

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

БИРСКИЙ ФИЛИАЛ БАШГУ
ФАКУЛЬТЕТ ФИЗИКИ И МАТЕМАТИКИ
КАФЕДРА ИНФОРМАТИКИ И ЭКОНОМИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ БАКАЛАВРИАТА

КАРАМОВ ТИМУР ИРИКОВИЧ

РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА СПЕЦИАЛИСТА
ОТДЕЛА ДОПОЛНИТЕЛЬНЫХ ПРОФЕССИОНАЛЬНЫХ ПРОГРАММ

Выполнил:
Студент 4 курса очной формы обучения
Направление подготовки
09.03.03 Прикладная информатика
Направленность (профиль)
Прикладная информатика в
информационной сфере

Руководитель
к.х.н., доцент
С.В. Пихтовников

БИРСК – 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	5
1.1. Структурно-функциональная диаграмма организации «Как есть» и ее описание	5
1.2. Обзор программных средств и технологий	7
Выводы по главе 1	11
ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА.....	12
2.1. Структурно-функциональная диаграмма «Как должно быть».....	12
2.2. Диаграммы последовательностей.....	14
2.3. Диаграммы деятельности	16
2.4. Даталогическая модель данных	16
2.5. Проектирование интерфейса.....	22
Выводы по главе 2	23
ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА	24
3.1. Реализация интерфейса.....	24
3.2. Реализация подключения интерфейса к базе данных.....	29
3.3. Реализация базы данных.....	30
3.4. Руководство пользователя	31
Выводы по главе 3	41
ЗАКЛЮЧЕНИЕ	42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	43

ВВЕДЕНИЕ

В настоящее время ведется повсеместная автоматизация различных процессов ведения данных. Для упрощения работы с данными многие организации используют механизм автоматизированного рабочего места.

Благодаря АРМ, упрощается работа внесения, хранения и обработки данных потому, что нет необходимости вручную систематизировать данные, обрабатывать их и на их основе создавать отчеты. Система сама систематизирует данные, сама формирует отчеты, благодаря чему упрощается работа

Таким образом, создание автоматизированного рабочего места для упрощения сбора и систематизации данных является важной и актуальной задачей.

Объект исследования – информационный процесс хранения данных и ведения отчетности отдела дополнительных профессиональных программ.

Предмет исследования – автоматизация информационного процесса хранения данных и ведения отчетности отдела дополнительных профессиональных программ.

Цель – разработка автоматизированного рабочего места специалиста отдела дополнительных профессиональных программ для автоматизированной обработки данных.

Задачи:

- Провести анализ предметной области.
- Спроектировать компоненты автоматизированного рабочего места.
- Разработать компоненты автоматизированного рабочего места специалиста отдела дополнительных профессиональных программ.

Структура выпускной квалификационной работы: Выпускная квалификационная работа состоит из трех глав, заключения и списка используемых источников и литературы.

В первой главе расположен анализ предметной области и обзор программных средств и технологий для разработки приложения.

Во второй главе выполнено проектирование автоматизированного рабочего места и выявлены требования к интерфейсу.

В третьей главе описана реализация подключения интерфейса к базе данных, создание базы данных и написано руководство пользователя.

Заключение содержит краткие результаты по каждой главе.

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Структурно-функциональная диаграмма организации «Как есть» и ее описание

Нотация IDEF0 предназначена для формализации и описания процессов [3, 14]. Благодаря её средствам создана контекстная диаграмма (Рисунок 1.1).

На представленной контекстной диаграмме можно выявить следующие типы стрелок:

- Вход – Ввод информации о человеке, программах. Представляет собой объекты, которые используются или преобразованы для получения результата (выхода).
- Выход – Результат работы системы.
- Управление – Нормативно-правовые акты.
- Механизм – специалист ДПП. Является ресурсом, выполняющим работу.

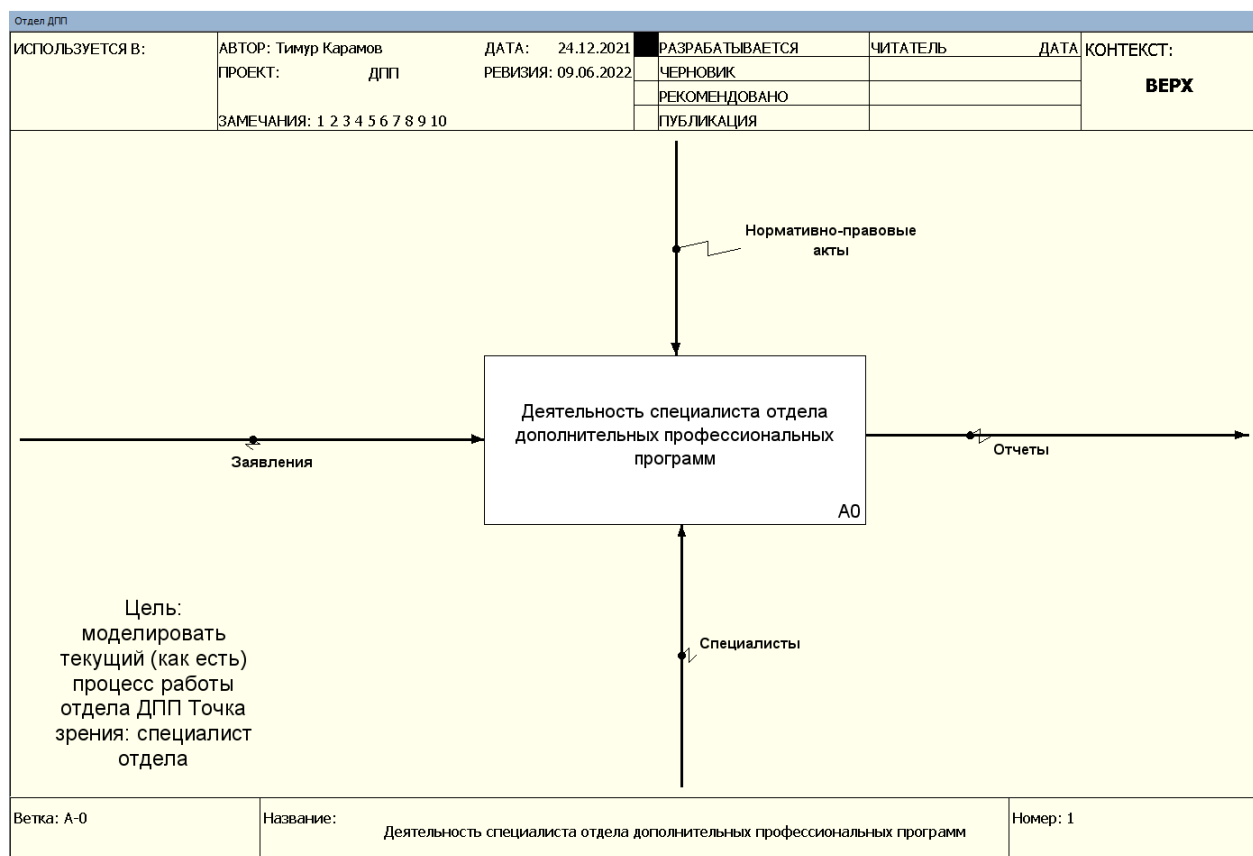


Рисунок 1.1 Контекстная диаграмма "Как есть"

Декомпозиция контекстной диаграммы включает в себя следующие функциональные блоки (Рисунок 1.2):

- Получение заявлений обучающихся.
- Обработка данных обучающихся.
- Формирование отчетов вручную.

