

Smørås Fotball Technical Documentation

Overview

This document provides a comprehensive overview of the Smørås Fotball application, a Django-based web application designed for football team management. The application supports multiple teams, players, matches, and includes features for match session management, player statistics, formation templates, and video clip creation.

Project Structure

The application follows the standard Django project structure with a main project folder ('smorasfotball') containing the Django project settings and multiple apps, primarily the 'teammanager' app which contains most of the business logic.

```
smorasfotball/
├── smorasfotball/ # Project settings and configuration
│   ├── settings.py # Django settings (database, authentication, etc.)
│   ├── urls.py # Main URL routing configuration
│   ├── wsgi.py # WSGI application entry point
│   └── asgi.py # ASGI application entry point
├── teammanager/ # Main application with team management logic
│   ├── models.py # Database models for teams, players, matches, etc.
│   ├── models_video.py # Video-related models separated for clarity
│   ├── views.py # Core views for basic functionality
│   └── views_*.py # Specialized view modules (auth, lineup, match management)
│   ├── forms.py # Form definitions for data input
│   ├── urls.py # URL routing for teammanager app
│   ├── admin.py # Django admin site configuration
│   ├── templates/ # HTML templates for rendering pages
│   │   ├── teammanager/ # App-specific templates
│   │   └── static/ # Static assets (CSS, JavaScript, images)
│   └── templates/ # Project-wide templates
│       ├── base.html # Base template with common structure
│       ├── registration/ # Authentication-related templates
│       └── static/ # Project-wide static files
├── locale/ # Translation files for internationalization
└── manage.py # Django command-line utility for administration
```

Additionally, the project includes various deployment and database management scripts in the root directory for automating deployment workflows and database migrations.

Key Technologies

The application leverages the following technologies:

- 1 1. **Django**: Core web framework for routing, database access, and template rendering
- 2 2. **PostgreSQL**: Primary database for production use
- 3 3. **Django ORM**: Object-Relational Mapping for database interactions
- 4 4. **JavaScript/AJAX**: For dynamic interactions on the client side
- 5 5. **Bootstrap**: CSS framework for responsive design
- 6 6. **i18n**: Internationalization support for multiple languages (English and Norwegian)

Database Schema

The database consists of several interconnected models:

Core Models

- 1 1. **User & UserProfile**
 - Django's built-in User model extended with a UserProfile
 - Supports multiple roles: Admin, Coach, Player
 - Includes approval workflow for new user registration
- 1 1. **Team**
 - Represents football teams
 - Contains name, description, and creation timestamp
- 1 1. **Player**
 - Stores player information (name, position, contact details)
 - Can be associated with teams and matches
- 1 1. **Match**
 - Records match details (teams, date, location, scores)
 - Links to the opponent team as a text field
 - Supports different match types (Friendly, League, Cup, Tournament)
- 1 1. **MatchAppearance**
 - Junction model connecting Players to Matches
 - Records statistics like minutes played, goals, assists, cards

Match Management Models

- 1 1. **MatchSession**
 - Manages active match tracking
 - Configures periods, period length, substitution intervals
 - Tracks match time and status
- 1 1. **PlayerSubstitution**
 - Records substitution events during a match
 - Stores which player went in, which went out, and when
- 1 1. **PlayingTime**
 - Tracks actual playing time for each player in a match
 - Indicates whether a player is currently on the pitch

Formation Models

- 1 1. **FormationTemplate**

- • Predefined formations (e.g., 4-4-2, 4-3-3)
- • Supports different team sizes (5er, 7er, 9er, 11er)
- 1 1. **LineupPosition**
- • Defines positions on a football pitch (GK, DEF, MID, FWD)
- 1 1. **Lineup** & **LineupPlayerPosition**
- • Creates lineups for matches with player positions
- • Stores visual coordinates for pitch positioning

Video Models

- 1 1. **VideoClip**
- • Stores videos of match highlights
- • Associates with players, matches, and game time
- 1 1. **HighlightReel** & **HighlightClipAssociation**
- • Compiles multiple clips into a highlight reel
- • Manages the order of clips in the reel

Key Features

User Authentication and Role Management

The application implements a multi-role authentication system:

- • **Admin**: Full system access, can approve users and manage all aspects
- • **Coach**: Can manage teams, players, and matches
- • **Player**: Limited access to view relevant information

New users must be approved by an admin before gaining full access to the system.

