

FlexField
Technical Design Document
- Team 3(Vanguardians)

Introduction

Purpose: This Technical Design Document (TDD) elaborates on the technical architecture, design patterns, data model, and technology stack required to implement the insurance platform outlined in the Functional Requirements Document (FRD) version. It serves as a blueprint for the development team to build and deploy the system.

Intended Audience: This document is intended for the development team (software engineers, database administrators, DevOps engineers), architects, technical leads, and quality assurance engineers involved in the insurance platform project.

Scope: This TDD covers the high-level and low-level technical design aspects of the core functionalities described in the FRD, including user management, games management, team registration model, turf booking, team and player finder and upcoming tournaments management. It also addresses the non-functional requirements related to performance, usability, reliability, and scalability.

Relationship to FRD: This document directly references and expands upon the functional and non-functional requirements defined in the FRD. Each technical design decision aims to fulfill one or more of the stated requirements.

I. Data Model

The data model will be designed to efficiently store and retrieve information related to users, policies, claims, and other relevant entities. Below is a high-level overview of key entities and their attributes:

- **Individual_type:**
 - type_id (Int, Primary Key, Auto-Increment)
 - type_name (Varchar(50))
 - **Values:** “Individual”, “Team”

- **Registered_players:**

- player_id (Int,Primary Key, Auto-Increment)
- player_name(Varchar(50))
- player_date_of_birth(Date)
- gender(Varchar(10))
- player_type_id(Int, Foreign Key, references individual_type(id))
- player_contact_number(Varchar(15))
- player_email(Varchar(100))
- player_pincode(Varchar(10))
- player_password(Varchar(255))
- player_booking(Boolean)

- **Sports:**

- sport_id (Int,Primary Key, Auto-Increment)
- sport_name(Varchar(50))

➤ **Values:** “Cricket”, “Football”

- **Teams:**

- team_id (Int,Primary Key, Auto-Increment)
- team_name(Varchar(50))
- team_contact_number(Varchar(15))
- team_email(Varchar(100))
- team_sport_id(Int, Foreign Key, references sports(sport_id))

