*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***к курсовому проекту по дисциплине   
"Технологии программирования"***

***на тему***

*Проектирование и разработка* *программной системы*

*«Дистанционное обучение»*

*Выполнил: студент гр. ИСТ-220*

*Егоров Е.А.*

*Принял: Доцент*

*Кафедры ИСПИ*

*Вершинин В.В.*

*Владимир, 2023*

**АННОТАЦИЯ**

В рамках курсового проекта производилась разработка программной системы для прохождений онлайн курсов в информационной системе «Дистанционное обучение». Данный проект содержит в себе три этапа проектирования, состоящих из описания предметной области, описания функционала, словаря предметной области, а также схем данных и разрабатываемого кода. Для реализации информационной системы использовался язык программирования C#.

Разработанная информационная система может быть использована для хранения данных о пользователях, менеджерах, админов, лаборантов, курсах, заявка на покупку курса. Она содержит в себе модели данных, базу данных, контроллеры, представления, благодаря которым процесс работы с программной системой автоматизирован.

Курсовой проект представлен на 27 страницах, содержит в себе 14 рисунков, 4 приложений, и 3 использованных источника.

**ANNOTATION**

As part of the course project, a software system was developed for online courses in the "Distance Learning" information system. This project contains three stages of design, consisting of a description of the subject area, a description of the functional, a dictionary of the subject area, as well as data schemas and the code being developed. The C# programming language was used to implement the information system.

The developed information system can be used to store data about users, managers, admins, laboratory assistants, courses, an application for the purchase of a course. It contains data models, a database, controllers, views, thanks to which the process of working with the software system is automated.

The course project is presented on 27 pages, contains 14 drawings, 4 appendices, and 3 used sources.

**СОДЕРЖАНИЕ**

1. ВВЕДЕНИЕ 5

2. Описание предметной области 6

2 Проектирование системы 8

2.1 Минимальный набор функций 8

2.2 Функциональные требования 8

2.3 Диаграмма прецедентов 8

2.4 Диаграмма классов 10

**2.5 Диаграммы некоторых процессов** 11

**2.6 Диаграмма состояний** 13

**2.7 Макеты типовых компонентов системы** 13

Рисунок 5. Приветственный интерфейс 13

4 РЕАЛИЗАЦИЯ СИСТЕМЫ 17

4.1 Технологический стек 17

4.2 Реализация модели на сервере 17

4.3 Реализация аутентификации 17

4.4 Модели системы 17

4.5 Реализация контроллеров 20

4.6 Реализация представлений 20

5 ЗАКЛЮЧЕНИЕ 20

6 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21

ПРИЛОЖЕНИЕ А 22

ПРИЛОЖЕНИЕ Б 23

ПРИЛОЖЕНИЕ В 25

ПРИЛОЖЕНИЕ Г 27

1. ВВЕДЕНИЕ

В ходе выполнения курсового проекта требовалось разработать программную систему для прохождений онлайн курсов в информационной системе «Дистанционное обучение» для автоматизации учета и хранения всех курсов, а также для выдачи их студентам курса.

Цель работы: разработать программную систему учета пациентов в информационной системе «Дистанционное обучение», облегчить студентам получать знания и продвигать данное направление

Для достижения поставленной цели необходимо решить следующие задачи:

* хранение данных о пользователях, менеджерах, админах, о курсах, о стоимости курса;
* хранение данных о заявках на покупку пользователем курса;
* хранение данных о содержание каждого курса;
* предоставление пользователю возможности просматривать данные о курсе, все доступные модули;
* предоставление менеджеру возможности добавлять и изменять курсы, а также предоставить просмотр заработка с выбранного курса;
* предоставление админу возможности добавлять, изменять менеджеров, а также удалять пользователей;

2. Описание предметной области

В данной работе в качестве предметной области рассматривается программная система для прохождений онлайн курсов в информационной системе «Дистанционное обучение».

Информационная система «Дистанционное обучение» позволяет:

1. пользователю просматривать данные о курсе, его модулях, просмотр стоимости курса;
2. любому человеку зарегистрироваться как пользователь системы;
3. зарегистрированным менеджерам присвоить роль;
4. пользователю записываться и оплачивать выбранный им курс для прохождения;
5. менеджеру предоставить следующий функционал, добавление курса (название, описание и стоимость), добавление модулей (название, описание, вставка картинки, вставка видеоизображения), редактирование курса, редактирование модулей, просмотр окупаемости курса;
6. админу предоставить следующий функционал, добавление, изменение, удаление менеджеров, удаление пользователей;

**2.1 Пользователи разрабатываемой подсистемы**

*Пользователь (User) –* каждый зарегистрированный человек в системе, пользователь может просмотреть все курсы, выбрать курс, пополнить средства на своем аккаунте, купить выбранный курс, проходить купленный курс.

*Менеджер (Manager) –* зарегистрированные пользователи, которым присвоим роль менеджера администратор, менеджеры могут добавлять, изменять и удалять всю информацию связанных с курсами, а также просматривать заработок курса

*Администратор (Admin)* – пользователь системы, имеющий высшую роль, админ может удалять пользователей за нарушение правил пользование системы.

*Курсы (Courses)* – Объект в системе хранящий о себе информацию по выбранной менеджером тематике, хранит в себе такие поля как: модули, стоимость

**2.2 Словарь предметной области;**

*Курсы –* Это любые обучающие программы, которые позволяют получить профессию и начать работать. Бывают онлайн и офлайн, для новичков и продолжающих, охватывают всю профессию или её часть.

*Модуль –* программы курса по конкретной дисциплине, комплекс предметов или программа учебного курса.

2 Проектирование системы

В ходе анализа предметной области была выбрана монолитная архитектура.

