|  |
| --- |
| MatchAware |
| Architecture Design and Software Structure |
| Patrick Borrelli |

Table of Contents

[Introduction 3](#_Toc471559720)

[Design and Implementation 3](#_Toc471559721)

[MatchAware REST API Specification 3](#_Toc471559722)

[Access Request 3](#_Toc471559723)

[Age Group 3](#_Toc471559724)

[Change Request 4](#_Toc471559725)

[Club 4](#_Toc471559726)

[Club Member 4](#_Toc471559727)

[Event 5](#_Toc471559728)

[Event Type 5](#_Toc471559729)

[Facility 5](#_Toc471559730)

[Field 6](#_Toc471559731)

[Field Size 6](#_Toc471559732)

[League 6](#_Toc471559733)

[League Team 6](#_Toc471559734)

[League Type 7](#_Toc471559735)

[Message 7](#_Toc471559736)

[Notification 7](#_Toc471559737)

[Organization 8](#_Toc471559738)

[Reschedule Rule 8](#_Toc471559739)

[Role 8](#_Toc471559740)

[Rule 9](#_Toc471559741)

[Team 9](#_Toc471559742)

[Team Coach 9](#_Toc471559743)

[Team Player 10](#_Toc471559744)

[User 10](#_Toc471559745)

[User Invite 10](#_Toc471559746)

[User Role 11](#_Toc471559747)

[User Settings 11](#_Toc471559748)

[Front-End Architecture Design 12](#_Toc471559749)

[Web Front-End 12](#_Toc471559750)

[Mobile Front-End 12](#_Toc471559751)

[Database Schema 13](#_Toc471559752)

[Access Request 13](#_Toc471559753)

[Age Group 13](#_Toc471559754)

[Change Request 13](#_Toc471559755)

[Club 14](#_Toc471559756)

[Club Member 14](#_Toc471559757)

[Event 14](#_Toc471559758)

[Event Type 15](#_Toc471559759)

[Facility 15](#_Toc471559760)

[Field 16](#_Toc471559761)

[Field Size 16](#_Toc471559762)

[League 16](#_Toc471559763)

[League Team 17](#_Toc471559764)

[League Type 17](#_Toc471559765)

[Message 17](#_Toc471559766)

[Notification 17](#_Toc471559767)

[Organization 18](#_Toc471559768)

[Reschedule Rule 18](#_Toc471559769)

[Role 18](#_Toc471559770)

[Rule 19](#_Toc471559771)

[Team 19](#_Toc471559772)

[Team Coach 19](#_Toc471559773)

[Team Player 20](#_Toc471559774)

[User 20](#_Toc471559775)

[User Invite 20](#_Toc471559776)

[User Role 21](#_Toc471559777)

[User Settings 21](#_Toc471559778)

[Communication 23](#_Toc471559779)

[Conclusions 23](#_Toc471559780)

[References 24](#_Toc471559781)

# Introduction

MatchAware is a comprehensive and easy to use Field Scheduling solution for youth sports clubs of all sizes. MatchAware provides an easy set of interfaces to provide scheduling of field and facility resources for league games, practice sessions, scrimmages, and training camps for any number of teams, as well as supporting workflow based change requests, rescheduling notifications, and referee and trainer assignments. The initial release of the software is Soccer Club specific, and future releases will add functionality and terminology applicable to other youth sports clubs.

# Design and Implementation

MatchAware REST API Specification

The complete API specification is presented here. Due to the short development lifecycle, some functionality will not be implemented in the first release of the MatchAware application. Indications of all deferred functionality will be called out in the conclusion of this document. Endpoints are presented alphabetically below.

### Access Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Description** | **Returned** | **Body** |
| GET | /access\_request | Retrieve all access requests | ACCESS\_REQUEST | optional |
| POST | /access\_request | Create access request | ACCESS\_REQUEST | yes |
| DELETE | /access\_request | Delete all access requests |  | optional |
| GET | /access\_request/:access\_requestId | Retrieve access request by ID | ACCESS\_REQUEST |  |
| PUT | /access\_request/:access\_requestId | Update access request by ID | ACCESS\_REQUEST |  |
| DELETE | /access\_request/:access\_requestId | Delete access request by ID |  |  |
| GET | /access\_request/findByReviewer/:userId | Retrieve all access requests pending acknowledgement by the specified user | ACCESS\_REQUEST |  |
| GET | /access\_request/findByStatus/:status | Retrieve all access requests in the specified status | ACCESS\_REQUEST |  |

