

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования**

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___» _____ 2024 г.

**Разработка компьютерной настольной справочной системы
для федеральных и мировых судей по судебным решениям**

ВЫПУСНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.04.2024.308-331.ВКР

Научный руководитель,
доцент кафедры СП, к.т.н.

_____ М.В. Сухов

Автор работы,
студент группы КЭ-404

_____ М.А. Братенев

Ученый секретарь
(нормоконтролер)

_____ И.Д. Володченко

«___» _____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП

Л.Б. Соколинский
29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра
студенту группы КЭ-404
Братеневу Максиму Алексеевичу,
обучающемуся по направлению
09.03.04 «Программная инженерия»

- 1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)
Разработка компьютерной настольной справочной системы для федеральных и мировых судей по судебным решениям.
- 2. Срок сдачи студентом законченной работы:** 03.06.2024 г.
- 3. Исходные данные к работе**
 - 3.1. Основы баз данных. [Электронный ресурс] URL:
<https://mipt.ru/dnbic/content/db.pdf> (дата обращения: 01.04.2024 г.).
 - 3.2. Самоучитель MS Access. [Электронный ресурс] URL:
<https://accesshelp.ru/samouchitel-ms-access/> (дата обращения: 01.04.2024 г.).
- 4. Перечень подлежащих разработке вопросов**
 - 4.1. Провести анализ предметной области.
 - 4.2. Провести анализ методов разработки и архитектуру справочных систем.
 - 4.3. Реализовать и протестировать справочную систему для поиска судебных решений.
- 5. Дата выдачи задания:** 29.01.2024 г.

Научный руководитель,
доцент кафедры СП, к.т.н.

М.В. Сухов

Задание принял к исполнению

М.А. Братенев

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Предметная область проекта	7
1.2. Анализ аналогичных проектов	8
2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ.....	10
2.1. Функциональные требования к проектируемой системе	10
2.2. Нефункциональные требования к проектируемой системе	11
2.3. Диаграмма вариантов использования.....	13
2.4. Краткое описание вариантов использования.....	14
3. АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ	16
3.1. Общее описание архитектурны системы.....	16
3.2. Описание компонентов, составляющих систему	18
3.3. Модель базы данных.....	20
3.4. Реализация архитектуры системы.....	22
4. РЕАЛИЗАЦИЯ СИСТЕМЫ	26
4.1. Реализация компонентов системы	26
4.2. Реализация интерфейса системы.....	29
5. ТЕСТИРОВАНИЕ СИСТЕМЫ.....	32
ЗАКЛЮЧЕНИЕ	37
ЛИТЕРАТУРА.....	38
ПРИЛОЖЕНИЯ.....	40
Приложение А. Спецификация вариантов использования.....	40
Приложение Б. Скрипт создания физической структуры базы данных ИСС	44

ВВЕДЕНИЕ

Актуальность

Существующие на сегодняшний день справочно-правовые системы (СПС), пройдя определенную государственную регистрацию и сертификацию, становятся вполне доступными для потенциальных пользователей (юридических или физических лиц), последние имеют возможность, обращаясь к содержимому СПС, устранять существующие пробелы в области знаний правового характера. В середине XX в. во всех развитых странах начался стремительный рост правовой информации. Число законодательных и иных нормативных правовых актов достигает сотен тысяч, что приводит к возникновению информационного кризиса. Изначально считалось, что СПС создают лишь дополнительные удобства при работе с информацией, а без их использования вполне можно обойтись. Однако, когда начались масштабные пересмотр и перестройка всего российского законодательства, уследить за огромным потоком вновь принятых нормативных актов стало невозможно. Поэтому к середине 1994 г. число потенциальных покупателей пакетов юридических программ значительно возросло.

Традиционные методы получения информации, например, использование периодических печатных изданий или других печатных источников, не были способны разрешить задачу по правовому обеспечению всех заинтересованных лиц. Компьютер же представлял собой замечательное средство аккумуляции, хранения и выдачи правовой информации. Сегодня большинству профессиональных юристов важна возможность получения максимально полной справочной, информационной и правовой информации. Ускорившийся ритм бизнеса и новые условия хозяйствования потребовали создания компьютерных правовых справочных систем, представляющих собой правовые базы данных [1].

Уникальность программных технологий и схем сервисного обслуживания российских СПС состоит в том, что в них почти не используются готовые технологические решения и не повторяется чей-либо зарубежный

опыт. Это связано с такими особенностями России, как огромная территория и относительно слабое развитие телекоммуникаций наряду с необходимостью оперативного предоставления пользователю доступа к большому числу новых документов, поскольку современное российское законодательство быстро развивается и пока еще далеко от стабильности. Отметим, что жесткая конкуренция отечественных компаний и активное развитие СПС на протяжении почти десяти лет обеспечили эффективность информационного сервиса и вывели его на мировой уровень услуг в этой области. Можно выделить следующие основные причины интенсивного развития в России компьютерных технологий, направленных на правовую информатизацию.

1. Стремительное совершенствование и удешевление персональных компьютеров в последнее десятилетие. На смену ЭВМ, которые десять лет назад были доступны лишь экспертам крупных научных центров, пришли компьютеры, ставшие по-настоящему персональными не только для сотен тысяч специалистов, но и для миллионов обычных людей.

2. Реформы политической и экономической жизни в стране породили буквально вал нормативных и иных правовых документов. Только за 1992–2000 гг. федеральными и региональными органами власти и управления было принято около 2 млн актов, при этом число актов, принимаемых ежегодно, не имеет тенденции к уменьшению. Работать с таким массивом документов без современных компьютерных технологий просто невозможно.

Сегодня существует множество практических задач, направлений, связанных с обращением к правовой информации, эффективно решать которые можно лишь при использовании компьютерных СПС (например, работа юристов, экономистов или бухгалтеров) [2].

Постановка задачи

Основной целью данной работы является разработка компьютерной настольной справочной системы для федеральных и мировых судей по судебным решениям.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выполнить анализ предметной области;
- 2) спроектировать и реализовать функционал базы данных;
- 3) спроектировать и реализовать функционал интерфейса;
- 4) провести тестирование разработанной системы.

Структура и содержание работы

Работа состоит из введения, пяти глав, заключения, литературы. Объем работы составляет 46 страниц, объем библиографии – 15 наименований.

В первой главе «Анализ предметной области» описывается принцип работы компьютерных справочных систем и анализ существующих решений.

Вторая глава «Анализ требований к программной системе» посвящена пунктам, которые надо было реализовать для корректной работы программного продукта.

Третья глава «Архитектура программной системы» описывает саму архитектуру системы и взаимодействие всех компонентов в ней между друг другом.

Четвертая глава «Реализация системы» раскрывает разработку программного продукта и его реализацию.

В пятой главе «Тестирование системы» описаны пройденные тесты.

В приложении А содержится диаграмма вариантов использования.

В приложении Б содержится листинг скрипта создания физической структуры базы данных ИСС.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Предметная область проекта

Целью данной работы является разработка компьютерной справочной системы для федеральных и мировых судей по судебным решениям. Компьютерная СПС, она же справочная правовая система – это программный комплекс, включающий в себя массив правовой информации и программные инструменты, позволяющие специалисту работать с этим массивом информации: производить поиск конкретных документов или их фрагментов, формировать подборки документов и т.д.

Разрабатываемая СПС является программой, позволяющей пользователю взаимодействовать с базой данных посредством интерфейса и производить поиск по заданным параметрам.

СПС работает с данными, полученными из ресурса «ГАС Правосудие» [3]. Данные на этом ресурсе хранятся хаотично и бессистемно, имеют затруднительную и не универсальную систему поиска, в которой сложно ориентироваться. Например, на рисунке 1 приведена форма поиска в одной из подсистем картотеки арбитражных дел «Электронное правосудие».

ЭЛЕКТРОННОЕ ПРАВОСУДИЕ

КАРТОТЕКА СТРАЖ БАНК РЕШЕНИЙ КАЛЕНДАРЬ

Фильтр дел Как это работает? А Административные Г Гражданские Б Банкротные С Судебные поручения

Участник дела
название, ИНН или ОГРН Любой +

Судья
фамилия судьи +

Суд
название суда +

Номер дела
например, А50-5568/08 +

Дата регистрации дела
с дд.мм.гггг по дд.мм.гггг

Найти
Сбросить все
Дел в картотеке
0 2 7 7 2 6 1 0 6
PRAVO TECH

Укажите атрибуты дела для поиска

Электронный страж Банк решений арбитражных судов Календарь заседаний

Рисунок 1 – Форма картотеки арбитражных дел «Электронное правосудие»

На рисунке 2 показана общая форма поиска дел в системе «Государственная автоматизированная система Российской Федерации «Правосудие»».

The screenshot shows the search interface of the State Automated System of the Russian Federation 'Justice'. The URL at the top is `sudrf.ru/index.php?id=300&var=true#sp`. The page title is 'Государственная автоматизированная система Российской Федерации «Правосудие»'. The main heading is 'Информация о деятельности судебных участков мировых судей'. Below the heading, there are three links: 'Найти судебный участок мирового судьи', 'Поиск дела' (highlighted in a red box), and 'Поиск по территориальной подсудности'. A paragraph explains that the search form provides information on cases in progress in a district court, with an example: '2-303/2012', where '2-303' is the district number and '2012' is the registration year. The form includes a dropdown menu for 'Субъект Российской Федерации:' and a text input for 'Номер дела:' with a placeholder 'Номер дела' and an example 'Пример: 2-303/2012'. At the bottom, there are two buttons: 'Найти' (highlighted in a red box) and 'Очистить форму'.

Рисунок 2 – Форма поиска «Государственная автоматизированная система Российской Федерации «Правосудие»»

Как видно из рисунков 1 и 2, система поиска в системе очень неудобная и не вариативная. Такой системой поиска пользоваться неэффективно.

1.2. Анализ аналогичных проектов

На сегодняшний день существуют несколько готовых и зарекомендовавших себя справочно-правовых систем: «Гарант», «КонсультантПлюс», ИПС «Кодекс» [4–6]. Все эти службы используются для предоставления доступа к обширной законодательной и правовой базе, которая есть в общем доступе, позволяют ознакомиться с нормативными и правовыми актами, выводят свод действующих, в том числе утративших силу законов и т.д.

Все существующие справочно-правовые системы реализованы по одному принципу создания парсера из общего реестра и последующей структуризацией для упрощенной работы с базой данных по определенным

фильтрам сортировки. Чаще всего они представляют собой локальную базу данных на самом компьютере пользователя, но также есть версии приложений, которые обновляются в реальном времени по мере поступления новых данных в базу.

Выводы по первой главе

Проведя анализ уже готовых приложений и доступных СПС, был сделан вывод, что решения, которые доступны на рынке на сегодняшний день не удовлетворяют ключевым требованиям целевой аудитории пользования, а именно: функционалу и удобству пользования. Все текущие решения являются собой крайне сложные и неудобные в обращении для рядового пользователя интернет-сервисы, которые в условиях ограничений трафика, не всегда работают, что очень усложняет работу с ними. Немаловажным также является то, что крайне ограниченная возможность использовать поиск по определенным критериям, делает возможным поиск только по узким критериям, что зачастую приводит к тому, что необходимость поиска, в связи с этим, исчерпывает себя.

Проанализировав текущие решения, можно заключить, что они не совсем удовлетворяют текущим потребностям судебных участков и имеют неудобный интерфейс и ограниченный функционал, который заставляет тратить огромное количество времени на поиск нужных судебных решений.