Исходный код представлен в репозитории Github: https://github.com/EpicSWAGCoding/LimaMajor

2.1 Минимальный набор функций

* поддержка (авторизация) различных типов пользователей (администратор, менеджер, пользователь);
* поддержка удаление пользователей (адм.);
* поддержка добавление, изменение и удаление менеджеров (адм.);
* добавление курса (менеджер);
* покупка курса (пользователь);

2.2 Функциональные требования

* зарегистрироваться;
* авторизоваться;
* выйти из системы;
* добавление курса;
* добавление модулей для курса;
* начисление денег;
* покупка курса;
* отказ покупки курса из-за недостаточных средств на балансе;
* просмотр купленного курса пользователем;

2.3 Диаграмма прецедентов

В ходе анализа и проектирования функционала системы была создана диаграмма прецедентов. В качестве инструмента создания диаграммы была использована программа Microsoft Visio.

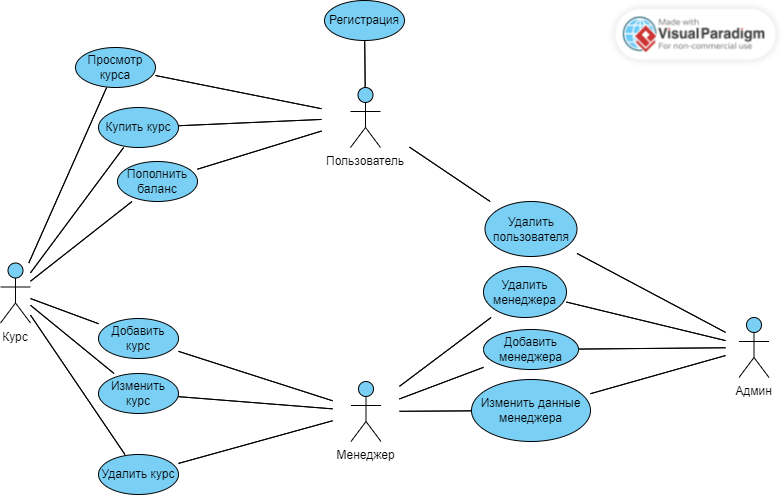


Рисунок 1 – Диаграмма прецедентов

**2.3.1 Описание прецедентов**

Расширенное описание прецедента «Добавить курс»

Название: «Добавить курс»

Предусловие: менеджер имеет доступ к добавлению, изменению и удалению курсов.

Действующее лица: менеджер, пользователь.

Основной поток: менеджер заполняет специальные поля для добавления курса, в ином случае либо изменяет уже существующие значения или вовсе удаляет их

Альтернативный поток: пользователь может просмотреть курс, купить его

Постусловие: -

Расширенное описание прецедента «Добавить менеджера»

Название: «Добавить менеджера»

Предусловие: админ может назначить выбранного пользователя менеджером, тем самым дает ему право вносить правки по курсам

Действующее лицо: админ, менеджер

Основной поток: При помощи роли админ, являющая самой высшей по иерархии, админ может добавить менеджера, изменить его данные или удалить его

Альтернативный поток: -

Постусловие: -

2.4 Диаграмма классов

Диаграмма классов представлена в приложении А. Сущности «User», «Manager», «Admin — это пользователи, которые внесли личные данные в систему и им присвоена соответствующая роль. В системе так же представлена сущность «Course» это таблица хранящая в себе информацию о курсах и уроках.

**2.5 Диаграммы некоторых процессов**

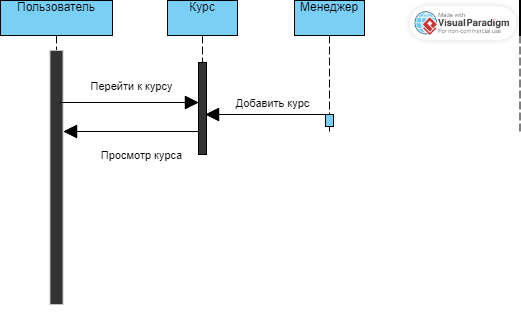


Рисунок 2 – цикл пользователя

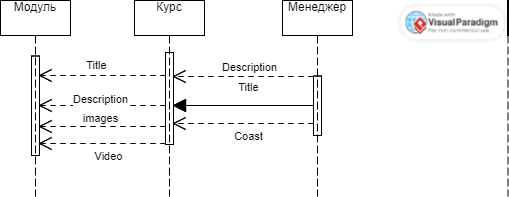


Рисунок 3 – создания курса и модуля

**2.6 Диаграмма состояний**

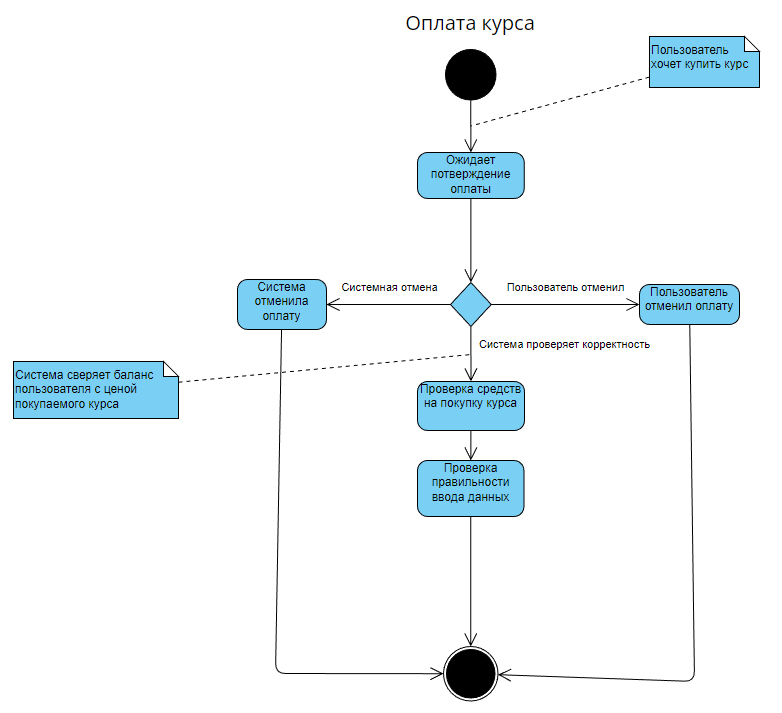


Рисунок 4 – диаграмма состояний для сущности «Заявка»