### Age Group

### 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /age\_group | Retrieve all age groups | AGE\_GROUP |  |
| POST | /age\_group | Create age group | AGE\_GROUP | yes |
| DELETE | /age\_group | Delete all age groups |  |  |
| GET | /age\_group/:age\_groupId | Retrieve age group by ID | AGE\_GROUP |  |
| PUT | /age\_group/:age\_groupId | Update age group by ID | AGE\_GROUP | yes |
| DELETE | /age\_group/:age\_groupId | Delete age group by ID |  |  |

### Change Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /change\_request | Retrieve all change requests | CHANGE\_REQUEST | optional |
| POST | /change\_request | Create change request | CHANGE\_REQUEST | yes |
| DELETE | /change\_request | Delete all change requests |  | optional |
| GET | /change\_request/:change\_requestId | Retrieve change request by ID | CHANGE\_REQUEST |  |
| PUT | /change\_request/:change\_requestId | Update change request by ID | CHANGE\_REQUEST | yes |
| DELETE | /change\_request/:change\_requestId | Delete change request by ID |  |  |
| GET | /change\_request/findByApprover/:userId | Retrieve all change requests for an approver | CHANGE\_REQUEST |  |
| GET | /change\_request/findBySubmitter/:userId | Retrieve all change requests for an approver | CHANGE\_REQUEST |  |

### Club

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /club | Retrieve all clubs | CLUB |  |
| POST | /club | Create club | CLUB | yes |
| DELETE | /club | Delete all clubs |  |  |
| GET | /club/:clubId | Retrieve clubs by user ID | CLUB |  |
| PUT | /club/:clubId | Update club by ID | CLUB | yes |
| DELETE | /club/:clubId | Delete club by ID |  |  |

### Club Member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| POST | /club\_member | Add user to club | CLUB\_MEMBER | yes |
| DELETE | /club\_member | Remove user from club |  | yes |
| GET | /club\_member/:clubId | Retrieve all user members of club | USER |  |
| GET | /club\_member/:userId | Retrieve all clubs user belongs to | CLUB |  |

### Event

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /event | Retrieve all events | EVENT | optional |
| POST | /event | Create event | EVENT | yes |
| DELETE | /event | Delete all events |  | optional |
| GET | /event/:eventId | Retrieve event by ID | EVENT |  |
| PUT | /event/:eventId | Update event by ID | EVENT | yes |
| DELETE | /event/:eventId | Delete event by ID |  |  |
| GET | /event/findByReferee/:userId | Retrieve all events for a referee | EVENT |  |
| GET | /event/findByStatus/:status | Retrieve all events with the provided status | EVENT |  |
| GET | /event/findByTeam/:teamId | Retrieve all events for a team | EVENT |  |
| GET | /event/findByTrainer/:userId | Retrieve all events for a trainer | EVENT |  |

### Event Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /event\_type | Retrieve all event types | EVENT\_TYPE |  |
| POST | /event\_type | Create event type | EVENT\_TYPE | yes |
| DELETE | /event\_type | Delete all event types |  |  |
| GET | /event\_type/:event\_typeId | Retrieve event type by ID | EVENT\_TYPE |  |
| PUT | /event\_type/:event\_typeId | Update event type by ID | EVENT\_TYPE | yes |
| DELETE | /event\_type/:event\_typeId | Delete event type by ID |  |  |

### Facility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /facility | Retrieve all facilities | FACILITY | optional |
| POST | /facility | Create facility | FACILITY | yes |
| DELETE | /facility | Delete all facilities |  | optional |
| GET | /facility/:facilityId | Retrieve facility by ID | FACILITY |  |
| PUT | /facility/:facilityId | Update facility by ID | FACILITY | yes |
| DELETE | /facility/:facilityId | Delete facility by ID |  |  |

### Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /field | Retrieve all fields | FIELD | optional |
| POST | /field | Create field | FIELD | yes |
| DELETE | /field | Delete all fields |  | optional |
| GET | /field/:fieldId | Retrieve field by ID | FIELD |  |
| PUT | /field/:fieldId | Update field by ID | FIELD | yes |
| DELETE | /field/:fieldId | Delete field by ID |  |  |
| GET | /field/findAvailableByQuery | Retrieve all available fields based on provided query parameters | FIELD | yes |

