要件定義書

# プロジェクト名

タスク管理システム

# 1. 目的

本システムは、複数のプロジェクトとタスクを一元管理し、タスクの進捗状況を可視化することで、チームメンバー間の円滑なコミュニケーションを支援することを目的としています。各ユーザーの役割に応じた権限を付与し、プロジェクトおよびタスクの作成や管理が可能です。最終的には法人向けプロジェクト管理システムとして、クライアントの管理機能および拡張性を考慮した設計がされています。

# 2. スコープ

課題:  
• 複数のプロジェクトにおけるタスクの進捗状況が不透明で、担当者の負荷が見えづらい。  
• クライアントと一般ユーザー間のコミュニケーションが不足し、タスクのステータス更新が遅れることがある。  
  
解決方法:  
• ダッシュボードを使用して、プロジェクトおよびタスクの進捗状況を一元管理し、担当者やステータスの変更を迅速に行えるようにする。  
• クライアントとユーザー間でコメント機能を活用し、タスクレベルでのコミュニケーションを効率化する。

# 3. 要件

## 3.1 機能要件

### 1. 管理者

管理者はシステム全体の管理を行う最高権限を持つユーザーです。システム全体の運用を監督し、ユーザーやプロジェクトの管理、システムのメンテナンスや設定変更を行うことが可能です。ただし、現在のバージョンでは削除機能は実装されておらず、今後の機能拡張として対応予定です。

管理者ができる機能:  
• ユーザー管理:  
 • 新規ユーザーの作成（管理者、クライアント、一般ユーザーの役割を設定）  
 • ユーザーの情報編集（名前、メールアドレス、パスワード、アクティブ・非アクティブの切り替え）  
 • 現在、ユーザーの削除は未対応（削除機能は今後の実装予定）  
• プロジェクト管理:  
 • プロジェクトの作成、編集（プロジェクトの名称、概要、クライアントの割り当て）  
 • プロジェクトへのユーザー割り当て（クライアントおよび一般ユーザーをプロジェクトに追加）  
 • プロジェクトの削除は未対応（削除機能は今後の実装予定）  
• タスク管理:  
 • 全プロジェクトに関連するタスクの閲覧、作成、編集  
 • タスクの担当者の変更  
 • タスクのステータス管理（pending、in\_progress、completed）  
• コメント管理:  
 • すべてのタスクに対してのコメントの閲覧、削除は未対応（削除機能は今後の実装予定）

### 2. クライアント

クライアントは、特定のプロジェクトに対する管理者的な役割を持ち、プロジェクト内のタスクやユーザーの管理を行います。ただし、クライアントの権限はその担当プロジェクト内に限定され、削除機能は実装されていません。今後のバージョンアップで削除機能が追加される予定です。  
  
クライアントができる機能:  
• プロジェクト管理:  
 • 自身の担当プロジェクトにおいて、プロジェクトの情報を編集（名称、説明など）  
 • プロジェクトに一般ユーザーを追加  
 • プロジェクトの削除は未対応（削除機能は今後の実装予定）  
• タスク管理:  
 • 担当するプロジェクト内でタスクの作成、編集  
 • タスクの担当者の割り当てや変更  
 • タスクのステータス管理（pending、in\_progress、completed）  
• コメント管理:  
 • プロジェクト内のタスクに対するコメントの閲覧  
 • コメントの削除は未対応（削除機能は今後の実装予定）

### 3. 一般ユーザー

一般ユーザーは、クライアントや管理者によってプロジェクトに割り当てられることで、そのプロジェクトのタスクに参加する権限を持ちます。一般ユーザーはタスクの閲覧や担当タスクの進捗管理を行いますが、プロジェクトやユーザー管理の権限はありません。削除機能については利用できず、今後の拡張として対応予定です。  
  
一般ユーザーができる機能:  
• タスクの管理:  
 • 自分が割り当てられたタスクのステータス変更（pending、in\_progress、completed）  
 • タスクの詳細を閲覧  
 • タスクに対するコメントを追加、閲覧  
 • 削除機能は未対応（今後の実装予定）

## 3.2 データ要件

ユーザーテーブル (user\_accounts)  
  
• id: ユーザーを一意に識別するための識別子（自動生成される整数）。  
• user\_id: ユーザーの識別情報を格納するフィールド。  
• name: ユーザーの名前を格納するフィールド。  
• email: ユーザーのメールアドレスを格納するフィールド。ユニークである必要があります。  
• password: ユーザーのパスワードをハッシュ化して格納するフィールド。  
• role: ユーザーの役割を示すフィールド（管理者、クライアント、一般ユーザー）。  
• active\_flag: ユーザーが有効か無効かを示すフラグ。  
• deleted\_flag: ユーザーが削除されたかどうかを示すフラグ（今後の実装予定）。  
• created\_at: ユーザーが作成された日時を記録するフィールド。  
• updated\_at: ユーザーが最後に更新された日時を記録するフィールド。

