

要件定義書

saku0512

目次

1	概要	3
2	プロジェクト構成	3
3	技術スタック	4
4	ユーザー種別と権限	5
5	共通機能要件	5
5.1	認証機能	5
5.2	初回ログイン時の表示名設定	5
6	管理者用アプリ機能要件	6
6.1	ダッシュボード	6
6.2	スポーツ管理 (CRUD)	6
6.3	クラス・チーム管理	6
6.4	学生管理	6
6.5	参加者登録	6
6.6	トーナメント・試合管理	6
6.7	QR コードリーダー機能	6
7	学生用アプリ機能要件	7
7.1	ダッシュボード	7
7.2	競技一覧・詳細閲覧	7
7.3	QR コード発行機能	7
7.4	得点一覧閲覧	7
8	データベース設計 (案)	8
9	画面遷移図 (案)	8
9.1	共通フロー	8

9.2	管理者アプリ	8
9.3	学生アプリ	8

1 概要

本プロジェクトは、学校のスポーツ大会運営を効率化するための Web アプリケーションを開発するものである。大会の準備から結果の記録までを一元管理する**管理者用アプリ**と、学生が大会情報を確認し、参加登録を行うための**学生用アプリ**の2つを開発する。

これにより、実行委員会の負担を軽減し、学生への情報伝達をスムーズにすることを目的とする。

2 プロジェクト構成

- **モノリポ (Monorepo)** : 一つの Git リポジトリで管理者用、学生用、バックエンドの3つのプロジェクトを管理する。これにより、コードの共通化やバージョン管理が容易になる。
 - packages/front/admin-app: 管理者用フロントエンド (Svelte 5)
 - packages/front/student-app: 学生用フロントエンド (Svelte 5)
 - packages/back: 共通バックエンド (Gin + GraphQL + MySQL)
- **アプリケーション**:
 1. **管理者用アプリ**: 大会情報の登録、更新、管理を行う Web アプリ。実行委員のみが利用。
 2. **学生用アプリ**: 大会情報の閲覧、および参加登録用 QR コードの発行を行う Web アプリ。全学生が利用。

3 技術スタック

表 1 技術スタック一覧

領域	技術・ライブラリ	理由
バックエンド	Go (Gin)	高速なパフォーマンスとシンプルな文法が特徴。小～中規模の API サーバー開発に適しており、今回の要件にマッチする。
API	GraphQL (gqlgen)	推奨 。トーナメント表や試合結果など、関連するデータを一度に効率よく取得する場面が多いため。フロントエンドが必要なデータだけをリクエストでき、REST API で起こりがちな過剰なデータ取得（オーバーフェッチ）や複数回のリクエスト（N+1 問題）を防げる。
フロントエンド	Svelte 5	コンパイル時に最適化されるため、高速な動作が期待できる。シンプルで学習コストが低く、迅速な開発に適している。
認証	Microsoft Authentication / Google Authentication	OAuth 2.0 を利用したセキュアな認証を実装する。学校で利用している Google Workspace や Microsoft 365 のアカウントを流用することで、ユーザー管理の負担を軽減する。
DB	PostgreSQL / MySQL (選択)	リレーショナルデータとの相性が良く、信頼性が高い。
インフラ	Docker	コンテナ化を利用した独立環境を構築することで、環境の整合性を高め、コードの構築やテストを簡単に行うことが可能になる。

4 ユーザー種別と権限

表 2 ユーザー権限

機能	管理者（実行委員）	学生
アプリへのログイン	✓ 両方	✓ 学生用のみ
スポーツ情報の登録・編集	✓	X
競技概要の登録・編集	✓	X
参加学生の登録	✓	X
試合結果の入力	✓	X
トーナメント表の閲覧	✓	✓ 閲覧のみ
学生情報の管理	✓	X
参加登録用 QR コード読取	✓	X
参加登録用 QR コード発行	X (※ 1)	✓

