェック	チェックの種類（単項目・相関・業務ロジック）と内容
メッセージ	文言と表示位置。全チェックにメッセージID
画面遷移	保存やエラーのときの行き先
テーブル変更	画面から逆算した列の追加と変更

青のシートが今日の中心です。変換は tools/md_to_xlsx.py が行います。

ID の振り方と項目定義の列

ID の振り方はテンプレートで決まっています

ID	例	振り方
項目	KIN030-I-001	画面ID-I-連番
チェック	KIN030-C-001	画面ID-C-連番
エラーのメッセージ	E-KIN030-001	E-画面ID-連番
情報のメッセージ	I-KIN030-001	I-画面ID-連番

項目定義の列

項目ID	項目名	型	桁	必須	初期値	入力者	備考
------	-----	---	---	----	-----	-----	----

型は 文字・数値・日付・選択・ボタンのどれか、必須は ○ か × で書きます。

備考には対応する要件ID と、締め後に編集できるかを書きます。

TERASOLUNA の入力チェックの分類

チェックは単項目・相関・業務ロジックの3種類に分けて書きます

アプリケーション層で行う入力チェック

単項目チェック

「単一のフィールドで
完結するチェック」

相関チェック

「複数のフィールドを
比較するチェック」

ドメイン層で行う

業務ロジックチェック

システムの状態によって
結果が変わるチェック

ガイドラインは「業務ロジックチェックは、ドメイン層で行うことをポリシーとする」としています。
研修ではコードを書かず、設計書の種類の欄をこの3つで書き分けます。

出典：NTT DATA 「TERASOLUNA Server Framework for Java (5.x) Development Guideline」

5.11.0.RELEASE 「4.2. 入力チェック」 2026年9月24日確認

<https://terasolunaorg.github.io/guideline/current/ja/ArchitectureInDetail/WebApplicationDetail/Validation.html>

単項目チェック

1つの項目の値だけを見て決まるチェック

作業時間

2.5

必須

空のまま保存しない

0.25時間単位

2.3 のような値を入れない

1行あたり24時間以内

1行に 25 と入れない

現行の KIN030 のチェック3つは、どれも作業時間の欄だけで決まります。

ガイドラインの例は入力必須・桁・型のチェックで、実装では Bean Validation を使います。

出典：NTT DATA 「TERASOLUNA Server Framework for Java (5.x) Development Guideline」

5.11.0.RELEASE 「4.2. 入力チェック」 2026年9月24日確認

<https://terasolunaorg.github.io/guideline/current/ja/ArchitectureInDetail/WebApplicationDetail/Validation.html>

関連チェック

画面の複数の項目を比べて決まるチェック

日次勤怠入力

勤務区分・出勤・退勤・休憩(分)を入力してください。実働は自動で計算します。

日付	勤務区分	出勤	退勤
2026-09-17(木)	出勤	09:00 AM	06:00 PM
2026-09-18(金)	出勤	08:45 AM	05:45 PM
2026-09-24(木)	出勤	09:00 AM	06:30 PM
2026-09-25(金)	出勤	09:15 AM	07:00 PM
2026-09-29(火)	選択してください	--:--	--:--

保存

KIN020 日次勤怠入力（実画面）

D2-2 で考えること

KIN030 の改修後の画面で、複数の項目を比べるチェックにはどんなものがあるか。

出典：NTT DATA「TERASOLUNA Server Framework for Java (5.x) Development Guideline」

5.11.0.RELEASE「4.2. 入力チェック」2026年9月24日確認

<https://terasolunaorg.github.io/guideline/current/ja/ArchitectureInDetail/WebApplicationDetail/Validation.html>

