Листинг А.1 - Метод «быстрого» поиска докторов

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Разраб.

Лапко М. Л.

Провер.

Дунина Е.Б.

Реценз.

Н. Контр.

Самусев А.М.

Утверд.

Казаков В.Е.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

public async Task<IEnumerable<Doctor>> GetFastSearchedDoctorsAsync(string searchTerm)

{

var values = searchTerm.Trim().Split(" ");

var doctorComparer = new DoctorComparer();

var doctors = new HashSet<Doctor>(doctorComparer);

foreach(var value in values)

{

IEnumerable<Doctor> temp = await FindByCondition(doc => doc.User.FirstName.ToLower().Contains(value.ToLower())

|| doc.User.LastName.ToLower().Contains(value.ToLower())

|| doc.User.Country.ToLower().Contains(value.ToLower())

|| doc.User.Country.ToLower().Contains(value.ToLower())

|| doc.Specialization.Name.ToLower()

.Contains(value.ToLower())).ToListAsync();

foreach (var item in temp) doctors.Add(item);

}

return doctors;

}

Листинг А.2 - Метод регистрации нового пользователя

public async Task<AuthenticationResultDto> RegisterAsync(UserRegistrationDto registrationDto)

{

var existingUser = await \_userManager.FindByEmailAsync(registrationDto.Email);

if (existingUser != null) return new AuthenticationResultDto

{

ErrorMessages = new List<string> { "Данная электронная почта уже зарегистрирована, попробуйте другую!" }

};

existingUser = await \_userManager. FindByNameAsync(registrationDto.UserName);

if (existingUser != null) return new AuthenticationResultDto

{

ErrorMessages = new List<string> { "Данное имя пользователя уже зарегистрировано!" }

};

var newUser = \_mapper.Map<User>(registrationDto);

if (registrationDto.Password != registrationDto.ConfirmPassword)

return new AuthenticationResultDto

{

ErrorMessages = new List<string> { "Пароль подтверждён неверно!" }

};

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

var createdUser = await \_userManager.CreateAsync(newUser, registrationDto.Password);

if (!createdUser.Succeeded)

return new AuthenticationResultDto

{

ErrorMessages = createdUser.Errors.Select(err => err.Description)

};

if (!string.IsNullOrEmpty(newUser.Type))

{

var result = await \_userManager.AddToRoleAsync(newUser, newUser.Type);

if(!result.Succeeded)

return new AuthenticationResultDto

{

ErrorMessages = result.Errors.Select(err => err.Description)

};

}

var token = await \_userManager.GenerateEmailConfirmationTokenAsync(newUser);

var confirmationLink = $"http://localhost:3000/emailConfirmed?

token={token}&email={newUser.Email}";

var message = new MessageOptions(new string[] { newUser.Email }, "Письмо подтверждения", confirmationLink);

await \_emailService.SendConfirmEmail(message);

return await GenerateAuthResultForUser(newUser);

}

Листинг А.3 - Метод вычисления списка доступного времени для записи к доктору на определённую дату

public async Task<IEnumerable<AvailableAppointmentTimeDto>> Handle(GetAvailableAppointmentTimeForDateCommand command,

CancellationToken cancellationToken)

{

var availableTime = new List<AvailableAppointmentTimeDto>();

var doctor = await \_repository.Doctor.GetDoctorByDoctorIdAsync(command.doctorId);

if (doctor == null) return availableTime;

var doctorAppointments = await \_repository.Appointment.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

GetAllAppointmentsForDoctorAsync(command.doctorId);

var appointmentsTime = doctorAppointments.Where(app => app.StartDate.Date.Equals(command.date.Date)).

Select(app => app.StartTime);

var doctorScedule = await \_repository.DoctorsScedule.

GetDoctorsScedule(command.doctorId);

var doctorWorkingDays = string.Join(" ", doctorScedule.

Select(sced => sced.dayOfWeek.ToString()));

if (!doctorWorkingDays.Contains(command.date.DayOfWeek.ToString()) || command.date.Date < DateTime.UtcNow.Date)

return availableTime;

var doctorDayOffs = await \_repository.DoctorDayOff.

