Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

Отчет по производственной практике

По ПМ.01 Разработка модулей программного обеспечения  
для компьютерных систем

НАТКиГ.720200.010.000

Разработал:

Студент группы ПР-21.101

Пахомов И. Е.

**Содержание**

[Введение 3](#_Toc170672766)

[1 Описание предметной области 4](#_Toc170672767)

[2 Проектирование информационной системы 5](#_Toc170672768)

[2.1 Постановка задачи 5](#_Toc170672769)

[2.2 Разработка базы данных 6](#_Toc170672770)

[2.3 Макет приложения 7](#_Toc170672771)

[3 Разработка приложения 9](#_Toc170672772)

[3.1 Описание используемых плагинов и библиотек 9](#_Toc170672773)

[3.2 Описание разработанных процедур и функций 10](#_Toc170672774)

[4 Тестирование 11](#_Toc170672775)

[4.1 Выбор стратегии тестирования 11](#_Toc170672776)

[4.2 Протокол тестирования 11](#_Toc170672777)

[5 Заключение 12](#_Toc170672778)

[6 Библиография 13](#_Toc170672779)

# Введение

Проект разработки системы учета уголовных и личных дел для МВД России направлен на создание удобного и функционального веб-приложения, которое упростит и оптимизирует процесс управления различными видами дел.

В настоящее время существующие методы учета могут быть недостаточно эффективными и не всегда позволяют оперативно получать полную информацию о состоянии дел. Цель данного проекта — устранение этих недостатков путем разработки специализированной базы данных и удобного веб-приложения для учета дел.

Задачами проекта в связи с указанной целью являются:

* изучение предметной области;
* определение ключевых функций приложения и проектирование интерфейса пользователя;
* разработка и реализация дизайна;
* написание кода приложения;
* тестирование полученного продукта.

Реализуемый продукт представляет собой веб-приложение для учета уголовных и личных дел.

# Описание предметной области

Министерство внутренних дел Российской Федерации занимается контролем и управлением большим количеством различных дел, включая уголовные, административные, гражданские и трудовые дела, а также личные дела сотрудников. Эти задачи требуют надежного и эффективного программного обеспечения для учета и обработки информации.

Министерство уделяет большое внимание качеству и надежности управления делами. Для повышения эффективности работы сотрудников МВД разрабатывается специализированная система, которая упростит и оптимизирует процесс учета и управления делами. Система будет включать все необходимые функции для работы сотрудников, обеспечивая удобный доступ к информации и высокую точность учета.

Разрабатываемая система предоставит интуитивно понятный интерфейс, упрощающий поиск и обработку данных. Благодаря автоматизации рутинных операций, система существенно повысит точность и удобство учета. Надежность и отказоустойчивость программного обеспечения позволит сотрудникам МВД быть уверенными в стабильности и качестве предоставляемых услуг.

В Министерстве внутренних дел существуют важные внутренние задачи, среди которых ключевую роль играет учет и управление делами. Для оптимизации и упрощения этого процесса разрабатывается специализированная база данных и удобная система. База данных будет аккумулировать все сведения о делах, включая количество, состояние и расположение. Это позволит сотрудникам МВД легко отслеживать изменения, вести учет поступлений и списаний, а также формировать необходимые отчеты.

# Проектирование информационной системы

## 2.1 Постановка задачи

Целью разработки базы данных и системы для учета уголовных и личных дел в МВД является упрощение и оптимизация процесса управления делами, что позволит сотрудникам легко находить нужную информацию и эффективно управлять делами.

Задачи:

* анализ требований и проектирование системы;
* разработка функционала для поиска и управления делами;
* обеспечение безопасности данных;
* интеграция и тестирование системы;
* обеспечение поддержки и обновлений.

Необходимый функционал:

* поиск дел по различным параметрам;
* удобный и интуитивно понятный пользовательский интерфейс;
* аутентификация и защита от несанкционированных изменений.

## 2.2 Разработка базы данных

Диаграмма прецедентов представлена на рисунке 1.