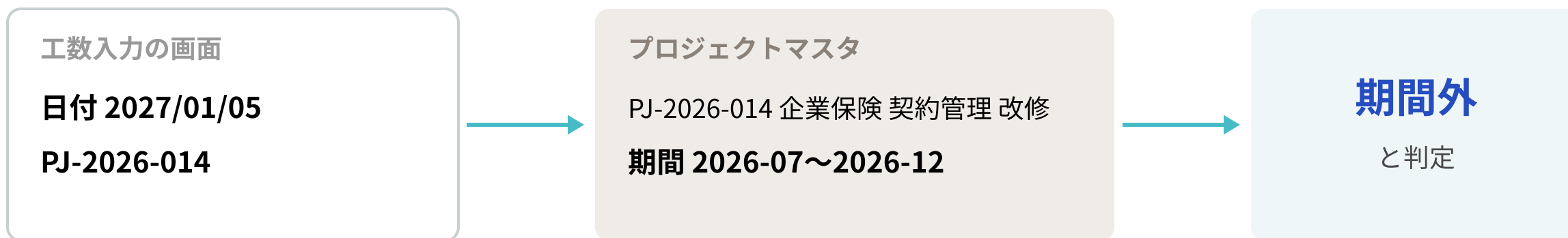
例

退勤は出勤より後の時刻

出勤の欄と退勤の欄を比べないと判定できません。
ガイドラインの例は、パスワードと
確認用パスワードの一致です。

業務ロジックチェック

画面の外の状態で結果が変わるチェックは業務ロジック



日付の欄とプロジェクトの欄だけを見ても、期間内かどうかは分かりません。マスタの内容が変われば、同じ入力でも結果が変わります。TERASOLUNA ではこの種類のチェックをドメイン層に置きます。

出典：NTT DATA 「TERASOLUNA Server Framework for Java (5.x) Development Guideline」
5.11.0.RELEASE 「4.2. 入力チェック」 2026年9月24日確認
<https://terasolunaorg.github.io/guideline/current/ja/ArchitectureInDetail/WebApplicationDetail/Validation.html>

チェックの種類の見分け方

迷ったら画面の外を見るかどうかから判定します



業務例外とシステム例外

利用者が直せる例外は**同じ画面**にメッセージを出します

業務例外

「ビジネスルールの違反を検知したことを通知する例外」

画面の出し方

入力した画面に戻り、
直し方の分かるメッセージを出す

例

保存しようとしたプロジェクトがすでに終了している

システム例外

「システムが、正常稼働している時に、発生してはいけない状態を検知したことを通知する例外」

画面の出し方

共通のエラー画面へ移る。
技術的な中身は画面に出さない

例

あるはずのマスタのデータが見つからない

出典：NTT DATA 「TERASOLUNA Server Framework for Java (5.x) Development Guideline」

5.11.0.RELEASE 「4.3. 例外ハンドリング」 2026年9月24日確認

<https://terasolunaorg.github.io/guideline/current/ja/ArchitectureInDetail/WebApplicationDetail/ExceptionHandling.html>

メッセージの書き方

全てのチェックにメッセージID を付けます

・2026-09-30 の行: 作業時間は1行24時間以内にしてください

工数入力

日付ごとに、プロジェクト・作業区分・作業時間・備考を1行で入力してください。作業時間は0.25時間単位、1行24時間以内です。

日付	プロジェクト	作業区分	作業時間
09/14/2026	PJ-COMMON-01 社内業務(会議・教育)	会議	1
09/14/2026	PJ-2026-011 信託財産管理システム 保守	保守対応	1.75
09/14/2026	PJ-2026-014 企業保険 契約管理 改修	設計	5.25
09/15/2026	PJ-2026-011 信託財産管理システム 保守	保守対応	4
09/15/2026	PJ-2026-014 企業保険 契約管理 改修	製造	4
09/30/2026	PJ-2026-011 信託財産管理システム 保守	保守対応	25
合計			41.00

行を追加

保存

メッセージのシートの列

列	書くこと
メッセージID	E-KIN030-001
種別	エラー・警告・情報
文言	利用者が直し方の分かる文
表示位置	項目の横か画面上部
使うチェックID	KIN030-C-001