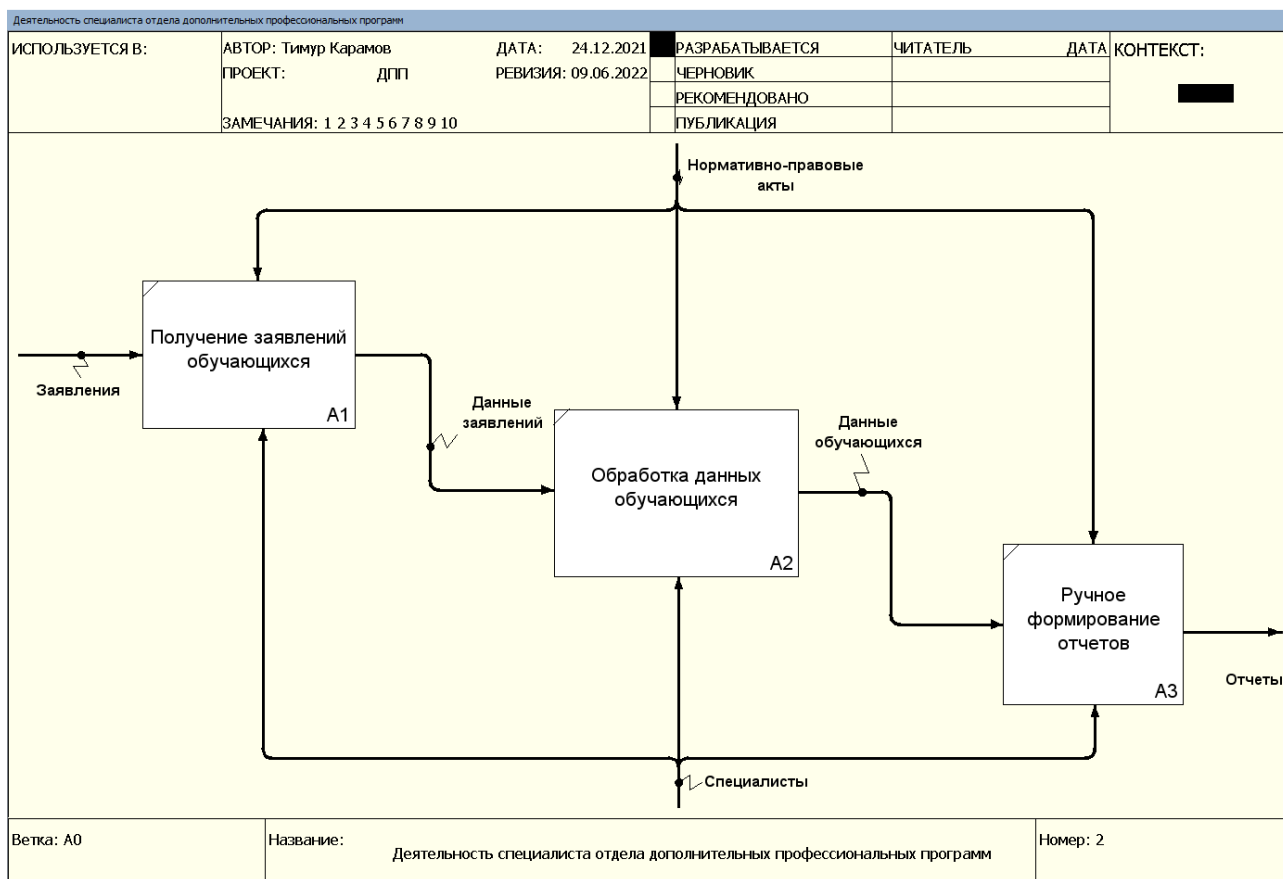


Рисунок 1.2 Результат декомпозиции контекстной диаграммы

Данная декомпозиция представляет собой иллюстрацию процесса работы системы. То есть получение данных пользователя, анализ данных пользователя, проверка данных и, по результату, запись пользователя на необходимую программу и формирование отчетов.

1.2. Обзор программных средств и технологий

В настоящее время, рынок современных средств разработки представлен широким ассортиментом. В данный момент популярны и востребованы на рынке клиент-серверные приложения, в частности веб-приложения.

Разработка приложения велась при помощи среды программирования Microsoft Visual Studio 2022.

Microsoft Visual Studio 2022 — это интегрированная среда разработки (IDE) для создания широкого спектра приложений для различных платформ.

Для создания сайта использован ASP.NET Core MVC.

ASP.NET Core – это кросс-платформенный фреймворк для создания веб-приложений на платформе .NET. Данная платформа разрабатывается компанией Майкрософт совместно с сообществом и имеет большую производительность по сравнению с ASP.NET. Имеет модульную структуру и совместима с такими операционными системами как Windows, Linux и macOS [4].

ASP.NET (Active Server Pages) расширяет .NET Core инструментами для создания серверных веб-приложений. Он использует архитектуру MVC для создания своего пользовательского интерфейса и веб-стандартов HTML, CSS и JavaScript для генерации поведения. Он включает обширную поддержку веб-API, облачных веб-приложений и масштабируемость.

ASP.NET Core включает в себя фреймворк MVC, который объединяет функциональность MVC, Web API и Web Pages. В предыдущих версии платформы данные технологии реализовались отдельно и поэтому содержали много дублирующей функциональности. Сейчас же они объединены в одну программную модель ASP.NET Core MVC [32].

ASP.NET Core предоставляет следующие преимущества:

- Единое решение для создания пользовательского веб-интерфейса и веб-API.
- Разработано для тестируемости.
- Pages упрощает написание кода для сценариев страниц и повышает его эффективность.
- Blazor позволяет использовать в браузере язык C# вместе с JavaScript. совместное использование серверной и клиентской логики приложений, написанных с помощью .NET.
- Возможность разработки и запуска в ОС Windows, macOS и Linux.
- Открытый исходный код и ориентация на сообщество.

- Интеграция современных клиентских платформ и рабочих процессов разработки.
- Поддержка размещения служб удаленного вызова процедур (RPC) с помощью gRPC.
- Облачная система конфигурации на основе среды.
- Встроенное введение зависимостей.
- Упрощенный высокопроизводительный модульный конвейер HTTP-запросов.
- Следующие возможности размещения:
 - Kestrel.
 - Службы IIS.
 - HTTP.sys.
 - Nginx.
 - Apache.
 - Docker.
- Управление параллельными версиями.
- Инструментарий, упрощающий процесс современной веб-разработки.

Выбор базы данных. Для разработки приложения использована СУБД Microsoft SQL Server [5, 8-9].

Microsoft SQL Server – это серверная система управления реляционными базами данных, разработанная компанией Microsoft [0].

Преимущества СУБД:

- Производительность.
- Надежность и безопасность.
- Простота использования.

Для работы с базой данных использован фреймворк Entity Framework.

Entity Framework – объектно-ориентированная технология от Microsoft, позволяющая работать с данными как с объектами, отделяя их от физической базы данных [4141]. Entity Framework работает поверх платформы .NET Core, благодаря чему его можно использовать при создании различных типов приложений, таких как:

- Консольные приложения [1, 22-24, 27].
- Приложения Windows Forms [6, 7].
- Приложения Windows Presentation Foundation (WPF) [26, 29-31].
- Приложения Universal Windows Platform (UWP) [34-36].
- Приложения ASP .NET Core [33].

Entity Framework поддерживает многие системы базы данных, благодаря чему программист может выбрать удобную ему базу данных для приложения.

Entity Framework позволяет удобным способом работать с БД, не используя SQL запросы, а обращаться к ней через объекты кода с помощью языка LINQ.

Согласно поставленной задаче, для разработки использованы следующие технологии:

- ASP .NET Core – платформа, предназначенная для разработки веб-приложений с использованием языка программирования C# на программной платформе .NET Core [32].
- MS SQL Server – СУБД от Microsoft [2].
- Entity Framework – набор технологий для работы с СУБД, позволяет генерировать SQL запросы посредством LINQ [41].
- LINQ – технология для выполнения высокоуровневых запросов на языке C# [4]. Является языком запросов к источникам данных [40].

Выводы по главе 1

В 1 главе исследована предметная область, для этого построена структурно-функциональная диаграмма отдела дополнительных профессиональных программ в нотации IDEF0.

Также произведен обзор программных средств и технологий, благодаря чему выбраны необходимые программные решения.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

2.1. Структурно-функциональная диаграмма «Как должно быть»

По ходу разработки бизнес – процесса почти полностью переработана диаграмма «Как есть». В итоге создана диаграмма «Как должно быть» [21, 25, 28].

