|  |  |
| --- | --- |
|  | Part 1  using System; |
|  | using System.Collections.Generic; |
|  |  |
|  | namespace TimeTable.Core.Models |
|  | { |
|  | public class Entity |
|  | { |
|  | public string Id { get; set; } |
|  | } |
|  |  |
|  | public class Course : Entity |
|  | { |
|  | public string Title { get; set; } |
|  | public List<TimeSlot> Slots { get; set; } |
|  | public Teacher Teacher { get; set; } |
|  | public List<StudentCourse> Students { get; set; } |
|  |  |
|  | } |
|  |  |
|  | public class Place : Entity |
|  | { |
|  | public string Name { get; set; } |
|  | public List<TimeSlot> TimeSlots { get; set; } |
|  | } |
|  |  |
|  | public class Student : Entity |
|  | { |
|  | public string Name { get; set; } |
|  | public string Email { get; set; } |
|  | public List<StudentCourse> Courses { get; set; } |
|  | public List<TimeSlot> TimeSlots { get; set; } |
|  | } |
|  |  |
|  | public class StudentCourse |
|  | { |
|  | public string CourseId { get; set; } |
|  | public string StudentId { get; set; } |
|  |  |
|  | public Student Student { get; set; } |
|  | public Course Course { get; set; } |
|  |  |
|  | } |
|  |  |
|  | public class Teacher : Entity |
|  | { |
|  | public List<Course> Courses { get; set; } |
|  | public string Name { get; set; } |
|  | } |
|  |  |
|  | public class TimeSlot : Entity |
|  | { |
|  | public DayOfWeek Day { get; set; } |
|  | public TimeSpan Start { get; set; } |
|  | public TimeSpan End { get; set; } |
|  | public string PlaceId { get; set; } |
|  | public Place Place { get; set; } |
|  | public string CourseId { get; set; } |
|  | public Course Course { get; set; } |
|  | public List<Student> Students { get; set; } |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/788c8c3e94d017a312e3ad9942925a82/raw/64a4f31e054aaa2922af4f5b08930651575b5ade/TimeTableModels.cs)[**TimeTableModels.cs**](https://gist.github.com/TheFo2sh/788c8c3e94d017a312e3ad9942925a82#file-timetablemodels-cs)hosted with ❤ by [**GitHub**](https://github.com/)

Part 2

|  |  |
| --- | --- |
|  | using Microsoft.EntityFrameworkCore; |
|  | using TimeTable.Core.Models; |
|  |  |
|  | namespace TimeTableServer.Context |
|  | { |
|  | public class DataContext : DbContext |
|  | { |
|  | public DbSet<StudentCourse> StudentCourse { get; set; } |
|  |  |
|  | public DbSet<Student> Students { get; set; } |
|  | public DbSet<Teacher> Teachers { get; set; } |
|  | public DbSet<Course> Courses { get; set; } |
|  | public DbSet<Place> Places { get; set; } |
|  | public DbSet<TimeSlot> TimeSlots { get; set; } |
|  |  |
|  | protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) |
|  | { |
|  | optionsBuilder.UseSqlServer( |
|  | @"Server=(localdb)\ProjectsV13;Database=TimeTableDb;Integrated Security=True"); |
|  | } |
|  |  |
|  | protected override void OnModelCreating(ModelBuilder modelBuilder) |
|  | { |
|  | modelBuilder.Entity<StudentCourse>().HasKey("StudentId", "CourseId"); |
|  | modelBuilder.Entity<Student>().HasMany(student => student.Courses); |
|  | modelBuilder.Entity<Course>().HasMany(course => course.Students); |
|  | modelBuilder.Entity<Student>().HasMany<TimeSlot>(student => student.TimeSlots); |
|  | modelBuilder.Entity<TimeSlot>().HasMany<Student>(slot => slot.Students); |
|  |  |
|  | base.OnModelCreating(modelBuilder); |
|  | } |
|  |  |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/4f1bdac9a82249da25f4ec060fd4d4d0/raw/65bc51f7fab154390556ebc814efd11ee6d7a000/TimeTableServerDataContext.cs)[**TimeTableServerDataContext.cs**](https://gist.github.com/TheFo2sh/4f1bdac9a82249da25f4ec060fd4d4d0#file-timetableserverdatacontext-cs)hosted with ❤ by [**GitHub**](https://github.com/)

