Football Game Matching Platform - PRD

1. Executive Summary

A Next.js web application that connects football players with local games. Organizers can post games seeking players, while individuals can advertise their availability. The platform prioritizes upcoming games by proximity and implements payment collection to reduce no-shows.

2. Core Features

2.1 User Authentication

- **Email/Password Registration** - Standard signup flow with email verification
- **Social Login** - Google, Facebook, Apple OAuth integration via NextAuth.js
- **Two-Factor Authentication** - TOTP-based (authenticator apps)
- **SAML/OIDC Support** - Enterprise identity provider integration

2.2 Game Creation & Management

- **Game Posting** - Organizers create games with:
  - Location (address with map integration)
  - Date and time
  - Skill level (Beginner/Intermediate/Advanced)
  - Player capacity (total needed)
  - Price per player
  - Additional details/rules
- **Automatic Game States**:
  - OPEN - Accepting players
  - FULL - At capacity (hidden from search)
  - EXPIRED - Past start time (archived)
- **Dynamic Visibility** - Games reappear if players leave before capacity

2.3 Player Discovery

- **"Looking for Game" Posts** - Individual availability advertising
- **Geographic Prioritization** - Location-based game ranking
- **Time-based Sorting** - Upcoming games prioritized
- **Skill Level Filtering** - Match appropriate competition levels

2.4 Payment & Booking System

- **Upfront Payment** - Players pay before confirmation to reduce no-shows
- **Payment Integration** - Stripe for card processing
- **Refund Policy** - Automated refunds for cancelled games
- **Organizer Payouts** - Minus platform fee, transferred after game completion

2.5 Waitlist Management

- **Automatic Queuing** - Players join waitlist when games are full
- **Priority Notifications** - Waitlisted players notified first when spots open
- **Payment Hold** - Waitlist payments processed only when spot becomes available
- **Time-limited Offers** - 15-minute window to accept waitlist promotion

2.6 Communication System

- **Direct Messaging** - Organizer ↔ Player conversations
- **Game-specific Threads** - Context-aware messaging per game
- **Real-time Notifications** - Game updates, messages, waitlist promotions
- **Future: Group Chat** - All confirmed players + organizer

3. Technical Architecture

3.1 Tech Stack

- **Frontend**: Next.js 14 with App Router
- **Database**: PostgreSQL with PostGIS for geospatial queries
- **Authentication**: NextAuth.js
- **Payments**: Stripe Connect
- **Real-time**: Server-Sent Events
- **Deployment**: Vercel/AWS
- **Cache**: Redis for session and game state management

3.2 Core Data Models

```typescript
```

```typescript
// User
interface User {
  id: string
  email: string
  name: string
  location?: Coordinates
  skillLevel: 'BEGINNER' | 'INTERMEDIATE' | 'ADVANCED'
  stripeCustomerId?: string
}

// Game
interface Game {
  id: string
  organizerId: string
  title: string
  location: Location
  dateTime: Date
  skillLevel: SkillLevel
  maxPlayers: number
  pricePerPlayer: number
  status: 'OPEN' | 'FULL' | 'EXPIRED' | 'CANCELLED'
  stripeProductId: string
}

// Game Participation
interface GameParticipant {
  gameId: string
  playerId: string
  status: 'CONFIRMED' | 'WAITLISTED'
  paymentIntentId: string
  joinedAt: Date
  waitlistPosition?: number
}
```

3.3 API Patterns

RESTful Endpoints:

- `GET /api/games` – Search games with geo/time filters
- `POST /api/games/{id}/join` – Join game + payment processing
- `DELETE /api/games/{id}/leave` – Leave game + refund logic
- `GET /api/games/{id}/participants` – Game roster management

Real-time Events:

- Game capacity changes

- Waitlist promotions

- New messages

- Payment confirmations

4. User Experience Flow

4.1 Game Discovery

1. User sets location preferences

2. Platform shows nearby upcoming games

3. Filter by skill level, time, distance

4. Click to view game details

4.2 Joining a Game

1. Select game → View details

2. Click "Join Game" → Payment form

3. Stripe payment processing

4. Confirmation + calendar invite sent

5. Access to game chat thread

4.3 Waitlist Experience

1. Game full → "Join Waitlist" option

2. Payment authorization (not charged)

3. Position #X in waitlist notification

4. Spot opens → 15min notification to claim

5. Accept → Payment processes, spot confirmed

5. Business Logic

5.1 Payment Processing

- **Upfront Collection** - Reduces no-shows by requiring commitment

- **Platform Fee** - 5% transaction fee

- **Refund Windows**:
  - Full refund: >24hrs before game

  - 50% refund: 24-2hrs before game

  - No refund: <2hrs before game

- **Organizer Payouts** – Transferred 24hrs after game completion

5.2 Game State Transitions

```
OPEN → (max capacity reached) → FULL
FULL → (player leaves) → OPEN
OPEN/FULL → (start time passed) → EXPIRED
ANY → (organizer cancels) → CANCELLED
```

5.3 Waitlist Logic

- FIFO queue with timestamps
- Automatic promotion when spots available
- Payment authorization → charge conversion
- 15-minute acceptance window before next in line

6. Success Metrics

6.1 Core KPIs

- **Game Fill Rate** – % of games reaching capacity
- **No-show Rate** – Payment impact measurement
- **User Retention** – Monthly active participants
- **Geographic Coverage** – Games per city/region

6.2 Quality Metrics

- **Time to Fill** – Average hours from posting to full capacity
- **Repeat Participation** – Users joining multiple games
- **Organizer Satisfaction** – Successful game completion rate

7. Development Phases

Phase 1: MVP (8-10 weeks)

- Basic auth + game CRUD
- Payment integration
- Simple messaging
- Core search/filter

Phase 2: Enhanced Discovery (4-6 weeks)

- Advanced geolocation

* Waitlist system

* Mobile optimization

* Push notifications

Phase 3: Community Features (6-8 weeks)

* Group messaging

* User ratings/reviews

* Recurring games

* Analytics dashboard

8. Risk Mitigation

8.1 Technical Risks

* **Payment Failures** – Robust error handling + retry logic

* **Scale Issues** – Database optimization + caching strategy

* **Security** – PCI compliance + data encryption

8.2 Product Risks

* **Low Adoption** – Strong onboarding + referral incentives

* **Seasonal Usage** – Indoor/outdoor game categorization

* **Market Competition** – Unique features + local partnerships

9. Security Considerations

* PCI DSS compliance for payment processing

* GDPR compliance for EU users

* Rate limiting on game creation/joining

* Input validation and SQL injection prevention

* Secure location data handling

This PRD serves as the technical foundation for development planning and engineering implementation of the football game matching platform.