現行はエラーを画面上部にまとめて並べる（実画面）

表示位置は現行と同じにするか、項目の横に変えるかを設計で決めます。

決めないままモックを作ると、AI が位置を決めます。

「退勤は出勤より後の時刻」はどの種類のチェックですか

A 単項目チェック

B 相関チェック

C 業務ロジックチェック

D システム例外

まず自分で考えてみてください。答えはチャットに記号だけ書いてください。

業務例外のメッセージはどこに出しますか

- A 共通のエラー画面
- B メニュー画面（KIN010）
- C 画面には出さずログにだけ残す
- D 入力した画面（項目の横か画面上部）

まず自分で考えてみてください。答えはチャットに記号だけ書いてください。

答えはB・Dの順です

Q1

B

関連チェック

出勤と退勤の2つの欄を比べて決まるので関連チェックです。
システム例外はチェックの種類ではありません。

Q2

D

入力した画面（項目の横か画面上部）

利用者が入力を直せば解消するので、同じ画面に戻して直し方を伝えます。
共通のエラー画面はシステム例外の出し方です。

予定工数を足す **KIN030** の画面設計書を書きます

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。

できあがりの目安

- ☐ 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
- ☐ 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
- ☐ 成果物/day2/画面設計書_KIN030_v0.1.xlsx が LibreOffice Calc で開ける

区分	ファイル
使う	テンプレート/画面設計書テンプレート.md、要件定義書_v0.2.md、現行画面/KIN030_工数入力.html
使う	資料/参考/TERASOLUNA_入力チェックと例外の書き分け.md
作る	成果物/day2/ の メモ_D2-2.md、画面設計書_KIN030_v0.1.md と .xlsx

予定工数の項目は手で5つ書いてから AI に渡します

■ 自分の手で ■ AIと一緒に

STEP1
[7min]

STEP2
[15min]

STEP3
[10min]

STEP4
[8min]

■ **STEP1** **予定工数の項目定義5つ**
[7min] 5項目と、Day1 で食い違いとして挙げた話題の決め方を書きます

■ **STEP2** **設計書の下書きと2つのチェック**
[15min] 設計書を書かせ、1日の合計のチェックの扱いを AI と詰めます

■ **STEP3** **Excel への変換と確認**
[10min] md_to_xlsx.py で Excel にし、Calc でチェックの数を数えます

■ **STEP4** **締め後と24時間チェックの判定**
[8min] 締め後に直せるかと、24時間チェックの種類を自分で決めます

STEP 1 で書いた5項目を差し込んでから送ります

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。

- 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

<ここに STEP 1 のメモを貼る>

要点だけ載せています。全文はハンズオンガイドの STEP 2 からコピーします。

続けて、1日の合計のチェックの扱いを、自分の考えを1～2行書いてから詰めます。

Excel にしたら LibreOffice Calc で開いて確かめます

画面設計書_KIN030_v0.1.md



tools/md_to_excel.py



画面設計書_KIN030_v0.1.xlsx

成果物/day2/画面設計書_KIN030_v0.1.md を tools/md_to_excel.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。

コマンドの確かめが出たら、tools/md_to_excel.py と 成果物/day2/ のファイルかを見て許可します。
python が見つからないと出たら「py で実行してください」と送ります。

Calc で見るところ

- ☐ シートがテンプレートの ## 見出しと同じ数だけ並んでいる
- ☐ 入力チェックのシートをオートフィルターで絞り、種類ごとの数を数える

もっともらしい桁と初期値ほど根拠を確かめます

見る観点

- チェックの種類が全部「単項目」になっていないか
- 全てのチェックにメッセージID があるか
- 桁と初期値の根拠が要件定義書にあるか。
無ければ（要確認）

よくある落とし穴

- AI が桁や初期値をもっともらしく決め、
要確認が付かない
- 要件定義書にない項目を足してくる
- エラーの表示位置を決めないまま次へ進む

早く終わった方は**困る人**の側から設計を見直します

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。

発展課題

画面の注意書き（ヘルプ文）を、メンバー向けに100字で AI に書かせます。

休憩

[10min]

講師の合図で再開します

再開後は、書いた設計書を画面のモックにして確かめます。

D2-3

設計書からの画面モック

[45min]

