

Минобрнауки России
федеральное государственное автономное образовательное учреждение высшего
образования "Санкт-Петербургский университет Петра Великого"
Институт дополнительного образования
Высшая инженерная школа

Выпускная квалификационная работа

по программе профессиональной переподготовки "Разработчик прикладного программного обеспечения(Язык C#)"

**"Создание прототипа информационной системы для отделения платных
образовательных услуг образовательного учреждения"**

Выполнил: Прохоров Д.В.
Руководитель: Туральчук К.А.

Санкт-Петербург
2022

Цель и задачи работы

- Цель ВКР: разработать многопользовательское веб-приложение с базой данных для работы администратора, преподавателей, а так же для информирования родителей учащихся.
- Задачи:
 - разработать backend часть приложения на ASP.NET Core MVC;
 - разработать frontend часть приложения в виде html страниц;
 - разработать базу данных для хранения информации;
 - провести тестирование приложения.

Функциональные требования

Приложение должно обеспечивать следующую функциональность:

- авторизация и аутентификация пользователей;
- предоставление администратору возможности создания пользователей и определение их роли;
- предоставление прав и возможностей пользователям при работе с приложением в зависимости от их роли;
- отображение данных из таблиц базы данных по запросам пользователей;
- возможность внесения корректировок в базе данных в зависимости от изменений в работе в образовательном учреждении.

Почему ASP.NET MVC ?

ASP.NET MVC является многофункциональной платформой для создания веб-приложений и API-интерфейсов с помощью структуры проектирования Model-View-Controller.

Преимущества ASP.NET MVC:

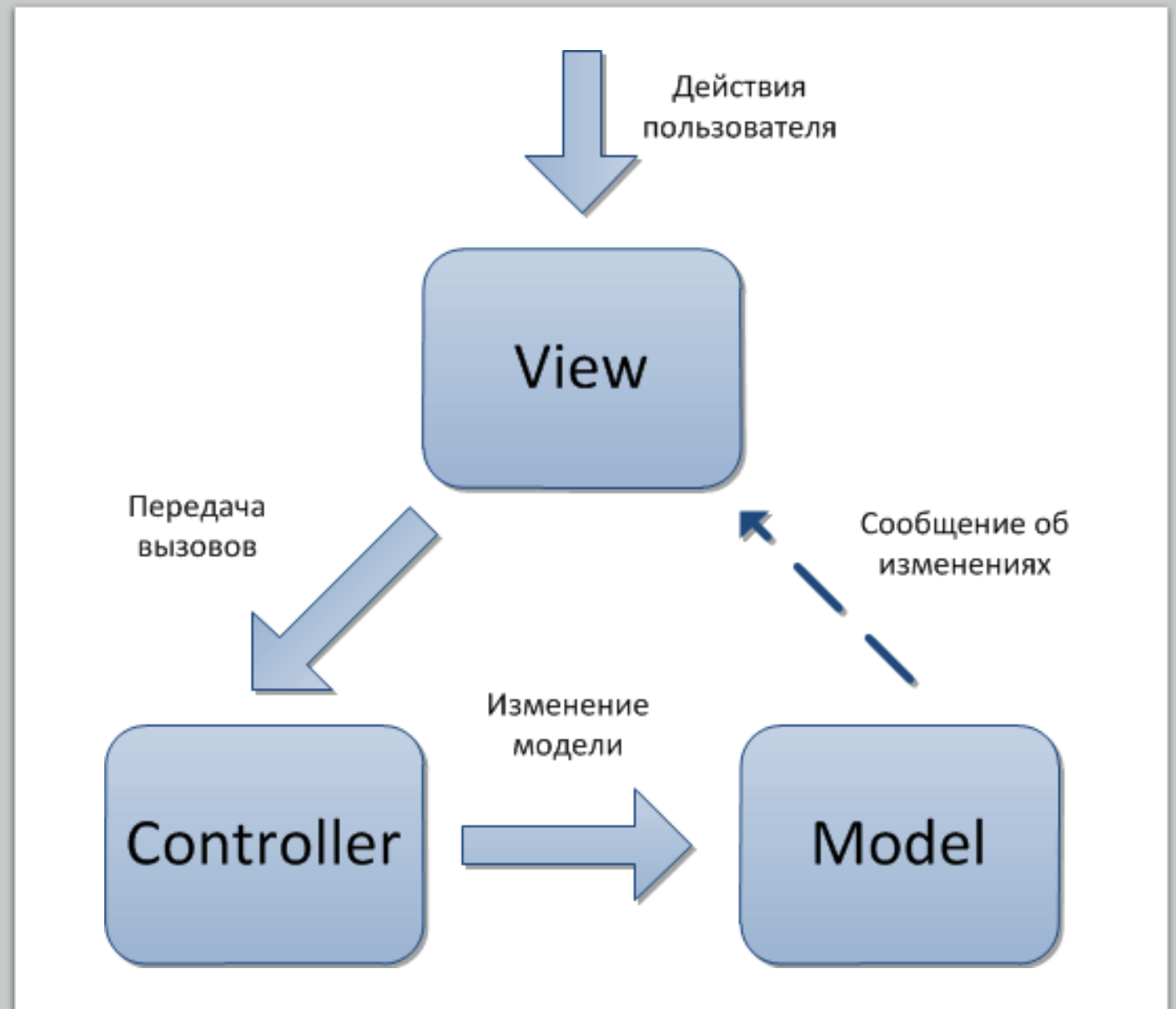
- высокая скорость разработки;
- разделение приложения на три компонента: модель, вид, контроллер.
- удобное тестирование и отладка кода;
- способность к развитию и усложнения проекта без потери качества;
- встроенная система авторизации и аутентификации;