**2.7 Макеты типовых компонентов системы**

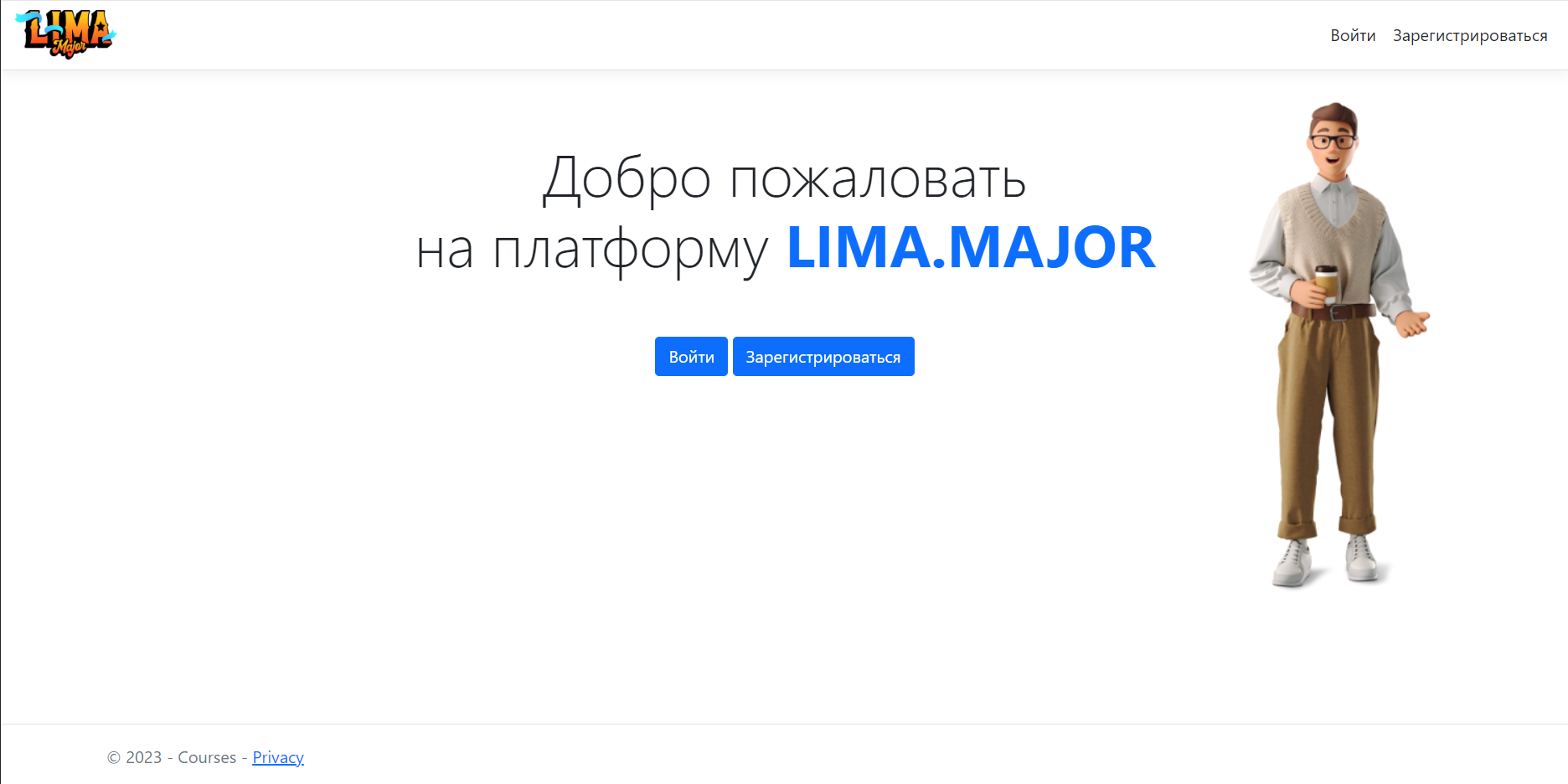


Рисунок 5. Приветственный интерфейс

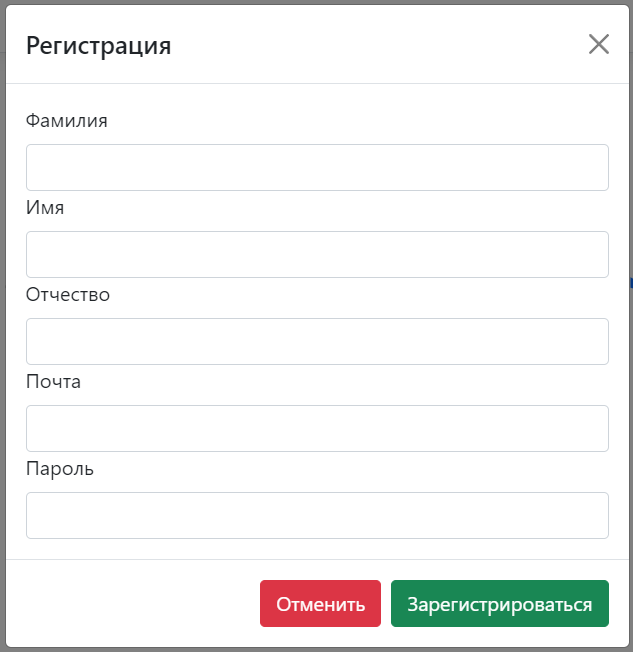


Рисунок 6 – типовая форма регистрации

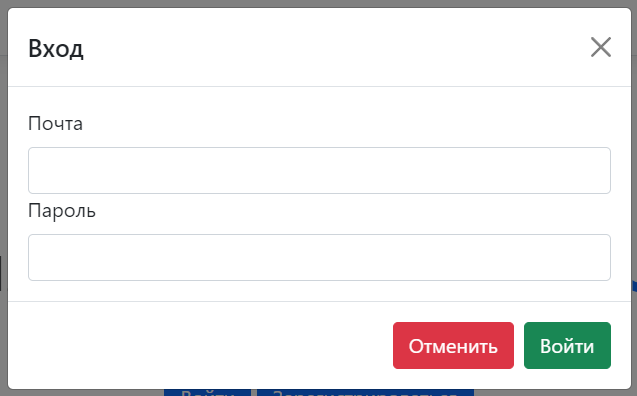


Рисунок 7 – типовая форма входа в аккаунт

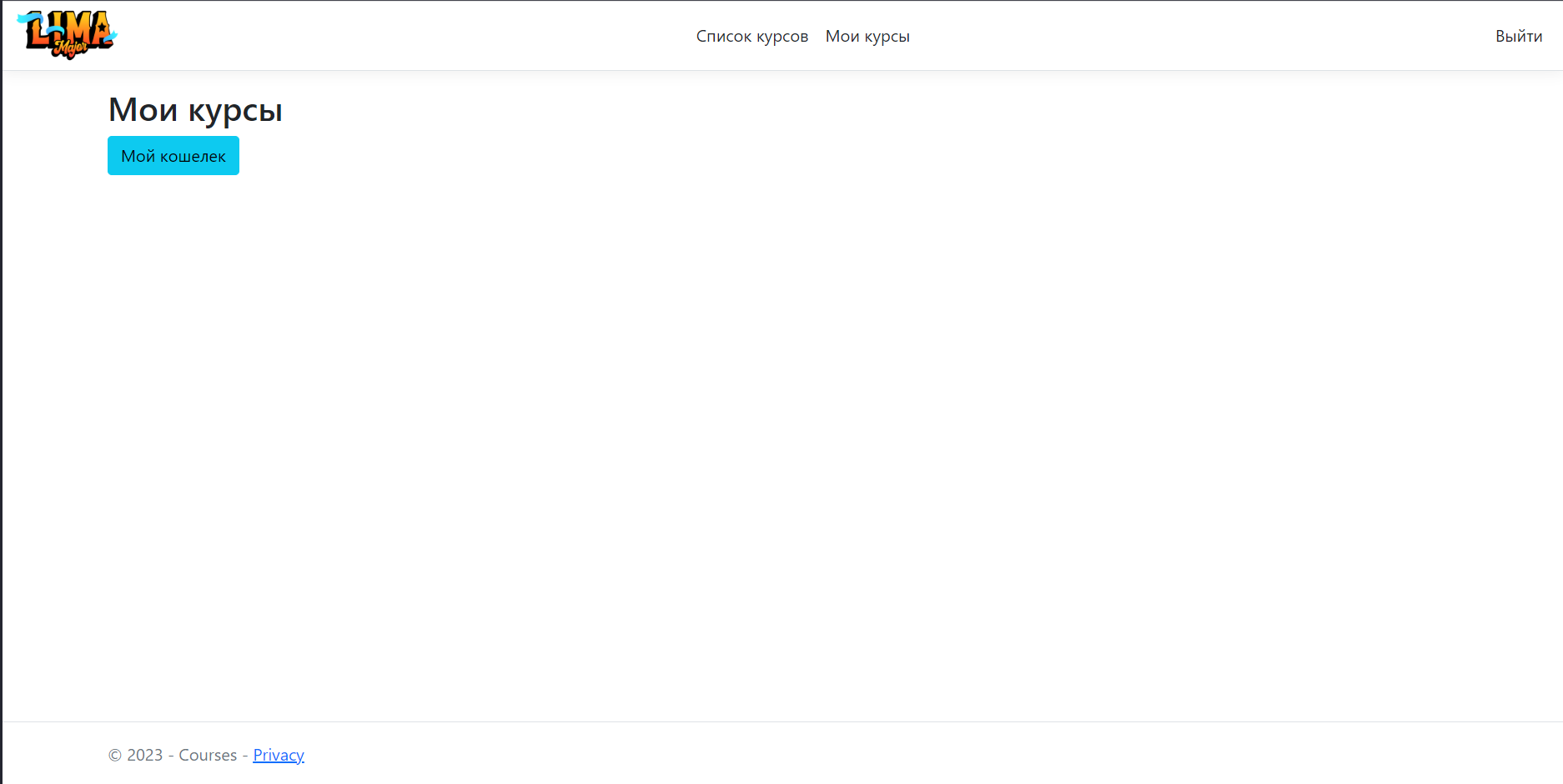


Рисунок 8 – Личный кабинет пользователя

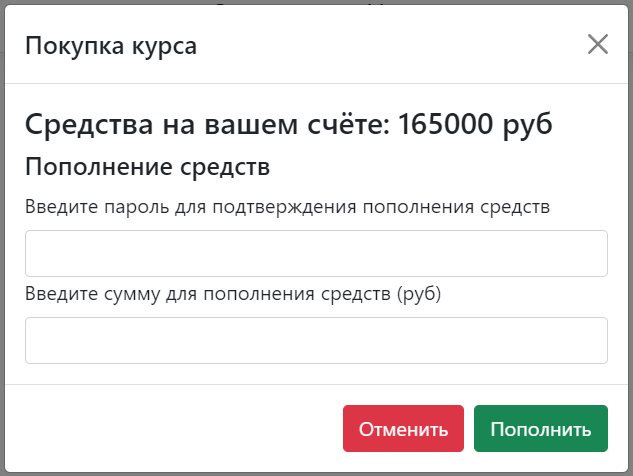


Рисунок 10 – типовая форма пополнения баланса

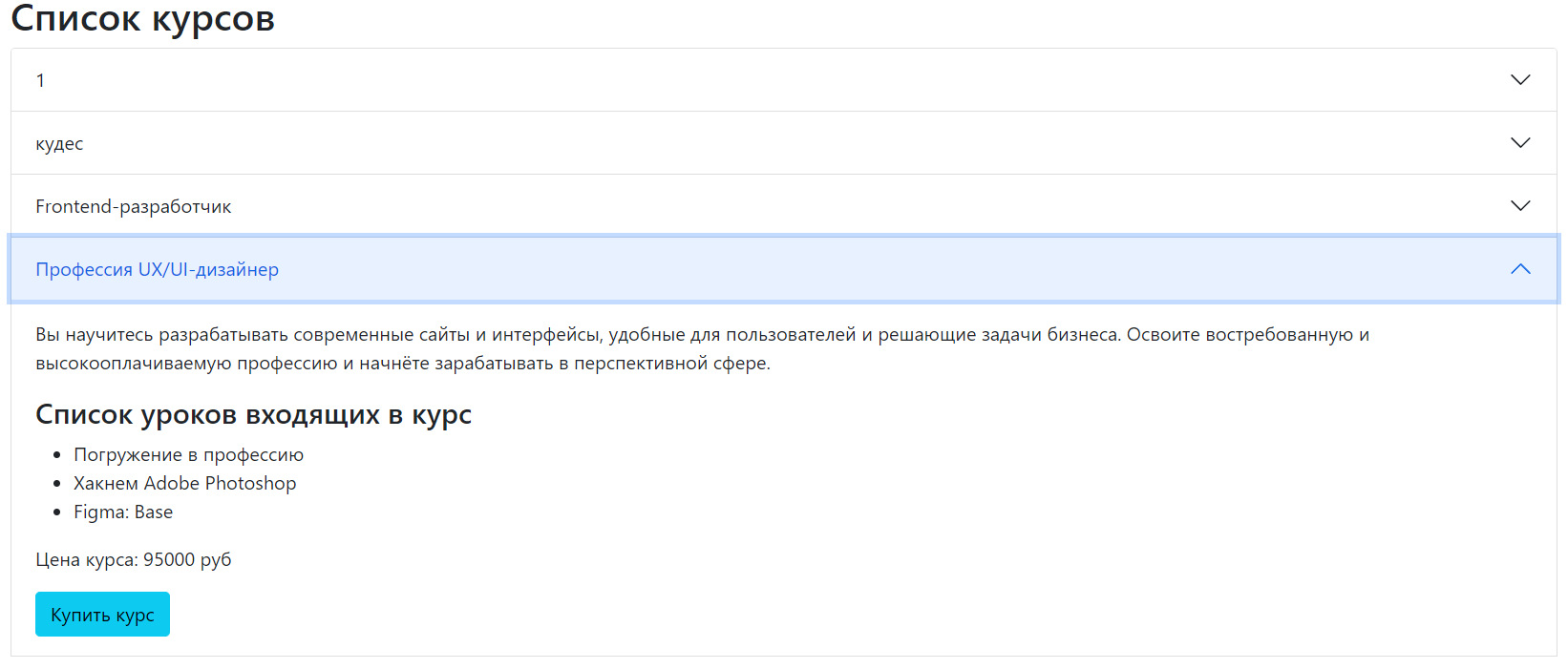


Рисунок 11 – Список доступных курсов для покупки

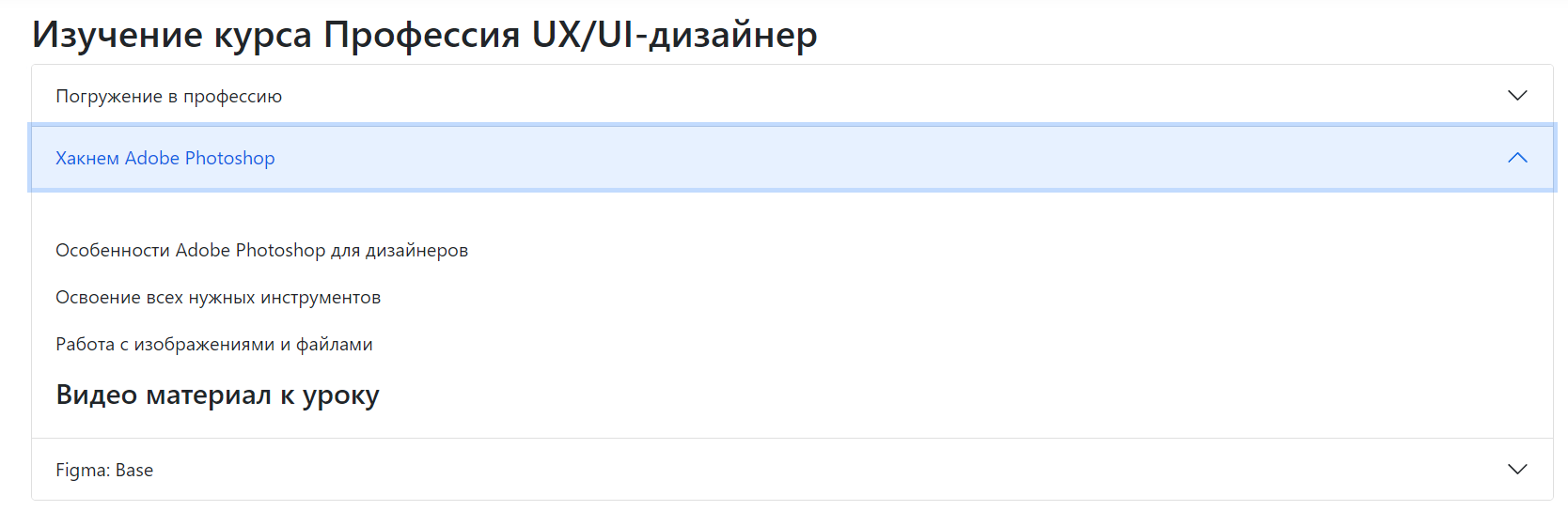


Рисунок 12 – Контентен курса

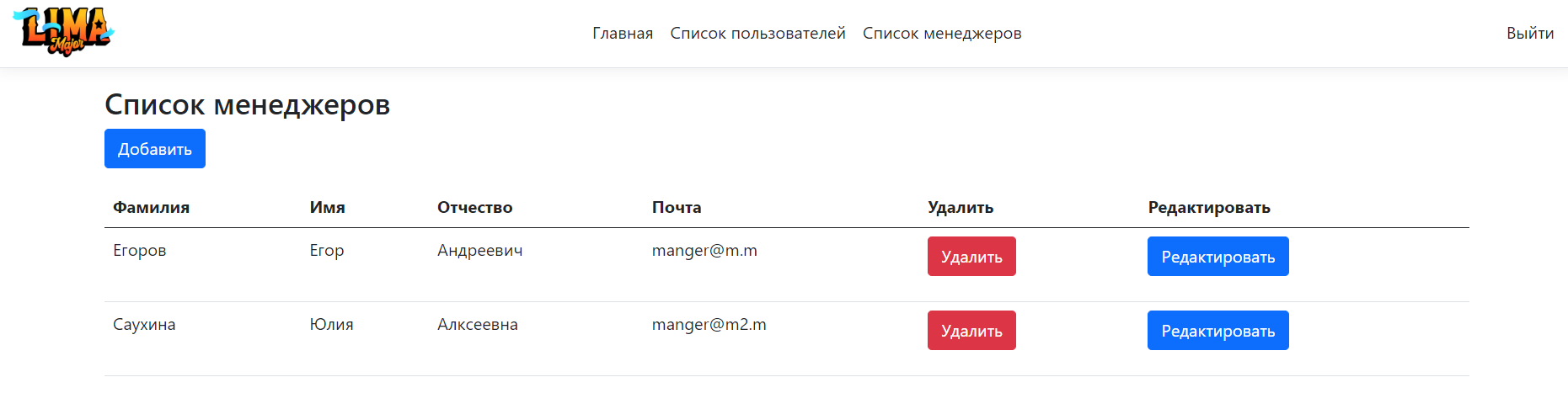


Рисунок 13 – Интерфейс админа по редактированию менеджеров

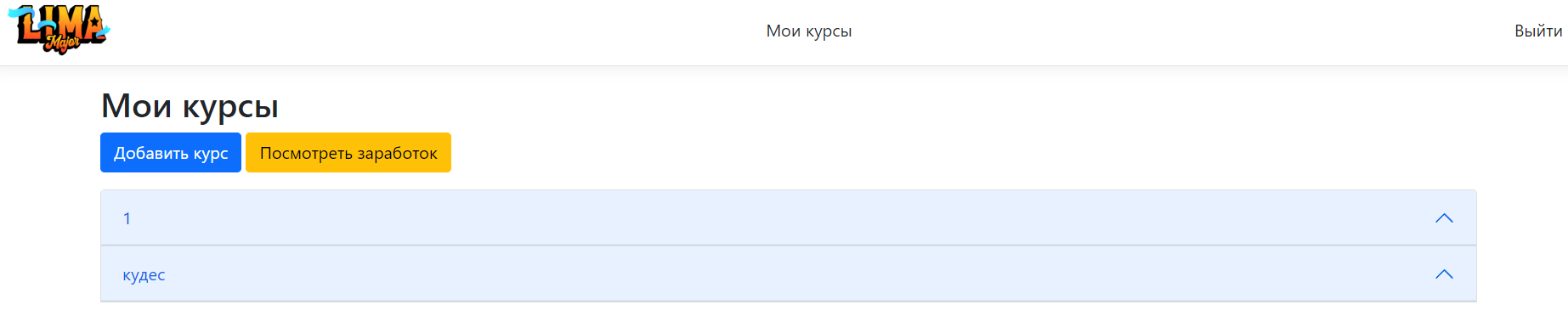


Рисунок 14 – Интерфейс менеджера по добавлению и просмотров курсов

4 РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1 Технологический стек

* реализация на платформе .net
* использование технологии asp.net core 6