Team and Player Management

- • Create, edit, and delete teams
- • Add players with detailed information
- • Import players from Excel spreadsheets
- • Assign players to teams for specific matches
- • View player statistics and participation history

Match Management

- • Create matches with detailed information
- • Track scores and statistics
- • Assign and manage players for specific matches
- • Generate match reports

Real-time Match Session Management

The application provides comprehensive real-time match session management:

- • Start/stop match timing with accurate time tracking

- • Configure match periods and period length
- • Track player substitutions and playing time
- • Reset match time and substitution timers
- • Manually set match periods
- • View recommended substitutions based on playing time

The match session pitch view shows:

- • Current players on the pitch with their positions
- • Real-time match clock with period tracking
- • Substitution countdown timer
- • Quick substitution interface

Formation and Lineup Builder

- • Create formation templates for different team sizes
- • Build visual lineups with player positioning on a pitch
- • Save lineups as templates for future use
- • Export lineups to PDF

Video Management

- • Upload and manage video clips from matches
- • Create highlight reels from multiple clips
- • Associate clips with specific players and match events
- • Categorize clips by action type (goal, assist, save, etc.)

Statistics and Visualization

- • Dashboard with key statistics and recent activity
- • Player participation matrix showing which players have played together
- • Match count charts for all teams
- • Individual player statistics

Internationalization

The application supports multiple languages:

- • English
- • Norwegian
- • Language selection with flag icons

Technical Implementation Details

Database Configuration

The application is designed to use PostgreSQL in production and SQLite for development. There's a sophisticated system for detecting the environment and configuring the appropriate database:

- 1 1. It checks for the `DATABASE_URL` environment variable to determine if PostgreSQL is available
- 2 2. In production, it automatically tries to create a PostgreSQL database if not configured
- 3 3. It includes failsafe mechanisms to prevent data loss during deployment

Match Time Management

The match time management system is particularly complex:

- 1 1. Server-side tracking of elapsed time with persistence across page reloads
- 2 2. Client-side clock for smooth updates without constant server requests
- 3 3. Support for stopping and resuming match time
- 4 4. Period tracking with accurate elapsed time calculations
- 5 5. Substitution timer with countdown display

The `update_playing_times` endpoint returns:

- • Player-specific playing times
- • Match status information
- • Elapsed time from previous periods
- • Current period information

CSRF Protection and AJAX Requests

The application implements special handling for CSRF protection in AJAX requests:

- • Custom CSRF exemption for critical endpoints
- • CSRF token included in AJAX request headers
- • Support for the Replit hosting environment

Deployment System

The project includes a comprehensive deployment system:

- • Automatic database backup before deployment
- • Database migration scripts for switching between SQLite and PostgreSQL
- • Production environment detection and configuration
- • Restoration process for deployed databases

Code Execution Flow

Match Session Flow

When a match session is active:

- 1 1. ****Initialization****:
 - • The `match_session_pitch_view` renders the interface
 - • JavaScript initializes client-side tracking
- 1 1. ****Match Start****:
 - • Server records `start_time` when match is started

- • Client begins real-time clock updates
- 1 1. ****During Match****:
- • Client updates display every 5 seconds (1 second during critical moments)
- • update_playing_times API endpoint provides current status
- • Substitutions are recorded and player times updated
- 1 1. ****Match Control****:
- • match_session_stop pauses the match
- • match_session_start resumes with accurate timing
- • reset_match_time resets the current period
- • set_match_period changes the period with proper time accounting

Player Substitution Flow

- 1 1. ****Normal Substitution****:
- • Coach selects players to substitute
- • substitution_create endpoint processes the change
- • Server updates all relevant playing time records
- 1 1. ****Quick Substitution****:
- • ajax_quick_sub provides a streamlined interface
- • Multiple players can be substituted at once
- 1 1. ****Recommended Substitutions****:
- • get_sub_recommendations suggests substitutions based on playing time
- • Algorithm balances playing time and positions

Conclusion

The Smørås Fotball application is a comprehensive solution for football team management with sophisticated features for match tracking, player management, and statistical analysis. The system is designed to be user-friendly while providing powerful tools for coaches and administrators.

The code architecture follows Django best practices with modular components, clear separation of concerns, and robust data models. The extensive use of AJAX and real-time updates creates a dynamic user experience, while the deployment scripts ensure reliable operation in production environments.