### Field Size

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /field\_size | Retrieve all field sizes | FIELD\_SIZE |  |
| POST | /field\_size | Create field size | FIELD\_SIZE | yes |
| DELETE | /field\_size | Delete all field sizes |  |  |
| GET | /field\_size/:field\_sizeId | Retrieve field size by ID | FIELD\_SIZE |  |
| PUT | /field\_size/:field\_sizeId | Update field size by ID | FIELD\_SIZE | yes |
| DELETE | /field\_size/:field\_sizeId | Delete field size by ID |  |  |

### League

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /league | Retrieve all leagues | LEAGUE |  |
| POST | /league | Create league | LEAGUE |  |
| DELETE | /league | Delete all leagues |  |  |
| GET | /league/:leagueId | Retrieve league by ID | LEAGUE |  |
| PUT | /league/:leagueId | Update league by ID | LEAGUE | yes |
| DELETE | /league/:leagueId | Delete league by ID |  |  |

### League Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| POST | /league\_team | Add team to league | LEAGUE\_TEAM | yes |
| DELETE | /league\_team | Remove team from league |  | yes |
| GET | /league\_team/:leagueId | Retrieve all teams in league | TEAM |  |
| GET | /league\_team/:teamId | Retrieve all leagues team plays in | LEAGUE |  |

### League Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /league\_type | Retrieve all league types | LEAGUE\_TYPE |  |
| POST | /league\_type | Create league type | LEAGUE\_TYPE |  |
| DELETE | /league\_type | Delete all league types |  |  |
| GET | /league\_type/:league\_typeId | Retrieve league type by ID | LEAGUE\_TYPE |  |
| PUT | /league\_type/:league\_typeId | Update league type by ID | LEAGUE\_TYPE | yes |
| DELETE | /league\_type/:league\_typeId | Delete league type by ID |  |  |

### Message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /message | Retrieve all messages | MESSAGE | optional |
| POST | /message | Create message | MESSAGE | yes |
| DELETE | /message | Delete all messages |  |  |
| GET | /message/:messageId | Retrieve message by ID | MESSAGE |  |
| PUT | /message/:messageId | Update message by ID | MESSAGE | yes |
| DELETE | /message/:messageId | Delete message by ID |  |  |
| GET | /message/findByEvent/:eventId | Retrieve all messages for this event | MESSAGE |  |
| GET | /message/findByRecipient/:userId | Retrieve all messages for this user | MESSAGE |  |

### Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /notification | Retrieve all notifications | NOTIFICATION | optional |
| POST | /notification | Create notification | NOTIFICATION | yes |
| DELETE | /notification | Delete all notifications |  | optional |
| GET | /notification/:notificationId | Retrieve notification by ID | NOTIFICATION |  |
| PUT | /notification/:notificationId | Update notification by ID | NOTIFICATION | yes |
| DELETE | /notification/:notificationId | Delete notification by ID |  |  |
| GET | /notification/findByRecipient/:userId | Retrieve all notifications for this user | NOTIFICATION |  |

### Organization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /organization | Retrieve all organizations | ORGANIZATION |  |
| POST | /organization | Create organization | ORGANIZATION | yes |
| DELETE | /organization | Delete all organizations |  |  |
| GET | /organization/:clubId | Retrieve all organizations working for the provided club | ORGANIZATION |  |
| GET | /organization/:organizationId | Retrieve organization by ID | ORGANIZATION |  |
| PUT | /organization/:organizationId | Update organization by ID | ORGANIZATION | yes |
| DELETE | /organization/:organizationId | Delete organization by ID |  |  |

### Reschedule Rule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /reschedule\_rule | Retrieve all reschedule rules | RESCHEDULE\_RULE |  |
| POST | /reschedule\_rule | Create reschedule rule | RESCHEDULE\_RULE | yes |
| DELETE | /reschedule\_rule | Delete all reschedule rules |  |  |
| GET | /reschedule\_rule/:reschedule\_ruleId | Retrieve reschedule rule by ID | RESCHEDULE\_RULE |  |
| PUT | /reschedule\_rule/:reschedule\_ruleId | Update reschedule rule by ID | RESCHEDULE\_RULE | yes |
| DELETE | /reschedule\_rule/:reschedule\_ruleId | Delete reschedule rule by ID |  |  |

### Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /role | Retrieve all roles | ROLE |  |
| POST | /role | Create role | ROLE | yes |
| DELETE | /role | Delete all roles |  |  |
| PUT | /role/:roleId | Update role by ID | ROLE | yes |
| DELETE | /role/:roleId | Delete role by ID |  |  |
| GET | /role/:userId | Retrieve roles by user ID | ROLE |  |