2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ

Исходя из проведенного анализа предметной области, было принято решение самостоятельно реализовать компьютерную справочную систему в виде локальной базы данных на самом компьютере судьи с использованием подходящей для этого СУБД Access [7], которая является файл-серверной (локальной) и языка программирования С# [8] на среде разработки Visual Studio [9]. При этом в СПС необходимо реализовать систему многокритериального поиска информации по судебным решениям.

Для предпроектного анализа предметной области был выбран ресурс «Судебные и нормативные акты РФ» [10]. На основании изучения структуры выдачи информации, ее вида и атрибутов были составлены основные объекты информационной модели проектируемой информационно-справочной системы (ИСС). Вся информация находится в открытом доступе и не подразумевает собой персональные данные граждан.

2.1. Функциональные требования к проектируемой системе

ИСС «Банк судебных решений» должна выполнять следующие функции:

- авторизацию пользователя;
- ведение вспомогательного справочника типов документов (постановление / решение / определение / приговор и т.д.);
- ведение вспомогательного справочника регионов;
- ведение вспомогательного справочника категорий дел;
- ведение вспомогательного справочника судебных участков;
- ведение вспомогательного справочника – реестра судей;
- просмотр автонаполняемого справочника истцов и ответчиков;
- ведение вспомогательного справочника видов дел (уголовное / гражданское / административное и т.д.);

- ведение вспомогательного справочника статей закона, являющихся предметом спора в делах;
- регистрация новых документов с возможным указанием истца и ответчика;
- поиск судебных решений и дел по судебному участку;
- поиск судебных решений и дел по судье;
- поиск судебных решений и дел по категории;
- поиск судебных решений и дел по региону;
- поиск судебных решений и дел по предмету спора;
- поиск судебных решений и дел по истцу / ответчику;
- поиск судебных решений и дел по номеру дела;
- поиск и просмотр документов по базе данных системы.

2.2. Нефункциональные требования к проектируемой системе

Важной частью нефункциональных требований являются требования к надежности. Требования к надежности системы должны определять такие свойства системы, как ее зрелость, устойчивость и восстанавливаемость. Надежность системы должна обеспечиваться за счет:

- применения технических средств, системного и базового ПО, соответствующих классу решаемых задач;
- соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;
- обучения пользователей и обслуживающего персонала;
- стабильной работы сервера СУБД.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы,

при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановку программных средств. Время восстановления после отказа, вызванного неисправностью на сервере, не должно превышать времени, требуемого на устранение неисправностей.

Должно быть соблюдено разграничение прав доступа к функциональности ИСС. СУБД системы должна быть защищена паролем администратора. Доступ к структурам таблиц базы данных должен иметь только уполномоченный системный администратор.

Взаимодействие пользователей с ИСС должно осуществляться посредством визуального пользовательского графического интерфейса. Интерфейс ИСС должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Ввод-вывод данных системы, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс ИСС должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы. Интерфейс информационной системы должен максимально соответствовать основным принципам Нильсена-Молиха [11] проектирования пользовательского интерфейса системы.

Требования к персональному рабочему месту пользователя должны определяться системными требованиями, достаточными для запуска и нормальной стабильной работы операционной системы Microsoft Windows версии 10.

В качестве СУБД применить MS Access версии 2007 или выше.

В качестве прикладного ПО, программа должна быть реализована на языке программирования C# .NET Windows Forms / WPF [12].

2.3. Диаграмма вариантов использования

На основании описанных в пункте 2.1 функциональных требований разработана диаграмма вариантов использования (прецедентов) [13] ИСС «Банк судебных решений». Данная диаграмма приведена на рисунке 3.

Диаграмма вариантов использования системы:

- показывает модель предметной области системы;
- отражает состав пользователей системы и разграничение прав доступа между ними;
- визуально выражает пользовательские требования к системе.

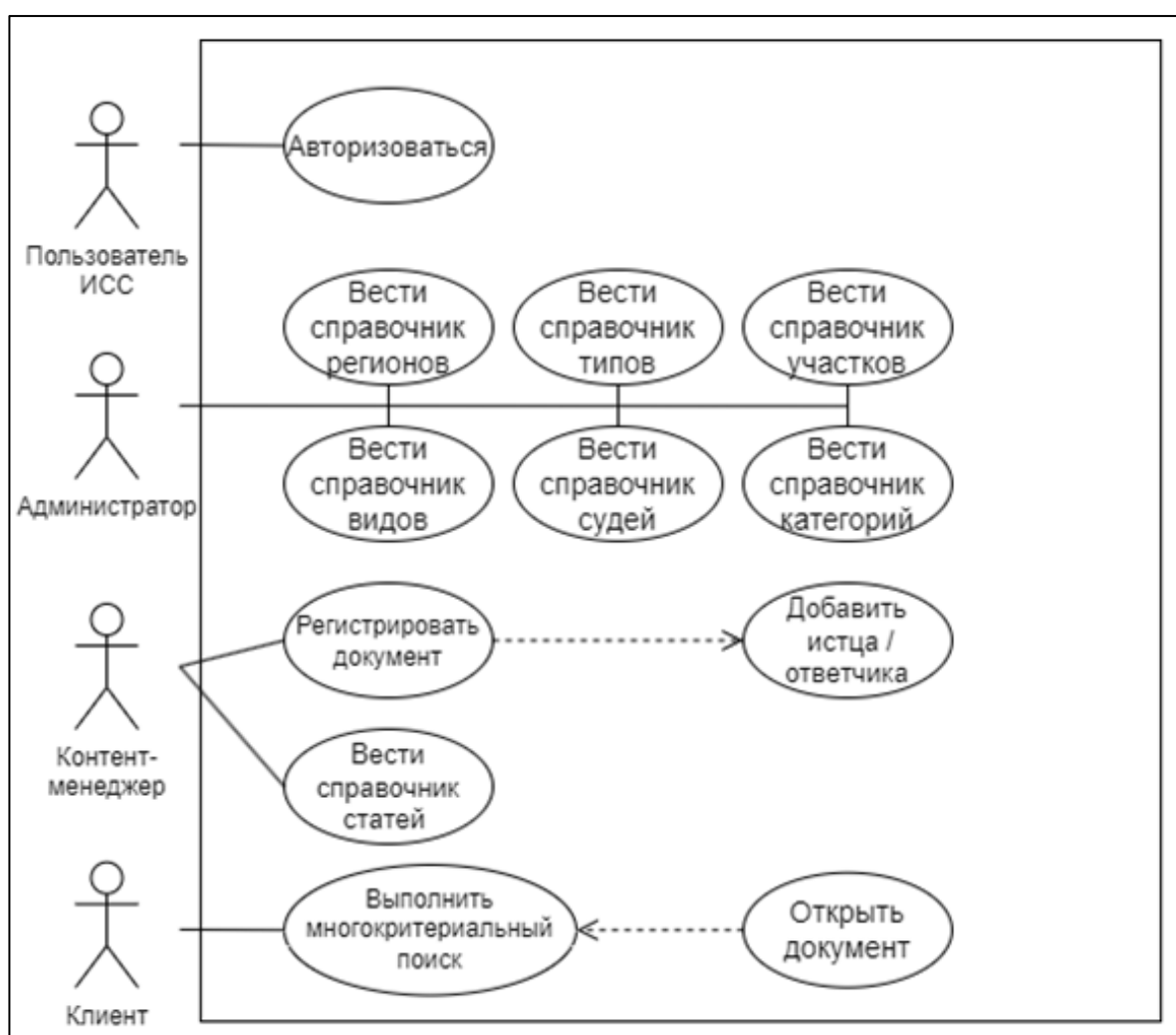


Рисунок 3 – Диаграмма прецедентов ИСС «Банк судебных решений»

Основные актеры, взаимодействующие с системой

Актерами ИСС «Банк судебных решений» являются:

- администратор (ID = A), выполняющий административные функции по наполнению условно-постоянных справочников системы (тех справочников, информация в которых меняется редко);
- контент-менеджер (ID = CM), который выполняет функцию наполнения справочной системы оперативной информацией: новые документы судебных решений, статьи закона, отраженные как предмет спора в судебных делах;
- клиенты (ID = CL), функцией которых является поиск и получение справочной информации по судебным решениям в соответствии с заданным многокритериальным запросом.

Все перечисленные пользователи имеют один общий тип – актера «Пользователь ИИС» (ID = UR), основной вариант использования которого – авторизация в системе.

2.4. Краткое описание вариантов использования

Перечень вариантов использования ИСС:

- 1) авторизоваться – позволяет пользователю авторизоваться в системе и получить соответствующие права доступа;
- 2) вести справочник судебных участков – позволяет заполнить данными справочник судебных участков;
- 3) вести справочник регионов – позволяет заполнить данными справочник регионов, на которых размещены судебные участки;
- 4) вести справочник видов дел – позволяет заполнить данными справочник видов дел (уголовное / гражданское / административное и т.д.);
- 5) вести справочник категорий дел – позволяет заполнить данными справочник категорий, на которые подразделяется множество рассматриваемых судебных дел;

6) добавить истца / ответчика – позволяет для каждого дела актуализировать информацию об истце и ответчике;

7) вести справочник судей – позволяет заполнить данными справочник судей, ведущих дела;

8) вести справочник типов документов – позволяет заполнить данными справочник типов документов (постановление / решение / определение / приговор и т.д.);

9) вести справочник истцов / ответчиков – позволяет просматривать справочник истцов и ответчиков, а также выполнять поиск по ним;

10) регистрировать документ – позволяет внести в справочную систему данные нового судебного решения, включающие: тип документа, вид дела, ведущего судью, регион и судебный участок, а также основное содержание документа (основные разделы документа – текстовая часть – прикрепляется как отдельный файл формата *.rtf);

11) вести справочник статей закона – позволяет заполнить данными справочник статей закона, являющихся предметом спора в делах;

12) выполнить многокритериальный поиск документов – позволяет найти и выдать подборку документов в соответствии с запросом по критерию: номер дела, статья закона, категория, истец / ответчик, дата, судья, судебный участок, регион, любая комбинация приведенных критериев;

13) открыть документ – позволяет открыть содержимое найденного документа из базы данных ИСС.

Выводы по второй главе

Проведя анализ требований к справочной правовой системе, определены и выявлены функциональные и нефункциональные требования к приложению. Это способствовало составлению диаграммы использования, которая отображает как пользователи могут взаимодействовать с системой.

3. АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ

3.1. Общее описание архитектурны системы

Архитектура решения проектируется с помощью инструментальных средств CASE-системы Enterprise Architecture версии 15.0 [14].

В рамках архитектурного проекта ИСС «Банк судебных решений» разработаны модели: логической структуры, компонентов, базы данных, последовательности при выполнении основной системной операции поиска документов.

Архитектура ИСС построена по модульному принципу с использованием шаблона проектирования MVC.

Модель данных доведена до третьей нормальной формы включительно в контексте принятых ограничений бизнес-логики приложения.

ИСС «Банк судебных решений» является информационной и не предполагает никаких функций управления. Целесообразно архитектурно выделить классы, описывающие:

- модель системы (модель информации);
- пользовательский интерфейс.

Информационная модель включает в себя структурированные классы, позволяющие описать сущности предметной области. Главные из них:

- документ (Document);
- судья (Judge);
- судебный участок (Court);
- person – обобщенный класс для частных реализации участников судебных процессов (судья, истец, ответчик).

Также в системе имеются вспомогательные сущности (тип дела, вид решения, регион, должность судьи, статья закона). Все эти вспомогательные сущности тривиальны и описываются одним классом SimpleDictionary, имеющим один атрибут – наименование.

