

注文管理システム 仕様書

この資料は、Next.js フルスタック学習パスの卒業課題として使用します。Todoアプリチュートリアルで学んだ知識を応用し、より実践的なシステム開発を体験できます。

ドキュメント構成

ファイル	内容
README.md (本ファイル)	システム仕様書（機能、API設計、バリデーション）
database.md	データベース設計（ER図、テーブル定義、SQL）
implementation-guide.md	実装ガイド（ステップバイステップの作業手順）
mockups/	モック画面（完成イメージのHTML）

概要

飲食店や小売店向けのシンプルな注文管理システムです。顧客情報、商品情報、注文情報を管理し、CRUD操作に加えてリレーション（テーブル間の関連付け）や集計機能を学習できます。

学習目標

Todoアプリからのステップアップとして、以下を習得します：

項目	Todoアプリ	注文管理システム
テーブル数	1テーブル	4テーブル
リレーション	なし	外部キー、1対多、多対多
データ操作	単純なCRUD	JOINを使った複合クエリ
ステータス管理	完了/未完了	複数ステータス（受付→調理中→完了→キャンセル）
集計機能	なし	売上集計、期間検索
入力項目	1項目（タイトル）	複数項目（バリデーション強化）

主な機能

顧客管理機能

- 顧客の新規登録
- 顧客一覧の表示・検索
- 顧客情報の編集
- 顧客の削除（注文履歴がある場合は削除不可）

商品管理機能

- 商品の新規登録（名前、価格、カテゴリ、在庫状況）
- 商品一覧の表示・検索
- 商品情報の編集
- 商品の削除（注文履歴がある場合は削除不可）
- 販売中/販売停止の切り替え

注文管理機能

- 新規注文の作成（顧客選択、商品選択、数量指定）
- 注文一覧の表示・検索
- 注文詳細の表示
- 注文ステータスの更新（受付→調理中→完了）

- 注文のキャンセル
- 注文履歴の確認

ダッシュボード機能

- 本日の注文件数・売上表示
- 注文ステータス別の件数表示
- 最近の注文一覧

画面一覧

画面名	説明
ダッシュボード	メイン画面。本日の売上サマリー、最近の注文一覧
顧客一覧画面	顧客の一覧表示、検索、新規登録ボタン
顧客登録/編集画面	顧客情報の入力フォーム
商品一覧画面	商品の一覧表示、カテゴリ絞り込み、新規登録ボタン
商品登録/編集画面	商品情報の入力フォーム
注文一覧画面	注文の一覧表示、ステータス・期間での絞り込み
注文作成画面	顧客選択、商品選択、数量入力
注文詳細画面	注文内容の詳細表示、ステータス更新ボタン

画面イメージ

ダッシュボード

注文管理システム

[顧客] [商品] [注文] [設定]

本日の注文 15件	本日の売上 ¥45,600	未完了注文 3件
--------------	------------------	-------------

ステータス別
[受付: 2] [調理中: 1] [完了: 12] [キャンセル: 0]

最近の注文

[+ 新規注文]

#1015 山田太郎 ¥2,400 調理中 12:30 [詳細]
#1014 佐藤花子 ¥1,800 受付 12:25 [詳細]
#1013 田中一郎 ¥3,200 完了 12:15 [詳細]

注文作成画面

新規注文作成

[戻る]

顧客選択

山田太郎 ▼

商品を追加

商品名	単価	数量	小計
カレーライス	¥800	[2]	¥1,600
サラダセット	¥400	[1]	¥400

[+ 商品を追加]

合計: ¥2,000

[キャンセル] [注文を確定する]

API設計 (REST API)

顧客API

メソッド	エンドポイント	説明
GET	/api/customers	顧客一覧取得
GET	/api/customers/{id}	顧客詳細取得
POST	/api/customers	顧客登録
PUT	/api/customers/{id}	顧客更新
DELETE	/api/customers/{id}	顧客削除

商品API

メソッド	エンドポイント	説明
GET	/api/products	商品一覧取得
GET	/api/products/{id}	商品詳細取得
POST	/api/products	商品登録
PUT	/api/products/{id}	商品更新
DELETE	/api/products/{id}	商品削除

注文API

メソッド	エンドポイント	説明
GET	/api/orders	注文一覧取得
GET	/api/orders/{id}	注文詳細取得（明細含む）
POST	/api/orders	注文作成
PUT	/api/orders/{id}/status	注文ステータス更新
DELETE	/api/orders/{id}	注文キャンセル

ダッシュボードAPI

メソッド	エンドポイント	説明
GET	/api/dashboard/summary	本日のサマリー取得
GET	/api/dashboard/recent-orders	最近の注文取得

リクエスト/レスポンス例

顧客一覧取得

```
GET /api/customers?search=山田
```

レスポンス:

```
{
  "customers": [
    {
      "id": 1,
      "name": "山田太郎",
      "email": "yamada@example.com",
      "phone": "090-1234-5678",
      "created_at": "2025-01-01T10:00:00Z"
    }
  ],
  "total": 1
}
```