### Rule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /rule | Retrieve all rules | RULE | optional |
| POST | /rule | Create rule | RULE | yes |
| DELETE | /rule | Delete all rules |  | optional |
| GET | /rule/:ruleId | Retrieve rule by ID | RULE |  |
| PUT | /rule/:ruleId | Update rule by ID | RULE | yes |
| DELETE | /rule/:ruleId | Delete rule by ID |  |  |

### Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /team | Retrieve all teams | TEAM | optional |
| POST | /team | Create team | TEAM | yes |
| DELETE | /team | Delete all teams |  | optional |
| GET | /team/:teamId | retrieve by ID | TEAM |  |
| PUT | /team/:teamId | Update team by ID | TEAM | yes |
| DELETE | /team/:teamId | Delete team by ID |  |  |
| GET | /team/findByQuery | Retrieve all teams based on parameters | TEAM | yes |

### Team Coach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| POST | /team\_coach | Add coach to team | TEAM\_COACH | yes |
| DELETE | /team\_coach | Remove coach from team |  | yes |
| GET | /team\_coach/:coachId | Retrieve all teams associated with coach | TEAM |  |
| GET | /team\_coach/:teamId | Retrieve all coaches on a team | USER |  |

### Team Player

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| POST | /team\_player | Add player to team | TEAM\_PLAYER | yes |
| DELETE | /team\_player | Remove player from team |  | yes |
| GET | /team\_player/:teamId | Retrieve all players on team | USER |  |
| GET | /team\_player/:userId | Retrieve all teams player belongs to | TEAM |  |
| GET | /team\_player/getRosterSize/:teamId | Get a count of all players associated with a team | INTEGER |  |

### User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /user | Retrieve a list of users | USER | optional |
| GET | /user/:userId | retrieve user by ID | USER |  |
| PUT | /user/:userId | Update user by ID | USER | yes |
| DELETE | /user/:userId | Delete a user account by ID |  |  |
| POST | /user/facebook | log in using a facebook account | USER | yes |
| GET | /user/findByClub/:clubId | Get all users in a club | USER |  |
| GET | /user/findByOrganization/:orgId | Get all users working for a training organization | USER |  |
| GET | /user/findByRefereeAssignor/:userId | Get all referees reporting to a ref assignor | USER |  |
| POST | /user/login | log in to a user account | USER | yes |
| POST | /user/logout | log out of a user account |  |  |
| POST | /user/register | Create a user account | USER | yes |

### User Invite

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /user\_invite | Retrieve all user invites | USER\_INVITE |  |
| POST | /user\_invite | Create user invites | USER\_INVITE | yes |
| DELETE | /user\_invite | Delete all user invites |  |  |
| GET | /user\_invite/:user\_inviteId | Retrieve user invite by ID | USER\_INVITE |  |
| PUT | /user\_invite/:user\_inviteId | Update user invite by ID | USER\_INVITE | yes |
| DELETE | /user\_invite/:user\_inviteId | Delete user invite by ID |  |  |
| GET | /user\_invite/findByKey/:inviteKey | Retrieve invite by key | USER\_INVITE |  |

### User Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| POST | /user\_role | Add role to user | USER\_ROLE | yes |
| GET | /user\_role/:roleId | Retrieve all users in role | USER |  |
| GET | /user\_role/:userId | Retrieve all roles for user | ROLE |  |
| DELETE | /user\_role/:userId | Remove all roles for user |  |  |
| DELETE | /user\_role/:userId/:roleId | Remove specified role from user |  |  |

### User Settings

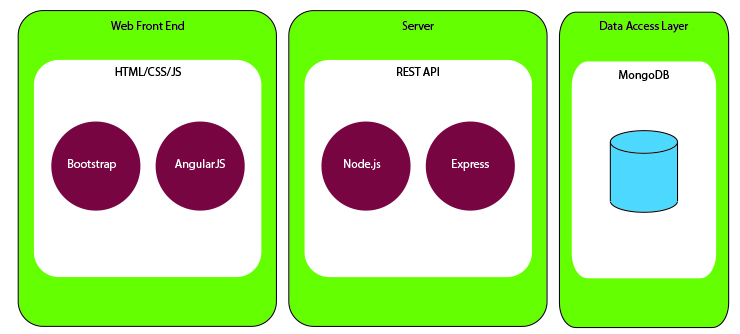
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **URL** | **Result** | **Returned** | **Body** |
| GET | /user\_settings | Retrieve all user settings | USER\_SETTINGS |  |
| POST | /user\_settings | Create user setting | USER\_SETTINGS | yes |
| DELETE | /user\_settings | Delete all user settings |  |  |
| GET | /user\_settings/:user\_settingsId | Retrieve user setting by ID | USER\_SETTINGS |  |
| PUT | /user\_settings/:user\_settingsId | Update user setting by ID | USER\_SETTINGS | yes |
| DELETE | /user\_settings/:user\_settingsId | Delete user setting by ID |  |  |
| GET | /user\_settings/findByUser/:userId | Retrieve user settings for this user | USER\_SETTINGS |  |