Анализ предметной области позволил установить связи классов информационной модели, определить атрибуты и поведение сущностей, что станет полями и методами классов соответственно.

Теперь необходимо определить ключевые формы для реализации интерфейса программной системы. Поскольку всего типов пользователей будет равно трем, то и пользовательских интерфейсов должно быть ровно столько же.

Пользовательский интерфейс имеет основную форму – MainForm, подтипами которой являются формы конкретной авторизованной пользовательской роли: ContentManagerForm, AdministratorForm, ClientForm. Каждая из форм имеет набор операций в соответствии с полномочиями пользовательской роли:

- ContentManagerForm: добавление документов, работа со справочником статей;
- AdministratorForm: работа со справочниками регионов, судебных участков, типов документов, видов решений, судей;
- ClientForm: поиск и просмотр документов, настройка фильтров поиска документов.

Помимо этих, в системе еще предусматривается еще одна форма – форма документа (DocumentForm), позволяющая добавлять и просматривать документы.

На рисунке 4 приведена диаграмма классов ИСС «Банк судебных решений», разработанная в соответствии с описанной выше концепцией. Диаграмма классов является основной моделью программной системы, отражающей ее логическую структуру и описывающей взаимодействие компонентов программы между собой.

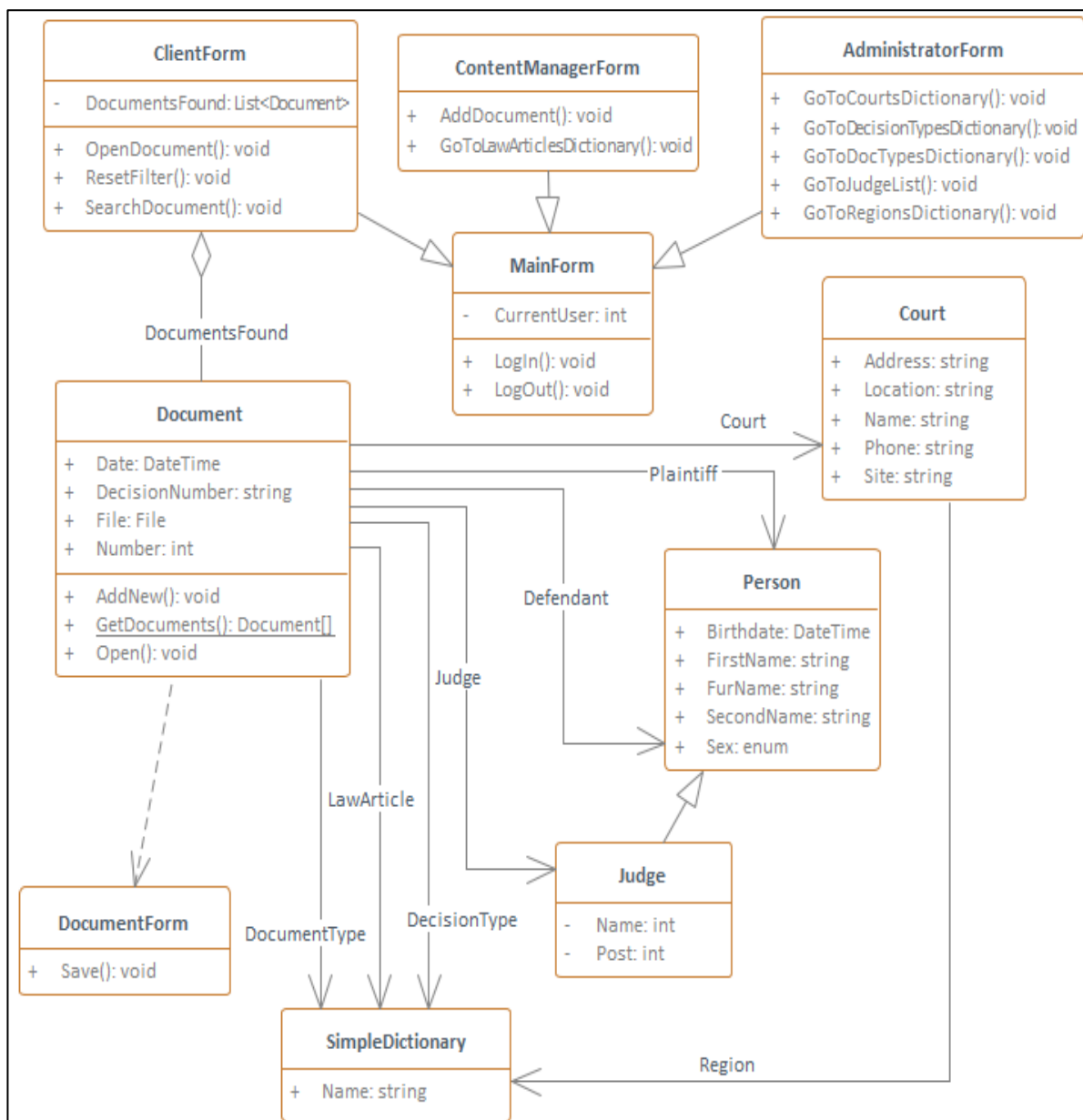


Рисунок 4 – Диаграмма классов ИСС «Банк судебных решений»

3.2. Описание компонентов, составляющих систему

Архитектура ИСС «Банк судебных решений» построена по модульному принципу, что позволяет получить следующие преимущества:

- простоту понимания структуры системы;
- упрощение поддержки исходного кода;
- независимость отдельных компонентов друг от друга;
- обеспечение устойчивости к модификациям;
- расширяемость и масштабируемость.

Все программные и другие модули системы распределены по различным пакетам, которые приведены ниже.

1. Model – пакет представления контекста модели данных, основанной на объектной структуре системы.

2. View – пакет представления системы (слой пользовательского интерфейса), в который включены компоненты форм пользовательского интерфейса также.

3. Controller – пакет контроллера, управляющего поведением модели, содержит контроллер базы данных – DBController, инкапсулирующий все операции управления данными, и контекст данных модели – DataContext, через который происходит взаимодействие с источниками данных БД. Совокупное использование контроллера базы данных и контекста данных позволяет пользовательскому приложению работать с базой данных через спроектированный интерфейс с учетом контекста решаемых задач. Таким образом, в системе реализуется пользовательская работа с моделью данных. В контроллере также имеется компонент ExtendedFilter, в котором реализуются возможности многокритериального фильтра для поиска документов по запросам пользователей в банке данных системы.

Отдельным компонентом системы является файл базы данных – decisions.mdb. База данных в данном проекте выполнена на основе файл-серверной СУБД MS Access 2007.

Таким образом, ИСС «Банк судебных решений» построена на основе архитектурного шаблона MVC (Model-View-Controller). Данный шаблон проектирования позволяет грамотно структурировать программный код, что благотворно влияет на эффективность работы разрабатываемого приложения.

На рисунке 5 приведена архитектурная диаграмма пакетов / компонентов ИСС «Банк судебных решений».

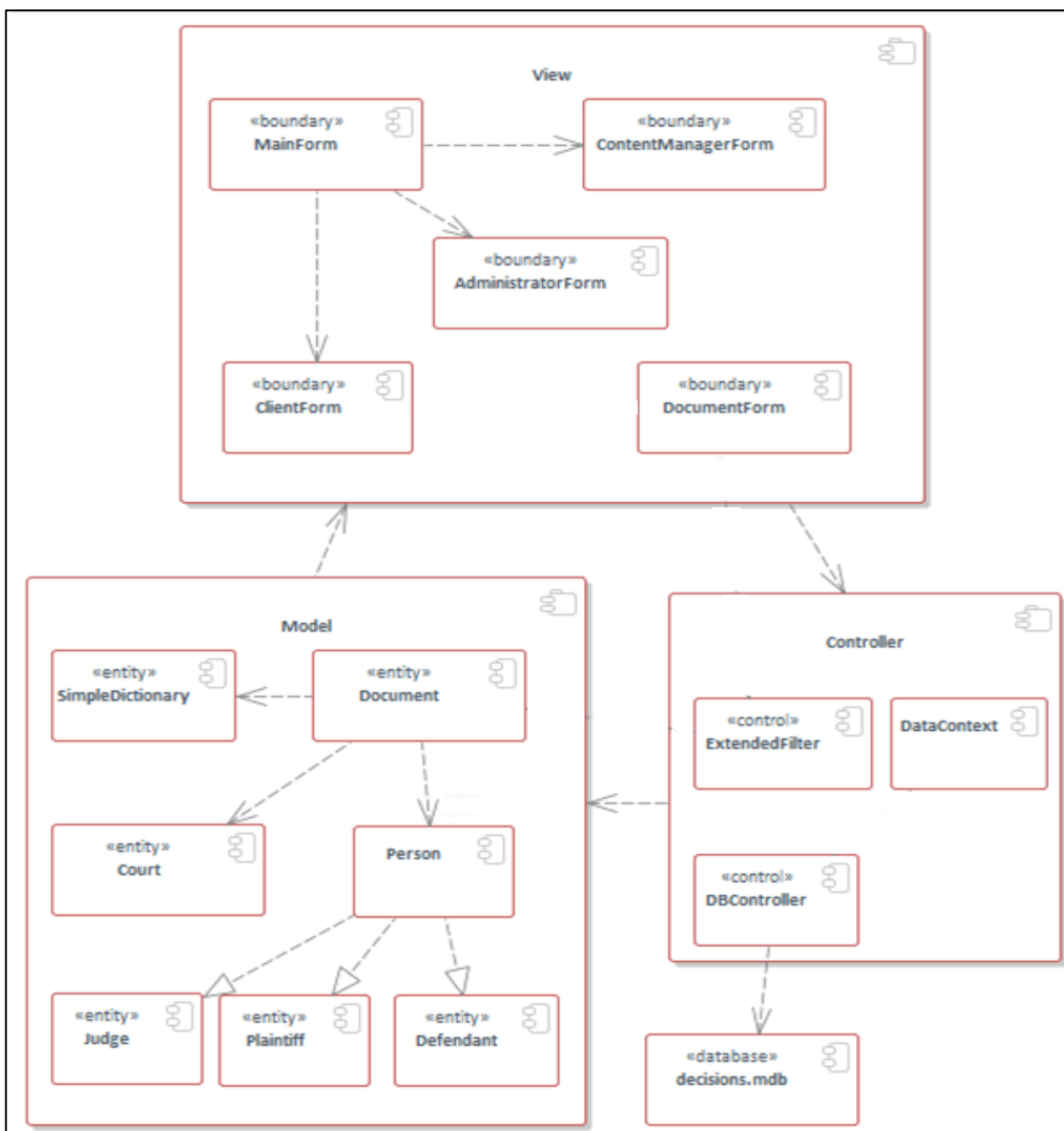


Рисунок 5 – Диаграмма компонентов ИСС «Банк судебных решений»

3.3. Модель базы данных

База данных проекта ИСС «Банк судебных решений» разработана средствами автоматизации инжиниринга данных CASE-системы Enterprise Architect 15.0. В данной системе спроектированы все таблицы – оперативные и справочные, назначены атрибуты полям и основные поведенческие свойства, например, ограничения целостности данных в соответствии с информационной моделью. Так, на рисунке 6 показана схема базы данных системы.