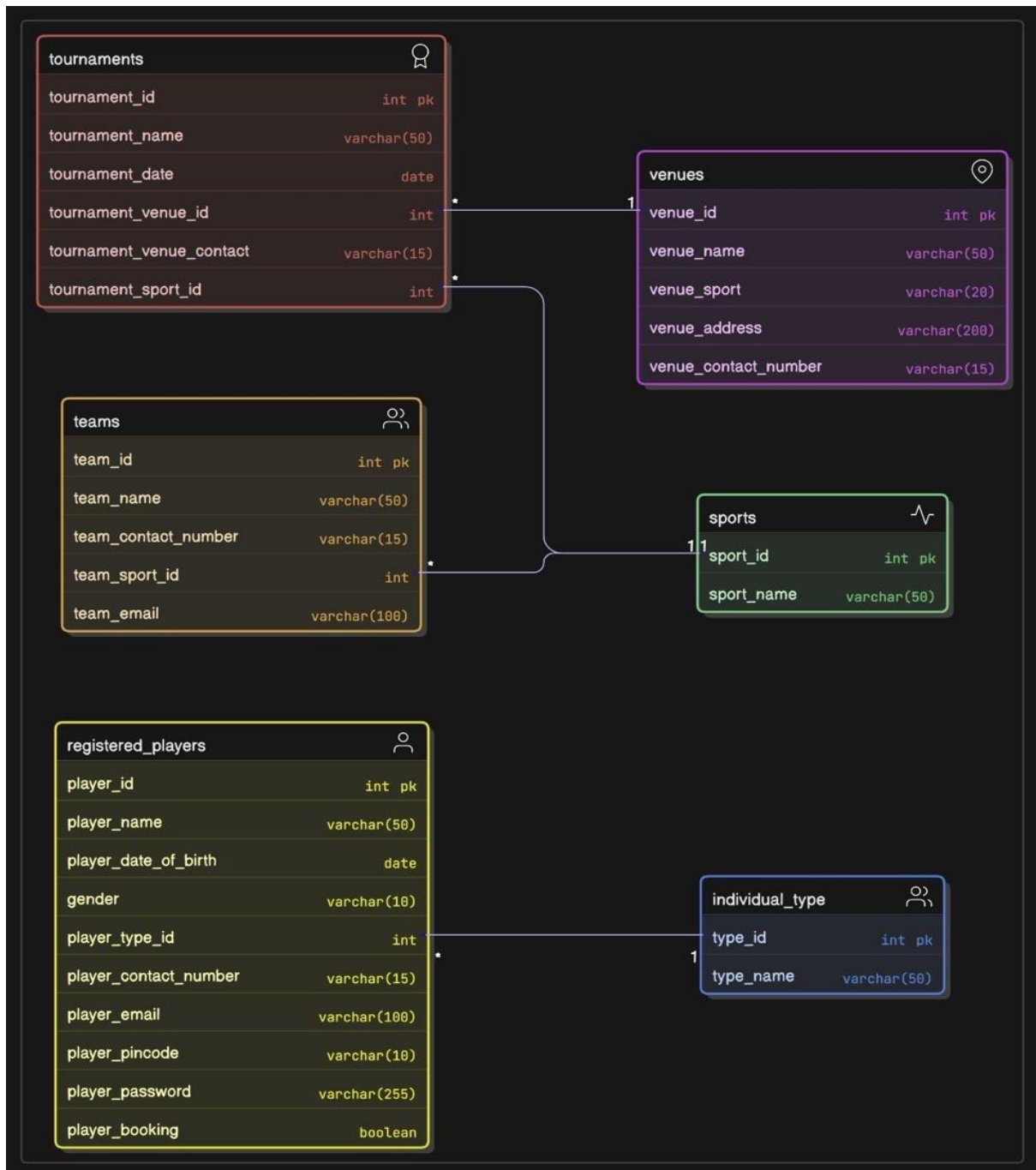
- **Venues:**

- venue_id (Int,Primary Key, Auto-Increment)
- venue_name(Varchar(50))
- venue_sport(Varchar(20))
- venue_address (Varchar(200))
- venue_contact_number(Varchar(15))

- **Tournaments:**

- tournament_id (Int,Primary Key, Auto-Increment)

- tournament_name(Varchar(50))
- tournament_date(Date)
- tournament_venue_id(Int, Foreign Key references venues(venue_id))
- tournament_venue_contact(Varchar(15))
- tournament_sport_id(Int, Foreign Key references sports(sport_id))



Entity Relationship Diagram

II. High-Level Architecture

We purpose a multi-tier architecture to ensure scalability, maintainability, and security. The key layers are:

2.1 Presentation Layer: This layer will be responsible for the user interface (UI) and user experience (UX). It will handle user interactions and display information to various user roles (individual and team).

Technology: React, HTML5, CSS3 and Bootstrap.

Key Components:

- User registration and login interfaces.
- dashboards and navigation.
- Page rendering for options Book Now, Fins a team, Find a player, View Tournament.
- Data display components (Book Turf Cricket, Book Turf Football, Find Available Cricket Player, Find Available Football Player, Registered Cricket Team, Registered Football Team, Upcoming Tournaments.
- Integration with mapping services for nearby garages.

Homepage (Before Login)

The homepage serves as the landing page for new users. Its purpose is to introduce the core offerings of the platform and encourage user sign-up or login. The layout is structured to present FlexField's features in a visually appealing and intuitive format.

Layout Overview:

1. Navigation Bar (Top)

- Logo (Top Left): Positioned prominently to maintain brand presence.
- Login Button (Top Right): Directs users to the login page.

2. Hero Section

- Main Banner Image: Centered and sized to immediately capture attention.
- Headline: "Book Turfs. Join Teams. Play More."

- About FlexField Subheading: Provides a brief mission statement.
- Description: Text describing how FlexField connects players to turfs, teams, and tournaments.

3. "How It Works" Section (Features)

- Feature 1: Book a Turf
 - Left-side: Title and description
 - Right-side: Image depicting turf booking
 - Feature 2: Join a Team
- Left-side: Image
- Right-side: Title and description
 - Feature 3: Play & Compete
- Left-side: Title and description
- Right-side: Image

4. "Why Choose FlexField?" Section

Emphasizes key platform advantages:

- Easy Booking: Real-time availability.
- Find Players: Location-based discovery.
- Local Tournaments: Access to curated events.
 - Images and text alternate layout between left and right for visual rhythm.

5. "What Players Say" Section (Testimonials)

Four identical testimonial blocks displayed in a responsive grid.

Each block contains:

- Quotation from a user
- Name and location

6. Footer

Logo (Bottom Left): Reinforces brand identity.

Quick Links Column:

- Home
- Book a Turf
- Find Teams
- Tournaments

Contact Column:

- Email
- Phone
- Location

LOGIN PAGE

The login page allows existing users to securely access their FlexField account. It provides a simple, focused interface for entering credentials with easy navigation back to signup if the user is new.