Front-End Architecture Design

The MatchAware application provides the user with two front ends, a fully-featured web interface, and a feature limited mobile interface. Both front ends interact with the same back end REST API and MongoDB instance.

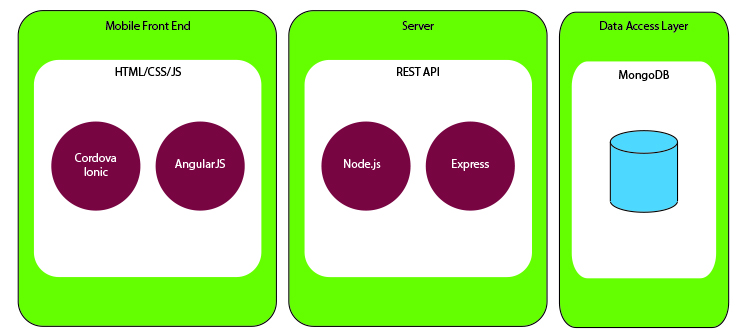
### Web Front-End

The web based front end for the MatchAware application is implemented as HTML/CSS/JavaScript, utilizing the Bootstrap framework and AngularJS.



### Mobile Front-End

The mobile front end for the MatchAware application is implemented as HTML/CSS/JavaScript, using the Ionic framework, Cordova and AngularJS, and will be available in Android and Apple platforms.

****

Database Schema

The database utilized is MongoDB, and the MatchAware schema is implemented using the Mongoose Schema module. The database schema is presented here both in tabular form to indicate all fields, and in a graphic form to indicate relationships.

### Access Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ACCESS\_REQUESTS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| USER | \_id | Y |  |  |
| CLUB | \_id | Y |  |  |
| ROLE | \_id | Y |  |  |
| TEAM | \_id |  |  |  |
| OWNER | \_id |  | coach, ref/train/club admin |  |
| STATUS | String |  | ['SENT', 'PENDING', 'ACCEPTED', 'REJECTED'] |  |

### Age Group

### 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AGE\_GROUPS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| BIRTH\_YEAR | String | Y |  |  |
| SOCCER\_YEAR | String | Y | must be upleveled each August 1st |  |
| NAME | String | Y |  |  |

### Change Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHANGE\_REQUESTS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| ORIGINAL\_EVENT | \_id | Y |  |  |
| CHANGED\_EVENT | \_id | Y |  |  |
| STATUS | String |  | enum |  |
| APPROVER | \_id | Y |  |  |
| SUBMITTER | \_id | Y |  |  |

### Club

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLUBS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  |  |
| SHORT\_NAME | String | Y |  |  |

### Club Member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLUB\_MEMBERS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| CLUB | \_id | Y |  |  |
| USER | \_id | Y |  |  |

### Event

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EVENTS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String |  |  |  |
| TYPE | \_id | Y |  |  |
| AWAY\_TEAM | \_id |  |  |  |
| HOME\_TEAM | \_id |  |  |  |
| LEAGUE | \_id |  |  |  |
| OFFICIALS | [USER] |  | array of referees |  |
| MAX\_AGE\_GROUP | \_id |  |  |  |
| MIN\_AGE\_GROUP | \_id |  |  |  |
| ORGANIZATION | \_id |  |  |  |
| TRAINERS | [USER] |  |  |  |
| DATE | String |  |  |  |
| END\_TIME | timestamp |  |  |  |
| FIELD | \_id |  |  |  |
| RECURRING | boolean |  |  |  |
| START\_TIME | timestamp |  |  |  |
| STATUS | String |  | enum |  |
| CREATOR | \_id |  |  |  |
| WEEKDAY | String |  | enum - for preseason requests |  |

### Event Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EVENT\_TYPES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  | Y |