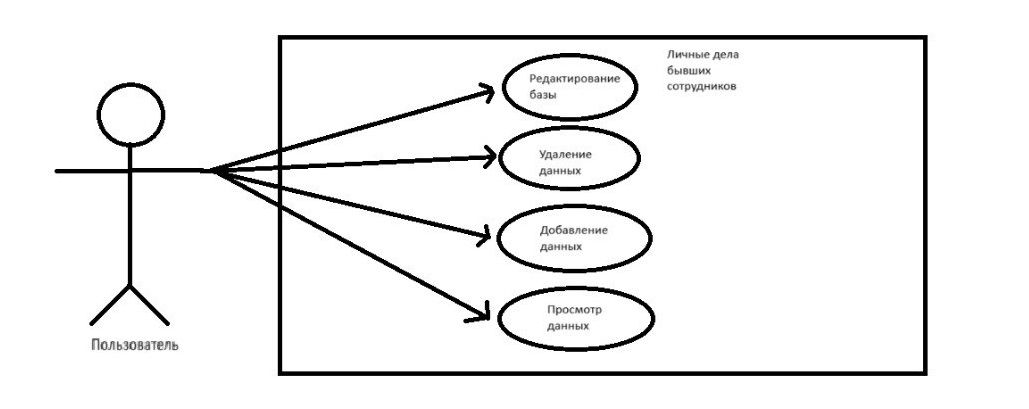


Рисунок 1 – Диаграмма прецедентов

Таблица 1 – Спецификация для прецедента «Редактирование данных».

|  |  |
| --- | --- |
| Раздел | Описание |
| Краткое описание | Пользователь редактирует данные |
| Действующие лица | Пользователь |
| Предусловия | Пользователь имеет доступ к базе данных |
| Основной поток | 1. Пользователь приступает к редактированию данных   А1. Неверно введены данные   1. Пользователь сохраняет данные выходом из столбца или нажатием кнопки Enter 2. Пользователь завершает редактирование данных и прецедент завершается |
| Альтернативный поток | А1. Неверно введены аргументы   1. Система выводит сообщение «Некорректный формат данных». 2. Возвращается на шаг 1. |
| Постусловие | Результатом будут успешно отредактированные данные |

Таблица 2 – Спецификация для прецедента «Удаление данных».

|  |  |
| --- | --- |
| Раздел | Описание |
| Краткое описание | Пользователь удаляет данные |
| Действующие лица | Пользователь |
| Предусловия | Пользователь имеет доступ к базе данных |
| Основной поток | 1 Пользователь приступает к работе с базой  2 Пользователь удаляет данные и прецедент завершается |
| Постусловие | Результатом будет успешное удаление данных |

Таблица 3 – Спецификация для прецедента «Добавление данных».

|  |  |
| --- | --- |
| Раздел | Описание |
| Краткое описание | Пользователь добавляет данные |
| Действующие лица | Пользователь |
| Предусловия | Пользователь имеет доступ к базе данных |
| Основной поток | 1. Пользователь приступает к добавлению данных   А1. Неверно введены данные   1. Пользователь добавляет данные нажатием кнопки для добавления данных 2. Пользователь завершает добавление данных данных и прецедент завершается |
| Альтернативный поток | А1. Неверно введены аргументы   1. Система выводит сообщение «Некорректный формат данных». 2. Возвращается на шаг 1. |
| Постусловие | Результатом будет успешное добавление данных |

В этом разделе рассматривается процесс создания специализированной базы данных для учета уголовных и личных дел в МВД России. Основное внимание уделяется оптимизации внутренних процессов, включая учет и отслеживание состояния дел, что позволит сотрудникам более эффективно управлять информацией.