Navigation Bar (Top):

Logo (Top Left)

1. "Login" Section

Headline: "Login" centre

2. User id Field (Email/Phone No.)

- Text area having placeholder (Enter your email or phone).
- Input type = "text", id = "email"

3. User password field (Password)

- Text area having placeholder (Enter your password)
- Input type = "password", id = "password"

4. Login Button

- Button "Login" to submit information.
- Input type = "submit"

5. Sign up link: Text with Sign up link.

SIGN UP PAGE

The sign-up page enables new users to create a FlexField account by providing personal and contact details, along with login credentials. The page offers a clean form with clear labels and easy navigation back to login if the user already has an account.

Navigation Bar (Top):

Logo (Top Left)

1. "Sign Up" Section

Headline: "Sign Up" centre

2. Name Field

- Text area having placeholder (Enter your name).
- Input type = "text", id = "name"

3. DOB and Gender Fields

- DOB: Date picker input, input type = "date", id = "dob"
- Gender: Radio buttons for Male/Female, input type = "radio", name = "gender"

4. Contact No. Field

- Text area having placeholder (Enter your phone number).
- Input type = "tel", id = "contact"

5. Email Field

- Text area having placeholder (Enter your email).
- Input type = "email", id = "email"

6. Pin Code Field

- Text area having placeholder (Enter your area pin code).
- Input type = "text", id = "pincode"

7. Player Type Field

- Radio buttons to register as Individual or Team.
- Input type = "radio", name = "playerType"

8. Password Fields

- Password input with placeholder (New), input type = "password", id = "password"
- Confirm password input with placeholder (Re-enter), input type = "password", id = "confirm"

9. Sign Up Button

- Button "Sign Up" to submit information.
- Input type = "submit"

Homepage After Login

The homepage after login serves as the central dashboard for logged-in users. It highlights key features of FlexField and provides easy access to core functionalities like booking turfs, joining teams, and managing profiles. The layout balances visual appeal with intuitive navigation to enhance user engagement.

1. Navigation Bar (Top)

- **Left:** FlexField logo (maintains brand identity)
- **Right:**

- **Games dropdown button:** Lists available games like Cricket, Football
- **Team Register button:** Opens a modal for team registration
- **Profile icon button:** Circular icon showing a user avatar; clicking reveals a dropdown with user details and a logout button

2. Hero + About Section (Centered below Navbar)

- **Centered large banner image:** Sports-themed, vibrant and engaging
- **Below image:**
 - Main headline: “Book Turfs. Join Teams. Play More.” (Large, bold)
 - Subheading: “About FlexField” (medium-sized, bold)
 - Short description paragraph explaining platform purpose and value

3. Features Sections (Grid with two columns on medium+ screens, stacked on small)

Alternating text and image layout to visually separate each feature:

- **Feature 1 – Book a Turf:**
 - Left column: Title + description
 - Right column: Turf booking image
- **Feature 2 – Join a Team:**
 - Left column: Image of players/teams
 - Right column: Title + description
- **Feature 3 – Play & Compete:**
 - Left column: Title + description
 - Right column: Image of gameplay/competition

4. Why Choose FlexField Section (Grid with alternating image and text)

Highlights key advantages with alternating layout to keep rhythm and visual interest:

- Easy Booking: Image on left, text on right
- Find Players: Text on left, image on right
- Local Tournaments: Image on left, text on right

5. Testimonials Section (Centered grid layout)

- Section title: “What Players Say”
- Four testimonial cards arranged in a responsive grid (1 to 4 columns depending on screen size)
- Each testimonial includes:
 - User quote (italicized)
 - User name and location (highlighted in brand color)

6. Team Register Modal (Hidden by default)

- Centered popup overlay triggered by “Team Register” button
- Contains a form with fields: Team Name, Sport (radio buttons), Phone Number, Email
- Submit button to register team
- Close button to dismiss modal

7. Footer (Three-column grid on medium+ screens, stacked on small)

- **Left column:** FlexField logo + brief about text
- **Middle column:** Quick navigation links (Home, Book a Turf, Find Teams, Tournaments)
- **Right column:** Contact details (Email, Phone, Location)

Cricket Landing Page

The Cricket landing page highlights cricket-related features of FlexField and it opens every time you click **Cricket** from the **Games** dropdown menu in the navigation bar. It showcases cricket-specific features in an interactive and responsive layout.