GetDoctorsDayOffsByDoctorId(command.doctorId);

var unavailableDays = doctorDayOffs.Where(day => day.Status != DoctorStatusEnum.Available).Select(day => day.Date);

foreach (var date in unavailableDays)

if (command.date.Date.Equals(date.Date)) return availableTime;

var time = doctor.WorkDayStart;

while(time < doctor.WorkDayEnd)

{

if (DateTime.UtcNow.Date == command.date.Date && DateTime.Now.TimeOfDay > time)

{

time = time.Add(new TimeSpan(0, doctor.AverageAppointmentTime, 0));

continue;

}

bool available = true;

foreach (var appTime in appointmentsTime)

if (time >= appTime && time <= appTime.Add(new TimeSpan(0, doctor.AverageAppointmentTime, 0)))

available = false;

if (available && time > doctor.DinnerStart.Subtract(new TimeSpan(0, doctor.AverageAppointmentTime, 0)) && time < doctor.DinnerEnd) available = false;

if (available) availableTime.Add(new AvailableAppointmentTimeDto

{

doctorId = command.doctorId,

Time = $"{time.Hours}:{time.Minutes}"

});

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

time = time.Add(new TimeSpan(0, doctor.AverageAppointmentTime, 0));

}

return availableTime;

}

Листинг А.4 - Тест GetDoctorsCommand\_ShouldReturnAllDoctors

[Fact]

public async Task GetDoctorsCommand\_ShouldReturnAllDoctors()

{

var command = new GetDoctorsCommand()

{

doctorParameters = new DoctorParameters { }

};

var doctorMock = new Mock<IDoctorRepository>();

doctorMock.Setup(repo => repo.GetAllDoctorsAsync(command.doctorParameters, false)).Returns(TestDoctors());

var mockRepoManager = new Mock<IRepositoryManager>();

mockRepoManager.Setup(repo => repo.Doctor).Returns(doctorMock.Object);

var mockConfigurationRoot = new Mock<IConfigurationRoot>();

mockConfigurationRoot.SetupGet(config => config[It.IsAny<string>()]).Returns("some setting");

var mapperMock = new Mock<IMapper>();

mapperMock.Setup(mapper => mapper.Map<IEnumerable<DoctorDto>>(It.

IsAny<IEnumerable<Doctor>>())).Returns(TestDoctorsDto());

var commandHandler = new GetDoctorsCommandHandler(mockRepoManager.Object,

mapperMock.Object);

var doctorsDto = await commandHandler.Handle(command, CancellationToken.None);

Assert.NotNull(doctorsDto);

Assert.IsAssignableFrom<IEnumerable<DoctorDto>>(doctorsDto);

Assert.Equal(TestDoctorsDto().Count(), doctorsDto.Count());

}

Листинг А.5 - Тест GetDoctorCommand\_ShouldReturnNull

[Fact]

public async Task GetDoctorCommand\_ShouldReturnNull()

{

var command = new GetDoctorCommand()

{

doctorId = Guid.NewGuid()

};

var doctorMock = new Mock<IDoctorRepository>();

doctorMock.Setup(repo => repo.GetDoctorByDoctorIdAsync(command.doctorId, false)).Returns(GetTestDoctor(command.doctorId));

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

var mockRepoManager = new Mock<IRepositoryManager>();

mockRepoManager.Setup(repo => repo.Doctor).Returns(doctorMock.Object);

var mockConfigurationRoot = new Mock<IConfigurationRoot>();

mockConfigurationRoot.SetupGet(config => config[It.IsAny<string>()]).Returns("some setting");

var mapperMock = new Mock<IMapper>();

mapperMock.Setup(mapper => mapper.Map<DoctorDto>(It.IsAny<Doctor>())).

Returns(GetTestDoctorDto(command.doctorId));

var commandHandler = new GetDoctorCommandHandler(mockRepoManager.Object, mapperMock.Object);

var doctorDto = await commandHandler.Handle(command, CancellationToken.None);

Assert.Null(doctorDto);

}

Листинг А.6 - Тест CreateDoctorCommand\_ShouldCreateNewDoctor

[Fact]

public async Task CreateDoctorCommand\_ShouldCreateNewDoctor()

{

var command = new CreateDoctorCommand()

{

doctor = new DoctorForCreationDto { }

};

var doctorId = Guid.NewGuid();

var doctorMock = new Mock<IDoctorRepository>();

doctorMock.Setup(repo => repo.CreateDoctor(new Doctor { Id = doctorId }));

var mockRepoManager = new Mock<IRepositoryManager>();

mockRepoManager.Setup(repo => repo.Doctor).Returns(doctorMock.Object);

mockRepoManager.Setup(repo => repo.SaveAsync()).Returns(Task.CompletedTask);

var mockConfigurationRoot = new Mock<IConfigurationRoot>();

mockConfigurationRoot.SetupGet(config => config[It.IsAny<string>()]).Returns("some setting");

var mapperMock = new Mock<IMapper>();

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

mapperMock.Setup(mapper => mapper.Map<Doctor>(It.IsAny<DoctorForCreationDto>())).

Returns(new Doctor { Id = doctorId });

var commandHandler = new CreateDoctorCommandHandler(mockRepoManager.Object,

mapperMock.Object);

var id = await commandHandler.Handle(command, CancellationToken.None);

Assert.True(id == doctorId);

}