

システム名

バーチャルフィットネスコーチ

目的

このシステムは、ユーザーが自宅でフィットネスを行うのを助けます。エクササイズのガイド、ポーズのチェック、運動の記録ができます。

機能一覧

- ユーザー認証（ログイン、新規登録、パスワード変更）
- ユーザープロフィール管理（身長、体重、目標などの編集）
- エクササイズ一覧の表示
- エクササイズの詳細情報表示
- ポーズ検出とリアルタイムフィードバック
- エクササイズ履歴の表示
- マイページでの個人情報と運動統計の確認

画面一覧

画面1: ログイン画面

- 目的: ユーザーがシステムにログインします。
- 機能: ユーザー名とパスワードを入力してログインします。新規登録画面やゲストログインへ移動できます。
- 入力項目: ユーザー名、パスワード
- 出力項目: なし
- 操作フロー:
 - ユーザーがユーザー名とパスワードを入力します。
 - 「ログイン」ボタンを押します。
 - 認証が成功すると、メイン画面へ移動します。
 - 認証が失敗すると、エラーメッセージが表示されます。
 - 「新規登録」ボタンで登録画面へ移動します。
 - 「ゲストログイン」ボタンでゲストとしてログインします。

- スクリーンショット:



バーチャルエクササイズコーチ...

ログイン

ユーザー名:

admin1

パスワード:

●●●●●●

☒ IDとパスワードを記憶する

ログイン

新規登録

ゲストとして続行

画面2: 新規登録画面

- 目的: 新しいユーザーアカウントを作成します。
- 機能: ユーザー名、メールアドレス、パスワードを入力してアカウントを登録します。
- 入力項目: ユーザー名、メールアドレス、パスワード、パスワード確認
- 出力項目: なし
- 操作フロー:

1. ユーザーがユーザー名、メールアドレス、パスワードを入力します。
2. 「登録」ボタンを押します。
3. 登録が成功すると、ログイン画面へ戻ります。
4. 登録が失敗すると、エラーメッセージが表示されます。
5. 「戻る」ボタンでログイン画面へ戻ります。

バーチャルエクササイズコーチ - 新規登録

新規登録

名:

姓:

ユーザー名:

メールアドレス:

パスワード:

パスワード確認:

- スクリーンショット:

画面3: メイン画面（バーチャルフィットネスコーチ画面）

- 目的: エクササイズの一覧を表示し、選択したエクササイズの詳細へ移動します。
- 機能: エクササイズをリストで表示します。マイページ、履歴、ログアウトの機能へアクセスできます。
- 入力項目: なし（エクササイズ選択）
- 出力項目: エクササイズ名、画像
- 操作フロー:
 1. 画面にエクササイズの一覧が表示されます。
 2. ユーザーがエクササイズを選択します。
 3. 選択したエクササイズの詳細画面へ移動します。
 4. 「マイページ」ボタンでマイページ画面へ移動します。
 5. 「履歴」ボタンでエクササイズ履歴画面へ移動します。
 6. 「ログアウト」ボタンでログイン画面へ戻ります。

- スクリーンショット:




画面4: エクササイズ詳細画面

- 目的: 選択したエクササイズの詳しい情報を提供します。
- 機能: エクササイズの説明、指示、動画URLなどを表示します。ポーズ検出を開始できます。
- 入力項目: なし
- 出力項目: エクササイズ名、説明、指示、動画URL、画像など
- 操作フロー:
 1. エクササイズの詳細情報が表示されます。
 2. 「ポーズ検出開始」ボタンを押すと、ポーズ検出画面へ移動します。
 3. 「戻る」ボタンでメイン画面へ戻ります。

- スクリーンショット:



画面5: ポーズ検出画面

- 目的: ユーザーのポーズをリアルタイムで検出し、フィードバックを提供します。また、エクササイズのポーズを記録できます。
- 機能: カメラ映像を表示し、ユーザーのポーズを分析します。エクササイズの指示とカウントダウンを表示します。エクササイズのポーズを記録する機能もあります。
- 入力項目: カメラ映像 (ユーザーの動き)
- 出力項目: リアルタイムのポーズ検出結果、フィードバック、タイマー、ステータスメッセージ
- 操作フロー:
 1. カメラが起動し、ユーザーの映像が表示されます。
 2. エクササイズが開始されます。
 3. システムがユーザーのポーズを検出し、フィードバックを表示します。
 4. エクササイズが終了すると、結果が表示され、履歴に記録されます。
 5. 「記録」ボタンを押すと、カウントダウンが始まり、現在のポーズが記録されます。これは、新しいエクササイズの参照ポーズとして使えます。
 6. 「終了」ボタンでエクササイズを中断し、前の画面へ戻ります。
- スクリーンショット:  スクリーンショット