* использование паттерна проектирования MVC
* использование entity framework
* использование iformfile
* использование базы данных MSSQL

4.2 Реализация модели на сервере

Сущности базы данных сгенерированы автоматически по классам, написанным на C# с использованием Entity Framework и миграций. А именно использовался подход Code First - подход, где сначала пишется код, а потом уже создается база данных.

4.3 Реализация аутентификации

Для реализации аутентификации использовался API asp.net core 6 – Identity. С помощью него реализовывался стандартный пользователь системой, а также его права доступа на базе ролей. Для поддержки разных типов пользователей помимо ролей использовались собственные таблицы со значениями, которые были связаны с таблицами, которые были сгенерированны Identity.

4.4 Модели системы

Листинг 1. Модель контекста базы данных

using Courses.Models;

using Microsoft.EntityFrameworkCore;

namespace Courses.Data

{

public class CoursesDbContext : DbContext

{

public CoursesDbContext(DbContextOptions<CoursesDbContext> options) : base(options) { }

//поля из таблиц

public DbSet<Admin> admins { get; set; }

public DbSet<Course> courses{ get; set; }

public DbSet<Lesson> lessons{ get; set; }

public DbSet<Manager> managers { get; set; }

public DbSet<User> users { get;set; }

public DbSet<Wallet> wallets { get; set; }

public DbSet<UserCourse> userCourses{ get; set; }

public DbSet<Image> images{ get; set; }

public DbSet<Video> videos { get; set; }

public DbSet<Content> contents{ get; set; }

}

}

Листинг 2. Модель «Админ»

namespace Courses.Models

{

public class Admin

{

public int Id { get; set; }

public string Email { get; set; }

public string Password { get; set; }

}

}

Листинг 3. Модель «Менеджер»

namespace Courses.Models

{

public class Manager

{

public int Id { get; set; }

public string Name { get; set; }

public string Email { get; set; }

public string Password { get; set; }

public string Surname { get; set; }

public string FatherName { get; set; }

}

}

Листинг 4. Модель «Пользователя»

namespace Courses.Models

{

public class User

{

public int Id { get; set; }

public string Name { get; set; }

public string Email { get; set; }

public string Password { get; set; }

public string Surname{ get; set; }

public string FatherName { get; set; }

}

}

Листинг 5. Модель «Курсы»

namespace Courses.Models

{

public class Course

{

public int Id { get; set; }

public string Title { get; set; }

public string Description { get; set; }

public int ManagerId { get; set; }

public Manager manager { get; set; }

public double Cost { get; set; }

}

}

Листинг 6. Модель «Модули»

namespace Courses.Models

{

public class Lesson

{

public int Id { get; set; }

public string Title { get; set; }

public int LessNum { get; set; }

public int CourseId { get; set; }

public Course course { get; set; }

}

}

Листинг 7. Модель «Кошелёк»