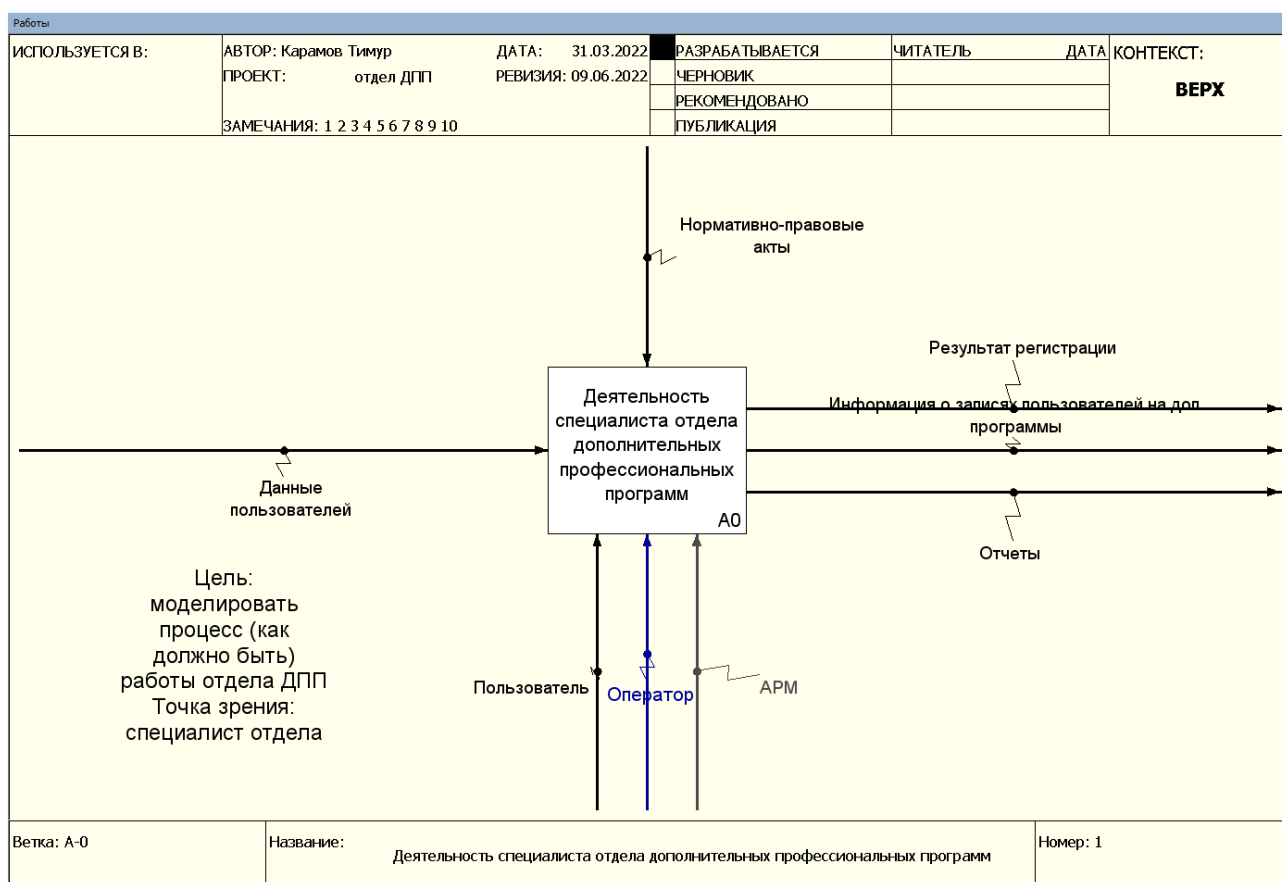


Рисунок 2.1 Контекстная диаграмма "Как должно быть"

После декомпозиции выявлены следующие блоки (Рисунок 2.5):

- Регистрация пользователя.
- Запись на различные программы.
- Просмотр заявок.
- Формирование отчетов.

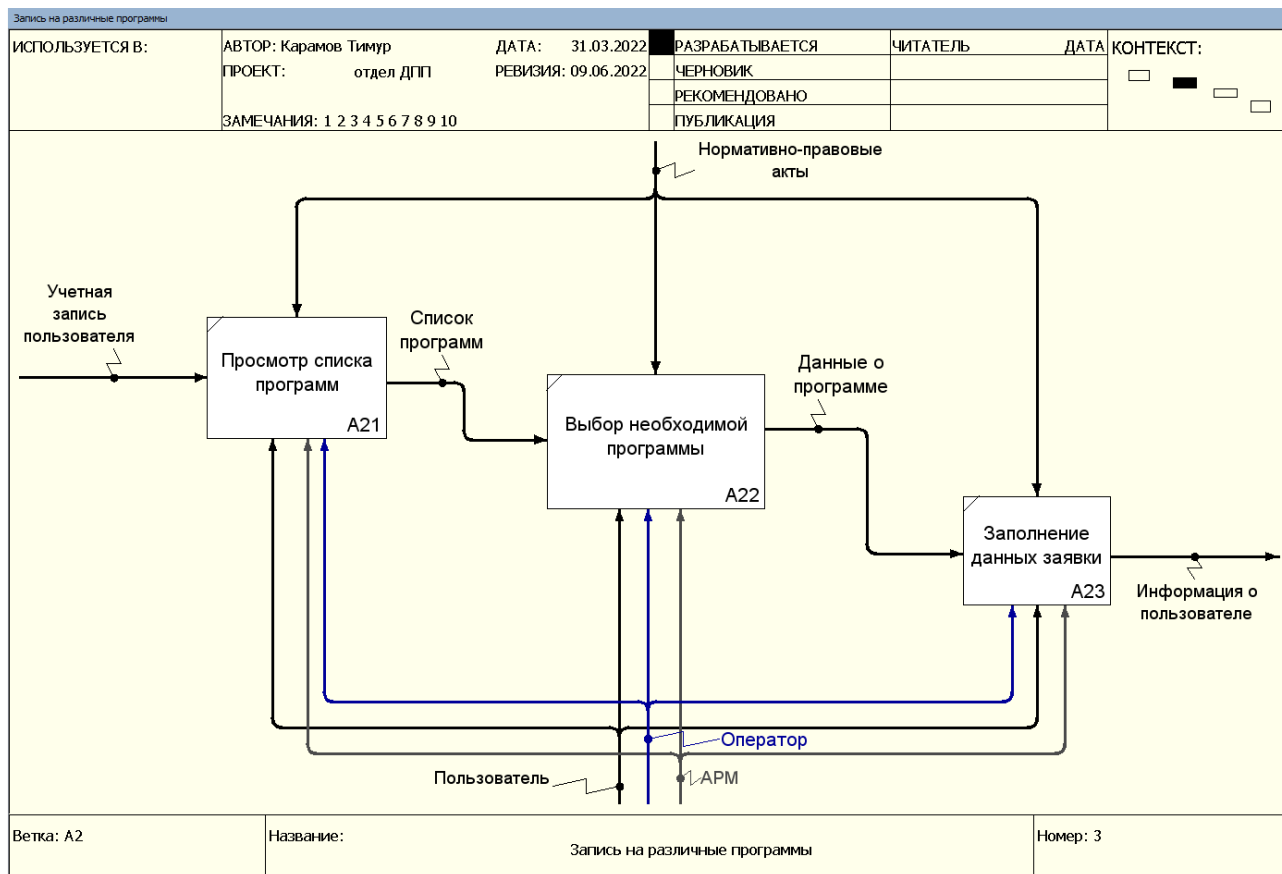


Рисунок 2.3 Декомпозиция блока «Запись на различные программы»

2.2. Диаграммы последовательностей

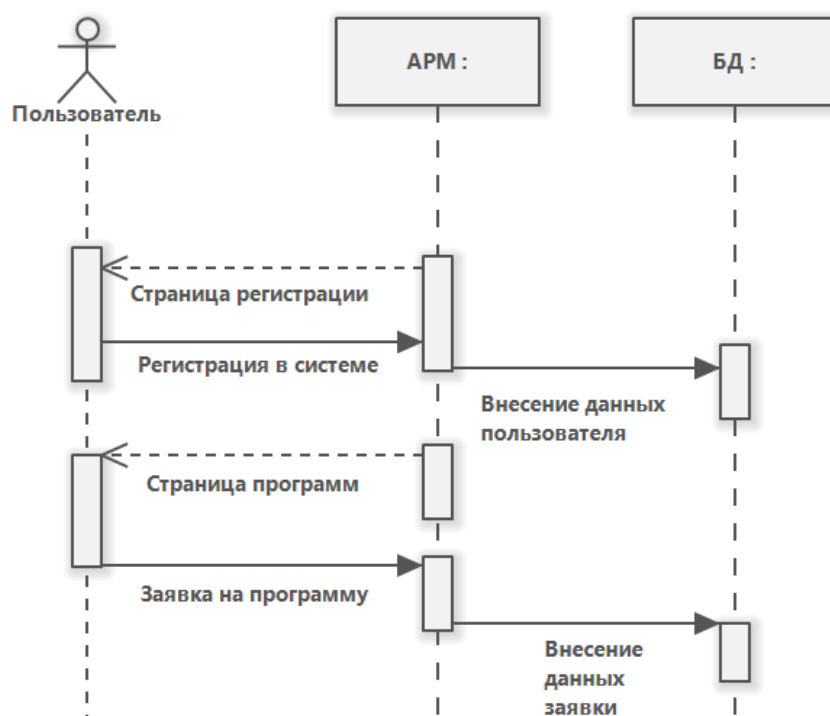


Рисунок 2.4 Диаграмма последовательности «Регистрация заявки пользователя на программы»

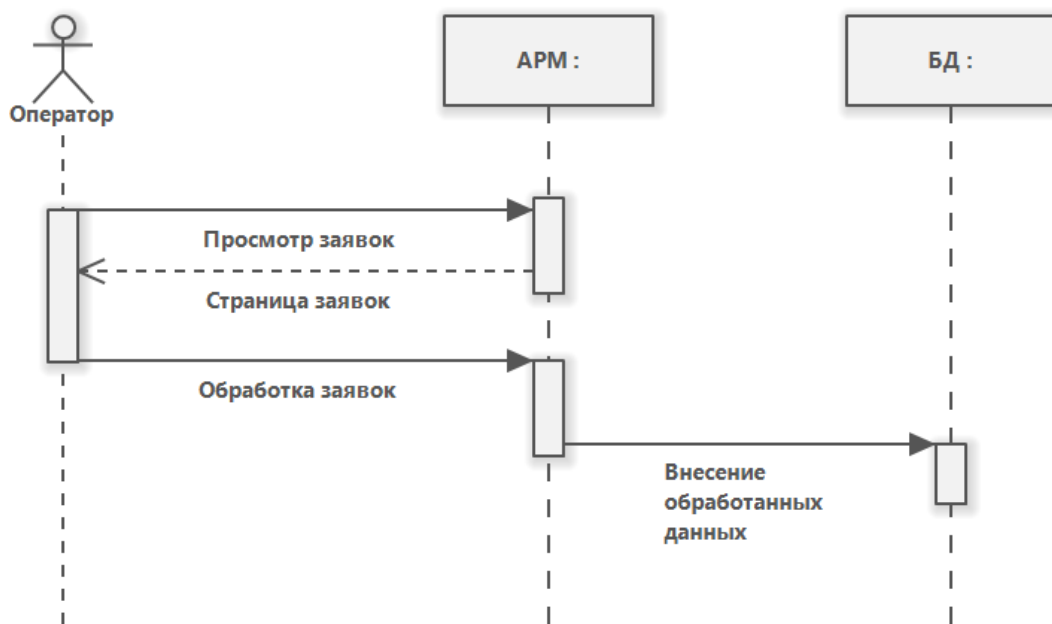


Рисунок 2.5 Диаграмма последовательности «Обработка заявок»

