MatchAware

Architecture Design and Software Structure

Patrick Borrelli

Table of Contents

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

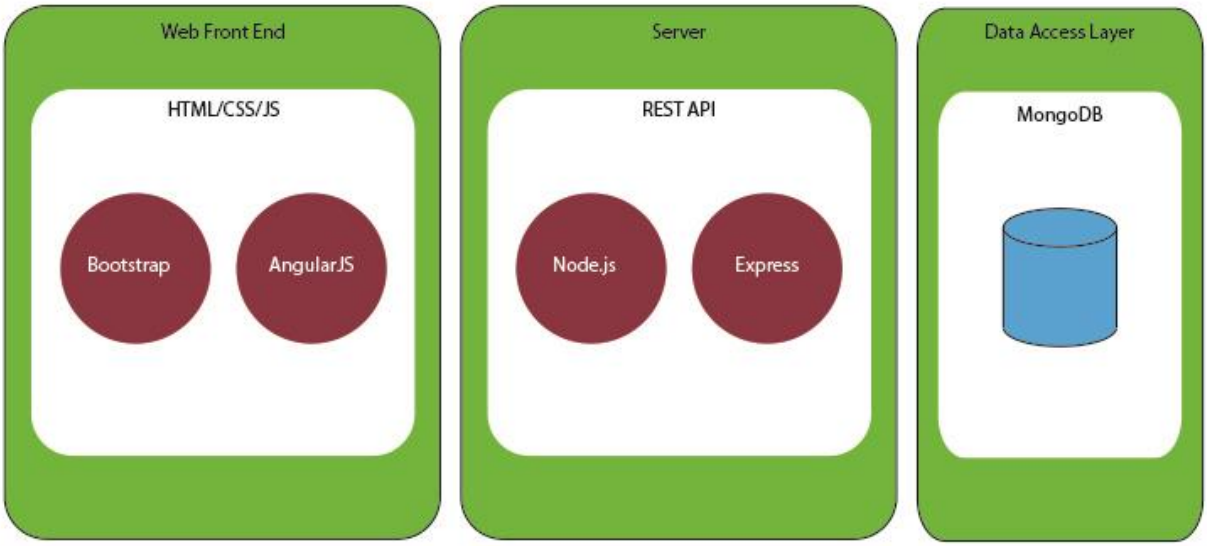
Action	URL	Result	Returned	Body
GET	/user_settings	Retrieve all user settings	USER_SETTINGS	
POST	/user_settings	Create user setting	USER_SETTINGS	yes
DELETE	/user_settings	Delete all user settings		
GET	/user_settings/:user_settingsId	Retrieve user setting by ID	USER_SETTINGS	
PUT	/user_settings/:user_settingsId	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.