### Facility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FACILITIES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  | Y |
| SHORT\_NAME | String | Y |  |  |
| ADDRESS | String | Y |  |  |
| CITY | String | Y |  |  |
| STATE | String | Y |  |  |
| POSTAL\_CODE | String | Y |  |  |
| GPS\_LOCATOR | String |  |  |  |
| SUN\_START\_TIME | Time |  |  |  |
| SUN\_STOP\_TIME | Time |  |  |  |
| MON\_START\_TIME | Time |  |  |  |
| MON\_STOP\_TIME | Time |  |  |  |
| TUE\_START\_TIME | Time |  |  |  |
| TUE\_STOP\_TIME | Time |  |  |  |
| WED\_START\_TIME | Time |  |  |  |
| WED\_STOP\_TIME | Time |  |  |  |
| THU\_START\_TIME | Time |  |  |  |
| THU\_STOP\_TIME | Time |  |  |  |
| FRI\_START\_TIME | Time |  |  |  |
| FRI\_STOP\_TIME | Time |  |  |  |
| SAT\_START\_TIME | Time |  |  |  |
| SAT\_STOP\_TIME | Time |  |  |  |
| INDOOR | Boolean |  |  |  |

### Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELDS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  | Y |
| FACILITY | \_id | Y |  |  |
| SIZE | \_id | Y |  |  |
| LIGHTS | boolean |  |  |  |
| CONDITION | integer |  |  |  |
| GAME | boolean |  |  |  |
| PRACTICE | boolean |  |  |  |
| TOURNAMENT | boolean |  |  |  |
| SURFACE | enum |  | ['GRASS','TURF'] |  |

### Field Size

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD\_SIZES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  | Y |
| MIN\_LENGTH | integer |  |  |  |
| MAX\_LENGTH | integer |  |  |  |
| MIN\_WIDTH | integer |  |  |  |
| MAX\_WIDTH | integer |  |  |  |

### League

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LEAGUES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  |  |
| SHORT\_NAME | String | Y |  |  |
| MIN\_AGE\_GROUP | \_id |  |  |  |
| MAX\_AGE\_GROUP | \_id |  |  |  |
| TYPE | \_id | Y |  |  |
| RESCHEDULE\_RULE | \_id | Y |  |  |

### League Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LEAGUE\_TEAMS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| LEAGUE | \_id | Y |  |  |
| TEAM | \_id | Y |  |  |

### League Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LEAGUE\_TYPES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  |  |

### Message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MESSAGES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| EVENT | \_id |  |  |  |
| TEXT | String |  |  |  |
| SENDER | \_id |  |  |  |
| RECIPIENT | \_id |  |  |  |

### Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOTIFICATIONS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| STATUS | String | Y | enum |  |
| TYPE | String | Y | enum |  |
| EVENT | \_id |  |  |  |
| TEXT | String |  |  |  |
| CREATED\_DATE | timestamp |  |  |  |
| SENDER | \_id |  |  |  |
| RECIPIENT | \_id |  |  |  |

### Organization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ORGANIZAZTIONS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  |  |
| SHORT\_NAME | String |  |  |  |
| ADMINISTRATOR | \_id |  |  |  |
| CLUB\_AFFILIATION | [\_id] |  |  |  |
| STAFF | [\_id] |  |  |  |

### Reschedule Rule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RESCHEDULE\_RULES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| TIMESPAN\_DAYS | integer | Y |  |  |
| CONSEQUENCE | enum |  | ['FORFEIT', |  |
|  |  |  | 'FINE', |  |
|  |  |  | 'FORFEIT\_FINE'] |  |
| FINE | float |  | 0 |  |

### Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ROLES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y | ['CLUB\_ADMIN', | N |
|  |  |  | 'FIELD\_ADMIN', |  |
|  |  |  | 'REFEREE\_ASSIGNOR', |  |
|  |  |  | 'TRAINING\_ADMIN', |  |
|  |  |  | 'COACH', |  |
|  |  |  | 'TRAINER', |  |
|  |  |  | 'REFEREE', |  |
|  |  |  | PARENT\_PLAYER'] |  |