商品一覧取得

```
GET /api/products?category=food&available=true
```

レスポンス:

```
{
  "products": [
    {
      "id": 1,
      "name": "カレーライス",
      "price": 800,
      "category": "food",
      "is_available": true,
      "created_at": "2025-01-01T10:00:00Z"
    },
    {
      "id": 2,
      "name": "サラダセット",
      "price": 400,
      "category": "food",
      "is_available": true,
      "created_at": "2025-01-01T10:00:00Z"
    }
  ],
  "total": 2
}
```

注文作成

```
POST /api/orders
Content-Type: application/json

{
  "customer_id": 1,
  "items": [
    { "product_id": 1, "quantity": 2 },
    { "product_id": 2, "quantity": 1 }
  ]
}
```

レスポンス:

```
{  
  "id": 1015,  
  "customer": {  
    "id": 1,  
    "name": "山田太郎"  
  },  
  "items": [  
    {  
      "product_id": 1,  
      "product_name": "カレーライス",  
      "unit_price": 800,  
      "quantity": 2,  
      "subtotal": 1600  
    },  
    {  
      "product_id": 2,  
      "product_name": "サラダセット",  
      "unit_price": 400,  
      "quantity": 1,  
      "subtotal": 400  
    }  
],  
  "total_amount": 2000,  
  "status": "pending",  
  "created_at": "2025-01-08T12:30:00Z"  
}
```

注文ステータス更新

```
PUT /api/orders/1015/status  
Content-Type: application/json  
  
{  
  "status": "cooking"  
}
```

ダッシュボードサマリー取得

```
GET /api/dashboard/summary
```

レスポンス:

```
{  
    "today_orders": 15,  
    "today_sales": 45600,  
    "pending_orders": 3,  
    "status_summary": {  
        "pending": 2,  
        "cooking": 1,  
        "completed": 12,  
        "cancelled": 0  
    }  
}
```

バリデーションルール

顧客

項目	ルール
名前	必須、1~100文字
メールアドレス	任意、有効なメール形式、一意
電話番号	任意、10~15文字（ハイフン含む）

商品

項目	ルール
商品名	必須、1~100文字
価格	必須、1~999999の整数
カテゴリ	必須、food/drink/otherのいずれか

注文

項目	ルール
顧客ID	必須、存在する顧客
商品リスト	必須、1件以上
数量	必須、1~99の整数

注文ステータス

ステータス	英語名	説明	次のステータス
受付	pending	注文を受け付けた状態	cooking, cancelled
調理中	cooking	調理・準備中	completed, cancelled
完了	completed	提供完了	- (終了状態)
キャンセル	cancelled	注文取消	- (終了状態)

エラーメッセージ例

状況	メッセージ
名前未入力	名前を入力してください
価格が不正	価格は1~999999の数値で入力してください
顧客が見つからない	指定された顧客は存在しません
商品が見つからない	指定された商品は存在しません
注文に商品がない	1つ以上の商品を選択してください
削除不可（関連データあり）	注文履歴があるため削除できません
ステータス遷移不可	この注文のステータスは変更できません

推奨チュートリアル構成

ステップ	内容	学習ポイント
1	環境構築	プロジェクト作成、Docker Compose設定
2	データベース設計	複数テーブル作成、リレーション設定
3	商品管理API	基本的なCRUD APIの実装
4	商品管理画面	一覧・登録・編集・削除の画面実装
5	顧客管理API	CRUD API、検索機能の実装
6	顧客管理画面	フォームバリデーション強化
7	注文作成API	複数テーブルへの同時書き込み
8	注文作成画面	動的フォーム、合計計算
9	注文一覧・詳細	JOIN、ステータス管理
10	ダッシュボード	集計クエリ、グラフ表示
11	仕上げ	エラー処理、UI改善、デプロイ

技術スタック（推奨）

この課題は学習パスで学んだ技術を活用します：

レイヤー	技術
フロントエンド	Next.js (App Router) + TypeScript
スタイル	Tailwind CSS
バックエンド	Python + FastAPI
ORM	SQLModel
データベース	PostgreSQL
環境構築	Docker Compose

設計方針

含める機能

- 複数テーブルのリレーション
- ステータス管理
- 検索・フィルタリング
- 基本的な集計（日別売上）
- フォームバリデーション

含めない機能（スコープ外）

- ユーザー認証・ログイン
- 権限管理（管理者/スタッフ）
- 在庫管理（数量追跡）
- 支払い処理
- 印刷機能（レシート等）
- 通知機能（メール/プッシュ）
- 多店舗対応
- 予約管理

対象レベル

Todoアプリチュートリアルを完了した方向け。複数テーブルを使ったリレーションナルデータベース設計と、より実践的なWebアプリケーション開発を学習できます。

完成後のスキル

この課題を完了すると、以下のスキルが身につきます：

- 複数テーブルを持つデータベース設計
- 外部キー制約とリレーションの理解
- JOINを使った複合クエリの作成
- ステータス管理を含むビジネスロジックの実装

- 動的なフォーム（明細行の追加・削除）の実装
- 集計機能とダッシュボード画面の作成
- より複雑なAPIエンドポイントの設計