診療ウェブ予約システム  
プログラム設計書

第2.0版

株式会社○○○○

目次

[1. 本書の位置付け 3](#__RefHeading___Toc18225_1494168553)

[2. 機能仕様 3](#__RefHeading___Toc18227_1494168553)

[3. ソフトウェア構成 3](#__RefHeading___Toc18229_1494168553)

[3.1 バックエンド 3](#__RefHeading___Toc18231_1494168553)

[3.2 フロントエンド 3](#__RefHeading___Toc18233_1494168553)

[4. ファイル構成 4](#__RefHeading___Toc18235_1494168553)

[5. クラス構成 6](#__RefHeading___Toc18237_1494168553)

[5.1 エンティティ（domain） 6](#__RefHeading___Toc18241_1494168553)

[5.2 DTO（dtoパッケージ） 8](#__RefHeading___Toc14674_3684750480)

[5.3 リポジトリ（repository: Mapperインターフェース） 8](#__RefHeading___Toc14676_3684750480)

[5.4 サービス（serviceパッケージ） 10](#__RefHeading___Toc14678_3684750480)

[5.5 コントローラー（controllerパッケージ） 11](#__RefHeading___Toc14680_3684750480)

[6. 設計思想・技術的特徴 12](#__RefHeading___Toc14978_3684750480)

[7. 改訂履歴 13](#__RefHeading___Toc18247_1494168553)