### Rule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RULES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| LEAGUE | \_id |  |  |  |
| AGE\_GROUP | \_id |  |  |  |
| DURATION\_MINUTES | integer |  |  |  |
| FIELDED\_PLAYERS | integer |  |  |  |
| GOALKEEPER | boolean |  |  |  |
| MAX\_FIELD\_LENGTH | integer |  |  |  |
| MAX\_FIELD\_WIDTH | integer |  |  |  |
| GOAL\_WIDTH\_FT | integer |  |  |  |
| GOAL\_HEIGHT\_FT | integer |  |  |  |
| NUM\_PERIODS | integer |  |  |  |
| PERIOD\_DURATION\_MINUTES | integer |  |  |  |
| BALL\_SIZE | integer |  |  |  |
| OFFSIDE | boolean |  |  |  |
| HEADING | boolean |  |  |  |
| BUILD\_OUT\_LINE | boolean |  |  |  |

### Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEAMS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| NAME | String | Y |  | Y |

### Team Coach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEAM\_COACHES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| TEAM | \_id | Y |  |  |
| COACH | \_id | Y |  |  |

### Team Player

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEAM\_PLAYERS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| TEAM | \_id | Y |  |  |
| PLAYER | \_id | Y |  |  |

### User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USERS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| FIRST\_NAME | String | Y |  |  |
| LAST\_NAME | String | Y |  |  |
| USERNAME | String | Y |  |  |
| PASSWORD | String |  |  |  |
| EMAIL\_ADDRESS | String | Y |  |  |
| PROFILE\_IMAGE | String |  |  |  |
| ADDRESS | String |  |  |  |
| CITY | String |  |  |  |
| STATE | String |  |  |  |
| COUNTRY | String |  |  |  |
| POSTAL\_CODE | String |  |  |  |
| ADMIN | Boolean |  | FALSE |  |
| CERTIFICATIONS | [\_id] |  | referee certifications |  |
| LICENCES | [\_id] |  | coaching licenses |  |
| OAUTH\_ID | String |  |  |  |
| OAUTH\_TOKEN | String |  |  |  |

### User Invite

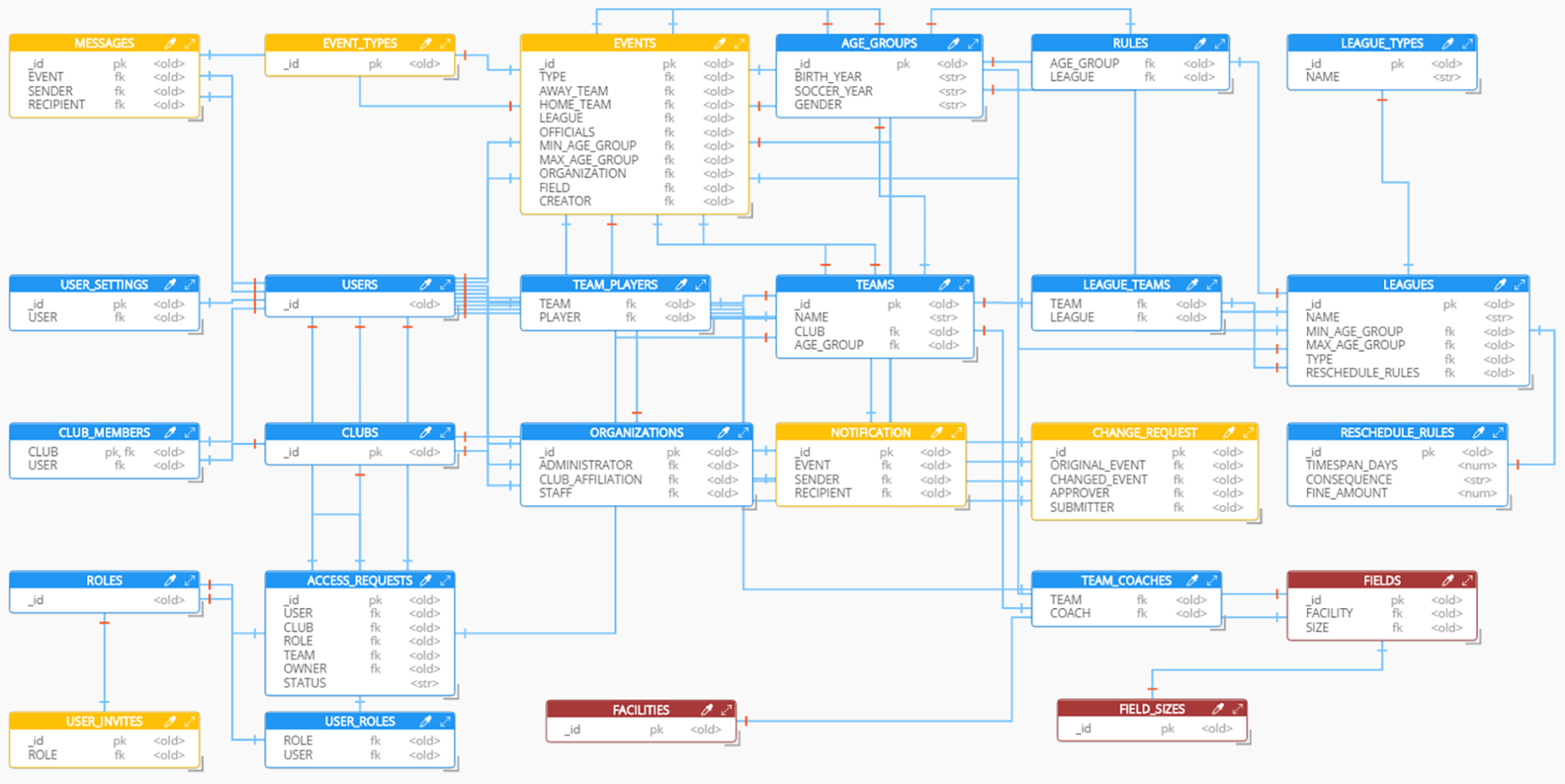
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USER\_INVITES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| INVITE\_KEY | numerical | Y |  | Y |
| EMAIL | String |  |  |  |
| MOBILE | String |  |  |  |
| ROLE | \_id | Y |  |  |
| STATUS | String |  | enum |  |