Part 3

|  |  |
| --- | --- |
|  | struct TimeSlotChromosome |
|  | { |
|  | public TimeSpan StartAt { get; set; } |
|  | public TimeSpan EndAt => StartAt.Add(TimeSpan.FromHours(3)); |
|  | public string CourseId { get; set; } |
|  | public string PlaceId { get; set; } |
|  | public string TeacherId { get; set; } |
|  | public List<string> Students { get; set; } |
|  | public int Day { get; set; } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/7026cd58d97edab9ee65c29ebd982171/raw/ef58ae5a69728de081a3e6b4f08f45a1963a9d6a/TimeSlotChromosome.cs)[**TimeSlotChromosome.cs**](https://gist.github.com/TheFo2sh/7026cd58d97edab9ee65c29ebd982171#file-timeslotchromosome-cs)hosted with ❤ by [**GitHub**](https://github.com/)

Part 4

|  |  |
| --- | --- |
|  | using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using Accord.Genetic; |
|  | using Microsoft.EntityFrameworkCore; |
|  |  |
|  | namespace TimeTable.Builder |
|  | { |
|  | class TimeTableChromosome:ChromosomeBase |
|  | { |
|  | private readonly DataContext \_dataContext; |
|  | static Random Random=new Random(); |
|  |  |
|  | static TimeSpan RandomStartTime() |
|  | { |
|  | return TimeSpan.FromMilliseconds(Random.Next((int) TimeSpan.FromHours(9).TotalMilliseconds, |
|  | (int) TimeSpan.FromHours(17).TotalMilliseconds)); |
|  | } |
|  |  |
|  |  |
|  | public List<TimeSlotChromosome> Value; |
|  |  |
|  | public TimeTableChromosome(DataContext dataContext) |
|  | { |
|  | \_dataContext = dataContext; |
|  | Generate(); |
|  | } |
|  | public TimeTableChromosome(List<TimeSlotChromosome> slots, DataContext dataContext) |
|  | { |
|  | \_dataContext = dataContext; |
|  | Value = slots.ToList(); |
|  | } |
|  | public override void Generate() |
|  | { |
|  | IEnumerable<TimeSlotChromosome> generateRandomSlots() |
|  | { |
|  | var courses = \_dataContext.Courses |
|  | .Include(course => course.Teacher) |
|  | .Include(course => course.Students).ToList(); |
|  |  |
|  | foreach (var course in courses) |
|  | { |
|  | yield return new TimeSlotChromosome() |
|  | { |
|  | Students = course.Students.Select(student => student.StudentId).ToList(), |
|  | CourseId = course.Id, |
|  | StartAt = RandomStartTime(), |
|  | PlaceId = \_dataContext.Places.OrderBy(place => Guid.NewGuid()).FirstOrDefault().Id, |
|  | TeacherId = course.Teacher.Id, |
|  | Day=Random.Next(1,5) |
|  | }; |
|  | } |
|  | } |
|  |  |
|  | Value= generateRandomSlots().ToList(); |
|  |  |
|  | } |
|  |  |
|  | public override IChromosome CreateNew() |
|  | { |
|  | var timeTableChromosome = new TimeTableChromosome(\_dataContext); |
|  | timeTableChromosome.Generate(); |
|  | return timeTableChromosome; |
|  | } |
|  |  |
|  | public override IChromosome Clone() |
|  | { |
|  | return new TimeTableChromosome(Value,\_dataContext); |
|  | } |
|  |  |
|  | public override void Mutate() |
|  | { |
|  | var index = Random.Next(0, Value.Count - 1); |
|  | var timeSlotChromosome = Value.ElementAt(index); |
|  | timeSlotChromosome.StartAt = RandomStartTime(); |
|  | timeSlotChromosome.Day = Random.Next(1, 5); |
|  | Value[index] = timeSlotChromosome; |
|  |  |
|  | } |
|  |  |
|  | public override void Crossover(IChromosome pair) |
|  | { |
|  | var randomVal = Random.Next(0, Value.Count - 2); |
|  | var otherChromsome = pair as TimeTableChromosome; |
|  | for (int index = randomVal; index < otherChromsome.Value.Count; index++) |
|  | { |
|  | Value[index] = otherChromsome.Value[index]; |
|  | } |
|  |  |
|  | } |
|  |  |
|  | public class FitnessFunction : IFitnessFunction |
|  | { |
|  | public double Evaluate(IChromosome chromosome) |
|  | { |
|  | double score = 1; |
|  | var values = (chromosome as TimeTableChromosome).Value; |
|  | var GetoverLaps = new Func<TimeSlotChromosome, List<TimeSlotChromosome>>(current => values |
|  | .Except(new []{current}) |
|  | .Where(slot=>slot.Day == current.Day) |
|  | .Where(slot =>slot.StartAt == current.StartAt |
|  | || slot.StartAt <= current.EndAt&&slot.StartAt >= current.StartAt |
|  | || slot.EndAt >= current.StartAt&&slot.EndAt <= current.EndAt) |
|  | .ToList()); |
|  |  |
|  |  |
|  |  |
|  | foreach (var value in values) |
|  | { |
|  | var overLaps= GetoverLaps(value); |
|  | score -= overLaps.GroupBy(slot => slot.TeacherId).Sum(x=>x.Count()-1); |
|  | score -= overLaps.GroupBy(slot => slot.PlaceId).Sum(x => x.Count()-1); |
|  | score -= overLaps.GroupBy(slot => slot.CourseId).Sum(x => x.Count()-1); |
|  | score -= overLaps.Sum(item => item.Students.Intersect(value.Students).Count()); |
|  | } |
|  |  |
|  | score -= values.GroupBy(v => v.Day).Count() \* 0.5; |
|  | return Math.Pow(Math.Abs(score),-1); |
|  | } |
|  | } |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/c9338717a83fc613b4aa99c583828c47/raw/59e81cb0eb1b0bcbbe58c70506c72d14d6a39c41/TimeTableChromosome.cs)[**TimeTableChromosome.cs**](https://gist.github.com/TheFo2sh/c9338717a83fc613b4aa99c583828c47#file-timetablechromosome-cs)hosted with ❤ by [**GitHub**](https://github.com/)