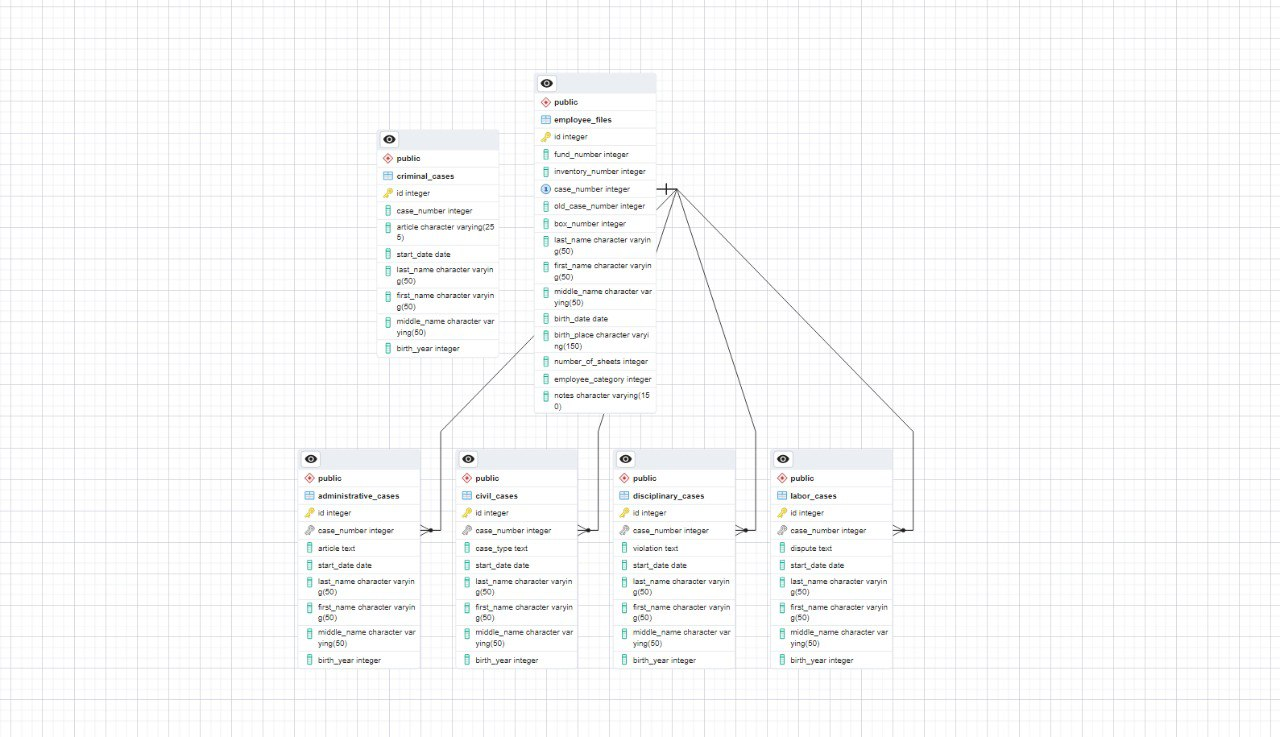
Диаграмма представлена на рисунке 2.

Рисунок 2 – Диаграмма базы данных

## 2.3 Макет приложения

На рисунке 3 представлен экран добавления уголовных дел.