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。

モックで確かめる理由

画面にすると書いていないことが目に見えます

設計書だけのとき

書き漏らしがあっても、文書はエラーを出しません。読む人が頭の中で補ってしまうので、抜けに気づくのはたいてい製造やテストの段階です。

モックを起こしたとき

初期値や並び順、エラーの出る位置を決めないと画面になりません。決めていない箇所はAIが埋めるので、設計書との差として見えてきます。

モックはHTMLとJavaScriptだけで作り、ダブルクリックでブラウザに表示します。JDKやWebサーバは使いません。

設計書に無くて AI が埋める箇所

設計書に書いていない箇所は AI が黙って決めます

設計書 v0.1 に書いたこと

項目と型・桁・必須

入力チェックの種類と内容

メッセージID と文言



モックを作るときに AI が決めがちなこと

初期値

予定工数の欄に最初から入れておく値

並び順

予定と実績を横に並べるか分けるか

エラーの出る位置

項目の横か、画面上部にまとめるか

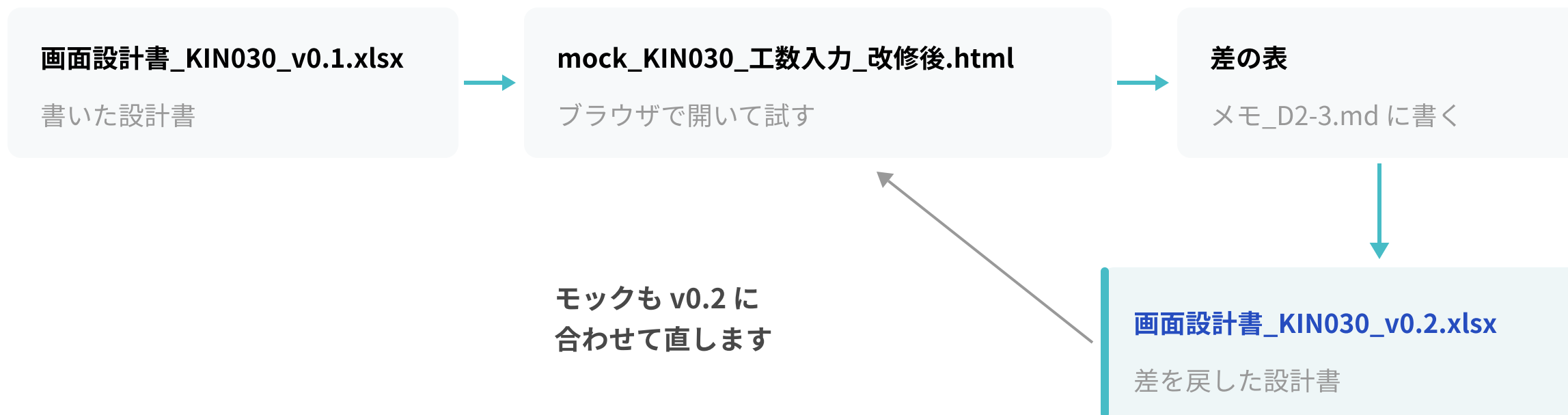
確かめる時点

入力した瞬間か、保存を押したときか

右の4つは、書いていなければ AI がそれらしく決めます。STEP 3 ではこれを探して差の表にします。

設計書とモックの往復

モックで見つけた差は設計書に戻して v0.2 にします



モックに合わせて設計書を黙って書き換えないでください。どちらに合わせたかを差の表に残します。

差の表の書き方

差は1行ずつ書き足す・直す・要確認のどれかに決めます

箇所	モックでの動き	設計書の記述	どうするか
エラーの出る位置	画面上部にまとめて出た	書いていない	設計書に書き足す

メモ_D2-3.md の表の例。行は見つけた差の数だけ足します

書き足す

モックの動きを設計書に書き足す

直す

モックの方が誤りなので直す

要確認

決めきれないので相手に聞く

「どうするか」は自分で書きます。要確認にした差は、誰に聞くかも書いておきます。

設計書から KIN030 のモックを起こして**抜け**を探します

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。

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件以上見つかっている
- ☐ 差を戻した 成果物/day2/画面設計書_KIN030_v0.2.xlsx がある

区分	ファイル
使う	成果物/day2/画面設計書_KIN030_v0.1.md、現行画面/KIN030_工数入力.html
作る	メモ_D2-3.md、mock_KIN030_工数入力_改修後.html、画面設計書_KIN030_v0.2.md と .xlsx

