



# AIと一緒に設計・開発！ChatGPTで 基本設計から実装までを体験しよう

## 1回目

2025/12/02

Kazuma Sekiguchi

# 自己紹介



関口和真

株式会社コムセントCTO  
Webシステム開発、スマートフォンアプリ制作、  
サーバー構築、運用など

スマートフォンを使った動画配信アプリの制作  
サーバーサイドシステムの作成  
フロントエンド部分の作成  
AI周りのいろいろ  
最近は仕様書とか書き物が多い

# 今回のAgenda

- AIを使って「基本設計書」を自動生成する
- AI活用の概要と注意点
- 基本設計に必要な要素の解説
- AIへの指示（プロンプト設計）の方法
- 実践：AIで基本設計を作成する

# 設計書とは

- ・ 設計書はシステム開発やソフトウェア開発の現場で、重要な成果物の一つ
  - ・ エンジニア・デザイナー・顧客の間で共通認識を持つために作成
- ・ システムやソフトウェアが「どのように動作するか」「何を実現するか」を明文化したドキュメント
  - ・ 設計書は実務においては仕様書と同義
    - ・ 仕様書だと明確なドキュメントを指定できていないので、少しほんやりした概念ではある
    - ・ 設計書は基本設計書、詳細設計書に大別できる

# 要件定義書

- ・顧客のニーズや実現したいことを文章としてまとめたものが要件定義書
  - ・現状のシステムや問題点、改善したいところをヒアリングした上で、システム化することで実現したいことができるかどうかを判断しつつまとめていく
  - ・クライアントからの要望を聞く能力、システムに落とし込むことができるかどうかの判断力、クライアントが真に欲しいものを想像する能力などが要求される
    - ・クライアントが明確にITに詳しいとは限らないため、きちんと言語化されるとは限らない
    - ・クライアントが実現したいといっているものが最適とは限らない

# 要件定義書

- ネットを通じてお問い合わせや予約が24時間いつでも取れるようしたい、というのが要件
  - 実際には要件定義書では他にも書かないといけない部分が多いが、要望を文書にまとめ上げたもの、となる
- 要件定義の段階でシステムの目的を失うことがあるため、きちんとシステム化する部分の範囲や目的を明確化することが重要
  - 範囲を広げすぎて、あれもこれもと盛り込むとシステム化に失敗する確率が上がっていく
  - ヒアリングや文字化する能力が必要なため、AIに代替しづらい部分

# RFP（提案依頼書）

- ・発注者がベンダー（システム制作会社など）に対して「このようなシステムが欲しい」と提示する文書
  - ・複数の提案を比較検討するために用いる
  - ・発注者の要望に対して、「こう作成します」と仕様書を作成して、発注者に返す
  - ・大枠の機能や目的を提示するのが一般的
- ・要件定義を発注者がある程度済ませている
  - ・実際には受注後に改めて要件定義をして、範囲を絞り込むことが多い
    - ・場合によっては、システム対象が拡大することもある

# システム化できるのかできないのか

- 実例（に近い）
  - AIを使ったシステムを作成して欲しい
  - ネット上でユーザーがある操作をすることを通じて、ユーザーの評価を行ない、スキルを可視化するシステムの作成
  - 社内などで社員スキル評価を計測するシステムを提供する
  - 10ヶ月でシステムを作成
- 結果
  - お断りしました
  - 評価軸が極めて曖昧である、非明示的なものを明示的なスキルという評価に落とし込むための変換式が不明確（それを作成するタスクだった模様）
  - AIに数値化または文字化できないものを伝達することは現状不可能
- 不明確であり、策定しなければいけない内容が多すぎて、担保できない
- システムに落とし込むには、考えるべき事項が多すぎる

# 設計書

- 基本設計書
  - 外部設計書
    - どのような仕組みでシステムを作成するのかの全体像を記述する
- 詳細設計書
  - 内部設計書
    - 基本設計を分解し、どう実装するかを具体的に決定して明文化
- ほかに機能仕様書（機能の説明）、画面設計書（UIや画面遷移）、データベース設計書（DB構造）などが存在

# 設計書

- ・開発者が必要な情報を漏れなく取得できる形にする
  - ・入力項目では入力形式・必須／任意・最大文字数などを定義し、バリデーションは入力値のチェック条件を記述
  - ・エラーを想定し、エラーが起きたときにどのような処理を行うのかも明示しておく
- ・システムの挙動や処理内容を決める「業務上の決まりごと」を業務ルールとして、処理内容などを記述することもある
- ・どのような画面を遷移して操作を行うのかを明確にするための画面遷移図や業務フロー図などの補足図を用いる

# AIによる設計書作成

- AIは「初稿生成」と「修正指示」に強い
  - 特に基本設計の文章化が劇的に効率化される
- ある程度要件定義書を与えて、基本設計書に落とし込むよう指示をすれば、基本設計書が作成できる
  - 図を作成するように指示すれば、必要な図を作成して貰うことも可能
  - 図はそのまま使うならPNG形式を、編集して使うなら、SVGとして出力させると便利
  - ただし無料版のChatGPTだと図は相当変になる
    - ChatGPT PlusなどのChatGPT-5.1でPNG指定すればまとも

# AIとの役割分担

- AIは以下の内容は高速に実現可能
  - 文章化（初稿の高速生成）
  - 要件の整理
  - 漏れの指摘
  - 基本設計書の標準構成を提案
  - 画面遷移図・業務フロー図の初期ドラフト
  - 実装コードのサンプル生成
- つまりある程度要件をまとめ上げることができれば、AIに通すことで、ある程度の文書化+図の作成は可能

