1.要件定義書

1.1 システム概要

システム名称:クリニックオンライン予約システム (Clinic Online Reservation System: CORS)

目的:

- 患者が来院せずにオンラインで診療予約ができるシステムを構築する

- 医師・クリニックスタッフの予約管理業務を効率化する

- 予約の重複を防止し、待ち時間を削減する

対象ユーザー:

- 患者（クリニックの診療を希望する一般ユーザー）

- 医師（診療を行う医療従事者）

- 受付スタッフ（予約・来院管理を行うクリニックスタッフ）

- システム管理者（システム設定・メンテナンスを行う担当者）

1.2 機能要件

1.2.1 患者向け機能

* オンライン診療予約機能（診療科・医師・日時の選択）
* 予約確認・変更・キャンセル機能
* 会員登録・ログイン機能（メールアドレス・パスワード）
* 予約履歴の閲覧機能
* 通知機能（予約完了・リマインドメール送信・キャンセル発生）
* 家族アカウント登録・管理機能（家族の予約代行など）

1.2.2 医師向け機能

* 自身の業務スケジュール登録・確認機能
* 自身の診療スケジュール自動生成機能
* 予約一覧の閲覧・詳細表示機能（患者情報含む）
* 予約ステータス更新機能（診療完了、キャンセルなど）
* コメント・メモ登録機能（診療内容・注意点の記録用）

1.2.3 受付スタッフ向け機能

* 予約状況の一覧表示・検索機能
* 予約の登録・変更・キャンセル代行機能
* 来院受付処理機能（患者の来院チェックイン）
* 日別・週別の診療スケジュール表示機能
* 患者情報の閲覧・修正機能

1.2.4 システム管理機能

* ユーザー管理機能（患者・医師・スタッフのアカウント管理）
* 予約枠・診療時間帯の設定機能
* 診療科・医師情報の登録・編集機能
* 通知テンプレート管理機能（メール内容・送信タイミングの設定）
* ログ管理機能（操作履歴の記録・確認）

1.3 非機能要件

1.3.1 性能要件

* 平常時において、1秒以内のレスポンスで予約・参照操作が完了すること。
* 同時アクセスは最大100ユーザーを想定し、パフォーマンスを維持できる設計とする。
* 急患発生などの非常時には、通常予約を自動で数枠調整するロジックを組み込むことで、急患対応を実現する。

1.3.2 セキュリティ要件

* 通信経路はすべてSSL/TLSで暗号化し、第三者による盗聴や改ざんを防止する。
* ユーザー情報・医療情報は保存時にAES等の暗号方式で保護する。
* アクセス権限をユーザー種別（患者・医師・受付スタッフ・管理者）ごとに厳密に設定し、不正アクセスを防止する。
* 管理者画面へのアクセスは2段階認証を導入する。
* サイバー攻撃（SQLインジェクション、XSS、CSRFなど）への対策を実装し、脆弱性診断を定期的に実施する。

1.3.3 可用性要件

* 稼働率99.9%以上（年間ダウンタイム合計9時間未満）を目標とする。
* 定期メンテナンスは深夜帯に実施し、事前告知を徹底する。
* 災害時やシステム障害時には、フェイルオーバー構成によりサービスを迅速に再開できること。
* システムダウン時にも、ローカルキャッシュまたは印刷リスト等で最低限の予約情報（当日分）を参照可能とする。

1.3.4 拡張性要件

* 新しい診療科やクリニックの追加、予約枠の変更などが柔軟に行える設計とする。
* 今後のモバイルアプリ対応や他システム（電子カルテなど）との連携を見越したAPI設計とする。
* 急患対応など運用上の例外処理にも柔軟に対応できる設計とし、通常予約への影響を最小限に抑えるロジックを備える。

1.4 外部インターフェース要件

1.4.1 ユーザーインターフェース

* 患者向け画面：スマートフォン、タブレット、PCに対応したレスポンシブデザインを採用。
* 医師・受付スタッフ・管理者向け画面：PC向けブラウザ最適化、業務用途に適した操作性を考慮。
* 予約カレンダー：ドラッグ&ドロップで変更可能なインタラクティブUI。
* 文字サイズや色のコントラストは高齢者にも配慮したユニバーサルデザインを採用。

1.4.2 システムインターフェース

* 電子カルテシステムとの連携（将来的拡張を想定しAPIで対応）。
* メール／SMS配信システムとの連携（通知・リマインダー送信用）。
* 外部認証サービス（例：Google、LINE）との連携機能（患者の簡易ログインオプション）。
* バックアップサーバー・クラウドストレージとの定期連携（予約データ・ログの保存）。

1.4.3 ハードウェアインターフェース

* サーバー：クラウド環境（例：AWS, Azure）上に構築。
* 受付用端末：PCまたはタブレット、バーコードリーダー等の周辺機器との接続を想定。
* 医師の診察室端末：Webブラウザ経由でアクセス可能なPC端末を想定。
* オフライン対応端末：非常時にローカルキャッシュを保持するための簡易端末対応も考慮。

1.5 制約条件

* 医療関連法規（個人情報保護法、医療法など）に準拠した運用・管理を行うこと。
* 利用者のITリテラシーの差を考慮し、直感的でシンプルな操作性を実現すること。
* 患者情報・診療予約情報は国外に保存せず、国内のクラウドインフラ上で管理すること。
* システム改修・メンテナンスは原則として稼働時間外に実施すること。
* インターネット接続環境がない患者に対しては、電話予約などの代替手段を残すこと。
* 導入クリニックのネットワーク・端末環境によっては、パフォーマンス保証に制限がある場合がある。

1.6 環境要件

1.6.1 システム環境

* OS：Linuxベース（例：Ubuntu, Amazon Linux）
* Webサーバー：Nginx または Apache
* アプリケーション：Node.js / Python (Django/Flask) / Ruby on Rails のいずれか
* データベース：PostgreSQL または MySQL
* クラウド：AWS / Azure / GCP（高可用性・自動バックアップ対応）

1.6.2 クライアント環境

* 患者：
  + OS：iOS / Android / Windows / macOS
  + ブラウザ：Chrome, Safari, Edge（最新版）
* 医師・受付スタッフ・管理者：
  + OS：Windows 10 以降 / macOS
  + ブラウザ：Chrome または Edge（業務端末向け推奨）
  + 必要スペック：4GB以上のRAM、インターネット常時接続可能な端末