namespace Courses.Models

{

public class Wallet

{

public int Id { get; set; }

public string WalletNum { get; set; }

public double Money { get; set; }

public int UserId { get; set; }

public User user{ get; set; }

}

}

Листинг 8. Модель «Контента»

namespace Courses.Models

{

public class Content

{

public int Id { get; set; }

public string ContentText { get; set; }

public int LessonId { get; set; }

public Lesson lesson { get; set; }

}

}

Листинг 9. Модель «Загрузка картинки»

namespace Courses.Models

{

public class Image

{

public int Id { get; set; }

public string FileName { get; set; }

public int LessonId { get; set; }

public Lesson lesson { get; set; }

}

}

Листинг 10. Модель «Загрузка видео»

namespace Courses.Models

{

public class Video

{

public int Id { get; set; }

public string FileName { get; set; }

public int LessonId { get; set; }

public Lesson lesson { get; set; }

}

}

4.5 Реализация контроллеров

Листинг типового контроллера представлен в приложении Б

4.6 Реализация представлений

Листинг типового представления со списком представлен в приложении В.

Листинг типового представления представлен в приложении Г.

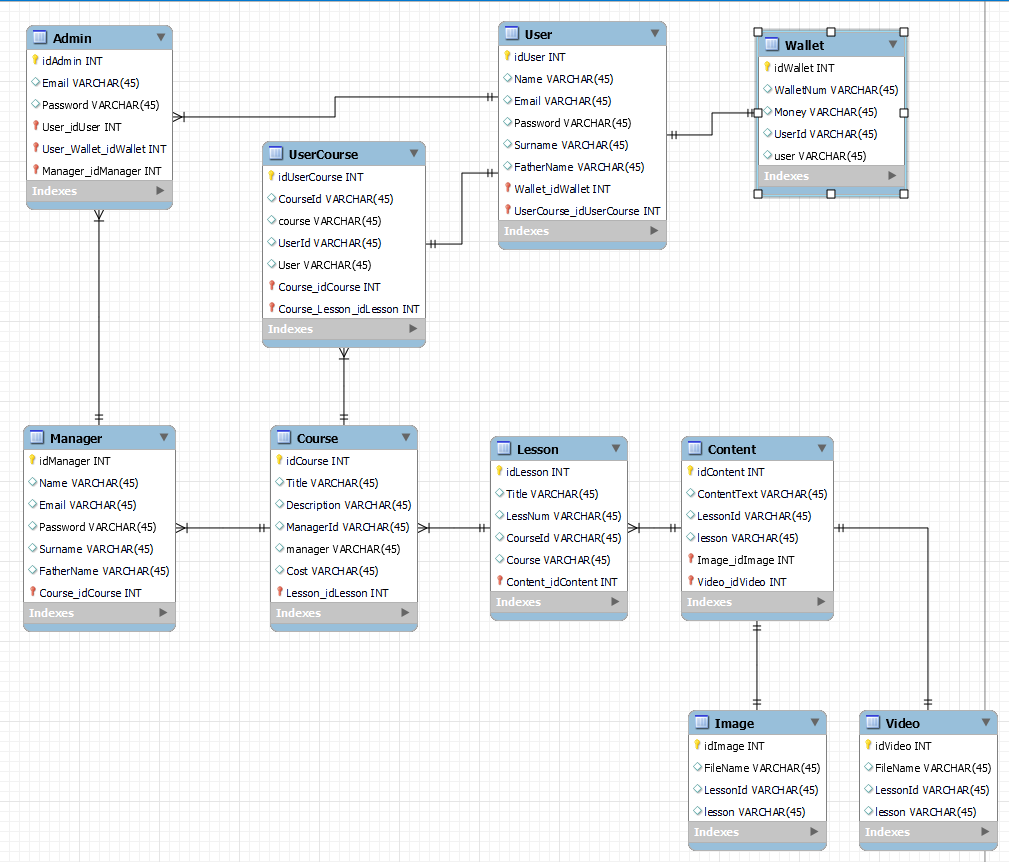
5 ЗАКЛЮЧЕНИЕ

В результате выполненной работы был произведен анализ предметной области «Дистанционное обучение», определены основные процессы, происходящие в системе, спроектирована система хранения данных, описана реализация бизнес-логики.

Были разработаны страницы регистрации, авторизации, меню, подтверждения данных, присвоения ролей пользователям, списка курсов, менеджеров, пользователей, добавления курса, пополнения баланса, покупки курса.

Текущее состояние системы позволяет ее использование в реальной сфере инфобиза.

6 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Введение в Identity в ASP.NET Core // https://learn.microsoft.com/ URL: https://learn.microsoft.com/ru-ru/aspnet/core/security/authentication/identity?view=aspnetcore-6.0&tabs=visual-studio (дата обращения: 9.12.22).
2. Новые возможности в ASP.NET Core 6.0 // https://learn.microsoft.com/ URL: https://learn.microsoft.com/ru-ru/aspnet/core/release-notes/aspnetcore-6.0?view=aspnetcore-6.0 (дата обращения: 10.12.22).
3. https://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm // tutorialspoint.com URL: Design Patterns - MVC Pattern (дата обращения: 10.12.22).
4. ПРИЛОЖЕНИЕ А

ПРИЛОЖЕНИЕ Б

Контроллер для сущности Админ

using Courses.Data;

using Courses.Models;

using Microsoft.AspNetCore.Mvc;

namespace Courses.Controllers

{

public class AdminController : Controller

{

private readonly CoursesDbContext \_context;

public AdminController(CoursesDbContext context)

{

\_context = context;

}

public IActionResult Index()

{

return View();

}

public IActionResult Users()

{

ViewData["users"]= \_context.users.ToList();

return View();

}

public IActionResult Managers()

{

ViewData["managers"] = \_context.managers.ToList();

return View();

}

public async Task<IActionResult> DeleteUser(int id)

{

\_context.users.Remove(\_context.users.Where(a => a.Id == id).FirstOrDefault());

await \_context.SaveChangesAsync();

return Redirect("~/Admin/Users");