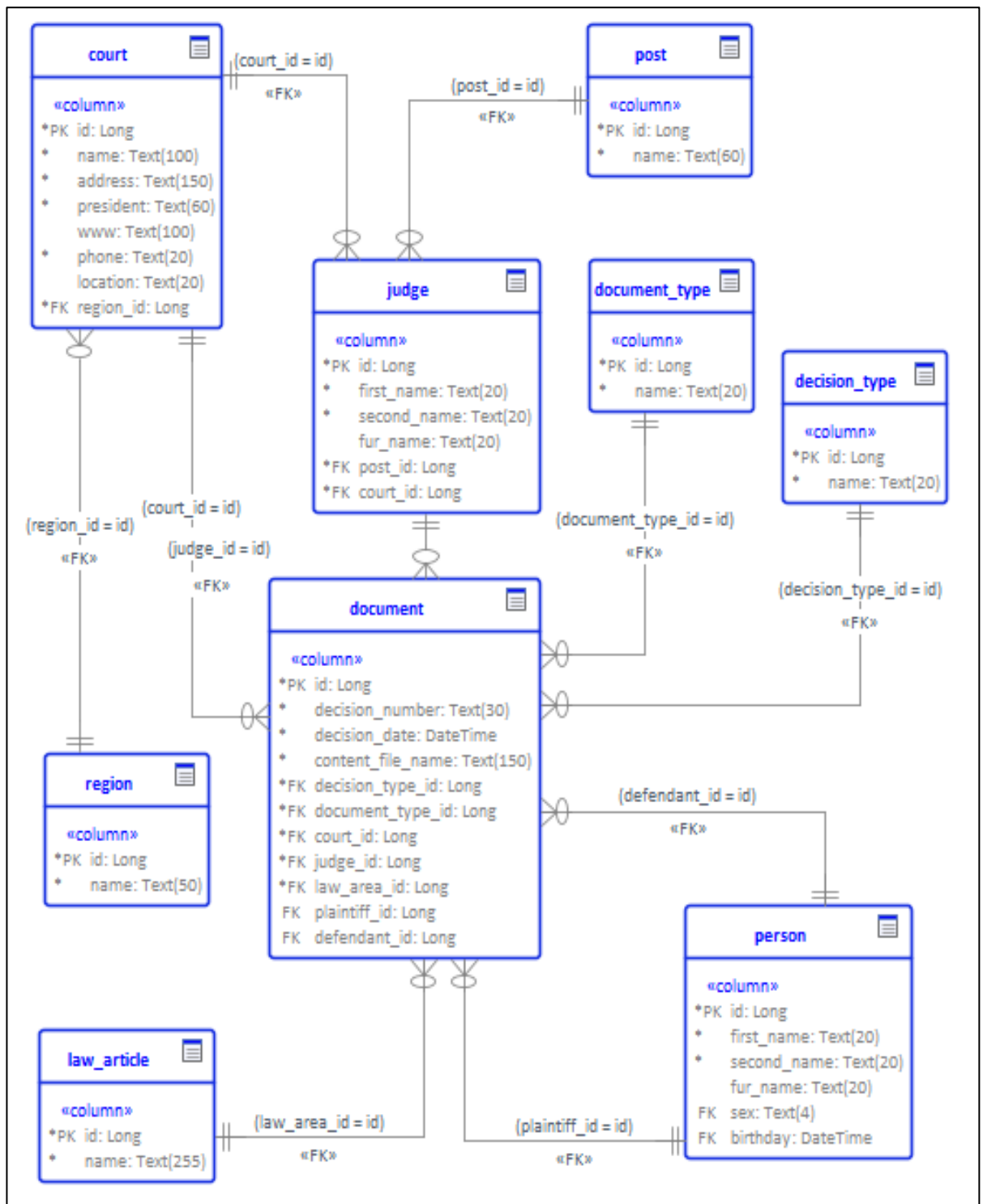


Рисунок 6 – Диаграмма базы данных ИСС «Банк судебных решений»

Разработанная модель данных приведена к третьей нормальной форме включительно в контексте принятых ограничений бизнес-логики приложения. CASE-система Enterprise Architect 15.0 позволяет автоматически сгенерировать скрипт на языке описания данных (DDL, Data Description

Language), позволяющий создать физическую структуру данных в выбранной СУБД. На рисунке 7 приведена форма CASE-системы, в которой настраивается набор таблиц, для которых необходимо сгенерировать скрипт формирования физической структуры БД.

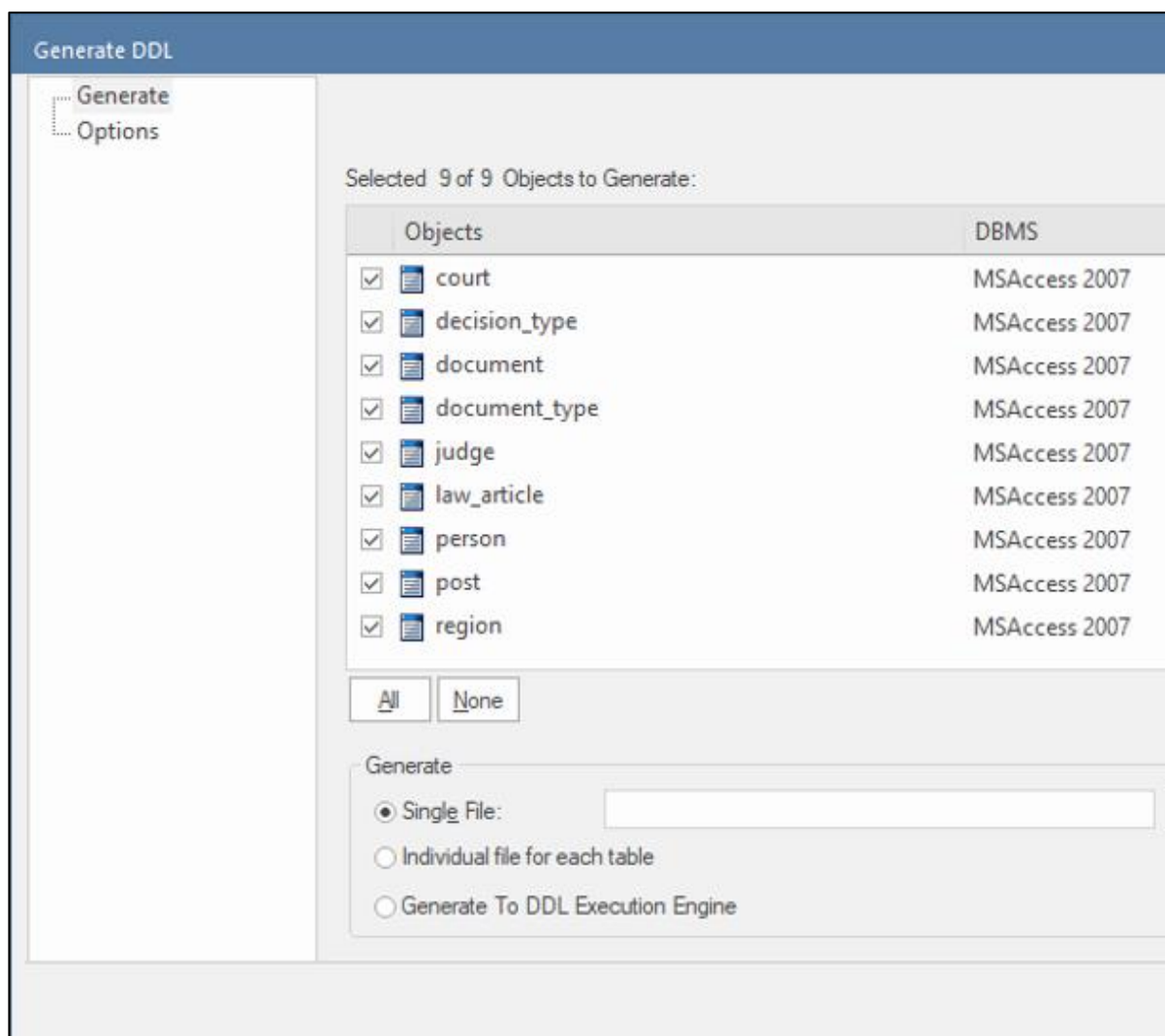


Рисунок 7 – Выбор объектов базы данных ИСС «Банк судебных решений» для формирования скрипта создания физической структуры базы данных

3.4. Реализация архитектуры системы

Основной системной операцией ИСС «Банк судебных решений» является возможность клиента осуществлять многокритериальный поиск необходимых документов по разным фильтрам, таким как номер дела, адрес участка, фамилия судьи, истца/ответчика или даже по типу дела и судебного решения. С точки зрения интерактивного взаимодействия между объектами

эта операция может иметь следующую последовательность обмена сообщениями / вызовами процедур:

- 1) клиент устанавливает фильтры поиска (судья, номер дела, участок, тип дела, вид решения, истец / ответчик, категория дела, регион и т.д.) с помощью инструментов, предложенных формой интерфейса пользователя;
- 2) клиент, взаимодействуя с формой, инициирует команду, выполняющую процедуру запуска поиска;
- 3) форма клиента обращается к статическому методу класса документа с намерением послать в БД запрос поиска документов в соответствии с заданными значениями фильтра;
- 4) статический метод класса документа обращается к контроллеру БД;
- 5) контроллер БД формирует соответствующий SELECT-запрос и отправляет его в БД;
- 6) контроллер БД с помощью драйвера подключения к БД получает выборку данных в формате DataTable;
- 7) полученная выборка проходит процедуру парсинга, в результате чего формируется массив объектов «Документ», удовлетворяющих запросу пользователя, этот массив передается на клиентскую форму для отображения пользователю;
- 8) пользователь выбирает любой документ из найденных и выполняет на форме команду открытия документа;
- 9) форма снова обращается к классу документа с соответствующим методом открытия документа;
- 10) создается новая форма документа;
- 11) экземпляр новой созданной формы документа инициализирован и представлен пользователю на экране.

Описанная последовательность приведена на рисунке 8 на диаграмме последовательности при выполнении основной системной операции поиска документов в ИСС «Банк судебных решений».