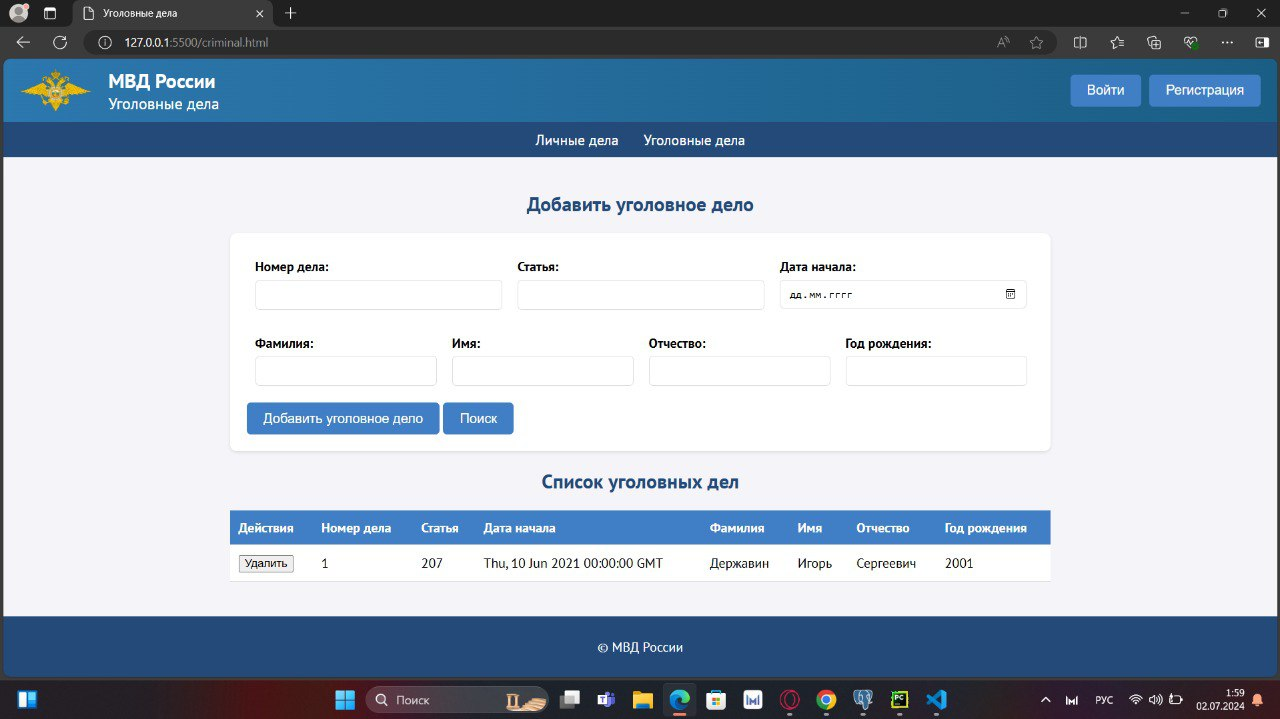


Рисунок 3 – Экран добавления уголовных дел

На рисунке 4 представлен экран добавления записи.