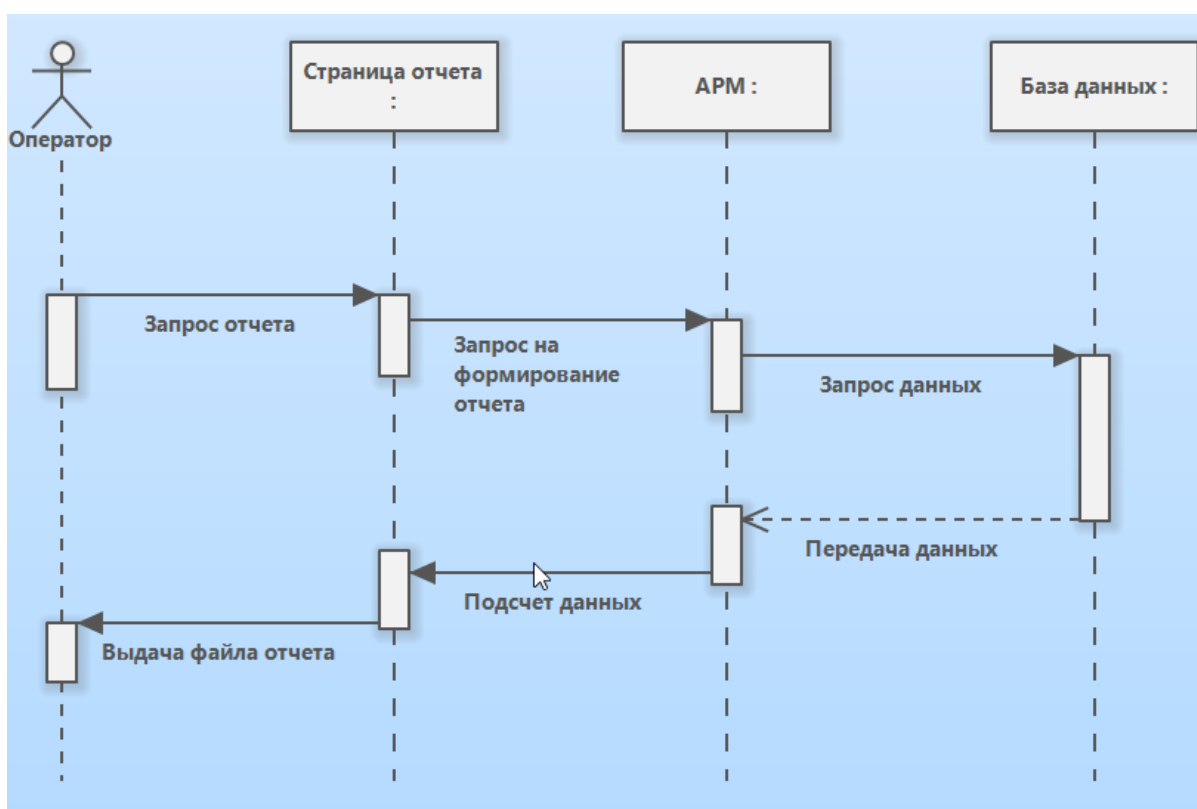


Рисунок 2.6 Диаграмма последовательности «Получение отчета»

2.3. Диаграммы деятельности

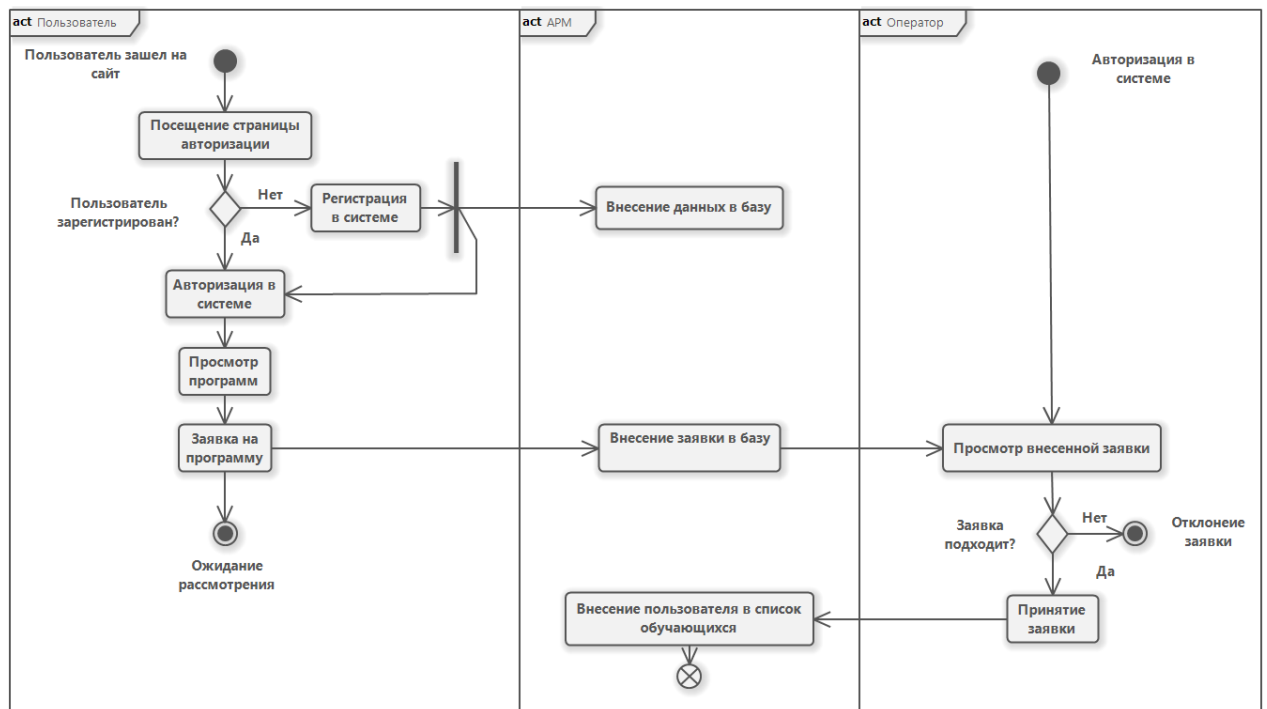


Рисунок 2.7 Диаграмма деятельности «Запись на программу и рассмотрение заявки»

2.4. Дatalogическая модель данных

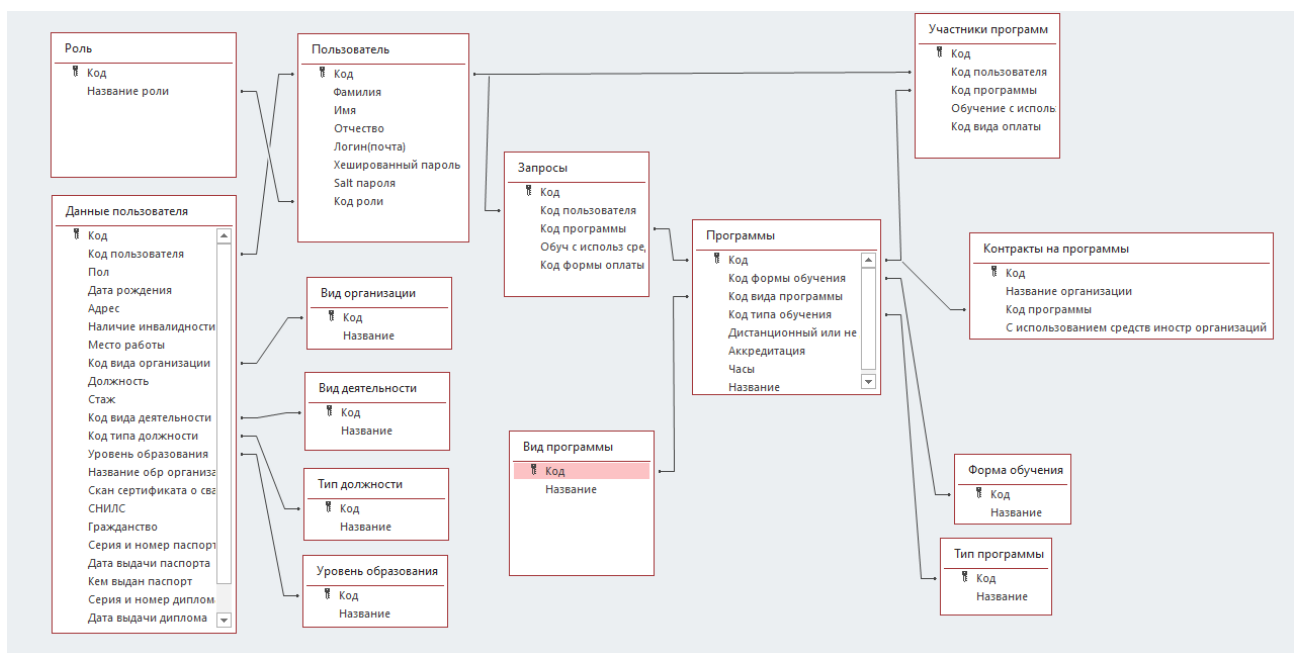


Рисунок 2.8 Дatalogическая модель данных

Таблица 2.1

Сущность: Пользователь

Атрибут	Тип	Описание
Код пользователя	Числовой	Содержит учетные записи
Имя	Текст	
Фамилия	Текст	
Отчество	Текст	
Логин	Текст	
Hash пароля	Текст	
Salt (соль) пароля	Текст	
Код роли	Числовой	