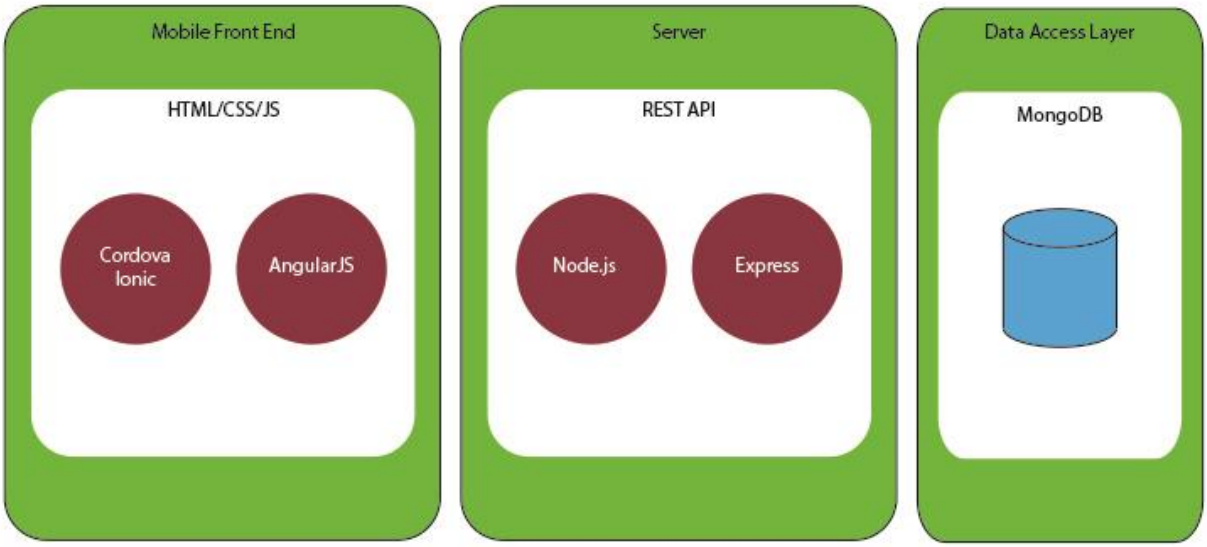
Web Front-End

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

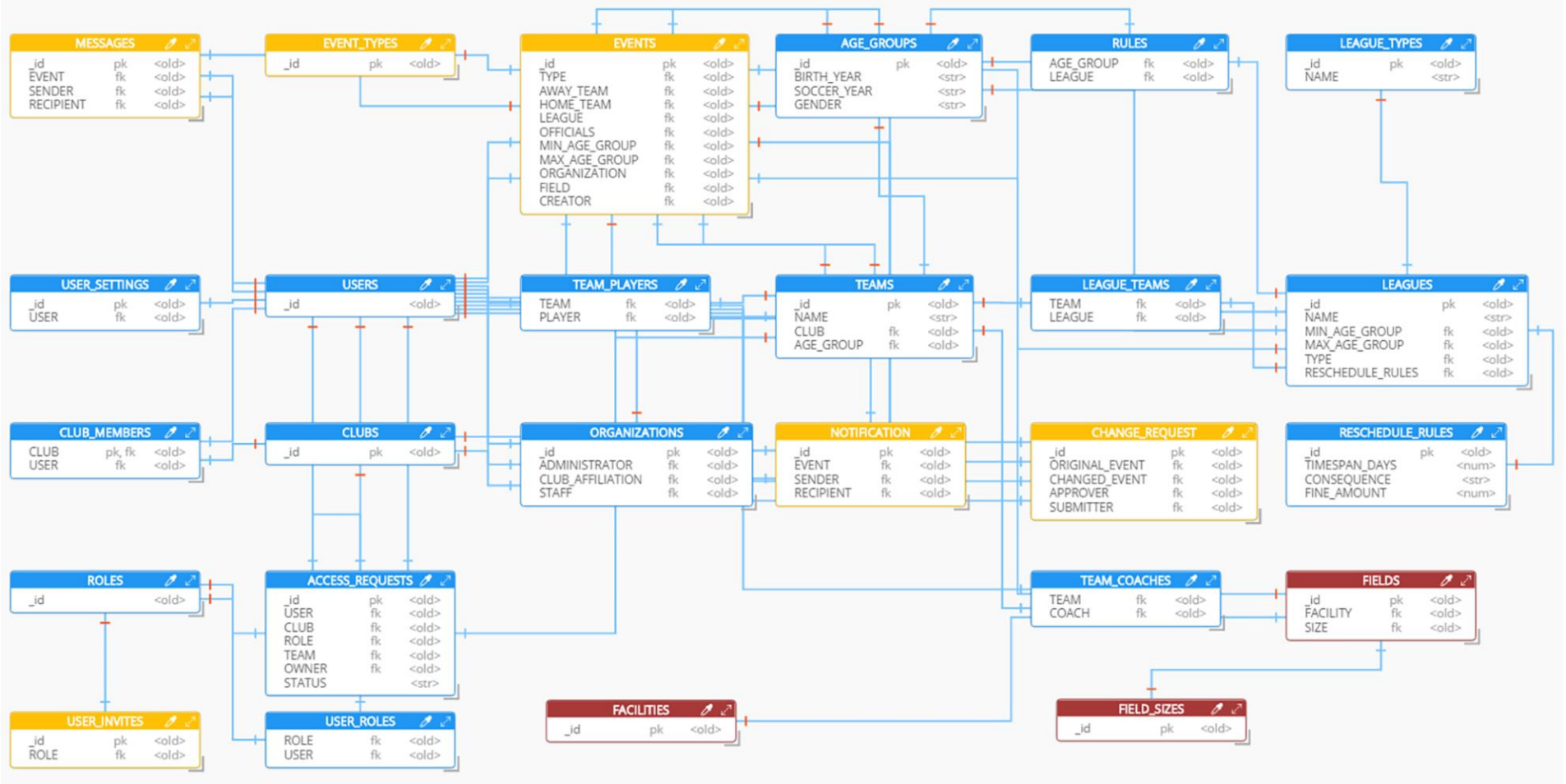
USER_INVITES				
field	**data type**	**required**	**default**	**unique**
ID	_id	Y	mongo generated	Y
INVITE_KEY	numerical	Y		Y
EMAIL	String			
MOBILE	String			
ROLE	_id	Y		
STATUS	String		enum	