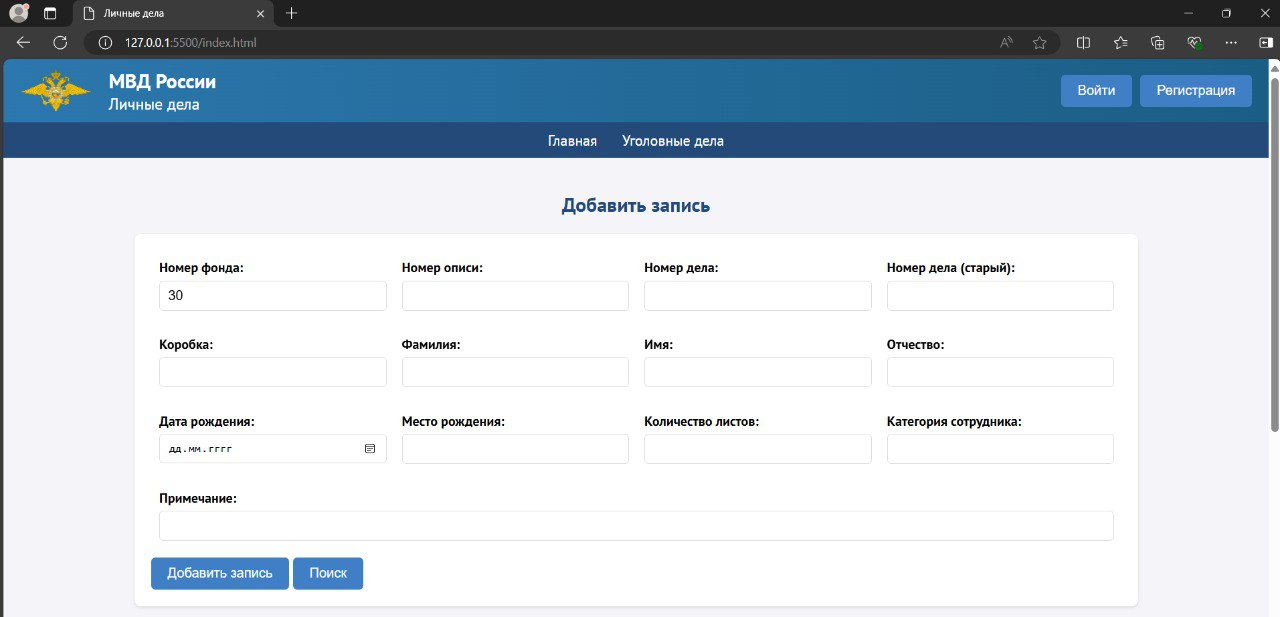


Рисунок 4 – Экран добавления записи

На рисунке 5 представлен экран списка дел бывших сотрудников.

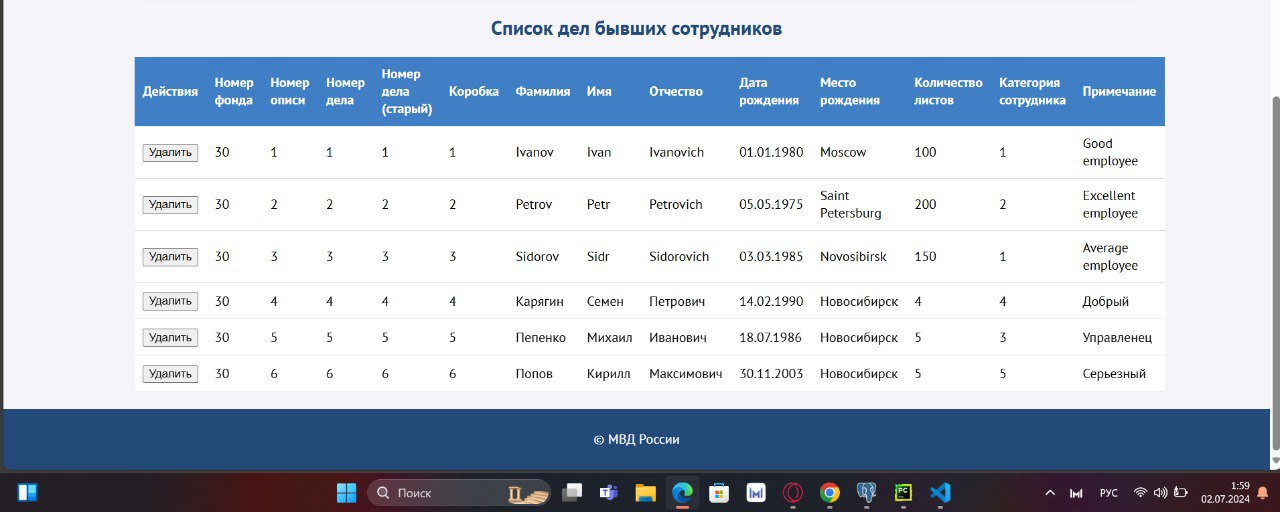


Рисунок 5 – Экран списка дел бывших сотрудников

# Разработка приложения

## 3.1 Описание используемых плагинов и библиотек

В проекте используются библиотеки с различными компонентами и функциями. Список всех библиотек, а также их описание представлен в таблице 2.

Таблица 2 – Библиотеки и их описание

|  |  |
| --- | --- |
| Библиотека | Описание |
| flask\_cors | Библиотека для создания веб-приложений на Python |
| psycopg2 | Модуль, который используется для работы с PsSQL |

Данные библиотеки являются неотъемлемой частью приложения, обеспечивая его правильное функционирование и реализацию всех задуманных возможностей. Без этих библиотек приложение не сможет работать в полной мере, так как они предоставляют необходимые инструменты и функции, которые необходимы для его работы.