確かめたいことを先に3つ書いてから作らせます

■ 自分の手で

■ AIと一緒に

STEP1

[5min]

STEP2

[15min]

STEP3

[15min]

STEP4

[10min]

■ STEP1
[5min]

確かめたいこと3つ

設計書の中で自信のないところを3つ、メモ_D2-3.md に書きます

■ STEP2
[15min]

モックの作成

設計書に無いことを決めた箇所には、コメントを残させます

■ STEP3
[15min]

ブラウザでの確認と差の洗い出し

わざと誤った入力を試し、AI が決めたところを表にします

■ STEP4
[10min]

v0.2 への反映

どうするかを自分で書き、v0.2 の md と xlsx を作らせます

モックの条件は毎回同じ**5つ**を付けます

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 属性と項目の横のラベルで付ける
- 入力チェックは設計書のとおり to 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ら「<!-- 設計書に無い: 何を決めたか -->」を残す

このモックで確かめたいことは次の3つです。

<ここに STEP 1 の3つを貼る>

要点だけ載せています。全文はハンズオンガイドの STEP 2 からコピーします。

「設計書に無い」のコメントは、STEP 3 で AI が決めたところを拾う目印です。

モックと設計書を横に並べて確かめます

1 開く

エクスプローラーで 成果物/day2/ を開き、
mock_KIN030_工数入力_改修後.html をダブルクリックします

2 並べる

Chrome か Edge の横に、LibreOffice Calc で開いた設計書を置きます

3 試す

STEP 1 の3つを順に試し、わざと誤った入力もします。
1行に 25 時間、0.3 時間、1日の合計が24時間超、締めたあとの月への入力

4 直したら

ブラウザで再読み込みして、同じ操作をもう一度試します

開かないときは、拡張子が .html か、ファイル名に余分な文字が付いていないかを見ます。

見るのはモックの出来より**設計書との差**です

見る観点

- 全ての項目に設計書の項目ID が付いているか
- エラーが設計書のメッセージID と文言で出るか
- 設計書に無いチェックが動いていないか

よくある落とし穴

- 外部のファイルを読みに行き、白い画面になる
- 設計書にない機能（一括登録やグラフ）が付いてくる
- モックに合わせて設計書を黙って書き換える

早く終わった方は締め後の見え方も確かめます

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。

発展課題

締め状態（締め前・締め後）を切り替えるボタンを付け、締め後の見え方を確かめます。

D2-4

請求回収の業務フロー

[20min]

経理のヒアリングメモから、手作業と例外のある業務フローを描きます。

ヒアリングメモ_04 の中身

経理の山本さんに請求から入金確認までを聞いたメモです

メモの見出し	中身
請求書を作る	月末に請求書を作り、承認を経てお客様に送るまで
入金の確認	毎朝の入金明細と請求一覧の突き合わせ
金額や名前が合わないとき	請求と入金的一致しない場合の扱い
督促	支払期日を過ぎた入金の確認と督促
月次の報告	売掛金の残高を課長に出すまで

発展課題で使うデータ

資料/データ/請求回収/ の 請求一覧_202608.csv と 入金明細_202609.csv

業務フローを AI に描かせるときの渡し方

人と帳票を先に挙げさせてから図にします

1

**登場する人と帳票を
列挙させる**

メモに出てくる人、部署、
帳票、ファイルを表にさせる

2

As-Is を描かせる

今のやり方を、人ごとの
列に分けて描かせる

3

To-Be を描かせる

手作業が減った箇所を、
図の横に書かせる

列挙を先にすると、図から抜けた人や帳票に気づけます。いきなり図を頼むと、メモに1回しか出てこない人や帳票が落ちやすくなります。

手作業と例外がどこにあるかを図で見ます

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。

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

区分	ファイル
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
発展で使う	資料/データ/請求回収/ の 請求一覧_202608.csv、入金明細_202609.csv
作る	成果物/day2/ の メモ_D2-4.md、業務フロー_請求回収.html

メモを読んで**手作業の箇所**に印を付けてから始めます

■ 自分の手で ■ AIと一緒に

STEP1
[5min]

STEP2
[8min]