1. Navigation Bar (Top)

- Left: FlexField logo
- Right:
 - **Games dropdown** (shows Cricket and Football options on click)
 - **Team Register** link
 - **Profile icon** that reveals user info and logout button on click*Dropdown menus toggle visibility on click for smooth interaction.*

2. Background and Overlay

- Cricket-themed background image with a semi-transparent white overlay to keep content readable

3. Hero Section (Grid of 4 Cards with Buttons)

- Each card features an image with a dark overlay and a prominent button:
 - **Book Now** — to book cricket turfs
 - **Find a Team** — to search for cricket teams
 - **Find a Player** — to find cricket players
 - **View Tournaments** — to explore cricket tournaments*Cards have a subtle scale and shadow effect on hover, enhancing interactivity.*

Football Landing Page

The Football landing page highlights football-related features of FlexField and opens every time you click **Football** from the **Games** dropdown menu in the navigation bar. It showcases football-specific features in an interactive and responsive layout.

1. Navigation Bar (Top)

- Left: FlexField logo
- Right:
 - **Games dropdown** (shows Cricket and Football options on click)
 - **Team Register** link
 - **Profile icon** that reveals user info and logout button on clickDropdown menus toggle visibility on click for smooth interaction.

2. Background and Overlay

- Football-themed background image with a semi-transparent white overlay to keep content readable

3. Hero Section (Grid of 4 Cards with Buttons)

- Each card features an image with a dark overlay and a prominent button:
 - **Book Now** — to book football turfs
 - **Find a Team** — to search for football teams
 - **Find a Player** — to find football players
 - **View Tournaments** — to explore football tournamentsCards have a subtle scale and shadow effect on hover, enhancing interactivity and user engagement.

Turf Booking Page

The Turf Booking page opens whenever you click **Book Now** on either the Cricket or Football landing pages. It dynamically displays turfs available for the specific sport you chose—cricket turfs if you clicked from cricket, football turfs if from football.

1. Navigation Bar

- Contains the FlexField logo, Games dropdown, Team Register link, and Profile icon with user details and logout option.
- Consistent across all pages for seamless navigation.

2. Page Header

- Displays a clear heading like " Book Cricket Turf" or "Book Football Turf" based on the selected sport.

3. Search Filters

- Date picker to select booking date.
- Location input for filtering turfs by area.
- Search button to apply filters.

4. Turf Listings

- Grid layout showcasing multiple turf cards.
- Each card includes:
 - Turf name and address
 - Rating stars
 - Available time slots
 - Price per hour
 - Contact number
- Cards have hover effects for better interactivity.

Team Find Page

This page opens when the user clicks **Find a Team** from either the Cricket or Football main pages. It dynamically adjusts content based on the selected sport.

1. Navigation Bar

Shows the FlexField logo, navigation links to Games and Team Register, and a profile icon for user account options.

2. Page Header

Displays a clear heading such as "Registered Cricket Teams" or "Registered Football Teams" according to the sport selected.

3. Player Cards Grid

Displays a grid of Registered Team details that were stored in the database when a user uses the team register feature from navigation bar and register his team there.

- Each card includes:

- Team name
 - Location
 - phone number
 - Email
- Cards have hover effects for interactivity.

Player Find Page

This page opens when the user clicks **Find a Player** from either the Cricket or Football main pages. It dynamically adjusts content based on the selected sport.

1. Navigation Bar

Shows the FlexField logo, navigation links to Games and Team Register, and a profile icon for user account options.

2. Page Header

Displays a clear heading such as "Find Available Football Players" or "Find Available Cricket Players" according to the sport selected.

3. Search Filters

- Location input field to filter players by geographic area.
- Role dropdown lets users filter players by specific roles relevant to the sport (e.g., Football roles: Goalkeeper, Defender, Midfielder, Forward; Cricket roles could be Batsman, Bowler, All-Rounder, Wicket-Keeper).
- These filters help users quickly narrow down player options.

4. Player Cards Grid

- Displays a grid of player profiles matching the search criteria.
- Each card includes:
 - Player's name
 - Role
 - Location
 - Contact phone number
- Cards have hover effects for interactivity.

Upcoming Tournaments Page

This page is displayed when the user clicks "**Upcoming Tournaments**" from either the **Cricket** or **Football** landing pages. It presents a list of nearby sports events and is visually consistent with the rest of the FlexField platform.