## 3.2 Описание разработанных процедур и функций

В приложении реализованы следующие методы (таблица 3):

Таблица 3 – Методы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| document.addEventListener('DOMContentLoaded',  function() {...}) | Этот метод запускает выполнение скрипта после полной загрузки HTML-документа. Он вызывает функцию fetchRecords() для получения списка записей и добавляет обработчик событий для формы добавления записи. |
| fetchRecords() | Этот метод отправляет GET-запрос на сервер для получения всех записей из базы данных. После получения данных создает строки таблицы для каждой записи и добавляет их в HTML-документ. |
| document.getElementById('addRecordForm').  addEventListener('submit', function(event) {...}) | Этот метод предотвращает стандартное поведение формы при отправке (перезагрузку страницы), собирает данные формы и отправляет POST-запрос на сервер для добавления новой записи. После успешного добавления записи обновляет список записей и сбрасывает форму. |
| deleteRecord(id) | Этот метод отправляет DELETE-запрос на сервер для удаления записи по указанному идентификатору. После успешного удаления обновляет список записей. |
| updateRecord(id, field, value) | Этот метод отправляет POST-запрос на сервер для обновления конкретного поля записи по указанному идентификатору. Запрос включает в себя новое значение поля в формате JSON. После успешного обновления данных обновляет список записей. |

# Тестирование

## 4.1 Выбор стратегии тестирования

* модульное тестирование;
* функциональное тестирование;
* тестирование пользовательского интерфейса;

Для тестирования нашего приложения было решено использовать комбинированную стратегию, включающую модульное, функциональное тестирование, а также UI тестирование.

## 4.2 Протокол тестирования

Необходимо проверить функционал приложения. Для этого, для каждой функции были разработаны TestCase.

В таблице 4 предоставлено тестирование функции добавление записи.

Таблица 4 – Тестирование функции авторизации

|  |  |  |
| --- | --- | --- |
| Название | Добавление записи в систему | |
| Функция | Добавление записи в систему | |
| Предусловие | Открыть приложение и перейти на форму добавления записи | |
| Действие | Ожидаемый результат | Результат теста |
| Шаги теста(positive)  1. Ввести корректные данные | Новая запись добавлена и отображается в списке | пройден |
| Шаги теста(negative)  1. Оставить обязательные поля пустыми | Появляется сообщение об ошибке и запись не добавлена | пройден |

Тестирование удобства использования. Была произведена проверка юзабилити:

* приложение должно быть простым в использовании;
* экран просмотра дел бывших сотрудников должен быть построен в логической последовательности.

# Заключение

В процессе разработки системы учета уголовных и личных дел для МВД России были учтены предпочтения и потребности пользователей. Система обеспечивает удобный доступ к информации о делах и предоставляет сотрудникам возможности для эффективного управления данными.

Интерфейс системы разработан с акцентом на удобство использования и минимизацию количества действий, необходимых для достижения целей пользователей. Основные экраны и функции спроектированы так, чтобы каждый элемент был функциональным и интуитивно понятным.

Важным этапом разработки стало тестирование, которое подтвердило корректность работы и успешное функционирование всех компонентов системы. Это гарантирует, что разработанная система учета дел соответствует всем поставленным целям и требованиям.

В результате разработки системы учета дел для МВД России был создан функциональный и удобный инструмент, полностью учитывающий потребности пользователей. Проделанная работа успешно достигла поставленных целей и является надежной основой для дальнейшего развития и улучшения проекта.

# Библиография

Электронные ресурсы:

1. GitHub API [Электронный ресурс]. – Документация по API GitHub. – URL: <https://docs.github.com/en/rest> (дата обращения: 01.06.2024)
2. Retrofit [Электронный ресурс]. – Библиотека для сетевых запросов в Android. – URL: https://square.github.io/retrofit/ (дата обращения: 01.06.2024)
3. Kotlin Coroutines [Электронный ресурс]. – Документация по Kotlin Coroutines. – URL: https://kotlinlang.org/docs/coroutines-overview.html (дата обращения: 01.06.2024)