STEP3
[5min]

STEP4
[2min]

■ **STEP1** **手作業の拾い出し**
[5min] 手で何かをしている箇所を3つ以上、行番号つきで書きます

■ **STEP2** **人と帳票の列挙と As-Is**
[8min] 人と帳票を一覧にさせてから、As-Is を描かせます

■ **STEP3** **To-Be の追加**
[5min] To-Be を足し、手作業が減った箇所を図の横に書かせます

■ **STEP4** **例外の確かめ**
[2min] 一部入金・手数料の差し引き・合算入金の分かれ道を数えます

列挙と作図を分けて頼みます

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。

<ここに STEP 1 のメモを貼る>

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・部署ごとに帯を分け、例外は分かれ道として描く

要点だけ載せています。全文はハンズオンガイドの STEP 2 からコピーします。

図に**例外の流れ**があるかを最後に確かめます

見る観点

- 自分が印を付けた手作業が、As-Is に全部あるか
- 例外が本筋と別の流れとして描かれているか
- To-Be で手作業が減った理由が書いてあるか

よくある落とし穴

- 本筋だけの、きれいな一本道の図になる
- As-Is にメモに無い作業が入る
- To-Be の案とメモにあることが色で分かれていない

設計書とモックのずれ

ずれは設計書の書き漏らしが見えた印です

ずれの種類	モックでの見え方	行き先
書いていない	AI が決めた形で画面ができている	決めて設計書に足す
書いたが画面で成り立たない	項目が並ばない、チェックの順が合わない	設計書を直す
書いたとおりだが使いにくい	入力の手間が増える、見比べにくい	要確認にして相手に聞く

今日見つけたずれは、Day3 で別の設計書をレビューするときの観点になります。

設計書に無い表示位置でエラーが出たとき最初にすることは

- A モックを消して作り直す
- B そのまま次の演習に進む
- C 表示位置を決めて設計書に書き、v0.2 に戻す
- D AI に同じ文面をもう一度送る

まず自分で考えてみてください。答えはチャットに記号だけ書いてください。

要件ID の紐づけが無い画面が見つかったときの扱いは

- A 要件定義書か画面一覧の抜けとして確かめる
- B 問題はないので次に進む
- C 区分を新設に変える
- D 画面遷移の図から消す

まず自分で考えてみてください。答えはチャットに記号だけ書いてください。

答えはC・Aの順です

Q1 C 表示位置を決めて設計書に書き、v0.2 に戻す

モックで見えた差は、設計書の書き漏らしです。
決めた内容を設計書に戻すと、次に作る人も同じ画面を作れます。

Q2 A 要件定義書か画面一覧の抜けとして確かめる

要件の無い画面は、要件定義書の抜けか、画面一覧の区分の誤りのどちらかです。
消す前に理由を確かめます。

影響する画面を一覧に並べてから書く 設計書はモックにして確かめる

入力チェックは単項目・相関・業務ロジックに分け、全てのチェックにメッセージID を付けました。

今日の成果物

画面一覧、画面遷移、画面設計書_KIN030_v0.2.xlsx、モック、請求回収の業務フロー。
どれも 成果物/day2/ にあります。

Day3 の予告

Day3 は別チームが書いた設計書をレビューします

Day3 設計レビューと収支ダッシュボード

2026年10月22日（木）

演習	中身
D3-1	観点を分けた設計レビュー
D3-2	要件と画面の対応表
D3-3	収支ダッシュボードの作成
D3-4	成果物のまとめ

Day3 で使う今日の成果物

画面一覧.md

画面設計書_KIN030_v0.2.md

画面設計書_KIN030_v0.2.xlsx

終わらなかった方には、Day3 の始めに合流版を配ります。

質疑とアンケート

ご質問はチャットか音声でどうぞ

アンケートの URL は事務局からチャットでお送りします

いただいた回答は Day3 の準備に使います。

今日作った 成果物/day2/ のファイルは、消さずにそのまま残してください。



株式会社ギブリー

〒150-0036 東京都渋谷区南平台町15-13 帝都渋谷ビル8F

<https://givery.co.jp/>

Day3 でお会いしましょう。