Таблица 2.2

Сущность: Роль

Атрибут	Тип	Описание
Код роли	Числовой	Содержит роли для учетных записей
Название роли	Текст	

Таблица 2.3

Сущность: Данные пользователя

Атрибут	Тип	Описание
Код данных	Числовой	Содержит данные пользователя
Код пользователя	Числовой	
Пол	Текст	
Дата рождения	Дата	
Адрес	Текст	
Место работы	Текст	
Должность	Текст	

Наличие инвалидности	Логический	
Код типа организации	Текст	
Код вида должности	Числовой	
Код вида деятельности	Числовой	
Стаж	Числовой	
Квалификация/ специальность/ профессия по образованию	Текст	
Код уровня образования	Числовой	
Имя организации, выдавшей документ об образовании	Текст	
Серия/номер диплома	Текст	
Дата выдачи диплома	Дата	
СНИЛС	Текст	
Гражданство	Текст	
Серия и номер паспорта	Текст	
Кем выдан паспорт	Текст	
Дата выдачи паспорта	Дата	
Скан диплома	Текст	
Приложение к диплому	Текст	
Копия свидетельства о браке	Текст	

Таблица 2.4

Сущность: Уровень образования

Атрибут	Тип	Описание
Код уровня	Числовой	Содержит уровни образования
Имя	Текст	

Таблица 2.5

Сущность: Тип должности

Атрибут	Тип	Описание
Код типа	Числовой	Содержит типы должностей
Имя	Текст	

Таблица 2.6

Сущность: Вид организации

Атрибут	Тип	Описание
Код вида	Числовой	Содержит виды организации
Имя	Текст	

Таблица 2.7

Сущность: Вид деятельности

Атрибут	Тип	Описание
Код вида	Числовой	Содержит виды деятельности
Имя	Текст	

Таблица 2.8

Сущность: Образовательная программа

Атрибут	Тип	Описание
Код курса	Числовой	Содержит курсы
Код формы обучения	Числовой	

Код типа обучения	Числовой	
Код вида программы	Числовой	
Полностью дистанционный/ электронный/ сетевой	Логический	
Аккредитован	Логический	
Название	Текст	
Часы	Числовой	

Таблица 2.9

Сущность: Контракт с организациями

Атрибут	Тип	Описание
Код контракта	Числовой	Содержит список договоров, заключенных с организациями на реализацию образовательных программ
Имя организации	Текст	
Код курса	Числовой	
С использованием ресурсов иностранных организаций	Логический	

Таблица 2.10

Сущность: Форма обучения

Атрибут	Тип	Описание
Код формы	Числовой	Содержит формы обучения
Имя	Текст	

Таблица 2.11

Сущность: Тип обучения

Атрибут	Тип	Описание
Код типа	Числовой	Содержит типы обучения
Имя	Текст	

Таблица 2.12

Сущность: Участник программ

Атрибут	Тип	Описание
Код	Числовой	Содержит список обучающихся в программах
Код обучающегося	Числовой	
Код программы	Числовой	
Обуч. с использ. ресурсов иностр. организаций	Логический	
Код вида оплаты	Числовой	

Таблица 2.13

Сущность: Заявка обучающегося

Атрибут	Тип	Описание
Код	Числовой	Содержит список заявок обучающихся
Код обучающегося	Числовой	
Код курса	Числовой	
Обучение с использованием ресурсов иностр. организаций	Логический	
Код вида оплаты	Числовой	

Сущность: Вид программы

Атрибут	Тип	Описание
Код типа	Числовой	Содержит типы обучения
Имя	Текст	

2.5. Проектирование интерфейса

Веб-интерфейс – веб-страницы, предоставляющие пользовательский интерфейс для взаимодействия с сайтом посредством графического интерфейса.

Для удобного доступа к сайту необходимо создать веб-интерфейс, отвечающий следующим требованиям:

- Общая стилистика – страницы должны иметь общий внешний вид.
- Удобство использования – страницы должны иметь простой, понятный интерфейс.
- Последовательность интерфейса – интерфейс должен соблюдать последовательность для простоты работы с ним.

Веб-интерфейс должен содержать следующие страницы:

- Страница авторизации – для авторизации пользователей системы.
- Страница регистрации – для создания новой учетной записи.
- Профиль пользователя – для просмотра данных пользователя.
- Страница редактирования профиля – для изменения данных пользователя.
- Страницы для образовательных программ – для просмотра списка и выполнения различных действий над программами – заявка на программу для пользователя, редактирование, удаление, добавление и запись пользователя для операторов.

- Страницы для договоров с организациями – для просмотра, добавления, изменения и удаления образовательных договоров.
- Страница со списком обучающихся – страница для просмотра и удаления обучающихся.
- Страницы списка пользователей – для просмотра, удаления, редактирования данных и ролей, записи пользователей на образовательные программы.
- Страница для получения отчета – для скачивания отчета по обучающимся.

Выводы по главе 2

Во 2 главе проведен анализ и проектирование модели базы данных, построены диаграммы последовательности, диаграмма в нотации IDEF0, даталогическая модель базы данных, а также выявлены требования и произведено проектирование интерфейса.

ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

3.1. Реализация интерфейса

Интерфейс программы представляет собой специальный формат страниц CSHTML, который использует HTML и код C# [10-13, 15-20]. Пример кода приведен ниже:

```
@model RoleModel;

@{
    ViewData["Title"] = "Изменение роли пользователя";
}

<body>
    <div class="container mx-auto text-justify">
        <form class="justify-content-center container"
style="width:40%" asp-action="ChangeRole" method="post">
            <h3 class="h3 text-center">Изменение роли</h3>
            @Html.HiddenFor(r => r.UserId)
            <div style="width:40%">
                <div class="form-group mb-4">
                    <label class="form-text">Текущая роль
- @Model.RoleName</label>
                </div>
                <div class="form-group mb-4">
                    <label class="form-text" asp-
for="RoleId">Новая роль</label>
                    <select asp-for="RoleId" required asp-
items="ViewBag.Roles" class="form-control"></select>
                </div>
                <div class="col-md-8 mx-auto mt-4">
                    <input type="submit" value="Изменить"
class="btn btn-primary" />
                </div>
            </div>
        </form>
    </div>
</body>
```


Для передачи данных на страницу используются Tag-Helper'ы, которые подключаются сверху страницы с помощью конструкции @model «Название_модели».

Также, для передачи данных на страницу можно использовать специальные объекты – ViewBag, которые могут хранить в себе любые необходимые данные. Ниже приведен пример создания ViewBag, которые хранят в себе списки для <select>:

```
var edulvl = await _context.EduLevels.Select(x =>
x).ToListAsync<EduLevel>();
var subw = await _context.SubWorks.Select(x =>
x).ToListAsync<SubWork>();
var workp = await _context.WorkplaceTypes.Select(x =>
x).ToListAsync<WorkplaceType>();
var workt = await _context.WorkTypes.Select(x =>
x).ToListAsync<WorkType>();
ViewBag.EduLevels = new SelectList(edulvl, "Id", "Name");
ViewBag.SubWork = new SelectList(subw, "Id", "Name");
ViewBag.WorkplaceType = new SelectList(workp, "Id",
"Name");
ViewBag.WorkType = new SelectList(workt, "Id", "Name");
```

В коде самой страницы данные из ViewBag можно привязать к тегу <select> при помощи Tag-Helper'a asp-items:

```
<div class="form-group form-outline mb-4">
<label asp-for="WorkTypeId">Вид деятельности*</label>
<select required asp-for="WorkTypeId" asp-
items="ViewBag.WorkType" class="form-control"></select>
</div>

<div class="form-group form-outline mb-4">
<label asp-for="WorkplaceTypeId">Тип организации*</label>
<select required asp-for="WorkplaceTypeId" asp-
items="ViewBag.WorkplaceType" class="form-control"></select>
</div>
```

Благодаря ASP.NET не требуется каждый раз подключать панель навигации, программа сама подключает ее, если это необходимо программисту.