• active\_flag: ユーザーが有効か無効かを示すフラグ。  
• deleted\_flag: ユーザーが削除されたかどうかを示すフラグ（今後の実装予定）。  
• created\_at: ユーザーが作成された日時を記録するフィールド。  
• updated\_at: ユーザーが最後に更新された日時を記録するフィールド。

プロジェクトテーブル (projects)  
  
• id: プロジェクトを一意に識別するための識別子（自動生成される整数）。  
• name: プロジェクトの名称を格納するフィールド。  
• description: プロジェクトの概要や詳細な説明を格納するフィールド。  
• owner\_id: プロジェクトの所有者を示す識別子。  
• created\_at: プロジェクトが作成された日時を記録するフィールド。  
• updated\_at: プロジェクトが最後に更新された日時を記録するフィールド。

タスクテーブル (tasks)  
  
• id: タスクの一意識別子。タスク間での重複を避けるために自動生成されます。  
• title: タスクの簡単なタイトル。  
• description: タスクの詳細な説明。  
• status: タスクの進捗状況を表すフィールド（pending、in\_progress、completed）。  
• assigned\_to: タスクを担当するユーザーのID。  
• project\_id: タスクが属するプロジェクトのID。  
• created\_at: タスクが作成された日時を記録するフィールド。  
• updated\_at: タスクが最後に更新された日時を記録するフィールド。

コメントテーブル (comments)  
  
• id: コメントの一意識別子。  
• task\_id: コメントが関連付けられているタスクのID。  
• user\_id: コメントを作成したユーザーのID。  
• content: コメントの内容を格納するフィールド。  
• created\_at: コメントが作成された日時を記録するフィールド。  
• updated\_at: コメントが最後に更新された日時を記録するフィールド。

# 4. 非機能要件

• パフォーマンス: システムは高負荷状態でも安定して動作し、タスクの追加や更新を迅速に処理することが求められます。  
• セキュリティ: ユーザーの個人情報や機密情報を保護するため、適切な認証および権限管理を行います。  
• 拡張性: 将来的な機能追加やユーザー増加に対応できるよう、システムは容易に拡張可能であるべきです。  
• ユーザビリティ: システムは直感的で使いやすいインターフェースを提供し、ユーザーが簡単に操作できるように設計されるべきです。  
• レスポンシブデザイン: モバイルデバイスやタブレットでも快適に利用できるよう、デザインを最適化します。

# 5. システム環境

システム環境は、タスク管理システムの運用に必要なハードウェア、ソフトウェア、ネットワーク構成を含みます。以下は具体的な要件です。  
  
• サーバー環境:  
Dockerを使用したコンテナベースの開発環境を構築します。これにより、各コンテナが独立して動作し、アプリケーションの開発・テスト・運用がスムーズに行えるようになります。  
  
• 使用技術:  
• フロントエンド: Reactを使用して、ユーザーインターフェースを構築します。  
• バックエンド: PHP（Laravel）を使用し、APIを構築します。  
• データベース: MySQLを使用して、データの永続化を行います。  
• コンテナ: Dockerを使用して、全体の開発・運用環境を管理します。

# 6. 今後の拡張

• 完全レスポンシブ対応:  
モバイルデバイスやタブレット向けに最適化されたユーザーインターフェースを提供し、あらゆるデバイスからスムーズな操作を実現します。  
• プロフィール機能の充実:  
ユーザーが自身のプロフィールを編集・管理できる機能を強化し、より使いやすいインターフェースを提供します。  
• クライアント機能の拡張:  
一般ユーザーに対して、クライアント機能を持たせることが可能となるよう、権限の拡張を行います。これにより、プロジェクトの管理をより効率的に行えるようになります。  
• 大規模ユーザー管理:  
法人向け機能として、大規模ユーザーの管理やアクセス制御を強化し、複雑な権限設定を実現します。

# 7. まとめ

本要件定義書は、タスク管理システムに関する全体像を明確にし、今後の開発や拡張に向けた基盤を提供することを目的としています。各ユーザーの役割や機能を明確にし、削除機能など未実装の機能についても明記することで、今後の開発方針を示しています。このドキュメントを基に、実際の開発が進められることを期待しています。