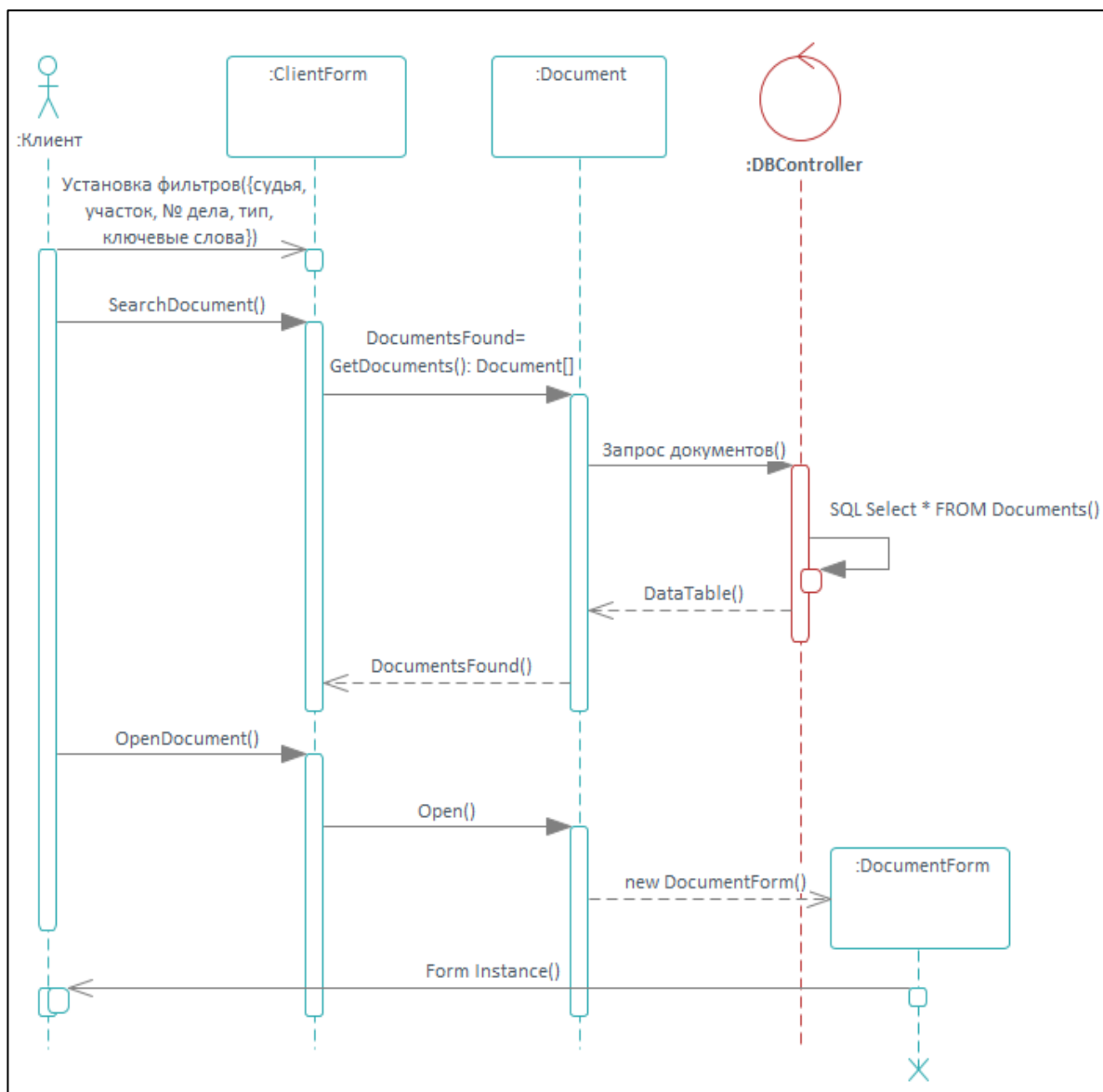


Рисунок 8 – Диаграмма последовательности при выполнении основной системной операции поиска документов в ИСС «Банк судебных решений»

Рассмотренная системная операция является основной в контексте функций системы. Все остальные функции сводятся к управлению справочными данными и выполняются средствами, предоставляемыми .NET: BindingSource, BindingNavigator, DataSource. Все эти средства сами отвечают и за слой представления и за работу с контекстом модели.

Выводы по третьей главе

В рамках этого раздела, были сделаны выводы о том, как должны работать и взаимодействовать программный интерфейс и база данных с предварительно загруженными решениями. Именно представление об их работе друг с другом, дало четкое понимание того, как будет выглядеть конечный продукт. Стоит отметить, что от слаженности работы интерфейса с базой данных и будет зависеть то, выполняет ли разрабатываемая ИСС основным требованиям или нет и будет ли она являться конкурентноспособным решением для судебного аппарата.

Основываясь на определении функциональных и нефункциональных требований, проведен анализ программной системы, в результате которого:

- составлена диаграмма классов и компонентов разрабатываемой системы;
- определено, как будут взаимодействовать компоненты системы друг с другом;
- была определена последовательность выполнения при основной функции поиска документа по базе данных.

Проведя тщательный анализ программной системы и выполнив все поставленные в данном разделе задачи, можно приступать к реализации ИСС.

4. РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1. Реализация компонентов системы

Основные этапы реализации ИСС «Банк судебных решений» заключаются в следующих работах:

- реализация контроллера;
- реализация модели;
- реализация представления (пользовательского интерфейса).

В ИСС «Банк судебных решений» предусматривается два контроллера: контроллер безопасности и контроллер работы с базой данных.

Контроллер безопасности отвечает за хеширование последовательность. В данном случае – хеширование паролей пользователей. В листинге 1 приведен код контроллера безопасности.

Листинг 1 – Код контроллера безопасности

```
using System.Text;
using System.Security.Cryptography;

namespace CourtDecisionsDictionary.Controller.Security
{
    /// <summary>
    /// Класс информационной безопасности
    /// </summary>
    public class Secure
    {
        // Хешировать по SHA256
        public static string Hash(string data)
        {
            string result = "";
            SHA256 hf = SHA256.Create();
            byte[] hash = hf.ComputeHash(Encoding.ASCII.GetBytes(data));
            for (int i = 0; i < hash.Length; i++)
                result += $"{hash[i]:X2}";
            return result;
        }
    }
}
```

Один из случаев применения контроллера безопасности – функция авторизации пользователя. За эту функцию отвечает метод `Login` в контроллере базы данных. Передаваемый в метод в качестве аргумента пароль предварительно хешируется перед запуском запроса на соответствие и поиска

авторизуемого аккаунта. В листинге 2 приведен фрагмент кода использования контроллера безопасности.

Листинг 2 – Код использования контроллера безопасности

```
public static UserRole Login(string login, string password)
{
    try
    {
        // Данные пользователя получены из БД
        foreach (var item in new DB_CDRSDataSetTableAdapters.usersTableAdapter().TryToLogin
            (login, Security.Secure.Hash(password)))
            return (UserRole)item.access_id;

        // Данные пользователя не получены из БД
        return UserRole.NotAuthorized;
    }
    catch (Exception ex)
    {
        LastErrorText = ex.Message;
        return UserRole.Failed;
    }
}
```

Контроллер базы данных отвечает за выполнение запросов к базе данных. Реализуются следующие виды запросов:

- поиск совпадения по логину и паролю пользователя (для процедуры авторизации);
- идентификация сущностей из модели по идентификаторам;
- управление данными в базе документов: регистрация, обновление, удаление, чтение базы данных судебных решений;
- заполнение таблиц данными, полученными путем выполнения SQL-запросов, отправленных в БД через соответствующие адаптеры.

Вся работа в контроллере БД производится с помощью специальных адаптеров, которые создаются автоматически средой Visual Studio при интеграции схемы данных в проект. Для работы с адаптерами в контроллере реализуется соответствующая обвязка, включающая:

- соотнесение данных с моделью;
- обработку полученных данных;
- обработку ошибок.

Один из случаев применения контроллера базы данных – функция многокритериального поиска пользователем документа. За эту функцию отвечает метод `FillTable` в окне пользователя. На основе выбранных пользователем параметров фильтра (элементы `CheckBox` и `TextBox`) составляется SQL-запрос, который через соответствующий адаптер в контроллере базы данных посылается в БД. В качестве ответа получается датасет, который отображается на форме пользователя в таблице как результат поиска. В листинге 3 приведен фрагмент кода использования контроллера базы данных в функции поиска документов.

Листинг 3 – Код использования контроллера базы данных функции поиска документов

```
foreach (var doc in GetDocumentsByUserQuery(
    chkFilterDecisionNumberOn.Checked ? d_f_number : "",
    chkFilterCourtOn.Checked ? d_f_court : "",
    chkFilterJudgeOn.Checked ? d_f_judge : "",
    chkFilterRegionOn.Checked ? d_f_region : "",
    chkFilterLawArticleOn.Checked ? d_f_law_article : ""))
```

Реализация модели представляет собой создание компонентов (модулей) системы в соответствии со схемой базы данных и диаграммой классов.

Объекты, представляющие собой модель системы, создаются в рамках задач реализации поведения в своей предметной области. Также объекты сохраняют / изменяют себя с сохранением результата в базе данных. Так, в граничном классе `NewDecisionPanel`, отвечающем за регистрацию нового документа в функции `documentBindingNavigatorSaveItem_Click` происходит создание объекта – документа для сохранения информации о нем в базе данных. В листинге 4 приведена эта функция.

Листинг 4 – Функция создания документа и регистрации в системе

```
var c_doc = Model.Document.CurrentDocument;

c_doc.DecisionNumber = decision_numberTextBox.Text.Trim();
c_doc.Date = decision_dateDateTimePicker.Value;
c_doc.ContentFileName = content_file_nameTextBox.Text.Trim();
c_doc.DecisionTypeID = Convert.ToInt32(decision_type_idComboBox.SelectedValue);
c_doc.DocumentTypeID = Convert.ToInt32(document_type_idComboBox.SelectedValue);
c_doc.CourtID = Convert.ToInt32(court_idComboBox.SelectedValue);
```

```

c_doc.JudgeID = Convert.ToInt32(judge_idComboBox.SelectedValue);
c_doc.LawArticleID = Convert.ToInt32(law_area_idComboBox.SelectedValue);

bool result_ok = create_new
    ? Controller.DB.AccessDB.SaveNewDocument(c_doc)
    : Controller.DB.AccessDB.SaveEditedDocument(c_doc);

```

4.2. Реализация интерфейса системы

Пользовательский интерфейс ИСС «Банк судебных решений» построен с помощью панелей, выполненных на базе UserControl. Панели «вкладываются» в контейнерный контрол (System.Windows.Forms.Panel) в качестве дочернего элемента. Генеральным контейнером является панель на основной форме системы – MainForm. Состав панелей представления системы приведен на рисунке 9.

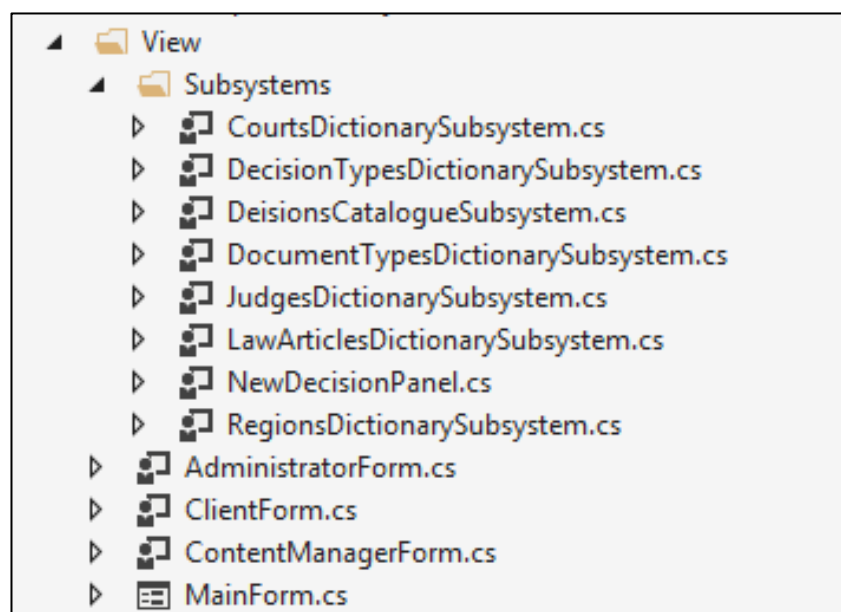


Рисунок 9 – Компоненты представления ИСС «Банк судебных решений»

В разрабатываемой ИСС присутствует возможность переключения между учетными записями. Так, например, можно провести повторную аутентификацию с учетной записи с правами клиента на учетную запись с правами администратора или обратно.

Реализована эта возможность при помощи загрузки соответствующей панели подсистемы в контейнерный объект необходимой пользователю формы.

На рисунке 10 приведен макет пользовательского интерфейса панели администратора ИСС «Банк судебных решений».

