

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CupPlaner.Helpers
{
    public class Validator
    {
        CupDBContainer db = new CupDBContainer();

        // Validates to check if teams in a match are available for a given time
        public bool areTeamsFree(Match m, DateTime startTime)
        {
            // Start time can not be earlier than the tournament stage start time
            if (m.TournamentStage.TimeInterval.StartTime > startTime)
            {
                return false;
            }
            foreach (Team team in m.Teams)
            {
                // Get the time interval for a team and for the parameter date
                TimeInterval timesForDate = team.TimeIntervals.First(x =>
x.StartTime.Date == startTime.Date);
                // Start time can not be earlier than teams start time and the match can
not end later than a teams end time
                if (startTime < timesForDate.StartTime ||
startTime.AddMinutes(m.Duration) > timesForDate.EndTime)
                {
                    return false;
                }
                // Check all the matches for the team
                foreach (Match match in team.Matches)
                {
                    if (match.IsScheduled && startTime.AddMinutes(m.Duration * 2) >
match.StartTime)
                    {
                        DateTime teamBreakDone =
match.StartTime.AddMinutes(match.Duration * 2);
                        if (startTime < teamBreakDone)
                        {
                            return false;
                        }
                    }
                }
            }
            return true;
        }
        //Function that checks for different parameters that needs to be fulfilled for the
cup to be able to be scheduled
        public bool IsScheduleReady(int tournamentId)
        {
            bool isValid = true;
            Tournament t = db.TournamentSet.Find(tournamentId);

            bool fieldIsValid;

```

```

        if(t.Divisions.Count < 1)
        {
            isValid = false;
            return isValid;
        }
        else
        {
            foreach (Division d in t.Divisions)
            {
                fieldIsValid = false;

                foreach(Field f in d.Tournament.Fields)
                {
                    if(f.Size == d.FieldSize)
                    {
                        fieldIsValid = true;
                    }
                }
                if (fieldIsValid == false)
                {
                    isValid = false;
                    return isValid;
                }
                if (d.Pools.Count < 1)
                {
                    isValid = false;
                    return isValid;
                }
                foreach (Pool p in d.Pools)
                {
                    if (p.Teams.Count <= 1)
                    {
                        isValid = false;
                        return isValid;
                    }
                }
            }
        }

        return isValid;
    }
}

```