# 本書の位置付け

本書は、診療ウェブ予約システム（以下、「本システム」）に関するプログラムの設計を記す。

# 機能仕様

機能仕様については、以下のドキュメントを参照のこと：

* システム機能一覧
* システム画面設計書
* システムデータ設計書

# ソフトウェア構成

以下に、本システムのソフトウェア構成を示す。

## バックエンド

Java/Spring Bootを基盤にしたWebアプリケーション。

データベースはMySQLを使用し、MyBatisでデータアクセス処理を実装。

サーバーは組み込み型Tomcat（Spring Boot標準）。

ユーザー認証・権限管理はSpring Security＋BCryptで実装。

業務ロジックをサービス層（Serviceクラス）に集約し、コントローラ層（Controllerクラス）で画面/リクエスト毎にエンドポイントを分担。

ログインや予約操作など、全ての処理がREST形式のコントローラーおよびThymeleaf画面に接続される。

## フロントエンド

HTML/CSS/JavaScriptで構築し、Thymeleafを使ってサーバーサイドで動的ページ生成。

デザインはBootstrap＋main.cssなどのカスタムCSSを利用し、スマホ対応・統一感のあるUIを実現。

画面の共通部分（ヘッダーやメッセージ表示）はThymeleafフラグメント（parts/common.html など）で管理。

フォームバリデーションや入力支援もサーバー側・クライアント側で実装。

# ファイル構成

以下に本システムのファイル構成を示す。

src/main/

├─java/com/unknownclinic/appointment/

│ │ ClinicAppointmentSystemApplication.java

│ │

│ ├─config/

│ │ SecurityConfig.java

│ │

│ ├─controller/

│ │ AdminBookingController.java

│ │ AdminBusinessDayController.java

│ │ AdminLoginController.java

│ │ BookingController.java

│ │ LoginController.java

│ │ MyPageController.java

│ │ PasswordResetController.java

│ │ RegisterController.java

│ │

│ ├─domain/

│ │ Admin.java

│ │ Booking.java

│ │ BusinessDay.java

│ │ TimeSlot.java

│ │ User.java

│ │

│ ├─dto/

│ │ AdminBookingView.java

│ │ AdminBusinessDayView.java

│ │ BookingView.java

│ │ RegisterForm.java

│ │ TimeSlotView.java

│ │

│ ├─repository/

│ │ AdminMapper.java

│ │ BookingMapper.java

│ │ BusinessDayMapper.java

│ │ TimeSlotMapper.java

│ │ UserMapper.java

│ │

│ └─service/

│ AdminBookingService.java

│ AdminUserDetailsService.java

│ BookingService.java

│ BookingServiceImpl.java

│ BusinessDayService.java

│ BusinessDayServiceImpl.java

│ NameSanitizer.java

│ PatientUserDetailsService.java

│ TimeSlotService.java

│ TimeSlotServiceImpl.java

│ UserService.java

│

│

│

│

│　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　次ページへ続く

└─resources/

│ application.properties

│

├─mapper/

│ BookingMapper.xml

│ BusinessDayMapper.xml

│ TimeSlotMapper.xml

│ UserMapper.xml

│

├─static/

│ └─css/

│ auth.css

│ bootstrap.min.css

│ main.css

│

└─templates/

│ confirm.html

│ login.html

│ main.html

│ mypage.html

│ register.html

│ reset-password.html

│

├─admin/

│ business-days.html

│ login.html

│ main.html

│

└─parts/

admin-common.html

auth-common.html

common.html

# クラス構成

## エンティティ（domain）

以下に各クラスの詳細を示す。

****【User】****

| フィールド名 | 型 | 説明 |
| --- | --- | --- |
| id | Long | ユーザーID |
| cardNumber | String | 診察券番号 |
| birthday | LocalDate | 生年月日 |
| name | String | 氏名 |
| phoneNumber | String | 電話番号 |
| isTemporary | Boolean | 仮診察券フラグ |
| password | String | パスワード（ハッシュ） |

**【Admin】**

| フィールド名 | 型 | 説明 |
| --- | --- | --- |
| id | Integer | 管理者ID |
| loginId | String | ログインID |
| password | String | パスワード（ハッシュ） |

**【Booking】**

| フィールド名 | 型 | 説明 |
| --- | --- | --- |
| id | Long | 予約ID |
| userId | Long | ユーザーID（外部キー） |
| date | LocalDate | 予約日 |
| timeSlotId | Long | 時間枠ID（外部キー） |
| status | String | 予約ステータス |

**【BusinessDay】**

| フィールド名 | 型 | 説明 |
| --- | --- | --- |
| id | Long | 営業日ID |
| businessDate | LocalDate | 営業日 |
| isActive | Boolean | 営業日有効フラグ |
| createdAt | LocalDateTime | 登録日時 |

**【TimeSlot】**

| フィールド名 | 型 | 説明 |
| --- | --- | --- |
| id | Long | 時間枠ID |
| businessDayId | Long | 営業日ID（外部キー） |
| slotType | String | 時間帯区分（例: null=終日, am, pm） |
| isAvailable | Boolean | 予約可否 |
| createdAt | LocalDateTime | 登録日時 |

※ 各エンティティはgetter/setterを備えています。

## ****DTO**（**dtoパッケ****ージ）

| **クラス名** | **用途** |
| --- | --- |
| TimeSlotView | 営業日・時間枠表示用DTO |
| BookingView | 患者マイページ・予約一覧DTO |
| AdminBookingView | 管理画面用 予約一覧表示DTO |

## ****リポジト**リ（**repository: Mapperインターフェー****ス）

****UserMapper****

| **メソッド名** | **説明** |
| --- | --- |
| User findByCardNumberAndBirthday(String, LocalDate) | 診察券番号と生年月日でユーザー検索 |
| User findById(Long) | ユーザーIDで取得 |
| void insert(User) | ユーザー新規登録 |
| List<User> findAll() | 全ユーザー取得 |

****BookingMapper****

| **メソッド名** | **説明** |
| --- | --- |
| List<Booking> findByUserId(Long) | ユーザーIDから予約履歴取得 |
| Booking findById(Long) | 予約IDで取得 |
| void insert(Booking) | 新規予約登録 |
| void update(Booking) | 予約内容更新（キャンセル時など） |
| List<Booking> findByDate(LocalDate) | 指定日の予約一覧取得（管理者用） |

****AdminMapper****

| **メソッド名** | **説明** |
| --- | --- |
| Admin findByLoginId(String) | ログインIDで管理者取得 |

****BusinessDayMapper****

| **メソッド名** | **説明** |
| --- | --- |
| List<BusinessDay> findAll() | 全営業日取得 |
| List<BusinessDay> findByMonth(YearMonth) | 指定月の営業日取得 |
| void insert(BusinessDay) | 営業日新規登録 |
| void update(BusinessDay) | 営業日情報更新 |

****TimeSlotMapper****

| **メソッド名** | **説明** |
| --- | --- |
| List<TimeSlot> findByBusinessDayId(Long) | 営業日IDの時間枠一覧取得 |
| void insert(TimeSlot) | 時間枠新規登録 |
| void update(TimeSlot) | 時間枠情報更新 |
| void deleteByBusinessDayId(Long) | 指定営業日の時間枠を全削除（再設定用） |