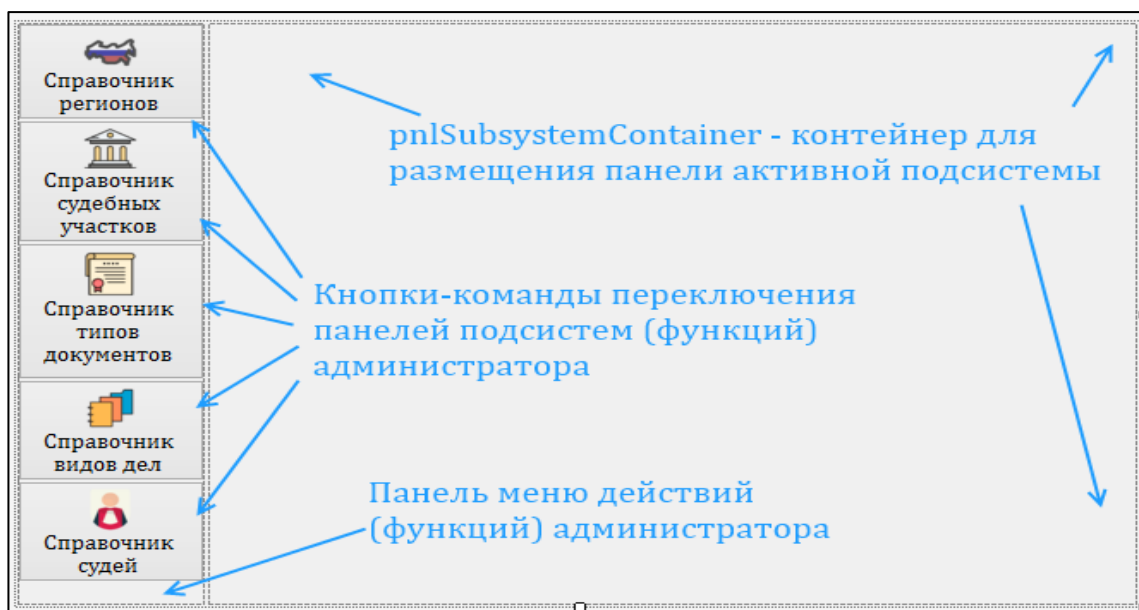


Рисунок 10 – Макет пользовательского интерфейса панели администратора ИСС «Банк судебных решений»

Основной механизм работы системы состоит в двух видах взаимодействия: система – пользователь и система – база данных. Таким образом, система взаимодействует с базой данных и представляет результат пользователю. Для этого в системе предусмотрены специальные компоненты и функции пользовательского интерфейса, предлагаемые средствами .NET Framework от Microsoft.

Так, ведение справочных и оперативных таблиц выполняется с помощью комбинации таких элементов, как DataGridView и ToolStrip. DataGridView отображает полученные данные из соответствующего набора DataSet. ToolStrip предоставляет строку активного контекстного меню подсистемы. DataGridView связан с источником данных (BindingSource), который обращается через набор данных (DataSet) в соответствующую таблицу физической базы данных – для этого используется специальный адаптер (DataTableAdapter).

Все созданные в СУБД таблицы и представления должны быть интегрированы в проект прикладного программного обеспечения системы. Среда разработки прикладного программного обеспечения (MS Visual Studio 2019) позволяет выполнить комплексную интеграцию базы данных в проект ПО. Для этого компанией Microsoft был разработан программный интерфейс ODBC и технология интегрирования БД с приложением. Являясь также продуктом Microsoft, MS Access со своей стороны поддерживает идею комплексной интеграции в IDE Visual Studio посредством специальных коннекторов.

Выводы по четвертой главе

В ходе выполнения этого раздела, было реализовано минимально жизнеспособное приложение, которое уже может выполнять все требующиеся от него задачи и функции.

Поскольку теперь имеется уже готовое приложение, то можно приступать как к отдельным тестам модулей, проверяя их работу независимо друг от друга, так и системы в целом, составив несколько тестовых запросов и проверить правильное выполнение каждой части разрабатываемой ИСС.

Протоколы тестирования вынесены в отдельную главу.

5. ТЕСТИРОВАНИЕ СИСТЕМЫ

В решение ПО ИСС «Банк судебных решений» интегрирован проект модульного тестирования (UnitTest) [15]. Тестироваться будет основная функция системы – клиентская, то есть поиск по базе данных судебных решений.

Клиентский фильтр поиска, изображенный на рисунке 11, имеет 7 критериев поиска: номер дела, фамилия судьи, вид спора (статья), судебный участок, регион (судебного участка), истец, ответчик.

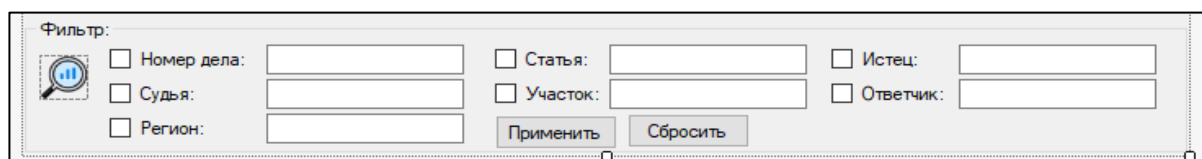


Рисунок 11 – Фильтр клиента

Тестируемый модуль – View.ClientForm.

Метод View.ClientForm, отвечающий за поиск по справочнику судебных решений – GetDocumentsByUserQuery, этот метод и будет тестироваться юнит-тестом. Код данного метода приведен в листинге 5.

Листинг 5 – Код метода View.ClientForm.GetDocumentsByUserQuery

```
public List<Model.Document> GetDocumentsByUserQuery(
    string FilterDocNumber = "", string FilterCourtName = "",
    string FilterJudgeSecondName = "", string FilterRegionName = "",
    string FilterLawArticleName = "", string FilterPlaintiff = "",
    string FilterDefendant = "")
{
    List<Model.Document> result = new List<Model.Document>();
    foreach (var doc in docs)
    {
        if (!string.IsNullOrEmpty(FilterCourtName))
            if (doc.ItsCourt.Name.ToLower().IndexOf(FilterCourtName.ToLower()) < 0)
                continue;
        if (!string.IsNullOrEmpty(FilterDocNumber))
            if (doc.DecisionNumber.ToLower().IndexOf(FilterDocNumber.ToLower()) < 0)
                continue;
        if (!string.IsNullOrEmpty(FilterJudgeSecondName))
            if (doc.ItsJudge.SecondName.ToLower().IndexOf(FilterJudgeSecondName.ToLower()) < 0)
                continue;
        if (!string.IsNullOrEmpty(FilterLawArticleName))
            if (doc.ItsLawArticle.Name.ToLower().IndexOf(FilterLawArticleName.ToLower()) < 0)
                continue;
```



```

        if (!string.IsNullOrEmpty(FilterRegionName))
            if (doc.ItsCourt.ItsRegion.Name.ToLower().IndexOf(FilterRegion-
Name.ToLower()) < 0)
                continue;
            if (!string.IsNullOrEmpty(FilterPlaintiff))
                if (doc.PlaintiffID > 0)
                    if (doc.ItsPlaintiff.SecondName.ToLower().IndexOf(FilterPlain-
tiff.ToLower()) < 0)
                        continue;
                    if (!string.IsNullOrEmpty(FilterDefendant))
                        if (doc.DefendantID > 0)
                            if (doc.ItsDefendant.SecondName.ToLower().IndexOf(FilterDefend-
ant.ToLower()) < 0)
                                continue;
                            result.Add(doc);
                        }
                    return result;
}
}

```

Метод `View.ClientForm`, отвечающий за представление на форме результата по найденным судебным решениям – `FillTable`, этот метод и будет тестироваться практически и визуально. Данный метод представлен в листинге 6.

Листинг 6 – Код метода `View.ClientForm.FillTable`

```

protected void FillTable()
protected void FillTable()
{
    // Filters
    string d_f_number = txtFilterDecisionNumber.Text.Trim().ToLower();
    string d_f_court = txtFilterCourt.Text.Trim().ToLower();
    string d_f_judge = txtFilterJudge.Text.Trim().ToLower();
    string d_f_region = txtFilterRegion.Text.Trim().ToLower();
    string d_f_law_article = txtFilterLawArticle.Text.Trim().ToLower();
    string d_f_plaintiff = txtFilterPlaintiff.Text.Trim().ToLower();
    string d_f_defendant = txtFilterDefendant.Text.Trim().ToLower();
    documentDataGridView.Rows.Clear();
    foreach (var doc in GetDocumentsByUserQuery(
        chkFilterDecisionNumberOn.Checked ? d_f_number : "",
        chkFilterCourtOn.Checked ? d_f_court : "",
        chkFilterJudgeOn.Checked ? d_f_judge : "",
        chkFilterRegionOn.Checked ? d_f_region : "",
        chkFilterLawArticleOn.Checked ? d_f_law_article : "",
        chkFilterPlaintiffOn.Checked ? d_f_plaintiff : "",
        chkFilterDefendantOn.Checked ? d_f_defendant : ""))
    {
        documentDataGridView.Rows.Add(new object[] {
            doc.ID,
            doc.DecisionNumber,    doc.Date,
            doc.LawArticleID,      doc.ContentFileName,
            doc.DecisionTypeID,    doc.DocumentTypeID,
            doc.CourtID,           doc.JudgeID,
            doc.PlaintiffID > 0 ? doc.PlaintiffID : (int?)null,
            doc.DefendantID > 0 ? doc.DefendantID : (int?)null
        });
    }
}

```

```

documentDataGridView.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
documentDataGridView.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;
documentDataGridView.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
documentDataGridView.AutoSizeRows(DataGridViewAutoSizeRowsMode.AllCells);
}

```

Тестовая выборка по поиску вида решений приведена на рисунке 12.

The screenshot shows a web application interface for viewing court cases. At the top, there is a user profile icon and the text "Вы вошли как: клиент". Below this is a header "Просмотр дела". The main part of the interface is a table with the following columns: "Дело №", "Дата", "Предмет спора", "Вид дела", "Вид решения", and "Участок". The table contains 14 rows of data. The row with case number "5-211/2017" is highlighted in blue. Below the table is a search filter panel with the following fields: "Номер дела:", "Судья:", "Регион:", "Статья:", "Участок:", "Истец:", and "Ответчик:". There are "Применить" and "Сбросить" buttons at the bottom of the filter panel.

Дело №	Дата	Предмет спора	Вид дела	Вид решения	Участок
2-473/2017	22.05.2017	Отношения,	Гражданское	Определение	Судебный участок № 10
2-1605/2017	03.07.2017	Прочие исковые	Гражданское	Определение	Судебный участок № 2
2-535/2017	10.04.2017	Споры, связанные с	Гражданское	Решение	Белогорский районный
2-1114/2017	18.05.2017	Прочие исковые	Гражданское	Определение	Судебный участок № 6
2-122/2017	30.03.2017	Споры, связанные с	Гражданское	Решение	Судебный участок Верх
5-221/2017	30.05.2017	ст. 12.26 ч. 1	Административное	Постановление	Белогорский районный
5-211/2017	25.05.2017	ст. 20.19 КоАП РФ	Административное	Постановление	Белогорский районный
3-450/2017	03.07.2017	ч.2 ст.12.2 КоАП РФ	Административное	Постановление	Судебный участок №10
3-451/2017	03.07.2017	ч.2 ст.12.2 КоАП РФ	Административное	Постановление	Судебный участок №10
5-459/2017	12.07.2017	КоАП РФ ст. 20.25	Административное	Постановление	Судебный участок № 5
5-571/2017	27.06.2017	ст. 7.5 КоАП РФ	Административное	Постановление	Светловский судебный
5-299/2017	17.04.2017	КоАП РФ ст. 20.25	Административное	Постановление	Судебный участок № 1
5-377/2017	26.05.2017	КоАП РФ ст. 20.25	Административное	Постановление	Судебный участок № 1

Рисунок 12 – Тестовая выборка судебных решений

Было рассмотрено два тестовых случая. Первый заключался в поиске по региону и судебному участку, а второй по множеству номеров судебных решений, судье и предмету спора.

