**COACH ASSISTANT**

*A little league web app to assist coaches   
and families during season*

Contents

[REQUIREMENTS 2](#_Toc39264259)

[TOPIC IDEAS: 2](#_Toc39264260)

[CLASSES & METHODS 2](#_Toc39264261)

[MODEL: 2](#_Toc39264262)

[METHODS: 4](#_Toc39264263)

[USER STORIES 4](#_Toc39264264)

[League 4](#_Toc39264265)

[TEAM 5](#_Toc39264266)

[TASKS LIST 6](#_Toc39264267)

# REQUIREMENTS

-Java 1.8

-Servlets

-JDBC

-SQL

-HTML/CSS

-Bootstrap

-AJAX

-JUnit

-log4j

Environment:

Tomcat

IntelliJ

-DBeaver

CONTAINERIZE – using docker

UNIT TESTING – 33% MUST BE TESTED – ALSO NEED TO DO COVERAGE REPORTS

TOPIC IDEAS:  
Little League App – an app that houses the leagues(Roster of players & coach), players(contact info, kids age/birthdate, parentsnames & address), Schedule, scores for the league & current league ladder. App will allow for both Coaches and Players to signup to the leagues, split the leagues into appropriate teams – and kick off a league schedule for playing. Leagues will be able to be renamed after creation

# CLASSES & METHODS

## MODEL:

* League
  + City (String)
  + Teams[] (Team)
  + Season(String) (format: YYYY-MM-DD)
* User
  + Username (String)
  + Password (String)
  + Email (String)
  + Useraccess (String) THIS SHOULD ONLY BE: admin, coach, player, user(defaulttype)
* SPONSOR
  + Name (String)
  + Phone (String(10))
  + Email (String)
* Team
  + Name(String)
  + Coach (Coach)
  + Players [] (player)
  + Schedule[](schedule)
  + GameScores[] (gameScore)
  + Sponsor (String)
* Schedule
  + GameDay(LocalDate)
  + teamOne(Team)
  + teamOne(Team)
  + ScoreTeamOne (int)
  + ScoreTeamTwo (int)
  + Forfiet (String)
* Person
  + Name(String)
  + Phone(Numeric(10) )
  + Phonecarrier(PhoneCarrier)
  + allowSMS (Boolean)
  + Team(team)
* Player (extends person)
  + Parent (string)
  + Age(int)
  + Position (Enum)
* Position (Enum)
  + Pitcher, Catcher, FirstBase, SecondBase,ShortStop, ThirdBase, LeftField, RightField, CenterField
* PhoneCarrier(Enum)
  + TMobile, Virgin, Cingular, Sprint, Verizon, GoogleFi
* GameScore
  + Score(int)
  + Win(Boolean)
  + DayPlayed(string)

## METHODS:

* LEAGUE
  + Register Coach
  + Register Player
  + Create Teams – this function creates an array of teams based on the registered coaches and number of players. Players are randomly split/selected – minimum of 9 per team. Teams are given a null value name and can be changed after creation.
  + Create Schedule – this function takes and pairs teams to play at one week intervals (on different days) for an 11 week season. Each week should be calculated the same night/day played.
* TEAM
  + RenameTeam
  + RecordScoreOnDay – record the score from a game on a particular day and mark as a win or loss
  + AddSponsor

# USER STORIES

## League

1. A league administrator needs to start a league season
   1. Software will check to see if enough players have registered for this season to fulfill at least 2 teams (min 18 players) & enough coaches (min 2)
   2. Software splits players into even distribution (random selection) into however many coaches are registered for the season.
   3. Teams are established (Coach + roster) and a default team name is given. An SMS message is sent to the coach and each player that a season has been setup with a date to confirm to the coach for final roster.
      1. Manual/automatic reminder via admin portal to coaches
   4. backend creates a schedule for the season
      1. 11 weeks
      2. Different playdays pitting 2 teams together so that everyone plays eachother a fair/even amount of times each week changes
2. A person in the world needs to register for a little league season
   1. User logs into the portal
   2. User navigates to the registration.html(.jsp??) and registers for the year season
   3. Upon successful registration user receives an email(SMS if available) that they are registered and awaiting team assignment
3. A Sponsor needs to register as a team sponsor in the league
   1. User logs into their portal
   2. User navigates to the Sponsor section and clicks on Registration (sponsorregistration.html) and registers
4. A User needs to modify/update their information
   1. User logs into portal and clicks UPDATE on their account tab

