# **Доклад**

**Первый слайд**

Здравствуйте! Я, студент группы П2-19, Ратников Владимир, хочу представить курсовой проект на тему «Разработка приложения для учета сотрудников производственного предприятия».

**Слайд 2: Предметная область**

Кадровик (специалист отдела кадров, менеджер по персоналу или инспектор по кадрам) – это сотрудник компании, который занимается ведением кадров и несет ответственность за формирование кадровой политики предприятия.

Слово «кадровик» широко использовалось и впервые было применено ещё в СССР. Сотрудник отдела кадров заполнял трудовые книги, ввёл личные дела сотрудников и записывал должностные инструкции. Работа была рутинной.

Задача кадровика — оформлять и вносить вправки в документы кадровой службы, проверять на соответствие актуальному законодательству.

**Слайд 3: Инструменты разработки**

На третьем слайде изображены иконки используемых приложений.

При выборе инструментов сравнивались все достоинства и недостатки каждого инструмента, знания и время на обучение.

В качестве языка программирования был выбран C#. Он имеет схожий синтаксис как у Python и C++, а также он объектно-ориентированный, что подходит для разработки графических приложений.

В качестве фреймворка был выбран Blazor Server. Он позволяет создавать серверные приложения с использованием языка программирования C# и компонентов Razor Pages.

В качестве среды разработки была выбрана Microsoft Visual Studio 2022, так как только на ней можно писать приложения на платформе Blazor Server. **Слайд 4: Диаграмма прецедентов**

На четвёртом слайде отображена диаграмма прецедентов, которая описывает работу сотрудника кадров.

Сотрудник отдела кадров при работе с приложением постоянно взаимодействует с базой данных. Через неё он может уволить сотрудника или добавить нового. Также, он может получить данные о уже записанных сотрудниках и отредактировать.

**Слайд 5: Проектирование сценария**

На этом слайде изображен сценарий использования программы.

Сотрудник отдела кадров запускает программу, открывает браузер и вводит в адресную строку адрес хоста. Обычно это <https://localhost:5001>.

Первоначально открывается страница со всеми, кроме удаленных, сотрудниками. Их можно редактировать, а также удалять.

При нажатии на Deleted, или же при добавлении в ссылку /counter, можно переключиться на страницу с удаленными пользователями. Точно также можно переключиться на страницу с отделами, нажав на Departments или добавив в адрес страницы /alldepartments.

Завершение программы происходит после закрытия EXE файла.

**Слайд 6: Диаграмма классов**

На этом слайде изображена диаграмма классов.

Класс «Base User» является базовым и хранит в себе только Id и Status Id. Он нужен чтобы в дальнейшем наследовать от него другие классы

Класс «Department» является наследником класса «Base User» и содержит поле DepartmentName

Класс «User» также является наследником класса «Base User», и содержит в себе поля Fio, Email, NumberPhone, Department, Department Position и DepartmentNumber. Department — одноименный объект класса Department.

Класс «UserServices» содержит методы класса «User».

**Слайд 7: Главная страница**

На седьмом слайде отображена главная страница

На ней сотрудник может добавить нового сотрудника, вводя данные в пустые поля, и просматривать список уже существующих сотрудников.

Данные сотрудников можно редактировать. Их можно удалять, перенося на страницу с удаленными сотрудниками.

Для перехода на следующие страницы служит левая панель.

**Слайд 8: Страница с удаленными пользователями**

На восьмом слайде отображена страница с удаленными сотрудниками.

Здесь мы можем редактировать или восстановить их.

**Слайд 9: Страница с отделами**

На девятом слайде отображена страница с отделами.

На этой странице, отображены все отделы. Также можно занести в базу данных новый отдел. Существующие отделы отредактировать или удалить.

**Заключительный слайд**

На этом мой доклад закончен. Спасибо за внимание!