### User Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USER\_ROLES** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ROLE\_ID | \_id | Y |  | N |
| USER\_ID | \_id | Y |  | N |

### User Settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USER\_SETTINGS** | | | | |
| **field** | **data type** | **required** | **default** | **unique** |
| ID | \_id | Y | mongo generated | Y |
| USER | \_id |  |  |  |
| SCHEDULE\_VIEW\_RANGE | String |  | enum |  |

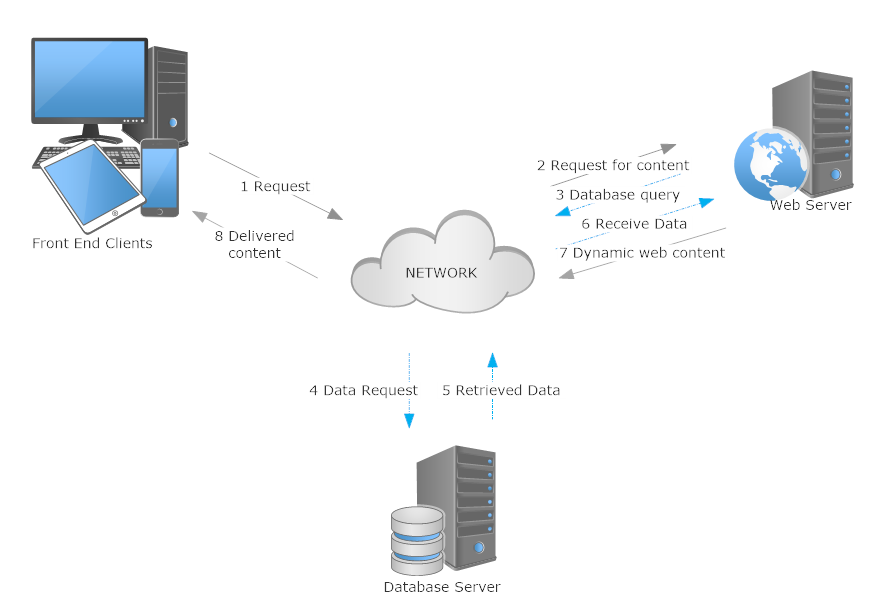


Communication

All communications between the client and server sides in the MatchAware application involve AngularJS on the client making requests to Node.js Express on the server side.

AngularJS will make HTTP requests, which are received and processed by Express on the server side, and Express sends HTTP responses to the requesting client, which are then handled in the client code by AngularJS.

All communications are defined in the REST API provided earlier in this document.



# Conclusions

MatchAware will rollout with its first release containing minimally the Club Admin, Field Admin, Coach and Parent/Player functionality for both the web and mobile front ends. Later releases of the product will add on the Referee Assignor, Referee, Training Admin and Trainer functionalities.

# References

Communication diagram designed in SmartDraw - https://www.smartdraw.com/  
Architecture diagrams designed in Microsoft Illustrator

Data modeling diagram designed in Hackolade - <http://hackolade.com/>