1. Navigation Bar

Displays at the top of the page and includes:

- FlexField logo on the left
- Navigation links:
 - **Games**
 - **Team Register**
- A profile icon button on the right for user account access

2. Page Header

- A centered title: " Upcoming Tournaments"
- Subtitle text: "Discover the latest sports events near you!"
- Both styled for clarity and emphasis

3. Tournament Cards Grid

Displays upcoming tournament information in a responsive 2–3 column grid depending on screen size.

Each card includes:

- A sport type badge at top left of card (Cricket in green, Football in blue)
- A date range at top right of card(e.g., "June 5–10")
- Tournament name
- Venue address
- Organizer name

III. Application Layer (API Layer): This layer will act as an intermediary between the presentation layer and the business logic layer. It will expose RESTful APIs for the UI to interact with the backend.

Backend Coding (Spring Boot Structure)

Typical Structure:

src

```
└── main/java/com/flexfield
    ├── controller // API endpoints
    ├── service    // Business logic
    ├── model      // JPA Entities
    └── repository // Data access (Spring Data JPA)
```

└─ security // JWT auth/config
└─ dto // Data Transfer Objects
└─ exception // Global error handling

Useful Tools/Libraries:

- Spring Security(JWT)
- Spring Data JPA(Repository pattern)
- ModelMapper/MapStruct(DTO mapping)
- Validation(javax.validation, e.g., @Email, @NotNull)

IV. Validation

Frontend (React)

- All required fields enforced in forms (can't submit empty)
- Validate:
 - Email (syntactic check)
 - Phone numbers
 - Password strength (length, characters, etc.)
- Show field-level error messages before sending to backend.

Backend (Spring Boot)

- Use @Valid and bean validation annotations in DTOs
 - @Email , @NotNull , @Pattern , @Size , etc.
- Manual checks (e.g. slot overlap, uniqueness constraints)
- Return detailed error messages (400/422) for invalid fields

Example DTO validation:

```
public class SignupRequest {  
    @NotBlank  
    private String name;  
  
    @Email
```



```

@NotBlank

private String email;

@Pattern(regexp="^[0-9]{10}$")
private String contactNo;

// ...
}

```

V. API Design

Authentication

- POST /api/auth/login (email/phone, password) → JWT token (+ user info)
- POST /api/auth/signup (user fields) → user created

Venues

- GET /api/venues?location=&sport=
- GET /api/venues/{id}

Booking

- POST /api/bookings (needs auth) – create booking
- GET /api/bookings?userId= – user bookings
- DELETE /api/bookings/{id} – cancel

Teams

- POST /api/teams/register (needs auth)
- GET /api/teams?location=&sport=

Players

- GET /api/players?location=&role=&sport=

Tournaments

- GET /api/tournaments?sport=

Reviews

- POST /api/reviews (venue_id, rating, comment)

Standard responses:

- 200/201 for success, 400/422 validation error, 401 unauthorized, 500 server error

VI. Services

Service Layer Classes:

- **AuthService:** handles login, registration, password hashing, JWT token handling
- **UserService:** user profile management
- **VenueService:** venue search, listing
- **BookingService:** check slot availabilities, booking logic, pricing, prevent double-booking; email notification stub
- **TeamService:** team create/list logic, join/leave
- **PlayerService:** player profile, search
- **TournamentService:** tournament list
- **ReviewService:** handle one review per booking
- **NotificationService:** (optional/future) email/SMS
- **ValidationService:** any custom re-usable checks

Business Logic Example:

Before confirming booking, BookingService must:

- Check venue/turf availability for given slot/date
- Prevent overlapping for user/venue
- Only proceed if check passes

Example BookingService(Pseudo- Java):

@Service

```
public class BookingService {
```

```
    @Autowired
```

```
    private BookingRepository bookingRepo;
```

```
    public BookingResponse createBooking(BookingRequest req, User user) {
```

```
        boolean available = bookingRepo.isSlotAvailable(req.getVenueId(), req.getDate(), req.getSlot());
```

```

    if (!available) {
        throw new BookingException("Slot Already Booked");
    }
    Booking booking = new Booking(/* fields from req + user */);
    booking.setStatus(Status.CONFIRMED);
    bookingRepo.save(booking);
    // (Optional) call notificationService.sendBookingConfirmation(...)
    return new BookingResponse(booking);
}
}

```

Security

- All POST/DELETE/PUT endpoints (except login/signup) require JWT bearer token
- Use `@PreAuthorize/@Secured` for endpoint-level security
- Passwords stored with Bcrypt

Error Handling

- Use `ControllerAdvice` for global exception handling (return JSON error messages)
- Return 400 for validation, 401 for unauthorized, 404 for not found

Utility

- Use environment variables (`application.properties/yaml`) for DB, JWT secrets, mail
- Use standard logging (`slf4j`, `logback`)
- DB migrations handled with Flyway/Liquibase if versioning is needed

Detailed Design – Key Features

User Management

Login: The login functionality enables existing users to securely access their accounts through a React-based interface that handles form inputs and validations. On the backend, Spring Boot services manage authentication processes, leveraging Spring Security for secure

password validation and session management. User credentials and profiles are stored in a MySQL database to maintain persistent and reliable user data.