Part 5

|  |  |
| --- | --- |
|  | class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  |  |
|  | using (var dataContext = new DataContext()) |
|  | { |
|  | Population population = new Population(1000, new TimeTableChromosome(dataContext), |
|  | new TimeTableChromosome.FitnessFunction(), new EliteSelection()); |
|  |  |
|  | int i = 0; |
|  | while (true) |
|  | { |
|  | population.RunEpoch(); |
|  | i++; |
|  | if (population.FitnessMax >= 0.99 || i >= 1000) |
|  | { |
|  | break; |
|  | } |
|  | } |
|  |  |
|  | var timetable = (population.BestChromosome as TimeTableChromosome).Value.Select(chromosome => |
|  | new TimeSlot() |
|  | { |
|  | CourseId = chromosome.CourseId, |
|  | PlaceId = chromosome.PlaceId, |
|  | Day = (DayOfWeek) chromosome.Day, |
|  | Start = chromosome.StartAt, End = chromosome.EndAt, |
|  | Id = Guid.NewGuid().ToString() |
|  | } |
|  | ).ToList(); |
|  | dataContext.TimeSlots.AddRange(timetable); |
|  | dataContext.SaveChanges(); |
|  | } |
|  |  |
|  |  |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/8d8ae6410761e80583f365e6012abd46/raw/0474ecf1648989179105e29e1384835710551e4f/TimeTableBuilder.cs)[**TimeTableBuilder.cs**](https://gist.github.com/TheFo2sh/8d8ae6410761e80583f365e6012abd46#file-timetablebuilder-cs)hosted with ❤ by [**GitHub**](https://github.com/)