Результаты тестирования приведены на рисунках 13 и 14.

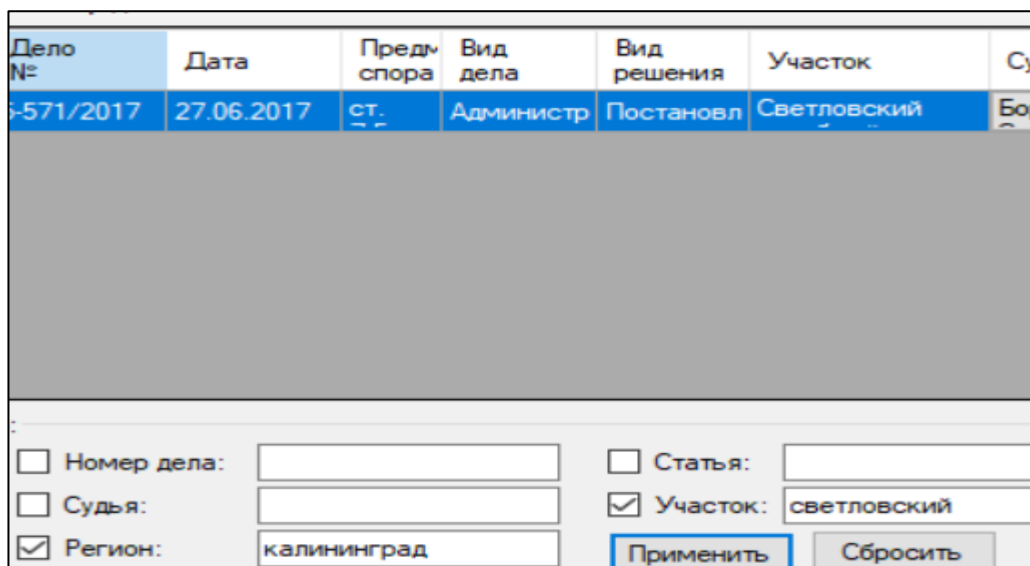


Рисунок 13 – Результат функционального тестирования тестового случая 1

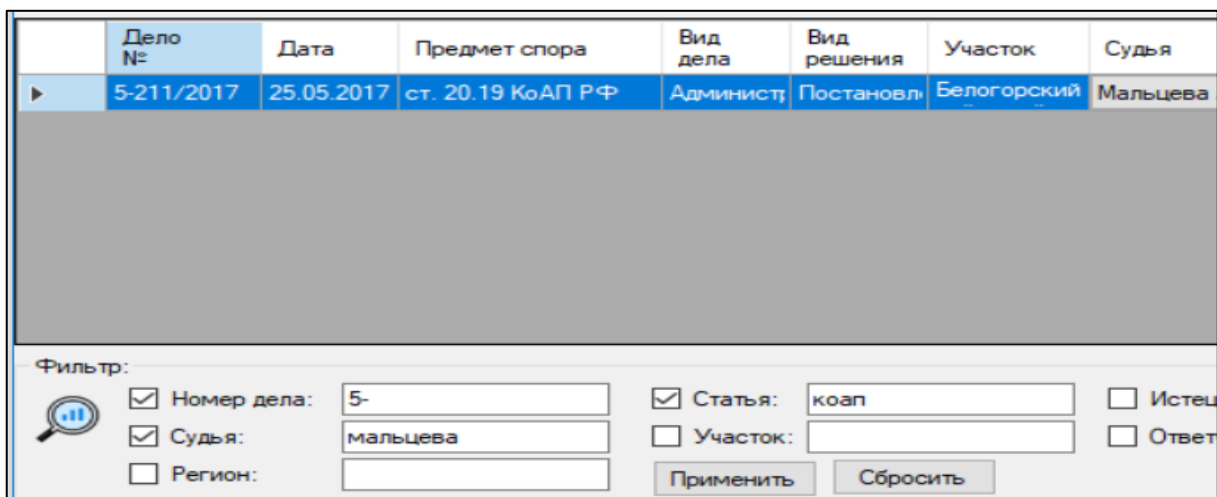


Рисунок 14 – Результат функционального тестирования тестового случая 2

Ниже представлен исходный код в листинге 7 для тестовых случаев.

Листинг 7 – Исходный код модульных тестов для тестовых случаев

```

namespace UnitTestProject
{
    [TestClass]
    public class UnitTest
    {
        [TestMethod]
        public void TestSearch_Case_1()
        {
            string filter_region = "калининград";
            string filter_court = "Светловский";
            string answer = "5-571/2017";
            CourtDecisionsDictionary.View.ClientForm cf =
            new CourtDecisionsDictionary.View.ClientForm();
            List<CourtDecisionsDictionary.Model.Document> docs =

```

```

        cf.GetDocumentsByUserQuery("", filter_court, "", filter_region,
        "");
        string response = docs.Count > 0 ? docs[0].DecisionNumber :
string.Empty;
        Assert.AreEqual(answer, response);
        Assert.AreEqual(1, docs.Count);
    }
    [TestMethod]
    public void TestSearch_Case_2()
    {
        // Test values
        string filter_number = "5-";
        string filter_judge_sname = "Мальцева";
        string filter_article = "коап";
        // True answer (expected)
        string answer = "5-211/2017";
        // Create object and test method
        CourtDecisionsDictionary.View.ClientForm cf =
            new CourtDecisionsDictionary.View.ClientForm();
        List<CourtDecisionsDictionary.Model.Document> docs =
            cf.GetDocumentsByUserQuery(filter_number, "", fil-
ter_judge_sname, "", filter_article);
        string response = docs.Count > 0 ? docs[0].DecisionNumber :
string.Empty;
        // .. - номер найденного судебного решения
        Assert.AreEqual(answer, response);
        Assert.AreEqual(1, docs.Count);
    }
}
}
}

```

Результаты выполнения автоматических модульных тестов приведены на рисунке 15.

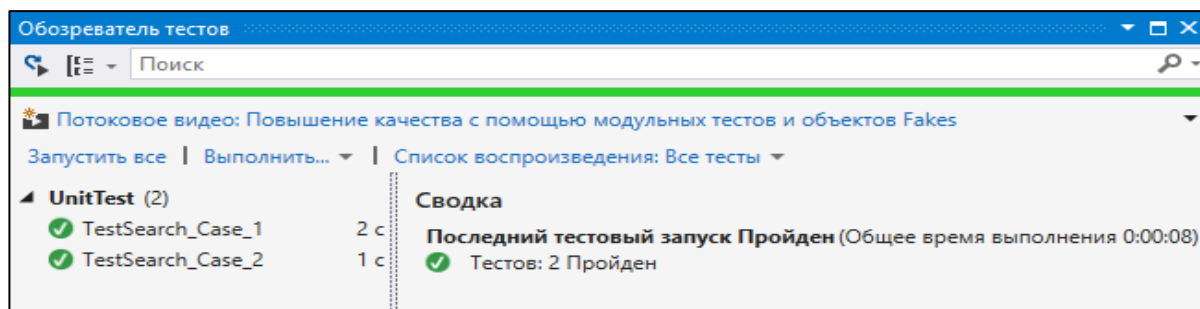


Рисунок 15 – Результаты выполнения автоматических модульных тестов

Выводы по пятой главе

В пятой главе были проведены базовые тесты, чтобы проверить работоспособность приложения. Результаты тестирования ИСС «Банк судебных решений» доказали корректность и полноту реализации системой заявленных функций.

ЗАКЛЮЧЕНИЕ

Целью данной работы являлась разработка компьютерной настольной справочной системы для федеральных и мировых судей по судебным решениям.

В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

- 1) выполнен анализ предметной области;
- 2) спроектирован и реализован функционал базы данных;
- 3) спроектирован и реализован функционал интерфейса программы;
- 4) проведено тестирование разработанной системы.

В рамках выполнения выпускной квалификационной работы была реализована база данных для поиска по решениям федеральных и мировых судей, а также реализован интерфейс к ней.

Данная справочная система позволит упростить поиск по уже вынесенным судебным решениям для судей и судейского аппарата, в условиях ограниченного доступа к сети интернет.

В результате выполнения выпускной квалификационной работы были решены все поставленные задачи, таким образом, цель данной работы достигнута.

ЛИТЕРАТУРА

1. Прохоров А.Е. Справочные правовые системы // Компьютер-Пресс 2002. – № 39. – 14 с.
2. Развитие справочных систем в юриспруденции. [Электронный ресурс] URL: http://prepod-shmu.ucoz.ru/lekcii/lekcija_15_po_iptd.pdf (дата обращения: 09.04.2024 г.).
3. ГАС «Правосудие». [Электронный ресурс] URL: <https://sudrf.ru/> (дата обращения 09.04.2024 г.).
4. Официальный сайт Гарант. [Электронный ресурс] URL: <https://www.garant.ru/> (дата обращения: 09.04.2024 г.).
5. Официальный сайт КонсультантПлюс. [Электронный ресурс] URL: <http://www.consultant.ru/> (дата обращения: 09.04.2024 г.).
6. Официальный сайт Кодекс. [Электронный ресурс] URL: <https://kodeks.ru/> (дата обращения: 09.04.2024 г.).
7. Документация Access. [Электронный ресурс] URL: <https://coderlessons.com/tutorials/microsoft-technologies/izuchite-microsoft-access/ms-access-kratkoe-rukovodstvo> (дата обращения: 09.04.2024 г.).
8. Документация C#. [Электронный ресурс] URL: <https://metanit.com/c/tutorial/> (дата обращения: 09.04.2024 г.).
9. Документация VisualStudio. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/cpp/?view=msvc-170> (дата обращения: 09.04.2024 г.).
10. Судебные и нормативные акты РФ. [Электронный ресурс] URL: <https://sudact.ru/magistrate/> (дата обращения: 09.04.2024 г.).
11. Общие принципы проектирования взаимодействия Якоба Нильсена. [Электронный ресурс] URL: <https://vc.ru/design/243284-10-obshchih-principov-proektirovaniya-vzaimodeystviya-yakoba-nilsena> (дата обращения: 09.04.2024 г.).

12. Руководство по программированию в Windows Forms. [Электронный ресурс] URL: <https://metanit.com/sharp/windowsforms/> (дата обращения: 09.04.2024 г.).

13. Использование диаграммы вариантов использования. [Электронный ресурс] URL: <https://habr.com/ru/articles/566218/> (дата обращения: 09.04.2024 г.).

14. Применение технологии Enterprise Arcitect. [Электронный ресурс] URL: <https://habr.com/ru/companies/lanit/articles/352826/> (дата обращения: 09.04.2024 г.).

15. Руководство по применению модульного тестирования. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/visualstudio/test/unit-test-basics?view=vs-2022> (дата обращения: 09.04.2024 г.).

ПРИЛОЖЕНИЯ

Приложение А. Спецификация вариантов использования

В таблицах 1–5 приведена детальная спецификация основных вариантов использования.

Таблица 1 – Спецификация варианта использования «Авторизоваться»

Прецедент: Авторизоваться
ID: 1
Краткое описание: позволяет пользователю авторизоваться в системе и получить соответствующие права доступа
Главные актеры: Пользователь
Второстепенные актеры: Нет
Предусловия: система запущена, приглашение к авторизации
Основной поток: 1. Пользователь вводит логин и пароль 2. ИИС проверяет логин и пароль по базе 3. ИИС загружает подсистему авторизованного пользователя (альтернативный поток I)
Постусловия: загружена подсистема авторизованного пользователя
Альтернативные потоки: I. Логин / пароль неверны: 1. Система выдает соответствующее сообщение 2. Переход к шагу 1 основного потока.