## TEAM

1. A Coach is notified through email/sms that their team is assigned
   1. Coach needs to log into the portal & verify/confirm each member of their team
   2. Coach can drop people that reply as non-participants as well manually add people from an available list of registered (but non-teamed) people in the dbase
      1. Dropped people are removed from registered players table
   3. Coach can at this time rename the team
2. A Coach needs to add a sponsor to their team
   1. Coach logs into the portal and goes to their team page
   2. Sponsor will show up as unassigned – with a + button to add
      1. A list of available sponsors in the database will be listed for selection.
         1. Selecting a sponsor will assign that sponsor to the team appropriately
         2. Sponsors table will be updated for the appropriate sponsor with the selected team
3. A Coach needs to setup a practice schedule for their team
   1. Coach logs into the portal and goes to their team page
   2. PracticeDay field is set to ‘unassigned’ with a + button to schedule
      1. A list of daytimes will populate from the dbase for available date/time to schedule a 1.5 hour practice.
         1. Selecting the timeslot will secure that team for the timeslot selected in the Practices table
         2. Once schedule is setup – an SMS/Email is sent to each member of the TEAM to notify
4. A Game has been played and scores neeed to be input
   1. Coach logs into portal and goes to the schedule
   2. Coach clicks scheduled game and is able to enter scores
   3. Scores are stored in SCHEDULES table for the appropriate fields
5. A Game is scheduled – but one team needs to forfeit for any reason
   1. Coaches log into each respective portal and enter scores
   2. Forfeit team enters 0 and clicks check/radio for FORFEIT
   3. Submit button writes to SCHEDULE for pertinent transaction
6. A Coach needs to modify positions on a team
   1. Coach logs into portal and goes to team setup – and selects positions
   2. Coach is able to go to each position (diagram) and drops down and selects player from list to slot to that position
      1. Updates both page and database for position assignment
      2. SMS/email given for update change to team roster (OPTIONAL)

# TASKS LIST

* Create Project Template – Web App
* Create Models
* Create Database Schema
* Create/Define Userstories
* Generate Tasks from UserStories
* GREG
  + Design newuserregistration.html
    - Form for user table data
  + Create UserRegistration servlet
    - Implement UserRegistration from newuseregistration form submission
      * Adding row into Users table
  + WEB PAGES
  + Add “new user registration” link to homepage header
* Insert header into index.html and most all other pages
  + Contact.jsp
    - Simple information page with a form at the bottom to send site administrator a message (fake email for now)
  + Springtraining.jsp
    - Simple information page with a signup form at the bottom
      * Form simply does an email to site administrator sending information (Fake email for now)
* Userportal.jsp (DAN)
  + - Redirect to 3 separate pages depending on loginobject:useraccess
      * Login notification if not currently loggedin
  + Sponsorship.jsp
    - Shows list of current sponsors
    - Register as a sponsor
* Registration.jsp (GREG)
  + Create Coaches & Players tables from schema diagram
  + Create Servlet “PersonRegistration”
    - Implement functionality for registering a person into appropriate database from selected option @ web page.
  + Create class/methods for the “PersonDAO” & “PersonDAOImpl\_PGR” & “PersonService”
    - Register as a coach/player
      * Form updates appropriate table on selection of type (coach/player).
      * MUST BE LOGGED IN
  + Season.jsp
    - Shows calendar schedule
      * Click to show schedule information
* SERVLET
  + UserLogin
  + UserRegistration
  + UserPortal
  + SeasonRegistration
  + Sponsorship
* DOCKER
* EnvironmentVariables setup into docker
* Project1\_UName - postgres
* Project1\_UPassword - Cavalier93!
* Project1\_URL - jdbc:postgresql://database-1.cdr3hmlqxdcv.us-east-2.rds.amazonaws.com:5432/project1

EXTERNAL DOCS TO PRODUCE:

* Config.cfg