}

public async Task<IActionResult> DeleteManager(int id)

{

\_context.managers.Remove(\_context.managers.Where(a => a.Id == id).FirstOrDefault());

await \_context.SaveChangesAsync();

return Redirect("~/Admin/Managers");

}

[HttpPost]

public async Task<IActionResult> AddManager(string Email,string Password,string Surname, string Name, string FatherName)

{

Manager manager = new Manager();

manager.Email = Email;

manager.Password = Password;

manager.Surname = Surname;

manager.Name = Name;

manager.FatherName = FatherName;

\_context.managers.Add(manager);

await \_context.SaveChangesAsync();

return Redirect("~/Admin/Managers");

}

[HttpPost]

public async Task<IActionResult> EditManager(int id,string Email, string Password, string Surname, string Name, string FatherName)

{

Manager manager = new Manager();

manager.Email = Email;

manager.Password = Password;

manager.Surname = Surname;

manager.Name = Name;

manager.FatherName = FatherName;

manager.Id = id;

\_context.managers.Update(manager);

await \_context.SaveChangesAsync();

return Redirect("~/Admin/Managers");

}

}

}

ПРИЛОЖЕНИЕ В

@{

ViewData["Title"] = "Home Page";

}

<div class="text-center justify-content-center">

<div class="container position-relative">

<img class="img-fluid position-absolute top-0 end-0" src="images/Man1.png" alt="man" />

<h1 class="display-4 p-5">Добро пожаловать <br /> на платформу <a class="text-uppercase text-decoration-none fw-bold">Lima.Major</a></h1>

</div>

<a class="btn btn-primary mb-3" data-bs-toggle="modal" data-bs-target="#Login">Войти</a>

<a class="btn btn-primary mb-3" data-bs-toggle="modal" data-bs-target="#Register">Зарегистрироваться</a>

</div>

<div class="modal fade" id="Register" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="exampleModalLabel">Регистрация</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form asp-action="Register">

<div class="modal-body">

<label class="form-label">Фамилия</label>

<input type="text" class="form-control" name="Surname" />

<label class="form-label">Имя</label>

<input type="text" class="form-control" name="Name" />

<label class="form-label">Отчество</label>

<input type="text" class="form-control" name="FatherName" />

<label class="form-label">Почта</label>

<input type="email" class="form-control" name="Email" />

<label class="form-label">Пароль</label>

<input type="password" class="form-control" name="Password" />

</div>

<div class="modal-footer">

<button type="button" class="btn btn-danger" data-bs-dismiss="modal">Отменить</button>

<button type="submit" class="btn btn-success">Зарегистрироваться</button>

</div>

</form>

</div>

</div>

</div>

<div class="modal fade" id="Login" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="exampleModalLabel">Вход</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form asp-action="Login">

<div class="modal-body">

<label class="form-label">Почта</label>

<input type="email" class="form-control" name="Email" />

<label class="form-label">Пароль</label>

<input type="password" class="form-control" name="Password" />

</div>

<div class="modal-footer">

<button type="button" class="btn btn-danger" data-bs-dismiss="modal">Отменить</button>

<button type="submit" class="btn btn-success">Войти</button>

</div>

</form>

</div>

</div>

ПРИЛОЖЕНИЕ Г

@{

Layout = "\_LayoutAdmin";

List<Manager> managers = (List<Manager>)ViewData["managers"];

}

<h3>Список менеджеров</h3>

<a class="btn btn-primary mb-3" data-bs-toggle="modal" data-bs-target="#addManager">Добавить</a>

<table class="table">

<thead>

<tr>

<th>Фамилия</th>

<th>Имя</th>

<th>Отчество</th>

<th>Почта</th>

<th>Удалить</th>

<th>Редактировать</th>

</tr>

</thead>

<tbody>

@foreach (var item in managers)

{

<tr>

<td>@item.Surname</td>

<td>@item.Name</td>

<td>@item.FatherName</td>

<td>@item.Email</td>

<td><a asp-action="DeleteManager" asp-route-id="@item.Id" class="btn btn-danger">Удалить</a></td>

<td><a class="btn btn-primary mb-3" data-bs-toggle="modal" data-bs-target="#editManager\_@item.Id">Редактировать</a></td>

</tr>

<div class="modal fade" id="editManager\_@item.Id" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="exampleModalLabel">Добавление урока</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form asp-action="EditManager">

<div class="modal-body">

<label class="form-label">Фамилия</label>

<input type="text" class="form-control" name="Surname" value="@item.Surname"/>

<label class="form-label">Имя</label>

<input type="text" class="form-control" name="Name" value="@item.Name"/>

<label class="form-label">Отчество</label>

<input type="text" class="form-control" name="FatherName" value="@item.FatherName"/>

<label class="form-label">Почта</label>

<input type="text" class="form-control" name="Email" value="@item.Email"/>

<label class="form-label">Пароль</label>

<input type="text" class="form-control" name="Password" value="@item.Password"/>

<input type="hidden" name="id" value="@item.Id">

</div>

<div class="modal-footer">

<button type="button" class="btn btn-danger" data-bs-dismiss="modal">Отменить</button>

<button type="submit" class="btn btn-success">Редактировать</button>

</div>

</form>

</div>

</div>

</div>

}

</tbody>

</table>

<div class="modal fade" id="addManager" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="exampleModalLabel">Добавление урока</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form asp-action="AddManager">

<div class="modal-body">

<label class="form-label">Фамилия</label>

<input type="text" class="form-control" name="Surname" />

<label class="form-label">Имя</label>

<input type="text" class="form-control" name="Name" />

<label class="form-label">Отчество</label>

<input type="text" class="form-control" name="FatherName" />

<label class="form-label">Почта</label>

<input type="text" class="form-control" name="Email" />

<label class="form-label">Пароль</label>

<input type="text" class="form-control" name="Password" />

</div>

<div class="modal-footer">

<button type="button" class="btn btn-danger" data-bs-dismiss="modal">Отменить</button>

<button type="submit" class="btn btn-success">Добавить</button>

</div>

</form>

</div>

</div>

</div>