Оформление панели описано в отдельном файле Layout.cshtml. Этот файл состоит из нескольких блоков. В теге <head> описаны все необходимые для создания интерфейса компоненты:

```
<link rel="stylesheet"
href="~/lib/bootstrap/dist/css/bootstrap.css" />
<link rel="stylesheet" href="~/css/bootstrap.css" />
<link rel="stylesheet" href="~/css/site.css" />
<link rel="preconnect"
href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@
400&display=swap" rel="stylesheet">
```

В теге <body> файла расположено оформление:

- навигационной панели сайта;

```
<header>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a class="navbar-brand" asp-controller="Home" asp-
action="Index">ДПП</a>
<button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarTogglerDemo02" aria-
controls="navbarTogglerDemo02" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse"
id="navbarTogglerDemo02">
<ul class="navbar-nav mr-auto mb-2 mb-lg-0">
```

- основная часть страницы, вызываемая специальным методом RenderBody();

```
<div class="container">
<main role="main" class="pb-3">
@RenderBody()
```

```

        </main>
    </div>

```

- footer сайта.

```

<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2021 - ДПП - <a asp-area="" asp-
controller="Home" asp-action="Privacy">Privacy</a>
    </div>
</footer>

```

Для внешнего оформления страниц использована библиотека Bootstrap.

Bootstrap – это бесплатный JS, HTML, CSS фреймворк, используемый для быстрой верстки адаптивных сайтов и веб-приложений [38].

Данный фреймворк содержит в себе множество различных стилей и методов для создания интерфейса.

Например, для создания навигационной панели сайта используются стили для <nav>:

```

<header>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a class="navbar-brand" asp-controller="Home" asp-
action="Index">ДПП</a>
<button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarTogglerDemo02" aria-
controls="navbarTogglerDemo02" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse"
id="navbarTogglerDemo02">
<ul class="navbar-nav mr-auto mb-2 mb-lg-0">

```

Рисунок 3.1 Внешний вид навигационной панели

Также, для создания вложенных списков в панели можно использовать различные стили и методы Bootstrap:

```
<li class="nav-item dropdown">
  <a class="nav-link text-dark dropdown-toggle" role="button"
id="dropdownAdmin"          data-toggle="dropdown"          aria-
expanded="false">Администрирование</a>
  <ul class="dropdown-menu" aria-labelledby="dropdownAdmin">
    <li><a class="dropdown-item" asp-action="GetReport" asp-
controller="Report">Отчеты</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" asp-action="Requests" asp-
controller="Course">Список заявок</a></li>
    <li><a class="dropdown-item" asp-action="ParticipantsList"
asp-controller="Course">Список обучающихся</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" asp-action="EducationContracts"
asp-controller="Course">Список
договоров с организациями</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" asp-action="UserList" asp-
controller="User">Список пользователей</a></li>
  </ul>
</li>
```

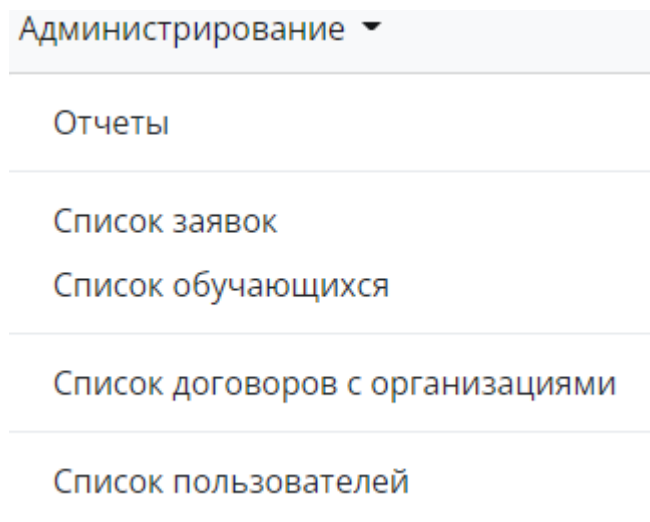


Рисунок 3.2 Внешний вид списка

3.2. Реализация подключения интерфейса к базе данных

Подключение интерфейса к базе данных реализуется за счет фреймворка MVC (model – view – controller), который позволяет привязать различные действия из интерфейса к методам программы:

```
<a class="nav-link active" asp-controller="Home" asp-action="Index">Главная</a>
```

Для передачи данных в метод есть 2 способа:

- Передача данных в привязанную модель через Tag-Helpers `asp-for`

```
<label asp-for="Password">Введите новый пароль</label>
<input type="password" asp-for="Password" class="form-control" />
<span asp-validation-for="Password" />
```

- Передача данных в метод через Tag-Helpers `asp-route-«название переменной в методе»`

```
<a class="btn btn-primary btn-sm mb-2" asp-action="EditCourse"
  asp-controller="Course" asp-route-id="@c.Id">Изменить</a>
```

Каждый метод находится в своем контроллере и реализует связь с базой данных с помощью Entity Framework и LINQ to Entities:

```
public async Task<IActionResult> Courses()
{
    var model = await _context.Courses.Include(c =>
c.EduForm).Include(c => c.EduType).ToListAsync();
    if (User.IsInRole("User"))
    {
        var user = await
        _context.Users.FirstOrDefaultAsync(u => u.Login ==
        User.Identity.Name);
        var uid = user.UserId;
        var cparts = _context.CParts.Where(cparts =>
        cparts.UserId == uid).Select(cp => cp.CourseId);
        foreach (var i in cparts)
        {
```

```

        model = model.Where(m => m.Id != i).ToList();
    }
}
return View(model);
}

```

3.3. Реализация базы данных

Различные действия с базой данных (запись, запрос, изменение, удаление и т.д.) также производятся через Entity Framework.

Entity Framework реализует подход Code First, подразумевающий написание кода модели (таблицы) на языке C# и создания таблицы в базе данных на его основе:

```

public class User
{
    public int UserId { get; set; }
    public string Name { get; set; }
    public string Surname { get; set; }
    public string? Patronymic { get; set; }
    public string Login { get; set; }
    public string PasswordHash { get; set; }
    public byte[] PasswordSalt { get; set; }
    public Role Role { get; set; }
    public int RoleId { get; set; }
}

```

Для привязки моделей к базе данных используется контекст, который, в свою очередь, создает модели в базе и позволяет работать с таблицами БД с помощью LINQ.

```

public class DPPContext: DbContext
{
    public DbSet<EduLevel> EduLevels { get; set; }
    public DbSet<User> Users { get; set; }
    public DbSet<UserInfo> UserInfos { get; set; }
    public DbSet<Role> Roles { get; set; }
    public DbSet<EducationType> EduTypes { get; set; }
    public DbSet<EducationForm> EduForms { get; set; }
}

```

```

public DbSet<Course> Courses { get; set; }
    public DbSet<EducationContract> EduContracts { get;
set; }
public DbSet<CourseParticipant> CParts { get; set; }
public DbSet<CourseRequest> Requests { get; set; }
public DbSet<PaymentForm> PaymentForms { get; set; }
public DbSet<SubWork> SubWorks { get; set;}
    public DbSet<WorkplaceType> WorkplaceTypes { get;
set; }
public DbSet<WorkType> WorkTypes { get; set; }
public DPPContext(DbContextOptions<DPPContext> options):
base(options)
    {
        Database.EnsureCreated();
    }
}

```

Для реализации запросов к базе данных используется LINQ.

Language Integrated Query (LINQ) – это SQL-подобный язык запросов, используемый для обращения к различным объектам в коде [37].

Для реализации запросов к базе данных используется LINQ to Entites – обращение к базе данных через Entity Framework с использованием LINQ.

Запросы могут быть выполнены как напрямую к контексту, так и через паттерн «Репозиторий», который являются оберткой LINQ запросов к БД:

Взаимодействие через репозитории:

```

await _course.CreateAsync(course);
await _course.SaveAsync();

```

Взаимодействие напрямую:

```

var _ef = await _context.EduForms.ToListAsync();
var _et = await _context.EduTypes.ToListAsync();

```

3.4. Руководство пользователя

Обычный пользователь. При заходе на главную страницу сайта (Рисунок 3.3) пользователь должен авторизоваться (Рисунок 3.4) или зарегистрироваться (Рисунок 3.5).

Добро пожаловать

Рисунок 3.3 Главная страница сайта

Авторизация

admin@mail.ru

Войти

Рисунок 3.4 Страница авторизации

Регистрация

Введите фамилию

Введите имя

Введите отчество

Введите E-mail

admin@mail.ru

Введите пароль

Введите пароль еще раз

Пол

Мужской

Дата рождения *

дд.мм.гггг



Адрес *

Рисунок 3.5 Страница регистрации

После авторизации пользователь может просмотреть свой профиль (Рисунок 3.6) или список образовательных программ (Рисунок 3.7).