Архитектурный шаблон MVC

Модель(model) предоставляет данные и методы работы с ними.

Представление(View) отвечает за получение необходимых данных из модели и отправляет их пользователю.

Контроллер(Controller) обеспечивает связь между пользователем и системой, направляет данные от пользователя к системе и наоборот.

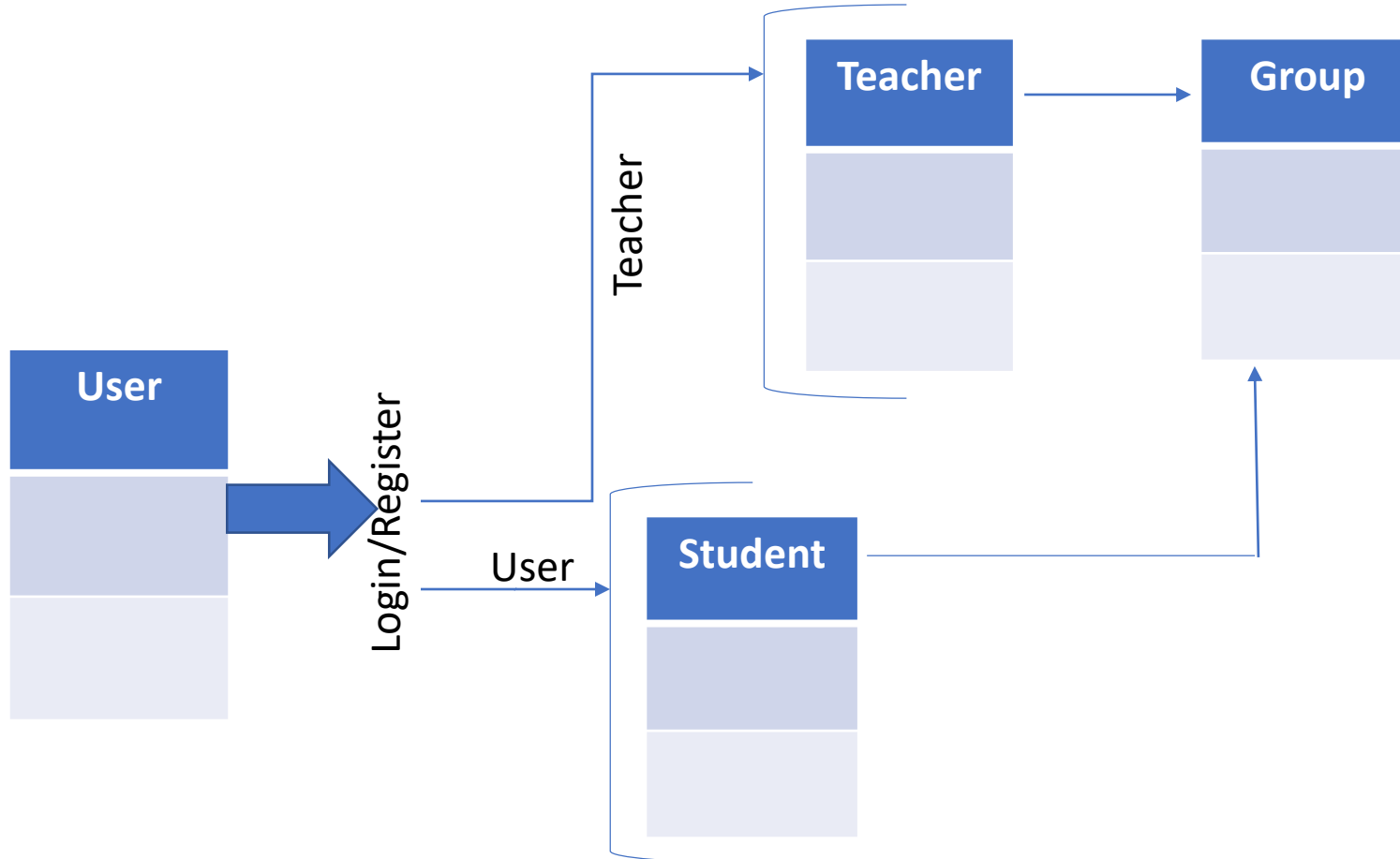


Авторизация и Аутентификация

ASP.NET Identity

IdentityDbContext	IdentityUser	UserManager	RoleManager
Users	Id	ChangePasswordAsync(user, old,new)	CreateAsync
Roles	UserName	CreateAsync(user)	UpdateAsync
RoleClaims	Email	DeleteAsync(user)	DeleteAsync
UserLogins	PasswordHash	UpdateAsync(user)	UpdateAsync(user)
UserRoles	PhoneNumber	Users	Roles
UserTokens	Roles	RemoveFromRoleAsync (name, role)	FindByNameAsync

