

cafeMuji 注文管理システム 概要説明書

1. システム概要

1.1 システム名

cafeMuji 注文管理システム

1.2 開発目的

cafe&meal MUJIの業務効率化を目的とした注文管理用Webアプリケーションの開発

1.3 システムの特徴

- 役割別UI設計: レジ、キッチン、各担当者用の専用インターフェース
- リアルタイム更新: 注文状態の部分更新による効率的な業務管理
- 多商品対応: フード、アイスクリーム、かき氷の統合管理
- クラウド対応: Renderを利用した外部アクセス可能なシステム

2. システム構成

2.1 アーキテクチャ

- フレームワーク: Django 5.2.1
- データベース: SQLite（開発） / PostgreSQL（本番推奨）
- Webサーバー: gunicorn
- デプロイ環境: Render.com

2.2 主要コンポーネント

cafeMuji/	
├── food/	# フード注文管理
├── ice/	# アイスクリーム注文管理
├── shavedice/	# かき氷注文管理
├── mobile/	# モバイル対応
├── common/	# 共通機能
└── config/	# プロジェクト設定

3. 主要機能

3.1 フード注文管理（food）

- メニュー: からあげ丼、ルーロー飯

- 注文管理: 仮注文→本注文のワークフロー
- 状態管理: 受注→作成中→完了の追跡
- クリップ管理: 色・番号による注文識別

3.2 アイスクリーム注文管理 (ice)

- サイズ: シングル/ダブル
- 容器: カップ/コーン
- フレーバー: 12種類 (ジャージー牛乳、お茶、マンゴー等)
- 注文状態: OK/STOP/保留の管理

3.3 かき氷注文管理 (shavedice)

- フレーバー: 抹茶、いちご、ゆず
- 注文状態管理
- クリップ色による識別

3.4 共通機能

- セッション管理による仮注文の一時保存
- 注文グループ管理 (複数商品の一括注文)
- 完了時刻の自動記録
- 備考欄による詳細情報の管理

4. 技術的特徴

4.1 データベース設計

- 正規化されたテーブル構造
- 適切なインデックス設計
- 外部キー制約によるデータ整合性

4.2 セッション管理

- Django標準セッション機能の活用
- 仮注文データの一時保存
- セキュリティを考慮した実装

4.3 レスポンシブ対応

- モバイル端末での利用を考慮
- タッチ操作に最適化されたUI

- 画面サイズに応じたレイアウト調整

5. 運用環境

5.1 開発環境

- OS: macOS
- Python: 3.x
- Django: 5.2.1
- データベース: SQLite

5.2 本番環境

- ホスティング: Render.com
- データベース: SQLite（本番ではPostgreSQL推奨）
- 静的ファイル: Render CDN

6. セキュリティ対策

6.1 認証・認可

- Django標準認証システムの活用
- セッション管理による状態保持
- CSRF保護の実装

6.2 データ保護

- 入力値の妥当性チェック
- SQLインジェクション対策
- XSS対策

7. 今後の拡張性

7.1 機能拡張

- 在庫管理システムの統合
- 売上分析・レポート機能
- 顧客管理システムの連携

7.2 技術的拡張

- マイクロサービス化
 - API化による他システム連携
 - リアルタイム通信（WebSocket）の導入
-

作成日: 2025年8月 **作成者:** 村岡 優次郎 **バージョン:** 1.0