Part 6

|  |  |
| --- | --- |
|  | using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using System.Threading.Tasks; |
|  | using Microsoft.AspNetCore.Mvc; |
|  | using Microsoft.EntityFrameworkCore; |
|  | using TimeTable.Core.Models; |
|  | using TimeTableServer.Context; |
|  |  |
|  | // For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860 |
|  |  |
|  | namespace TimeTableServer.Controllers |
|  | { |
|  | [Route("api/[controller]")] |
|  | [ApiController] |
|  | public class TimeTableController : ControllerBase |
|  | { |
|  | public TimeTableController() |
|  | { |
|  |  |
|  | } |
|  |  |
|  | // GET: api/<TimeTableController> |
|  | [HttpGet] |
|  | public IEnumerable<TimeSlotDTO> Get() |
|  | { |
|  | using (var context=new DataContext()) |
|  | { |
|  | var timeslots = context.TimeSlots |
|  | .Include(slot => slot.Course) |
|  | .ThenInclude(course=>course.Teacher) |
|  | .Include(slot => slot.Course) |
|  | .ThenInclude(course => course.Students) |
|  |  |
|  | .Include(slot => slot.Place).ToList(); |
|  | return timeslots.Select(slot => new TimeSlotDTO() |
|  | { |
|  | Course = slot.Course.Title, |
|  | Place = slot.Place.Name, |
|  | Start = slot.Start.TotalMilliseconds, |
|  | End = slot.End.TotalMilliseconds, |
|  | Students=slot.Course.Students.Count, |
|  | Day = slot.Day |
|  | }).ToList(); |
|  |  |
|  | } |
|  | } |
|  |  |
|  |  |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/bcfc070481f62110afc06a763e37f7b0/raw/0bd2d1225422906e4527317c7ca35e112059c7c1/TimeTableController.cs)[**TimeTableController.cs**](https://gist.github.com/TheFo2sh/bcfc070481f62110afc06a763e37f7b0#file-timetablecontroller-cs)hosted with ❤ by [**GitHub**](https://github.com/)

|  |  |
| --- | --- |
|  | Part 7  namespace TimeTable.App.Views |
|  | { |
|  | public class TimeSlotDTO |
|  | { |
|  | public int Day { get; set; } |
|  | public double Start { get; set; } |
|  | public double End { get; set; } |
|  | public string Place { get; set; } |
|  | public string Course { get; set; } |
|  | public int Students { get; set; } |
|  | } |
|  |  |
|  | // Learn more about making custom code visible in the Xamarin.Forms previewer |
|  | // by visiting https://aka.ms/xamarinforms-previewer |
|  | [DesignTimeVisible(false)] |
|  | public partial class MainPage : ContentPage |
|  | { |
|  | public MainPage() |
|  | { |
|  | InitializeComponent(); |
|  | Device.InvokeOnMainThreadAsync(Init); |
|  |  |
|  | } |
|  |  |
|  | public async Task Init() |
|  | { |
|  | var currentWeek = DateTime.Today.AddDays(DateTime.Today.Day % 7); |
|  | using (var httpClient = new HttpClient()) |
|  | { |
|  |  |
|  | var serverResponse = await httpClient.GetStringAsync("http://d73f96fd2125.ngrok.io/api/timetable"); |
|  | var data = JsonConvert.DeserializeObject<List<TimeSlotDTO>>(serverResponse); |
|  | schedule.DataSource = data.Select(item => new ScheduleAppointment() |
|  | { |
|  | Subject = item.Course, |
|  | StartTime = currentWeek.AddDays(item.Day).AddMilliseconds(item.Start), |
|  | EndTime = currentWeek.AddDays(item.Day).AddMilliseconds(item.End), |
|  | Location = item.Place |
|  |  |
|  | }).ToList(); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/TheFo2sh/c91279e82c011f7825d3973f5ba06bc0/raw/71ca122a51bafe01af9f4cddd3373a0443c791a6/TimeTableMainPage.cs)[**TimeTableMainPage.cs**](https://gist.github.com/TheFo2sh/c91279e82c011f7825d3973f5ba06bc0#file-timetablemainpage-cs)hosted with ❤ by [**GitHub**](https://github.com/)