Диаграмма классов



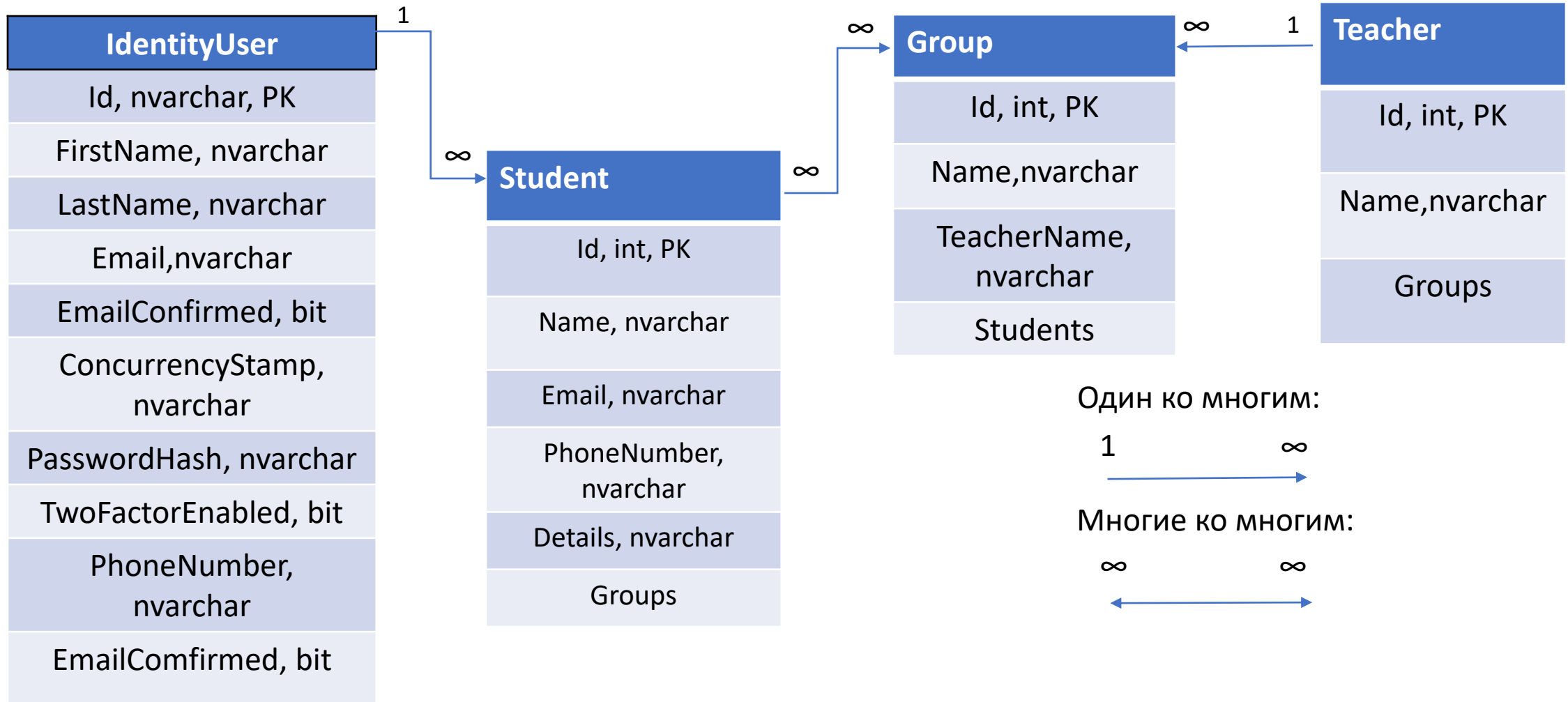
База данных. EntityFramework

Метод code first

Классы:

- **DbContext**: определяет контекст данных, позволяет создавать запросы, отслеживать изменения и сохранять данные.
- **DbSet**: Описывает набор сущностных классов, который затем можно использовать в коде для создания запросов CRUD (create, read, update, delete) к данным. С помощью экземпляров этого класса описываются различные объекты базы данных (таблицы, представления, хранимые процедуры и т.д.)

Схема базы данных



Заключение

В рамках выпускной квалификационной работы были поставлены и выполнены следующие задачи:

- разработать back-end часть приложения с помощью ASP.NET Core MVC;
- разработать front-end часть с помощью HTML-страниц;
- разработать структуру базы данных;
- тестирование функциональности приложения проводилось;

В итоге создан прототип приложения, на основе которого в дальнейшем можно добавлять новый функционал, который облегчит работу сотрудников учреждения.

Спасибо за внимание!