Профиль

Изменить

Основная информация

ФИО: Карамов Тимур Ирикович

Пол: Мужской

Дата рождения: 05.02.2001

Адрес: Интернациональная д.76, кв. 48

Информация о работе

Место работы:

Должность:

Опыт работы (лет):

Информация об образовании

Образование: Программист

Уровень образования: Подготовка кадров высшей квалификации

Название образовательной организации: БФ БашГУ

Документы

СНИЛС: 123-123-123 12

Гражданство: Российская Федерация

Паспорт, серия/номер: 1111 111111

Паспорт, кем выдано: МВД Бирского района

Паспорт, дата выдачи: 15.02.2021

Информация о дипломе

Диплом, серия/номер: 123456 1234567

Диплом, дата выдачи: 11.11.2021

Сканы диплома

Рисунок 3.6 Профиль пользователя

ДПП Главная Программы								П
Id	Название	Форма обучения	Тип программы	Часы	Форма проведения	Аккредитован	Функции	
1	Курс 4	Очно-заочная	Программа профессиональной переподготовки	90	Дистанционная	Нет	Функции ▼	
2	Курс 3	Очная	Программа повышения квалификации	1000	Дистанционная	Нет	Функции ▼	
3	Курс 2	Заочная	Программа профессиональной переподготовки	76	Дистанционная	Нет	Функции ▼	
4	Курс 1	Очно-заочная	Программа профессиональной переподготовки	60	Не дист./Частично дист.	Да	Функции ▼	

Рисунок 3.7 Список образовательных программ

В списке программ пользователь может подать заявку, заполнив необходимые данные

ДПП Главная Программы

Название программы: Курс 4

Обучение с использованием ресурсов иностранных организаций ☒

Форма оплаты:

Обучение за счет бюджетных ассигнований федерального бюджета

Отправить

Рисунок 3.8 Страница заявки

Также пользователь может просмотреть свой список заявок (Рисунок 3.9) и программ (Рисунок 3.10):

Список заявок

Код	Название программы	Обуч. с использ. ресурсов иностр. орг.	Форма оплаты	Функции
1	Курс 4	Да	Обучение за счет средств физических лиц	Функции ▼

Отменить

Рисунок 3.9 Список заявок пользователя

Список программ

Код	Название программы	Обуч. с использ. ресурсов иностр. орг.	Форма оплаты
6	Курс 4	Да	Обучение за счет средств физических лиц

Рисунок 3.10 Список программ пользователя

Руководство оператора. После авторизации администратор может выполнить следующие действия:

1. Просмотреть список программ, чтобы удалить, изменить существующую, записать на нее пользователя (Рисунки 3.11, 3.12), добавить новую программу (Рисунок 3.13)

Главная

Программы

Администрирование

Профиль

Выход

Добавить программу

Id	Название	Форма обучения	Тип программы	Часы	Форма проведения	Аккредитован	Функции
1	Курс 4	Очно-заочная	Программа профессиональной переподготовки	90	Дистанционная	Нет	<div>Функции</div> <div>Изменить</div> <div>Удалить курс</div>
2	Курс 3	Очная	Программа повышения квалификации	1000	Дистанционная	Нет	<div>Записать пользователя на программу</div>
3	Курс 2	Заочная	Программа профессиональной переподготовки	76	Дистанционная	Нет	
4	Курс 1	Очно-заочная	Программа профессиональной переподготовки	60	Не дист./Частично дист.	Да	

Рисунок 3.11 Страница с программами

ДПП Главная Программы Администрирование ▾

Запись пользователя на программу

Название программы:

Курс 4

Выберите пользователя

Семенов Семен Петрович ▾

Обучение с использ. ресурсов иностр. организаций

☐

Форма оплаты

Обучение за счет бюджетных ассигнований федерального бюджета ▾

Записать

Рисунок 3.12 Запись пользователя

ДПП Главная Программы Администрирование ▾

Добавить программу

Название

Тип

Программа профессиональной переподготовки ▾

Форма

Очно-заочная ▾

Часы

Аккредитация

☐

Дистанционная

☐

Добавить

Рисунок 3.13 Добавление новой программы

2. Просмотреть список заявок и обработать их (Рисунок 3.14)

ДПП Главная Программы Администрирование ▾ Профиль Выйти

Список заявок

Код	Название программы	ФИО пользователя	Обуч. с использ. ресурсов иностр. орг.	Форма оплаты	Функции
1	Курс 4	Карамов Тимур Ирикович	Да	Обучение за счет средств физических лиц	Функции ▾

Принять
Отклонить

Рисунок 3.14 Страница заявок

3. Просмотреть список обучающихся (Рисунок 3.15)

ДПП Главная Программы Администрирование ▾ Профиль Выйти

Список обучающихся

Код	Название программы	ФИО пользователя	Обуч. с использ. ресурсов иностр. орг.	Форма оплаты	Функции
1	Курс 2	Семенов Семен Петрович	Нет	Обучение за счет средств физических лиц	Функции ▾
2	Курс 3	Петров Петр Иванович	Нет	Обучение за счет средств юридических лиц	Удалить
3	Курс 4	Семенов Семен Петрович	Да	Обучение за счет средств физических лиц	Функции ▾

Рисунок 3.15 Страница списка обучающихся

4. Просмотреть список договоров с организациями (Рисунки 3.16-3.18)

ДПП Главная Программы Администрирование ▾

Профиль Выйти

Добавить контракт

Id	Название	Название программы	Обуч. с использ. ресурсов иностр. организ.	Функции
1	Организация 2	Курс 4	Нет	Функции ▾
2	Организация 1	Курс 1	Да	

Изменить

Удалить контракт

Рисунок 3.16 Страница договоров

ДПП Главная Программы Администрирование ▾

Добавление договора

Название

Программа

Курс 4 ▾

Использ. ресурсов иностр. организаций

☐

Добавить

Рисунок 3.17 Добавление нового договора

ДПП Главная Программы Администрирование ▾

Название организации

Организация 2

Программа

Курс 4 ▾

Обуч. с использ. ресурсов иностр. организ. ☐

Сохранить

Рисунок 3.18 Изменение существующего договора

5. Просмотреть список пользователей, чтобы добавить, изменить данные (Рисунок 3.19), просмотреть профиль, изменить роль (Рисунок 3.20), удалить пользователя (Рисунок 3.21) или записать на программу (Рисунок 3.22).

ДПП Главная Программы Администрирование ▾

Редактирование данных

Введите фамилию

Семенов

Введите имя

Семен

Введите отчество

Петрович

Введите новый пароль

Пол

Мужской ▾

Дата рождения *


01.01.0001 

Рисунок 3.19 Изменение данных пользователя

ДПП Главная Программы Администрирование ▾

Изменение роли

Текущая роль - User

Новая роль

User ▾

Изменить

Рисунок 3.20 Изменение роли

ДПП Главная Программы Администрирование ▾

Список пользователей

Добавить нового пользователя

# пользователя	ФИО	Роль	Функции
1	Семенов Семен Петрович	User	Функции ▾
2	Петров Петр Иванович	User	Профиль
3	Иванов Иван Иванович	MainAdmin	Изменить роль
4	Карамов Тимур Ирикович	User	Изменить данные
			Удалить

Рисунок 3.21 Удаление пользователя

ДПП Главная Программы Администрирование ▾

Запись пользователя на программу

ФИО Пользователя:

Семенов Семен Петрович

Выберите программу

Курс 1 ▾

Обучение с использ. ресурсов иностр. организаций

☒

Форма оплаты

Обучение за счет средств юридических лиц ▾

Записать

Рисунок 3.22 Запись пользователя на программу

6. Сформировать отчет по программам и обучающимся (Рисунки 3.23, 3.24)