Sign-Up Page: The sign-up page allows new users to register by entering personal, contact, and login details via React-controlled forms with Bootstrap styling for responsiveness. Data submitted is validated both client-side and server-side, with Spring Boot handling the backend logic to create new user records in the MySQL database, ensuring data integrity and secure storage.

Profile Management: Allows users to view their profile and update their details. Users can enter their details in sign-up page and later write about themselves in the provided field.

Games Management

Cricket Specific feature: The Cricket landing page focuses on cricket-specific features, opening when the user selects Cricket from the games dropdown. Backend services built with Spring Boot expose APIs that provide cricket-specific team, players, turf, and tournament data from the MySQL database.

Football Specific feature: Similar, the football landing page utilizes React and Bootstrap for UI rendering and styling. The Spring Boot backend serves football-related data via RESTful APIs interfacing with MySQL, enabling users to interact with real-time football-specific content.

Team Registration Modal

Functionality: Triggered by the team register button in the navigation, this modal allows users to register their team by submitting the team name, sport selection, phone number, and email. Validation occurs client-side, and submissions are processed by Spring Boot endpoints that persist team data in MySQL.

Turf Booking Feature

Search and Filter Functionality: Users can filter turf availability by selecting a booking date through an interactive date picker and entering their desired location. A search button triggers the retrieval of relevant turfs, refining results based on user inputs to provide precise options.

Turf Listings Display: Available turfs are showcased in a responsive grid layout. Each turf card presents essential.

Booking Process: Selecting a turf enables users to view and choose from available venues. The booking process integrates real-time availability checks via asynchronous API calls to ensure accurate reservation status before confirming the booking.

Team Finder Feature

Functionality: This page lists registered teams based on sport filters. React dynamically renders the team cards, while backend Spring Boot services retrieve team information from MySQL databases. This page lists registered teams based on sport filters. React dynamically renders the team cards, while backend Spring Boot services retrieve team information from MySQL databases. Bootstrap manages the responsive grid layout, and API endpoints ensure up-to-date data delivery.

Player Finder Feature

Functionality: Players are searchable by location and role filters using React-managed state and UI components. Spring Boot APIs handle filtered queries against player data in MySQL. Bootstrap ensures responsive and accessible UI presentation.

Upcoming Tournaments Management

Functionality: Upcoming tournaments for cricket and football are displayed using React with Bootstrap grids. Tournament data is fetched through Spring Boot RESTful services querying MySQL, enabling dynamic updates and filtering. Sport-specific badges and date ranges are rendered client-side for clarity.

Deployment Architecture

The platform uses a multi-tier deployment architecture for scalability, maintainability, and security.

Application Servers:

Backend application logic is hosted on auto-scaling server groups such as Apache Tomcat that adjust instances based on demand.

Database:

Managed relational database service ensures reliability and scalability with MySQL.

Security and Performance:

Load balancers distribute traffic, and Spring Security secures communication protocols to protect user data.

WireFrame Link:

<https://www.figma.com/design/APFHR4Z2kO7odXDfifQM4j/FlexField?node-id=0-1&p=f&t=MPC4f2cTv4axSs46-0>

WorkFlow:**New User:**

ROOTPAGE → LOGIN → SIGNUP → LOGIN → HOMEPAGE

Existing User:

ROOTPAGE → LOGIN → HOMEPAGE

Individual Player:

HOMEPAGE → GAMES → CRICKET / FOOTBALL → TEAM FIND / BOOK TURF / UPCOMING TOURNAMENTS

Team Player:

HOMEPAGE → TEAM REGISTER → GAMES → CRICKET / FOOTBALL → PLAYER FIND / BOOK TURF / UPCOMING TOURNAMENTS

Conclusion

This Technical Design Document details the core functionality, features, and technology stack for FlexField's sports booking platform. It outlines key user flows, interface design, and backend integrations necessary to deliver a scalable and user-friendly experience. This living document will evolve as development progresses, ensuring alignment with project goals and requirements.