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	

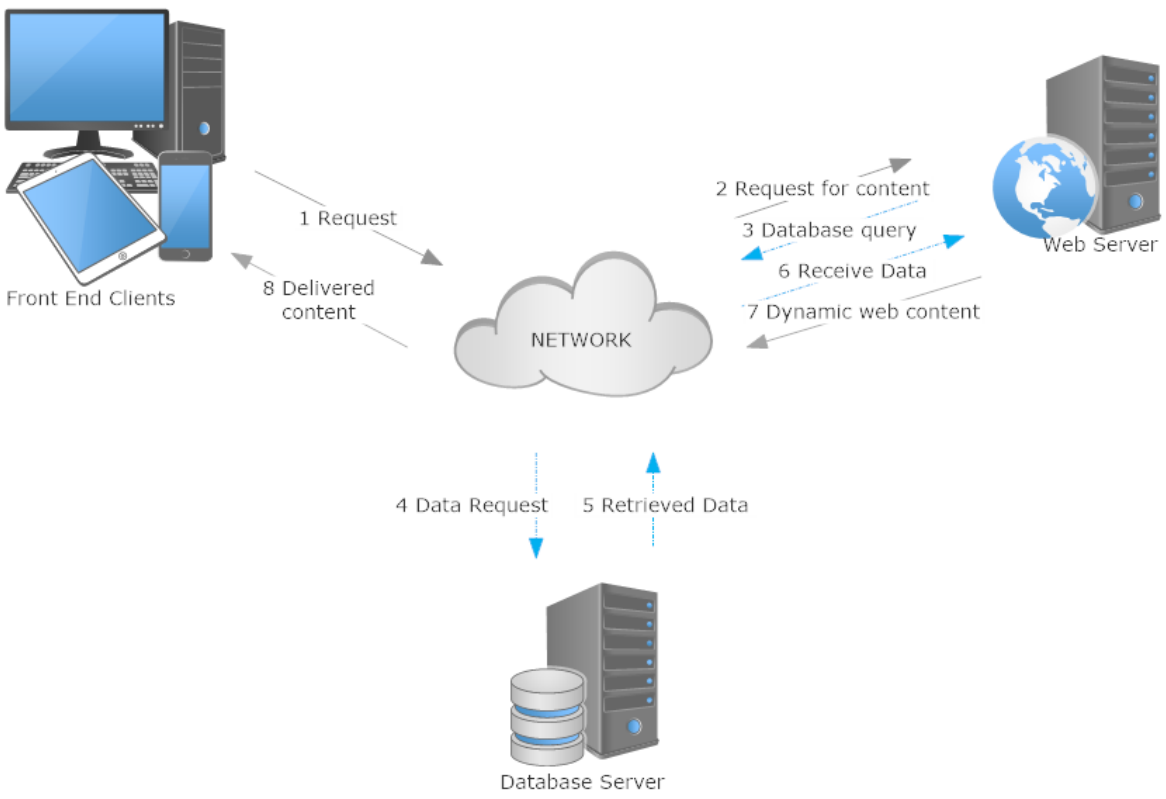

MESSAGES
_id pk <old>
EVENT fk <old>
SENDER fk <old>
RECIPIENT fk <old>
EVENT_TYPES
_id pk <old>
EVENTS
_id pk <old>
TYPE fk <old>
AWAY_TEAM fk <old>
HOME_TEAM fk <old>
LEAGUE fk <old>
OFFICIALS fk <old>
MIN_AGE_GROUP fk <old>
MAX_AGE_GROUP fk <old>
ORGANIZATION fk <old>
FIELD fk <old>
CREATOR fk <old>
AGE_GROUPS
_id pk <old>
BIRTH_YEAR <str>
SOCCER_YEAR <str>
GENDER <str>
RULES
AGE_GROUP fk <old>
LEAGUE fk <old>
LEAGUE_TYPES
_id pk <old>
NAME <str>
USER_SETTINGS
_id pk <old>
USER fk <old>
USERS
_id <old>
TEAM_PLAYERS
TEAM fk <old>
PLAYER fk <old>
TEAMS
_id pk <old>
NAME <str>
CLUB fk <old>
AGE_GROUP fk <old>
LEAGUE_TEAMS
TEAM fk <old>
LEAGUE fk <old>
LEAGUES
_id pk <old>
NAME <str>
MIN_AGE_GROUP fk <old>
MAX_AGE_GROUP fk <old>
TYPE fk <old>
RESCHEDULE_RULES fk <old>
CLUB_MEMBERS
CLUB pk, fk <old>
USER fk <old>
CLUBS
_id pk <old>
ORGANIZATIONS
_id pk <old>
ADMINISTRATOR fk <old>
CLUB_AFFILIATION fk <old>
STAFF fk <old>
NOTIFICATION
_id pk <old>
EVENT fk <old>
SENDER fk <old>
RECIPIENT fk <old>
CHANGE_REQUEST
_id pk <old>
ORIGINAL_EVENT fk <old>
CHANGED_EVENT fk <old>
APPROVER fk <old>
SUBMITTER fk <old>
RESCHEDULE_RULES
_id pk <old>
TIMESPAN_DAYS <num>
CONSEQUENCE <str>
FINE_AMOUNT <num>
ROLES
_id <old>
ACCESS_REQUESTS
_id pk <old>
USER fk <old>
CLUB fk <old>
ROLE fk <old>
TEAM fk <old>
OWNER fk <old>
STATUS <str>
TEAM_COACHES
TEAM fk <old>
COACH fk <old>
FIELDS
_id pk <old>
FACILITY fk <old>
SIZE fk <old>
USER_INVITES
_id pk <old>
ROLE fk <old>
USER_ROLES
ROLE fk <old>
USER fk <old>
FACILITIES
_id pk <old>
FIELD_SIZES
_id pk <old>

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/