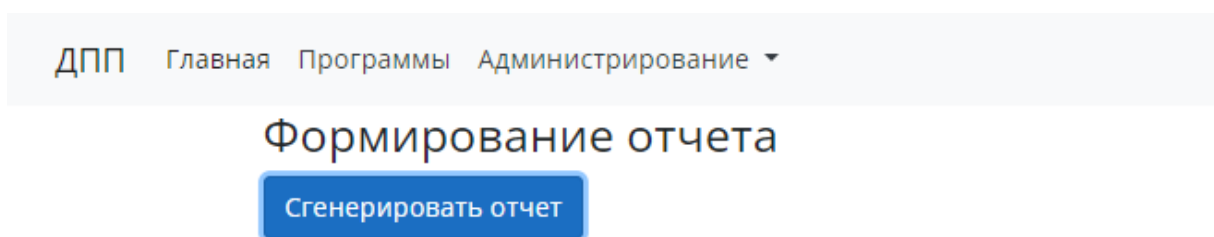


Рисунок 3.23 Страница генерации отчета

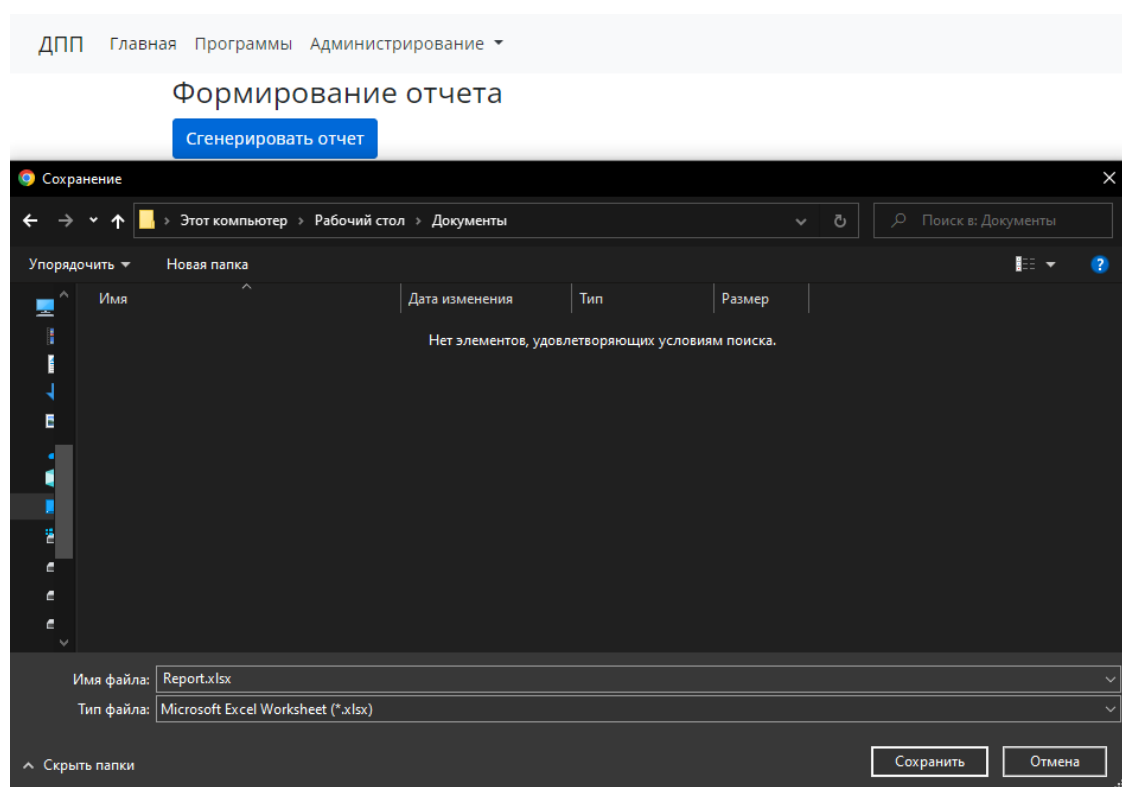


Рисунок 3.24 Отчет

Выводы по главе 3

В ходе работы разработан и реализован интерфейс, реализована связь интерфейса с кодом программы, реализовано подключение к базе данных и составлено руководство пользователя программы.

ЗАКЛЮЧЕНИЕ

Проведено исследование предметной области отдела дополнительных профессиональных программ, выполнено моделирование текущего состояния в нотации IDEF0, выявлены необходимые для автоматизации функции.

Автоматизированное рабочее место спроектировано с помощью UML-диаграмм, диаграммы работы системы в нотации IDEF0, даталогической модели базы данных и выявлены требования к интерфейсу.

Приложение разработано с помощью IDE Visual Studio 2022 (C#). Для разработки приложения использовался фреймворк ASP.NET Core MVC. Интерфейс создан с помощью стандартных средств HTML и C#, с использованием графического фреймворка Bootstrap. Подключение интерфейса к базе данных реализовано с помощью Entity Framework и LINQ. Также создано руководство пользователя.

*Я подтверждаю, что настоящая работа написана мной лично,
не нарушает интеллектуальные права третьих лиц и не
содержит сведений, составляющих государственную тайну*

Карамов / Карамов Тимур Иринович

09.06.2022

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Агуров П. В. С#. Сборник рецептов / Агуров П. В. – СПб.:БХВ-Петербург, 2007. – 412с.
2. Аллен, Г. Тейлор SQL для чайников / Аллен Г. Тейлор. - М.: Диалектика, Вильямс, 2015. - 416 с.
3. Анисимова Г. Б., Романенко М. В. Выбор методологии проектирования информационных систем. Структурный подход // Научное обозрение. 2017. №22. С. 74–79.
4. Арсеновски Даниэль Рефакторинг в С# и ASP.NET для профессионалов; Вильямс - М., 2010. - 528 с.
5. Баженова, И. Ю. Основы проектирования приложений баз данных / Баженова И. Ю. – М.: Национальный Открытый Университет "ИНТУИТ", 2016. – 215 с.
6. Ватсон, К. С#. / К. Ватсон, М. Беллиназо, О. Корнс, Д. Эспиноза. - М.: Лори, 2005. - 863 с.
7. Ватсон, К. Visual С# 2010: Полный курс. - М.: Диалектика, 2013. - 356 с.
8. Вирт, Н. Алгоритмы и структуры данных. - СПб: Невский диалект, 2008.
9. Горев А., Макашарипов С., Ахаян Р. Эффективная работа с СУБД. - СПб: «Питер», 2004. - 704с.
10. Дейтел Ч. С# / Дейтел Ч., Дейтел П., Листфилд Дж. и др. – СПб.:БХВПетербург, 2006. – 1056с.
11. Джозеф Албахари, Бен Албахари. С# 6.0. Справочник. Полное описание языка = С# 6.0 in a Nutshell: The Definitive Reference. — М.: «Вильямс», 2018. — 1040 с.
12. Джон Скит. С# для профессионалов: тонкости программирования, 3-е издание, новый перевод = С# in Depth, 3rd ed. — М.: «Вильямс», 2014. — 608 с.

13. Ишкова Э. А. Самоучитель C#. Начала программирования; Наука и техника - М., 2013. - 496 с.
14. Каменнова М.С. Моделирование бизнес-процессов. Часть 1: учебник и практикум для академического бакалавриата. – М.: Издательство Юрайт, 2019. — 282 с.
15. Климов А. C#. Советы программистам; БХВ-Петербург - М., 2008. - 544 с.
16. Кристиан Нейгел и др. C# 5.0 и платформа .NET 4.5 для профессионалов = Professional C# 5.0 and .NET 4.5. — М.: «Диалектика», 2013. — 1440 с.
17. Кристиан Нейгел, Карли Уотсон и др. Visual C# 2010: полный курс = Beginning Microsoft Visual C# 2010. — М.: Диалектика, 2010.
18. Культин Н. Microsoft Visual C# в задачах и примерах; БХВ-Петербург - М., 2015. - 320 с.
19. Культин Никита Борисович Основы программирования в Microsoft Visual C# 2010; БХВ-Петербург - М., 2011. - 384 с.
20. Мак-Дональд Мэтью, Шпушта Марио Microsoft ASP.NET 2.0 с примерами на C# 2005 для профессионалов; Вильямс - М., 2007. - 920 с.
21. Маклаков С.В. Моделирование бизнес-процессов. М.: ДИАЛОГМИФИ, 2002. – 180 с.
22. Марченко, А. Л. Основы программирования на C# 2.0 / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. - 552 с.
23. Пахомов Борис, C# для начинающих; БХВ-Петербург - М., 2014. - 917 с.
24. Петцольд Ч. Программирование для Microsoft Windows на C#. Пер. с англ.— М.: Издательско-торговый дом «Русская Редакция», 2002. – 576с.
25. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: Манн, Иванов и Фербер, 2013. – 544 с.

26. Рихтер Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#; Питер - М., 2012. - 928 с.
27. Секунов Н. Самоучитель C#; БХВ-Петербург - М., 2001. - 576 с.
28. Силич В.А., Силич М.П. Моделирование и анализ бизнес-процессов: учебное пособие. – Томск: Издательство Томск. гос. ун.-та систем управления и радиоэлектроники, 2011. – 212 с.
29. Скит Джон C# для профессионалов. Тонкости программирования; Вильямс - М., 2014. - 608 с.
30. Смоленцев, Н. К. MATLAB. Программирование на Visual C#, Borland JBuilder, VBA (+ CD-ROM) / Н.К. Смоленцев. - М.: ДМК Пресс, 2011. - 456 с.
31. Стилмен Э. Изучаем C#: учебное пособие / Э. Стилмен, Дж. Грин — 3-е изд. — Санкт-Петербург: Питер, 2014. — 816 с.
32. Фримен Адам ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов; Вильямс - М., 2015. - 736 с.
33. Фримен Адам, Сандерсон Стивен ASP.NET MVC 3 Framework с примерами на C# для профессионалов; Вильямс - М., 2011. - 672 с.
34. Шилдт Г. Полное руководство C# 4.0 [Текст]: учебное пособие / Г. Шилдт — пер. с англ. Берштейн И. В. — Москва: Вильямс, 2012.— 1051 с.
35. Э. Стилмен, Дж. Грин. Изучаем C#. 2-е издание = Head First C#, 2ed. — СПб.: «Питер», 2012. — 704 с.
36. Эндрю Троелсен Язык программирования C# 5.0 и платформа .NET 4.5; Диалектика / Вильямс - М., 2015. - 126 с.
37. Фримен Адам , Раттц-мл. Джозеф LINQ. Язык интегрированных запросов в C# 2010 для профессионалов; Вильямс - М., 2011. - 656 с.
38. Introduction [Электронный ресурс]. Режим доступа: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>, свободный. – (Дата обращения 18.04.2022)

39. Microsoft SQL Server [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Microsoft_SQL_Server, свободный. – (Дата обращения 12.05.2022)

40. Основы LINQ [Электронный ресурс]. Режим доступа: <https://metanit.com/sharp/tutorial/15.1.php>, свободный. – (Дата обращения 21.11.2021)

41. Что такое Entity Framework Core [Электронный ресурс]. Режим доступа: <https://metanit.com/sharp/efcore/1.1.php>, свободный. – (Дата обращения 28.04.2022)

Карашин / Карашин Т.И.