# 人がやるべき内容

- ・この辺りは文書化することがAIには不可能
  - ・顧客の真意を汲み取ること（要件定義）
  - ・曖昧な仕様の意思決定
  - ・業務ルールの策定
  - ・セキュリティ要件・性能要件の確定
  - ・法的な制約の判断（個人情報、著作権、契約など）
  - ・不明点の仮決めとリスク管理
  - ・開発スケジュール

この辺りは個別にAIに  
聞けば分かることが多い

# AIによる誤解

- AIはもっともらしい嘘を付く
  - 分からないということをAIは言わないので、何となく正しいような内容を提示することが多々ある
  - 特に不明確な部分を不明確である、と指摘してくることは少ないため、不明確な部分は適当に補われるため、意図と異なる内容になることが多い
  - 必ず提示された内容を確認し、意図した通りの内容になっているかどうかを精査する必要がある
  - 意図と異なっている場合は、指摘して、不明確な点を補って方向修正をして貰う

# AIによる誤解

- ・「ここを修正して」と指示をしたとしても、他の箇所も併せて修正することが非常に多い
  - ・他の箇所は正しい内容であったにも関わらず、正しく無い内容に変化していることもあるため、毎回確認が必要
  - ・変化していないかどうかを確認するならDIFF（差分を取るためのツール）などを使って確認する

# AIによる失敗

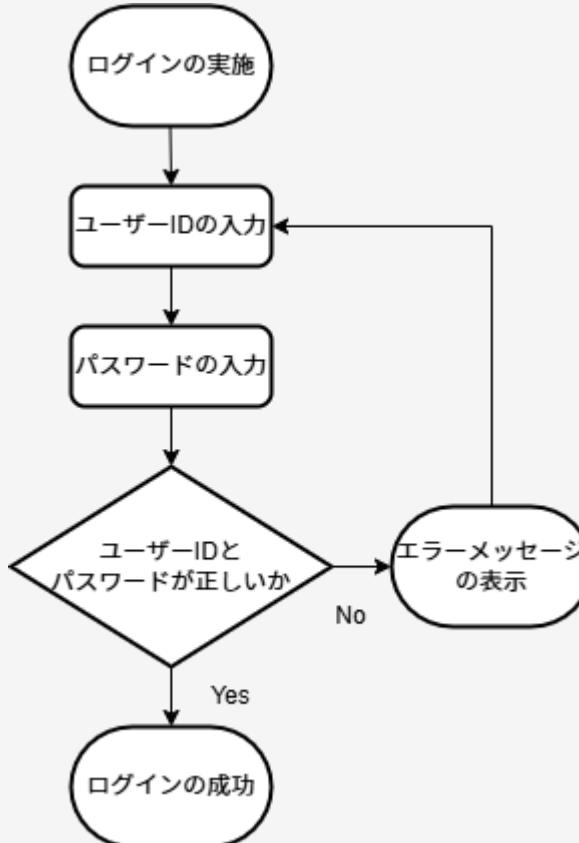
- ・要件定義を作らないで聞いたままの内容をAIに投げ込むと目的と異なるシステムとなってしまう可能性が高い
  - ・整合性の取れていない文章を入れると、ズレが激しくなる
- ・要件定義を確定して、目的を明確にした上で、AIに依頼をすることでズレを防ぐことができる
- ・複雑な業務ルールや暗黙知は理解できない
  - ・その会社独自のルールなどは、きちんと説明しないとAIには理解不能
- ・判定基準が曖昧な評価系は難しい
  - ・判定基準を明確化しないと、かなり怪しい文章を出す
- ・セキュリティ要件の抜け漏れが多い（人的レビュー必須）
  - ・どこまでを確保するべきかなどの線引きはAIには不可能
- ・実際のインフラ・負荷試験・運用はAIでは設計不可

# AIの利用・プロンプト

- 明確な要件定義を作成しておく
  - 要件定義作成時点で壁打ち的にAIを使うのはあり
- 作成した要件定義を投げ込んで、プロンプトを指定
  - 基本設計書のドラフトを作成することを明示
  - 出力して欲しい項目を明示する
    - 例：システム概要、画面一覧、機能一覧、非機能要件、想定されるリスクと確認事項など
  - 不明確な点や要件定義で抜けている点を指摘するように記述
- 図は基本設計書のドラフトを出力させてから、改めてプロンプトを投げて、図を作成させた方が良い

# 補足図

- 良く使われるのは2種類
  - フローチャート
    - 処理や手順を記号と矢印で表す
  - ER図
    - データ構造と関係性を示す
    - 通常データベースの設計を示す



inquiries			
	id	PK	問い合わせID
varchar	customer_name		顧客名
varchar	email		メールアドレス
varchar	subject		件名
text	body		本文
varchar	company		会社名
varchar	department		部署名
varchar	phone		電話番号
varchar	source		受領経路(web/phone等)
varchar	status		ステータス(未対応/対応中/完了)
datetime	created_at		
datetime	updated_at		

inquiry_status_history			
	id	PK	
int	inquiry_id	FK	お問い合わせID
varchar	status		変更後ステータス
varchar	updated_by		変更者社員番号
datetime	updated_at		

ありがとうございました。  
また次回。