(※ 1) 管理者も学生として競技に参加する場合を考慮し、学生用アプリにログインすれば発行可能。

5 共通機能要件

5.1 認証機能

- Google または Microsoft アカウントによる OAuth 認証を実装する。
- **ドメイン制限:** 学校から指定されたドメイン（@sendai-nct.jp）を持つアカウントのみログインを許可する。
- **ホワイトリスト:** 上記ドメイン制限に加え、事前に登録されたメールアドレスリスト（ホワイトリスト）に含まれるユーザーのみがログインできるようにする。これにより、関係者以外のアクセスを完全に遮断する。

5.2 初回ログイン時の表示名設定

- いずれのアプリでも、ユーザーが初めてログインした際に、アプリ内で使用する**表示名**を設定する画面にリダイレクトする。
- ダッシュボードなどのメイン機能は、表示名が設定されるまで利用できないようにする。
- 設定した表示名は、後からプロフィールページ（要追加検討）で変更できるようにする。

6 管理者用アプリ機能要件

6.1 ダッシュボード

登録済みのスポーツ一覧、本日の試合予定、未入力 of 試合結果などを一覧表示する。

6.2 スポーツ管理 (CRUD)

- 大会で行うスポーツを登録・編集・削除できる。
- 各スポーツの概要、ルール、トーナメント表 (後々アプリ側で完全ランダム生成するようにする) を登録できる。

6.3 クラス・チーム管理

参加単位となるクラスやチームを登録・管理する。

6.4 学生管理

- ログインした全学生のリスト (表示名、メールアドレス) を閲覧できる。
- 学生の役割 (一般学生 or 実行委員) を変更できる。

6.5 参加者登録

- 各スポーツに、参加する学生をクラス単位または個人単位で割り当てる。
- UI は、クラスリストから参加するスポーツへドラッグ&ドロップするような直感的な操作を検討する。

6.6 トーナメント・試合管理

- 登録された参加者情報を基に、トーナメント表を自動生成する。
- 試合結果 (スコアなど) を入力し、トーナメント表に反映させる。勝者が自動的に次のコマに進むようにする。
- 試合結果に応じて、自動で得点付与処理を行う。

6.7 QR コードリーダー機能

- デバイスのカメラを使用し、学生が提示する QR コードを読み取る。
- 読み取った QR コードから学生情報を特定し、競技への参加受付 (チェックイン) 処理を行う。

7 学生用アプリ機能要件

7.1 ダッシュボード

自分が参加する競技の一覧や、本日の試合予定などを表示する。

7.2 競技一覧・詳細閲覧

- 開催される全スポーツの一覧を閲覧できる。
- 各スポーツの詳細ページで、ルールや概要、現在のトーナメントの進捗状況を確認できる。試合結果もリアルタイムで反映される。

7.3 QR コード発行機能

- ログイン中の学生に紐づいた、一意の QR コードを画面に表示する。
- この QR コードには、学生を識別するための ID 情報（例：データベース上のユーザー ID）が含まれる。
- 管理者がこの QR コードを読み取ることで、本人確認と参加登録が完了する。

7.4 得点一覧閲覧

- 全クラスの現在の得点状況を閲覧できる。
- 自分のクラス名や競技名、現在の上位クラス等をソートして表示できる。

8 データベース設計（案）

表 3 主要なテーブル構成案

テーブル名	カラム	説明
users	id, google_id, microsoft_id, email, display_name, role (admin or student), created_at	ユーザー情報。‘role’で管理者と学生を区別する。
sports	id, name, description, rules, format (tournament など)	競技情報。
teams	id, name, class_name	参加チーム（クラスなど）。
participants	id, user_id, team_id	ユーザーとチームを紐付ける中間テーブル。
sport_entries	id, sport_id, team_id	どのチームがどの競技に参加するかを管理する。
matches	id, sport_id, round_number, team1_id, team2_id, winner_id, score, match_time	試合情報。トーナメントの各試合を記録する。

9 画面遷移図（案）

9.1 共通フロー

ログインページ → (Google/Microsoft 認証) → 認証成功 → (初回ログインか?)

- Yes: 表示名設定ページ → (設定完了) → 各アプリのダッシュボード
- No: 各アプリのダッシュボード

9.2 管理者アプリ

ダッシュボード → スポーツ管理 / 参加者登録 / 試合結果入力 ...

9.3 学生アプリ

ダッシュボード → 得点閲覧 / 競技一覧 → 競技詳細・トーナメント表 / QR コード表示