画面6: エクササイズ履歴画面

- 目的: ユーザーの過去のエクササイズ記録を表示します。
- 機能: 実施したエクササイズ、日時、期間、消費カロリーなどを一覧で表示します。
- 入力項目: なし
- 出力項目: エクササイズ履歴の一覧
- 操作フロー:
 1. 過去のエクササイズ履歴がリストで表示されます。
 2. 「戻る」ボタンでマイページまたはメイン画面へ戻ります。
- スクリーンショット:



画面7: マイページ画面

- 目的: ユーザーの個人情報と運動統計を表示し、プロフィール編集やパスワード変更へ移動します。
- 機能: ユーザー名、メールアドレス、身長、体重、フィットネスレベルなどの情報を表示します。
- 入力項目: なし
- 出力項目: ユーザー情報、運動統計
- 操作フロー:
 1. ユーザーのプロフィール情報と運動統計が表示されます。
 2. 「プロフィール編集」ボタンでプロフィール編集画面へ移動します。
 3. 「パスワード変更」ボタンでパスワード変更画面へ移動します。
 4. 「履歴を見る」ボタンでエクササイズ履歴画面へ移動します。

- 5. 「ログアウト」ボタンでログイン画面へ戻ります。
- 6. 「戻る」ボタンでメイン画面へ戻ります。
- スクリーンショット:



画面8: プロファイル編集画面

- 目的: ユーザーのプロファイル情報を更新します。
- 機能: 身長、体重、フィットネスレベル、目標などを編集し、保存します。
- 入力項目: 身長、体重、フィットネスレベル、目標、好みなど
- 出力項目: なし
- 操作フロー:
 1. 現在のプロファイル情報が表示されます。

2. ユーザーが情報を編集します。
3. 「保存」ボタンを押すと、情報が更新され、マイページ画面へ戻ります。
4. 「キャンセル」ボタンを押すと、変更を破棄してマイページ画面へ戻ります。



ユーザー情報編集

ユーザー名: admin1

メールアドレス: ada@example.com

名: sumit

姓: punn

保存 キャンセル

- スクリーンショット:

画面9: パスワード変更画面

- 目的: ユーザーのパスワードを変更します。
- 機能: 現在のパスワードと新しいパスワードを入力してパスワードを更新します。
- 入力項目: 現在のパスワード、新しいパスワード、新しいパスワード確認
- 出力項目: なし
- 操作フロー:
 1. ユーザーが現在のパスワードと新しいパスワードを入力します。
 2. 「変更」ボタンを押すと、パスワードが更新され、マイページ画面へ戻ります。
 3. 「キャンセル」ボタンを押すと、変更を破棄してマイページ画面へ戻ります。



パスワード変更

現在のパスワード: |

新しいパスワード:

新しいパスワード再入力:

☐ パスワードを表示する

パスワードを変更する 戻る

- スクリーンショット:

データベース設計

- テーブル名: users
 - id: SERIAL ・ ユーザーID (主キー)
 - username: VARCHAR(50) ・ ユーザー名 (ユニーク、必須)
 - email: VARCHAR(100) ・ メールアドレス (ユニーク、必須)
 - password_hash: VARCHAR(255) ・ パスワードのハッシュ値 (必須)