Таблица 2 – Спецификация варианта использования «Вести справочник судебных участков»

Прецедент: Вести справочник регионов
ID: 2
Краткое описание: позволяет заполнить данными справочник регионов
Главные актеры: Администратор
Второстепенные актеры: Нет
Предусловия: Администратор авторизован, загружена подсистема администратора
<p>Основной поток:</p> <ol style="list-style-type: none"> 1. ИИС выводит системное меню команд управления данными справочника 2. Администратор выполняет команду «добавить запись» (альтернативный поток I, II) 3. ИИС выводит форму ввода данных (поле: «Наименование») 4. ИИС выводит системное меню команд управления данными справочника 6. Администратор выполняет команду «добавить запись» (альтернативный поток I, II) 7. ИИС выводит форму ввода данных (поле: «Наименование») 8. Администратор заполняет поля ввода и выполняет команду «Подтвердить» 9. ИИС сохраняет запись / изменение записи в БД 10. ИИС обновляет вид справочника в соответствии с изменениями
Постусловия: внесены изменения в справочник (запись добавлена, отредактирована или удалена)
<p>Альтернативные потоки:</p> <p>I. Выполнена команда «редактировать данные»:</p> <ol style="list-style-type: none"> 1. ИИС загружает актуальные значения редактируемой сущности из справочника для отображения на форме редактирования 2. Переход к шагу 3 основного потока. <p>II. Выполнена команда «удалить данные»: ИИС запрашивает у пользователя подтверждение удаления записи</p> <ol style="list-style-type: none"> 1. Администратор дает подтверждение – запись удаляется из БД (альтернативный поток III) 2. Переход к шагу 6 основного потока. <p>III. Подтверждение на удаление записи не дано:</p> <p>Переход к шагу 1 основного потока.</p>

Таблица 3 – Спецификация варианта использования «Регистрировать документ»

Прецедент: Регистрировать документ
ID: 3
Краткое описание: позволяет внести в справочную систему данные нового судебного решения
Главные актеры: Контент-менеджер
Второстепенные актеры: Нет
Предусловия: Контент-менеджер авторизован, загружена подсистема контент-менеджера
<p>Основной поток:</p> <ol style="list-style-type: none"> 1. Контент-менеджер выполняет команду «регистрировать новый документ» 2. ИИС выводит форму ввода данных (поля-списки, синхронизированные с соответствующими справочниками: «Тип документа», «Вид дела», «Ведущий судья», «Регион», «Судебный участок», «Суть спора (статья)», другие поля: «Номер дела», «Дата вынесения», «Файл документа» – загружается указанный файл формата «*.RTF» с диска). 3. Контент-менеджер заполняет поля ввода и выполняет команду «Подтвердить» 4. ИИС сохраняет запись / изменение записи в БД (альтернативный поток I) 5. ИИС регистрирует в БД новый документ
Постусловия: новый документ зарегистрирован в базе данных документов ИИС
<p>Альтернативные потоки:</p> <p>I: Не все поля заполнены:</p> <ol style="list-style-type: none"> 1. Система выдает соответствующее сообщение с указанием списка полей, которые не заполнены, но обязательны к заполнению 2. Переход к шагу 3 основного потока.

Таблица 4 – Спецификация варианта использования «Выполнить многокритериальный поиск документов»

Прецедент: Выполнить многокритериальный поиск документов
ID: 4
Краткое содержание: позволяет найти и выдать подборку документов в соответствии с запросом по критериям поиска
Главные актеры: Клиент
Второстепенные актеры: Нет
Предусловия: Клиент авторизован, загружена подсистема поиска документов
Основной поток: 1. Клиент вводит критерии поиска: номер дела (текстовое поле), статья закона (текстовое поле), дата (выбор диапазона дат), судья (текстовое поле), судебный участок (текстовое поле), регион (текстовое поле), любая комбинация приведенных критериев (с помощью «чек-бокса») 2. ИИС выполняет поиск документов в БД по выбранным критериям поиска (альтернативный поток I) 3. ИИС выводит на экран список документов, удовлетворяющих заданным критериям поиска – список в кратком виде [«Тип документа» от «Дата» по делу номер «Номер дела»; «Судебный участок» – «Вид дела»; Суть спора: «Статья»] 4. Опционально: Клиент выбирает любой документ из списка найденных и инициирует его просмотр – выполняется подчиненный сценарий 5
Постусловия: список документов сформирован в соответствии с заданными критериями поиска и выведен на экран
Альтернативные потоки: I: критерии не указаны: 1. Система выдает соответствующее сообщение 2. Переход к шагу 1 основного потока.

Таблица 5 – Спецификация варианта использования «Открыть документ»

Прецедент: Открыть документ
ID: 5
Краткое содержание: позволяет открыть содержимое найденного документа из базы данных ИСС
Главные актеры: Клиент
Второстепенные актеры: Нет
Предусловия: по результатам сценария 4 список документов сформирован в соответствии с критериями поиска и выведен на экран
Основной поток: 1. Клиент выбирает любой документ из списка найденных и инициирует его просмотр командой «Просмотр» 2. ИИС открывает специальный браузер rtf-файлов и загружает в него содержимое документа по ссылке, указанной как путь к файлу в базе данных для соответствующего документа
Постусловия: содержимое rtf-файла загружено

Приложение Б. Скрипт создания физической структуры базы данных ИСС

В листинге 1 приложения Б приведен скрипт создания физической структуры базы данных ИСС «Банк судебных решений» для СУБД MS Access 2007.

Листинг 1 – Скрипт создания физической структуры базы данных

```
ALTER TABLE [court] DROP CONSTRAINT [FK_court_region]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_court]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_decision_type]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_document_type]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_judge]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_law_article]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_person]
ALTER TABLE [document] DROP CONSTRAINT [FK_document_person_02]
ALTER TABLE [judge] DROP CONSTRAINT [FK_judge_court]
ALTER TABLE [judge] DROP CONSTRAINT [FK_judge_post]
ALTER TABLE [person] DROP CONSTRAINT [FK_judge_court]
ALTER TABLE [person] DROP CONSTRAINT [FK_judge_post]

DROP TABLE [court]
DROP TABLE [decision_type]
DROP TABLE [document]
DROP TABLE [document_type]
DROP TABLE [judge]
DROP TABLE [law_article]
DROP TABLE [person]
DROP TABLE [post]
DROP TABLE [region]

CREATE TABLE [court]
(
    [id] AUTOINCREMENT NOT NULL,
    [name] Text(100) NOT NULL,
    [address] Text(150) NOT NULL,
    [president] Text(60) NOT NULL,
    [www] Text(100) NULL,
    [phone] Text(20) NOT NULL,
    [location] Text(20) NULL,
    [region_id] Long NOT NULL
)

CREATE TABLE [decision_type]
(
    [id] AUTOINCREMENT NOT NULL,
    [name] Text(20) NOT NULL
)

CREATE TABLE [document]
(
    [id] AUTOINCREMENT NOT NULL,
    [decision_number] Text(30) NOT NULL,
    [decision_date] DateTime NOT NULL,
    [content_file_name] Text(150) NOT NULL,
    [decision_type_id] Long NOT NULL,
```

Продолжение листинга 1 приложения Б

```

    [document_type_id] Long NOT NULL,
    [court_id] Long NOT NULL,
    [judge_id] Long NOT NULL,
    [law_area_id] Long NOT NULL,
    [plaintiff_id] Long NULL,
    [defendant_id] Long NULL
)

CREATE TABLE [document_type]
(
    [id] AUTOINCREMENT NOT NULL,
    [name] Text(20) NOT NULL
)

CREATE TABLE [judge]
(
    [id] AUTOINCREMENT NOT NULL,
    [first_name] Text(20) NOT NULL,
    [second_name] Text(20) NOT NULL,
    [fur_name] Text(20) NULL,
    [post_id] Long NOT NULL,
    [court_id] Long NOT NULL
)

CREATE TABLE [law_article]
(
    [id] AUTOINCREMENT NOT NULL,
    [name] Text(255) NOT NULL
)

CREATE TABLE [person]
(
    [id] AUTOINCREMENT NOT NULL,
    [first_name] Text(20) NOT NULL,
    [second_name] Text(20) NOT NULL,
    [fur_name] Text(20) NULL,
    [sex] Text(4) NULL,
    [birthday] DateTime NULL
)

CREATE TABLE [post]
(
    [id] AUTOINCREMENT NOT NULL,
    [name] Text(60) NOT NULL
)

CREATE TABLE [region]
(
    [id] AUTOINCREMENT NOT NULL,
    [name] Text(50) NOT NULL
)

ALTER TABLE [court] ADD CONSTRAINT [PK_court]
PRIMARY KEY ([id])

ALTER TABLE [decision_type] ADD CONSTRAINT [PK_decision_type]
PRIMARY KEY ([id])

ALTER TABLE [document] ADD CONSTRAINT [PK_document]
PRIMARY KEY ([id])
```

Окончание листинга 1 приложения Б

```
CREATE INDEX [IXFK_document_person] ON [document] ([plaintiff_id] ASC)

CREATE INDEX [IXFK_document_person_02] ON [document] ([defendant_id] ASC)

ALTER TABLE [document_type] ADD CONSTRAINT [PK_document_type]
PRIMARY KEY ([id])

ALTER TABLE [judge] ADD CONSTRAINT [PK_judge]
PRIMARY KEY ([id])

ALTER TABLE [law_article] ADD CONSTRAINT [PK_low_article]
PRIMARY KEY ([id])

ALTER TABLE [person] ADD CONSTRAINT [PK_judge]
PRIMARY KEY ([id])

ALTER TABLE [post] ADD CONSTRAINT [PK_post]
PRIMARY KEY ([id])

ALTER TABLE [region] ADD CONSTRAINT [PK_region]
PRIMARY KEY ([id])

/* Create Foreign Key Constraints */

ALTER TABLE [court] ADD CONSTRAINT [FK_court_region]
FOREIGN KEY ([region_id]) REFERENCES [region] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_court]
FOREIGN KEY ([court_id]) REFERENCES [court] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_decision_type]
FOREIGN KEY ([decision_type_id]) REFERENCES [decision_type] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_document_type]
FOREIGN KEY ([document_type_id]) REFERENCES [document_type] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_judge]
FOREIGN KEY ([judge_id]) REFERENCES [judge] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_law_article]
FOREIGN KEY ([law_area_id]) REFERENCES [law_article] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_person]
FOREIGN KEY ([plaintiff_id]) REFERENCES [person] ([id])

ALTER TABLE [document] ADD CONSTRAINT [FK_document_person_02]
FOREIGN KEY ([defendant_id]) REFERENCES [person] ([id])

ALTER TABLE [judge] ADD CONSTRAINT [FK_judge_court]
FOREIGN KEY ([court_id]) REFERENCES [court] ([id])

ALTER TABLE [judge] ADD CONSTRAINT [FK_judge_post]
FOREIGN KEY ([post_id]) REFERENCES [post] ([id])

ALTER TABLE [person] ADD CONSTRAINT [FK_judge_court]
FOREIGN KEY ([birthday]) REFERENCES ( )

ALTER TABLE [person] ADD CONSTRAINT [FK_judge_post]
FOREIGN KEY ([sex]) REFERENCES ( )
```