## ****サービ**ス（**serviceパッケー****ジ）

| **サービス名** | **主なメソッド（引数）** | **概要** |
| --- | --- | --- |
| UserService | login(String, LocalDate) | ユーザー認証 |
|  | registerTemporaryUser(String, LocalDate, String) | 仮登録 |
|  | findUserById(Long) | ユーザーIDで取得 |
| BookingService | createBooking(Long, LocalDate, Long) | 予約作成 |
|  | getAvailableTimeSlots(LocalDate) | 空き時間枠取得 |
|  | getUserBookings(Long) | 予約履歴取得 |
|  | cancelBooking(Long, Long) | 予約キャンセル |
|  | getBookingById(Long) | 予約詳細取得 |
| AdminService | login(String, String) | 管理者認証 |
| AdminBookingService | getBookingsByDate(LocalDate) | 指定日の全予約取得 |
|  | getBookingsByDateAndSlot(LocalDate, Long) | 指定日・時間枠ごとの予約取得 |
| BusinessDayService | getBusinessDaysInMonth(YearMonth) | 月単位の営業日取得 |
|  | updateBusinessDays(List<BusinessDay>) | 営業日情報更新 |
| TimeSlotService | getAllTimeSlots() | 全時間枠取得 |
|  | getTimeSlotById(Long) | 時間枠IDで取得 |
|  | updateTimeSlots(List<TimeSlot>) | 時間枠情報更新 |

## ****コントローラ**ー（**controllerパッケー****ジ）

| **コントローラー名** | **メソッド/対応画面・機能** |
| --- | --- |
| LoginController | showLoginForm()：ログイン画面表示 |
|  | processLogin(String, LocalDate)：ユーザー認証 |
| RegisterController | showRegisterForm()：仮登録画面表示 |
|  | processRegister(UserForm)：仮登録処理 |
| BookingController | showBookingForm(Model)：予約画面表示（/main） |
|  | selectTimeSlot(LocalDate, Long)：時間枠選択→確認画面 |
|  | showConfirmPage()：予約確認画面表示（/confirm） |
|  | submitBooking()：予約確定 |
|  | cancelConfirm()：確認キャンセル |
| MypageController | showMypage()：マイページ表示 |
|  | cancelBooking(Long)：予約キャンセル |
| AdminLoginController | showAdminLoginForm()：管理者ログイン画面表示 |
|  | processAdminLogin(String, String)：管理者認証 |
| AdminBookingController | showAdminBookingList(LocalDate)：指定日の予約一覧 |
| AdminBusinessDayController | showBusinessDaySettings(YearMonth)：営業日設定画面表示 |
|  | updateBusinessDays(List<LocalDate>)：営業日設定保存 |

# 設計思想・技術的特徴

・レイヤー構造の分離

業務ロジック、データアクセス、画面制御、データ表現（DTO）を明確に分けることで、保守性や拡張性を高める設計とする。

・DTO設計の導入

画面表示やレスポンスの要件に応じて、エンティティとは独立したDTO（データ転送オブジェクト）を用意し、各レイヤ間の依存を最小限に抑える。

・データベース設計の簡素化

冗長なテーブルや複雑なリレーションを避け、シンプルなテーブル設計により理解しやすく保守しやすいシステム構造とする。

・セキュリティ設計

パスワードハッシュ化、認証・認可の分離設計を前提とし、今後の拡張や運用を見据えたセキュリティアーキテクチャとする。

・テンプレート再利用性の重視

共通パーツ（ヘッダー、フッター等）はテンプレートエンジン（Thymeleaf）のフラグメントで分割し、各画面で再利用できる設計とする。

・UI設計方針

フレームワーク（Bootstrap）＋オリジナルCSSの組み合わせで、統一感と拡張性のあるUIを設計段階から目指す。

・バリデーション設計

二重予約防止や入力チェックなど、業務要件に基づいた独自バリデーションを設計段階で明確化しておく。

# 改訂履歴

|  |  |  |  |
| --- | --- | --- | --- |
| 版 | 日付 | 改訂者 | 内容 |
| 1.0 | 2025/7/10 | yshtis | 初版作成 |
| 2.0 | 2025/8/15 | yshtis | 機能追加 |