- `first_name`: VARCHAR(50) ・ 名
 - `last_name`: VARCHAR(50) ・ 姓
 - `created_at`: TIMESTAMP ・ 作成日時 (デフォルト現在時刻)
 - `updated_at`: TIMESTAMP ・ 更新日時 (デフォルト現在時刻)
 - `last_login`: TIMESTAMP ・ 最終ログイン日時
 - `is_active`: BOOLEAN ・ アカウントが有効か (デフォルトTRUE)
- テーブル名: `user_sessions`
 - `id`: SERIAL ・ セッションID (主キー)
 - `user_id`: INTEGER ・ ユーザーID (`users` テーブル参照、必須)
 - `token_hash`: VARCHAR(255) ・ JWTトークンのハッシュ値 (必須)
 - `expires_at`: TIMESTAMP ・ 有効期限 (必須)
 - `created_at`: TIMESTAMP ・ 作成日時 (デフォルト現在時刻)
 - `is_valid`: BOOLEAN ・ セッションが有効か (デフォルトTRUE)
- テーブル名: `user_profiles`
 - `id`: SERIAL ・ プロファイルID (主キー)
 - `user_id`: INTEGER ・ ユーザーID (`users` テーブル参照、必須)
 - `age`: INTEGER ・ 年齢
 - `height_cm`: INTEGER ・ 身長 (cm)
 - `weight_kg`: DECIMAL(5,2) ・ 体重 (kg)
 - `fitness_level`: VARCHAR(20) ・ フィットネスレベル (デフォルト'beginner')
 - `goals`: TEXT ・ 目標
 - `preferences`: TEXT ・ 好み
 - `created_at`: TIMESTAMP ・ 作成日時 (デフォルト現在時刻)
 - `updated_at`: TIMESTAMP ・ 更新日時 (デフォルト現在時刻)
- テーブル名: `exercises`
 - `id`: SERIAL ・ エクササイズID (主キー)
 - `name`: VARCHAR(255) ・ エクササイズ名 (必須)
 - `image_path`: VARCHAR(255) ・ 画像パス
 - `category`: VARCHAR(100) ・ カテゴリ
 - `primary_muscle`: VARCHAR(100) ・ 主要な筋肉
 - `secondary_muscles`: TEXT ・ 補助的な筋肉
 - `equipment`: TEXT ・ 使用器具
 - `difficulty_level`: VARCHAR(50) ・ 難易度
 - `instructions`: TEXT ・ 指示
 - `tips`: TEXT ・ ヒント
 - `video_url`: VARCHAR(255) ・ 動画URL
 - `reps_sets_suggestion`: VARCHAR(255) ・ 回数・セットの提案
 - `benefits`: TEXT ・ 効果
 - `common_mistakes`: TEXT ・ よくある間違い

- `variations`: TEXT・バリエーション
 - `calories_burned_estimate`: VARCHAR(50)・消費カロリーの目安
 - `created_at`: TIMESTAMP・作成日時（デフォルト現在時刻）
 - `updated_at`: TIMESTAMP・更新日時（デフォルト現在時刻）
- テーブル名: `exercise_history`
 - `id`: SERIAL・履歴ID（主キー）
 - `user_id`: INTEGER・ユーザーID（`users` テーブル参照、必須）
 - `exercise_id`: INTEGER・エクササイズID（`exercises` テーブル参照、必須）
 - `session_date`: TIMESTAMP・セッション日時（デフォルト現在時刻）
 - `duration_minutes`: INTEGER・期間（分）
 - `calories_burned`: INTEGER・消費カロリー
 - `notes`: TEXT・メモ
 - `status`: VARCHAR(50)・ステータス（例: '確認', '試行', '実行', デフォルト'確認'）
 - `performed_seconds`: INTEGER・実際に行った時間（秒）

使用技術

- フロントエンド: GTKMM (C++ GUI Toolkit)
- バックエンド: C++ (アプリケーションロジック)
- データベース: PostgreSQL
- その他: TensorFlow Lite (ポーズ検出モデル)

システム構成図

このシステムは、ユーザーインターフェース（GTKMMで開発されたデスクトップアプリケーション）、アプリケーションロジック（C++）、データベース（PostgreSQL）、そしてポーズ検出モデル（TensorFlow Lite）で構成されています。

1. **ユーザーインターフェース**: ユーザーはGTKMMアプリケーションを通じてシステムと対話します。ログイン、エクササイズ選択、プロフィール管理などの操作を行います。
2. **アプリケーションロジック**: C++で書かれたコアロジックが、ユーザーからの入力処理、データベースとの通信、ポーズ検出モデルの呼び出し、ビジネスルールの適用を担当します。`AuthManager` が認証を、`DatabaseManager` がデータベース操作を、各 `*Window` クラスが画面表示とイベント処理を担当します。
3. **データベース**: PostgreSQLがユーザー情報、エクササイズ情報、運動履歴などのデータを保存します。`DatabaseManager` を通じてアクセスされます。
4. **ポーズ検出モデル**: TensorFlow Liteモデル（`movenet_singlepose_lightning.tflite`）が、`PoseDetectionWindow` 内でカメラからの映像を分析し、ユーザーのポーズをリアルタイムで検出します。

システムはクライアントサイドで動作し、データベースとポーズ検出モデルをローカルまたはネットワーク経由で利用する構成です。